

Scriptie

J-Layer auto-scaling

Naam: Rogier Schipper
Studentnummer: 1583842
Examinator: Peter van Rooijen
Opleiding: Informatica Deeltijd
Datum: 12-10-2014



Managementsamenvatting

Probleem

Jitscale heeft een applicatie ontwikkeld waarmee IT omgevingen beheerd worden, genaamd J-Layer. Met J-Layer is Jitscale in staat om instances/servers te starten en te stoppen (schalen) aan de hand van de behoeften op het huidige moment. Daarmee wordt veel downtime voorkomen, maar niet altijd. Bij een snelle stijging van de behoeften is het namelijk niet mogelijk om nieuwe instances gestart te hebben op het moment dat ze nodig zijn, wat zorgt voor downtime en de daarbij horende kosten. Daarbij is het met de huidige manier van schalen nodig om instances langer door te laten draaien, totdat zeker is dat de behoeften gedaald is en het zeker is dat ze niet meer nodig zijn. Dit zorgt voor overcapaciteit en overbodige kosten. Er moet dus een oplossing gerealiseerd worden waarmee downtime voorkomen wordt en kosten verminderd worden.

Advies

Voor het oplossen van het probleem is ervoor gekozen om aan de hand van time serie data van de benodigde behoeften uit het verleden te voorspellen wat het benodigd aantal instances zijn in de toekomst. Het voorspellen van de het benodigde aantal instance wordt gedaan in een applicatie die los van J-Layer functioneerd waar J-Layer via een API mee communiceert.

Argumentatie

Door te voorspellen hoeveel instances er in de toekomst nodig zijn wordt downtime voorkomen, doordat eerder bekend is dat de behoeften gaat stijgen, waardoor er eerder bijgeschaald wordt. Door de instances eerder bij te schalen is er meer starttijd, waardoor ze gestart zijn op het moment dat voorzien moet worden in de behoeften. Daarbij is er door te voorspellen ook bekend wanneer de behoeften daalt waarmee het mogelijk is om instances eerder te stoppen.

Er is gekozen voor het ontwikkelen van een applicatie naast J-Layer voor twee redenen:

1. Hiermee wordt de oplossing schaalbaar zodat er extra computer resources ingezet kunnen worden voor het doen van voorspellingen, zonder dat hiervoor bij de ontwikkeling van J-Layer rekening mee gehouden hoeft te worden.
2. De applicatie zal, naast J-Layer, voorspellingen kunnen gaan doen voor andere doeleinden binnen Jitscale.

Consequentie

Voor het voorspellen van het aantal instances is het van belang dat er time serie data uit het verleden beschikbaar is. Zonder deze data kan er geen voorspelling gedaan worden. Daarbij is het belangrijk dat voorspellingen die gedaan worden kloppen, zodat er niet instances gestart worden die niet nodig zijn of geen instances gestart worden als ze wel nodig zijn.

Inhoudsopgave

1. Inleiding	1
2. Jitscale.....	1
3. Doelstellingen	1
4. Opdracht.....	2
5. Onderzoeksvragen.....	3
5.1. Hoofdvraag	3
5.2. Deelvragen	3
6. Wat is J-Layer?	3
6.1. Onderzoeksmethode.....	3
6.2. J-Layer.....	3
6.3. Huidige manier van schalen.....	5
7. Veranderingsonderzoek	6
7.1. Onderzoeksmethoden.....	6
7.2. Huidige situatie	7
7.3. Problemen en doelstellingen.....	7
7.4. Veranderingsanalyse	7
7.4.1. SWOT Analyse	8
7.4.2. Effect cause effect analyse	8
7.4.3. Veranderingsbehoeften.....	9
7.4.4. Veranderingsmaatregelen.....	10
7.5. Conclusie	11
8. Voorspellen van benodigde instances.....	11
8.1. Onderzoeksmethoden.....	11
8.2. Voorspellingsmethode eisen	12
8.2.1. Voorspelling binnen 15 minuten	12
8.2.2. Voorspellen in de toekomst met time serie data	12
8.2.3. Voorspellen aan de hand van meerdere time series	12
8.2.4. Trends herkennen.....	12
8.2.5. Seasonality herkennen	13
8.2.6. Outliers detectie.....	13

8.2.7.	Under- en overfitting voorkomen	14
8.3.	Voorspellingsmethode selectie.....	14
8.3.1.	Voorspellingsmethodes.....	14
8.4.	Cross reference analyse	15
8.4.1.	Voorspelling binnen 15 minuten	15
8.4.2.	Noise en outliers kunnen detecteren en negeren	16
8.4.3.	Trends herkennen.....	16
8.4.4.	Under- en overfitting voorkomen	16
8.4.5.	Meerdere seasonalities herkennen:	17
8.4.6.	Voorspellen op basis van meerdere time series.....	17
8.4.7.	Resultaat	18
8.5.	Conclusie	18
9.	Functioneel onderzoek.....	19
9.1.	Onderzoeksmethoden.....	19
9.2.	Functionele en niet-functionele eisen	19
9.3.	Werking van clustered linear regression	20
9.4.	K-Fold cross validation.....	23
9.5.	Outlier detection.....	24
9.6.	Functionele oplossing	26
9.6.1.	Use cases.....	26
9.6.2.	Business rules	27
9.6.3.	Nieuwe manier van schalen	28
9.7.	Conclusie	30
10.	Technischonderzoek	31
10.1.	Onderzoeksmethode	31
10.2.	Klassendiagram.....	31
10.3.	Koppeling tussen J-Layer en J-Predictor	34
10.3.1.	JSON	35
10.4.	Ruby on Rails	35
10.5.	RServe	36
10.6.	Database.....	36
10.6.1.	Database engine	37

10.7.	Strategy pattern.....	37
10.8.	Beveiliging.....	37
10.9.	Webserver.....	37
10.10.	Prototype.....	38
10.11.	Conclusie	40
11.	Implementatie.....	41
11.1.	Onderzoeksmethode	41
11.2.	Server setup.....	41
11.3.	Backup	41
11.4.	Monitoring	41
11.5.	IT infrastructuur	41
12.	Conclusie	43
13.	Aanbevelingen.....	45
14.	Evaluatie	45
15.	Bibliografie	46
Bijlage 1 – Plan van aanpak.....		
Bijlage 2 – Evaluatie		
Bijlage 3 – Functioneel ontwerp		
Bijlage 4 – Technisch ontwerp		
Bijlage 5 – Implementatieplan		
Bijlage 6 – Voorspellingsmethodes		

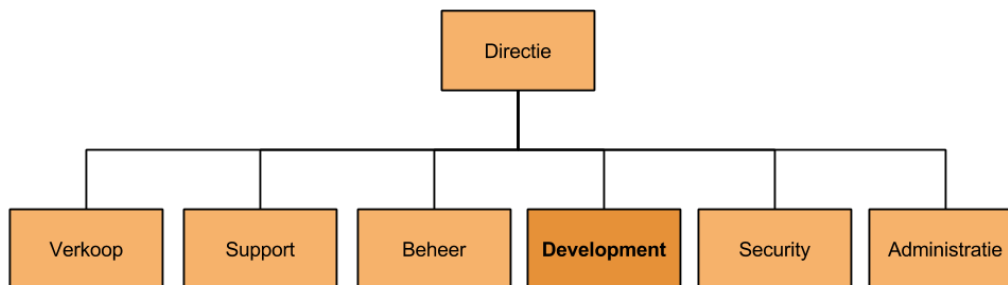
1. Inleiding

Voor Jitscale moet een oplossing gezocht en ontwikkeld worden waarmee downtime voorkomen wordt en kosten bespaard worden. In deze scriptie wordt beschreven hoe dit onderzoek verlopen is, wat de resultaten zijn en hoe het ontwikkelde prototype werkt. Als eerste wordt uitleg gegeven over de context van de opdracht, de doelstelling en de opdracht met bijbehorende deelvragen. Daarna worden de onderzoeksresultaten beschreven en toegelicht waaruit zal blijken hoe downtime voorkomen wordt, kosten bespaard worden en hoe de oplossing werkt. Als laatste worden in een conclusie de onderzoeksresultaten en de antwoorden op de deelvragen en de hoofdvraag beschreven.

2. Jitscale

Dit project heeft plaats gevonden bij Jitscale. Jitscale is een hosting provider die gespecialiseerd is in het hosten van cloud infrastructuur oplossingen. Daarbij ligt de focus van Jitscale op 100% availability, 100% performance en 100% security. Voor het realiseren van 100% availability, performance en security heeft Jitscale de J-Cloud en J-Layer ontwikkeld. J-Cloud is de cloud oplossing van Jitscale, en de enige ISO gecertificeerde cloud oplossing in Nederland. En met J-Layer heeft Jitscale het mogelijk gemaakt om vanuit één simpel dashboard (Jitscale, z.d.) een IT infrastructuur gemakkelijk bij verschillende cloud providers te beheren.

Mijn werkzaamheden bij Jitscale zijn het (door)ontwikkelen van software oplossingen voor de primaire werkzaamheden van Jitscale, zoals J-Layer. De ontwikkeling binnen Jitscale wordt gedaan op de development afdeling. afbeelding 2.1 toont de bedrijfsstructuur van Jitscale en de Development afdeling.



Afbeelding 2.1 - Development afdeling binnen Jitscale

3. Doelstellingen

Het bedenken en het ontwikkelen van een oplossing die binnen J-Layer geïmplementeerd kan worden waarmee gezorgd wordt dat er op elk moment precies genoeg instances zijn om te voorzien in de behoeften zodat de IT infrastructuur 100% beschikbaar blijft en zo min mogelijk instances nodig zijn zodat de kosten zo laag mogelijk blijven. (Appendix, z.d)

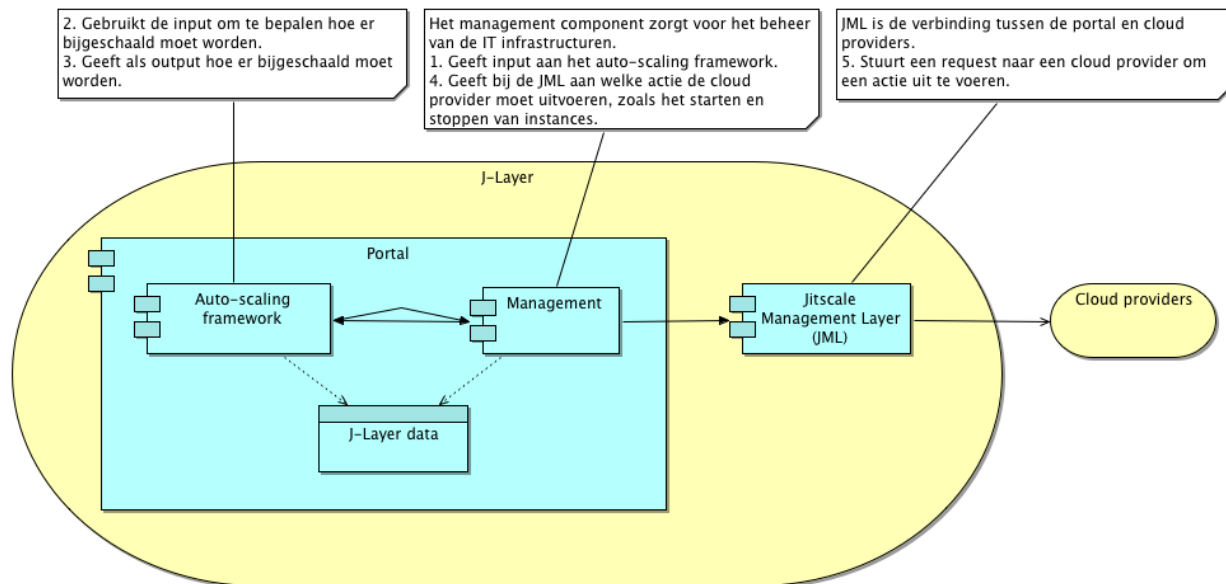
4. Opdracht

Het ontwikkelen van een pluggable auto-scaling framework binnen J-Layer waarmee aan de hand van voorspellingsalgoritmes voorspeld kan worden hoeveel resources er nodig gaan zijn. Door te voorspellen hoeveel resources nodig zullen zijn, kan 100% availability gegarandeerd worden doordat hiermee instances precies op het juiste moment gestart kunnen worden, net voordat ze nodig zijn. (SDART, z.d)

Door gebruik te maken van voorspellingsalgoritmes en aan de hand van data uit het verleden kan er berekend word hoeveel resources er nodig gaan zijn. De data kunnen bijvoorbeeld het uurlijks aantal HTTP requests op een bepaald tijdstip over de afgelopen twee maanden zijn. Een voorspellingsalgoritme kan aan de hand van de data voorspellen hoeveel HTTP requests er morgen op het zelfde tijdstip zijn, waarmee bepaald kan worden hoeveel resources/instances er nodig zullen zijn. (Wikipedia, Plug-in (computing), 2014)

Het framework moet pluggable zijn zodat voorspellingalgoritmes onafhankelijk van elkaar als plugins toegevoegd kunnen worden. Dit zorgt voor flexibiliteit doordat bij het toevoegen van algoritmes geen rekening gehouden hoeft te worden met de andere algoritmes.

In afbeelding 4.1 is te zien hoe het auto-scaling framework geïmplementeerd kan worden binnen J-Layer.



Afbeelding 4.1 - Implementatie binnen J-Layer

5. Onderzoeksvragen

5.1. Hoofdvraag

Op welke manier kan er binnen J-Layer voor gezorgd worden dat er op elk moment precies genoeg instances gestart zijn en het kosten besparend is?

5.2. Deelvragen

1. Wat is J-Layer?
2. Op welke wijze kan ervoor gezorgd worden dat er precies genoeg instances gestart zijn met J-Layer?
 - a. Hoe wordt er nu geprobeerd om precies genoeg instances te starten?
 - b. Hoe komt het dat het nu niet lukt om precies genoeg instances te starten?
3. Op welke wijze kan ervoor gezorgd worden dat er kosten bespaard worden met J-Layer?
 - a. Hoe wordt er nu voor gezorgd dat er kosten bespaard worden?
4. Wat is de beste oplossing om te implementeren binnen J-Layer?
 - a. Wat zijn de functionele en niet-functionele eisen?
 - b. Wat zijn de constraints?
 - c. Hoe kan de oplossing geïmplementeerd worden binnen J-Layer?

Om zeker te zijn dat de opdracht van de opdrachtgever in hoofdstuk 4 de beste oplossing is, is eerst onderzocht welke mogelijke oplossingen er zijn en wat de beste oplossing is voor het probleem. Dit is gedaan met deelvragen 1 en 2. Met deelvraag 1 is onderzocht wat J-Layer is en hoe het werkt, zodat er een duidelijk beeld bestaat van J-Layer voor de implementatie van de oplossing. Vervolgens is met deelvraag 2 en 3 onderzocht welke maatregel het meest bijdraagt aan het precies genoeg starten van instances en het besparen van kosten. Nadat bekend is geworden welke maatregel het meest bijdraagt is met deelvraag 4 bepaald wat de beste oplossing is voor het implementeren van de maatregel binnen J-Layer.

6. Wat is J-Layer?

Om voor deelvraag 1 duidelijk te krijgen wat J-Layer, wordt in dit hoofdstuk beschreven wat het doel en de werking van J-Layer is. Het is van belang dat de werking van J-Layer duidelijk is, zodat bij de implementatie van de oplossing bekend is hoe dit kan gaan werken binnen J-Layer. Als eerste wordt beschreven wat het doel is van J-Layer en wat de mogelijkheden zijn. Daarna wordt het proces van de huidige manier van instances stoppen en starten beschreven.

6.1. Onderzoeksmethode

Om in kaart te brengen wat J-Layer is zijn de developers van J-Layer geïnterviewd en is de documentatie van J-Layer geraadpleegd.

6.2. J-Layer

J-Layer is een applicatie die ontwikkeld is door Jitscale voor het beheren van IT-infrastructuren waarbij gebruik wordt gemaakt van cloud computing. Cloud computing is een on-demand service model waarmee IT diensten geleverd worden. Dit zijn IT diensten zoals servers, storage, netwerken en applicaties. Deze diensten kunnen op ieder moment van de dag binnen een paar minuten gestart en gestopt worden met minimale beheerlast waarbij alleen betaald hoeft te worden zolang er gebruik wordt gemaakt van de dienst. (wiki, z.d), (Apprenda, z.d), (Wikipedia, IT-infrastructuur, 2014)

Een server in de cloud wordt ook wel een instance of VPS genoemd. Zo'n instance of VPS is een virtuele server welke gehost en verhuurd wordt door een cloud provider en waarbij de huurder zelf helemaal verantwoordelijk is voor het beheer van de virtuele server. Zo'n instance of VPS heeft een vaste hoeveelheid resources tot zijn beschikking van één of meer onderliggende servers. De onderliggende

server deelt zijn resources dus met één of meerdere instances/VPS. (Wikipedia, Virtual private server, 2014), (Rackspace, z.d), (Hogben, 2009)

Het doel van Jitscale is het 100% beschikbaar houden van de IT-infrastructuur, waarbij cloud computing een grote bijdrage levert. Dit komt omdat het met cloud computing mogelijk is om op ieder moment van de dag een IT dienst, zoals een instance te starten of te stoppen. Hiermee wordt het voor Jitscale mogelijk om op elk moment van de dag te voorzien in de behoeften van de klant. De behoefte kan in zo'n geval zijn dat een website veel bezoekers heeft waardoor het niet mogelijk is om met één instance in die behoefte te voorzien. In dat geval kan er met cloud computing een extra instance gestart worden, die helpt met het afhandelen van de behoefte.

Een sterk punt van J-Layer is dat het mogelijk om de IT-diensten van verschillende cloud providers te beheren via de API's van deze cloud providers. Hierdoor is het met J-Layer mogelijk om IT-infrastructuren te creëren en te beheren op basis van de cloud provider IT-diensten. Zo'n IT-infrastructuur kan bestaan uit een aantal instances in een eigen netwerkgroep, zoals een database en applicatie instance. Dit is te vergelijken met een IT-infrastructuur waarbij fysieke servers gebruikt worden en de servers via een eigen netwerk met elkaar communiceren maar dan in de cloud. In afbeelding 6.1 is een IT-infrastructuur te zien van twee Wordpress PHP instances die samen de behoeften van een Wordpress website afhandelen, waarbij de bezoekers verdeeld worden over de twee instances. Daarbij is ook te zien wat voor type(specs) instances het zijn, wat aangeeft hoeveel resources de instance heeft. In dit geval gaat het om twee m1.xlarge instances. Op het moment dat er meer of minder instances nodig zijn kan handmatig of met de knop 'scale up' een extra instance gestart worden, die identiek is aan de andere twee instances, of met de knop 'scale down' een instance gestopt worden.

Wordpress PHP server acting as web application server

settings ▾

monitoring deployments scaling behavior

Name	Status	Specs	Architecture	Zone	Performance	Time	DNS	Monitored	
jtcprprodapp1	running	m1.xlarge	Linux / 64-bit	eu-west-1b	003	2013-12-17 13:19 CET	✓	✓	actions ▾
jtcprprodapp2	running	m1.xlarge	Linux / 64-bit	eu-west-1b	001	2014-01-21 09:23 CET	✓	✓	actions ▾

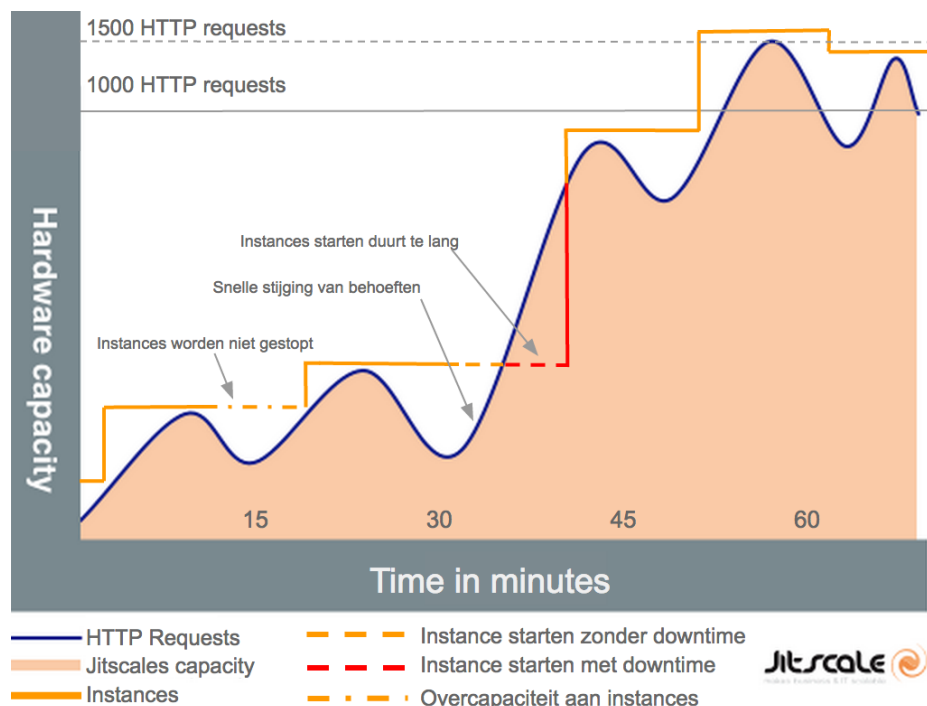
show service information

scale down deploy configuration scale up

Afbeelding 6.1 - J-Layer

Doordat J-Layer gebruik maakt van de API's van de cloud providers heeft dit als voordeel dat de IT-diensten te automatiseren zijn. In J-Layer is dit gedaan door het automatische starten en stoppen (schalen) van een instance wanneer de behoeften meer worden dan de ingestelde threshold. Met de threshold wordt aangegeven wanneer er een instance gestart moet worden, bijvoorbeeld bij 1500 HTTP requests.

In afbeelding 6.2 is te zien hoe instances bij schalen op het moment dat de HTTP requests (behoeften) meer of minder worden waarmee vaak voldaan wordt aan de behoeften, behalve als de load erg snel steigt. (Wikipedia, Cloud API, 2013), (BusinessDictionary, z.d)



Afbeelding 6.2- Jitscale infrastructuur

6.3. Huidige manier van schalen

Om duidelijk inzicht te krijgen in hoe J-Layer op dit moment instances start en stopt is in afbeelding 6.3 een procesmodel van het huidige schaalproces te zien is. In het model is te zien dat er elke 3 minuten aan de hand van de load van een instance, bepaald wordt of, en hoeveel instances er gestart moeten worden. Bij deze manier van schalen zijn de volgende vijf instellingen van belang:

- Threshold
- Breach and sample periode
- Cool down periode (voor het starten van instances)
- Warm up periode (voor het stoppen van instances)

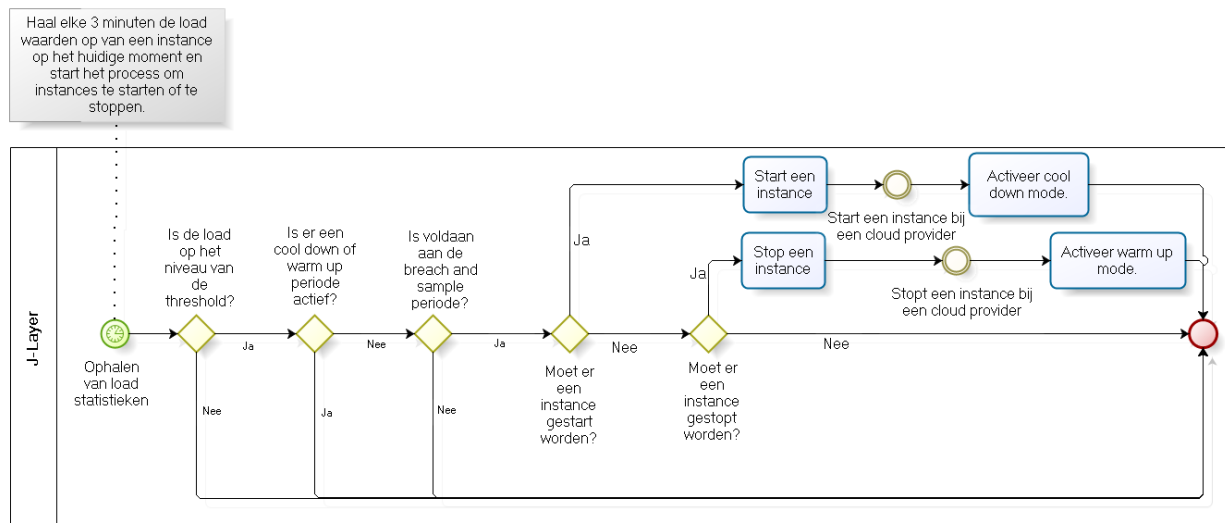
Met de threshold wordt bepaald of de load, waaruit af te leiden is hoe druk een instance het heeft, op een bepaald niveau is waardoor er instances gestart of gestopt moeten worden.

Met de breach and sample periode wordt aangegeven dat de load voor een bepaalde tijd, achtereenvolgens, op het niveau van de threshold moet zijn voordat instances gestart of gestopt wordt.

Met de cool down periode wordt aangegeven dat er voor een bepaalde tijd, na het starten van instances geen instances gestart mogen worden. Dit zorgt ervoor dat niet te veel instances gestart worden doordat de instances die al aan het starten zijn de tijd hebben om de load naar beneden te brengen, waardoor door de load niet meer op het niveau is van de threshold.

De warm up periode is het omgekeerde van de cool down periode en geldt voor het stoppen van instances. Met de warm up periode wordt er dus voor gezorgd dat er niet te veel instances gestopt worden als er al instances aan het stoppen zijn. Dit is belangrijk doordat hiermee voorkomen wordt dat er te veel gestopt worden waardoor de load weer zal stijgen en er weer instances gestart moeten worden.

Het nadeel bij deze manier van schalen is dat er niet precies bekend is hoeveel instances er gestart en gestopt moeten worden. Hierdoor is het nodig om met de “breach time and sample time” voor een bepaalde periode te wachten voordat er instances gestart of gestopt mogen worden. Het zelfde geldt voor de cool down en warm up periode waarbij afgewacht wordt of de load verder zal stijgen of dalen voordat er instances gestart of gestopt worden. Dit zorgt ervoor dat het voorkomt dat instances te laat gestart zijn waardoor er downtime ontstaat, omdat niet optijd voorzien kan worden in de behoeften. En dit zorgt er ook voor dat instances langer zullen blijven draaien terwijl ze mogelijk al gestopt hadden kunnen worden, waardoor er overcapaciteit ontstaat. Dit is te zien in afbeelding 6.2.



Afbeelding 6.3 - Huidige manier van schalen

7. Veranderingsonderzoek

In dit hoofdstuk wordt antwoord gegeven op deelvraag 2 en 3, en bijbehorende sub vragen. Als eerste wordt uitleg geven over de gebruikte onderzoeksmethoden en daarna wordt aan de hand van de onderzoeksmethodes, antwoord gegeven op de subvragen 2a, 2b, 3a en deelvragen 2 en 3.

7.1. Onderzoeksmethoden

Om in kaart te brengen wat de (kern) problemen, doelstellingen en veranderingsbehoeften zijn, zijn de volgende onderzoeksmethodes gebruikt voor het analyseren van de huidige situatie en het bepalen wat de beste veranderingsmaatregel zal zijn:

- **Interview:** Als eerste is aan de hand van een interview met Jitscale informatie vergaard over de huidige situatie en vastgesteld wat de problemen en doelstellingen zijn.
- **SWOT analyse:** Na het interview zijn met een SWOT analyse sterkte, zwakte, kansen en bedreigingen van de huidige situatie in kaart gebracht.
- **Effect cause effect analyse:** Met de effect cause effect analyse is vastgesteld wat de kernproblemen zijn van de huidige situatie. Het vaststellen van de kernproblemen is gedaan door het analyseren van de doelstellingen en problemen, die naar voren zijn gekomen tijdens het interview.
- **Veranderingsanalyse:** Met de veranderingsanalyse is vastgesteld welke veranderingsmaatregel het meest bijdraagt aan het oplossen van de problemen van de huidige situatie en het meest voorziet in het realiseren van de doelstellingen van Jitscale.

7.2. Huidige situatie

Om antwoord te geven op deelvraag 2 en 3 was het van belang om de huidige situatie in kaart te brengen. Dit is gedaan met subvraag 2a, 2b en 3a. De huidige situatie is van belang zodat bekend is waar de maatregel op aan moet sluiten en voorkomen wordt dat er een maatregel bedacht wordt die al gerealiseerd is.

Deelvraag 2a: Hoe wordt nu geprobeerd om precies genoeg instances te starten?

Antwoord: De manier waarop nu precies genoeg instances worden gestart, is beschreven in hoofdstuk 6.3 - Huidige manier van schalen, door te kijken naar de behoeften op het huidige moment en op dat moment te bepalen of er een instance gestart of gestopt moet worden.

Deelvraag 2b: Hoe komt het dat het nu niet lukt om precies genoeg instances te starten?

Antwoord: Doordat er met de huidige manier van schalen pas een instance gestart wordt op het moment dat de behoeften stijgen. Hierdoor komt het voor dat de instance niet op tijd de behoeften kan opvangen, doordat de instance nog aan het starten is. Tijdens dit proces kunnen andere instances overbelast raken.

Deelvraag 3a: Hoe wordt er nu voor gezorgd dat er kosten bespaard worden?

Antwoord: Door met J-Layer instances te starten op het moment dat ze nodig zijn en instances te stoppen op het moment dat ze niet nodig zijn, waarbij gekeken wordt naar de behoeften op dat moment.

7.3. Problemen en doelstellingen

Uit een interview met Jitscale over de problemen en doelstellingen van Jitscale zijn de volgende doelstellingen en problemen naar voren gekomen.

Doelstellingen:

1. Downtime voorkomen.
2. Best mogelijke performance kunnen garanderen.
3. Kosten verminderen.

Problemen:

1. Downtime bij snelle stijging van benodigde behoeften.
2. Performance verlies door gebrek aan resources.
3. Onnodige kosten door downtime, zoals:
 - a. Kosten voor het niet kunnen nakomen van SLA afspraken omtrent de tijdsduur van downtime.
 - b. Kosten voor de klant doordat hun IT infrastructuur down is.
4. Onnodige kosten door overcapaciteit aan resources.
5. Kosten zijn hoger dan de afgesproken norm in de SLA.
6. Er is meer downtime dan de afgesproken norm in de SLA.
7. Performance voldoet niet aan de afgesproken norm in de SLA.

7.4. Veranderingsanalyse

Voor het vinden van een antwoord op deelvraag 2 en 3 is een veranderingsanalyse uitgevoerd. Deze analyse bestaat uit een aantal deelonderzoeken, die samen resulteren in een veranderingsmaatregel. De veranderingsmaatregel is de maatregel die het meest bijdraagt aan het oplossen van het probleem en die gebruikt is voor het beantwoorden van deelvraag 4.

7.4.1.SWOT Analyse

In tabel 7.1 is de SWOT analyse te zien die is uitgevoerd met de interview resultaten.

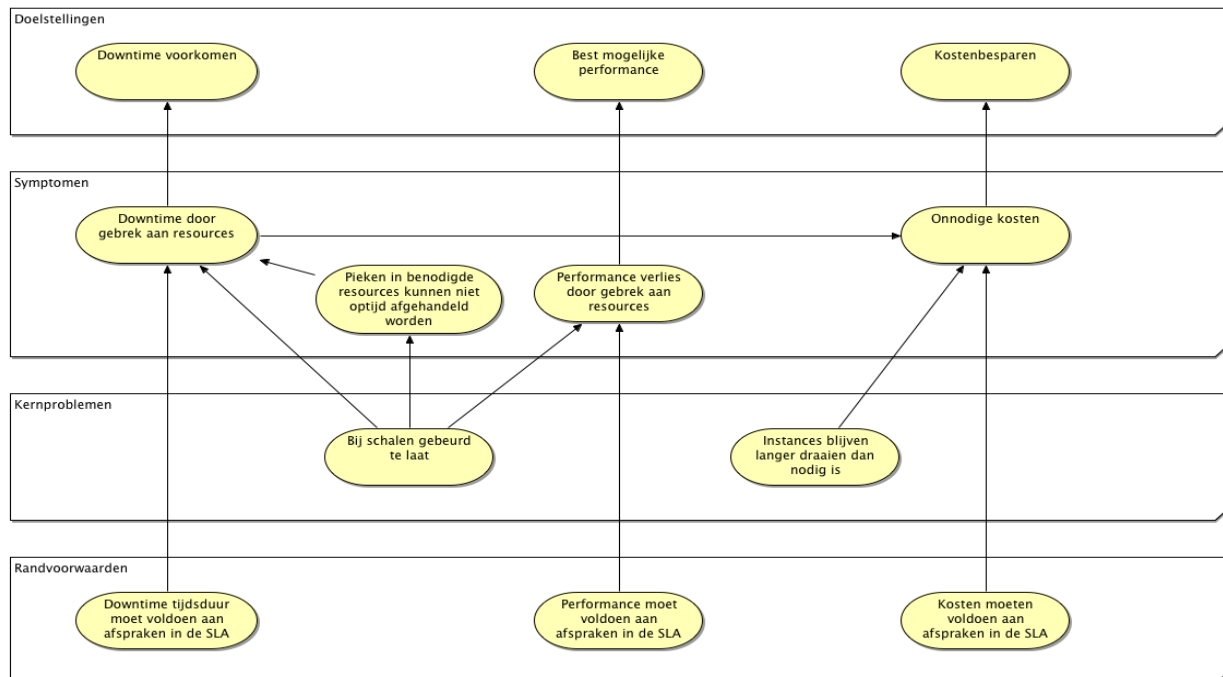
Sterkte	Zwakte
<ul style="list-style-type: none"> - Bij schalen gaat automatisch. - Starten en configureren van instance gaat snel. - Beheer van cloud diensten is gemakkelijk in J-Layer. - Met J-Layer is het gebruik van meerdere cloud providers mogelijk. - Downtime wordt in veel gevallen voorkomen door bij te schalen als de capaciteitsbehoeften meer worden. 	<ul style="list-style-type: none"> - Downtime is nog steeds mogelijk bij een snelle stijging van de behoeften. - Onnodige kosten door downtime. - Performance verlies door te laat bij schalen.
Kansen	Bedreigingen
<ul style="list-style-type: none"> - Downtime voorkomen. - Kosten besparen door minder overcapaciteit. - Kosten besparen door minder downtime. - Performance verbeteren door voldoende resources 	<ul style="list-style-type: none"> - Concurrenten kunnen meer uptime garanderen. - Klanten verliezen doordat ze meer uptime willen dan we kunnen leveren.

Tabel 7.1 - SWOT analyse

Uit de SWOT analyse is af te leiden dat het automatisch bij schalen een sterk punt is maar dat er met deze manier nog steeds kans is op downtime wat voor onnodige kosten en performance verlies zorgt. Doordat met J-Layer het beheer van cloud diensten een stuk gemakkelijker is en het gebruik van cloud diensten geautomatiseerd kan worden, creëert dit kansen voor het nog verder terug brengen van downtime, kosten en performance verlies. Hoe deze kansen gerealiseerd kunnen worden is onderzocht met een veranderingsbehoefte analyse in paragraaf 7.4.3. Uit de SWOT analyse blijkt ook dat als concurrenten meer uptime kunnen garanderen, dit kan leiden tot mogelijk verlies van klanten.

7.4.2.Effect cause effect analyse

In afbeelding 7.1 is de effect cause effect analyse (ECEA) te zien waarin de doelstellingen en de problemen verwerkt zijn. Door de doelstellingen en de problemen die met elkaar verband houden te koppelen in een ECEA wordt duidelijk wat de kern problemen zijn, die opgelost moet worden.



Afbeelding 7.1 - Effect cause effect analyse

Uit de effect cause effect analyse komt duidelijk naar voren dat de kernproblemen zijn dat instances te laat bij schalen en te lang blijven draaien (overcapaciteit). Dit bleek ook uit het interview met Jitscale dus hierbij dient de ECEA als onderbouwing van deze feiten.

7.4.3. Veranderingsbehoeften

Na het analyseren van de effect cause effect diagram en bespreking met Jitscale zijn vijf veranderingsbehoeften naar voren gekomen. Deze veranderingsbehoeften zijn verwerkt in de veranderingsbehoefte analyse waarmee bepaald is welke veranderingsbehoeften het meest bijdragen aan het oplossen van de problemen.

Veranderingsbehoeften:

1. Instaat zijn te bepalen hoeveel instances op een bepaald tijdstip nodig zijn.
2. Overcapaciteit verminderen.
3. Downtime voorkomen.
4. Kosten verminderen.
5. Performance verlies voorkomen.

In tabel 7.2 is de veranderingsbehoefte analyse te zien. In de tabel wordt vertikaal met kruisen aangegeven welke veranderingsbehoeften bijdragen aan het oplossen van de problemen en het realiseren van de doelstellingen. Vervolgens wordt aan de hand van het aantal kruisen, dat een veranderingsbehoefte vertikaal heeft, en in overleg met Jitscale bepaald wat de prioriteit is van een veranderingsbehoefte. Hierbij is 1 de hoogste prioriteit.

Veranderingsbehoeften						
V1	Instaat zijn te bepalen hoeveel instances er op een bepaald tijdstip nodig zijn.					
V2	Overcapaciteit verminderen.					
V3	Downtime voorkomen.					
V4	Kosten verminderen.					
V5	Performance verlies voorkomen.					
Problemen		V5	V4	V3	V2	V1
P1	Downtime bij snelle stijging van benodigde behoeften.			X		X
P2	Performance verlies door gebrek aan resources.	X		X		X
P3	Onnodige kosten door downtime.		X	X		X
P4	Onnodige kosten door overcapaciteit aan resources.		X		X	X
P5	Kosten zijn hoger dan de afgesproken norm in de SLA.		X	X	X	X
P6	Er is meer downtime dan de afgesproken norm in de SLA.			X		X
P7	Performance voldoet niet aan de afgesproken norm in de SLA.	X				X
Doelstellingen						
D1	100% availability kunnen garanderen.	X	X	X		X
D2	Kosten lager houden dan de afgesproken norm in de SLA.	X	X	X	X	X
D3	Performance voldoet aan de afgesproken norm in de SLA.	X	X	X		X
Prioriteit veranderingsbehoeften		2	2	1	3	1

Tabel 7.2 - Veranderingsbehoefte analyse

Uit de veranderingsbehoefte analyse blijkt dat twee veranderingsbehoeften het meest bijdragen aan het oplossen van de problemen en het realiseren van de doelstellingen:

V1: Instaat zijn te bepalen hoeveel instances er op een bepaald tijdstip nodig zijn.

V3: Downtime voorkomen.

De reden hiervoor is dat V1 en V3 ook bijdragen aan de realisatie van andere veranderingsbehoeften. Zo zorgt V1 ook voor:

- Het verminderen van overcapaciteit (V2) doordat instances eerder gestopt kunnen worden.
- Het voorkomen van performance verlies doordat er precies genoeg instances gestart kunnen worden.

En V3 zorgt voor:

- het verminderen van kosten als gevolg van downtime.

7.4.4. Veranderingsmaatregelen

Voor het realiseren van de veranderingsbehoeften zijn maatregelen nodig. Aan de hand van de veranderingsbehoeften zijn drie maatregelen bedacht.

1. Benodigde instances voorspellen, zodat eerder bekend is hoeveel instances er nodig zijn en ze precies op tijd gestart of gestopt kunnen worden.
2. Meer instances draaien dan nodig is, zodat downtime voorkomen wordt omdat er genoeg instances draaien.
3. Instance configuratie versnellen, zodat het starten van een instance sneller gaat.

Om duidelijk te krijgen welke maatregel voorziet in de meeste en belangrijkste veranderingsbehoeften is er een veranderingsmaatregel analyse gedaan, welke te zien is in tabel 7.3.

	Veranderingsbehoeften		Maatregelen	
V1	Instaat zijn te bepalen hoeveel instances er op een bepaald tijdstip nodig zijn	M1	Benodigde instances voorspellen.	V1, V2, V3, V4, V5
V2	Overcapaciteit verminderen.	M2	Meer instances draaien dan nodig is.	V2, V3, V5
V3	Downtime verminderen.	M3	Instance configuratie versnellen.	V3, V4, V5
V4	Kosten verminderen.			
V5	Performance verlies voorkomen.			

Tabel 7.3 - Veranderingsmaatregel analyse

De veranderingsmaatregel tabel toont de veranderingsbehoeften, drie maatregelen en de veranderingsbehoeften, die de maatregel realiseert. Uit de veranderingsmaatregel analyse blijkt dat M1 "Benodigde instances voorspellen" voorziet in alle veranderingsbehoeften en dat M1 ook de veranderingsbehoefte met de hoogste prioriteit, zoals bepaald is in veranderingsbehoeften analyse, realiseert.

7.5. Conclusie

Na het doen van de SWOT, effect cause effect analyse, veranderingsbehoeften en veranderingsmaatregel analyse blijkt dat het automatisch bij schalen een sterk punt is van J-Layer en dat wanneer er voorspeld gaat worden hoeveel instances nodig zijn op een bepaald tijdstip in de toekomst, downtime helemaal voorkomen kan worden en kosten verminderd kunnen worden.

Met het resultaat uit de veranderingsmaatregel analyse zijn deelvraag 2 en 3 beantwoord:

Deelvraag 2: Op welke wijze kan ervoor gezorgd worden dat er precies genoeg instances?

Deelvraag 3: Op welke wijze kan ervoor gezorgd worden dat er kosten bespaard worden met J-Layer?

Antwoord: Door te voorspellen hoeveel instances er nodig zijn op een bepaald tijdstip. Daardoor kunnen instances eerder gestart en gestopt worden, omdat met zekerheid gezegd kan worden of een instance wel of niet nodig is.

Voor de veranderingsmaatregel is een methode nodig voor het voorspellen van het aantal instances. Hieruit komen twee nieuwe subvragen voort die vallen onder deelvraag 4:

Deelvraag 4d: Aan welke eisen moet de voorspellingsmethode voldoen?

Deelvraag 4e: Welke voorspellingsmethode voldoet aan deze eisen?

Het onderzoek van deze sub vragen wordt in het volgende hoofdstuk beschreven, en daarna wordt beschreven hoe deelvraag 4 en sub vraag 4a, 4b en 4c onderzocht zijn.

8. Voorspellen van benodigde instances

Om te kunnen bepalen hoeveel instances op een bepaald tijdstip in de toekomst nodig zijn is het nodig om dit te voorspellen. Voor het doen van voorspellingen zijn een reeks voorspellingsmethodes beschikbaar, die voorspellingen kunnen doen voor een reeks aan verschillende situaties. Voor het voorspellen van een instance moet een voorspelling in de tijd gedaan worden wat betekent dat dit gedaan moet worden aan de hand van time serie data. Door het voorspellen van de benodigde instances is het mogelijk om verschillende time serie data te gebruiken, zolang er uit af te leiden is hoe druk de instances het hebben gehad op een bepaald tijdstip.

8.1. Onderzoeksmethoden

Voor de selectie van een voorspellingsmethode zijn er, zoals beschreven is in de conclusie van hoofdstuk 7 twee nieuwe subvragen: 4d en 4e. Als eerste is een antwoord gezocht op deelvraag 4d waarbij een bronnenonderzoek gedaan is naar de verschillende voorspellingsmethodes, en gekeken is aan welke eisen deze moeten voldoen voor het doen van een accurate voorspelling. Daarbij is ook een interview gedaan bij Jitscale om duidelijk te krijgen of Jitscale eisen stelt aan de voorspellingsmethode. De eisen waaraan de voorspellingsmethode moet voldoen zijn hieronder beschreven.

Vervolgens wordt subvraag 4e beantwoord, waarbij geselecteerd is welke voorspellingsmethode het meeste aan de eisen voldoet. Voor het beantwoorden van de vraag was het nodig om eerst te bepalen welke voorspellingsmethodes er zijn en welke de meeste potentie hebben om te gebruiken voor het voorspellen van het aantal instances. Dit is gedaan door het doen van een bronnenonderzoek. De beschrijving van de voorspellingsmethode die gebruikt zullen worden bij de selectie is te vinden in Bijlage 6 – Voorspellingsmethodes. Daarna is een cross reference analyse gedaan om te bepalen welke van de voorspellingsmethodes het beste voldoet aan de gestelde eisen.

8.2. Voorspellingsmethode eisen

Uit het bronnenonderzoek en het interview met Jitscale is gebleken dat de voorspellingsmethode aan de volgende eisen moet voldoen:

1. Voorspelling kunnen doen binnen 15 minuten.
2. Voorspellen in de toekomst met time serie data.
3. Voorspellen aan de hand van meerdere time series.
4. Trends herkennen.
5. Seasonality herkennen.
6. Outliers detectie, zoals downtime en ongebruikelijke pieken.
7. Under- en overfitting voorkomen.

8.2.1. Voorspelling binnen 15 minuten

Uit het interview met Jitscale kwam naar voren dat voor het bepalen van het aantal benodigde instances het belangrijk is dat er altijd snel een voorspelling is, ook als de time serie zodanig verandert waardoor de instellingen van de voorspellingsmethode opnieuw bepaald moeten worden. Jitscale heeft als richtlijn aangegeven dat het mogelijk moet zijn om binnen 15 minuten nieuwe instellingen te bepalen en een accurate voorspelling te doen. De reden voor deze eis is omdat het belangrijk is dat de voorspelling op tijd bekend is zodat de instances op tijd gestart kunnen worden. Deze eis is erg belangrijk en weegt zwaar bij de selectie van een voorspellingsmethode.

Met accuraat wordt bedoeld dat bij een time serie met herkenbare seasonaliteit en trend met 90% zekerheid het aantal instances bepaald kan worden. Of er een accurate voorspelling gedaan kan worden ligt sterk aan de time serie data. Als een time serie bestaat uit slechte data, zoals random waarden met geen seasonality, zal er ook geen betrouwbare voorspelling gedaan kunnen worden.

8.2.2. Voorspellen in de toekomst met time serie data

Doordat er bepaald moet worden hoeveel instances er nodig zijn op een bepaald tijdstip in de toekomst zal er voorspeld moeten worden aan de hand van data uit het verleden hoeveel instances er nodig gaan zijn. Dit zal gedaan worden met time serie data. Time serie data is een reeks aan datapunten die gemeten zijn over een bepaalde periode in tijd. Aan de hand van de waarde van time serie data kan een voorspelling gedaan worden in de toekomst. (Wikipedia, Time series, 2014)

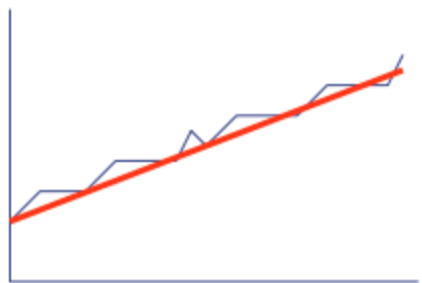
8.2.3. Voorspellen aan de hand van meerdere time series

Hoe druk een instance het heeft kan afgeleid worden aan verschillende statistieken, zoals het netwerk verkeer of de schrijfacties van de harde schijf. Doordat er verschillende statistieken zijn is het belangrijk dat de voorspellingsmethode de voorspellingen kan doen aan de hand van verschillende statistieken. Door een voorspelling te doen op basis van verschillende statistieken waaraan af te leiden is wat de behoeften is kan de voorspelling accurater worden. (Neal Wagner, 2011)

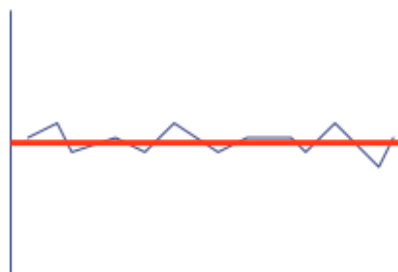
8.2.4. Trends herkennen

Een time serie kan verschillende trends hebben die de voorspellingmethode moet kunnen opmerken voor een nauwkeurige voorspelling. Een trend in een time serie kan bijvoorbeeld opwaarts of afwaarts zijn. Een instance die het afgelopen jaar drukker heeft gekregen zal een opwaartse trend hebben waaraan af te leiden is hoe snel de behoefte stijgt. Als de voorspellingsmethode geen trend kan herkennen zou het kunnen zijn dat de voorspelling te hoog of te laag uitkomt, doordat de groei of de daling van de behoeften niet meegenomen worden in de voorspelling.

In afbeelding 8.1 is een voorbeeld te zien van een time serie met een opwaartse trend en in afbeelding 8.2 een time serie zonder trend. (Crone, 2005)



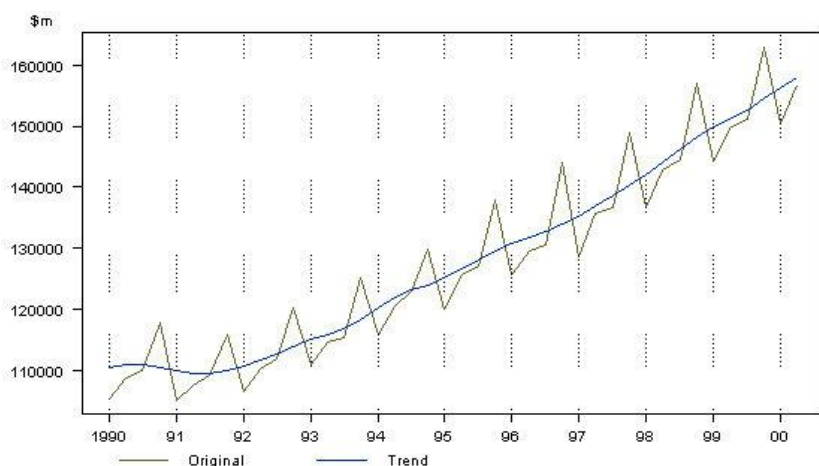
Afbeelding 8.1 - Opwaardse trend



Afbeelding 8.2 - Geen trend

8.2.5. Seasonality herkennen

Seasonality is het herhalend terugkeren van patronen in een periode. Een seasonality kan bijvoorbeeld een uurlijks, dagelijks, wekelijks, maandelijks of jaarlijks patroon zijn, maar het is ook mogelijk dat het patroon zich elke 6 uur voordoet. Door het herkennen van seasonalities is de voorspellingsmethode in staat te weten wanneer de behoeften minder of meer worden. Het kan zijn dat de trend opwaarts is maar dat betekent niet dat de behoeften altijd zullen stijgen. Dit is het geval wanneer er bijvoorbeeld dagelijks om 12 uur een daling in de behoeften is. In afbeelding 8.3 is een jaarlijkse seasonality te zien met een opwaartse trend. (Laborator, z.d).

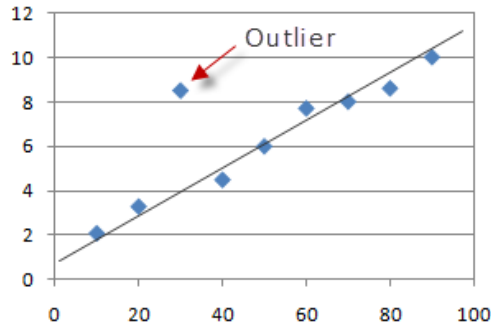


Afbeelding 8.3 - Jaarlijkse seasonality met opwaardsetrend

8.2.6. Outliers detectie

Een situatie die zich voor kan doen is dat de time serie data mist op bepaalde tijdstippen. In zo'n geval kan dit misleidend zijn voor de voorspellingsmethode doordat de voorspellingsmethode kan denken dat de instance het rustiger gehad heeft in de periode waar de data ontbreken. Dit zorgt ervoor dat de voorspelling niet nauwkeurig is en niet representatief is met het werkelijk aantal benodigde instances. Het ontbreken van data kan bijvoorbeeld komen doordat er downtime geweest is.

Data die niet kloppen met de werkelijkheid of misleidend zijn, zijn te herkennen doordat ze boven of onder de rest van de data uitschieten. Dit worden ook wel outliers (uitschieters) genoemd. Bij outliers is het overigens wel belangrijk dat er goed bepaald wordt of het inderdaad outliers zijn. Als er elke dag op het zelfde tijdstip een outlier is dan is het goed mogelijk dat dit een dagelijkse seasonality is in plaats van een outlier. In afbeelding 8.4 is te zien hoe de outlier vergeleken met de rest van de datapunten een stuk hoger ligt. (Wikipedia, Outlier, 2014), (Hans-Peter Kriegel, 2010), (Reference, 2009)



Afbeelding 8.4 - Outlier

8.2.7. Under- en overfitting voorkomen

Voor het doen van een voorspelling is het van belang dat de voorspelling niet leidt aan under- of overfitting. Underfitting wordt veroorzaakt als de voorspellingsmethode niet in staat is de onderliggende trend te herkennen. En overfitting wordt veroorzaakt als de voorspellingsmethode zijn voorspelling te veel baseert op noise in plaats van op onderliggende relaties tussen data. Bij overfitting is de voorspelling erg accuraat als er een voorspelling gedaan wordt op de huidige time serie maar op het moment dat time serie wijzigt zal de voorspelling verre van accuraat zijn. Dit komt doordat bij overfitting de voorspellingsmethode de voorspelling precies baseert op de huidige time serie in plaats van op de onderliggende trends. Tussen under- en overfitting moet een midden weg gevonden worden. (Cai, 2014), (data, 2013), (Fortmann-Roe, 2012), (Bizzarodavid, 2011), (Wikipedia, Overfitting, 2014)

8.3. Voorspellingsmethode selectie

Na het in kaart brengen van de eisen waaraan de voorspellingsmethodes moeten voldoen, is er bepaald welke voorspellingsmethode gebruikt gaat worden voor het voorspellen van het aantal benodigde instances. Als eerste wordt uitleg gegeven over de voorspellingsmethodes waar tussen geselecteerd is en vervolgens wordt beschreven hoe en op basis waarvan de voorspellingsmethode geselecteerd is.

8.3.1. Voorspellingsmethodes

Bij het onderzoeken van de voorspellingsmethodes bleek dat er veel methodes zijn voor het doen van voorspellingen en dat deze voor time serie te categoriseren zijn in de volgende categorieën:

- Autoregressive integrated moving average (ARIMA)
- Linear prediction
- Moving average
- Artificial intelligence

(Wikipedia, Autoregressive integrated moving average, 2014), (Wikipedia, Linear regression, 2014), (Investopedia, Moving Average - MA, z.d), (Wikipedia, Artificial intelligence, 2014)

Voor deze categorieën zijn vijf voorspellingsmethodes geselecteerd die mogelijk gebruikt kunnen worden voor het voorspellen van het aantal instances en die de moeite waard zijn voor verder onderzoek omtrent de gestelde eisen:

- ARIMA: Box-Jenkins
- Linear prediction: Clustered linear regression
- Moving average: Triple exponential smoothing
- Artificial intelligence:
 - Artificial neural networks
 - Support vector machines

(Software, z.d), (Danny Yuan, 2013), (Laboratory, Triple Exponential Smoothing, z.d), (Laboratory, Example of Triple Exponential Smoothing), (DTREG), (Investopedia, Neural Network, z.d), (Crone, 2005)

Voor de selectie van de vijf voorspellingsmethodes is de werking van de methodes uitgebreid onderzocht, zodat bepaald kon worden of het mogelijk is om de methode te gebruiken voor het voorspellen van het

aantal instances. Bij de selectie is rekening gehouden dat de voorspellingsmethodes voorspellingen kunnen doen op time series en is gekeken in hoeverre de methode te automatiseren is. Hoe meer er te automatiseren is, hoe minder de Jitscale medewerker, die de voorspellingsmethode moet configureren, hoeft te weten van de werking van de onderliggende werking van de methode.

8.4. Cross reference analyse

Uit de vijf voorspellingsmethode die geselecteerd zijn uit het bronnen onderzoek is één methode geselecteerd welke gebruikt gaat worden in de oplossing waarmee instances voorspeld gaan worden. Voor de selectie is een cross reference analyse gedaan welke te zien is in tabel 8.1.

Eisen						
Het moet mogelijk zijn om binnen 15 minuten met nieuwe instelling een accurate voorspelling te doen.						
Noise en outliers kunnen detecteren en negeren.						
Trends herkennen.						
Under- en overfitting voorkomen.						
Meerdere seasonalities herkennen.						
Voorspellen op basis van meerdere time series.						
Voorspellingsmethodes						
Clustered linear regression	/	X	/	X		X
Artificial neural networks	X	X	/	X		
Support vector machine	X	X	/	X		
Box-Jenkins	/	/	/	X		/
Triple exponential smoothing	/	X	/	X		X
Prioriteit	2	2	2	2	2	1

Tabel 8.1 - Cross reference analyse

In de cross reference analyse is horizontaal te zien aan de hand van de (halve) kruizen welke voorspellingsmethodes er voldoen aan een eis. Halve kruizen geven aan dat het mogelijk is maar dat de voorspellingsmethode hiervoor uitgebreid moet worden. In de tabel is ook, verticaal, de prioriteit van de eisen te zien. Daarbij heeft het doen van een voorspelling binnen 15 minuten, zoals eerder beschreven, een hoger prioriteit. Voorspellingmethodes die aan deze eis voldoen, zullen voorrang krijgen bij de selectie.

8.4.1. Voorspelling binnen 15 minuten

Het grote voordeel van Clustered linear regression is dat het doen van voorspellingen met lineaire regressie snel is en het model niet getraind hoeft te worden voordat er een voorspelling gedaan kan worden, zoals bij Artificial neural networks en Support vector machine het geval is. Hierdoor is het mogelijk om de accuraatheid van nieuwe instellingen snel te testen en te wijzigen als de voorspelling niet accuraat genoeg is. (Een model is een combinatie aan instellingen waarmee voorspellingen gedaan worden.)

Bij Box-Jenkins moet een procedure doorlopen worden om te bepalen welk model mogelijk een accurate voorspelling geeft. Daarna moet getest worden of het model accuraat is en zo nodig aangepast moet worden, waarbij de procedure opnieuw doorlopen moet worden. Er zijn veel verschillende modellen mogelijk waardoor het kiezen van het juiste model complex kan zijn en veel tijd in beslag kan nemen. De complexiteit is afhankelijk van de time serie. Dit komt doordat het model bepaald wordt aan de hand van

patronen in de time serie. Als de time serie minder duidelijke patronen heeft wordt het bepalen van het model complexer. Doordat de tijdsduur van het bepalen van het model afhankelijk is van de time serie is het mogelijk dat het langer duurt dan 15 minuten. In Bijlage 6 – Voorspellingsmethodes is beschreven hoe met Box-Jenkins een model bepaald wordt. (Stellwagen, z.d)

Net zoals bij lineaire regressie is het bij Triple exponential smoothing niet nodig om het model van te voren te trainen waardoor nieuwe instellingen snel getest kunnen worden en zo nodig aangepast kunnen worden als de voorspelling niet accuraat genoeg is.

Bij Artificial neural networks en Support vector machine moet eerst bepaald worden welke instellingen er gebruikt worden voor het model, waarna het model getraind moet worden aan de hand van trainingsdata. De tijd dat het model nodig heeft om te trainen is afhankelijk van de instellingen en trainings data. Om die reden is het niet zeker of de training klaar is binnen 15 minuten. Er zijn mogelijkheden om de trainingstijd te versnellen maar het bepalen van de juiste instellingen waarmee de training versneld kan worden, neemt ook tijd in beslag. (Martí & El-Fallahi, 2002), (Arnosti & Kalita, z.d)

8.4.2.Noise en outliers kunnen detecteren en negeren

Uit onderzoek blijkt dat bij alle voorspellingsmethodes outliers en noise invloed hebben op de voorspelling, en zelf geen methode hebben om dit te voorkomen. Het is daarom aan te raden om bij elke methode outlier detectiemethodes te gebruiken om outliers en noise te vinden, zodat deze genegeerd kunnen worden bij het doen van een voorspelling. Hoe outliers gedetecteerd en genegeerd worden is beschreven hoofdstuk 9.5 - Outlier detection. (Stellwagen, z.d), (Scikit-learn, z.d), (Tomas Cipra, 2011), (Beliakov, Kelarev, & Yearwood, 2011)

8.4.3.Trends herkennen

Het is met alle vijf de voorspellingsmethodes mogelijk om voorstellingen te doen als er een trend (opwaarts, stationair of naar beneden) in zit. Bij Box-Jenkins en Clustered Linear Regression moet de time serie eerst stationair gemaakt worden voordat het model bepaald kan worden (dit is nodig voor het bepalen van de seasonality). Artificial Neural Network en Support Vector Machine zijn zelflerende voorspellingsmethodes, dus deze herkennen de trend automatisch. En bij Exponential Smoothing kunnen trends herkend worden als gebruik gemaakt wordt van Double Exponential Smoothing of Triple exponential smoothing. (Wikipedia, Linear regression, 2014), (Term, 2011), (Kalekar, 2004), (Stergiou, z.d), (Yu, Chen, Wang, & Lai, 2008),

8.4.4.Under- en overfitting voorkomen

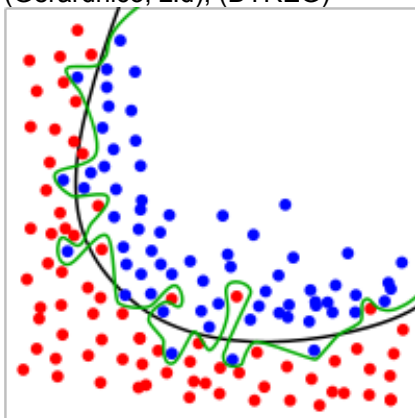
Bij Clustered linear regression en Triple exponential smoothing (Holt-Winters seasonal smoothing) is er een kans op under- en overfitting, die op te lossen is door K-fold cross validation te gebruiken, zodat de beste instellingen bepaald kunnen worden. Uitleg over K-fold cross validation is te vinden in Bijlage 6 – Voorspellingsmethodes. (Gaer, Ceulemans, & Mechelen), (Duke, z.d)

Bij Box-Jenkins/ARIMA is het mogelijk dat er overfitting plaatsvindt. Om dit te controleren kan een overfitting techniek/proces, gebaseerd op de BIC criteria, toegepast worden op het ARIMA model om te testen of het gekozen model juist is. Dit wordt gedaan door in het ARIMA model de autoregressive en moving average waarden te veranderen en te testen of de nieuwe waarden zinvol zijn voor de voorspelling. Zie Bijlage 6 – Voorspellingsmethodes voor verdere uitleg over het ARIMA model. (Alonso & Garcia-Martos, 2012), (Term, 2011), (Wikipedia, Bayesian information criterion, 2014)

Bij Artificial neural networks is er een kans op under- of overfitting als te weinig of te veel hidden neurons gebruikt worden. Door het gekozen model te testen op test data waar niet op getraind is, kan gecontroleerd worden of er under- of overfitting plaats vindt, waarna het model aangepast kan worden om under- en overfitting te voorkomen. Uitleg over de werking van Artificial neural networks is te vinden in Bijlage 6 – Voorspellingsmethodes. (Okkels, 2014)

Support vector machine is echter minder gevoelig voor overfitting. Dit komt doordat de voorspellingen gebaseerd worden op de support vectors en niet op de rest van de data. Toch is er bij Support vector

machine wel een kans op overfitting. Overfitting komt voor op het moment dat de data niet goed van elkaar zijn te onderscheiden. Op dat moment wordt geprobeerd de data alsnog te onderscheiden wat resulteert in heel veel optimale hyperplanes en support vectors. Dit kan voorkomen worden door met de cost parameter aan te geven dat het toegestaan is dat niet alle data te onderscheiden zijn wat resulteert in een hogere trainings error. In afbeelding 8.5 is te zien dat de groene lijn alle datapunten probeert te onderscheiden, wat zorgt voor overfitting. De zwarte lijn is het resultaat van de cost parameter waarbij te zien is dat sommige rode punten rechts en blauwe punten links van de lijn toegestaan zijn, wat resulteert in een hogere trainings error. Uitleg over de werking van Support vector machine hoe het gebruik maakt van support vectors en hyperplanes is te vinden in Bijlage 6 – Voorspellingsmethodes vector machine. (Gerardnico, z.d), (DTREG)



Afbeelding 8.5 - SVM overfitting

8.4.5. Meerdere seasonalities herkennen:

Ook de herkenning van meerdere seasonalities is met alle vijf de voorspellingsmethodes mogelijk. Met Clustered linear regression is dit mogelijk door het clusteren van datapunten. Doordat het hiermee mogelijk is verschillende seasonalities te clusteren, kan hierop vervolgens lineaire regressie uitgevoerd worden. (Wikipedia, Linear regression, 2014), (Danny Yuan, 2013)

Ook met Box-Jenkins is het mogelijk om meerdere seasonalities te gebruiken maar hierbij moet wel het ARIMA model aangepast worden naar SARIMA, waarbij de S staat voor seasonal. (COORAY, z.d), (Suhartono, 2011), (Au, Ma, & Yeung, z.d), (Wikipedia, Autoregressive integrated moving average, 2014), (UMBRELLA, 2013),

Artificial neural networks en Support vector machine leren zelf de patronen herkennen dus voorspellingen doen met meerdere seasonalities is mogelijk. Als gebruik gemaakt wordt van Triple exponential smoothing dan is het ook mogelijk om voorspellingen te doen op het moment dat meerdere seasonalities aanwezig zijn. (Kalekar, 2004), (Taylor, 2003), (Kourentzes & Crone, z.d), (Nguyen & Chan, 2003)

8.4.6. Voorspellen op basis van meerdere time series

Om met Clustered linear regression voorspellingen te doen op basis van meerdere time series, moet in plaats van “simple linear regression”, “multi linear regression” gebruikt worden. Het is dus mogelijk maar er is wel een aanpassing nodig aan de manier waarop linear regression toegepast wordt. (ReStore, z.d), (Statistics, z.d)

Bij Box-Jenkins wordt gebruik gemaakt van het ARIMA model maar voor het doen van voorspellingen op meerdere time series is het ARIMAX model nodig, wat gezien kan worden als een multi regression model. Voor het bepalen van het ARIMAX model is de Box-Tiao methode ontwikkeld. Dus voor het doen van voorspellingen op meerdere time series moet Box-Tiao gebruikt worden in plaats van Box-Jenkins. (Institute, 2012), (Little, 2012), (Andrews, Dean, Swain, & Cole, 2013), (SAS)

Met Triple exponential smoothing is het niet mogelijk om voorspellingen te doen aan de hand van meerdere time series maar het is wel mogelijk om exponential smoothing uit te breiden, zodat dit wel

mogelijk wordt. Dit wordt ook wel Multivariate Exponential Smoothing genoemd. (Athanasopoulos & Silva, 2010), (Multivariate Exponential Smoothing: prediction, signal extraction and control)

Bij Artificial neural networks kan meer dan één time series als input gegeven worden aan het neural network waardoor het mogelijk is om een voorspelling te doen op meerdere time series. (Crone, 2005)

Met Support vector machine is het ook mogelijk om voorspellingen te doen aan de hand van meerdere time series. Hiervoor kan de multi kernel LS-SVR gebruikt worden. (GUO, RAN, LI, MA, & ZHANG, 2013)

8.4.7. Resultaat

Uit de cross reference analyse blijkt dat alle voorspellingsmethodes voldoen aan een groot gedeelte van de eisen. Sommige methodes hebben een uitbreiding nodig om te voldoen aan de eis maar daarmee wordt wel gezorgd dat de methode bruikbaar is, dus dit is geen probleem. Bij de selectie van de voorspellingsmethode die gebruikt gaat worden speelt de prioriteit van de eisen een grote rol. Zo is het met Clustered linear regression, Triple exponential smoothing en Box-Jenkins mogelijk om binnen 15 minuten een voorspelling te doen, waarbij er bij Box-Jenkins wel een kans is dat het langer duurt dan 15 minuten. Clustered linear regression en Triple exponential smoothing voldoen beide aan de gestelde eisen, dus hierin zit geen verschil. Er moet dus een keuze gemaakt worden tussen Clustered linear regression en Triple exponential smoothing.

De voorkeur gaat hierbij uit naar Clustered linear regression. Dit komt doordat het met daarmee mogelijk is om aan de hand van het clusteren van datapunten aan te geven welke data meer of minder belangrijk zijn, wat Clustered linear regression flexibiliteit geeft bij het doen van een voorspelling. Daarnaast is voor het doen van een voorspelling met Clustered linear regression, alleen de data nodig die geclusterd is voor een voorspelling op een bepaald tijdstip. Dit zorgt ervoor dat Clustered linear regression minder data hoeft te gebruiken wat zorgt voor winst in de snelheid van de voorspelling.

Triple exponential smoothing is ook zeker een goede voorspellingsmethode en heeft zeker potentie om als volgende voorspellingsmethode geïmplementeerd te worden binnen J-Predictor.

8.5. Conclusie

Voor de selectie van de voorspellingsmethode moesten twee deelvragen beantwoord worden.

Deelvraag 4d: Aan welke eisen moet de voorspellingsmethode voldoen?

Deelvraag 4e: Welke voorspellingsmethode voldoet aan deze eisen?

Deelvraag 4d is beantwoord door het bepalen van eisen naar aanleiding van een bronnenonderzoek en de interviews. De eisen die deelvraag 4d beantwoorden zijn beschreven in paragraaf 8.2 -

Voorspellingsmethode eisen. Voor het beantwoorden van deelvraag 4e is een bronnenonderzoek en cross reference analyse gedaan waaruit blijkt dat Clustered linear regression en Exponential smoothing het beste voldoen aan de eisen. Hierbij is Clustered linear regression gekozen als voorspellingsmethode die geïmplementeerd gaat worden.

9. Functioneel onderzoek

Na het onderzoeken van verschillende voorspellingsmethodes en het selecteren van de voorspellingsmethode die gebruikt gaat worden is er een oplossing ontwikkeld waarmee de voorspellingsmethode gebruikt wordt voor het bepalen van het aantal benodigde instances. Als eerste wordt in dit hoofdstuk uitleg gegeven over hoe het functioneel onderzoek verricht is en vervolgens zullen de resultaten beschreven worden.

9.1. Onderzoeksmethoden

Voor het vaststellen van wat er ontwikkeld is, is er een functioneel onderzoek gedaan. Voor dit onderzoek was het nodig om interviews te doen bij Jitscale met daarnaast een (intern) bronnenonderzoek. Bij dit onderzoek worden twee sub vragen beantwoord:

Deelvraag 4a: Wat zijn de functionele en niet-functionele eisen?

Deelvraag 4b: Wat zijn de constraints?

En het functionele gedeelte voor deelvraag 4 wordt beantwoord:

Deelvraag 4: Wat is de beste oplossing om te implementeren binnen J-Layer?

Het volgende is onderzocht:

Deelvraag 4a (Interviews, intern bronnenonderzoek):

- De functionele eisen.
- De niet functionele eisen.

Deelvraag 4b (Interviews, intern bronnenonderzoek):

- De business rules (constraints).

Deelvraag 4 (bronnenonderzoek):

- Hoe Clustered linear regression (CLR) functioneel gaat werken.
- Hoe under- en overfitting voorkomen kan worden. (functionele eis)
- Hoe outliers gedetecteerd kunnen worden. (functionele eis)
- Hoe de oplossing functioneel gaat werken.

Het resultaat van het functioneel onderzoek is te vinden in Bijlage 3 – Functioneel ontwerp.

9.2. Functionele en niet-functionele eisen

Voor het beantwoorden van deelvraag 4a zijn Jitscale medewerkers geïnterviewd en is een intern bronnenonderzoek gedaan. Hieruit zijn de functionele en niet-functionele eisen vastgesteld, die beschreven worden in Bijlage 3 – Functioneel ontwerp. Hieronder zullen de functionele en niet-functionele eisen toegelicht worden die van belang zijn voor het bepalen van de werking van Clustered linear regression (CLR), under- en overfitting voorkomen, outlier detectie en de manier waarop de oplossing functioneel gaat werken.

Voor CLR zijn er de volgende functionele eisen van belang:

- FE3: Het bepalen van het aantal instances wordt gedaan door een voorspelling op basis van time serie data.
- FE14: Er kan aangegeven worden welke periode voorspeld moet worden.
- FE27: De voorspellingsmethode kan meerdere seasonalities herkennen.
- FE28: De voorspellingsmethode kan trends herkennen.

Deze eisen zijn van belang, zodat bij het bepalen van hoe CLR functioneel gaat werken het mogelijk is om meerdere seasonalities en trends te herkennen. Daarbij moet CLR in staat zijn om een voorspelling te doen aan de hand van een time serie en moet aangegeven worden voor welke periode een voorspelling gedaan moet worden. Om ervoor te zorgen dat CLR hiertoe in staat is, was het nodig om te bepalen welke instellingen er van belang zijn voor de werking van CLR. Deze instellingen zijn beschreven in paragraaf 9.3 - Werking van clustered linear regression.

Naam: Rogier Schipper

Studentnummer: 1583842

Opleiding: Informatica Deeltijd

Datum: 12-10-2014

Voor het bepalen hoe under- en overfitting voorkomen wordt, waren de volgende functionele en niet-functionele eisen van belang:

- FE30: De voorspellingsmethode kan under- en overfitting voorkomen.
- FE20: De beste settings voor de voorspellingsmethode kunnen automatisch bepaald worden.

Om aan deze functionele eisen te voldoen wordt K-fold cross validation gebruikt, wat wordt beschreven in paragraaf 9.4 - K-Fold cross validation. Naast het voorkomen van under- en overfitting is het daarmee mogelijk automatisch de beste settings te bepalen.

Voor outlier detectie zijn de volgende functionele en niet-functionele eisen van belang:

- FE29: De voorspellingsmethode is in staat om noise en outliers te detecteren en negeren.
- NFE24: De voorspellingsmethode moet een accurate voorspelling kunnen doen. Met accuraat wordt bedoeld dat bij een time serie met herkenbare seasonaliteit en trend met 90% zekerheid het aantal instances bepaald kan worden.

Outliers zorgen ervoor dat de voorspelling minder accuraat is. Daardoor is het van belang dat outliers gedetecteerd en genegeerd worden. Hoe dit gedaan wordt is beschreven in paragraaf 9.5 - Outlier detection.

Binnen de niet-functionele eisen zijn drie noemenswaardige eisen:

- NFE1: J-Predictor moet schaalbaar zijn, zodat er extra servers ingezet kunnen worden.
- NFE2: J-Predictor moet ingezet kunnen worden voor andere voorspellingsdoeleinden dan het voorspellen van het aantal instances.
- NFE20: Het moet mogelijk zijn om voorspellingsmethodes pluggable toe te voegen.

Voor NFE20 is in het technisch ontwerp, Bijlage 4 – Technisch ontwerp, rekening gehouden met het feit dat de voorspellingmethode 'loosely coupled' is, waarmee gezorgd wordt dat een voorspellingsmethode gewijzigd of toegevoegd kan worden zonder dat dit effect heeft op de rest van de applicatie. (Wikipedia, Coupling (computer programming), 2014)

Door NFE1 is er in het ontwerpen van de applicatie rekening meegehouden dat de oplossing schaalbaar is. Daarbij moet voor NFE2 gezorgd worden dat de oplossing gebruikt kan worden voor andere doeleinden. Voor die reden is gekozen om de oplossing waarmee voorspeld wordt niet te implementeren binnen J-Layer, zoals aangegeven wordt in het plan van aanpak. In plaats daarvan is er een losse applicatie ontwikkeld waar J-Layer gebruik van maakt voor het doen van voorspellingen, genaamd J-Predictor.

Doordat de oplossing niet geïmplementeerd wordt binnen J-Layer is er meer flexibiliteit waardoor het mogelijk is om de J-Predictor los van J-Layer te schalen. Dit zorgt ervoor dat er extra computerkracht ingezet kan worden voor het doen van voorspellingen en voor minder coupling met J-Layer doordat in het design van J-Layer geen rekening gehouden hoeft te worden met het schalen van de voorspellingsoplossing.

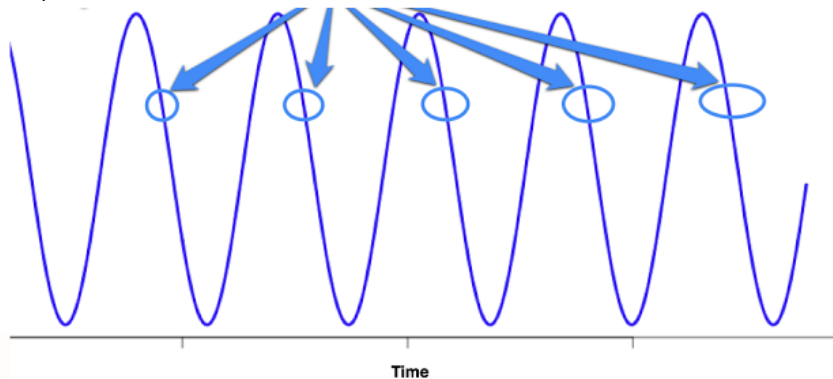
9.3. Werking van clustered linear regression

CLR is geïmplementeerd als voorspellingsmethode binnen J-Predictor. Voor de implementatie van CLR was echter een uitgebreider onderzoek nodig naar CLR dan al gedaan was. Dit is gedaan aan de hand van een bronnenonderzoek en met dit onderzoek wordt een gedeelte van deelvraag 4 beantwoord, hoe CLR functioneel gaat werken.

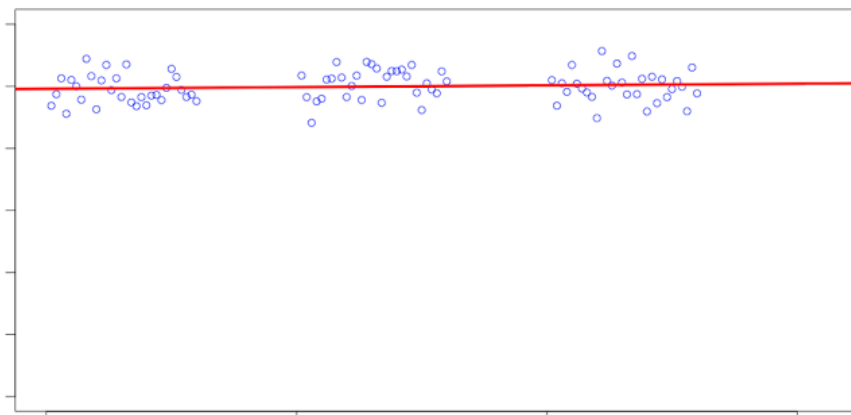
CLR is een manier van lineaire regressie waarbij de lineaire regressie op clusters van datapunten wordt uitgevoerd. Een voorbeeld waarbij CLR gedaan wordt, is te zien in afbeelding 9.1 en afbeelding 9.2. Met de cirkels in afbeelding 9.1 wordt aangegeven op welke punten in de time serie datapunten worden geclusterd.

Als een time serie een seasonality van een dag heeft en je wilt weten wat de waarde gaat zijn van de datapunten vandaag tussen 19:00 uur en 19:15 uur clustert CLR datapunten van de afgelopen dagen

tussen 19:00 en 19:15 uur, zoals te zien is in afbeelding 9.1. Na het clusteren voert CLR lineaire regressie uit op de geclusterde data, zoals te zien is in afbeelding 9.2, waarbij de rode lijn de regressie lijn is. Door het doen van lineaire regressie wordt het mogelijk om met extrapolatie te voorspellen wat de waarde zal zijn van de data punt de volgende dag tussen 19:00 en 19:15 uur. Bij CLR zijn dus twee dingen erg belangrijk, welke data er geclusterd worden en de seasonality. De clusters zijn van belang zodat er meer datapunten zijn waar de lineaire regressie op gedaan kan worden. Daarnaast de seasonality zodat datapunten die overeenkomsten met elkaar hebben, en dus dicht bij elkaar liggen qua waarden, geclusterd worden. Hoe meer overeenkomsten de waarden hebben hoe accurater lineaire regressie een voorspelling kan doen. (Wikipedia, Linear regression, 2014), (Wikipedia, Extrapolatie, 2014)



Afbeelding 9.1 - Clusters in een time serie



Afbeelding 9.2 - Lineaire regressie op geclusterde data

Bij het onderzoek naar de implementatie van CLR is vastgesteld dat CLR kan werken aan de hand van een aantal instellingen. Bij het vaststellen van deze instellingen is rekening gehouden met het feit dat de configuratie van CLR goed te begrijpen is door mensen die weinig wiskundige ervaring hebben. De instellingen die nodig zijn, zijn:

Seasonality: Hiermee wordt aangegeven wat de periode van de seasonality is zoals elke 7 dagen. De verschillende mogelijke seasonalities zijn: minuten, uren, dagen, weken, maanden, jaren.

Maximale cluster grootte: Hiermee wordt aangegeven hoeveel datapunten een cluster maximaal mag zijn.

Minimale cluster grootte: Hiermee wordt aangegeven hoeveel datapunten een cluster minimaal moet hebben.

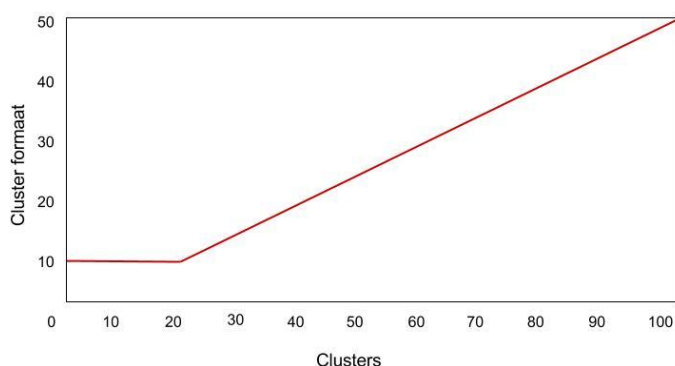
Historische relevantie: Hiermee wordt aangegeven tot waar in de time serie de data relevant is voor de voorspelling. Als hierbij wordt aangegeven dat 80% van de time serie relevant is, wordt 80% van het aantal clusters de maximale grootte zijn, en 20% de minimale grootte.

Historische relevantie afname: Hiermee wordt aangegeven hoe snel de relevantie van historische data afneemt. Als hier aangegeven wordt dat de historische relevantie afneemt met een snelheid van 100%, betekent dit dat cluster verder in het verleden steeds kleiner worden. Waarbij het laatste cluster gelijk is aan de minimale cluster grootte en de meeste recente cluster gelijk is aan de maximale cluster grootte.

In afbeelding 9.3 is ter verduidelijking te zien hoe de cluster grootte afneemt bij met de volgende instellingen:

- Historische relevantie afname: 100%
- Historische relevantie: 80%
- Maximale cluster grootte: 50
- Minimale cluster grootte: 10

Zoals in afbeelding 9.3 te zien is, is de laatste 20% (van clusters 0 tot 20) de minimale cluster grote. Dit komt doordat maar 80% van de data relevant is en de laatste 20% niet groter mag zijn dan de minimale cluster grootte. Als de historische relevantie afname snelheid 0% zou zijn, dan zou de rode lijn tussen cluster 20 en cluster 100 horizontaal zijn met een cluster formaat van 50, en tussen 0 en 20 ook horizontaal met een cluster formaat van 10.



Afbeelding 9.3 – Historische relevantie

Begin en einde van de voorspelling: Hiermee wordt de periode aangegeven die voorspeld moet worden. Als het begin 1 is en het einde 60 betekent dit dat er een voorspelling gedaan wordt voor een uur in de toekomst.

De reden dat seasonality een instelling is, is omdat met CLR aan de hand van de seasonality weet waar in de time serie data geclusterd moet worden. Bij de seasonality kan als type opgegeven worden: minuten, uren, dagen, weken, maanden, jaren. Met deze periodetypes kunnen vrijwel alle benodigde seasonalities ingesteld worden. Met minuten bijvoorbeeld elke 30 minuten en met jaren kan een decennium aangegeven worden mocht dat nodig zijn.



CLR is afhankelijk van clusters van data voor het doen van een voorspelling. Voor het bepalen van de clusters is het van belang dat de aangegeven wordt wat het formaat van de cluster is. Het formaat van een cluster bepaalt hoe belangrijk een cluster is voor een voorspelling (gewicht). Dus door aan te geven hoe groot een cluster moet zijn kan het resultaat van de voorspelling beïnvloed worden, doordat lineaire regressie de datapunten gebruikt voor het bepalen van de trendlijn. Hoe meer datapunten met waarden 10, hoe meer de trendlijn bij 10 zal zitten.

Het bepalen van hoe belangrijk oude of recente data is voor een voorspelling, wordt gedaan met de settings 'historische relevantie' en 'historische relevantie afname'. Bij het doen van een voorspelling kan het zijn dat recente gebeurtenissen in een time serie een belangrijke rol spelen bij het doen van een voorspelling. Zo kan het zijn dat bij een time serie van een jaar aan data er de laatste twee weken een opwaartse trend is die meegenomen moet worden in de voorspelling. Als alle clusters even groot zijn over een periode van een jaar betekent dit dat de opwaartse trend in maar 2 weken van alle weken in een

jaar niet genoeg gewicht heeft om de voorspelling hoger uit te laten komen. De overige weken in dat jaar houden de trendlijn lager doordat die weken samen meer gewicht hebben dan de laatste 2 weken. De reden voor de setting 'historische relevantie' en 'historische relevantie afname' is om aan te geven dat de laatste 2 weken belangrijk zijn voor het doen van voorspellingen. Met 'historische relevantie' kan ingesteld worden dat de laatste 10% het meest relevant is waardoor die clusters de maximale grootte hebben en de rest (90% van de time serie) de minimale cluster grootte. Daarbij kan het in sommige gevallen ook nodig zijn dat de relevantie van data per week gelijkmatig afneemt. Door de setting 'historische relevantie afname' is het ook mogelijk om dit in te stellen.

Op het moment dat een voorspelling gedaan wordt, wordt dit gedaan voor een bepaald punt of periode in de tijd. Om die reden is het mogelijk om in te stellen voor welke periode er een voorspelling gedaan moet worden met het begin en het einde van de voorspelling. Hierbij is het getal 1 de eerste waarde na de time serie waar de voorspelling op gebaseerd is.

Doordat het mogelijk is dat de interval tussen lags per time serie kunnen verschillen, een lag interval kan bijvoorbeeld 1 minuut of 15 minuten zijn, is besloten dat alle time series geconverteerd worden zodat een lag interval altijd 1 minuut is. (Een lag is een data punt in de time serie. Een time serie met 100 datapunten heeft dus 100 lags.) Hierdoor zijn de settings die afhankelijk zijn van de lag interval universeel te gebruiken op alle time series. Dit betekent bijvoorbeeld dat settings waarmee de begin- en eind periode worden aangegeven altijd in minuten is. Door de eenmalige conversie van de time serie is het niet nodig om per time serie en per setting een conversie te doen. De gebruiker kan de periode in stellen als bijvoorbeeld uren en dagen, zoals te zien is in afbeelding 9.4. Deze waarden zullen onderliggend geconverteerd worden naar minuten.

Prediction period: start: days 
end: days 

Afbeelding 9.4 - Voorspellingsperiode

9.4. K-Fold cross validation

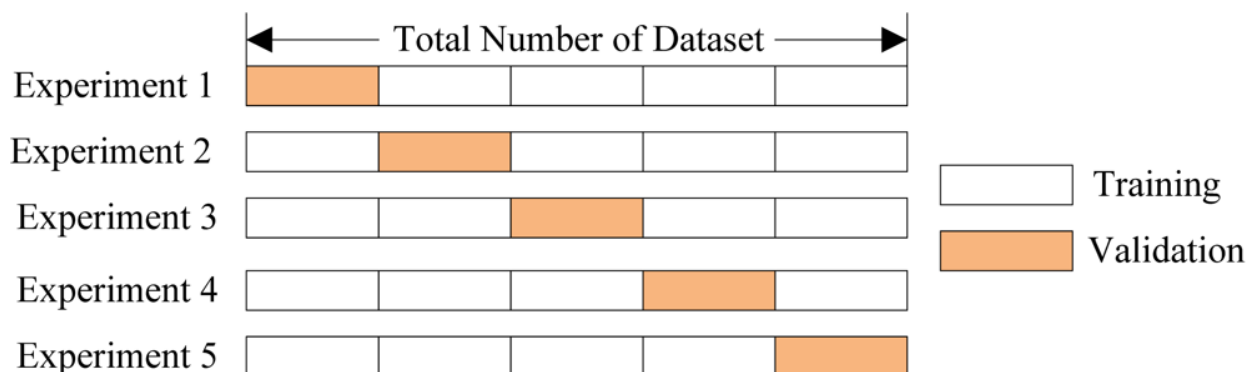
Bij het bepalen van de eisen in hoofdstuk 8.2 is één van de eisen dat under- en overfitting voorkomen moet worden. CLR blijkt hier zelf geen oplossing voor te hebben. Om die reden is een bronnenonderzoek gedaan om te bepalen welke oplossing geïmplementeerd moet worden en dit alsnog mogelijk te maken. Met dit onderzoek wordt een gedeelte van deelvraag 4 beantwoord, hoe under- en overfitting voorkomen kan worden.

Voor het doen van een voorspelling moet een combinatie aan instellingen (model) gevonden worden waarmee een zo goed mogelijke voorspelling gedaan wordt. Voor CLR gaat het hierbij om de instellingen: seasonality, maximum cluster grootte, minimum cluster grootte, historische relevantie en historische relevantie afname.

Het juiste model is een model dat voorspellingen doet die overeenkomen met de werkelijkheid maar waarbij geen under- of overfitting plaats vindt. Voor het valideren of een model juist is, is ervoor gekozen om K-fold cross validation te gebruiken.

Voor het valideren van een model moet er test data beschikbaar zijn waarmee gevalideerd wordt of een voorspelling klopt. Aangezien het van belang is dat de test data niet gebruikt worden voor het trainen van de data, zoals beschreven is in Bijlage 6 – Voorspellingsmethodes, betekent dit dat er bij data sets met weinig data belangrijke data niet gebruikt kunnen worden voor het trainen van de voorspelling. Om die reden is voor K-fold cross validation gekozen, doordat K-fold cross validation alle data gebruikt voor het trainen en het testen van data. (De werking van K-fold cross validation is beschreven in Bijlage 6 – Voorspellingsmethodes.) Door de manier van trainen en testen is K-fold cross validation in staat om nieuwe modellen te vinden bij nieuwe datasets of bij datasets die veranderen waardoor het huidige model geen accurate voorspelling meer kan doen en er dus een nieuw model nodig is. K-fold cross validation test elk model en bepaald per model wat de mean square error is. Aan de hand van de mean square

error wordt bepaald welk model het beste is. Met de mean square error wordt bepaald wat het verschil is tussen de waarde van de voorspellingen en de waarde van de daadwerkelijke data. Hoe lager de mean square error, hoe accurater het model voorspellingen doet. In afbeelding 9.5 is te zien hoe k-fold cross validation werkt, waarbij de time serie opgesplitst wordt in test en training data. De data wordt opgedeeld in vijf blokken wat betekent dat het om 5-fold cross validation gaat. (Wikipedia, Mean squared error, 2014)



Afbeelding 9.5 - K-fold cross validation

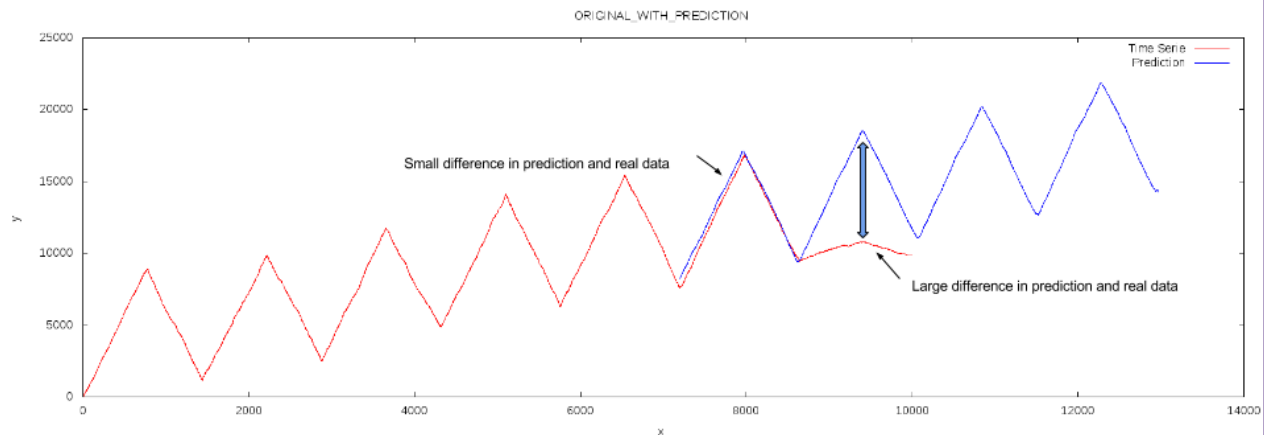
Bij K-fold cross validation moet er bepaald worden in hoeveel delen de data set opgedeeld wordt (K-fold). Hoe meer delen, hoe langer het duurt om te trainen. Voor grote data sets kunnen minder delen, bij voorbeeld 3-fold, gebruikt worden voor het vinden van een juist model, maar bij een kleine dataset is het nodig om de data set op te delen in meer delen. Bij kleine datasets kan het zelfs nodig zijn leave-one-out cross validation(loocv) te gebruiken wat als ware K-fold cross validation is waarbij K het aantal datapunten in de data set is. Bij elke training wordt hierbij maar één datapunt weggelaten voor het doen van de test. Het voordeel van loocv is dat er geen data verspeeld wordt, in vergelijking tot bijvoorbeeld 10-fold cross validation waarbij 10% van de data verspeeld wordt. Het nadeel van loocv is dat het qua performance erg intensief is doordat er veel meer getraind en getest moet worden. Bij grote data sets zal loocv daarom langer duren dan bijvoorbeeld 10-fold cross validation.

Naast het voorkomen van under- en overfitting wordt K-fold cross validation gebruikt voor het automatisch bepalen van het model dat gebruikt wordt voor de voorspelling. Dit wordt gedaan door met K-fold cross validation automatisch verschillende modellen te testen, waaruit zal blijken welk model de meest accurate voorspelling geeft. (Wikipedia, Mean squared error, 2014), (Gutierrez-Osuna, z.d), (Moore, z.d), (MathWorks, z.d), (Hyndman, 2010)

9.5. Outlier detection

Bij het bepalen van de eisen in hoofdstuk 8.2 is één van de eisen dat outliers gedetecteerd moeten worden. Omdat CLR hiervoor zelf geen oplossing heeft is een bronnenonderzoek gedaan om te bepalen hoe hier wel voor gezorgd kan worden. Met dit onderzoek wordt een gedeelte van deelvraag 4 beantwoord, hoe outliers gedetecteerd kunnen worden, zoals beschreven is in paragraaf 9.1.

Bij het doen van een voorspelling is het van belang dat de time serie data bevat die juist zijn. Als er een storing is geweest kan het zijn dat de time serie data mist waardoor de time serie niet juist is doordat het outliers bevat. Dit is te zien in afbeelding 9.6, waarbij de zevende seasonality erg afwijkt van de rest van de time serie. Dit kan ervoor zorgen dat de voorspelling zegt dat er minder HTTP requests zullen zijn omdat op het moment van de downtime er ook minder HTTP request waren. Om dit te voorkomen zal J-Predictor moeten controleren of de time serie overeenkomt met de voorspelling. Op het moment dat het verschil groot is vergeleken met eerdere voorspellingen betekent dit dat de time serie niet juist is en niet gebruikt moet worden voor het doen van een voorspelling.



Afbeelding 9.6 - Voorspelling met downtime

Outlier detectie is een uitgebreid onderwerp waardoor voor verdere implementatie van outlier detectie in J-Predictor een uitgebreid onderzoek gedaan moet worden. Doordat voor een gedegen onderzoek meer tijd nodig is dan beschikbaar binnen het afstudeertraject, is ervoor gekozen om dit onderzoek buiten het afstudeertraject te laten plaats vinden.

Bij outlier detectie zit de uitdaging in het bepalen wat wel of niet een outlier is. Het kan namelijk zo zijn dat een datapunt een outlier lijkt maar in werkelijkheid noise is. Daarom is het belangrijk om goed te analyseren of een datapunt daadwerkelijk een outlier is. Veel outlier detectie algoritmes geven de data een score wat aangeeft in hoeverre de data een mogelijke outlier is. De score geeft niet precies aan wat een outlier is, maar is een indicatie van wat mogelijke een outlier is, zodat de mogelijke outliers verder onderzocht kunnen worden. Daarnaast zijn er outlier detectie algoritmes die precies aangeven wat outliers zijn gebaseerd op een threshold op de outlier score van data. Aan de hand van de threshold wordt bepaald wat wel en wat niet een outlier is. Daarbij moet wel bepaald worden wat de threshold is.

Doordat het complex is om eenduidig te kunnen zeggen wat wel en niet een outlier is, is hiervoor het nodige onderzoek nodig. Daarbij komt ook dat de oplossing moet werken voor veel verschillende time series van verschillende IT infrastructuren binnen J-Layer wat betekent dat outlier detectie geautomatiseerd moet worden omdat het niet mogelijk is om handmatig 24 uur per dag de outliers te zoeken. (Aggarwal, z.d), (Kriegel, Kröger, & Zimek, 2010)

9.6. Functionele oplossing

Aan de hand van de (niet) functionele eisen en de werking van CLR en K-fold cross validation is bepaald hoe de applicatie functioneel gaat werken, waarmee het functionele gedeelte van deelvraag 4 beantwoord wordt. De functionele werking is uitgewerkt in use cases die beschrijven hoe de functies functioneel gaan werken. Aan de hand van de use cases konden de business rules worden bepaald, waarmee deelvraag 4b beantwoord is.

9.6.1. Use cases

Bij de uitwerking van de use cases is rekening gehouden met NFE18 “Alle functies van J-Predictor moeten via die via de GUI te gebruiken zijn moet ook via de API te gebruiken zijn”. De use cases zijn te zien in tabel 9.1. De uitgebreide uitwerking van de use cases zijn te vinden in Bijlage 3 – Functioneel ontwerp. Aan de hand van deze use case zijn twee proces modellen gemaakt, die te zien in in paragraaf 9.6.3 - Nieuwe manier van schalen. De modellen geven een duidelijk beeld van de werking van de oplossing en wat de taken van J-Layer en J-Predictor zijn.

Code	Use Case
UC1	Nieuwe predictive auto scaler aanmaken.
UC2	Predictor vanuit J-Layer configureren.
UC3	Nieuwe responsive auto scaler aanmaken.
UC4	Time serie data pushen naar J-Predictor.
UC5	Bepalen van het scaling plan.
UC6	Nieuwe voorspeller aanmaken via GUI.
UC7	Nieuwe voorspeller aanmaken via API.
UC8	Time serie updaten via GUI.
UC9	Time serie updaten via API.
UC10	Voorspeller configuratie via een GUI.
UC11	Voorspeller configuratie via een API.
UC12	Bepalen van seasonality via GUI.
UC13	Bepalen van seasonality via API.
UC14	Automatische bepalen van configuratie via GUI.
UC15	Automatische bepalen van configuratie via API.
UC16	Voorspelling via GUI.
UC17	Voorspelling via API.
UC18	Automatisch voorspelling.
UC19	Voorspelling accuraatheid check.
UC20	Automatisch configuratie tuning.

Tabel 9.1 - Use cases

9.6.2. Business rules

Aan de hand van de use cases konden de business rules bepaald worden door de use cases met Jitscale te bespreken en een intern bronnenonderzoek van de werking van J-Layer te doen. In tabel 9.2 - business rules zijn de business rules te zien.

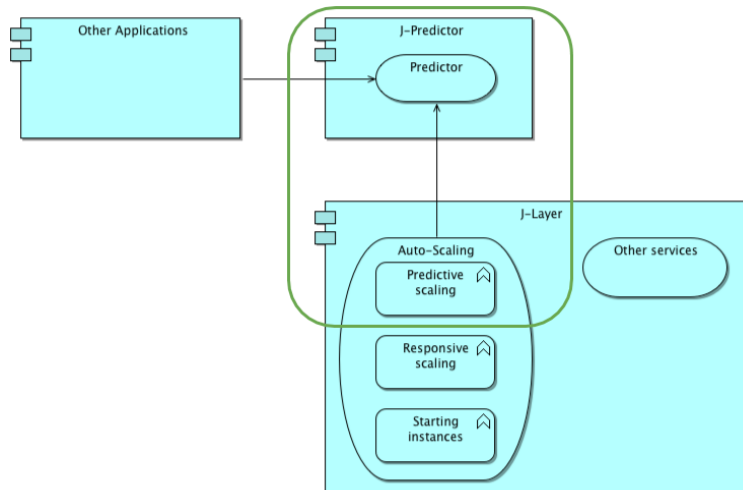
Use Case	Business rule
Algemeen	De GUI's en API's moet alleen beschikbaar zijn binnen het netwerk van Jitscale.
Algemeen	Bij een API's request moet het referentie ID meegegeven worden bij het geen wat opgevraagd of gewijzigd worden.
Algemeen	Time serie moet de volgende structuur hebben: float, float.
Algemeen	Bij het aanmaken van een nieuwe object via de API geeft J-Predictor het ID van het aangemaakte object.
Algemeen	De API moet de zelfde functies bevatten als de GUI.
UC1, UC3	Minimum instances moeten minder of gelijk zijn aan maximum instances.
UC1	De predictive auto-scaler wordt pas aangemaakt als het ID van de predictor in J-Predictor ontvangen is.
UC2	Er moeten monitoring time series beschikbaar zijn in J-Layer.
UC4	De X waarden van de time serie moet een integer zijn.
UC5	De begin periode moet lager zijn dan de eind periode.
UC6	Elke predictor heeft een unieke referentie ID.
UC7	Voordat er voorspeld kan worden met een predictor moet deze ingesteld zijn met UC10
UC9	De API request bevat time serie data.
UC10	Voorspellingsinterval moet voor minuten ≤ 60 zijn en voor uren ≤ 24 .
UC11	Voorspellingsinterval moet voor minuten ≤ 60 zijn en voor uren ≤ 24 .
UC11	Relevantie van historische data: Moet float zijn welke ≥ 0 en ≤ 100
UC11	Historische relevantie afname: Moet float zijn welke ≥ 0 en ≤ 100
UC13	J-Predictor stuurt afbeeldingen van de grafieken via de API.
UC14 UC15	Het automatisch bepalen van de configuratie gebeurt op de achtergrond.
UC16 UC17	Een voorspelling wordt weergegeven in een grafiek en als tekst.
UC18	De X en Y waarde van de voorspelling zijn beide een float.
UC18	De job scheduler doet voorspellingen op ingestelde interval.
UC19	J-Predictor houdt in de gaten of een voorspelling accuraat is en geeft hierover een waarschuwing als dat niet het geval is.

Tabel 9.2 - Business rules

9.6.3. Nieuwe manier van schalen

In hoofdstuk 6.3 is de huidige manier van schalen beschreven (IST) waar een nieuwe situatie voor bedacht moet worden. Na het bepalen van hoe de oplossing functioneel gaat werken is het mogelijk om het proces van de nieuwe manier van schalen (SOLL) te beschrijven. Dit is te zien in afbeelding 9.7.

In de nieuwe situatie zullen J-Predictor en J-Layer samenwerken aan het realiseren van de oplossing, waarbij J-Predictor de taak heeft om een voorspelling te doen en J-Layer aan de hand van de voorspelling moet bepalen hoeveel instances nodig zijn. In afbeelding 9.7 is de scope van het project te zien waarbij te zien is dat J-Predictor en een nieuwe functionaliteit binnen J-Layer genaamd Predictive scaling ontwikkeld wordt.



Afbeelding 9.7 - Project scope

J-Predictor en J-Layer hebben bij het bepalen van het aantal benodigde instances een aantal taken die ze moeten verrichten,

De taken van J-Predictor zijn:

- Het ontvangen en opslaan van de time series.
- Het analyseren van time series.
- Het doen van een voorspelling op basis van een time serie op het ingestelde tijdstip.
- Het opslaan en beschikbaar stellen van voorspellingen.

De taken van J-Layer zijn:

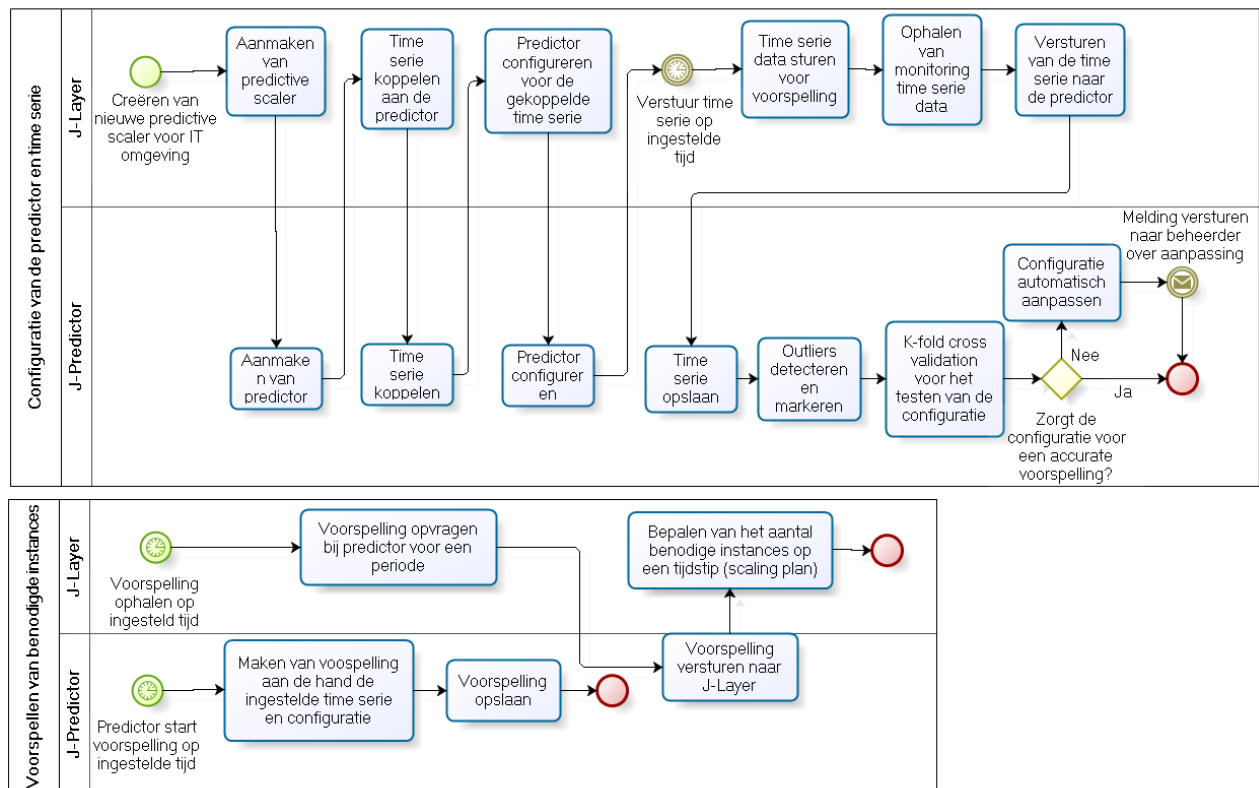
- Het aanmaken en configureren van een predictor en deze koppelen aan een IT omgeving binnen J-Layer.
- Het versturen van monitoring time series naar de J-Predictor op het ingestelde tijdstip.
- Het opvragen van de voorspelling uit J-Predictor.
- Het scaling plan bepalen aan de hand van de ingestelde instance threshold.
- Het starten van de instances volgens het scaling plan. (Dit is al mogelijk in J-Layer en valt daarom buiten de scope van dit project.)

In de afbeelding 9.8 zijn twee proces modellen te zien die gemaakt zijn aan de hand van de use cases uit paragraaf 9.6.1 - Use cases. In de proces modellen is te zien is hoe de taken van J-Layer en J-Predictor samen zorgen voor het bepalen van het aantal benodigde instances. Het realiseren van een voorspelling van het benodigde aantal instances bestaat uit het voorbereiden van de voorspelling:

- Configureren van de predictor en time serie.

En de voorspelling:

- Voorspellen van benodigde instances.



Afbeelding 9.8 - Nieuwe manier van schalen

Bij de nieuwe manier van schalen zijn er twee schermen die erg belangrijk zijn, het configureren van de predictor en het configureren van de voorspellingsmethode instellingen. Deze zullen hier toegelicht worden. In afbeelding 9.9 is te zien hoe de predictor geconfigureerd wordt. Dit wordt gedaan aan de hand van de volgende instellingen:

Prediction period: voor het bepalen hoever in de toekomst voorspeld moet worden.

Prediction interval: voor het aangeven wanneer een voorspelling gedaan moet worden. J-Layer gebruikt de interval ook voor het bepalen wanneer een nieuwe time serie verstuurd moet worden, zodat J-Layer weet wanneer er een nieuwe voorspelling opgevraagd kan worden.

Time serie used for prediction: voor het bepalen welke time serie gebruikt wordt voor de voorspellingen.

Afbeelding 9.9 - Predictor configuratie

In afbeelding 9.10 is te zien hoe de voorspellingsmethode geconfigureerd wordt. Dit formulier is een dynamisch formulier doordat het mogelijk is dat er een andere voorspellingsmethode ingesteld wordt die andere instellingen nodig heeft. De instellingen die in dit scherm te zien zijn, zijn die van CLR die beschreven zijn in hoofdstuk 9.3 - Werking van clustered linear regression.

Afbeelding 9.10 - Voorspellingsmethode configuratie

9.7. Conclusie

Uit het functionele onderzoek blijkt welke (niet) functionele eisen en business rules er zijn, die benoemd zijn in paragraaf 9.2 - Functionele en niet-functionele eisen en paragraaf 9.6.2 - Business rules. Hiermee wordt deelvraag 4a en 4b beantwoord. Bij het analyseren van de niet functionele eisen bleek vervolgens dat het een betere keuze was om de oplossing niet binnen J-Layer te implementeren maar als een losse applicatie.

Voor het beantwoorden van de vraag hoe CLR functioneel gaat werken is bepaald welke instellingen nodig zijn en waar deze voor dienen. Om under- en overfitting te voorkomen wordt K-fold cross validation gebruikt en voor outlier detectie wordt gecontroleerd in hoeverre de voorspelling afwijkt met de time serie.

Daarbij is nog een uitgebreid onderzoek nodig voor het vinden van outlier detectie mogelijkheden die te automatiseren zijn.

Aan de hand van het beantwoorden van deze vragen was het mogelijk te bepalen hoe de oplossing functioneel gaat werken wat beschreven is aan de hand van use cases en twee procesmodellen. Hiermee wordt deelvraag 4, wat is de beste oplossing is om te implementeren is, beantwoord.

10. Technischonderzoek

Na het onderzoek hoe de oplossing functioneel gaat werken is onderzocht hoe de applicatie technisch gaat werken en geïmplementeerd wordt. Dit is nodig voor het beantwoorden van deelvraag 4c en het technische gedeelte van deelvraag 4.

Deelvraag 4c: Hoe kan de oplossing geïmplementeerd worden binnen J-Layer?

Deelvraag 4: Wat is de beste oplossing om te implementeren binnen J-Layer?

Zoals uit het functioneel onderzoek blijkt zal de oplossing bestaan uit een losse applicatie genaamd J-Predictor en nieuwe functionaliteiten binnen J-Layer, dus de oplossing wordt niet alleen binnen J-Layer geïmplementeerd.

In dit hoofdstuk is onderzocht hoe de oplossing technisch gaat werken, welke methodes en software gebruikt zijn waarbij de gemaakte keuzes toegelicht worden. Voor de oplossing is er een klassendiagram gemaakt waarin de technische werking van de applicatie beschreven is, en daarbij is bepaald welk framework, programmeertaal, database, webserver, communicatiemethode tussen J-Layer en J-Predictor en beveiligingsmethode er gebruikt gaat worden.

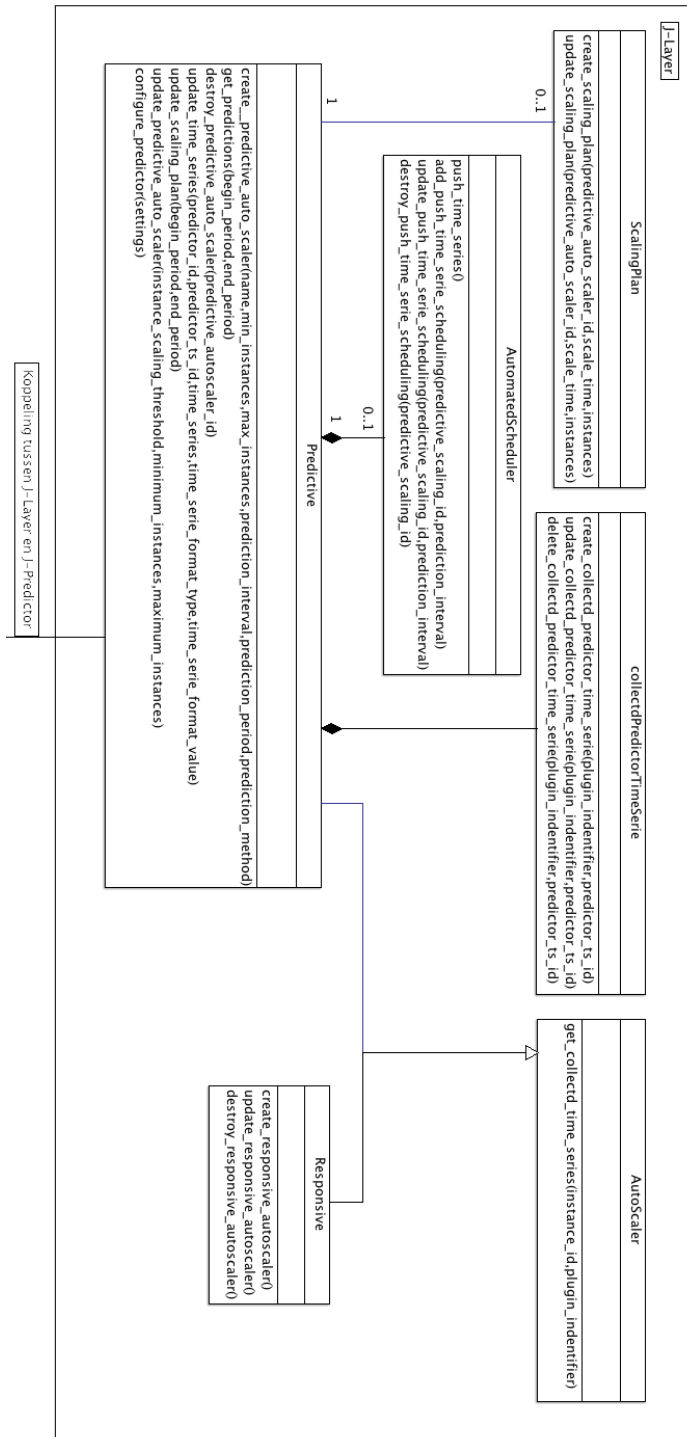
10.1. Onderzoeksmethode

Voor het onderzoek is een bronnenonderzoek gedaan waarmee keuzes gemaakt konden worden tussen de verschillende methodes en software. Voor het bepalen hoe J-Layer en J-Predictor met elkaar kunnen communiceren is een intern bronnenonderzoek gedaan en zijn medewerkers van Jitscale geïnterviewd.

10.2. Klassendiagram

In het klassendiagram (Bijlage 4 – Technisch ontwerp) zijn de klassen van J-Predictor en J-Layer te zien en met welke klassen J-Predictor en J-Layer gekoppeld zijn. Bij J-Layer is te zien dat er twee manieren van auto-scalen zijn, predictive en responsive. Responsive is de huidige manier van bij schalen waarbij gekeken wordt naar de huidige behoeften en predictive is de manier waarmee aan de hand van een voorspelling uit J-Layer bepaald wordt hoeveel instances er nodig gaan zijn. Aan de klasse Predictive zitten drie klasse gekoppeld, ScalingPlan, AutomatedScheduler en CollectdPredictorTimeSerie, wat te zien is in afbeelding 10.1.

ScalingPlan zorgt voor het bepalen van het scaling plan aan de hand van de voorspelling uit J-Predictor. CollectdPredictorTimeSerie is de klasse waar de monitoring time serie gekoppeld wordt met de time serie in J-Predictor. En AutomatedScheduler zorgt ervoor dat de monitoring time serie op het juiste moment naar J-Predictor verstuurd wordt en op het juiste moment de voorspelling uit J-Predictor opgehaald wordt voor het bepalen van het scaling plan.



Afbeelding 10.1 - J-Layer klassendiagram

Afbeelding 10.2 - J-Predictor klassendiagram

In het klassendiagram van J-Predictor, die te zien is in afbeelding 10.2 staan drie klassen centraal, Predictor, PredictionMethod en TimeSerieAnalyzer. Predictor is de klasse waarmee de voorspellingsmethode beheerd en aangestuurd wordt. Vanuit deze klasse kunnen ook de time series geanalyseerd worden met de klasse TimeSerieAnalyzer. Predictor is ook de klasse waarmee J-Layer gekoppeld is met J-Predictor. De taken van de klasse Predictor zijn het:

- ontvangen en opslaan van de time serie met de klasse TimeSeries.
- starten van een time serie analyse.
- configureren van de voorspellingsmethode.
- starten van een voorspelling met de ingestelde voorspellingsmethode.
- opslaan van de voorspellingen met de klasse Predictions.
- plotten van de grafiek van de time series met de voorspellingen die gedaan zijn.

In het klassendiagram van J-Predictor is te zien dat voor de voorspellingsmethode de strategy pattern gebruikt is welke bestaat uit de volgende klassen, Predict, PredictionMethod(interface) en ClusteredLinearRegression. Zoals ClusteredLinearRegression gekoppeld is aan de interface kunnen meerdere voorspellingsmethode gekoppeld worden. (SourceMaking, z.d)

De voorspellingsmethodes moeten allemaal de methodes predict, set_settings en get_settings hebben. Wat de methode predict doet en welke settings er ingesteld kunnen worden zal per voorspellingsmethode verschillen, maar de manier waarop de methode predict aangeroepen wordt of de instellingen ingesteld worden zal voor elke voorspellingsmethode het zelfde zijn. Daarbij heeft elke voorspellingsmethode die toegevoegd wordt aan J-Predictor de mogelijkheid om gebruik te maken van de klasse PredictionTools. De methodes in PredictionTools zijn methodes die bijdragen leveren aan het doen van voorspellingen en gebruikt zullen worden door meerdere voorspellingsmethodes. Dit kan bijvoorbeeld een berekening zijn zoals die verschillende voorspellingsmethodes nodig hebben, zoals root mean square. (Rosettacode, 2014)

Met de klasse TimeSerieAnalyzer kunnen de time series die toegevoegd zijn met Predictor geanalyseerd worden. TimeSerieAnalyzer maakt voor het doen van de analyses gebruik van de analysetools die zich bevinden in de klasse AnalyzeTools, welke ook een onderdeel is van een strategy pattern. Voor het weergeven van de resultaten van de analyse(s) maakt AnalyzeTools gebruik van de klasse plotter waarmee grafieken geplot worden van de resultaten.

10.3. Koppeling tussen J-Layer en J-Predictor

Voor de koppeling tussen J-Layer en J-Predictor wordt REST gebruikt. Er is gekozen voor REST vanwege de niet functionele eis dat alle configuratie van J-Predictor via zowel de GUI als API ingesteld moet kunnen worden. REST biedt hiervoor de oplossing doordat precies de zelfde URL van de GUI gebruikt kunnen worden voor het configureren van J-Predictor via de API. Hierdoor is geen extra methode zoals een façade klasse nodig die zorgt voor de koppeling tussen J-Predictor en externe applicaties.

Daarbij is het met REST ook mogelijk om informatie aan te leveren in verschillende formaten, zoals JSON, XML etc. Als J-Predictor informatie in JSON beschikbaar stelt, zou bijvoorbeeld de time serie op de volgende manier opgevraagd kunnen worden: GET j-predictor.jitscale.com/predictor/1/time_serie/3.json. (Sutterer, 2010), (Microformats, z.d), (Wikipedia, Representational state transfer, 2014), (SourceMaking, Facade Design Pattern, z.d.)

10.3.1. JSON

Het formaat waarin gecommuniceerd wordt tussen J-Layer en J-Predictor is JSON. De reden dat JSON gebruikt wordt is omdat JSON:

- leesbaar is voor mensen.
- het gemakkelijk te importeren is doordat de data worden opgeslagen als array's en records. Bij XML moeten de data eerst omgezet worden naar bruikbare data voordat deze geïmporteerd kunnen worden.
- JSON laat maar een beperkt aantal datatypes toe waardoor het vrijwel met elke programmeertaal werkt.
- JSON pakketten zijn kleiner dan die van XML. Hierdoor gaat het versturen van JSON pakketten sneller dan XML pakketten en er wordt minder bandbreedte, CPU en geheugen verbruikt.

(Developerfeed, 2010), (Perkins, z.d), (Mikoluk, 2014), (Wikipedia, JSON, 2014), (JSON, z.d), (Secretgeek, 2006)

10.4. Ruby on Rails

Als programmeertaal en framework is Ruby on Rails(RoR) gebruikt worden. Ruby is de programmeertaal en Rails het framework. Binnen Jitscale wordt een groot deel van de applicaties, waaronder J-Layer, ontwikkeld met RoR dus in die zin sluit deze keuze goed aan bij de werkzaamheden binnen Jitscale. J-Layer en J-Predictor zijn gescheiden applicaties maar doordat ze beide gebruik maken van RoR wordt de kans verkleind dat er onverwachts problemen ontstaan in de koppeling tussen de twee applicaties. Ook is het hiermee mogelijk, mocht het nodig zijn, code te verplaatsen van J-Layer naar J-Predictor of vice versa. Voor development heeft dit als voordeel dat de developer geen kennis hoeft te hebben van andere programmeertalen als hij/zij ontwikkeld aan de koppeling tussen J-Layer en J-Predictor. Als dat wel het geval zou zijn, zou de situatie zich voor kunnen doen dat er twee developers nodig zijn, één voor J-Layer en één voor J-Predictor.

De reden dat Ruby gebruikt wordt voor de ontwikkeling van J-Predictor is, omdat het object georiënteerd is, het duidelijke en logische syntaxis heeft, zoals 'end' voor het einde van een if statement, waardoor de code gemakkelijk te lezen en te begrijpen is. Daarbij is alles in Ruby een object waardoor het mogelijk is om complexe acties uit te voeren door het aanroepen van een simpele methode, zoals het berekenen van 5x5 met de methode `time(5.times)`, wat mogelijk is doordat het getal 5 een object is. (Ruby-lang, z.d), (Tutorialspoint, z.d), (BOJČIĆ, z.d), (Atwood, Why Ruby?, 2013)

Ruby beschikt ook over een package manager waarmee libraries genaamd gems gemakkelijk beheerd kunnen worden. Met gems is het mogelijk om Ruby uit te breiden met nieuwe functies, afhankelijk van wat er nodig is. Een voorbeeld van zo'n gem is 'active record'. Met active record wordt het mogelijk om een database tabellen te koppelen aan klassen en het zorgt voor een abstracte data laag. Hierdoor kan de database (MySQL, PostgreSQL, etc) veranderd worden zonder dat hiervoor de code gewijzigd hoeft te worden. Door gebruik te maken van active record hoeven er namelijk geen queries geschreven te worden. In plaats daarvan wordt er gebruik gemaakt van methodes die active record vertaald naar queries, zoals "new" voor het toevoegen van een nieuwe record in de database of "find" voor het zoeken van een record. (Rubyonrails, Active Record, z.d), (Rubydoc, z.d)

De reden dat Rails als framework gekozen is heeft meerdere redenen, Rails:

1. is een full-stack framework.
2. is gebaseerd op het model view controller (MVC) principe.
3. focussed op het gebruik van software engineering patterns, zoals don't repeat yourself (DRY), convention over configuration (CoC) en active record pattern.
4. heeft een RESTful design.

(Wikipedia, Ruby on Rails, 2014), (Sendhil, 2012), (Wikipedia, Active record pattern, 2014), (Wikipedia, Convention over configuration, 2014), (Wikipedia, Don't repeat yourself, 2014), (Rubyonrails, Getting Started with Rails, z.d)

Een full-stack framework is een framework dat hulpmiddel is voor het development op alle lagen van de UI-laag tot de data laag. Dit heeft als voordeel dat er geen verschillende tools gebruikt hoeven worden voor development op verschillende lagen. (Wikipedia, Multilayered architecture, 2014)

Bij de development van J-Predictor wordt er gewerkt via het MVC principe dus een framework dat gebaseerd is op dit principe sluit hier goed bij aan. Met het MVC principe wordt er voor gezorgd dat de code leesbaarder en herbruikbaar is. Dit wordt gedaan door het scheiden van de presentatie laag (view), applicatie logica laag (controller) en data laag (model). (Atwood, Understanding Model-View-Controller, 2008)

Daarbij maakt Rails ook gebruik van het DRY principe waarbij ervoor gezorgd wordt dat code herbruikbaar is. Een simpel voorbeeld hiervoor is dat er een partial template gemaakt wordt van een formulier waarmee dit formulier op meerdere plaatsen gerenderd kan worden. En met het CoC principe zorgt Rails ervoor dat standaard handelingen niet geconfigureerd hoeven te worden, dit wordt automatisch gedaan door Rails. Een voorbeeld hiervan is dat bij het laden van de methode 'update' (voor het updaten van een object) in een controller automatisch de 'update' template geladen wordt. Hierdoor is er in de code geen verwijzing naar de template nodig. (Rubyonrails, Layouts and Rendering in Rails, z.d),

Het design van Rails is gebaseerd op REST wat aansluit bij de koppeling tussen J-Layer en J-Predictor waar REST voor gebruikt wordt. Met REST wordt een HTTP method (GET, POST, PUT and DELETE) gekoppeld aan een URL waarmee bepaald wordt wat de URL moet doen, zoals informatie ophalen met 'GET' of informatie versturen met 'POST'. Daarbij is het met REST mogelijk om op verschillende manieren informatie te presenteren, waaronder HTML, XML en JSON. (Wirdemann & Baustert, 2007)

10.5. RServe

Voor het doen van analyses en voorspellingen op time serie is het voor J-Predictor nodig om gebruik te maken van statische functies, waaronder:

1. (Partial) Autocorrelatie
2. Root-mean-square berekening
3. Differentiatie

Het is mogelijk om deze functie zelf te programmeren maar het is ook mogelijk om R te gebruiken voor het uitvoeren van deze functies. R beschikt over een uitgebreid aantal statistische functies, waaronder de bovengenoemde functies. Daarbij is het met R ook mogelijk de functies te gebruiken in Ruby door gebruik te maken van de applicatie Rserve en de Ruby gem RServe. Rserve zorgt voor de koppeling tussen R en andere applicaties. Doordat Rserve de communicatie doet met R zorgt Rserve ervoor dat de communicatie snel verloopt. Dit is ook één van de redenen dat Rserve gebruikt wordt voor J-Predictor, de andere reden is dat er actief ontwikkeld wordt aan Rserve waardoor Rserve werkt met recente versies van Ruby. (R-project, z.d), (Sheong, 2010)

10.6. Database

De database die gebruikt wordt voor J-Predictor en J-Layer is MySQL. Voor J-Layer wordt al MySQL gebruikt dus om die reden is voor verdere ontwikkeling van J-Layer dezelfde database gebruikt. Voor J-Predictor is ook gekozen voor het gebruik van MySQL, omdat MySQL:

1. schaalbaar is.
2. snel is.
3. goede beveiliging mogelijkheden heeft.

MySQL is schaalbaar doordat het beschikt over de mogelijkheid om te repliceren. Door replicatie zijn er meerdere kopieën van de database waardoor de load verdeeld kan worden over meerdere databases waardoor het benaderen van de database snel blijft. (mysql.com, Chapter 17 Replication, z.d), (Wikipedia, Load (computing), 2014), (Anchor, z.d), (Tezer, 2014)

MySQL beschikt ook over een uitgebreid aantal security features, waaronder:

- Het hebben van een copy database waarmee dataverlies voorkomen wordt.
- Gebruikers toelaten op basis van IP, wachtwoord en permissies.
- Het hebben van SSL en two-factor authentication beveiliging.

(Dobrzanski, 2013), (Digicert, z.d)

10.6.1. Database engine

Bij gebruik van MySQL moet er gekozen worden tussen de database engine die gebruikt gaat worden. Voor J-Predictor is gekozen voor MyISAM in plaats van InnoDB. De reden hiervoor is dat MyISAM in tegenstelling tot InnoDB:

- gebruikte ruimte gelijk vrijgeeft na het verwijderen van de data.
- sneller is.

InnoDB heeft echter wel een beter crash recovery en foreign key support, maar doordat er gebruikt gemaakt van backups en de foreign key relaties afgehandeld worden door Active Records in Rails is MyISAM een betere keuze. (mysql.com, InnoDB File-Per-Table Mode, z.d), (Nieto, 2013), (Joshi, 2013), (Yang, 2009), (mysql.com, The MyISAM Storage Engine, z.d)

10.7. Strategy pattern

Voor het implementeren van de voorspellingsmethodes en analysetools is ervoor gekozen om deze te implementeren aan de hand van de strategy pattern. De reden hiervoor is omdat ze herbruikbaar moeten zijn en omdat er nieuwe voorspellingsmethodes en analysetools onafhankelijk van elkaar toegevoegd moeten kunnen worden. Daarbij moet het ook mogelijk zijn om op runtime te selecteren welke voorspellingsmethode of analysetool gebruikt wordt. In Bijlage 4 – Technisch ontwerp wordt uitgelegd hoe de strategy pattern geïmplementeerd wordt voor de voorspellingsmethodes en analysetools. (SourceMaking, z.d), (Alm, 2013), (Wikibooks, 2014)

10.8. Beveiliging

Voor de beveiliging van J-Predictor is gekozen om Htaccess te gebruiken. De reden voor deze keuze is omdat J-Predictor binnen het netwerk van Jitscale komt te draaien waarbij J-Predictor automatisch afgeschermd is van de buiten wereld. Hierbij is het niet nodig om J-Predictor af te schermen aan de hand van een gebruikersaccount en is het alleen nodig om bepaalde IP's binnen het netwerk toegang te geven, zoals het IP van J-Layer. Dit kan gedaan worden met Htaccess en daarbij heeft Htaccess ook de mogelijkheid om alsnog J-Predictor af te schermen aan de hand van een gebruikersnaam en wachtwoord. (Wikipedia, .htaccess, 2014), (htaccess-guide.com, What is .htaccess?, z.d), (htaccess-guide.com, Deny visitors by IP address, z.d), (htaccess-guide.com, Password protection, z.d)

10.9. Webserver

Als webserver wordt Apache gebruikt. De redenen voor het gebruik van Apache zijn:

- Htaccess beveiliging support.
- betrouwbaar omdat het al een lange tijd bestaat waardoor de meest bugs eruit zijn.
- Apache veel functies bevat, zoals: caching, proxying, load balancing en authenticatie en autorisatie (htaccess).

Daarbij heeft Apache een minder goede performance dan Nginx maar voor J-Predictor is de performance ruim voldoende doordat er niet duizenden requests per seconden afgehandeld hoeven te worden. Nginx heeft als nadeel dat het minder functionaliteiten bevat doordat de focus ligt op performance. (Venezia, 2014), (ISSAC, 2013), (ocean, 2013), (Wikivs, 2014), (agency, 2013)

10.10. Prototype

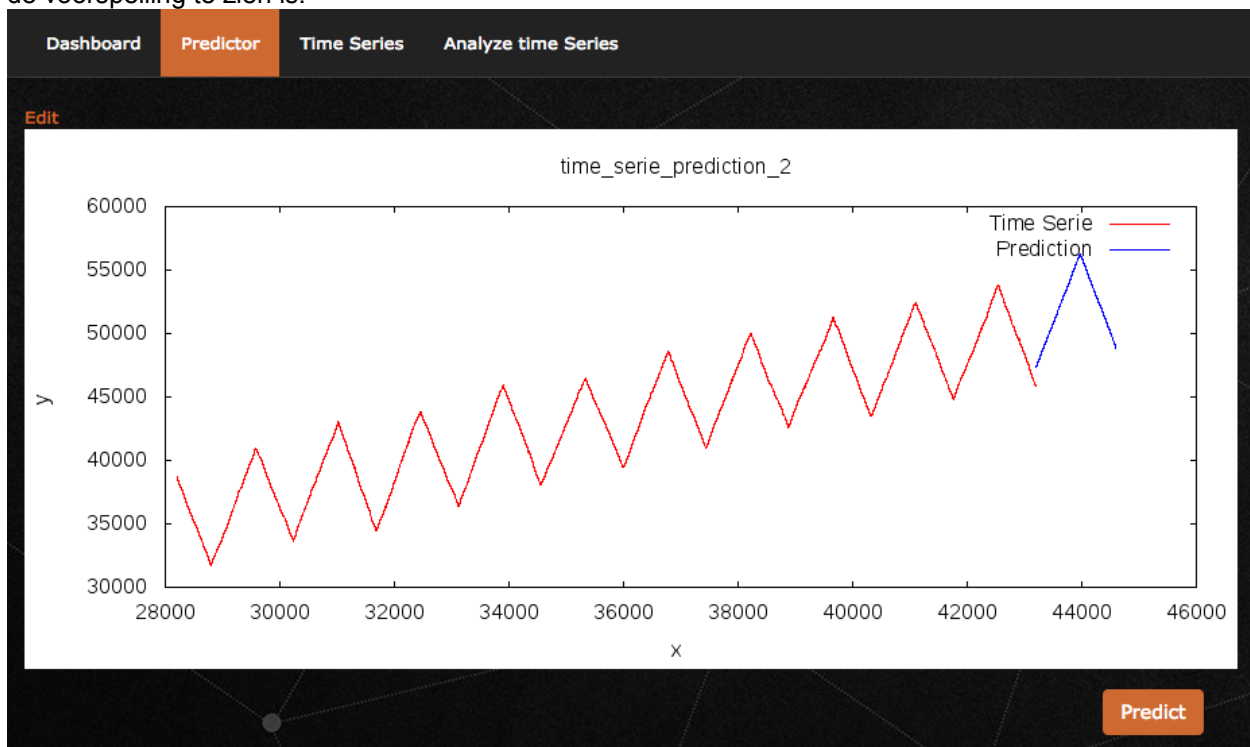
Het prototype is ontwikkeld zoals te zien is in de mockups in Bijlage 3 – Functioneel ontwerp. Bij de ontwikkeling beschikt het prototype nog niet over alle eisen, deze zullen later geïmplementeerd worden. Zo is het met het prototype mogelijk om een voorspelling te doen aan de hand van één seasonaliteit en time serie, en nog niet met meerdere seasonalities en time series. Bij de ontwikkeling is er wel rekening meegehouden dat dit later geïmplementeerd gaat worden, want er kunnen al wel meerdere time series toegevoegd worden.

De focus voor het prototype ligt als eerste bij de ontwikkeling van J-Predictor, en daarna bij de extra functionaliteiten waar J-Layer over moet beschikken. Wel is er bij de ontwikkeling van het prototype rekening gehouden met de koppeling tussen J-Layer en J-Predictor waarvoor een RESTful design en JSON nodig is.

Het prototype van J-Predictor beschikt over de volgende functionaliteiten:

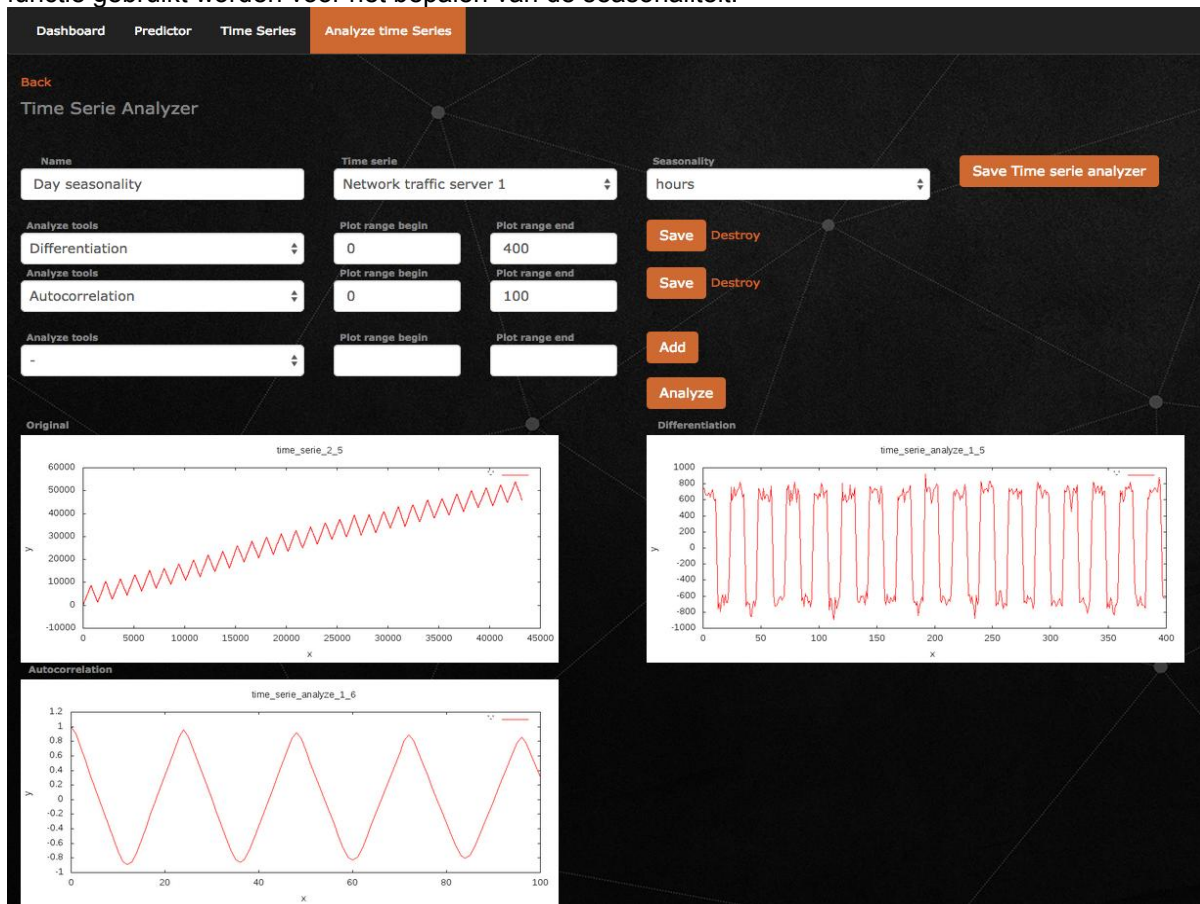
- Clustered lineaire regression als voorspellingsmethode.
- Predictor aanmaken.
- Time series incrementeel toevoegen.
- Time serie analyse tool met (partial) autocorrelation en differentiation.
- Time serie configureren met de instellingen van de gekozen voorspellingsmethode.
- Voorspelling doen aan de hand van één time serie.
- Voorspelling opvragen voor een bepaalde periode.
- Grafieken voor het visualiseren van de time series.
- Automatisch de instellingen voor de time serie configuratie bepalen.

Afbeelding 10.3 toont de pagina waar handmatig de voorspelling uitgevoerd wordt, en de time serie met de voorspelling te zien is.



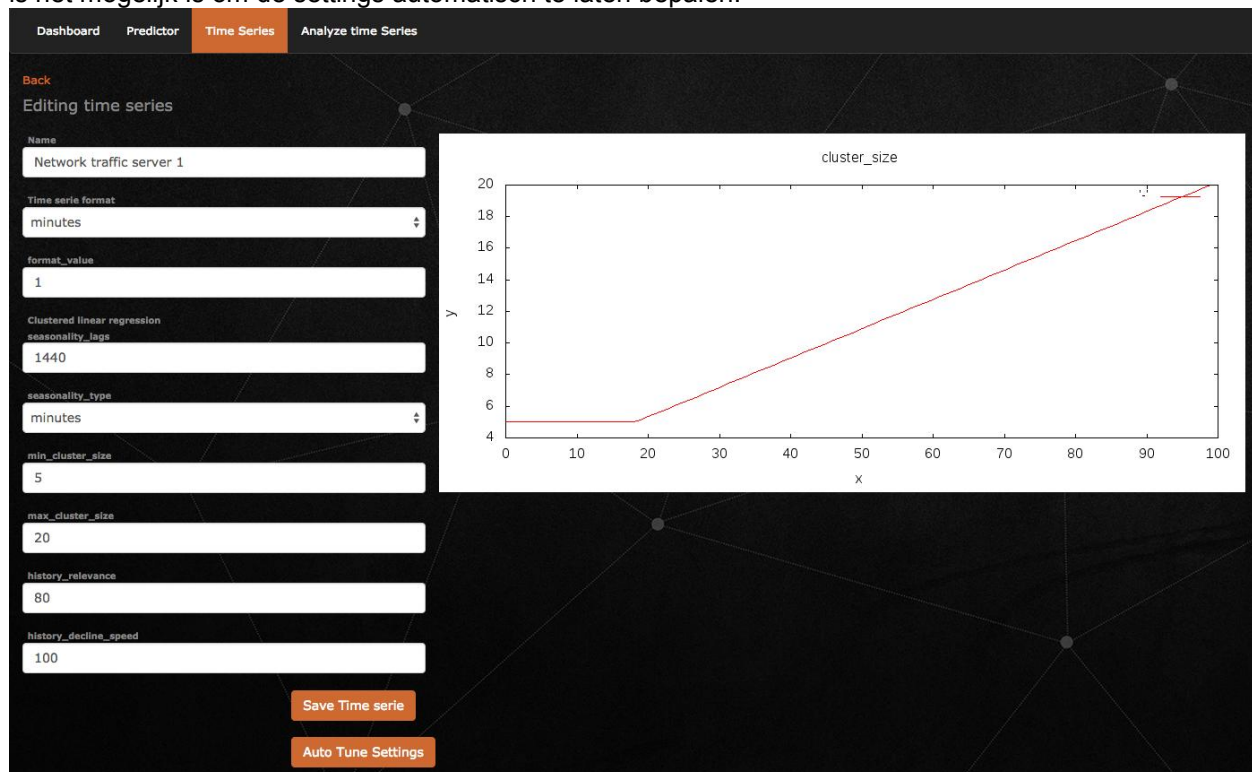
Afbeelding 10.3 – Voorspelling

Afbeelding 10.4 toont de pagina waar de time serie geanalyseerd kan worden, onder andere voor het bepalen van de seasonality. In de afbeelding is te zien hoe de differentiation functie en autocorrelation functie gebruikt worden voor het bepalen van de seasonaliteit.



Afbeelding 10.4 - Time serie analyzer

Afbeelding 10.5 toont de pagina waar de settings van een time serie geconfigureerd worden. Daarbij zijn een aantal settings afhankelijk van de voorspellingsmethode die gebruikt wordt, in dit geval is dat Clustered linear regression. Aan de hand van de “cluster size” grafiek kan afgeleid worden wat het format van de clusters zal zijn bij het doen van Clustere linear regression, en met de knop “Auto Tune Settings” is het mogelijk is om de settings automatisch te laten bepalen.



Afbeelding 10.5 - Time serie configuratie

10.11. Conclusie

In het technisch onderzoek is onderzocht hoe de oplossing technisch gaat werken. Voor het beantwoorden van deelvraag 4c is geselecteerd welke methodes en software gebruikt gaan worden. Hierbij wordt als framework Ruby on Rails gebruikt, als database MySQL en als webserver Apache. Daarbij wordt REST in combinatie met JSON gebruikt voor de communicatie tussen J-Layer en J-Predictor.

Voor deelvraag 4 was bij de functionele oplossing gekozen om een losse applicatie te ontwikkelen, bij het technische gedeelte van de oplossing moest de beste technische oplossing bepaald worden. Hiervoor is gekozen om J-Predictor te ontwikkelen in Ruby on Rails waarbij de voorspellingsmethode geïmplementeerd wordt met een strategy design pattern, en de koppeling tussen J-Layer en J-Predictor gedaan wordt met REST en JSON.

11. Implementatie

Voor het beantwoorden van deelvraag 4c “Hoe kan de oplossing geïmplementeerd worden binnen J-Layer?” is onderzocht hoe de IT omgeving gaat werken. De oplossing wordt niet geïmplementeerd binnen J-Layer dus er is een nieuwe IT omgeving nodig. Daarbij moet de oplossing overgedragen worden aan de medewerkers van Jitscale die het moeten gaan beheren en ermee moeten gaan werken. Voor de implementatie van J-Predictor is een implementatieplan opgesteld dat te vinden is in Bijlage 5 – Implementatieplan. In het implementatieplan wordt ingegaan op de vraag hoe de server setup gaat worden, hoe de backup en monitoring geregeld is, welke software nodig is en hoe J-Predictor overgedragen wordt aan de support- en beheerafdeling van Jitscale.

11.1. Onderzoeksmethode

Voor het onderzoeken van de implementatie is een bronnenonderzoek gedaan voor het bepalen hoe de server setup het beste geïmplementeerd kan worden.

11.2. Server setup

Voor de server setup is ervoor gekozen om J-Predictor in de Amazon cloud te draaien, waarbij gebruik gemaakt wordt van Amazon instances, Relational Database Service (RDS), Elastic Load Balancing (ELB) en cloud storage (S3). De reden dat gekozen is om J-Predictor in de cloud te draaien is, omdat J-Predictor hiermee schaalbaar is doordat er extra instances gestart kunnen worden als er meer rekenkracht nodig is voor het doen van voorspellingen. En de reden dat gekozen is voor Amazon is omdat Amazon betrouwbaar is en doordat ze een groot aanbod aan cloudservices hebben, zoals loadbalancing (ELB), schaalbare database (RDS) en gedeelde opslag tussen instances (S3). De betrouwbaarheid blijkt uit het feit dat Jitscale Amazon al een aantal jaren gebruikt zonder noemenswaardige problemen. (Amazon, Amazon EC2 Instances, z.d), (Amazon, Amazon RDS, z.d), (Amazon, Elastic Load Balancing, z.d), (Amazon, Amazon S3, z.d), (Amazon, Amazon EC2, z.d)

11.3. Backup

Het voordeel van het gebruik van boven genoemde Amazon dienst is dat ze allemaal backup mogelijkheden hebben. Hierdoor is het niet nodig om een extra backup functionaliteit in te richten.

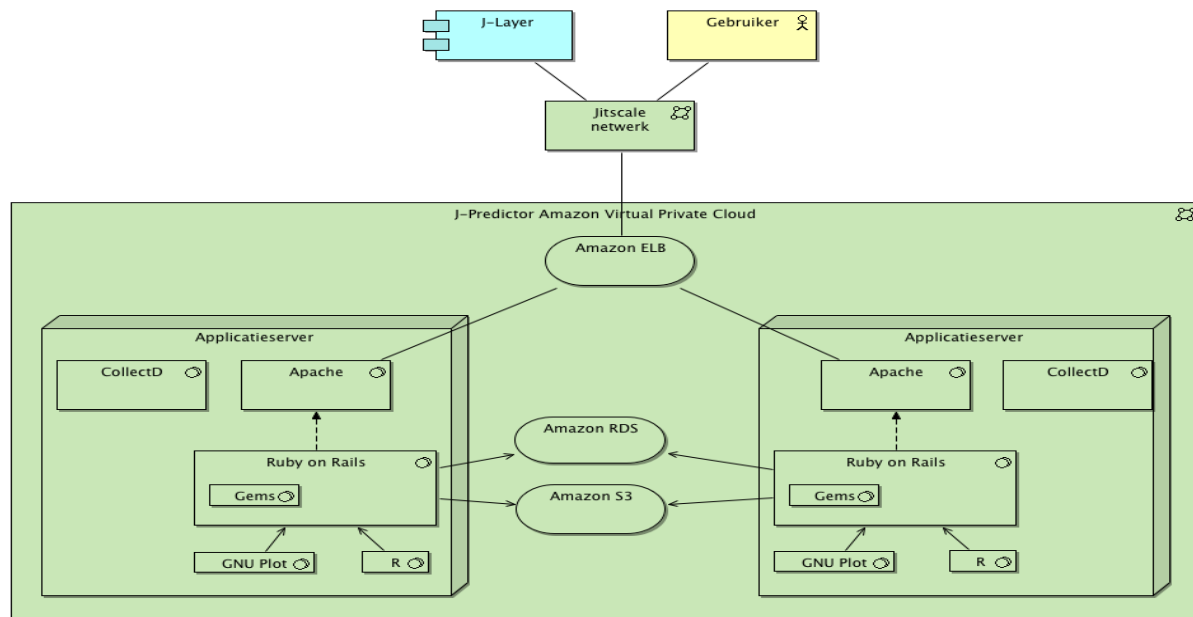
11.4. Monitoring

De monitoring wordt gedaan door CollectD. CollectD is een gratis monitoring tool dat statistische informatie over de server kan vergaren zoals load- en geheugenverbruik. Daarbij slaat CollectD de gegevens op in RRD bestanden welk bedoeld zijn voor het genereren van time serie grafieken en dus ook goed te gebruiken zijn voor J-Predictor voor het doen van voorspellingen. (Collectd, z.d), (Wikipedia, RRDtool, 2014)

11.5. IT infrastructuur

Om een duidelijk beeld te scheppen van de infrastructuur van J-Predictor is in afbeelding 11.1 de infrastructuur te zien. In de afbeelding is te zien:

- hoe de software met elkaar gekoppeld is.
- hoe de ELB zorgt voor het verdelen van het verkeer tussen twee applicatieservers.
- hoe de applicatieserver gebruik maakt van een gedeelde RDS en S3 opslag.
- dat J-Layer en gebruikers via het Jitscale netwerk verbinding kunnen maken met J-Predictor.



Afbeelding 11.1 - J-Predictor infrastructuur

12. Conclusie

Voor Jitscale moest een oplossing bedacht en ontwikkeld worden waarmee binnen J-Layer ervoor gezorgd wordt dat er altijd precies genoeg instances gestart zijn en dit ook kosten besparend werkt. Dit is onderzocht door het beantwoorden van vier deelvragen. De antwoorden op deze deelvragen samen zorgen voor het antwoord op de hoofdvraag: Op welke manier kan er binnen J-Layer voor gezorgd worden dat er op elk moment precies genoeg instances gestart zijn en het kosten besparend is?

Deelvraag 1: Wat is J-Layer?

Antwoord: J-Layer is een applicatie die ontwikkeld is door Jitscale voor het beheren van cloud computing IT-infrastructuren waarbij gebruik gemaakt wordt van IT diensten van verschillende cloud providers aan de hand van API's. Met J-Layer is Jitscale in staat snel nieuwe instances te starten en te stoppen afhankelijk van de huidige behoeften.

Uit het onderzoek blijkt dat er één oplossing is het beantwoorden van deelvraag 2 en 3.

Deelvraag 2: Op welke wijze kan ervoor gezorgd worden dat er precies genoeg instances gestart zijn met J-Layer?

Deelvraag 3: Op welke wijze kan ervoor gezorgd worden dat er kosten bespaard worden met J-Layer?

Antwoord: Door te voorspellen hoeveel instances er nodig zijn op een bepaald tijdstip. Daardoor kunnen instances eerder gestart en gestopt worden, omdat met zekerheid gezegd kan worden of een instance wel of niet nodig is.

Deelvraag 4: Wat is de beste oplossing om te implementeren binnen J-Layer?

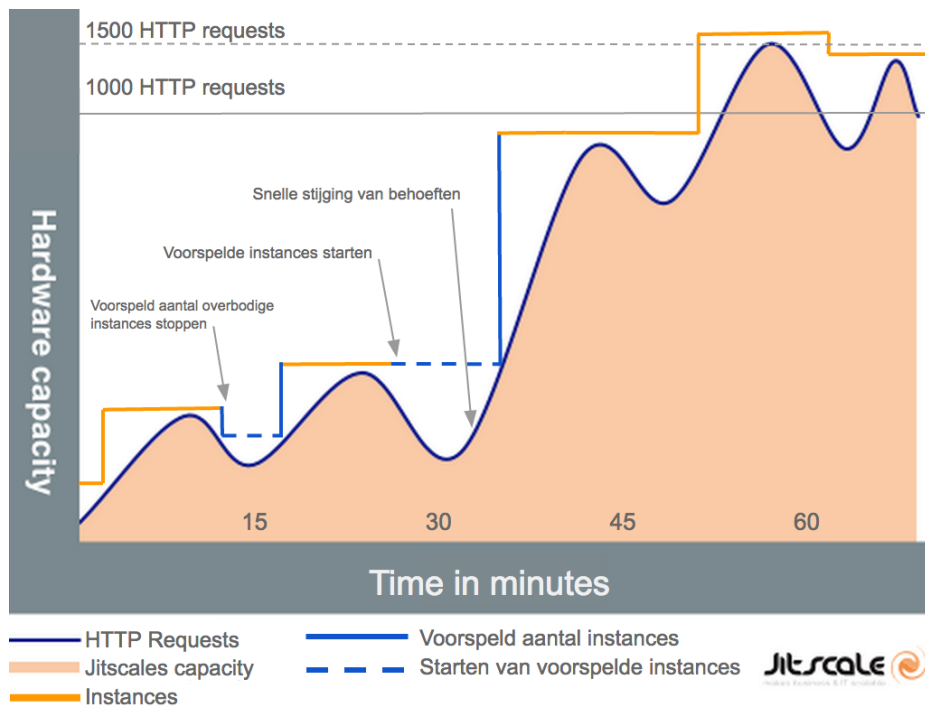
Antwoord: Bij het onderzoeken van deelvraag 4 en met de resultaten uit deelvraag 2 en 3 blijkt dat de beste oplossing is om een applicatie, genaamd J-Predictor, te ontwikkelen naast J-Layer waarmee de behoeften voorspeld wordt. Hierbij is J-Predictor ontwikkeld aan de hand van de designs en software die beschreven worden in Bijlage 4 – Technisch ontwerp. De oplossing wordt dus niet binnen J-Layer geïmplementeerd.

Na het beantwoorden van de deelvragen kan de hoofdvraag beantwoord worden.

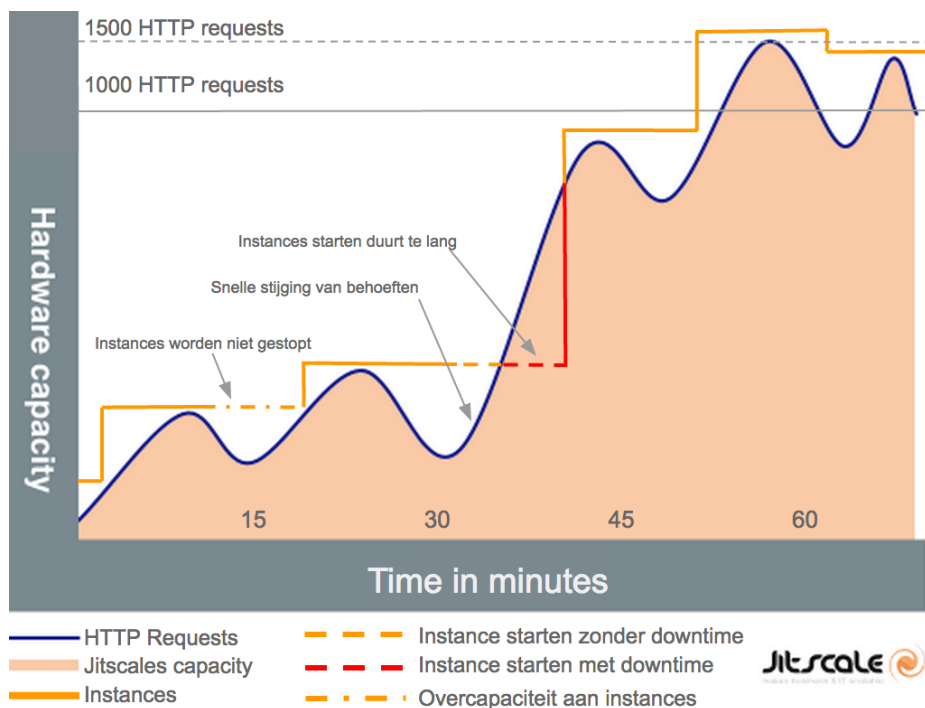
Hoofdvraag: Op welke manier kan er binnen J-Layer voor gezorgd worden dat er op elk moment precies genoeg instances gestart zijn en het kosten besparend is?

Antwoord: Door het ontwikkelen van een applicatie waarmee de behoeften in de toekomst voorspeld worden, zodat J-Layer weet hoeveel instances er nodig zijn en precies op tijd een instance kan starten en stoppen. En door een instance precies op tijd te starten wordt downtime en bij behorende kosten voorkomen, en door een instance precies op tijd te stoppen hoeft een instance niet onnodig lang te draaien waarmee overcapaciteit voorkomen wordt en kosten bespaard worden.

In afbeelding 12.1 is te zien hoe met de nieuwe manier van schalen ervoor gezorgd wordt dat er altijd genoeg instances gestart zijn en instances precies op tijd gestopt kunnen worden. Dit in vergelijking met de huidige manier van schalen (afbeelding 12.2) waarbij downtime ontstaat doordat instances te laat gestart worden en er een overcapaciteit aan instances is.



Afbeelding 12.1 - Nieuwe manier van schalen



Afbeelding 12.2 - Huidige manier van schalen

13. Aanbevelingen

In dit project zijn er een aantal punten in mindere mate behandeld die bij uitbreiding van J-Predictor een bijdrage kunnen leveren aan het doen van accurate voorspellingen. Mijn aanbeveling is om bij verdere ontwikkeling het mogelijk te maken om met meerdere seasonalities en time series een voorspelling te kunnen doen. Daarbij is outlier detectie een onderwerp dat tijdens dit project wel behandeld is maar voor een goede implementatie is meer tijd nodig is. Daarom raad ik aan om hier verder onderzoek naar te doen.

Uit de voorspellingsmethode selectie bleek dat ook Triple exponential smoothing en Box-Jenkins voorspellingsmethodes goede voorspellingsmethodes zijn om te implementeren. Daarom raad ik ook aan deze methodes verder te onderzoeken voor implementatie binnen J-Predictor. Daarbij is tijdens dit project niet gekeken om voorspellingsmethodes te combineren voor het doen van een voorspelling maar dit is voor in de toekomst zeker de moeite waard zijn om te onderzoeken. Hiermee kunnen bijvoorbeeld met verschillende voorspellingsmethodes voorspellingen gedaan worden op een time serie waarbij het gemiddelde van de voorspellingen als uitkomst gebruikt wordt.

14. Evaluatie

Terug kijkend op dit project is dit project succesvol verlopen en met een functioneel en technisch ontwerp, implementatie plan en prototype is het project met succes afgerond. Tijdens dit project zat de grootste uitdaging in het onderzoeken van voorspellingsmethodes. Hierbij heb ik veel nieuwe kennis opgedaan over welke voorspellingsmethodes er zijn, hoe ze werken en wat er nodig is voor het doen van een accurate voorspelling. Met de kennis was het uiteindelijk mogelijk om te beslissen welke voorspellingsmethode geïmplementeerd moest worden.

Bij het project is scrum gebruikt als ontwikkelmethode wat erg heeft bijgedragen aan het succesvol afronden van dit project. Door gebruik te maken van scrum en de gebruikelijke scrum sprints was het mogelijk het project stap voor stap te doorlopen waarmee het mogelijk werd om tijdens het project gemakkelijk bij te sturen. Zo was het met scrum mogelijk om in plaats van de oplossing te implementeren binnen J-Layer een losse applicatie te ontwikkelen, doordat regelmatig met Jitscale de voortgang besproken werd.

In het begin van het project was het lastig in te schatten hoeveel tijd er nodig zou zijn voor het onderzoek. Dit kwam mede doordat het doen van voorspellingen een complex onderwerp is. Voor iemand die niet bekend is met voorspellingsmethodes zijn veel termen onbekend waardoor het begrijpen van de werking van een voorspellingsmethode lastig was, wat bij mij het geval was. Na veel onderzoek en lezen werden mij de begrippen en de werking van de voorspellingsmethodes duidelijk waarna ik de kennis kon verwerken in het voorspellingsmethode selectie onderzoek. Na het onderzoeken van de voorspellingsmethodes was er nog ruim tijd over voor de ontwikkeling van de oplossing.

15. Bibliografie

- agency, T. o. (2013, Maart). *APACHE VS NGINX {PERFORMANCE COMPARISON}*. Opgeroepen op Oktober 7, 2014, van The organic agency: <http://www.theorganicagency.com/apache-vs-nginx-performance-comparison/>
- Aggarwal, C. C. (z.d). *Outlier analysis*. Opgeroepen op Oktober 7, 2014, van Charu Aggarwal: <http://www.charuaggarwal.net/outlierbook.pdf>
- Alm, J. (2013, Augustus 16). *Combining Strategy Pattern and Spring*. Opgeroepen op Oktober 7, 2014, van CapTech: <http://captechconsulting.com/blog/jens-alm/combining-strategy-pattern-and-spring>
- Alonso, A. M., & Garcia-Martos, C. (2012, Juli). *Time Series Analysis*. Opgeroepen op Oktober 2, 2014, van Escuela tecnica superior de ingenieros industriales: <http://www.etsii.upm.es/ingor/estadistica/Carol/TSAtema10petten.pdf>
- Amazon. (z.d). *Amazon EC2*. Opgeroepen op Oktober 7, 2014, van Amazon: <http://aws.amazon.com/ec2/>
- Amazon. (z.d). *Amazon EC2 Instances*. Opgeroepen op Oktober 7, 2014, van Amazon: <http://aws.amazon.com/ec2/instance-types/>
- Amazon. (z.d). *Amazon RDS*. Opgeroepen op Oktober 7, 2014, van Amazon: <http://aws.amazon.com/rds/>
- Amazon. (z.d). *Amazon S3*. Opgeroepen op Oktober 7, 2014, van Amazon: <http://aws.amazon.com/s3/>
- Amazon. (z.d). *Elastic Load Balancing*. Opgeroepen op Oktober 7, 2014, van Amazon: <http://aws.amazon.com/elasticloadbalancing/>
- Anchor. (z.d). *MySQL vs. PostgreSQL comparison*. Opgeroepen op Oktober 7, 2014, van Anchor: http://www.anchor.com.au/hosting/dedicated/mysql_vs_postgres
- Andrews, B. H., Dean, M. D., Swain, R., & Cole, C. (2013, Augustus). *Building ARIMA and ARIMAX Models for Predicting Long-Term Disability Benefit Application Rates in the Public/Private Sectors*. Opgeroepen op 10 4, 2014, van Society of Actuaries: <https://www.soa.org/Files/Research/Projects/research-2013-arma-arimax-ben-appl-rates.pdf>
- Appenda. (z.d). *Definition of Cloud Instance (Single & Multi)*. Opgeroepen op Oktober 2, 2014, van <http://appenda.com/library/glossary/definition-cloud-instance-single-multi/>
- Arnosti, N. A., & Kalita, J. K. (z.d). *Cutting Plane Training for Linear Support Vector*. Opgeroepen op Oktober 2, 2014, van Stanford University: <http://web.stanford.edu/~narnosti/Arnosti-Kalita.pdf>
- Athanasopoulos, G., & Silva, A. d. (2010, Februari 22). *Multivariate exponential smoothing for forecasting tourist arrivals to Australia and New Zealand*. Opgeroepen op 10 4, 2014, van Monash University: <http://www.buseco.monash.edu.au/ebs/pubs/wpapers/2009/wp11-09.pdf>
- Atwood, J. (2008, Mei 5). *Understanding Model-View-Controller*. Opgeroepen op Oktober 7, 2014, van Codinghorror: <http://blog.codinghorror.com/understanding-model-view-controller/>
- Atwood, J. (2013, Maart 22). *Why Ruby?* Opgeroepen op Oktober 7, 2014, van Codinghorror: <http://blog.codinghorror.com/why-ruby/>

- Au, S. T., Ma, G.-Q., & Yeung, S.-N. (z.d). *Automatic Forecasting of Double Seasonal Time Series with Applications on Mobility Network Traffic Prediction*. Opgeroepen op 10 4, 2014, van AT&T Researchers: http://www.research.att.com/techdocs/TD_100381.pdf
- Beliakov, G., Kelarev, A., & Yearwood, J. (2011, Oktober 2). *Robust artificial neural networks and outlier detection*. Opgeroepen op Oktober 2, 2014, van arXiv: <http://arxiv.org/pdf/1110.0169.pdf>
- Bizarro David. (2011, Januari 7). *How to diagnose problems with your statistical learning algorithm*. Opgeroepen op Oktober 2, 2014, van Anti Anti: <http://antianti.org/?p=175>
- BOJČIĆ, K. (z.d). *After Two Decades of Programming, I Use Rails*. Opgeroepen op Oktober 7, 2014, van Toptal: <http://www.toptal.com/ruby-on-rails/after-two-decades-of-programming-i-use-rails>
- BusinessDictionary. (z.d). *Threshold*. Opgeroepen op Juni 27, 2014, van Businessdictionary: <http://www.businessdictionary.com/definition/threshold.html>
- Cai, E. (2014, Maart 20). *Machine Learning Lesson of the Day – Overfitting and Underfitting*. Opgeroepen op Oktober 2, 2014, van Statsblogs: <http://www.statsblogs.com/2014/03/20/machine-learning-lesson-of-the-day-overfitting-and-underfitting/>
- Collectd. (z.d). *collectd – The system statistics collection daemon*. Opgeroepen op Oktober 7, 2014, van Collectd: <https://collectd.org/>
- COORAY, T. (z.d). *ANLYSIS AND FORECASTING OF MULTIPLE SEASONAL TIME SERIES MODELS*. Opgeroepen op 10 4, 2014, van Department of Mathematics University of Moratuwa: <http://www.math.mrt.ac.lk/TMJAC/Papers/Paper%20%2023.pdf>
- Crone, S. F. (2005, December 15). *Forecasting with Artificial Neural Networks*. Opgeroepen op Oktober 2, 2014, van Neural Forecasting: http://www.neural-forecasting.com/Downloads/EVIC05_tutorial/EVIC'05%20Slides%20-%20Forecasting%20with%20Neural%20Networks%20Tutorial%20SFCrone.pdf
- Danny Yuan, N. J. (2013, December 4). *Scryer: Netflix's Predictive Auto Scaling Engine - Part 2*. Opgeroepen op Oktober 2, 2014, van Netflix: <http://techblog.netflix.com/2013/12/scryer-netflix-predictive-auto-scaling.html>
- data, T. s. (2013, Maart 26). *General regression and over fitting*. Opgeroepen op Oktober 2, 2014, van Wordpress - The shape of data: <http://shapeofdata.wordpress.com/2013/03/26/general-regression-and-over-fitting/>
- Developerfeed. (2010, Maart 2). *What are the benefits of using JSON over XML?* Opgeroepen op Oktober 7, 2014, van Developerfeed: <http://www.developerfeed.com/what-are-benefits-using-json-over-xml>
- Digicert. (z.d). *Behind the Scenes of SSL Cryptography*. Opgeroepen op Oktober 7, 2014, van Digicert: <https://www.digicert.com/ssl-cryptography.htm>
- Dobrzanski, M. (2013, Februari 6). *MySQL Security: Overview of MySQL security features*. Opgeroepen op Oktober 7, 2014, van PSCE: <http://www.psce.com/blog/2013/02/06/mysql-security-overview-of-mysql-security-features/>

- DTREG. (sd). *SVM - Support Vector Machines*. Opgeroepen op Oktober 2, 2014, van DTREG: <http://www.dtreg.com/svm.htm>
- Duke. (z.d). *Data transformations and forecasting models: what to use and when*. Opgeroepen op Oktober 11, 2014, van Duke : <http://people.duke.edu/~rnau/whatuse.htm>
- Fortmann-Roe, S. (2012, Juni). *Understanding the Bias-Variance Tradeoff*. Opgeroepen op Oktober 2, 2014, van Scott fortmann-roe: <http://scott.fortmann-roe.com/docs/BiasVariance.html>
- Gaer, E. V., Ceulemans, E., & Mechelen, I. V. (sd). *An extensive evaluation of the performance of clusterwise regression and its multilevel extension*. Opgeroepen op Oktober 2, 2014, van Inria: https://www.rocq.inria.fr/axis/COMPSTAT2010/slides/slides_437.pdf
- Gerardnico. (z.d). *Data Mining - Support Vector Machines (SVM) algorithm*. Opgeroepen op Oktober 2, 2014, van Gerardnico: http://gerardnico.com/wiki/data_mining/support_vector_machine#overfitting
- GUO, Y.-M., RAN, C.-B., LI, X.-L., MA, J.-Z., & ZHANG, L. (2013). *Weighted prediction method with multiple time series using multi-kernel least squares support vector regression*. Opgeroepen op 10 4, 2014, van http://yadda.icm.edu.pl/yadda/element/bwmeta1.element.baztech-article-BAT6-0015-0020/c/httpwww_ein_org_plsitesdefaultfiles2013-02-16.pdf
- Gutierrez-Osuna, R. (z.d). *Lecture 13: Validation*. Opgeroepen op Oktober 7, 2014, van http://research.cs.tamu.edu/prism/lectures/iss/iss_113.pdf
- Hans-Peter Kriegel, P. K. (2010). *Outlier Detection Techniques*. Opgeroepen op Oktober 2, 2014, van [siam: http://www.siam.org/meetings/sdm10/tutorial3.pdf](http://www.siam.org/meetings/sdm10/tutorial3.pdf)
- Hogben, D. C. (2009, november). *Cloud computing security risk assessment*. Opgeroepen op Juni 7, 2014, van Enisa: <http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-riskassessment/>
- htaccess-guide.com. (z.d). *Deny visitors by IP address*. Opgeroepen op Oktober 7, 2014, van htaccess-guide: <http://www.htaccess-guide.com/deny-visitors-by-ip-address/>
- htaccess-guide.com. (z.d). *Password protection*. Opgeroepen op Oktober 7, 2014, van htaccess-guide: <http://www.htaccess-guide.com/password-protection/>
- htaccess-guide.com. (z.d). *What is .htaccess?* Opgeroepen op Oktober 7, 2014, van Htaccess-guide: <http://www.htaccess-guide.com/>
- Hyndman, R. J. (2010, Oktober 4). *Why every statistician should know about cross-validation*. Opgeroepen op Oktober 7, 2014, van HYNDSIGHT: <http://robjhyndman.com/hyndsight/crossvalidation/>
- Institute, S. (2012). *SAS/ETS 12.1 User's Guide*. Opgeroepen op 10 4, 2014, van Google books: <http://books.google.nl/books?id=OE0UfAhit4kC&pg=PA28&lpg=PA28&dq=box+jenkins+for+%22ARIMAX%22&source=bl&ots=pgBwBsda-v&sig=IVpcPUo9kOFaohXDUikKgp1QDDA&hl=en&sa=X&ei=EEf4U9-2Ds3mPKa3gNgK&ved=0CE8Q6AEwBg#v=onepage&q&f=false>

- Investopedia. (z.d). *Moving Average - MA*. Opgeroepen op Oktober 2, 2014, van Investopedia: <http://www.investopedia.com/terms/m/movingaverage.asp>
- Investopedia. (z.d). *Neural Network*. Opgeroepen op Oktober 2, 2014, van Investopedia: <http://www.investopedia.com/terms/n/neuralnetwork.asp>
- ISSAC, L. P. (2013, November 20). *Nginx Vs Apache: Nginx Basic Architecture and Scalability*. Opgeroepen op Oktober 7, 2014, van The geek stuff: <http://www.thegeekstuff.com/2013/11/nginx-vs-apache/>
- Jitscale. (z.d). *Solution for every demand*. Opgeroepen op 10 5, 2014, van Jitscale: <https://www.jitscale.com/cloud-solutions>
- Joshi, N. (2013, September 25). *How to reclaim space in InnoDB when innodb_file_per_table is ON*. Opgeroepen op Oktober 7, 2014, van Percona: http://www.percona.com/blog/2013/09/25/how-to-reclaim-space-in-innodb-when-innodb_file_per_table-is-on/
- JSON. (z.d). *JSON: The Fat-Free Alternative to XML*. Opgeroepen op Oktober 7, 2014, van JSON: <http://www.json.org/xml.html>
- Kalekar, P. S. (2004, December 6). *Time series Forecasting using Holt-Winters Exponential Smoothing*. Opgeroepen op Oktober 2, 2014, van Kanwal Rekhi School of Information Technology: http://www.it.iitb.ac.in/~praj/acads/seminar/04329008_ExponentialSmoothing.pdf
- Kourentzes, N., & Crone, S. F. (z.d). *Automatic modelling of neural networks for time series prediction – in search of a uniform time series prediction – in search of a uniform*. Opgeroepen op 10 4, 2014, van Sven F. Cron: [www.sven-crone.com/papers/Kourentzes,%20Crone%20\(2008\)%20Automatic%20modelling%20of%20neural%20networks%20for%20Time%20Series%20Prediction%20-%20in%20search%20for%20a%20uniform%20methodology%20across%20varying%20time%20frequencies%20-%20ESTSP08%20V1.5.pdf](http://www.sven-crone.com/papers/Kourentzes,%20Crone%20(2008)%20Automatic%20modelling%20of%20neural%20networks%20for%20Time%20Series%20Prediction%20-%20in%20search%20for%20a%20uniform%20methodology%20across%20varying%20time%20frequencies%20-%20ESTSP08%20V1.5.pdf)
- Kriegel, H.-P., Kröger, P., & Zimek, A. (2010). *Outlier Detection Techniques*. Opgeroepen op Oktober 7, 2014, van Department of Computer Science: <http://www.dbs.ifi.lmu.de/~zimek/publications/KDD2010/kdd10-outlier-tutorial.pdf>
- Laborator, I. T. (z.d). *Seasonality*. Opgeroepen op Oktober 2, 2014, van Information Technology Laboratory: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc443.htm>
- Laboratory, I. T. (sd). *Example of Triple Exponential Smoothing*. Opgeroepen op Oktober 2, 2014, van Information Technology Laboratory: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc436.htm>
- Laboratory, I. T. (z.d). *Triple Exponential Smoothing*. Opgeroepen op Oktober 2, 2014, van Information Technology Laboratory: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc435.htm>
- Little, A. (2012). *Evaluating the Effectiveness of Groundwater Conservation Districts in the Gulf Coast Aquifer of Texas*. Opgeroepen op 10 4, 2014, van Department of Statistics Texas A&M University: http://w1.stat.tamu.edu/sheathermemorial/PDF/ALittle_MSProject.pdf

- Martí, R., & El-Fallahi, A. (2002, Juli 2014). *Multilayer Neural Networks: An Experimental*. Opgeroepen op Oktober 2, 2014, van <http://www.uv.es/~rmarti/paper/docs/nn2.pdf>
- MathWorks. (z.d). *cvpartition*. Opgeroepen op Oktober 7, 2014, van MathWorks: <http://www.mathworks.nl/help/stats/cvpartition.html>
- Microformats. (z.d). *URL Conventions*. Opgeroepen op Oktober 7, 2014, van Microformats: <http://microformats.org/wiki/rest/urls>
- Mikoluk, K. (2014, Augustus 16). *JSON vs XML: How JSON Is Superior To XML*. Opgeroepen op Oktober 7, 2014, van Udemy: <http://www.udemy.com/blog/json-vs-xml/>
- Moore, A. W. (z.d). *Cross-validation for detecting and preventing overfitting*. Opgeroepen op Oktober 7, 2014, van Auton Lab: <http://www.autonlab.org/tutorials/overfit10.pdf>
- Multivariate Exponential Smoothing: prediction, signal extraction and control*. (sd). Opgeroepen op 10 4, 2014, van European University Institute: <http://www.eui.eu/Documents/DepartmentsCentres/Economics/Researchandteaching/Conferences/TSEConference-NewDevelopmentsinTimeSeries/Sbrana.pdf>
- mysql.com. (z.d). *Chapter 17 Replication*. Opgeroepen op Oktober 7, 2014, van MySQL: <http://dev.mysql.com/doc/refman/5.6/en/replication.html>
- mysql.com. (z.d). *InnoDB File-Per-Table Mode*. Opgeroepen op Oktober 7, 2014, van MySQL: <http://dev.mysql.com/doc/refman/5.7/en/innodb-multiple-tablespaces.html>
- mysql.com. (z.d). *The MyISAM Storage Engine*. Opgeroepen op Oktober 7, 2014, van MySQL: <http://dev.mysql.com/doc/refman/5.5/en/myisam-storage-engine.html>
- Neal Wagner, Z. M. (2011, April 6). *Intelligent techniques for forecasting multiple time series in real-world systems*. Opgeroepen op Oktober 2, 2014, van The university of Adelaide: <https://cs.adelaide.edu.au/~zbyszek/Papers/p51.pdf>
- Nguyen, H. H., & Chan, C. W. (2003, Oktober 27). *Multiple neural networks for a long term time series forecast*. Opgeroepen op 10 4, 2014, van http://avellano.fis.usal.es/~lalonso/compt_soft/articulos/timeseriesprediction.pdf
- Nieto, M. A. (2013, Augustus 20). *Why is the ibdata1 file continuously growing in MySQL?* Opgeroepen op Oktober 7, 2014, van Percona: <http://www.percona.com/blog/2013/08/20/why-is-the-ibdata1-file-continuously-growing-in-mysql/>
- ocean, D. (2013, April 30). *How To Set Up HTTP Authentication With Nginx On Ubuntu 12.10*. Opgeroepen op Oktober 7, 2014, van Digital ocean: <https://www.digitalocean.com/community/tutorials/how-to-set-up-http-authentication-with-nginx-on-ubuntu-12-10>
- Okkels, C. B. (2014, Juni 1). *Financial Forecasting*. Opgeroepen op Oktober 2, 2014, van University of copenhagen: http://qdev.nbi.ku.dk/student_theses/pdf_files/Financial_Forecasting_-_MSc_Thesis_-_Christian_Birch_Okkels.pdf
- Perkins, L. (z.d). *Why JSON will continue to push XML out of the picture*. Opgeroepen op Oktober 7, 2014, van Appfog: <https://blog.appfog.com/why-json-will-continue-to-push-xml-out-of-the-picture/>

- Rackspace. (z.d). *Definitie van cloudcomputing*. Opgeroepen op Juni 7, 2014, van Rackspace: http://www.rackspace.nl/cloud-computing/definition?utm_medium=ppc_gen
- Reference, M. O. (2009). *Outlier*. Opgeroepen op Oktober 2, 2014, van Math Open Reference: <http://www.mathopenref.com/outlier.html>
- ReStore. (z.d). *restore*. Opgeroepen op 10 4, 2014, van ReStore: http://www.restore.ac.uk/srme/www/fac/soc/wie/research-new/srme/modules/mod3/module_3_-_multiple_linear_regression.pdf
- Rosettacode. (2014, September 1). *Averages/Root mean square*. Opgeroepen op Oktober 7, 2014, van Rosettacode: http://rosettacode.org/wiki/Averages/Root_mean_square
- R-project. (z.d). *R-project*. Opgeroepen op Oktober 7, 2014, van R-project: <http://www.r-project.org/>
- Rubydoc. (z.d). *Active Record – Object-relational mapping put on rails*. Opgeroepen op Oktober 7, 2014, van Rubydoc: <http://rubydoc.info/gems/activerecord/4.1.5/frames>
- Ruby-lang. (z.d). *About Ruby*. Opgeroepen op Oktober 7, 2014, van Ruby-lang: <https://www.ruby-lang.org/en/about/>
- Rubyonrails. (z.d). *Active Record*. Opgeroepen op Oktober 7, 2014, van Rubyonrails: <http://api.rubyonrails.org/classes/ActiveRecord/Base.html>
- Rubyonrails. (z.d). *Getting Started with Rails*. Opgeroepen op Oktober 7, 2014, van Rubyonrails: http://guides.rubyonrails.org/v4.0.6/getting_started.html
- Rubyonrails. (z.d). *Layouts and Rendering in Rails*. Opgeroepen op Oktober 7, 2014, van Rubyonrails: http://edgeguides.rubyonrails.org/layouts_and_rendering.html
- SAS. (sd). *The ARIMA Procedure*. Opgeroepen op 10 4, 2014, van University of Montreal: <http://www.dms.umontreal.ca/~duchesne/chap7.pdf>
- Scikit-learn. (z.d). *Novelty and Outlier Detection*. Opgeroepen op Oktober 2, 2014, van Scikit-learn: http://scikit-learn.org/stable/modules/outlier_detection.html
- SDART. (z.d). *Forecasting and Prediction Algorithms*. Opgeroepen op Juni 30, 2014, van sdart: http://www.sdart.co.uk/forecasting_and_prediction_algorithms
- Secretgeek. (2006, Juli 22). *What is JSON: the 3 minute JSON Tutorial*. Opgeroepen op Oktober 7, 2014, van Secretgeek: http://secretgeek.net/json_3mins
- Sendhil. (2012, Oktober 22). *Full-Stack frameworks vs. Non Full-Stack frameworks*. Opgeroepen op Oktober 7, 2014, van Codingarchitect: <http://codingarchitect.wordpress.com/2012/10/22/full-stack-frameworks-vs-non-full-stack-frameworks/>
- Sheong, C. S. (2010). *Ruby and R*. Opgeroepen op Oktober 7, 2014, van Slideshare: <http://www.slideshare.net/sausheong/rubyand-r>
- Software, N. S. (z.d). *The Box-Jenkins*. Opgeroepen op Oktober 2, 2014, van NCSS Statistical Software: http://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/NCSS/The_Box-Jenkins_Method.pdf

- SourceMaking. (z.d). *Strategy Design Pattern*. Opgeroepen op Oktober 7, 2014, van SourceMaking: http://sourcemaking.com/design_patterns/strategy
- SourceMaking. (z.d.). *Facade Design Pattern*. Opgeroepen op 10 11, 2014, van SourceMaking: http://sourcemaking.com/design_patterns/facade
- Statistics, L. (z.d). *Multiple Linear Regression*. Opgeroepen op 10 4, 2014, van LSA Statistics: <http://dept.stat.lsa.umich.edu/~kshedden/Courses/Stat401/Notes/401-multreg.pdf>
- Stellwagen, E. (z.d). *Forecasting 101: Box-Jenkins Forecasting*. Opgeroepen op Oktober 2, 2014, van Forecast pro: <http://www.forecastpro.com/Trends/forecasting101June2012.html>
- Stergiou, C. (z.d). *What is a Neural Network?* Opgeroepen op Oktober 2, 2014, van Imperial College London: http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/cs11/article1.html
- Suhartono. (2011). *Time Series Forecasting by using Seasonal Autoregressive Integrated Moving Average: Subset, Multiplicative or Additive Model*. Opgeroepen op 10 4, 2014, van Science publications: <http://thescipub.com/PDF/jmssp.2011.20.27.pdf>
- Sutterer, N. (2010, Oktober 28). *Rails Misapprehensions: CRUD is not REST*. Opgeroepen op Oktober 7, 2014, van Apotomo: <http://nicksda.apotomo.de/2010/10/rails-misapprehensions-crud-is-not-rest/>
- Taylor, J. W. (2003). *Short-Term Electricity Demand Forecasting Using Double Seasonal Exponential Smoothing*. Opgeroepen op 10 4, 2014, van University of Oxford: <http://users.ox.ac.uk/~mast0315/ExpSmDoubleSeasonal.pdf>
- Term, M. (2011). *A Study of Sales Data using Box-Jenkins ARIMA Techniques*. Opgeroepen op Oktober 2, 2014, van Department of Statistics: <http://www.stats.ox.ac.uk/~burke/Autocorrelation/AP2%20Wk8%20Sample%20Report.pdf>
- Tezer, O. (2014, Mei 30). *SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems*. Opgeroepen op Oktober 7, 2014, van DigitalOcean: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
- Tomas Cipra, T. H. (2011). *Exponential smoothing for time series with outliers*. Opgeroepen op Oktober 2, 2014, van Kybernetika: <http://www.kybernetika.cz/content/2011/2/165/paper.pdf>
- Tutorialspoint. (z.d). *Ruby Syntax*. Opgeroepen op Oktober 7, 2014, van Tutorialspoint: http://www.tutorialspoint.com/ruby/ruby_syntax.htm
- UMBRELLA, R. P. (2013, Juli 29). *Seventh framework programme*. Opgeroepen op 10 4, 2014, van Umbrella project: www.e-umbrella.eu/download/82
- Venezia, P. (2014, Juni 16). *Apache or Nginx? Go with both*. Opgeroepen op Oktober 7, 2014, van Infoworld: <http://www.infoworld.com/article/2607928/data-center/apache-or-nginx--go-with-both.html>
- wiki, I. I. (z.d). *Cloud provider*. Opgeroepen op Oktober 2, 2014, van Wikia: http://itlaw.wikia.com/wiki/Cloud_provider

Wikibooks. (2014, September 22). *Programmeren in Java/Design Patterns/Strategy Pattern*. Opgeroepen op Oktober 7, 2014, van Wikibooks: http://nl.wikibooks.org/wiki/Programmeren_in_Java/Design_Patterns/Strategy_Pattern

Wikipedia. (2013, September 12). *Cloud API*. Opgeroepen op Juni 26, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Cloud_API

Wikipedia. (2014, September 30). *.htaccess*. Opgeroepen op Oktober 7, 2014, van Wikipedia: <http://en.wikipedia.org/wiki/.htaccess>

Wikipedia. (2014, Oktober 2). *Active record pattern*. Opgeroepen op Oktober 7, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Active_record_pattern

Wikipedia. (2014, Oktober 2). *Artificial intelligence*. Opgeroepen op Oktober 2, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Artificial_intelligence

Wikipedia. (2014, Oktober 1). *Autoregressive integrated moving average*. Opgeroepen op Oktober 2, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average

Wikipedia. (2014, Augustus 6). *Bayesian information criterion*. Opgeroepen op Oktober 2, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Bayesian_information_criterion

Wikipedia. (2014, Oktober 7). *Convention over configuration*. Opgeroepen op Oktober 7, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Convention_over_configuration

Wikipedia. (2014, September 10). *Coupling (computer programming)*. Opgeroepen op Oktober 7, 2014, van Wikipedia: [http://en.wikipedia.org/wiki/Coupling_\(computer_programming\)](http://en.wikipedia.org/wiki/Coupling_(computer_programming))

Wikipedia. (2014, September 2014). *Don't repeat yourself*. Opgeroepen op Oktober 7, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Don%27t_repeat_yourself

Wikipedia. (2014, Augustus 24). *Extrapolatie*. Opgeroepen op Oktober 7, 2014, van Wikipedia: <http://nl.wikipedia.org/wiki/Extrapolatie>

Wikipedia. (2014, Februari 13). *IT-infrastructuur*. Opgeroepen op Oktober 2, 2014, van Wikipedia: <http://nl.wikipedia.org/wiki/IT-infrastructuur>

Wikipedia. (2014, September 29). *JSON*. Opgeroepen op Oktober 7, 2014, van Wikipedia: <http://en.wikipedia.org/wiki/JSON>

Wikipedia. (2014, September 27). *Linear regression*. Opgeroepen op Oktober 2, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Linear_regression

Wikipedia. (2014, September 9). *Load (computing)*. Opgeroepen op Oktober 7, 2014, van Wikipedia: [http://en.wikipedia.org/wiki/Load_\(computing\)](http://en.wikipedia.org/wiki/Load_(computing))

Wikipedia. (2014, September 24). *Mean squared error*. Opgeroepen op Oktober 7, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Mean_squared_error

Wikipedia. (2014, Februari 27). *Multilayered architecture*. Opgeroepen op Oktober 7, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Multilayered_architecture

- Wikipedia. (2014, Oktober 1). *Outlier*. Opgeroepen op Oktober 2, 2014, van Wikipedia:
<http://en.wikipedia.org/wiki/Outlier>
- Wikipedia. (2014, September 9). *Overfitting*. Opgeroepen op Oktober 2, 2014, van Wikipedia:
<http://en.wikipedia.org/wiki/Overfitting>
- Wikipedia. (2014, April 7). *Plug-in (computing)*. Opgeroepen op Juni 29, 2014, van
http://en.wikipedia.org/wiki/Plug-in_%28computing%29
- Wikipedia. (2014, Oktober 4). *Representational state transfer*. Opgeroepen op Oktober 7, 2014, van
Wikipedia: http://en.wikipedia.org/wiki/Representational_state_transfer
- Wikipedia. (2014, Augustus 17). *RRDtool*. Opgeroepen op Oktober 7, 2014, van Wikipedia:
<http://en.wikipedia.org/wiki/RRDtool>
- Wikipedia. (2014, Oktober 5). *Ruby on Rails*. Opgeroepen op Oktober 7, 2014, van Wikipedia:
http://en.wikipedia.org/wiki/Ruby_on_Rails
- Wikipedia. (2014, September 29). *Time series*. Opgeroepen op Oktober 2, 2014, van Wikipedia:
http://en.wikipedia.org/wiki/Time_series
- Wikipedia. (2014, Juni 10). *Virtual private server*. Opgeroepen op Juni 26, 2014, van Wikipedia:
http://en.wikipedia.org/wiki/Virtual_private_server
- Wikivs. (2014, Augustus 20). *Apache vs nginx*. Opgeroepen op Oktober 7, 2014, van Wikivs:
http://www.wikivs.com/wiki/Apache_vs_nginx
- Wirdemann, R., & Baustert, T. (2007, Maart 26). *RESTful Rails Development*. Opgeroepen op Oktober 7, 2014, van b-simple: http://b-simple.de/download/restful_rails_en.pdf
- Yang, Y. (2009, September 2). *MySQL Engines: InnoDB vs. MyISAM – A Comparison of Pros and Cons*.
Opgeroepen op Oktober 7, 2014, van Kavoir: <http://www.kavoir.com/2009/09/mysql-engines-innodb-vs-mysam-a-comparison-of-pros-and-cons.html>
- Yu, L., Chen, H., Wang, S., & Lai, K. K. (2008). *Evolving Least Squares Support Vector Machines for Stock Market Trend Mining*. Opgeroepen op Oktober 2, 2014, van Unitec:
http://moodle.unitec.ac.nz/pluginfile.php/91205/mod_resource/content/0/Papers/04632148.pdf

Plan van aanpak

J-Layer auto-scaling



Inhoudsopgaven

Inleiding	2
Probleem en doelstellingen	2
Aanleiding	2
J-Layer	6
Probleem	8
Doelstelling	8
Opdracht	9
Hoofdvraag	10
Deelvragen	10
Op te leveren producten	12
Software ontwikkeling	12
Project planning	12
Projectrisico's	12
Projectorganisatie	13
Bedrijfs en persoonsgegevens	13
Te raadplegen literatuur	14
Bronvermelding	16

Bijlagen 1. Contract afstudeeropdracht

Inleiding

Het afstudeerproject zal plaats gaan vinden bij Jitscale. Jitscale is een hosting provider die gespecialiseerd is in het hosten van bedrijfskritische IT infrastructuur. Met bedrijfskritisch wordt bedoeld dat de IT infrastructuur cruciaal is voor het draaiend houden van het bedrijf, wat betekent dat zonder de IT infrastructuur het voor het bedrijf bijna tot niet mogelijk is te bestaan. (Wikipedia, (2014), IT-infrastructuur)

Omdat Jitscale weet dat het cruciaal is voor een bedrijf dat de bedrijfskritische IT infrastructuur altijd beschikbaar moet zijn heeft Jitscale als motto “garanteed 100% availability”. Dit is wat Jitscale belooft aan de klant en waar Jitscale dag in dag uit naar streeft.

Dit motto is ook de aanleiding van dit project. Met dit project is het doel ervoor te zorgen dat IT infrastructuur 100% beschikbaar blijven.

Ik ben momenteel werkzaam op de afdeling development waarbij mijn taken zijn, het ontwikkelen van de applicaties voor intern gebruik en ter verbetering van de primaire werkzaamheden van Jitscale.

Probleem en doelstellingen

Aanleiding

Jitscale maakt gebruik van verschillende manieren van hosting. De hosting manieren die gebruikt worden zijn hosting van IT infrastructuur op dedicated servers en op virtual private servers(VPS) via cloud computing.

(Wikipedia, (2014), Virtual private server)

(Wikipedia, (2014), Dedicated hosting service)

(Cloud Computing, (2014), Cloud Computing)

Dedicated servers zijn servers die gekocht zijn door onze klanten of door Jitscale en die Jitscale beheert in datacenters. Bij dedicated servers moet er van te voren bepaald worden hoeveel servers er nodig zijn. Als er op de server bijvoorbeeld een website komt te draaien die gemiddeld zorgt voor 1000 HTTP requests per seconde met pieken van 1500 HTTP requests per seconden dan moeten er een server aangeschaft worden die de piek van 1500 HTTP requests per seconde kan afhandelen. Een HTTP request wordt gedaan als een bezoeker een pagina van een website opvraagt. (Rve.org.uk, (z.d), What's in an HTTP request?)

Het nadeel van dedicated servers is dat als er minder dan 1500 requests per seconde zijn de server niet al zijn resources gebruikt zoals de central processing unit (CPU) en het geheugen. Op dat moment is het financieel voordeliger om een server te hebben welke precies het aantal requests op dat moment kan afhandelen en geen overbodige resources heeft. De resources bepalen namelijk de kosten van de server, hoe meer

resources hoe duurder de server wordt. (The web hosting show, (2007), What are Server Resources?), (Wikipedia, (2014), Central processing unit)

Een nadeel van dedicated servers is ook de aanschaf en installatietijd, omdat het aanschaffen en installeren van een dedicated server langer dan een dag kan duren. Hierdoor is het niet mogelijk extra resources beschikbaar te krijgen als er onverwachts meer requests gedaan worden dan de huidige server kan afhandelen wat voor downtime zorgt. Onder downtime wordt verstaan het niet in staat zijn de primaire functie te vervullen, wat in dit geval het hosten van een website is waarbij HTTP requests afgehandeld moeten worden. (Wikipedia. (2014). Downtime).

Op het moment dat er downtime is zorgt dit voor kosten door verschillende redenen. Zo kan het zijn dat de eigenaar van de website potentiële klanten misloopt doordat de klanten de site niet kunnen bereiken. Maar het kan ook zijn dat de website langer down is dan dat Jitscale heeft afgesproken met de klant in de service level agreement (SLA). In dat geval kost het Jitscale ook geld, omdat Jitscale dan een vergoeding moet geven aan de klant voor het niet nakomen van de afspraak. (Wikipedia, (2014), Service-level agreement)

Deze nadelen kunnen opgelost worden door gebruik te maken van cloud computing. Cloud computing is een on-demand service model waarmee IT diensten geleverd worden. Dit zijn IT diensten zoals servers, storage, netwerken en applicaties. Deze diensten kunnen op ieder moment van de dag binnen een paar minuten gestart en gestopt worden met minimale beheerlast waarbij alleen betaald hoeft te worden zolang er gebruik wordt gemaakt van de dienst. Er is minder beheerlast dan bij dedicated servers omdat er bij cloud computing geen rekening gehouden hoeft te worden met het aanschaffen en installeren van de benodigde hard- of software (zoals servers, netwerk switches en routers, bekabeling en software licenties), dit wordt namelijk geregeld door de cloud provider. (It law wiki, (z.d), Cloud provider)

Een server in de cloud wordt ook wel een instance of VPS genoemd. Zo'n instance of VPS is een virtuele server welke gehost en verhuurd wordt door een cloud provider en waarbij de huurder zelf helemaal verantwoordelijk is voor het beheer van de virtuele server. Zo'n instance of VPS heeft een vaste hoeveelheid resources tot zijn beschikking van één of meer onderliggende servers. De onderliggende server deelt zijn resources dus met één of meerdere instances/VPS.

(Wikipedia, (2014), Virtual private server)

(Rackspace. (z.d.). Definitie van cloudcomputing)

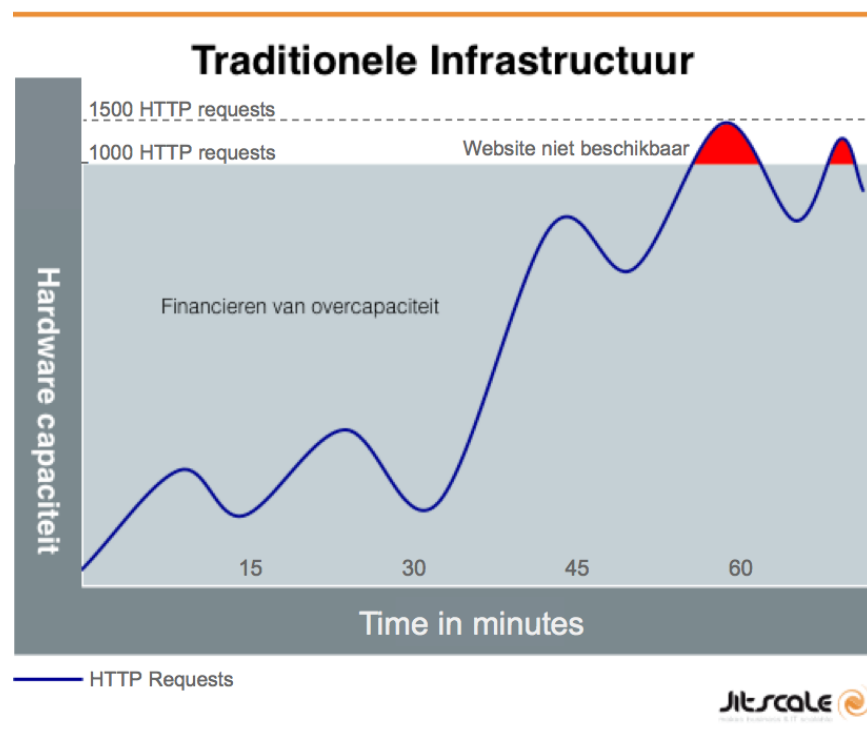
(Apprenda. (z.d.). Definition of Cloud Instance (Single & Multi))

(Daniele Catteddu and Giles Hogben. (2009). Cloud computing security risk assessment)

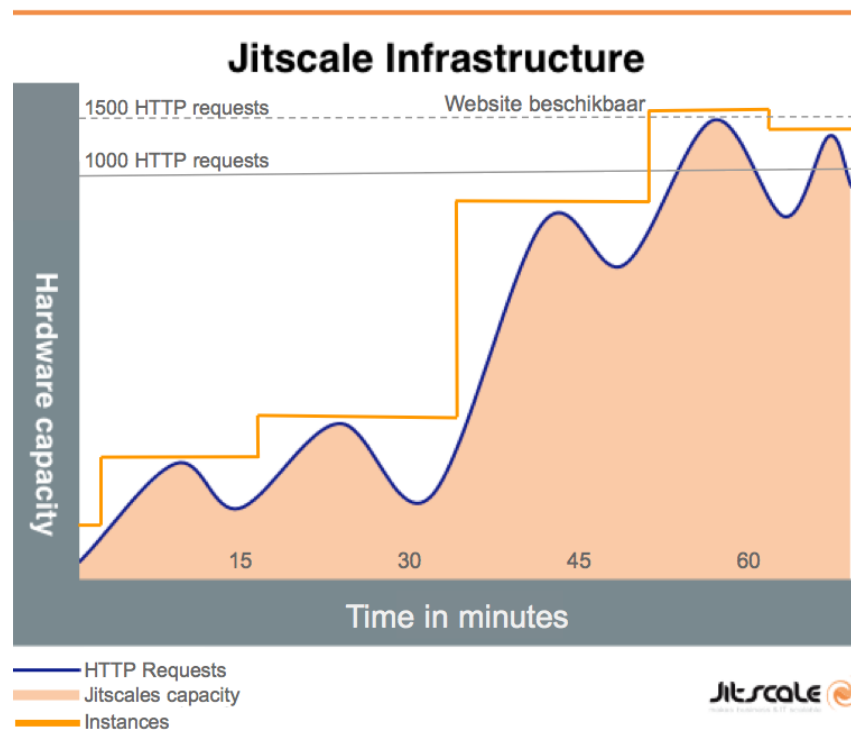
Eén van de voordelen van instances ten op zichten van dedicated servers is dat het met instances wel mogelijk is om onverwachts veel HTTP requests af te handelen als dit er meer zijn dan dat de huidige instances aankunnen. Dit komt omdat het mogelijk is om binnen 30 minuten een werkende instance te starten.

In afbeelding 1.0 is te zien dat het bij dedicated servers (traditionele infrastructuur) niet mogelijk is om de piek van 1500 HTTP requests af te handelen, omdat de hardware capaciteit maar 1000 HTTP requests kan afhandelen. En in afbeelding 1.1 is te zien dat het aantal instances altijd net boven het aantal HTTP requests blijft door het starten en stoppen van instances, waarmee ervoor gezorgd wordt dat de website beschikbaar blijft.

Daarbij is in afbeelding 1.0 ook te zien dat bij het gebruik van dedicated servers er de eerste 45 minuten een overcapaciteit is aan hardware (financieren van overcapaciteit) doordat er meer hardware is dan nodig voor het aantal HTTP requests. Bij het gebruik van instances is dat niet het geval zoals te zien is in afbeelding 1.1, omdat er instances gestart of gestopt worden op het moment dat er meer of minder HTTP requests zijn. Dit is de reden wat cloud computing goedkoper maakt dan het gebruik van dedicated servers, op het moment dat de instance niet nodig is kan deze uitgezet worden en hoeft er ook niet voor betaald te worden, en op het moment dat er meer HTTP requests zijn kan er downtime, en kosten door downtime, voorkomen worden door extra instances te starten. (Reuven Cohen, (2012). Is Cloud Computing Really Cheaper?)



Afbeelding 1.0: Traditionele infrastructuur met dedicated servers.



Afbeelding 1.1: Jitscale infrastructuur met instances.

Doordat de marge tussen het aantal gebruikte en benodigde instances zo laag mogelijk gehouden wordt om de kosten zo laag mogelijk te houden ontstaat er een kans dat er bij een (snelle) stijging van de behoeften (HTTP requests) niet op tijd een instance gestart kan worden, waardoor de instances overbelast raken wat voor downtime zorgt. Het is daarom van belang om in zo vroeg mogelijk stadium een instance te starten, zodat deze gestart is voordat de andere instances overbelast raken.

Het overbelast raken van een instance is te vergelijken met een file op een snelweg die ontstaat doordat er te veel automobilisten gebruik maken van de snelweg, waarbij de snelweg een instance is. Door het grote aantal automobilisten raakt de snelweg overbelast wat ervoor zorgt dat mensen in de file moeten wachten voordat ze door kunnen rijden. Zo'n file kan voorkomen worden door het aanleggen van een extra snelweg, waarmee de automobilisten verdeeld worden over de twee snelwegen. Het aanleggen van een extra snelweg is te vergelijken met het starten van een extra instance, waarmee de bezoekers van de website verdeeld worden over de instances.

Kort samengevat kan er dus downtime voorkomen worden en kosten bespaard worden door het gebruik van cloud computing. De focus van dit project ligt om die reden bij de hosting via cloud computing en niet bij hosting via dedicated servers.

J-Layer

Naast het feit dat het gebruik van instances in de cloud potentie heeft voor het voorkomen van downtime en het verminderen van kosten moet er meer gedaan worden om het ook daadwerkelijk goed bruikbaar te maken voor Jitscale. Dit komt doordat er een aantal taken zijn die moeten gebeuren voordat een instance bruikbaar is. Deze taken zijn het:

- detecteren of er een instance gestart moet worden.
- starten van een instance.
- configureren van een instance.
- koppelen van de instance met de IT infrastructuur.

Deze taken kunnen handmatig uitgevoerd worden maar dit neemt te veel tijd in beslag op het moment dat er een piek is in de benodigde behoeften, waardoor er alsnog downtime mogelijk is. Handmatig zouden de taken ongeveer 30 tot 60 minuten kunnen duren afhankelijk van de detectie en configuratie tijd. Daarom heeft Jitscale J-Layer ontwikkeld. J-Layer is een applicatie waarmee bovenstaande taken geautomatiseerd worden, waardoor het bruikbaar maken van een instance maar 15 minuten duurt. Dit wordt ook wel auto-scaling genoemd. (Cory Jansses, (z.d) Auto Scaling)

Het starten van een instance wordt gedaan door een start request te sturen naar de cloud application programming interfaces (Cloud API) van een cloud provider met het gewenste configuratie zoals het besturingssysteem en type instance. Het type instance bepaald hoeveel resource de instance heeft zoals CPU, schijfruimte en geheugen, dit bepaald overigens ook de kosten. Een instance met meer resources is duurder dan een instance met minder resources. (Wikipedia, (2013), Cloud API)

Als de instance gestart is moet deze geconfigureerd worden zodat deze instance zijn taak kan vervullen binnen de IT infrastructuur. Op een database instance moet bijvoorbeeld een database applicatie geïnstalleerd worden met een de huidige database.

Wat J-Layer ook doet is het detecteren wanneer er een instance gestart moet worden. Dit is te bepalen door te kijken naar hoe druk een instance het heeft. De drukte van een instance wordt gemeten aan de hand van thresholds die ingesteld kunnen worden op de resources van een instance. Als threshold kan bijvoorbeeld ingesteld worden dat de CPU niet meer dan 80% in gebruik mag zijn. Als dat wel het geval is zal er een extra instance gestart worden. Daarnaast kan de threshold ook ingesteld worden op het aantal processen van een applicatie. Als een applicatie maximaal 100 processen kan draaien op de server omdat de server anders te veel CPU gebruikt kan er als threshold

ingesteld worden dat er bij 90 processen een extra instance gestart wordt.
(BusinessDictionary, (z.d), Threshold)

Naast het starten van een instance controleert J-Layer ook wanneer er een instance gestopt kan worden. Dit gebeurt op de zelfde manier als bij het starten maar er wordt in dit geval gekeken of bijvoorbeeld de CPU minder dan 10% in gebruik is of als een applicatie minder dan 10 processen heeft.

Bij het starten en stoppen van instances wordt overigens wel rekening gehouden met het minimaal en maximaal aantal instances dat gestart of gestopt mag worden. Met het minimaal aantal instances wordt ervoor gezorgd dat er in ieder geval genoeg instances zijn om de minimale behoeften op te vangen. Hiermee kan er ook voor gezorgd worden dat er minimaal twee instances draaien voor het geval er één instance stopt met werken. In dat geval kan de andere instance zijn taak direct overnemen.

En met het maximaal aantal instances wordt ervoor gezorgd dat de kosten binnen de perken blijft of binnen de afgesproken maximum kosten die in de SLA met de klant zijn afgesproken.

Een situatie die zich voor kan doen is dat er een distributed denial-of-service (DDOS) aanval gedaan wordt waardoor de instances het erg druk krijgen. In zo'n geval is het niet wenselijk dat er oneindig veel instances op geschaald worden omdat dit voor hoge kosten kan zorgen. In plaats daarvan zal er onderzocht moeten worden waar vandaan de DDOS uitgevoerd wordt zodat de requests die gedaan worden geblokt kunnen worden op basis van IP adressen.

(Margaret Rouse, 2013, Distributed denial-of-service attack.)

(Wikipedia, 2014, IP-adres)

In afbeelding 1.3 is het configuratieformulier in J-Layer te zien waar een threshold van 80% wordt ingesteld op de CPU.

Scaling Up

Plugin	cpu (cpu-1)
Subject	system (cpu-system)
Datasources	value
Operator	> (greater than)
Threshold	80

Afbeelding 1.3

Er zijn twee vormen van op schalen, horizontaal en verticaal. Bij horizontaal op schalen (toevoegen of verminderen van resources) wordt er een extra instance gestart of gestopt en bij verticaal op schalen krijgt een instance meer of minder resources. Het nadeel van verticaal op schalen is dat de instance in sommige gevallen, zoals bij het aanpassen van de CPU, herstart moet worden. Omdat bij horizontaal op schalen de instance niet herstart hoeft te worden maakt J-Layer alleen gebruik van horizontaal op schalen. (Wikipedia. (2014). Scalability)

Probleem

Er is veel downtime te voorkomen door automatisch op te schalen maar in sommige gevallen is deze manier van op schalen niet voldoende. In sommige gevallen stijgt de load van een instance zo hard dat het starten van een extra instance zelfs in 15 minuten alsnog te lang duurt waardoor er toch downtime ontstaat.

Er zal voor die reden een oplossing gevonden moeten worden, binnen J-Layer, waarmee ervoor gezorgd wordt dat er op elk moment precies genoeg instances zijn om te voorzien in de behoeften, en zo min mogelijk instances zodat de kosten zo laag mogelijk blijven.

Doelstelling

Het bedenken en ontwikkelen van een oplossing die geïmplementeerd kan worden binnen J-Layer waarmee ervoor gezorgd wordt dat er op elk moment precies genoeg instances zijn om te voorzien in de behoeften zodat de IT infrastructuur 100% beschikbaar blijft, en zo minmogelijk instances zodat de kosten zo laag mogelijk blijven.

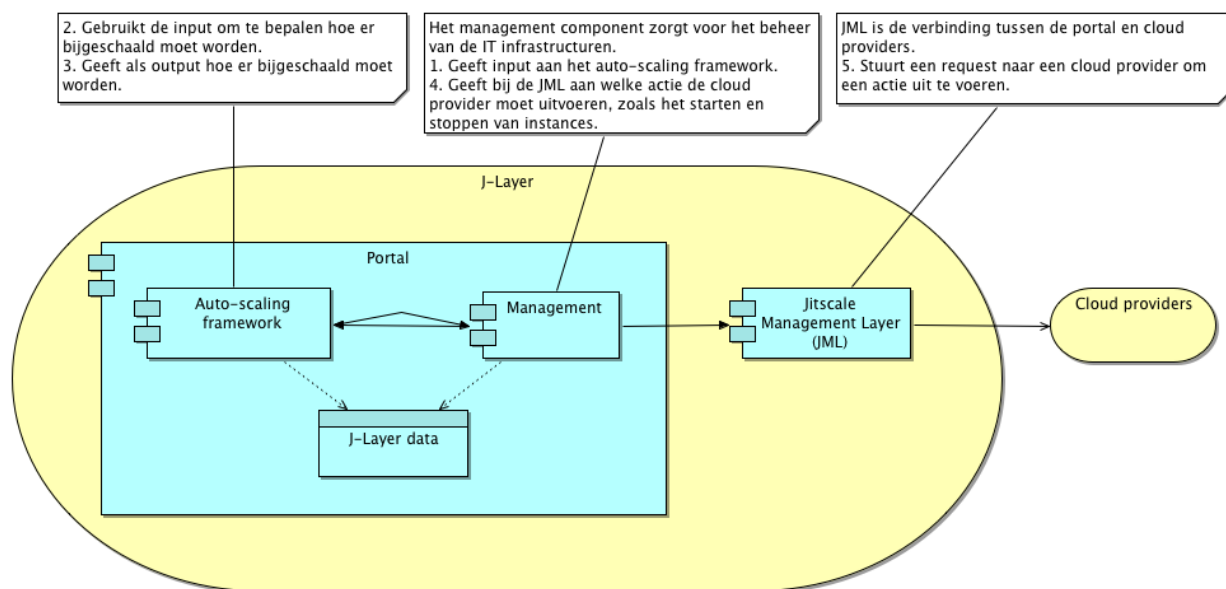
Opdracht

Het ontwikkelen van een pluggable auto-scaling framework binnen J-Layer waarmee voorspeld kan worden, aan de hand van voorspellingsalgoritmes, hoeveel resources er nodig gaan zijn. Door te voorspellen hoeveel resources er nodig zijn kan 100% availability gegarandeerd worden doordat hiermee instances precies op het juiste moment gestart kunnen worden, net voordat ze nodig zijn. (SDART, (z.d.), Forecasting and Prediction Algorithms)

Door gebruik te maken van voorspellingsalgoritmes kan er berekend worden aan de hand van data uit het verleden hoeveel resources er nodig gaan zijn. De data kan bijvoorbeeld zijn het aantal HTTP requests op een bepaald tijdstip van elk dag over de afgelopen twee maanden. Een voorspellingsalgoritme kan aan de hand van de data voorspellen hoeveel HTTP requests er morgen zullen zijn op het zelfde tijdstip, waarmee bepaald kan worden hoeveel resources/instances er nodig zullen zijn. (Wikipedia, (7 april 2014), Plug-in (computing))

Het framework moet pluggable zijn zodat voorspellingalgoritmes onafhankelijk van elkaar als plugins toegevoegd kunnen worden. Dit zorgt voor flexibiliteit doordat bij het toevoegen van algoritmes geen rekening gehouden hoeft te worden met de andere algoritmes.

In afbeelding 1.4 is te zien hoe het auto-scaling framework geïmplementeerd kan worden binnen J-Layer.



Afbeelding 1.4

Om zeker te zijn dat de gegeven opdracht van de opdrachtgever de beste oplossing is zal er eerst onderzocht gaan worden welke mogelijke oplossingen er zijn en wat de beste oplossing is voor het probleem.

Hoofdvraag

Op welke manier kan er binnen J-Layer voor gezorgd worden dat er op elk moment precies genoeg instances gestart zijn en het kosten besparend is?

Deelvragen

1. Wat is J-Layer?
2. Op welke wijze kan ervoor gezorgd worden dat er precies genoeg instances gestart zijn met J-Layer?
 - a. Hoe wordt er nu geprobeerd om precies genoeg instances te starten?
 - b. Hoe komt het dat het nu niet lukt om precies genoeg instances te starten?
3. Op welke wijze kan ervoor gezorgd worden dat er kosten bespaard worden met J-Layer?
 - a. Hoe wordt er nu voor gezorgd dat er kosten bespaard worden?
4. Wat is de beste oplossing om te implementeren binnen J-Layer?
 - a. Wat zijn de functionele en niet-functionele eisen?
 - b. Wat zijn de constraints?
 - c. Hoe kan de oplossing geïmplementeerd worden binnen J-Layer?

Deelvraag	Subvraag	Deel	Manier van Dataverzameling	Activiteiten / resultaten
1		Beschrijvend	Interne bronnen onderzoek, interviews	Onderzoeken en beschrijven wat J-Layer is.
2		Analyserend	Interne en internet bronnen onderzoek.	Veranderingsanalyse. - veranderingsalternatief
	a	Beschrijvend	Interne onderzoek, interviews	Vaststellen wat de huidige manier is om precies genoeg instances te starten. - SWOT analyse
	b	Verklarend	Interne onderzoek, interviews	Vaststellen wat de oorzaak is van het niet precies genoeg starten van instances. - Effect Cause Effect Diagram - veranderingsalternatief
3		Analyserend	Interne en internet bronnen onderzoek.	Veranderingsanalyse. - veranderingsalternatief
	a	Beschrijvend	Interne bronnen onderzoek, interviews	Vaststellen wat de huidige manier van kosten besparing is. - SWOT analyse
4		Beschrijvend	Intern onderzoek.	Implementatie onderzoek. - Functioneel ontwerp - Technisch ontwerp
	a	Beschrijvend	Intern onderzoek, interviews.	Functioneel onderzoek. - Functioneel ontwerp
	b	Beschrijvend	Intern onderzoek, interviews.	Constraints onderzoek. - Functioneel ontwerp.
	c	Analyserend	Intern onderzoek, interviews.	Implementatie onderzoek. - Technisch ontwerp - Implementatie plan

Op te leveren producten

Voor mijn afstudeeropdracht zal ik naar verwachting de volgende producten opleveren:

- scriptie
- functioneel ontwerp
- technisch ontwerp
- implementatie plan
- een prototype van de te ontwikkelen applicatie

Software ontwikkeling

Als software ontwikkelmethoden kies ik voor Scrum. De reden dat ik voor Scrum gekozen heb is omdat Scrum:

- voor een duidelijke planning zorgt.
- flexibel is.
- duidelijkheid geeft over de voortgang.
- een snel werkende producten geeft door het maken van korte sprints.
- zorgt voor snelle communicatie.

Project planning

Fase:	Week	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42
PvA																						
Deelvragen	D1																					
	D2																					
	D3																					
	D4																					
Scriptie																						

Deelopdracht doorlooptijd	
Deadline	
Tussentijdseoplevering	
Uitloop buffer	

Projectrisico's

Ziekte en vakantie:

Door te voorkomen dat het project gevaar loopt bij onverwachte ziekte of vakanties zorg ik voor de volgende maatregelen:

- Een buffer ingepland voor het geval het project uitloopt.
- Ruim van te voren plannen, zodat ik weet wat er moet gaan gebeuren en niet vastloop als er iemand niet is.

Onvoorziene taken met hoge prioriteit toegewezen krijgen die niet gerelateerd zijn aan het project:

- Hiervoor heb ik een buffer ingepland voor het geval het project uitloopt.
- Door gebruik te maken van Scrum kan ik duidelijk zien of ik op schema lig qua uren en hoeveel uren ik aan onvoorziene taken kan besteden.

Projectorganisatie

Teamleider:

Wessel van Heerde

Chief technologie and innovation:

Bart van de Garde

Developer:

Rogier Schipper

Bedrijfs en persoonsgegevens

Bedrijf: Jitscale

Adres: Einsteinbaan 4a, 3439 NJ Nieuwegein

Telefoonnummer: +31 88 00 22 700

Bedrijfsbegeleider: Wessel van Heerde

E-mailadres: wwheerde@jitscale.com

Telefoonnummer: +31 (0)88 00 22 752

Docentbegeleider: Peter van Rooijen

E-mailadres: peter.vanrooijen@hu.nl

Student: Rogier Schipper

Emailadres: rogier.schipper@gmail.com

Telefoonnummer: 06 460 73 103

Te raadplegen literatuur

Amir Atiya, Learning algoritmes for neural networks, (1991),
http://thesis.library.caltech.edu/3725/1/Atiya_a_1991.pdf

Baohua Wang, Hejiao Huang, Xiaolong Wang, (2012), A novel text mining approach to financial time series forecasting, <http://140.125.83.36/fuzzy/upload/reports/%E7%B6%AD%E5%80%AB/2012-05-16-%E7%B6%AD%E5%80%AB-A%20novel%20text%20mining%20approach%20to%20%E7%AC%81nancial%20time%20series%20forecasting.pdf>

Bertan Ari, H. Altay Güvenir, (2002), Clustered linear regression,
<http://yoksis.bilkent.edu.tr/pdf/files/10.1016-S0950-7051%2801%2900154-X.pdf>

Chris Bunch, Vaibhav Arora, Navraj Chohan, Chandra Krintz, Shashank Hegde, Ankit Srivastava, (z.d.), A Pluggable Autoscaling Service for Open Cloud PaaS Systems,
<http://cs.ucsb.edu/~ckrintz/papers/ucc12.pdf>

Daniel Jacobson, Danny Yuan, Neeraj Joshi, (2013), Scryer: Netflix's Predictive Auto Scaling Engine,
<http://techblog.netflix.com/2013/11/scryer-netflixs-predictive-auto-scaling.html>

Daniel Jacobson, Danny Yuan, Neeraj Joshi, (2013), Scryer: Netflix's Predictive Auto Scaling Engine - Part 2, <http://techblog.netflix.com/2013/11/scryer-netflixs-predictive-auto-scaling.html>

Duke University, (z.d.), Introduction to ARIMA: nonseasonal models,
<http://people.duke.edu/~rnau/411arim.htm>

E.M. Steenvoorden, (2013), Scriptie schrijven,
https://onderwijsteams.sharepoint.hu.nl/fnt/Cluster_ICT/afstuderen/Gedeelde%20documenten/Studiewijzer%20Scriptie%20Schrijven.pdf

Gmdh.net, (z.d.), Spectrum of GMDH algoritms, http://www.gmdh.net/GMDH_alg.htm

Ilya Grigorik, (2008), Support Vector Machines (SVM) in Ruby,
<https://www.igvita.com/2008/01/07/support-vector-machines-svm-in-ruby/>

Joseph J. LaViola Jr., (2003), A Testbed for Studying and Choosing Predictive Tracking Algorithms in Virtual, http://www.eecs.ucf.edu/~jjl/pubs/testbed_final_laviola.pdf

NYU Stern, (z.d.)The basics of ARMA models,
<http://pages.stern.nyu.edu/~churvich/TimeSeries/Handouts/Intro.pdf>

Oklahoma state university, (z.d.), The ARIMA Procedure, (z.d.),
<http://www.okstate.edu/sas/v8/saspdf/ets/chap7.pdf>

Scikit-learn, (z.d.), Clustering, <http://scikit-learn.org/stable/modules/clustering.html>

Steehouwer, M., Jansen, C., Mulder, J., Pool, E. van der & Zeijl, W. (2011). *Leren communiceren*. Groningen/Houten: Noordhoff Uitgevers.

Tania Llorido-Bostrán, José Miguel-Alonso, José A. Lozano, (2012), Auto-scaling Techniques for Elastic Applications in Cloud Environments, <http://www.chinacloud.cn/upload/2012-11/12112312255214.pdf>

Theresa Hoang Diem Ngo, Warner Bros. Entertainment Group, Burbank, CA, (2013), The Box-Jenkins Methodology for Time Series Models, <http://support.sas.com/resources/papers/proceedings13/454-2013.pdf>

University of Colorado, (2010), Time series analysis, http://www.colorado.edu/geography/class_homepages/geog_4023_s11/Lecture16_TS3.pdf

Wikipedia, (2014), Cluster analysis, http://en.wikipedia.org/wiki/Cluster_analysis

Wikipedia, (2014), Logistic regression, http://en.wikipedia.org/wiki/Logistic_regression

Wikipedia, (2014), Predictive modelling http://en.wikipedia.org/wiki/Predictive_modelling

Wu, Shih-Hung (Ph.D), (z.d.), Support Vector Machine Tutorial, http://www.csie.cyut.edu.tw/~shwu/PR_slide/SVM.pdf

Bronvermelding

Appenda. (z.d.). Definition of Cloud Instance (Single & Multi).

Geraadpleegd op 7 juni 2014, van

<http://appenda.com/library/glossary/definition-cloud-instance-single-multi/>

BusinessDictionary, (z.d), Threshold

Geraadpleegd op 27 juni 2014, van

<http://www.businessdictionary.com/definition/threshold.html>

Cloud Computing, (11 juni 2014), Cloud Computing

Geraadpleegd op 24 juni 2014, van

http://nl.wikipedia.org/wiki/Cloud_computing

Cory Jansses, (z.d) Auto Scaling.

Geraadpleegd op 27 juni 2014, van

<http://www.techopedia.com/definition/29432/auto-scaling>

Daniele Catteddu and Giles Hogben. (november 2009). Cloud computing security risk assessment.

Geraadpleegd op 7 juni 2014, van

http://www.enisa.europa.eu/activities/risk-management/files/deliverables/cloud-computing-risk-assessment/at_download/fullReport

It law wiki, (z.d), Cloud provider

Geraadpleegd op 24 juni 2014, van

http://itlaw.wikia.com/wiki/Cloud_provider

Margaret Rouse, mei 2013, Distributed denial-of-service attack.

Geraadpleegd op 26 juni 2014, van

<http://searchsecurity.techtarget.com/definition/distributed-denial-of-service-attack>

Rackspace. (z.d.). Definitie van cloudcomputing.

Geraadpleegd op 7 juni 2014, van

http://www.rackspace.nl/cloud-computing/definition?utm_medium=ppc_gen&utm_source=googlenl&utm_campaign=cloud&utm_term=cloud%20computing&gclid=CjgKEAjwzcWcBRCat43fy9e5i3ASJADXOBwuN5PR3-GwTbSQei2iThm30RlwcOUppyY9--lwFSxl0vD_BwE

Reuven Cohen, (8 maart 2012). Is Cloud Computing Really Cheaper?

Geraadpleegd op 7 juni 2014, van

<http://www.forbes.com/sites/reuvencohen/2012/08/03/is-cloud-computing-really-cheaper/>

Rve.org.uk, (z.d), What's in an HTTP request?

Geraadpleegd op 24 juni 2014, van

<http://rve.org.uk/dumprequest>

The web hosting show, (17 april 2007), What are Server Resources?

Geraadpleegd op 24 juni 2014, van

<http://www.webhostingshow.com/2007/04/17/what-are-server-resources/>

Wikipedia, (12 september 2013), Cloud API

Geraadpleegd op 26 juni 2014, van
http://en.wikipedia.org/wiki/Cloud_API

Wikipedia, 13 februari 2014, IT-infrastructuur.

Geraadpleegd op 27 juni 2014, van
<http://nl.wikipedia.org/wiki/IT-infrastructuur>

Wikipedia, (16 april 2014). Downtime.

Geraadpleegd op 7 juni 2014, van
<http://en.wikipedia.org/wiki/Downtime>

Wikipedia. (16 april 2014). Scalability.

Geraadpleegd op 8 juni 2014, van
http://en.wikipedia.org/wiki/Scalability#Horizontal_and_vertical_scaling

Wikipedia, 10 juni 2014, Virtual private server

Geraadpleegd op 25 juni 2014, van
http://en.wikipedia.org/wiki/Virtual_private_server

Wikipedia, (13 juni 2014), Service-level agreement

Geraadpleegd op 26 juni 2014, van
http://en.wikipedia.org/wiki/Service-level_agreement

Wikipedia, (16 juni 2014), Dedicated hosting service

Geraadpleegd op 24 juni 2014, van
http://en.wikipedia.org/wiki/Dedicated_hosting_service

Wikipedia, (23 juni 2014), Central processing unit

Geraadpleegd op 27 juni 2014, van
http://en.wikipedia.org/wiki/Central_processing_unit

Wikipedia, (9 juni 2014), IP-adres

Geraadpleegd op 1 juli 2014, van
<http://nl.wikipedia.org/wiki/IP-adres>

Wikipedia, (7 april 2014), Plug-in (computing)

Geraadpleegd op 29 juni 2014, van
http://en.wikipedia.org/wiki/Plug-in_%28computing%29

SDART, (z.d.), Forecasting and Prediction Algorithms

Geraadpleegd op 30 juni 2014, van
http://www.sdart.co.uk/forecasting_and_prediction_algorithms

Contract afstudeeropdracht
Instituut voor ICT
Nijenoord 1, 3552 AS, UTRECHT

NB: Dit contract dient te worden opgenomen als vast onderdeel van het plan van aanpak

Datum: 03-07-2014

Naam student: Rogier Schipper

Opleiding: Informatica

Variant: voltijd / deeltijd / duaal

Adres student: Achterstraat 18C

Postcode / Woonplaats student: 3441 EH, Woerden

Studentnummer: 1583842

Telefoonnummer privé: 06 46 07 31 03

E-mailadres: rogier.schipper@gmail.com

Naam bedrijf (afstuderen): Jitscale

Adres bedrijf: Einsteinbaan 4a

Postcode / Woonplaats bedrijf: 3439 NJ, Nieuwegein

Naam bedrijfsbegeleider: Wessel van Heerde

Telefoonnummer bedrijfsbegeleider: 06 53 75 18 34

E-mailadres bedrijfsbegeleider: wvheerde@jitscale.com

Beoogde datum van afstuderen: *November 2014*

Geheimhouding geaccordeerd door HU op: -

Ondergetekenden verklaren hiermee akkoord te gaan met de inhoud van bijgesloten PvA.

Handtekeningen

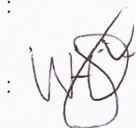
Student

: 

Docentbegeleider

:

Bedrijfsbegeleider^[1]

: 

¹ Door ondertekening van dit formulier verklaart de bedrijfsbegeleider minimaal een hbo- of vergelijkbare opleiding te hebben.

Bijlage 2 – Evaluatie

Ik heb mijn afstudeeropdracht bij Jitscale uitgevoerd waar een oplossing gezocht moest worden voor het voorkomen van downtime en het besparen van kosten bij de hosting van IT infrastructures in de cloud. Deze afstudeeropdracht was een grote uitdaging maar ik heb er veel van geleerd. Waar ik erg blij mee ben is dat ik een project van het begin tot het eind uitgevoerd heb waarbij alle facetten voorbij zijn gekomen zoals onderzoek, ontwikkeling en implementatie.

Voor het onderzoeken van de oplossing moest als eerste bepaald worden wat de beste oplossing zou worden. Hierbij heb ik geleerd om analytisch te werken door het in kaart brengen van feiten en deze te verwerken in een veranderingsanalyse. Door analytisch te werken heb ik ook geleerd beter te plannen en duidelijk in kaart te brengen naar welk resultaat er toegewerkt moet worden. In het begin van het project wist ik wat ik wilde onderzoeken maar ik had toen nog niet een duidelijke planning voor ogen hoe ik dit ging doen, waardoor ik minder gericht naar informatie zocht en het onderzoek mogelijk sneller had kunnen doen. Daarna ben ik vooraf goed bekeken wat voor resultaat ik zocht en wat ervoor nodig was om dat te behalen. Dit heeft uiteindelijk ook voor een duidelijk resultaat gezorgd. Bij dit onderzoek moest onderzocht worden welke voorspellingsmethode gebruikt zou gaan worden. Daarbij heb ik veel geleerd over de werking van verschillende voorspellingsmethodes wat natuurlijk erg handig is, maar mogelijk had ik mij hierbij meer moeten focussen op welke methode er wel of niet gebruikt kan worden op basis van feiten. Daarvoor hoef ik niet precies te weten hoe een voorspellingsmethode werkt, waardoor het onderzoek sneller had kunnen verlopen. Maar de kennis zal zeker bruikbaar zijn in de verdere ontwikkeling van de oplossing (J-Predictor).

De ontwikkeling van J-Predictor zelf is mij goed afgegaan. Voor de ontwikkeling moest de applicatie ontworpen en geprogrammeerd worden. Ik wist na het onderzoek duidelijk wat er gemaakt moest worden en aan welke eisen het moest voldoen, waardoor het ontwerpen goed te doen was. Het programmeren zelf was ook geen probleem doordat ik tijdens mijn werk en op school veel geprogrammeerd heb. Ik verwachtte hierbij ook geen probleem.

Voor de afstudeeropdracht moest ook een plan van aanpak en scriptie geschreven worden. Dit was voor mij een grote uitdaging. Tijdens de studie heb ik een aantal verslagen geschreven maar het blijft voor mij een uitdaging. Bij het schrijven van de plan van aanpak en scriptie heb ik veel geleerd over hoe een verslag opgebouwd wordt en dat het belangrijk is om van te voren duidelijk te hebben wat er beschreven gaat worden. Al met al heb ik dus veel geleerd en ben ik erg blij met het resultaat.

Functioneel ontwerp

J-Layer auto-scaling

Inhoudsopgave

1. Inleiding	1
2. Ontwerp beschrijving	1
3. Doelgroep & Doelen.....	2
4. Functionele eisen.....	2
5. Niet-functionele eisen	4
6. Use cases.....	5
6.1. Actoren	5
6.2. Use case diagram	6
6.3. Uitgebreide use cases	7
UC1 - Nieuwe predictive auto scaler aanmaken.....	7
UC2 - Predictive auto scaler en predictor configureren.	7
UC3 - Nieuwe responsive auto scaler aanmaken.....	8
UC4 - Time serie data pushen naar J-Predictor.	9
UC5 - Bepalen van scaling plan.	9
UC6 - Nieuwe Voorspeller aanmaken via GUI	10
UC7 - Nieuwe Voorspeller aanmaken via API	10
UC8 - Time serie updaten via een GUI	11
UC9 - Time serie updaten via een API	12
UC10 - Voorspeller configuratie via een GUI.....	12
UC11 - Voorspeller configuratie via een API	13
UC12 - Bepalen van seasonality via GUI	14
UC13 - Bepalen van seasonality via API	14
UC14 - Automatische bepalen van configuratie via GUI.....	15
UC15 - Automatische bepalen van configuratie via API	15
UC16 - Voorspelling via GUI	16
UC17 - Voorspelling via API.....	16
UC18 - Automatisch voorspelling	17
UC19 - Voorspelling accuraatheid check.....	18
UC20 - Automatisch configuratie tuning	18
7. Business rules	18
8. Functionele beschrijving.....	20

8.1.	Taken van J-Predictor	20
8.1.1.	Ontvangen en opslaan van de time series	20
8.1.2.	analyseren van time series	20
8.1.3.	Het doen van een voorspelling	21
8.1.4.	Opslaan en beschikbaar stellen van voorspellingen	21
8.1.5.	Analyseren van voorspellingen op juistheid	21
8.2.	Taken van J-Layer	21
8.2.1.	Aanmaken van een predictor	21
8.2.2.	Versturen van monitoring time series naar de J-Predictor	22
8.2.3.	opvragen van de voorspelling uit J-Predictor	22
8.2.4.	Scaling plan bepalen aan de hand van de ingestelde instance threshold	22
8.2.5.	Starten van de instances volgens het scaling plan	22
8.3.	Functioneel proces	22
9.	Voorspellingsmethode	23
10.	Voorspellingsmodel bepalen	25
11.	Outlier detectie	25
12.	Schermontwerpen	26
12.1.	New predictor	26
12.2.	Configure predictor	26
12.3.	Prediction method settings	27
12.4.	Time Serie Analyzer	28
12.5.	Auto tune settings	29
12.6.	Tune report	30
12.7.	Add time serie	31
12.8.	Time serie	32
12.9.	Dashboard	33
12.10.	Predictions	34
12.11.	New auto-scaler	35
12.12.	Predictive scaler configuration	35
12.13.	Responsive scaler configuration	36
13.	Bibliografie	37

1. Inleiding

Dit document bevat de functionele beschrijving over hoe J-Layer in staat zal gaan zijn om het aantal benodigde instances te gaan voorspellen. Als eerste zal een beschrijving gegeven worden van het ontwerp en met welk doel en voor welke doelgroepen het ontwikkeld wordt. Vervolgens worden de functionele eisen, use cases en business rules beschreven waarmee een beeld wordt gecreëerd van de werking van J-Predictor in samenwerking met J-Layer. Als laatste zijn de mockups van de schermen te zien, zoals ze ontwikkeld gaan worden in J-Predictor en J-Layer.

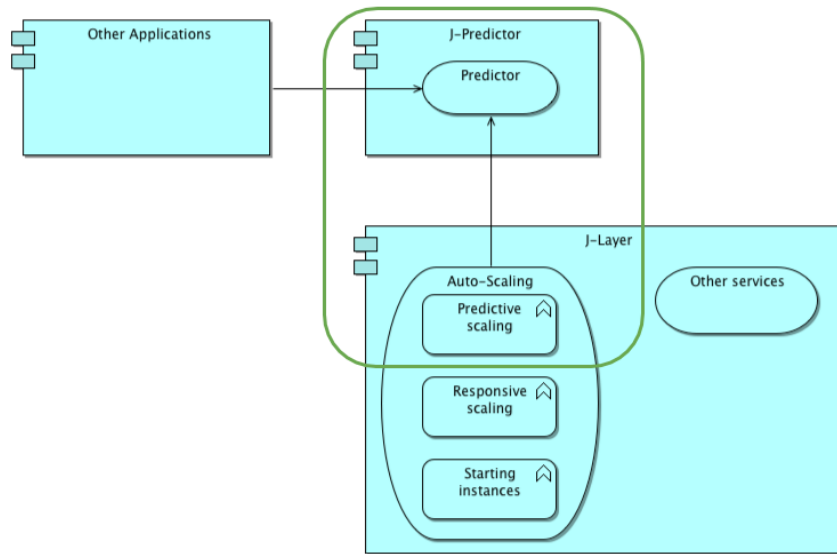
2. Ontwerp beschrijving

Voor Jitscale moet een applicatie ontwikkeld worden waarmee J-Layer in staat is te voorspellen hoeveel instances nodig gaan zijn. In eerste instantie zoals in het plan van aanpak vermeld staat was er vanuit gegaan dat dit een component zou worden binnen J-Layer. Tijdens de ontwerpfase bleek dat de ontwikkeling van een voorspellingsapplicatie, die als service te gebruiken is door J-Layer meer flexibiliteit en mogelijkheden levert dan het implementeren van een voorspellingscomponent binnen J-Layer zelf.

Twee redenen dragen bij aan deze beslissing.

1. Met een voorspellingsapplicatie als service is er meer flexibiliteit doordat het mogelijk is om de service los van J-Layer te schalen wat er voor zorgt dat er voor de voorspellingsservice extra computerkracht ingezet kan worden voor het doen van voorspellingen. Dit zorgt ook voor minder coupling met J-Layer doordat in het design van J-Layer geen rekening gehouden hoeft te worden met het schalen van de voorspellingsapplicatie.
2. De voorspellingsapplicatie is te gebruiken voor andere doeleinden dan J-Layer's auto-scaling services. Zo kan de voorspellingsservice ook gebruikt worden voor het voorspellen van de drukte op de support desk aan de hand van het aantal tickets dat aangemaakt wordt en het aantal monitoring meldingen.

In afbeelding 2.1 is de scope (groene omlijning) van dit project te zien en hoe J-Layer en andere applicaties gebruik kunnen maken van J-Predictor. In dit project zal J-Predictor en het predictive scaling componenten ontwikkeld worden. Het responsive scaling component is het component dat instances schaaft op basis van de huidige behoeften.



Afbeelding 2.1 - Project scope

3. Doelgroep & Doelen

Het doel is het ontwerpen van een applicatie waarmee door J-Layer voorspeld kan worden hoeveel instances nodig zullen zijn. Omdat J-Layer weet hoeveel instances nodig zullen zijn kunnen downtime en performance problemen voorkomen worden en kosten verminderd worden, doordat J-Layer hiermee in staat is om instances op het juiste moment te starten of te stoppen.

De doelgroep waarvoor de applicatie ontwikkeld wordt is Jitscale. Jitscale zal de applicatie intern gaan gebruiken voor het doen van voorspellingen voor J-Layer. Daarbij wordt er bij de ontwikkeling van J-Predictor ook rekening mee gehouden dat de applicatie gebruikt kan worden voor andere doeleinden, door J-Predictor niet te ontwikkelen binnen J-Layer maar als een losse applicatie waar J-Layer gebruik van kan maken.

4. Functionele eisen

Doordat J-Predictor als losse applicatie gaat werken en J-Layer hier gebruik van moet maken voor het voorspellen van het aantal instances, zijn er functionele eisen voor zowel J-Layer als J-Predictor. De communicatie tussen de twee applicaties zal verlopen via een API en daarbij zullen J-Layer en J-Predictor GUI's hebben waarmee de applicaties geconfigureerd kunnen worden.

Code	Functionele eisen:	Applicatie	GUI/API
FE2	Er kan aangegeven worden hoeveel instances minimaal en maximaal gestart moeten/mogen zijn.	J-Layer	GUI
FE3	Het bepalen van het aantal instances wordt gedaan door een voorspelling op basis van time serie data.	J-Layer J-Predictor	-
FE4	Met een threshold kan worden aangegeven wanneer een instance bij geschaald moet worden op basis van de uitkomst van de voorspelling.	J-Layer	GUI
FE5	Er wordt gebruik gemaakt van monitoring time serie data die al binnen J-Layer beschikbaar en op te vragen zijn.	J-Layer	-
FE6	Er is een lijst van time series waaruit geselecteerd kan worden.	J-Layer	GUI

FE7	De geselecteerde time serie kan gekoppeld worden met de J-Predictor.	J-Layer	GUI
FE8	De time serie data worden automatisch verstuurd naar J-Predictor.	J-Layer	-
FE9	Er wordt gekeken naar de huidige status van een instance om te bepalen of er bij geschaald moet worden. (Responsive scaling).	J-Layer	-
FE10	Het ophalen van een voorspelling uit J-predictor gaat op basis van de aangegeven periode.	J-Layer	-
FE11	Er kan een predictor worden aangemaakt waarbij de voorspellingsmethode geconfigureerd kan worden door de gebruiker.	J-Predictor	GUI
FE12	Het is mogelijk om te kiezen tussen verschillende voorspellingsmethodes.	J-Predictor	GUI
FE13	Er kan aangegeven worden met welke tijdsinterval een voorspelling gedaan moet worden.	J-Predictor	GUI/API
FE14	Er kan aangegeven worden welke periode voorspeld moet worden.	J-Predictor	GUI/API
FE15	De instellingen van de geselecteerde voorspellingsmethode kunnen ingesteld worden. Per voorspellingsmethode kunnen de instellingen verschillen.	J-Predictor	GUI/API
FE16	Per predictor kunnen time series toegevoegd worden die worden opgeslagen in de database.	J-Predictor	GUI/API
FE17	De time series kunnen incrementeel gestuurd worden, zodat historische data die al aanwezig zijn niet nogmaals verstuurd hoeven te worden.	J-Predictor	GUI/API
FE18	De toegevoegde time serie wordt weergegeven in een tabel en grafiek.	J-Predictor	GUI
FE19	De time serie kan geanalyseerd worden, zodat de instellingen voor de voorspellingsmethode zoals de seasonality bepaald kunnen worden.	J-Predictor	GUI
FE20	De beste settings voor de voorspellingsmethode kunnen automatisch bepaald worden.	J-Predictor	GUI
FE21	Het auto tunen van settings resulteert in een overzicht van settings en de combinatie en de accuraatheid van de voorspelling met die settings.	J-Predictor	GUI
FE22	Het is mogelijk om gevonden settings via auto tuning met één druk op de knop in te stellen	J-Predictor	GUI
FE23	De accuraatheid van de voorspelling kan getest worden.	J-Predictor	GUI
FE24	Er kan een eenmalige voorspelling gedaan worden onafhankelijk van de ingestelde tijdsinterval.	J-Predictor	GUI
FE25	De voorspellingen zijn te zien in een grafiek.	J-Predictor	GUI
FE26	De voorspellingen zijn op te vragen via de API.	J-Predictor	API
FE27	De voorspellingsmethode kan meerdere seasonalities herkennen.	J-Predictor	-
FE28	De voorspellingsmethode kan trends herkennen.	J-Predictor	-
FE29	De voorspellingsmethode is in staat om noise en outliers te detecteren en negeren.	J-Predictor	-
FE30	De voorspellingsmethode kan under- en overfitting voorkomen.	J-Predictor	-
FE31	J-Predictor bevat statistische analyse tools die gebruikt kunnen worden door elke voorspellingsmethode.	J-Predictor	-
FE32	De voorspellingsmethode is in staat om de juistheid van de voorspelling te valideren.	J-Predictor	-

FE33	J-Predictor controleert per predictor eenmaal per dag hoe accuraat de voorspelling is.	J-Predictor	-
------	--	-------------	---

5. Niet-functionele eisen

Net zoals bij de functionele eisen overkoepelen de niet-functionele eisen J-Layer en J-Predictor. De niet-functionele eisen zijn de verplichtingen (constraints) waaraan de applicatie moet voldoen. Bij het technische ontwerp wordt met deze verplichtingen rekening gehouden voor het ontwerpen van de applicatie.

Code	Niet-functionele eisen:
NFE1	J-Predictor moet schaalbaar zijn, zodat er extra server capaciteit ingezet kan worden.
NFE2	J-Predictor moet ingezet kunnen worden voor andere voorspellingsdoeleinden dan het voorspellen van het aantal instances.
NFE3	Alleen medewerkers van Jitscale hebben toegang tot J-Predictor.
NFE4	Er worden dagelijks incrementeel backups gemaakt met één maal in de week een volledige back-up van J-Predictor.
NFE5	J-Predictor moet gebruik kunnen maken van de applicatie R.
NFE6	Het moet mogelijk zijn om de time serie incrementeel naar J-Predictor te sturen. Dit voorkomt overbodig netwerk verkeer.
NFE8	J-Predictor moet 100% van de tijd beschikbaar zijn.
NFE10	J-Predictor moet 24/7 gemonitord worden om te controleren of het systeem nog functioneert. Bij problemen moet een Jitscale medewerker hier een bericht over krijgen.
NFE11	De voorspellingen met J-Predictor mogen niet langer duren dan 15 minuten.
NFE12	Een API request mag niet langer dan 3 seconden duren.
NFE13	Het laden van de J-Predictor GUI mag niet langer dan 3 seconden duren.
NFE14	J-Predictor moet kunnen functioneren op systemen met Debian Wheezy.
NFE16	J-Predictor mag alleen beschikbaar zijn binnen de infrastructuur van Jitscale.
NFE18	Alle functies van J-Predictor moeten via de GUI maar ook via de API te gebruiken zijn.
NFE19	De werking van J-Predictor moet gedocumenteerd zijn in de knowledge base van Jitscale.
NFE20	Het moet mogelijk zijn om voorspellingsmethodes pluggable toe te voegen.
NFE21	Berichten tussen J-Predictor en J-Layer moeten leesbaar zijn voor mensen.
NFE22	Time series worden 10 minuten voordat de voorspelling uitgevoerd wordt verstuurd door J-Layer naar J-Predictor.
NFE23	Als de nieuw verstuurde time serie de zelfde X waarde bevat als de time serie die al opgeslagen is in de database van J-Predictor zal de nieuwe Y waarde op de plaats van X de oude Y waarden in de database overschrijven. Hierbij wordt er vanuit gegaan dat nieuwe time serie data belangrijker zijn dan al bestaande time serie data.
NFE24	De voorspellingsmethode moet een accurate voorspelling kunnen doen. Met accuraat wordt bedoeld dat bij een time serie met herkenbare seasonaliteit en trend met 90% zekerheid het aantal instances bepaald kan worden.

6. Use cases

In dit hoofdstuk worden de use cases en de actoren beschreven die van de J-Predictor gebruik zullen gaan maken.

6.1. Actoren

Actor	Beschrijving
Client	Een client is een applicatie die gebruik maakt van J-Predictor.
J-Layer	J-Layer is de applicatie die aan de hand van de voorspelling van J-Predictor bepaalt hoeveel instances er bijgeschaald moeten worden. J-Layer is dus een client.
Medewerker	De medewerker is een Jitscale medewerker die J-Predictor zal gaan configureren voor het doen van voorspellingen.

6.2. Use case diagram

Afbeelding 6.1 toont het use case diagram waarin een overzicht gegeven worden van de use cases. De uitgebreide beschrijving van de use cases is beschreven in paragraaf 6.3.



Afbeelding 6.1 - Use case diagram

6.3. Uitgebreide use cases

UC1 - Nieuwe predictive auto scaler aanmaken.

Beschrijving	De gebruiker maakt in J-Layer een nieuwe predictive auto-scaler aan.
Actors	<ul style="list-style-type: none">- Gebruiker- J-Layer- J-Predictor
Voorwaarden	<ul style="list-style-type: none">- Alle velden moeten ingevuld zijn. <p>Parameters:</p> <ul style="list-style-type: none">- minimum_instances: integer- maximum_instances: integer
Resultaat	De predictive auto scaler in J-Layer en de predictor in J-Predictor zijn aangemaakt en aan elkaar gekoppeld.
Start	De gebruiker wil een nieuwe predictive auto scaler aanmaken.
Proces	
Nr	Beschrijving
1	De gebruiker opent de GUI in J-Layer voor het aanmaken van de predictive auto scaler.
2	J-Layer haalt de beschikbare voorspellingsmethodes op uit J-Predictor.
3	De gebruiker vult alle velden in.
4	J-Layer controleert of alle velden zijn ingevuld. Zo niet, dan moet de gebruiker terug naar stap 2.
5	J-Layer stuurt een API request naar J-Predictor voor het aanmaken van de predictor.
6	J-Layer ontvangt van J-Predictor het ID van de predictor.
7	J-Layer maakt de predictive auto scaler aan.

UC2 - Predictive auto scaler en predictor configureren.

Beschrijving	De gebruiker configureert in J-Layer de predictive auto-scaler en predictor, zodat er voorspellingen gedaan kunnen worden.
Actors	<ul style="list-style-type: none">- Gebruiker- J-Layer- J-Predictor
Voorwaarden	<ul style="list-style-type: none">- Alle velden moeten ingevuld zijn.- Er moeten time series beschikbaar zijn uit de monitoring. <p>Parameters:</p> <ul style="list-style-type: none">- Instance scaling threshold: integer- Prediction period: integer- Prediction interval: integer

	- Voorspellingsmethode settings.
Resultaat	De predictive auto-scaler en predictor zijn geconfigureerd, zodat time series naar de predictor verstuurd kunnen worden en er voorspellingen gedaan kunnen worden.
Start	De gebruiker opent het scherm om de predictive auto-scaler en predictor te configureren.
Proces	
Nr	Beschrijving
1	De gebruiker vult alle velden in en selecteert welke time serie(s) de predictor moet gebruiken.
2	J-Layer stuurt een API request naar J-Predictor voor het aanmaken van de configuratie van de predictor.
3	J-Layer en J-Predictor controleren of alle velden juist zijn ingevuld. Zo niet, dan moet de gebruiker terug naar stap 2.
4	J-Layer en J-Predictor configureren de instellingen.

UC3 - Nieuwe responsive auto scaler aanmaken.

Beschrijving	De gebruiker maakt in J-Layer een nieuwe responsive auto-scaler aan.
Actors	- Gebruiker - J-Layer
Voorwaarden	- Alle velden moeten ingevuld zijn. Parameters: - minimum_instances: integer - maximum_instances: integer - instance_scaling_threshold: integer - wait_time : integer. - breach_time: integer. - sample_time: integer.
Resultaat	De responsive auto scaler is aangemaakt en er wordt bijgeschaald volgens de ingestelde waarden.
Start	De gebruiker opent het scherm om de responsive auto-scaler aan te maken.
Proces	
Nr	Beschrijving
1	De gebruiker vult alle velden in.
2	J-Layer controleert of de velden juist zijn ingevuld. Zo niet, dan moet de gebruiker terug naar stap 1.
3	J-Layer controleert of de velden juist zijn ingevuld. Zo niet, dan moet de gebruiker terug naar stap 1.
4	J-Layer maakt de responsive auto scaler aan.

UC4 - Time serie data pushen naar J-Predictor.

Beschrijving	De time serie die wordt verstuurd naar J-Predictor, zodat J-Predictor de time serie data kan gebruiken voor het doen van een voorspelling.
Actors	- J-Layer - J-Predictor
Voorwaarden	Parameters: - J-Predictor predictor ID: integer - J-Predictor time serie ID: integer - time_serie: <integer, double> Note: De X en Y waarden van de time serie moeten een integer zijn. Als X een datum is, kan deze geconverteerd worden naar een timestamp.
Resultaat	J-Predictor doet voorspellingen op basis van de verstuurde time serie.
Start	De automated scheduler van J-Layer triggered de actie om time series te versturen naar J-Predictor.
Proces	
Nr	Beschrijving
1	J-Layer verstuurt de time serie(s) van de predictive auto scaler naar J-Predictor.
2	J-Predictor controleert of het format van de time serie(s) juist is. Zo niet, dan wordt er een foutmelding gegeven en moet de time serie opnieuw verstuurd worden.
3	J-Predictor koppelt time serie aan de predictor met het opgegeven predictor ID en time serie ID.

UC5 - Bepalen van scaling plan.

Beschrijving	Aan de hand van de voorspellingen uit J-Predictor word bepaald hoeveel instances nodig zijn.
Actors	- J-Layer - J-Predictor
Voorwaarden	Er moet een voorspelling beschikbaar zijn in de opgegeven periode. Parameter: predictor_id: integer begin_period: timestamp. end_period: timestamp.
Resultaat	Een scaling plan voor de opgegeven periode.
Start	Een gebruiker of de automated scheduler start het proces om het scaling plan te updaten.
Proces	
Nr	Beschrijving

1	J-Layer start het proces voor het update van het scaling plan wordt gestart 1 minuut nadat de voorspelling gedaan is door J-Predictor.
2	J-Layer stuurt elke 30 seconden een verzoek naar J-Predictor voor het ophalen van de voorspellingen in de aangegeven periode, totdat J-Layer de voorspelling ontvangen heeft.
3	J-Predictor stuurt de voorspellingen van de opgegeven periode.
4	J-Layer ontvangt de voorspellingen.
5	J-Layer vertaalt de voorspellingen naar het aantal benodigde instances op het tijdstip van de voorspelling aan de hand van de <code>instance_scaling_threshold</code> , <code>minimum_instances</code> en <code>maximum_instances</code> , wat resulteert in het scaling plan.

UC6 - Nieuwe Voorspeller aanmaken via GUI

Beschrijving	Het aanmaken van een nieuwe predictor via de GUI.
Actors	<ul style="list-style-type: none"> - Gebruiker - J-Predictor
Voorwaarden	<ul style="list-style-type: none"> - De GUI moet alleen beschikbaar zijn binnen het netwerk van Jitscale. - Alle velden moeten ingevuld zijn. - Voorspellingsmethode selectie zoals Clustered linear regression. Parameters: <ul style="list-style-type: none"> - Naam: Maximaal 255 karakters lang.
Resultaat	- Een nieuwe voorspeller die nog geconfigureerd moet worden voordat er mee voorspeld kan worden.
Start	De gebruiker opent de GUI waarmee de voorspelling geconfigureerd kan worden.
Proces	
Nr	Beschrijving
1	De gebruiker opent de GUI via een URL in de webbrowser.
2	De gebruiker vult alle velden in met de gewenste parameters.
3	De gebruiker geeft aan dat de voorspeller aangemaakt kan worden met de ingestelde parameters.
4	J-Predictor controleert of de ingevulde parameters voldoen aan de eisen. Zo niet dan moet de gebruiker terug naar stap 2.
5	J-Predictor slaat de voorspeller op met de ingestelde parameter.

UC7 - Nieuwe Voorspeller aanmaken via API

Beschrijving	Het aanmaken van een nieuwe Voorspeller via de API.
Actors	<ul style="list-style-type: none"> - Gebruiker - J-Predictor

Voorwaarden	<ul style="list-style-type: none"> - De GUI moet alleen beschikbaar zijn binnen het netwerk van Jitscale. - Alle velden moeten ingevuld zijn. - Voorspellingsmethode selectie zoals Clustered linear regression. - J-Predictor geeft een unieke referentie ID aan de gebruiker. Parameters: - Naam: Maximaal 255 karakters lang.
Resultaat	<ul style="list-style-type: none"> - Een referentie ID. - Een nieuwe voorspeller die nog geconfigureerd moet worden voordat ermee voorspeld kan worden.
Start	De gebruiker opent de API waarmee de voorspelling geconfigureerd kan worden.
Proces	
Nr	Beschrijving
1	De gebruiker opent de API via een URL in de webbrowser.
2	De gebruiker vult alle velden in met de gewenste parameters.
3	De gebruiker geeft aan dat de voorspeller aangemaakt kan worden met de ingestelde parameters.
4	J-Predictor controleert of de ingevulde parameters voldoen aan de eisen. Zo niet dan moet de gebruiker terug naar stap 2.
5	J-Predictor slaat de voorspeller op met de ingestelde parameter.
6	J-Predictor geeft een referentie ID aan de gebruiker.

UC8 - Time serie updaten via een GUI

Beschrijving	Het updaten van de time serie met nieuwe data via de GUI.
Actors	<ul style="list-style-type: none"> - Gebruiker - J-Predictor
Voorwaarden	- De GUI moet alleen beschikbaar zijn binnen het netwerk van Jitscale.
Resultaat	<ul style="list-style-type: none"> - De time serie is geüpdatet met nieuwe data. - De naam van de time serie is aangepast.
Start	De gebruiker opent de GUI waarmee de time serie data geüpload kan worden.
Proces	
Nr	Beschrijving
1	De gebruiker opent de GUI via een URL in de webbrowser.
2	De gebruiker vult de naam in van de time serie. (optioneel)
3	De gebruiker selecteert de time serie die geüpload moet worden.
4	J-Predictor controleert of de ingevulde parameters voldoen aan de eisen. Zo niet dan moet de gebruiker terug naar stap 2.

5	J-Predictor slaat de naam en time serie op in de database.
---	--

UC9 - Time serie updaten via een API

Beschrijving	Het updaten van de time serie met nieuwe data via de API.
Actors	- Gebruiker - J-Predictor
Voorwaarden	- De API moet alleen beschikbaar zijn binnen het netwerk van Jitscale. - De request bevat het referentie ID. - De request bevat het time serie referentie ID. - De request bevat de time serie data.
Resultaat	- De time serie met het time serie referentie ID is geüpdatet. - De naam van de time serie met het referentie ID is aangepast.
Start	De client doet een API request waarmee de time serie geüpdatet kan worden.
Proces	
Nr	Beschrijving
1	De client doet een API request met de parameters waarmee de time serie geüpdatet kan worden: time serie referentie ID, naam (optioneel), time serie data.
2	J-Predictor controleert of de ingevulde parameters voldoen aan de eisen. Zo niet dan moet de gebruiker terug naar stap 1.
3	J-Predictor slaat de naam en time serie op in de database.

UC10 - Voorspeller configuratie via een GUI

Beschrijving	Het configureren van de voorspeller met de juiste parameters via een GUI.
Actors	- Gebruiker - J-Predictor
Voorwaarden	- De GUI moet alleen beschikbaar zijn binnen het netwerk van Jitscale. - Alle verplichte settings van de voorspeller en voorspellingsmethode moeten ingevuld zijn. Parameters: - Interval: Moet van het type float of integer zijn. Voor minuten ≤ 60 en voor uren ≤ 24 . - Voorspellingsperiode: integer. Settings voor Clustered linear regression: - Relevantie van historische data: Moet float zijn welke ≥ 0 en ≤ 100 . - Historische relevantie afname: Moet float zijn welke ≥ 0 en ≤ 100 . - Time serie formaat: Moet een float of integer zijn. - Seasonality: Moet een integer of float zijn.
Resultaat	- Een geconfigureerde Predictor die door J-Predictor automatisch wordt uitgevoerd op de bepaalde tijdstippen.
Start	De gebruiker opent de GUI waarmee de voorspelling geconfigureerd kan worden.

Proces	
Nr	Beschrijving
1	De gebruiker opent de GUI via een URL in de webbrowser.
2	De gebruiker vult alle velden in met de gewenste parameters.
3	De gebruiker geeft aan dat de voorspelling aangemaakt kan worden met de ingestelde parameters.
4	J-Predictor controleert of de ingevulde parameters voldoen aan de eisen. Zo niet dan moet de gebruiker terug naar stap 2.
5	J-Predictor slaat de voorspelling op met de ingestelde parameter.

UC11 - Voorspeller configuratie via een API

Beschrijving	Het configureren van de Voorspeller met de juiste parameters via een API.
Actors	<ul style="list-style-type: none"> - Client - J-Predictor
Voorwaarden	<ul style="list-style-type: none"> - De API moet alleen beschikbaar zijn binnen het netwerk van Jitscale. - De request bevat het referentie ID. - Alle verplichte settings van de voorspeller en voorspellingsmethode moeten ingevuld zijn. - De API request moet alle benodigde parameters bevatten: <ul style="list-style-type: none"> - Naam - Voorspellingsinterval - Relevantie van historische data - Historische relevantie afname - Time serie format - Seasonality <p>Parameters:</p> <ul style="list-style-type: none"> - Interval: Moet van het type float of integer zijn. Voor minuten ≤ 60 en voor uren ≤ 24. - Voorspellingsperiode: integer. <p>Settings voor Clustered linear regression:</p> <ul style="list-style-type: none"> - Relevantie van historische data: Moet float zijn welke ≥ 0 en ≤ 100. - Historische relevantie afname: Moet float zijn welke ≥ 0 en ≤ 100. - Time serie format: Moet een float of integer zijn. - Seasonality: Moet een integer of float zijn.
Resultaat	<ul style="list-style-type: none"> - Een time serie referentie ID waarmee de time serie geïdentificeerd kan worden. Dit is nodig als er een voorspelling gedaan wordt met meerdere time series. - Een geconfigureerde voorspelling die door J-Predictor automatisch wordt uitgevoerd op de bepaalde tijdstippen.
Start	De client doet een API request met de parameters die ingesteld moeten worden.
Proces	
Nr	Beschrijving

1	De client doet een API request met de parameters voor het instellen van de voorspelling
2	J-Predictor controleert of de parameters voldoen aan de eisen en of het referentie ID bestaat. Zo niet dan moet de client een nieuwe API request doen.
3	J-Predictor slaat de voorspelling op met de verkregen parameter.

UC12 - Bepalen van seasonality via GUI

Beschrijving	Via een GUI bepalen wat de seasonality van de time serie is voor de configuratie van de voorspelling.
Actors	- Gebruiker - J-Predictor
Voorwaarden	- Time serie upload veld: De time serie data moet de volgende structuur hebben per regel: timestamp,value. - Voor het genereren van de seasonality grafieken moet er time serie data geupload zijn.
Resultaat	Grafieken waaruit de seasonality bepaald kan worden.
Start	De gebruiker upload de time serie data.
Proces	
Nr	Beschrijving
1	De gebruiker upload de time serie data
2	J-Predictor controleert of de data de juiste structuur heeft. Zo niet dan moet de gebruiker terug naar stap 1.
3	J-Predictor genereert (partial) auto correlation grafieken en differentieert de data als de time serie niet stationair is.
4	De grafieken worden in een GUI weergegeven.

UC13 - Bepalen van seasonality via API

Beschrijving	Via een API bepalen wat de seasonality van de time serie is voor de configuratie van de voorspelling.
Actors	- Client - J-Predictor
Voorwaarden	- De time serie data moet de volgende structuur hebben per regel, waarbij de X en Y waarden het type float zijn: <X waarde >, <Y waarde >. - Voor het genereren van de seasonality grafieken moet de time serie data meegestuurd worden met de API request. - De request bevat het referentie ID.
Resultaat	Grafieken waaruit de seasonality bepaald kan worden.
Start	De client doet een API request met de time serie data.
Proces	

Nr	Beschrijving
1	De client doet een API request met de time serie data.
2	J-Predictor controleert of de data de juiste structuur heeft en of het referentie ID bestaat. Zo niet dan moet de client terug naar stap 1.
3	J-Predictor genereert (partial) auto correlation grafieken en differentieert de data als de time serie niet stationair is.
4	J-Predictor stuurt de grafieken naar de client.

UC14 - Automatische bepalen van configuratie via GUI

Beschrijving	J-Predictor bepaalt zelf welke parameters de meest accurate voorspelling geven. (K-fold cross validation).
Actors	- Gebruiker - J-Predictor
Voorwaarden	- De time serie data moet de volgende structuur hebben per regel, waarbij de X en Y waarden het type float zijn: <X waarde >, <Y waarde >. - Voor het automatische bepalen van de configuratie moet de time serie data geüpload zijn.
Resultaat	Configuratie parameters waarmee een accurate voorspelling gedaan kan worden.
Start	De gebruiker opent het formulier waarmee de parameters bepaald kunnen worden.
Proces	
Nr	Beschrijving
1	De gebruiker upload de time serie data.
2	De gebruiker drukt op de knop waarmee het automatisch bepalen van de parameters gestart wordt.
3	J-Predictor controleert of de data de juiste structuur heeft. Zo niet dan moet de client terug naar stap 1.
4	J-Predictor controleert met K-fold cross validation welke parameters de beste voorspelling geven.
5	Het auto-scaling geeft de uitkomst weer aan de gebruiker.
6	De gebruiker beslist of de parameters ingesteld moeten worden.

UC15 - Automatische bepalen van configuratie via API

Beschrijving	J-Predictor bepaalt zelf welke parameters de meest accurate voorspelling geven (K-fold cross validation).
Actors	- Client - J-Predictor
Voorwaarden	- De time serie data moet de volgende structuur hebben per regel, waarbij de X en Y waarden het type float zijn: <X waarde >, <Y waarde >.

	<ul style="list-style-type: none"> - Voor het automatisch bepalen van de configuratie moet de time serie data met de API request meegestuurd worden. - De request bevat het referentie ID.
Resultaat	Configuratie parameters waarmee een accurate voorspelling gedaan kan worden.
Start	De client doet een API request voor het automatisch bepalen van de parameters.
Proces	
Nr	Beschrijving
1	De client doet een API request voor het automatisch bepalen van de parameters met de time serie data.
2	J-Predictor controleert of de data de juiste structuur hebben en of het referentie ID bestaat. Zo niet dan moet de client terug naar stap 1.
3	J-Predictor controleert met K-fold cross validation welke parameters de beste voorspelling geven.
4	Het auto-scaling verstuurt de uitkomst naar de client.

UC16 - Voorspelling via GUI

Beschrijving	J-Predictor doet een voorspelling aan de hand van de geüploade time serie data.
Actors	<ul style="list-style-type: none"> - Gebruiker - J-Predictor
Voorwaarden	- De time serie data moet de volgende structuur hebben per regel, waarbij de X en Y waarden het type float zijn: <X waarde>, <Y waarde >.
Resultaat	J-Predictor geeft het resultaat in de vorm van een grafiek en tekst in de volgende structuur: timestamp,value.
Start	De gebruiker opent het formulier waarmee de voorspelling gedaan kan worden.
Proces	
Nr	Beschrijving
1	De gebruiker upload de time serie data.
2	De gebruiker klikt op de knop 'voorspellen'
3	J-Predictor controleert of de data de juiste structuur heeft. Zo niet dan moet de gebruiker terug naar stap 1.
4	J-Predictor doet de voorspelling aan de hand van de time serie data.
5	J-Predictor geeft het resultaat in de vorm van een grafiek en tekst.

UC17 - Voorspelling via API

Beschrijving	J-Predictor doet een voorspelling aan de hand van de time serie data die de client stuurt via een API request.
---------------------	--

Actors	- Client - J-Predictor
Voorwaarden	- De time serie data moet de volgende structuur hebben per regel, waarbij de X en Y waarden het type float zijn: <X waarde >, <Y waarde >. - De request bevat het referentie ID.
Resultaat	J-Predictor geeft het resultaat in de vorm van een grafiek en tekst in de volgende structuur: <float>, <float>.
Start	De client doet een API request voor het doen van de voorspelling. Bij dit request wordt de time serie data en referentie ID meegegeven.
Proces	
Nr	Beschrijving
1	De client doet een API request voor het doen van de voorspelling. Bij dit request wordt de time serie data en referentie ID meegegeven.
2	J-Predictor controleert of de data de juiste structuur heeft en of het referentie ID bestaat. Zo niet dan moet de client terug naar stap 1.
3	J-Predictor doet de voorspelling aan de hand van de time serie data.
4	J-Predictor geeft het resultaat in de vorm van een grafiek en tekst.

UC18 - Automatisch voorspelling

Beschrijving	J-Predictor doet een voorspelling voor het ingestelde voorspellingsinterval en slaat het resultaat op in een database.
Actors	- J-Predictor
Voorwaarden	
Resultaat	De voorspelling waarbij de X en Y waarden een float is. J-Predictor slaat de voorspelling op in een database.
Start	De job scheduler start de voorspelling volgens de ingestelde interval.
Proces	
Nr	Beschrijving
1	De client stuurt het J-Predictor 10 minuten voordat er een voorspelling gedaan wordt nieuwe time serie data toe.
2	J-Predictor controleert of de beschikbare data de juiste structuur hebben en of het referentie ID bestaat. Zo niet dan kan er geen voorspelling gedaan worden en wordt er een waarschuwing gegeven.
3	J-Predictor doet de voorspelling aan de hand van de time serie data.
4	J-Predictor slaat de resultaten op in de database

UC19 - Voorspelling accuraatheid check

Beschrijving	J-Predictor controleert of de voorspelling accuraat genoeg is en geeft hier een waarschuwing over als dat niet zo is, zodat de parameters getuned kunnen worden.
Actors	- Client - J-Predictor
Voorwaarden	- Deze check gebeurt niet bij elke voorspelling, want het gaat ten kosten van de performance.
Resultaat	Een waarschuwing met de accuraatheid van de voorspelling
Start	Er wordt een voorspelling gedaan
Proces	
Nr	Beschrijving
1	Er wordt een voorspelling gedaan
2	J-Predictor controleert of de voorspelling accuraat is in vergelijking met vorige voorspellingen.
3	Als de voorspelling niet accuraat is geeft J-Predictor een waarschuwing.

UC20 - Automatisch configuratie tuning

Beschrijving	J-Predictor tuned zelf de parameters als de voorspelling niet accuraat genoeg is.
Actors	- J-Predictor
Voorwaarden	
Resultaat	Parameters waarmee een betere voorspelling gedaan kan worden.
Start	J-Predictor merkt op dat de voorspelling niet accuraat genoeg is.
Proces	
Nr	Beschrijving
1	J-Predictor start het proces voor het automatische bepalen van de configuratie.
2	J-Predictor vindt parameters waarmee een accurate voorspelling gedaan kan worden.
3	J-Predictor stelt de parameters voor de voorspelling.

7. Business rules

Use Case	Business rule
Algemeen	De GUI's en API's moet alleen beschikbaar zijn binnen het netwerk van Jitscale.

Naam: Rogier Schipper
Studentnummer: 1583842
Opleiding: Informatica Deeltijd
Datum: 12-10-2014

Algemeen	Bij een API's request moet het referentie ID meegegeven worden bij hetgeen wat opgevraagd of gewijzigd worden.
Algemeen	Time serie moet de volgende structuur hebben, waarbij de X en Y waarden het type float zijn: <X waarde >, <Y waarde >.
Algemeen	Bij het aanmaken van een nieuwe object via de API geeft J-Predictor het ID van het aangemaakte object.
Algemeen	De API moet de zelfde functies bevatten als de GUI.
UC1, UC3	Minimum instances moet minder of gelijk zijn aan maximum instances.
UC1	De predictive auto-scaler wordt pas aangemaakt als het ID van de predictor in J-Predictor ontvangen is.
UC2	Er moet monitoring time series beschikbaar zijn in J-Layer.
UC4	De X waarden van de time serie moet een integer zijn.
UC5	De begin periode moet lager zijn dan de eindperiode.
UC6	Elke predictor heeft een uniek referentie ID.
UC7	Voordat er voorspeld kan worden met een predictor moet deze ingesteld zijn met UC10.
UC9	De API request bevat time serie data.
UC10	Voorspellingsinterval moet voor minuten ≤ 60 zijn en voor uren ≤ 24 .
UC11	Voorspellingsinterval moet voor minuten ≤ 60 zijn en voor uren ≤ 24 .
UC11	Relevantie van historische data: Moet float zijn welke ≥ 0 en ≤ 100
UC11	Historische relevantie afname: Moet float zijn welke ≥ 0 en ≤ 100
UC13	J-Predictor verstuurt afbeeldingen van de grafieken via de API.
UC14 UC15	Het automatisch bepalen van de configuratie gebeurt op de achtergrond.
UC16 UC17	Een voorspelling wordt weergegeven in een grafiek en als tekst.
UC18	De X en Y waarde van de voorspelling zijn beide een float.
UC18	De job scheduler doet voorspellingen op ingestelde interval.
UC19	J-Predictor houdt in de gaten of een voorspelling accuraat is en geeft hier een waarschuwing over als dat niet het geval is.

8. Functionele beschrijving

J-Predictor en J-Layer werken samen aan het realiseren van de oplossingen, waarbij de taken van J-Predictor zijn:

- Het ontvangen en opslaan van de time series.
- Het analyseren van time series.
- Het doen van een voorspelling op basis van een time serie op het ingestelde tijdstip.
- Het opslaan en beschikbaar stellen van voorspellingen.
- Het analyseren van voorspellingen op juistheid.

En de taken van J-Layer zijn:

- Het aanmaken van een predictor en deze koppelen aan een IT omgeving binnen J-Layer.
- Het versturen van monitoring time series naar de J-Predictor op het ingestelde tijdstip.
- Het opvragen van de voorspelling uit J-Predictor.
- Het scaling plan bepalen aan de hand van de ingestelde instance threshold.
- Het starten van de instances volgens het scaling plan. (Dit is al mogelijk in J-Layer en valt daarom buiten de scope van dit project.)

8.1. Taken van J-Predictor

8.1.1. Ontvangen en opslaan van de time series

Het ontvangen en opslaan van de time series stelt J-Predictor in staat om de time serie te ontvangen van een externe bron (client). Dit kan een applicatie zijn die de time serie verstuurd naar J-Predictor, of een gebruiker(client) die een time serie uploadt via de GUI. J-Predictor zal in staat zijn om meerdere time series te onthouden per predictor, waardoor de gebruiker kan kiezen welke time serie gebruikt moet worden voor de voorspelling. De gebruiker kan onderscheid maken in de time series door ze een naam te geven. Daarbij zal het mogelijk zijn om time serie data incrementeel toe te voegen, zodat nieuwe data gecombineerd wordt met oude data. Het voordeel hiervan is dat de time serie datasets die verstuurd moeten worden minder groot zijn en de externe applicatie of gebruiker niet hoeft te beschikken over de oude data.

Het aanmaken van een nieuwe time serie voor een predictor (het object dat de gekozen voorspellingsmethode en configuratie bevat) wordt gedaan in het scherm 'Add Time Serie', waarbij het ook mogelijk is om een time serie te uploaden. De data die deze time serie bevat zal te zien zijn op de pagina 'Time Serie', waarvan het schermontwerp te zien is in paragraaf 12. Per time serie kan de voorspellingsmethode geconfigureerd worden, waarbij de instellingen alleen gelden voor die time serie. De reden hiervoor is omdat de voorspellingsmethode aan de hand van die instellingen een juiste voorspelling kan doen. Als een voorspelling gedaan wordt aan de hand van meerdere time series zal de voorspellingsmethode dus per time serie bekijken wat de instellingen zijn. De parameters die ingesteld kunnen worden zijn afhankelijk van de gebruikte voorspellingsmethode. De configuratie kan gedaan worden op de pagina 'Prediction Method Settings'. In het schermontwerp van deze pagina zijn de parameters te zien van de voorspellingsmethode Clustered linear regression.

8.1.2. analyseren van time series

Het analyseren van time series stelt de gebruiker in staat om aan de hand van een set aan statistische tools de settings te bepalen voor het configureren van de predictor. Dit wordt gedaan op de pagina 'Time Serie Analyzer'. Bij het configureren van een predictor moet de seasonality van de time serie ingesteld worden. De seasonality kan bepaald worden aan de hand van de time serie, maar doordat dit niet altijd even duidelijk te zien is in de time serie is het nodig om statistische tools te gebruiken om dit wel zichtbaar te maken. J-Predictor geeft de gebruiker toegang tot een set aan statistische tools waarmee de gebruiker de time serie kan analyseren. Met de autocorrelation functie is het bijvoorbeeld mogelijk om de seasonality zichtbaar te maken. Bij sommige time series is het van belang dat de time serie eerst bewerkt wordt door bijvoorbeeld te differentiëren, voordat de seasonality zichtbaar wordt met de autocorrelation

functie. Voor die reden zal het mogelijk zijn voor de gebruikers om zelf de volgorde te bepalen waarin de statistische tools uitgevoerd worden. Hierbij zijn differentiëren en de autocorrelation functie de statische tools.

8.1.3. Het doen van een voorspelling

Het doen van een voorspelling op basis van een time serie op het ingestelde tijdstip zorgt ervoor dat een voorspelling gedaan wordt op het tijdstip dat de J-Layer de voorspelling nodig heeft. Aan de hand van een taak planner (job scheduler) wordt het doen van een voorspelling automatisch gestart. De voorspelling zullen te zien zijn via de GUI in het scherm 'Predictions'. Hierbij kan aangegeven worden voor welke datum en tijd de voorspelling te zien moet zijn.

8.1.4. Opslaan en beschikbaar stellen van voorspellingen

De voorspelling zal opgeslagen worden in de database maar niet automatisch verstuurd worden naar de client. De cliënt zal zelf de voorspelling moeten opvragen bij J-Predictor. Hiermee wordt de verantwoordelijkheid van het ophalen van de voorspelling bij de cliënt gelegd. De reden hiervoor is dat de cliënt niet afhankelijk is van de planning van J-Predictor voor het versturen van de time serie, maar in plaats daarvan op elk moment van de dag de voorspellingen kan opvragen. J-Predictor is dus niet verantwoordelijk voor het versturen van de voorspellingen naar alle cliënten. Als J-Predictor dat wel zou zijn en er gaat iets mis in de planning voor het versturen zouden alle cliënten hier last van hebben, wat zorgt voor een single point of failure. (Wikipedia, 2014)

8.1.5. Analyseren van voorspellingen op juistheid

Bij het doen van een voorspelling is het van belang dat de time serie data bevat die juist is. Als er een storing is geweest kan het zijn dat de time serie data mist waardoor de time serie niet juist is. Dit kan ervoor zorgen dat de voorspelling voorspeld dat er minder HTTP requests zullen gaan zijn, omdat op het moment van de downtime er ook minder HTTP request waren. Om dit te voorkomen zal J-Predictor controleren of de time serie overeen komt met de voorspelling, zoals beschreven wordt in paragraaf 11- Outlier detectie.

Daarnaast kan het ook zijn dat de time serie wel juist is maar dat de voorspelling niet accuraat is in vergelijking met de werkelijkheid. Dit kan komen doordat de instellingen van de voorspellingsmethode niet juist zijn, zoals de seasonality. Om te testen of een voorspellingsmethode de juiste instellingen heeft gebruikt J-Predictor K-fold cross validation. Met K-fold cross validation worden meerdere voorspellingen gedaan over de gehele time serie, waarbij gewisseld wordt tussen de data waarmee de voorspelling gedaan wordt. Het is ook mogelijk dat de seasonality of trend van de time serie veranderd waardoor de settings opnieuw ingesteld moeten. Met K-fold cross validation is het in zo'n situatie mogelijk automatisch nieuwe instelling te vinden door te testen welke instellingen de meest accurate voorspellingen geeft.

8.2. Taken van J-Layer

8.2.1. Aanmaken van een predictor

Vanuit J-Layer wordt bij J-Predictor via de API een predictor aangemaakt. J-Layer zorgt voor de koppeling tussen de predictor en de predictive_scaler door het ID van de predictor op te slaan. Dit wordt gedaan vanuit de het scherm 'New Auto-Scaler'. In dit scherm moet gekozen worden voor het auto-scaling type 'Predictive', een voorspellingsmethode en aangegeven worden hoeveel instances minimaal gestart moeten/mogen zijn. Met het minimaal aantal instances wordt ervoor gezorgd dat er bijvoorbeeld altijd 2 instances draaien als een cluster voor het geval er 1 van de 2 uitvalt en er direct een instance nodig is voor het overnemen van de taken. Met het maximaal aantal instances wordt ervoor gezorgd dat niet oneindig veel instance gestart worden, zodat hoge kosten voorkomen worden bij bijvoorbeeld een DDoS-aanval. (Microsoft, 2014), (Wikipedia, Distributed denial-of-service, 2014)

Na het aanmaken van de predictive_scaler kan deze samen met de predictor geconfigureerd worden in het scherm 'Predictive scaler configuration'. In dit scherm kan naar het minimale en maximale instance de

instance threshold, prediction period en prediction interval ingesteld worden. Prediction period en prediction interval zijn parameters vanuit J-Predictor dus deze wordt via de API ingesteld in J-Predictor.

Met de 'instance threshold' kan aangegeven worden bij welke waarde een extra instance gestart moet worden. Als de threshold 1500 (HTTP requests) is en de voorspelling is 3400 (HTTP request) dan betekent dit dat er 3 instances gestart moeten worden, want $3400/1500$ is 2.2 wat betekent dat er meer dan 2 instances nodig zijn.

Met de 'prediction period' wordt in minuten aangegeven voor welke periode een voorspelling uitgevoerd kan worden. Met de start en eind periode wordt aangegeven over welke periode een voorspelling gedaan moet worden. Start periode 1 tot eind periode 60 betekent dat er een voorspelling gedaan moet worden voor het komende uur.

Met de 'prediction interval' wordt aangegeven om de hoeveel tijd J-Predictor een voorspelling moet doen. Hierbij worden vaste waarden gebruikt zoals elk uur of elke 3 uur. De reden hiervoor is zodat de periodes altijd even groot zijn en op het zelfde moment op en dag. Bij een interval van 18 uur zou de voorspelling de eerste dag om 18:00 uur zijn en de tweede dag om 12:00. Dit is niet consistent en kan voor verwarring zorgen bij bijvoorbeeld het versturen van time serie naar J-Predictor voor het doen van een voorspelling.

8.2.2. Versturen van monitoring time series naar de J-Predictor

Het versturen van monitoring time series naar de J-Predictor op het ingestelde tijdstip wordt gedaan door de scheduler van J-Layer, en aan de hand van de 'prediction interval'. J-Layer zorgt ervoor dat de time serie 10 minuten voordat de voorspelling gedaan wordt verstuurd zijn. Hierbij wordt oude time serie data in J-Predictor overschreven als de X-as waarde gelijk is aan een al bestaande X-as waarde in de database. Hiermee wordt ervoor gezorgd dat J-Predictor altijd over de meest recente (Y-as) data bevat. De Y-as bevat de data waarop de voorspelling gebaseerd wordt.

8.2.3. opvragen van de voorspelling uit J-Predictor

Het opvragen van de voorspelling uit J-Predictor wordt ook gedaan aan de hand van de 'prediction interval'. J-Layer zal 1 minuut na dat de voorspelling gedaan is door J-Predictor de voorspelling opvragen aan de hand van de start en eind periode van de voorspelling. Aan de hand van de 'prediction period' (voorspellingsperiode) controleert J-Predictor of er al een voorspelling is die verstuurd kan worden naar J-Layer. J-Layer doet elke 30 seconden een request naar J-Predictor voor het opvragen van de voorspelling, totdat J-Layer de voorspelling ontvangen heeft.

8.2.4. Scaling plan bepalen aan de hand van de ingestelde instance threshold

Het scaling plan wordt bepaald door aan de hand van de ingestelde 'instance threshold' en de voorspelling te bepalen hoeveel instances nodig zijn. De X-as van de voorspelling geeft de tijd aan en de Y-as de behoeften zoals de 3400 HTTP requests wat eerder besproken is in punt 1. Na het bepalen van het scaling plan wordt deze opgeslagen binnen J-Layer.

8.2.5. Starten van de instances volgens het scaling plan

Na het bepalen van het scaling plan zorgt de scheduler dat de instance(s) op het juist tijdstip bijgeschaald worden. Het schalen van de instances valt buiten de scope van dit project aangezien dit al een werkende functie binnen J-Layer is, dus de werking hiervan zal niet besproken worden.

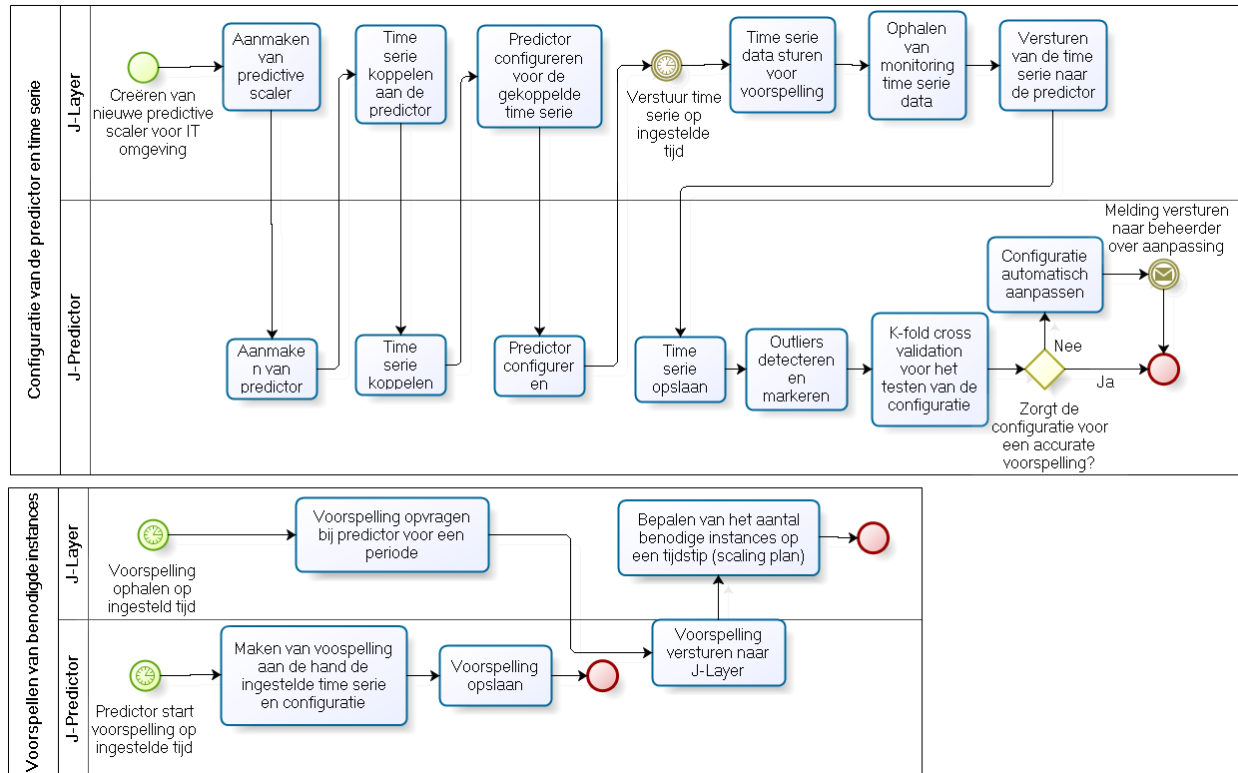
8.3. Functioneel proces

In de afbeelding 8.1 zijn twee proces modellen te zien die gemaakt zijn aan de hand van de use cases. In de proces modellen is te zien hoe de taken van J-Layer en J-Predictor samen zorgen voor het bepalen van het aantal benodigde instances. Het realiseren van een voorspelling van het benodigde aantal instances bestaat uit het voorbereiden van de voorspelling:

- Configureren van de predictor en time serie.

En de voorspelling:

- Voorspellen van benodigde instances.



Afbeelding 8.1 - Functioneel proces

9. Voorspellingsmethode

De voorspellingsmethode die geïmplementeerd wordt binnen J-Predictor is Clustered linear regression (CLR). De voorspellingsmethode zal in staat zijn tot de volgende punten:

- Voorspellen aan de hand van time series.
- Voorspellen op basis van seasonality.
- Het minimale en maximale aantal datapunten in een clusters bepalen. Met het aantal datapunten in een cluster wordt het gewicht van een cluster bepaald. Hoe meer datapunten hoe meer gewicht een cluster heeft.
- Bepalen hoe belangrijk historische data zijn en hoe snel historische data, over tijd, minder belangrijk worden. Hiermee wordt bepaald hoe snel de cluster grootte (gewicht) afneemt over een bepaald periode. Zo kan ingesteld worden dat elke dag terug in het verleden het data cluster 1 datapunt minder krijgt, waardoor de meest recente data het meest belangrijk zijn voor de voorspelling doordat recente clusters meer datapunten hebben.
- K-fold cross validation voor het valideren van de ingestelde parameters: seasonality, cluster grootte, en gewicht van historische data. Hierbij wordt met de root mean square error bepaald hoe accuraat de voorspelling is met de ingestelde parameters
- Outliers detecteren.

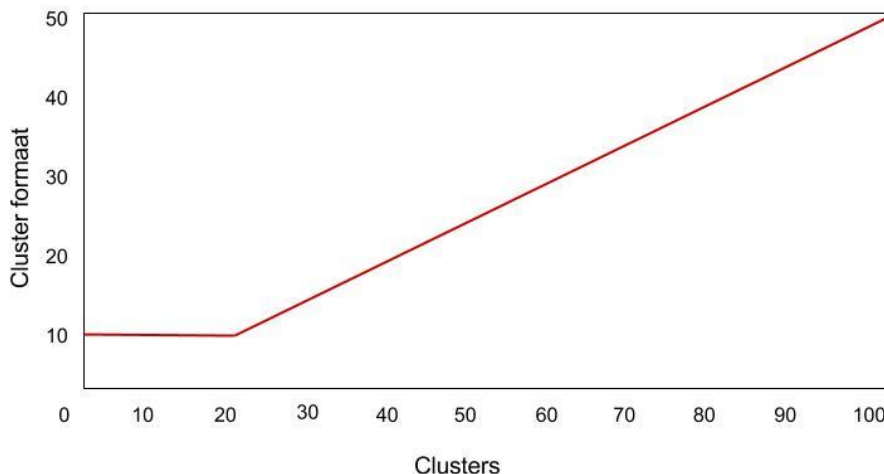
De parameters die gebruikt worden zijn voor het instellen van de voorspellingsmethode zijn:

- **Seasonality:** Hiermee kan aangegeven worden wat de periode van de seasonality is zoals elke 7 dagen. De verschillende mogelijke seasonalities zijn: minuten, uren, dagen, weken, maanden, jaren.
- **Maximale cluster grootte:** Hiermee wordt aangegeven hoeveel datapunten een cluster maximaal mag zijn.
- **Minimale cluster grootte:** Hiermee wordt aangegeven hoeveel datapunten een cluster minimaal moet hebben.
- **Historische relevantie:** Hiermee wordt aangegeven tot waar in de time serie de data relevant zijn voor de voorspelling. Als hierbij wordt aangegeven dat 80% van de time serie relevant is zal 80% van het aantal cluster de maximale grootte zijn en 20% de minimale grootte.
- **Historische relevantie afnamen:** Hiermee wordt aangegeven hoe snel de relevantie van historische data afneemt. Als hier aangegeven wordt dat de historische relevantie afneemt met een snelheid van 100%, betekent dit dat cluster verder in het verleden steeds kleiner worden. Waarbij het laatste cluster gelijk is aan de minimale cluster grootte en de meeste recente cluster gelijk is aan de maximale cluster grootte.
- **Begin en einde van de voorspelling:** Hiermee wordt de periode aangegeven die voorspeld moet worden. Als het begin 1 is en het einde 60 betekent dit dat er een voorspelling gedaan wordt voor een uur in de toekomst.

In afbeelding 9.1 is ter verduidelijking te zien hoe de cluster grootte afneemt bij met de volgende instellingen:

- Historische relevantie afname: 100%
- Historische relevantie: 80%
- Maximale cluster grootte: 50
- Minimale cluster grootte: 10

Zoals in afbeelding 9.1 te zien is, is de laatste 20% (van clusters 0 tot 20) de minimale cluster grootte. Dit komt doordat maar 80% van de data relevantie is en de laatste 20% niet grootte mag zijn dan de minimale cluster grootte. Als de historische relevantie afnamen snelheid 0% zou zijn zou de rode lijn tussen cluster 20 en 100 horizontaal zijn met een cluster formaat van 50, en tussen 0 en 20 ook horizontaal met een cluster formaat van 10.



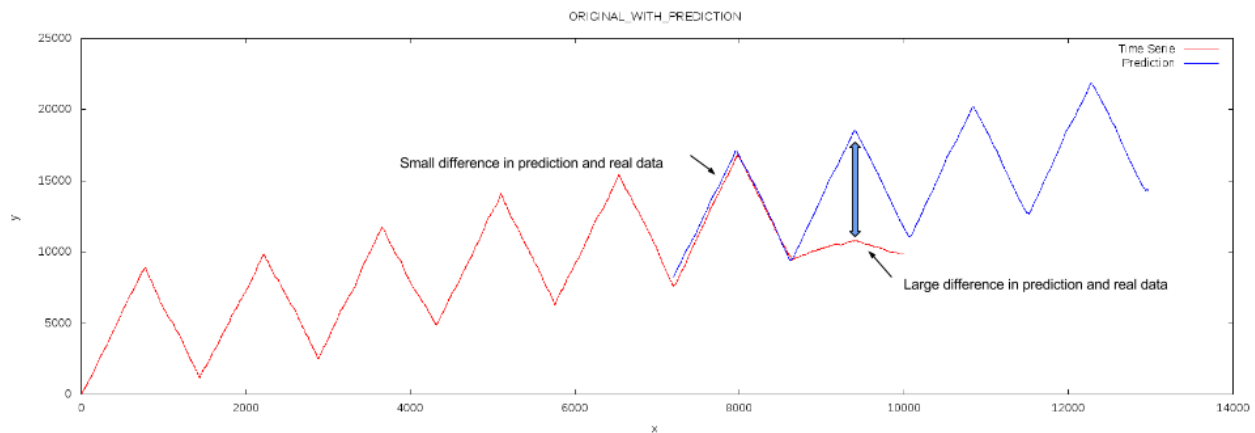
Afbeelding 9.1 – Historische relevantie

10. Voorspellingsmodel bepalen

Voor het doen van een voorspelling moet een model bepaald worden dat gebruikt wordt voor de voorspelling. Een model bestaat uit de instellingen die nodig zijn voor het doen van een voorspelling. Voor CLR zijn dat de instellingen die in hoofdstuk 9 besproken zijn. Doordat bij een verkeerd model kans is op under- of overfitting is het van belang dat een model getest wordt. Voor het testen zal K-fold cross validation gebruikt worden, zodat het juiste model bepaald kan worden. Daarbij heeft J-Predictor als functionele eis (FE20) dat de beste settings (het beste model), waarmee de meest accurate voorspelling gedaan kan worden, automatisch bepaald kan worden. Voor deze functionele eis zal ook K-fold cross validation gebruikt worden, waarbij K-fold cross validation verschillende instellingen uitprobeert en aan geeft welk model de meest accurate voorspelling zal geven.

11. Outlier detectie

Het detecteren van outliers zal gedaan worden door te kijken naar het verschil tussen de time serie en de voorspelling. Op het moment dat er een nieuwe time serie data toegevoegd wordt voor een periode waar al een voorspelling voor is, kan bekeken worden wat het verschil is tussen de time serie en de voorspelling. Dit is te zien in afbeelding 11.1. De laatste seasonality is in vergelijking met vorige seasonalities een stuk lager waardoor er een groot verschil is tussen de voorspelling en de time serie, wat niet het geval is bij de voorspelling van de vorige seasonality.



Afbeelding 11.1 - Error of prediction

12. Schermontwerpen

12.1. New predictor

The screenshot shows a web browser window titled "New Predictor". The address bar contains the URL "http://J-Predictor.jitscale.com/new". The main content area has the heading "New Predictor". Below the heading, there is a form with two fields: "Name:" followed by a text input box, and "Prediction method:" followed by a dropdown menu showing "Clustered Linear Regression". A "Create" button is located below the dropdown menu.

12.2. Configure predictor

The screenshot shows a web browser window titled "Configure Predictor". The address bar contains the URL "http://J-Predictor.jitscale.com/configure_predictor". The main content area has the heading "Configure Predictor" and a "Dashboard" button. Below the heading, there are four tabs: "Configure predictor", "CLR Settings", "Add time serie", and "Analyze time serie". The "Configure predictor" tab is active. The form contains the following fields and controls:

- Prediction period:** A section with "start" and "end" labels. The "start" field has a value of "0" and a "minutes" dropdown. The "end" field has a value of "60" and a "minutes" dropdown.
- Prediction interval:** A dropdown menu showing "15 minutes" as the selected option. The dropdown list includes: "30 minutes", "1 hour", "2 hours", "3 hours", "6 hours", "12 hours", and "24 hours".
- Time series used for prediction:** A section with three toggle switches:
 - ☒ HTTP requests
 - ☐ Memory usage
 - ☒ Disk I/O
- A "Save" button is located at the bottom right of the form.

12.3. Prediction method settings

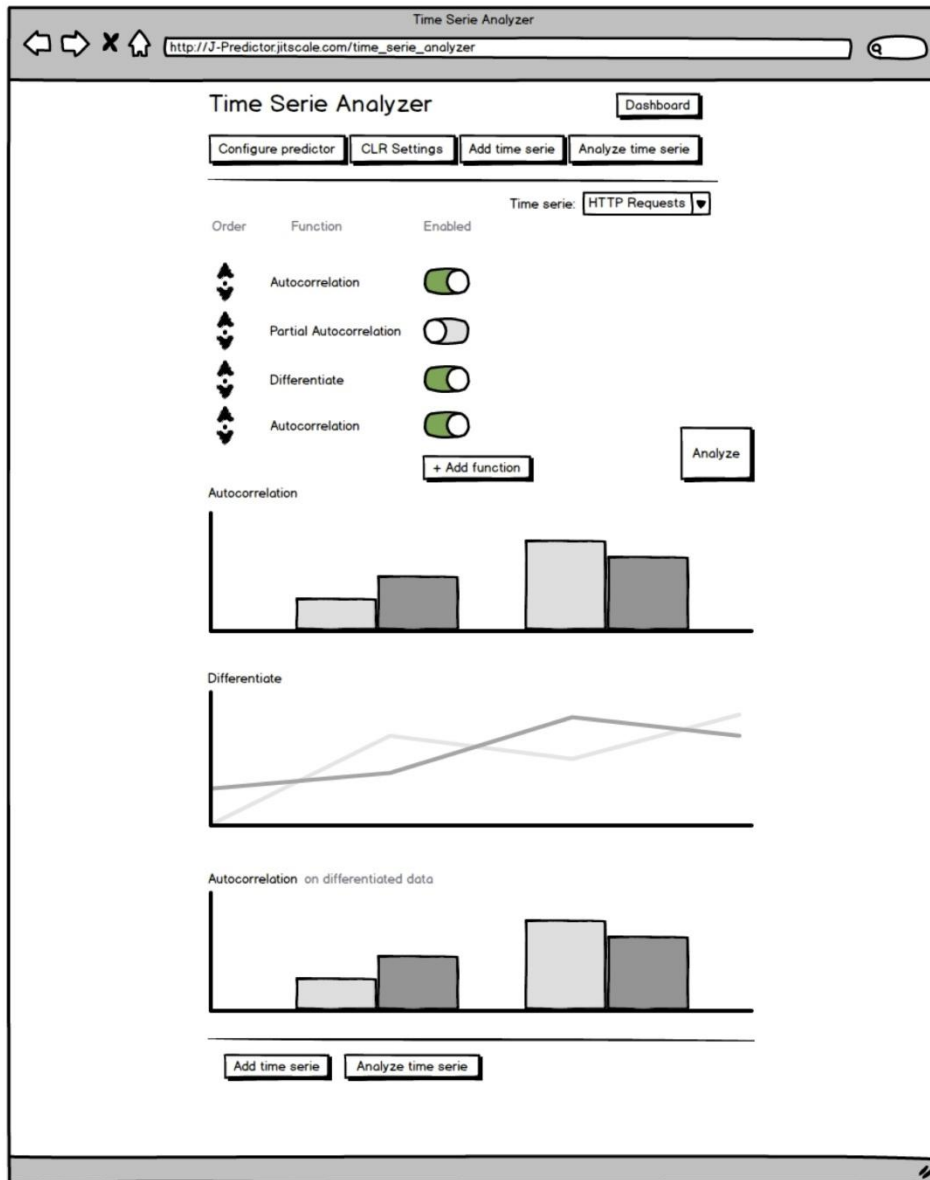
The screenshot shows a web browser window with the address bar displaying `http://J-Predictor.jitscale.com/prediction_method_settings`. The page title is "Prediction Method Settings". At the top right is a "Dashboard" button. Below the title are four tabs: "Configure predictor" (active), "CLR Settings", "Add time serie", and "Analyze time serie".

The main settings area includes:

- Use for prediction:** A toggle switch that is currently turned on.
- History relevance:** A horizontal slider.
- History relevance decline:** A horizontal slider.
- Minimum cluster size:** A text input field containing the value "10".
- Maximum cluster size:** A text input field containing the value "50".
- Seasonality:** A text input field containing "7" and a dropdown menu set to "daily".
- Time serie:** A dropdown menu set to "HTTP Requests".
- Test accuracy:** A box displaying "96%".

At the bottom of the settings area are two buttons: "Auto tune configuration" and "Save". To the right of the sliders is a small line graph showing a fluctuating trend.

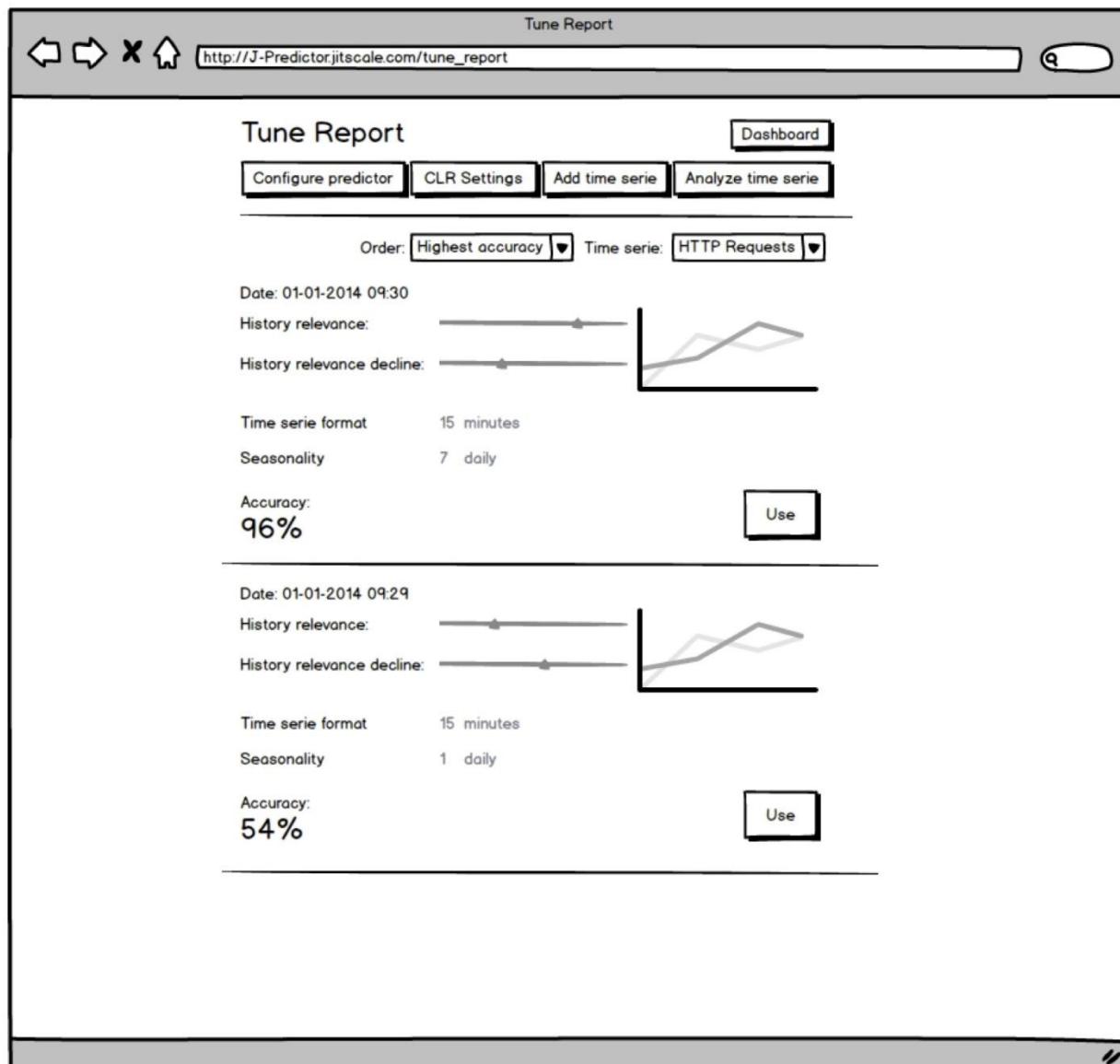
12.4. Time Serie Analyzer



12.5. Auto tune settings

The screenshot shows a web browser window with the address bar displaying `http://J-Predictor.jitscale.com/auto_tune_settings`. The page title is "Auto Tune Settings". At the top right, there is a "Dashboard" button. Below the title, there are four tabs: "Configure predictor", "CLR Settings", "Add time series", and "Analyze time series". The "CLR Settings" tab is currently selected. On the left, under the heading "Tune", there are four toggle switches, all of which are turned on. These are labeled: "History relevance:", "History relevance decline:", "Time serie format", and "Seasonality". To the right of these toggles are two horizontal sliders for "History relevance:" and "History relevance decline:". Further right, there are two input fields: "15" and "7". To the right of these are two dropdown menus: "minutes" and "daily". A "Tune" button is located below these settings. On the far right, there is a "Time serie:" label followed by a dropdown menu showing "HTTP Requests". To the right of the sliders and input fields is a small line graph showing a fluctuating trend.

12.6. Tune report

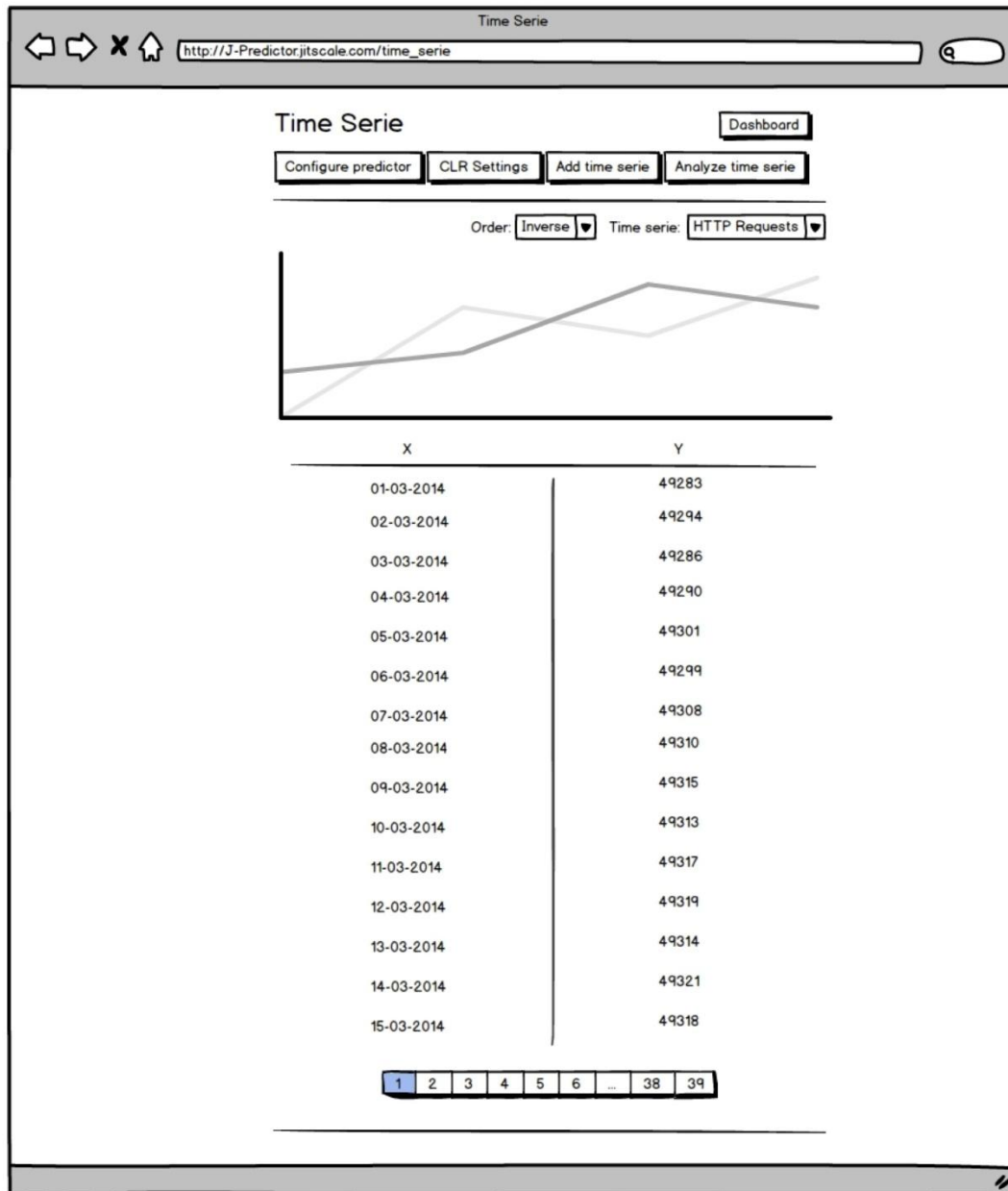


12.7. Add time serie

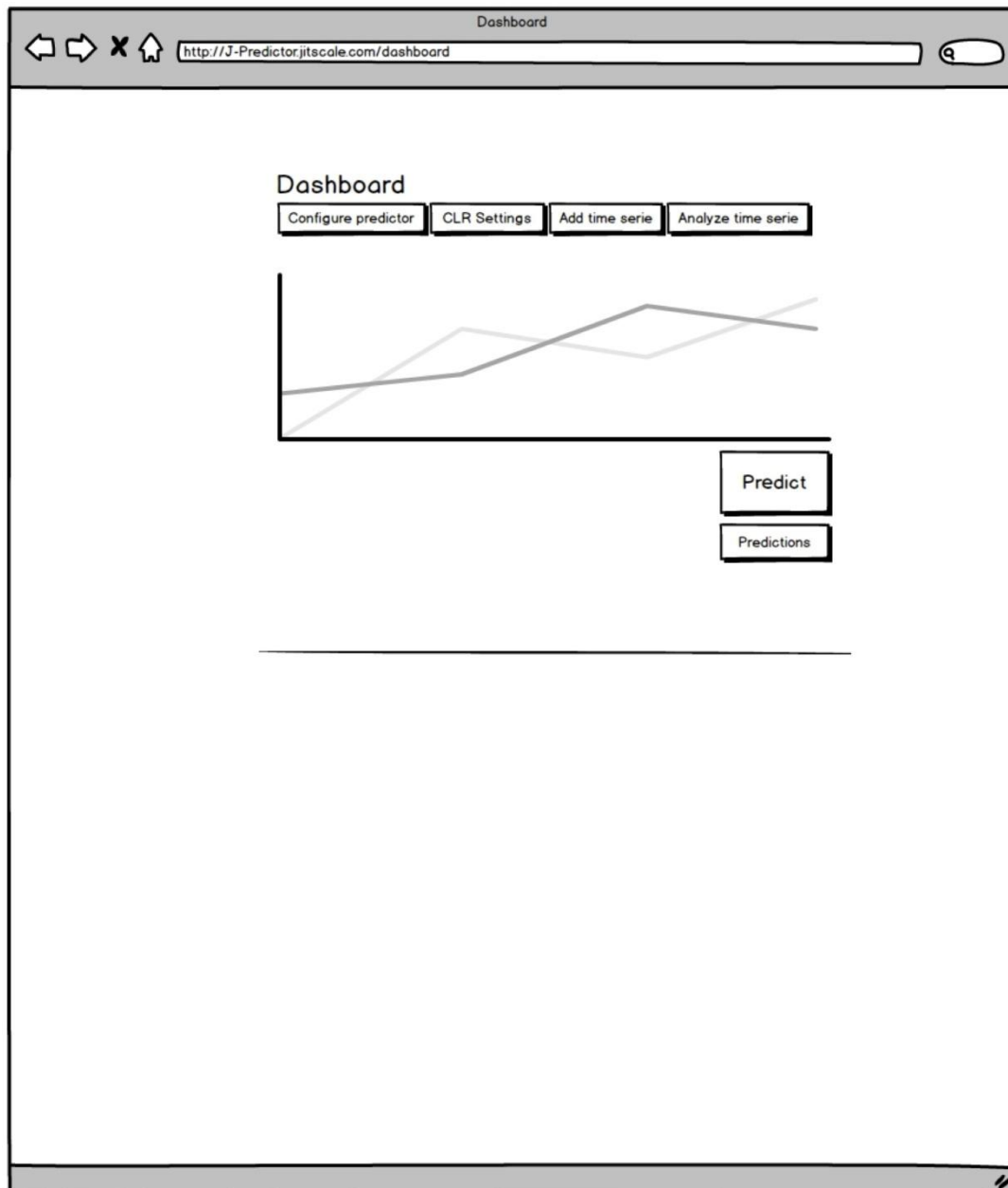
The screenshot shows a web browser window with the title 'Add Time Serie'. The address bar contains the URL 'http://J-Predictor.jitscale.com/add_time_serie'. The page content includes a navigation bar with buttons for 'Configure predictor', 'CLR Settings', 'Add time serie', and 'Analyze time serie'. A 'Dashboard' button is also present. The main form area has the following fields and controls:

- Name:** A text input field containing 'HTTP Requests'.
- Time serie format:** A dropdown menu showing '15' and 'minutes'.
- Upload:** A button labeled 'Select time serie' followed by an 'upload' button and the text '(optional)'.
- Create:** A button at the bottom of the form.

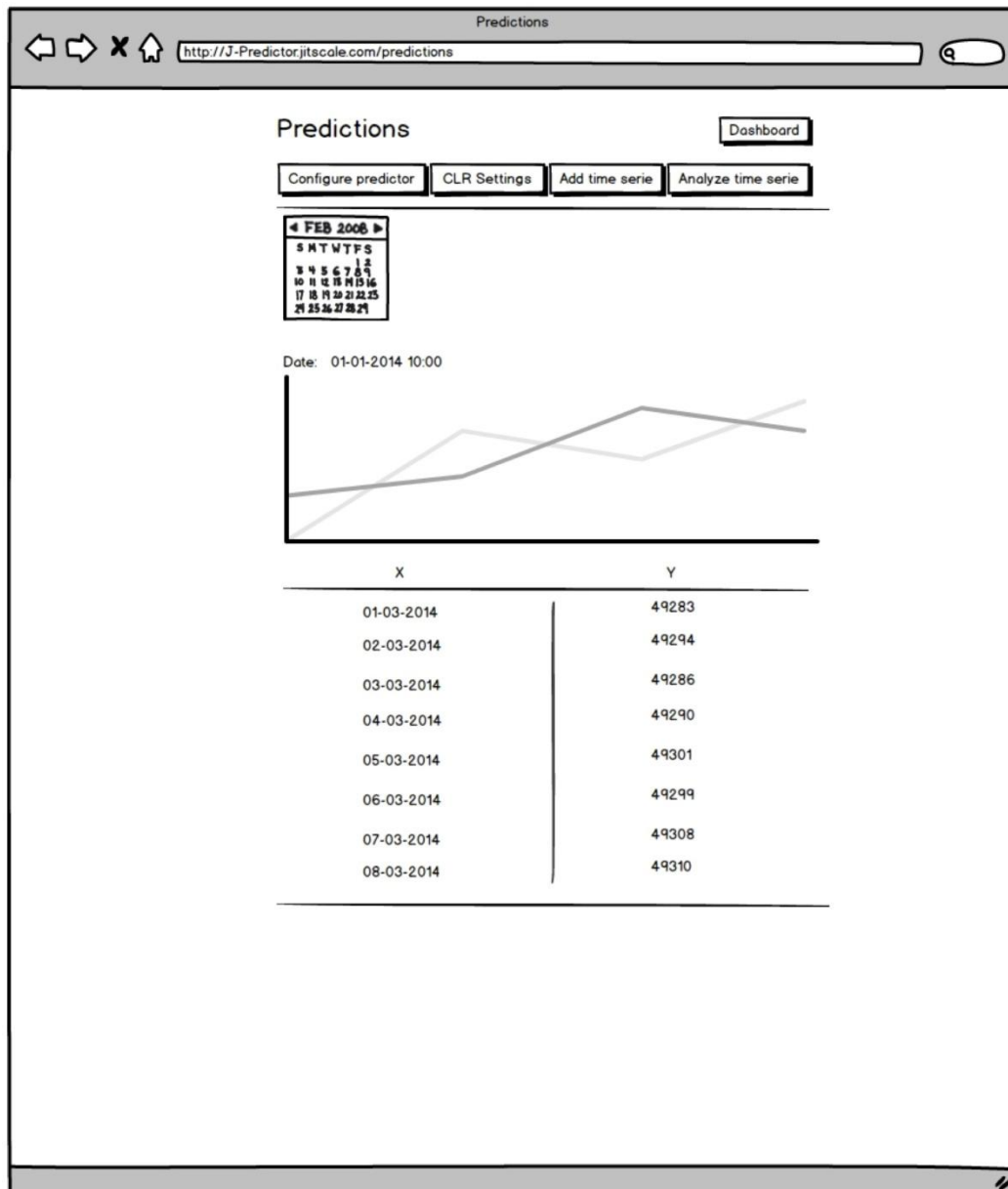
12.8. Time serie



12.9. Dashboard



12.10. Predictions



12.11. New auto-scaler

The screenshot shows a web browser window with the address bar displaying `http://J-Layer.jitscale.com/new_auto_scaler`. The page title is "New Auto-Scaler". The form contains the following fields and options:

- Name:** A text input field containing "Jitscale Corporate Websites".
- Type:** Two radio buttons: "Predictive" (selected) and "Responsive".
- Prediction method:** A dropdown menu showing "Clustered Linear Regression".
- Minimum instances:** A text input field containing "1".
- Maximum instances:** A text input field containing "10".
- Save:** A button located at the bottom right of the form.

12.12. Predictive scaler configuration

The screenshot shows a web browser window with the address bar displaying `http://J-Layer.jitscale.com/predictive_scaler_configuration`. The page title is "Predictive Scaler Configuration". The form includes a navigation bar with buttons: "Configure predictor", "CLR Settings", "Add time serie", and "Analyze time serie". A "Dashboard" button is also present in the top right. The main configuration area contains the following fields and options:

- Instance threshold:** A text input field containing "1500".
- Minimum instances:** A text input field containing "1".
- Maximum instances:** A text input field containing "10".
- Prediction period:** Two sets of input fields: "start" with "0" and "minutes" unit, and "end" with "60" and "minutes" unit.
- Prediction interval:** A dropdown menu showing "15 minutes" as the selected option. The dropdown list includes: "30 minutes", "1 hour", "2 hours", "3 hours", "6 hours", "12 hours", and "24 hours".
- Save:** A button located at the bottom right of the form.

12.13. Responsive scaler configuration

Responsive Scaler Configuration

[Dashboard](#)

Up Scaling

Plugin:

Subject:

Operator:

Threshold:

Scale Factor / Max: /

Cool down: Minutes

Breach time / Sample time: seconds within seconds

Down Scaling

Plugin:

Subject:

Operator:

Threshold:

Scale Factor / Max: /

Warming up: Minutes

Breach time / Sample time: seconds within seconds

13. Bibliografie

Microsoft. (2014, Maart 28). *What Is a Server Cluster?* Opgeroepen op Oktober 8, 2014, van Microsoft: [http://technet.microsoft.com/en-us/library/cc785197\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/cc785197(v=ws.10).aspx)

Wikipedia. (2014, Augustus 25). *Distributed denial-of-service*. Opgeroepen op Oktober 8, 2014, van Wikipedia: http://nl.wikipedia.org/wiki/Distributed_denial-of-service

Wikipedia. (2014, Oktober 3). *Single point of failure*. Opgeroepen op Oktober 8, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Single_point_of_failure

Technisch ontwerp

J-Layer auto-scaling

Inhoudsopgave

1. Inleiding	1
2. Beschrijving	1
3. Domein model.....	1
4. Klassendiagram	3
5. Data model	5
6. Logische lagen model	6
7. Strategy pattern implementatie	6
8. Programmeertaal en framework.....	8
9. Database	8
10. RESTful.....	8
11. JSON	8
12. Htaccess security	8
13. Webserver	8
14. Clustered linear regression.....	9

1. Inleiding

Dit document bevat de technische beschrijving van hoe J-Predictor zal werken in combinatie met J-Layer. Aan de hand van modellen en diagrammen wordt beschreven hoe J-Predictor en J-Layer ontwikkeld moeten worden en hoe J-Layer met J-Predictor gekoppeld is. Daarbij wordt ook beschreven welke software nodig zal zijn.

2. Beschrijving

Voor het voorspellen van het aantal benodigde instances is gekozen voor de ontwikkeling van een applicatie, genaamd J-Predictor, waarbij de voorspellingen en het bepalen van het aantal instances aan de hand van de voorspelling plaats zal vinden in J-Layer. Door deze keuzes zal de ontwikkeling van de oplossing zowel J-Layer als J-Predictor overkoepelen. In de modellen zal de koppeling en de scheiding tussen de twee applicaties duidelijk zichtbaar zijn.

3. Domein model

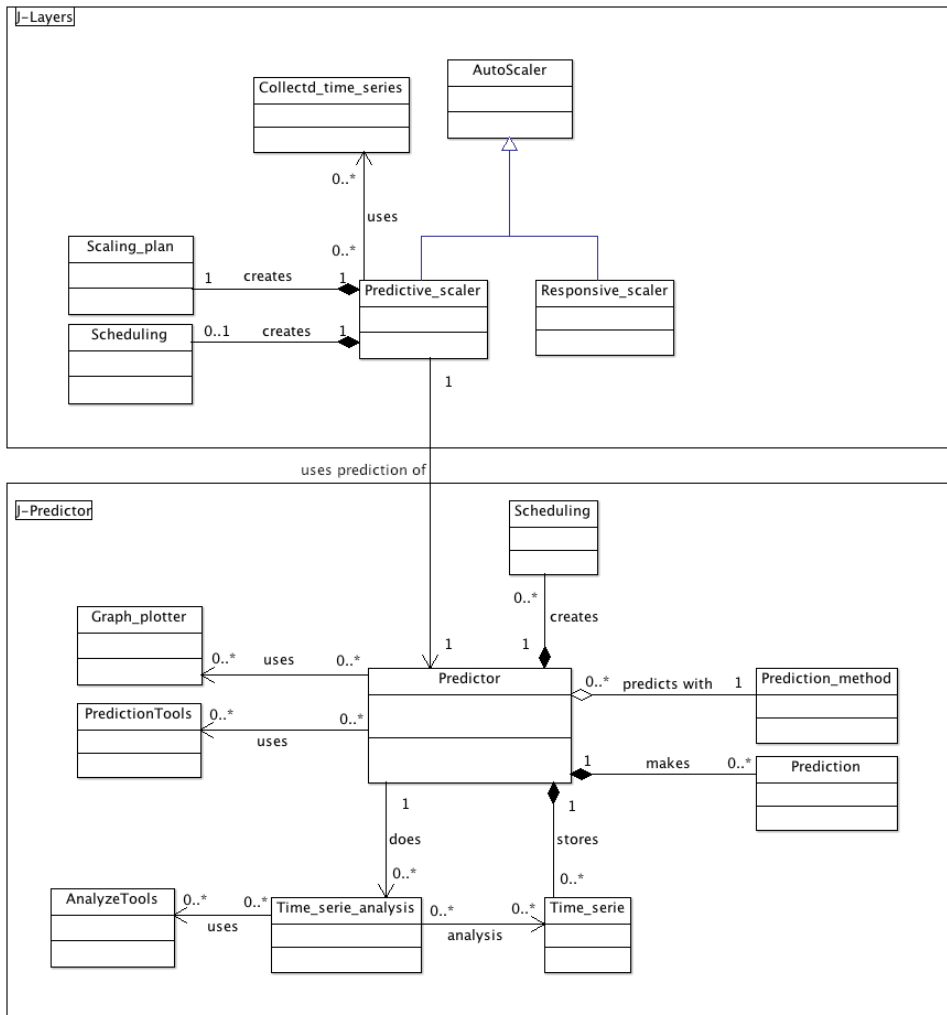
Het domein model (afbeelding 3.1) is ontwikkeld voor het in kaart brengen van de verschillende entiteiten die benodigd zijn. De entiteiten voor J-Layer zijn:

- AutoScaler
- PredictiveScaler
- ResponsiveScaler
- ScalingPlan
- Scheduling
- CollectDTimeSeries

Voor J-Predictor zijn de entiteiten:

- Predictor
- PredictorMethod
- TimeSerie
- Prediction
- TimeSerieAnalysis
- AnalyzeTool
- Scheduling
- StatisticTool
- GraphPlotter

De koppeling tussen J-Layer en J-Predictor zit tussen de entiteiten PredictiveScaler en Predictor waarbij de PredictiveScaler een nieuwe Predictor creëert voor het doen van voorspellingen.



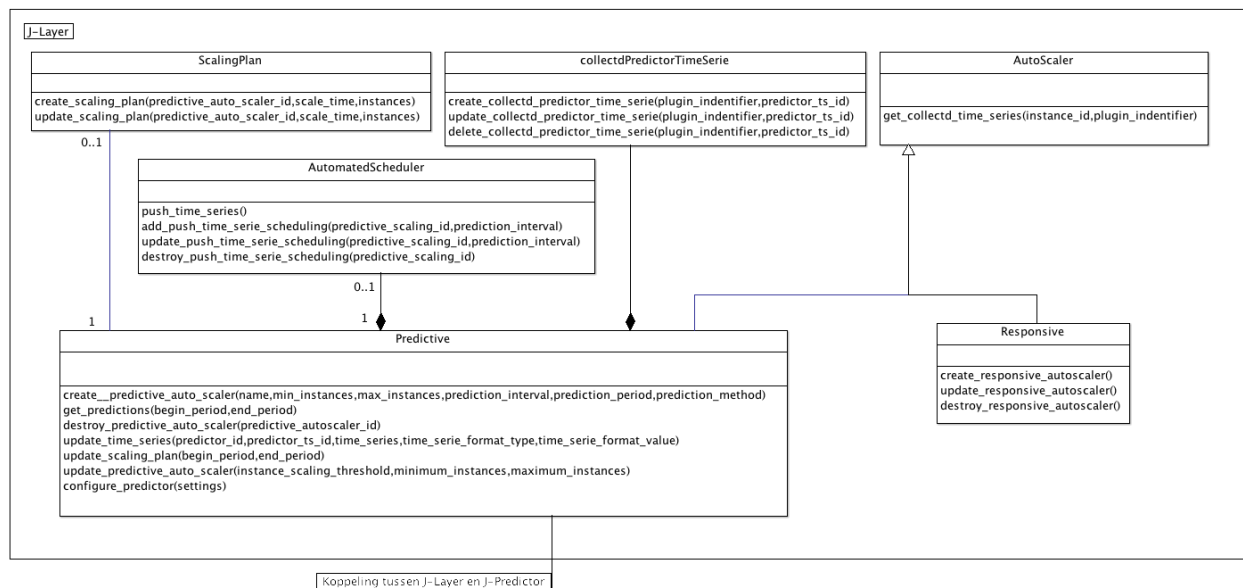
Afbeelding 3.1 - Domein model

4. Klassendiagram

Het klassendiagram (afbeelding 4.1 en afbeelding 4.2) is een uitgebreide uitwerking van hoe de J-Predictor eruit zal zien in combinatie met J-Layer. De entiteiten van het domein model zijn hierin vertaald naar klassen en er wordt beschreven welke methode en variabelen de klassen zullen bevatten. De communicatie tussen J-Layer en J-Predictor zal verlopen via Representational state transfer (REST) application programming interface (API), waarbij de data verstuurd worden in JavaScript Object Notation (JSON) formaat.

De voorspellingsmethodes en AnalyzeTool zijn beide gemoduleerd met het strategy design pattern, zodat nieuwe analysetools en voorspellingsmethodes toegevoegd kunnen worden zonder dat er in de klassen die hiervan gebruik maken, code gewijzigd hoeft te worden. Hiermee is het ook mogelijk om op runtime te bepalen welke statistische tools of voorspellingsmethode gebruikt moeten worden.

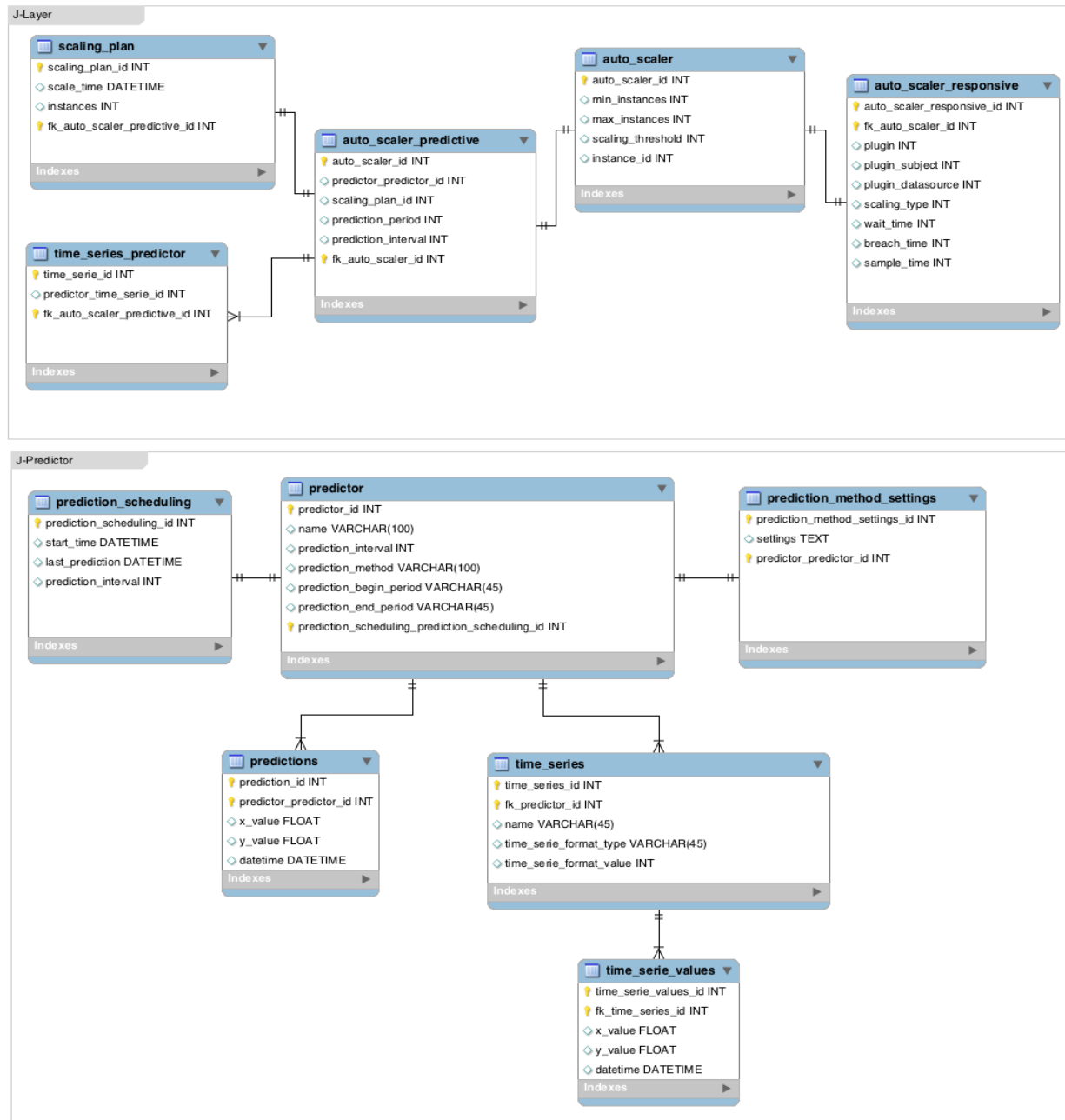
Binnen J-Predictor is de klasse Predictor de centrale klasse waaruit de andere klasse binnen J-Predictor aangeroepen kunnen worden. Bij J-Layer is de centrale klasse Auto-Scaler met de nadruk op PredictiveScaler aangezien er voorspeld moet worden hoeveel instances nodig zijn. De klasse ResponsiveScale is een klasse die al bestaat binnen J-Layer. Deze is ter verduidelijking aan het model toegevoegd maar de focus zal liggen bij de PredictiveScaler.



Afbeelding 4.1 – J-Layer klassendiagram

5. Data model

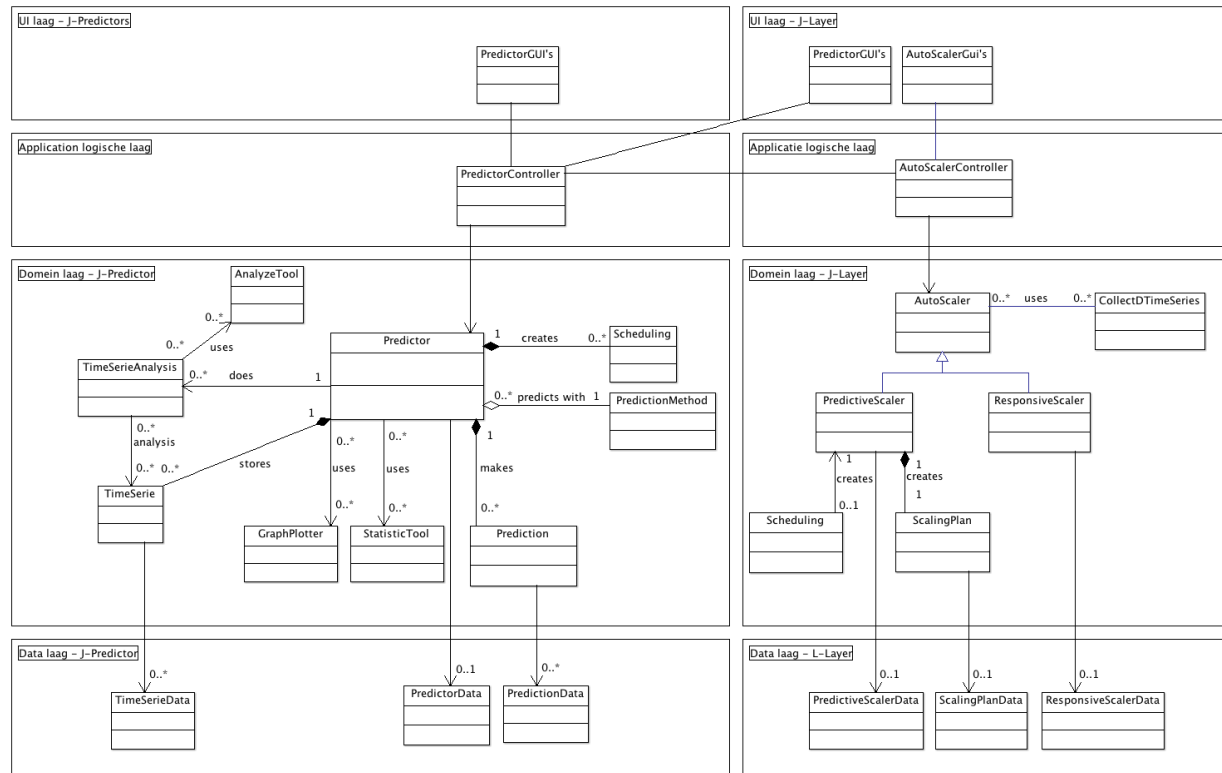
In het data model (afbeelding 5.1) zijn de database tabellen te zien van J-Layer en J-Predictor. Tussen de databases van J-Layer en J-Predictor zit geen koppeling aangezien de informatie gedeeld wordt via de REST API. De J-Layer database heeft kennis van J-Predictor doordat J-Predictor J-Layer voorziet van twee ID's, predictor_id en predictor_time_series_id. Met predictor_id is J-Layer in staat de juiste predictor aan te roepen voor het uitvoeren van opdrachten. En met predictor_time_series_id wordt de CollectD time serie gekoppeld met een predictor time serie. De koppeling tussen de time serie wordt gedaan aan de hand van J-Layer, omdat J-Layer de time serie verstuurt naar J-Predictor in plaats van dat J-Predictor de time serie opvraagt bij J-Layer. Op deze manier weet J-Layer welke time serie gestuurd moet worden.



Afbeelding 5.1 - Data model

6. Logische lagen model

Het logische lagen model (afbeelding 6.1) geeft een beeld hoe de communicatie loopt tussen verschillende lagen in de applicatie. De bovenste laag is de UI laag waar de GUI's zich bevinden die beschreven zijn in het functioneel ontwerp in de paragraaf "Schermontwerpen". J-Predictor bevat de GUI's voor het gebruik van de functies van de predictor. J-Layer heeft GUI's voor zowel de auto-scaler als de predictor. Met de auto-scaler GUI's worden de functies van de auto-scaler bediend en met de predictor GUI's kunnen net zoals bij J-Predictor de functies van de predictor bediend worden. De predictor GUI's bij J-Predictor en J-Layer doen in principe het zelfde waarbij de GUI's in J-Predictor direct met de predictor controller praten en de GUI's in J-Layer via de REST API met de predictor controller praten. Het is niet noodzakelijk dat J-Layer GUI's heeft voor de predictor maar hiermee is de gebruiker wel in staat om de predictor te bedienen van uit één applicatie, J-Layer.



Afbeelding 6.1 - Logische lagen model

7. Strategy pattern implementatie

Voor de voorspellingsmethodes en analysetools wordt de strategy pattern gebruikt, doordat er op runtime tussen gewisseld moet kunnen worden. De implementatie van de strategy pattern voor de voorspellingsmethodes is als volgt.

Elke voorspellingsmethode heeft een methode `get_settings`, `set_settings` en `predict`. Met `get_settings` wordt opgevraagd welke settings ingesteld moeten worden. De settings worden in een hashlist gegeven waarmee een formulier opgebouwd kan worden in de GUI. Een voorbeeld van een setting die de hashlist kan bevatten is: `setting naam, type, afspraak, input`. Dit zou er als volgt uit kunnen zien: `'minimum cluster size:', 'integer', '>= 0', ''`. In de GUI is het op deze manier mogelijk om de 'setting naam' weer te geven, het type invoerveld te bepalen aan het 'type' en aan te geven wat de verplichting is bij het invoeren van het veld. Na het invoeren wordt de hash teruggestuurd via de methode `set_settings` waarbij het 'input'

veld ingevuld is. In afbeelding 7.1 is te zien hoe dit er in de GUI uit ziet:

Minimum cluster size: >= 0

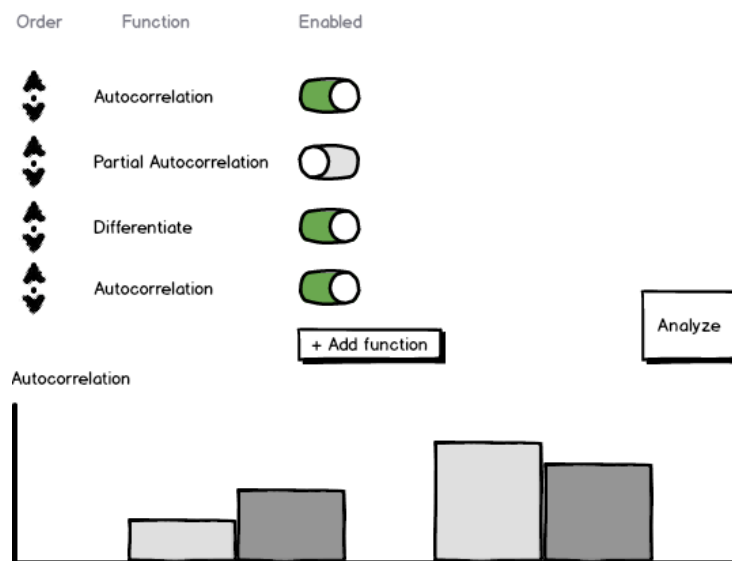
Afbeelding 7.1 - Minimum cluster size setting

De voorspellingsmethode controleert in `set_settings` de ingestelde waarden en stuurt een `HashList` terug met een error code en string om aan te geven of een setting goed of niet goed ingesteld is en wat er niet goed ingesteld is, bijvoorbeeld: 2, minimum cluster is < 0. Waarbij 2 de error code is en aangeeft dat er een fout is opgetreden. 1 zou dan kunnen betekenen dat er geen fouten waren.

Op het moment dat de settings juist ingesteld staan kan de methode `predict` aangeroepen worden voor het doen van een voorspelling, waarbij de time serie waarmee voorspeld moet worden meegegeven wordt. Als input en output van de methode geldt dat de time serie een hashlist moest zijn in de volgende structuur: `x_value => <integer>, y_value => <integer>`.

Voor het doen van analyses wordt ook de strategy pattern gebruikt. De reden hiervoor is dat de gebruiker moet kunnen bepalen welke analysetools er gebruikt worden voor het doen van een analyse op een time serie, zonder dat de gebruiker weet hoe de analysetools werken. En daarbij moet de gebruiker kunnen aangeven in welke volgorde de analyse tools uitgevoerd moeten worden.

In afbeelding 7.2 is een gedeelte van de 'Time Serie Anayzer' pagina te zien waar de gebruiker kan bepalen welke analysetools er gebruikt moeten worden voor de analyse.



Afbeelding 7.2 - Time Serie Anayzer

Het enige dat de analyse tools nodig hebben, is een time serie in het formaat van een ééndimensionale array en als output geeft de analyse tool een nieuwe ééndimensionale array van de time serie. De output kan vervolgens gebruikt worden in de volgende analyse tool, zodat de gebruiker met verschillende analyses het gewenste resultaat bereikt. Dit kan bijvoorbeeld nodig zijn als de autocorrelatie niet duidelijk laat zien wat de seasonality is. Om toch de seasonality inzichtelijk te krijgen is het nodig om de time serie te differentiëren voordat er autocorrelatie uitgevoerd wordt op de time serie.

8. Programmeertaal en framework

J-Layer is geprogrammeerd in Ruby met als framework Rails (Ruby on Rails) dus het uitbreiden van J-Layer met voorspellend auto-scalen zal ook ontwikkeld worden in Ruby.

J-Predictor zal net zoals J-Layer ontwikkeld worden in Ruby met als framework Rails. Ruby on Rails (RoR) is gebaseerd op de principes van model, view, controller (MVC) en RESTful design. Door MVC scheidt RoR duidelijk de UI, applicatie en domein laag en doordat RoR gebruik maakt van activerecords wordt ook de data laag gescheiden van de domein laag. Daarbij maar RoR ook gebruik van een RESTful design waardoor met RoR REST gebruikt kan worden voor de API.

9. Database

De database die bij J-Layer gebruikt wordt is MySQL dus voor de uitbreiding van J-Layer met voorspellend auto-scalen zal de zelfde database gebruikt worden. Als database voor J-Predictor wordt ook MySQL gebruikt met als storage engine MyISAM. MySQL is goed schaalbaar, heeft een goed performance bij het uitvoeren van query's en is gemakkelijk te beheren.

10. RESTful

De communicatie tussen J-Layer en J-Predictor zal gedaan worden met REST. Hiermee is het mogelijk om met een externe applicatie functies aan te roepen via URL's die gebruikt worden voor het aanroepen van GUI's. Hierdoor is geen extra method, zoals een façade klasse, nodig die zorgt voor de koppeling tussen J-Predictor en externe applicatie. Voor het aanmaken van een nieuwe predictor is dit voor zowel de GUI als REST: j-predictor.jitscale.com/predictor/new.

11. JSON

Voor uitwisselen van data tussen J-Layer en J-Predictor wordt JSON gebruikt. JSON is leesbaar voor mensen en het is gemakkelijk te importeren omdat de data worden opgeslagen als arrays en records. In vergelijking met XML is het data bestand dat verstuurd moet worden kleiner waardoor er minder bandbreedte nodig is en bij het exporteren en importeren minder CPU en geheugen verbruikt wordt. Hierdoor is het gebruik van JSON ook sneller dan XML.

12. Htaccess security

De beveiliging van J-Predictor zal geregeld worden met htaccess en SSL waarbij J-Predictor afgeschermd wordt op basis van IP en gebruikersnaam en wachtwoord. Hierdoor kan een gebruiker of client alleen J-Predictor bereiken vanaf een bepaald IP en met de juiste gebruikersnaam en wachtwoord.

13. Webserver

Als webserver zal Apache gebruikt gaan worden. Apache biedt de mogelijkheid om Htaccess security te gebruiken en biedt een ruim aanbod aan functionaliteiten, waaronder caching, loadbalancing en proxying. Daarbij heeft Apache een ruim aanbod aan documentatie en is het een betrouwbare webserver omdat het veel gebruikt wordt en het al bestaat sinds 1995. Daardoor zijn de meeste bugs eruit of worden snel gevonden.

14. Clustered linear regression

Clustered linear regression (CLR) zal geïmplementeerd worden binnen de strategy pattern Prediction_method. De parameters voor CLR worden ingesteld door de Predictor klasse en worden met de HashList settings verstuurd naar de Prediction_method Clustered_linear_regression welke de HashList omzet naar de parameter variabelen. En voor het doen van een accurate voorspelling maakt CLR gebruik van de methodes K_fold_cross_validation, root_mean_square_error en outlier_detection, die zich bevinden in de klasse Statistic_tools.

Implementatieplan

J-Layer auto-scaling

Inhoudsopgave

1.	Inleiding	1
2.	J-Predictor IT omgeving	1
2.1.	Servers setup	1
2.2.	Back-up	1
2.3.	Monitoring.....	1
2.4.	Software	2
2.5.	IT infrastructuur model	3
3.	Werken met J-Predictor	4
3.1.	Overdracht.....	4

1. Inleiding

In dit implementatie plan wordt beschreven worden hoe J-Layer geïmplementeerd zal worden binnen Jitscale. Er wordt besproken hoe de server setup is, wat de benodigde software is en hoe het beheer overgedragen zal worden naar de support- en beheerafdeling.

J-Layer is al geïmplementeerd binnen Jitscale dus voor de implementatie zal de focus vooral liggen bij J-Predictor.

2. J-Predictor IT omgeving

Voor de werking van J-Predictor is een IT omgeving nodig die bestaat uit een database, applicatieserver(s) en een oplossing om data op te slaan die gedeeld wordt over de verschillende applicatie servers. Daarbij is het ook van belang, dat de data geback-upt wordt, zodat data verlies voorkomen wordt en de IT omgeving gemonitord wordt.

2.1. Servers setup

Voor de server setup is gekozen om die als volgt uit te voeren:

Database	Amazon RDS met MySQL
Applicatieserver	Amazon m3.large instance
Load balancing	Amazon ELB
Storage	Amazon S3

2.2. Back-up

Doordat gebruik gemaakt wordt van Amazon RDS en S3 zullen RDS en S3 ook de back-up regelen.

Database	Amazon RDS automatische back-ups en snapshots.
Applicatieserver	Opslag van data vindt plaats in S3 dus de back-up zal gedaan worden door Amazon S3.

2.3. Monitoring

De monitoring zal gedaan worden met CollectD. CollectD komt te draaien op elke applicatieserver. Het volgende zal gemonitord worden:

voor J-Predictor

- load van applicatie server.
- geheugen van applicatie server.
- schijfruimte op applicatie server.
- verbinding tussen applicatieserver en database.
- verbinding tussen applicatieserver en S3.
- verbinding tussen applicatieserver en andere applicatieservers.
- beschikbaarheid van J-Predictor

voor J-Layer:

J-Layer beschikt al over monitoring dus hieronder staan alleen de monitoring checks die extra toegevoegd moeten worden voor J-Layer.

- verbinding tussen J-Layer en J-Predictor

De monitoring checks moeten aan de volgende eisen voldoen:

Check	Warning	Critical
Load	> 75% van CPU's	> 85% van CPU's
Geheugen	> 75% totale geheugen	> 85% totale geheugen
Schijfruimte	> 75% totale schijfruimte	> 85% totale schijfruimte
Verbinding tussen applicatieserver en database	> 10 seconden	> 15 seconden
Verbinding tussen applicatieserver en S3	> 10 seconden	> 15 seconden
Verbinding tussen applicatieserver en andere applicatieservers	> 10 seconden	> 15 seconden
Beschikbaarheid van J-Predictor	Laden van J-Predictor > 10 seconden	Laden van J-Predictor > 15 seconden
Verbinding tussen J-Layer en J-Predictor	> 10 seconden	> 15 seconden

2.4. Software

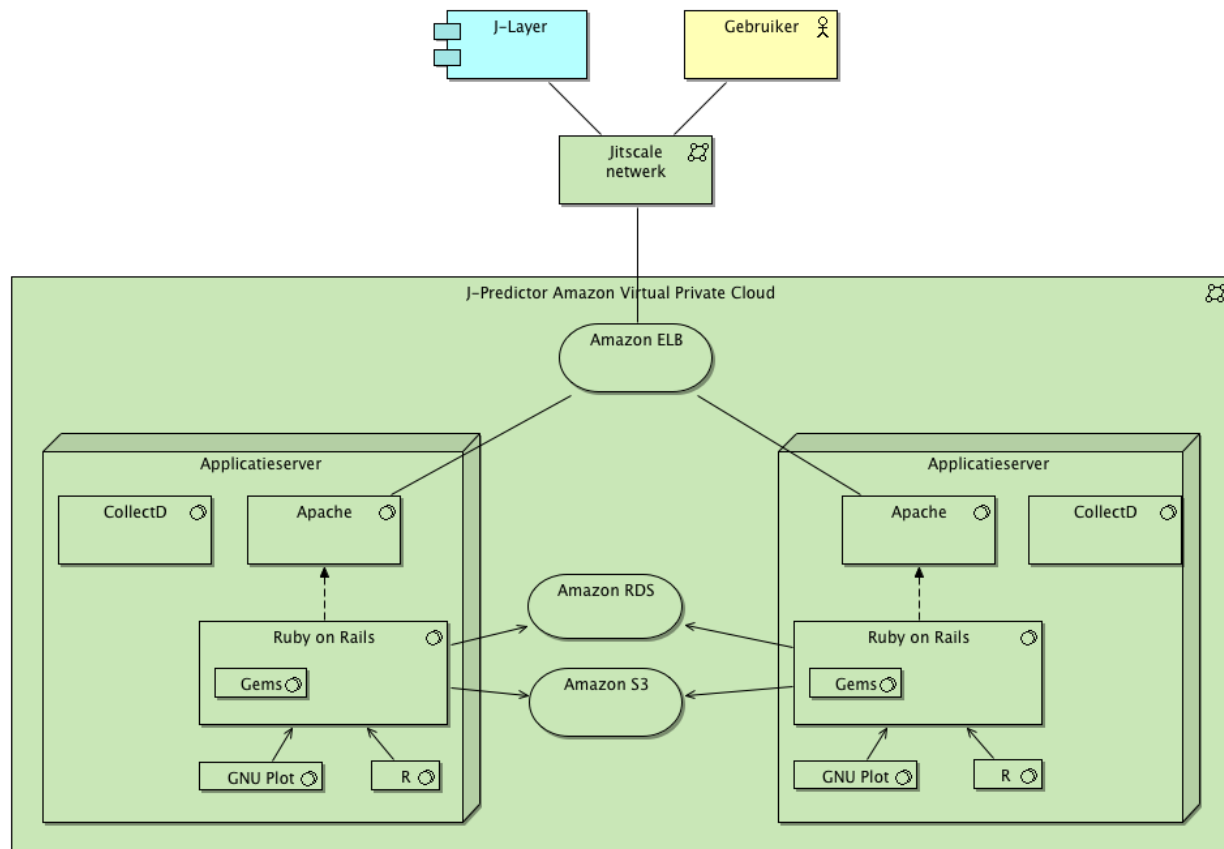
Op het gebied van software heeft J-Predictor een aantal benodigdheden die hieronder besproken worden. Daarbij wordt ook aangegeven met welke versie van de software J-Predictor in ieder geval werkt.

Applicatieserver besturingssysteem	Debian Wheezy	versie 7.0 (Wheezy)
Monitoring	CollectD	versie 5.1.0
Webserver	Apache	versie 2.2.22
Database	MySQL	versie 5.5
Framework	Ruby on Rails	versie 3.2.16
Programmeertaal	Ruby	versie 1.9.3
Gems	haml	versie 4.0.5

	gnuplot	versie 2.6.2
	mysql	versie 2.9.1
	rserve-client	versie 0.2.5
	linefit	versie 0.3.2
R	r-base	versie 2.15
Rserve	r-cran-rserve	versie 0.6
R libraries	r-cran-tseries	versie 0.10
GNU Plot	gnuplot	versie 4.6

2.5. IT infrastructuur model

Het IT infrastructuur model in afbeelding 2.1 toont hoe de server setup in combinatie met de software gaat werken. Daarbij is te zien hoe J-Layer via het Jitscale netwerk verbinding moeten maken met J-Predictor en hoe de Amazon ELB de requests verdeelt over twee applicatieservers. Waarbij de twee applicatieservers gebruikmaken van de zelfde database (RDS) en storage (S3).



Afbeelding 2.1 - IT infrastructuur model

3. Werken met J-Predictor

Hoe er gewerkt moet worden met J-Predictor wordt besproken in een overdracht sessie. Tijdens de overdracht wordt de werking van functies en de inrichting van de omgeving van J-Predictor besproken. Daarbij is in de knowledge base van Jitscale een uitgebreide documentatie beschikbaar van de werking en omgeving van J-Predictor.

3.1. Overdracht

De overdracht van J-Predictor zal gedaan worden voor twee doelgroepen:

- support afdeling
- beheer afdeling

Voor elke afdeling is het van belang dat bekend is hoe J-Predictor werkt en hoe de omgeving ingericht is. Voor de overdracht zal een sessie gepland worden met een presentatie en een demonstratie van J-Predictor. Daarbij wordt ook uitgelegd hoe J-Layer de voorspelling gebruikt voor het bepalen van het aantal instances.

Voorspellingsmethodes

J-Layer auto-scaling

Inhoud

1. Inleiding	1
2. Box-Jenkins	1
3. Clustered linear regression	5
3.1. Seasonality	7
3.2. Clusteren	7
3.3. Outliers	7
3.4. Under- en overfitting	8
4. Exponential smoothing.....	10
5. Artificial neural networks	12
6. Support vector machine	16
7. Bibliografie	20

1. Inleiding

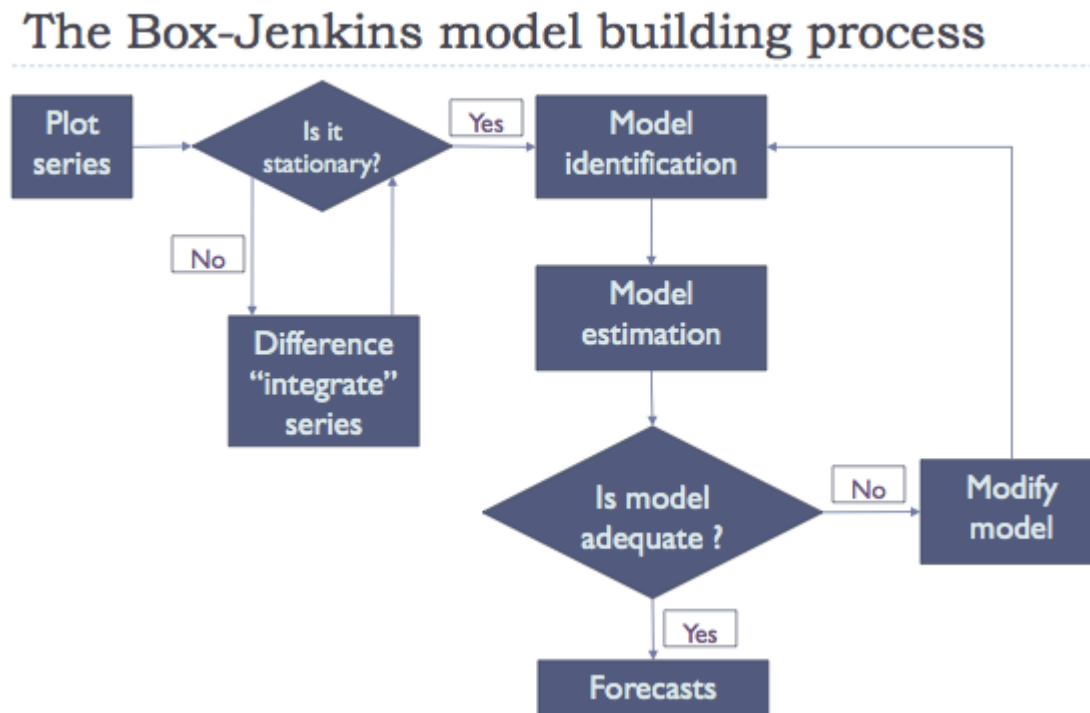
Voor het doen van een voorspelling op basis van time series zijn er een aantal voorspellingsmethodes waarmee dat mogelijk is. Elke voorspellingsmethode heeft zijn voor- en nadelen en eigen situaties waardoor de methode beter of slechter zal presteren. Hieronder zal duidelijk gemaakt worden welke voorspellingsmethodes gebruikt kunnen worden voor het doen van time serie voorspellingen, zodat duidelijk wordt hoe ze werken en wat de verschillen zijn.

2. Box-Jenkins

Box-Jenkins is een voorspellingsmethode die gebruikt maakt van ARIMA modellen. ARIMA staat voor "AutoRegressive Integrated Moving Average" en wordt ook wel ARIMA(p,d,q) genoemd. ARIMA modellen zijn modellen waarmee time serie data geanalyseerd kan worden en waarmee voorspellingen gedaan kunnen worden. (Wikipedia, Autoregressive integrated moving average, 2014)

Bij het doen van een voorspelling wordt een proces doorlopen waarbij de time serie data geanalyseerd wordt waarmee bepaald kan worden welk ARIMA model het beste gebruikt kan worden voor het doen van de voorspelling. Dit is nodig omdat het ene ARIMA model een betere voorspelling geeft bij een bepaalde time serie dan een ander ARIMA model.

Het proces dat doorlopen wordt bij Box-Jenkins is te zien in afbeelding 2.1.



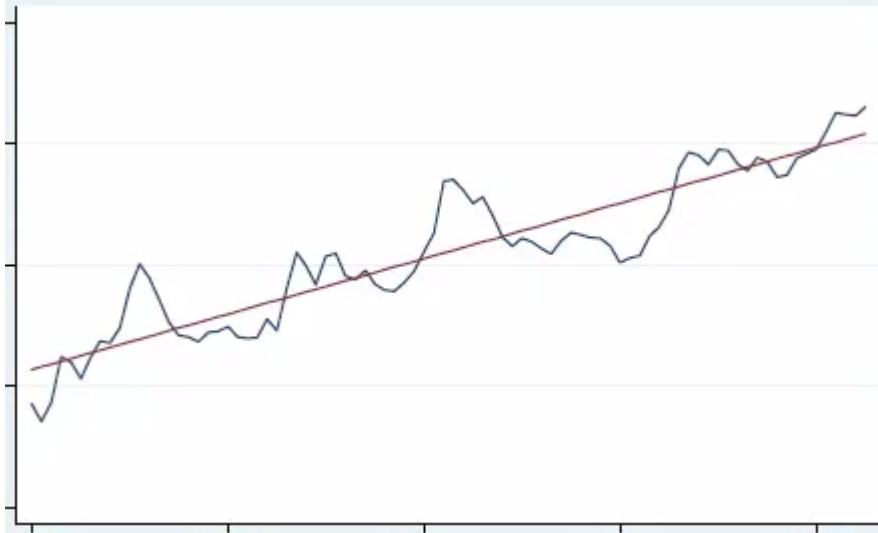
Afbeelding 2.1 – Box-Jenkins proces

Stap 1:

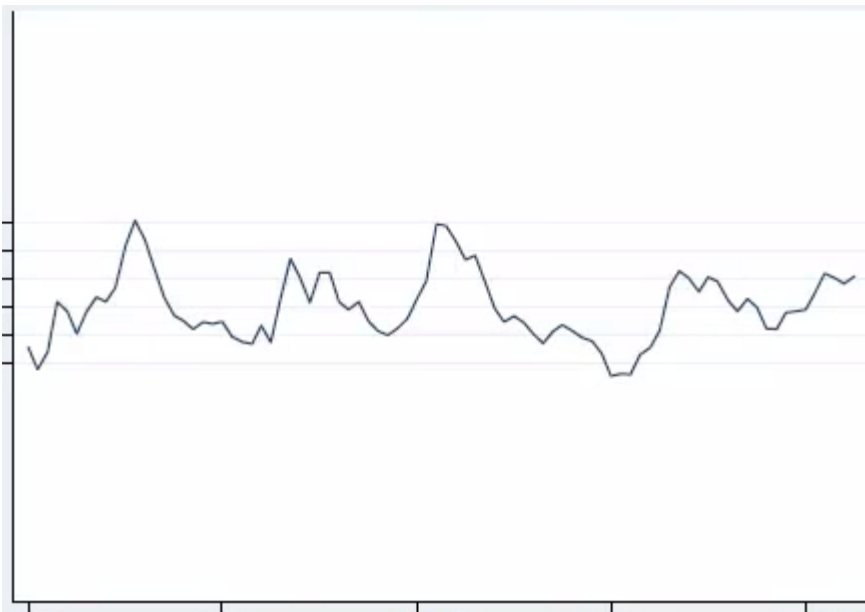
Hier wordt bepaald of de time serie stationair is of niet. Het is nodig dat de time serie stationair is omdat het met een stationaire time serie mogelijk is te bepalen of de time serie seasonality bevat wat in stap 3 bepaald wordt. Als de time serie niet stationair is zal stap 2 uitgevoerd moeten worden voor het stationair maken van de time serie. (Wikipedia, Stationary process, 2014), (Duke, z.d), (NIST, Seasonality, z.d)

Stap 2:

Als de time serie niet stationair is, zal deze stationair gemaakt moeten worden. Dit is te doen door de time serie te differentiëren. In afbeelding 2.2 is een time serie te zien die niet stationair is en die gedifferentieerd moet worden en in afbeelding 2.3 is een de zelfde time serie te zien na het differentiëren. Zoals te zien is, heeft de time serie geen stijgende trend meer, maar heeft nog wel de zelfde pieken en dalen welke belangrijk zijn voor stap 3.



Afbeelding 2.2 – Niet stationaire time serie

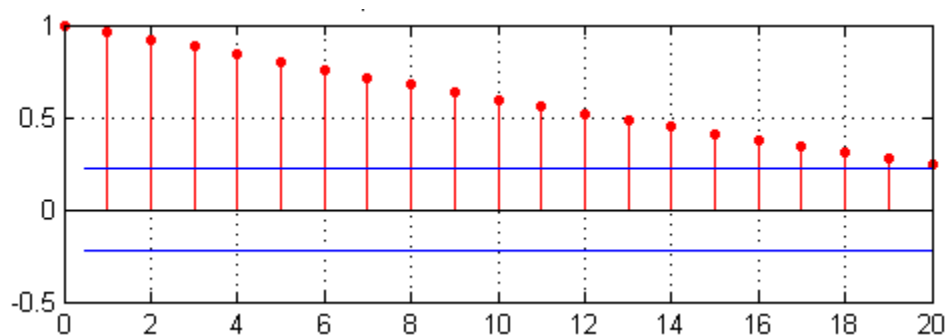


Afbeelding 2.3 - Time serie na differentiëren

Op het moment dat het nodig is de time serie 1 keer te differentiëren betekent dit dat I uit ARIMA 1 is. Dit kan op twee manieren geschreven worden als $I(1)$ of $ARIMA(0,1,0)$. Hierbij staat de eerste 0 voor $AR(0)$ en de tweede 0 voor $MA(0)$. De waarden voor AR en MA zullen in de stappen hierna bepaald worden.

Het bepalen of een time serie niet lineair is kan gedaan worden met de Dickey–Fuller test of door het uitvoeren van een autocorrelation functie op de time serie. Als het resultaat geplot wordt en er is een lineaire daling te zien dan betekent dit dat de time serie niet lineair is.

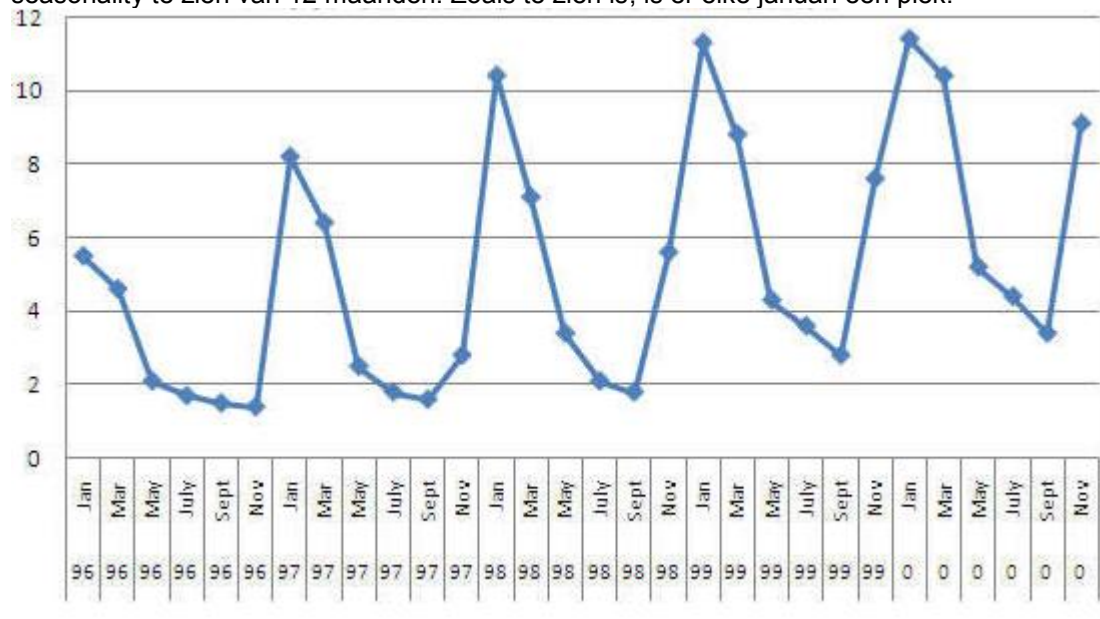
In afbeelding 2.4 is te de autocorrerlation te zien van een time serie die niet stationair is. (Eurostat, Trend, z.d), (Wikipedia, Autocorrelation, 2014), (Wikipedia, Dickey–Fuller test, 2014), (NIST, Box-Jenkins Model Identification, z.d)



Afbeelding 2.4 - Autocorrelation functie van een niet stationaire time serie

Stap 3:

Op het moment dat de time serie stationair is kan stap 3 uitgevoerd worden waar uitgezocht wordt welke seasonalities de time serie heeft. De seasonalities zijn nodig, want dit is één van de criteria waarmee bepaald wordt welk ARIMA model gebruikt moet worden. In afbeelding 2.5 is als voorbeeld een seasonality te zien van 12 maanden. Zoals te zien is, is er elke januari een piek.



Afbeelding 2.5 - Jaarlijkse seasonality

Bij een seasonality van 12 maanden bij een time serie waarbij 1 lag een maand is, zal dit betekenen dat dit een AR 12 en/of MA 12 model is. Dit wordt ook wel genoteerd als $ARIMA(p,d,q)_{12}$, waarbij de seasonality achteraan buiten de haken staat. (Eurostat, Lag, z.d), (Eurostat, Time lag, z.d)

Stap 4

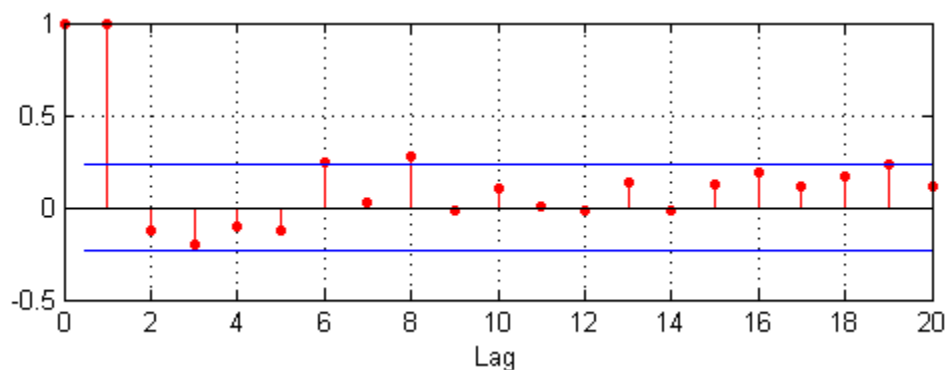
Hier zal bepaald worden of het model AR(autoregressive) en/of MA(moving average) order. Het bepalen of in het model AR en/of MA gebruikt moet worden en met welke order is te doen door het plotten van de autocorrelation functie (ACF) en partial autocorrelation(PACF) functie van de time serie.

De volgende tabel kan hierbij gebruikt worden voor het bepalen het model en de order:

Model	ACF	PACF
AR(p)	Geleidelijke daling	Snelle daling na p lags
MA(q)	Snelle daling na q lags	Geleidelijke daling
ARMA(p, q)	Geleidelijke daling	Geleidelijke daling

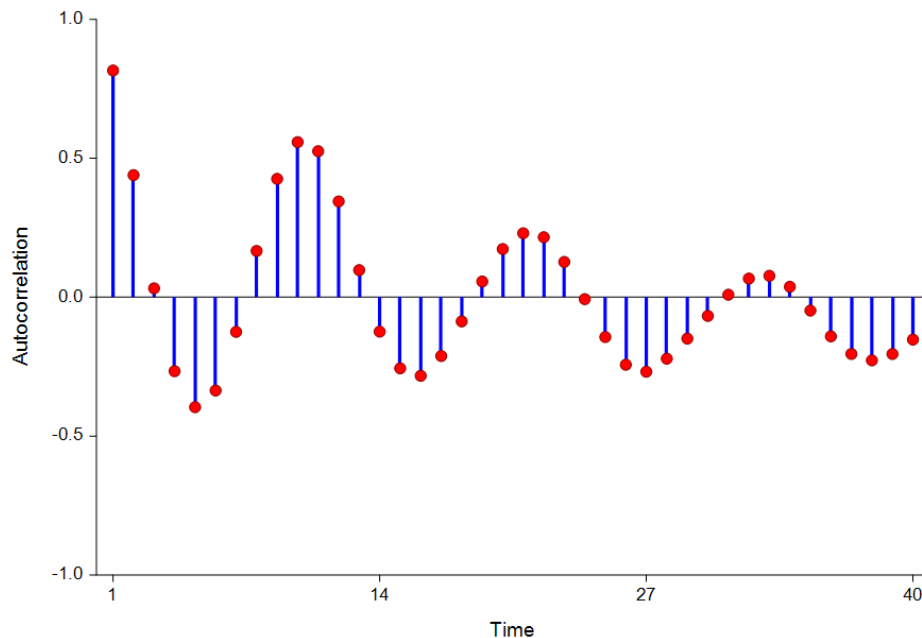
In afbeelding 2.4 is een ACF plot te zien die geleidelijk daalt, wat aanduidt dat het om een AR of ARMA model gaat. Als de zelfde time serie geplotted wordt in een PACF functie en er is een snelle daling na een p aantal lags (afbeelding 2.6) betekent dit dat het om een AR model gaat. In afbeelding 2.6 is te zien dat er een snelle daling is na lag 2 wat betekent dat de AR order 2 is. Dit wordt genoteerd als AR(2) of ARIMA(2, d , q).

Voor een MA model zal dit omgekeerd zijn, waarbij afbeelding 2.6 de ACF functie is van de time serie en afbeelding 2.4 de PACF functie.



Afbeelding 2.6 - Partial autocorrelation functie

Het is ook mogelijk dat de time serie een mix is van AR en MA, ARMA(p, q). Dit is het geval als de ACF een exponentiele daling en een gedempt sinusvormige patroon vertoont, zie hiervoor afbeelding 2.7. Het bepalen van de order p en q voor het AR en MA model is in dit geval erg lastig, en het is daarom verstandig om een aantal verschillende order combinaties te proberen en de gene te kiezen waarbij er geen grote autocorrelatie is. (MathWorks, z.d), (Data, z.d), (NCSS, z.d)

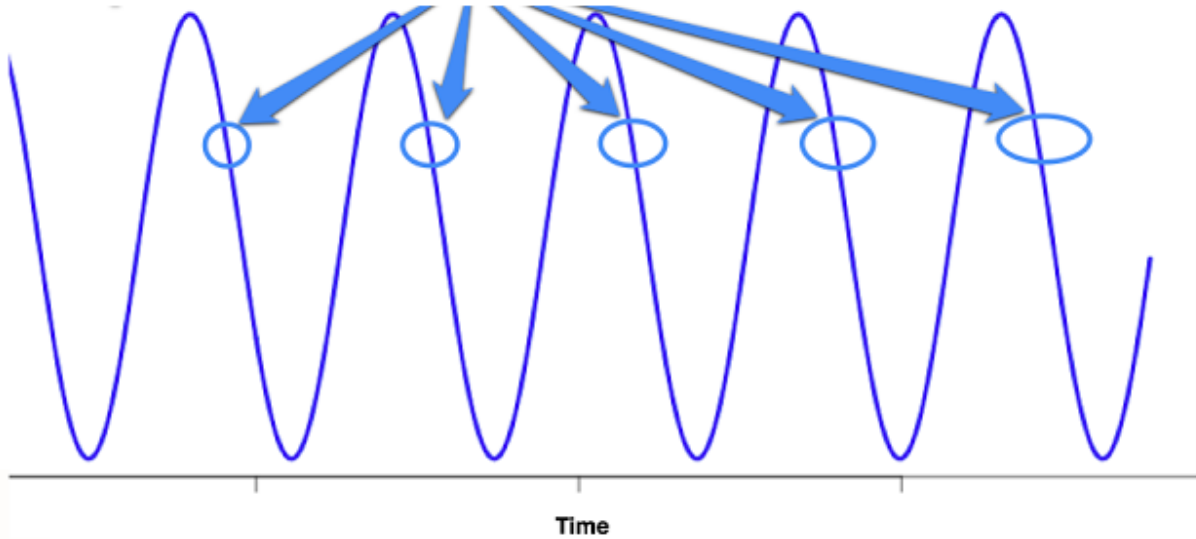


Afbeelding 2.7 - Sinusvormige autocorrelation

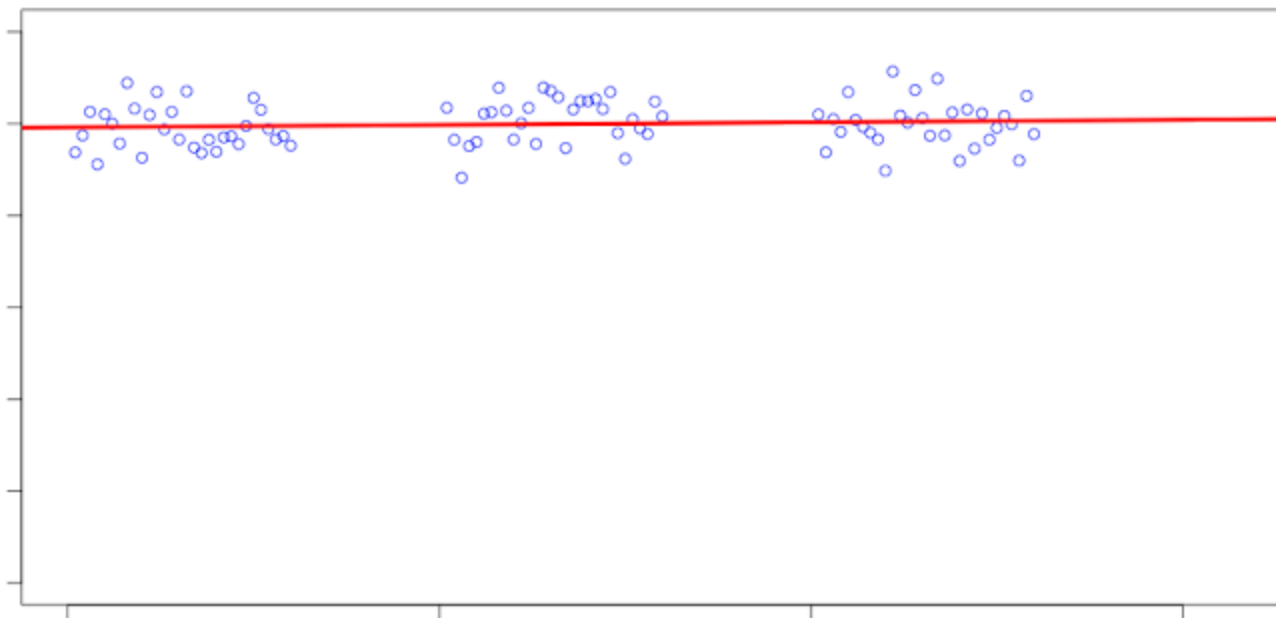
Als zowel AR als MA een order van 2 hebben, er 1 keer gedifferentieerd is en er een seasonality van 12 is, geeft dit het volgende ARIMA model: $ARIMA(2,1,2)_{12}$. Na het identificeren van het ARIMA model zal deze getest moeten worden om te controleren of de voorspelling juist is. Als de voorspelling niet klopt zal vanaf stap 3 weer gekeken moeten worden welke model het beste gebruikt kan worden. (Investopedia, Autoregressive, z.d), (Investopedia, Moving Average - MA, z.d), (Wikipedia, Partial autocorrelation function, 2014)

3. Clustered linear regression

Clustered linear regression (CLR) is een manier van lineaire regressie waarbij de lineaire regressie op clusters van datapunten wordt uitgevoerd. Een voorbeeld waarbij CLR gedaan wordt is te zien in afbeelding 3.1 en afbeelding 3.2. Met de cirkels in afbeelding 3.1 wordt aangegeven op welke punten in de time serie datapunten worden geclusterd. Als een time serie een seasonality van een dag heeft en je wilt weten wat de waarden zijn van de datapunten vandaag tussen 19:00 uur en 19:15 uur zal CLR datapunten van de afgelopen dagen (als er een dagelijkse seasonality is), tussen 19:00 en 19:15 uur, clusteren zoals te zien is in afbeelding 3.1. Na het clusteren voert CLR lineaire regressie uit op de geclusterde data zoals te zien is in afbeelding 3.2, waarbij de rode lijn de regressielijn is. Door het doen van lineaire regressie wordt het mogelijk om met extrapolatie te voorspellen wat de waarde zal zijn van de datapunten de volgende dag tussen 19:00 en 19:15 uur. (Wikipedia, Linear regression, 2014), (Wikipedia, Extrapolatie, 2013), (Yuan, Joshi, Jacobson, & Oberai, 2014)



Afbeelding 3.1 - Clusters in een time serie



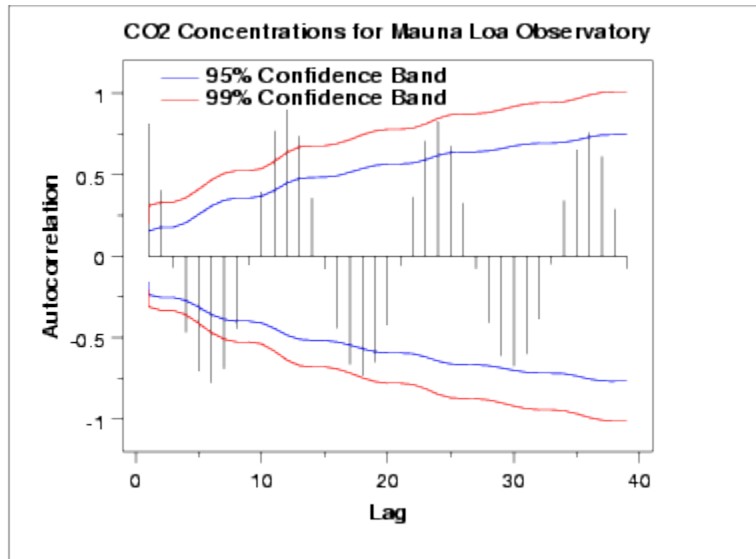
Afbeelding 3.2 - Lineaire regressie op geclusterde data

Voor het doen van CLR is het van belang dat de volgende stappen doorlopen worden:

1. Het bepalen van de seasonality.
2. Het bepalen van het cluster moment en de cluster grootte.
3. Het detecteren en negeren van outliers.
4. Het controleren of de voorspelling nauwkeurig is en niet lijdt aan over- of underfitting.

3.1. Seasonality

Wat erg belangrijk is bij CLR, is dat de time serie seasonality heeft en dat deze bekend is. Zonder seasonality kan er geen accurate voorspelling gedaan worden met CLR, omdat CLR afhankelijk is van een herhalend patroon voor het bepalen van de clusters. Het bepalen van de seasonality kan gedaan worden door gebruik te maken van de autocorrelation functie. Als time serie een seasonality heeft is dit te zien aan de lags die boven of onder de rest uitschieten, zie afbeelding 3.3.



Afbeelding 3.3 - Autocorrelation met een seasonality van 12 lags

Net zoals bij Box-Jenkins moet de time serie stationair zijn voordat de seasonality gevonden kan worden.

3.2. Clusteren

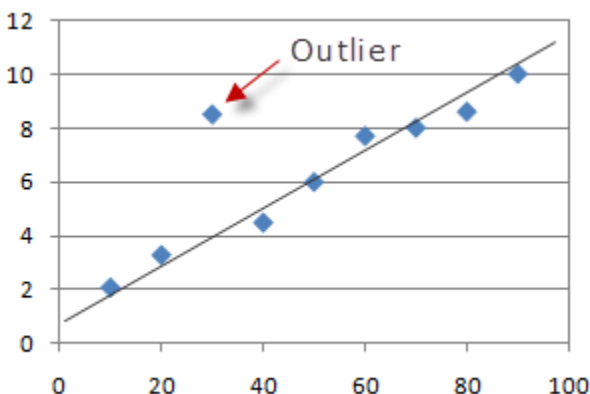
Voor het bepalen van de cluster zijn een aantal dingen belangrijk. Er moet als eerste bepaald worden welke periode de clusters omvatten, bijvoorbeeld van 19:00 uur tot 19:15 uur. Bij een dagelijkse seasonality wordt in dit geval elke dag de datapunten tussen 19:00 uur en 19:15 uur geclusterd, bij een wekelijkse seasonality zal dit bijvoorbeeld elke zaterdag tussen 19:00 uur en 19:15 uur kunnen zijn.

Daarnaast moet bepaald worden welke clusters belangrijker zijn voor de voorspelling. Dit is omdat het mogelijk is dat de time serie recent hogere pieken of dalen heeft (verandering in trend) waardoor de lineaire regressie hoger of lager uit moet gaan komen. Als een cluster belangrijker is zal dit cluster meer gewicht moeten krijgen bij de lineaire regressie. Meer gewicht geven kan gedaan worden door het cluster meer datapunten te geven dan de andere clusters.

3.3. Outliers

Het kan voorkomen dat de time serie niet de juiste data bevat voor het doen van de juiste voorspelling. Zo kan het zijn dat er een ongewoon hoge piek in zit, of dat er data ontbreken in een bepaalde periode. Zulke onregelmatigheden kunnen de voorspelling negatief beïnvloeden. Om dit te voorkomen moet de data geanalyseerd worden op outliers (uitschieters). Met outliers wordt data bedoeld die boven of onder

de rest uitschieten. Een mooi voorbeeld van een outlier is te zien in afbeelding 3.4.

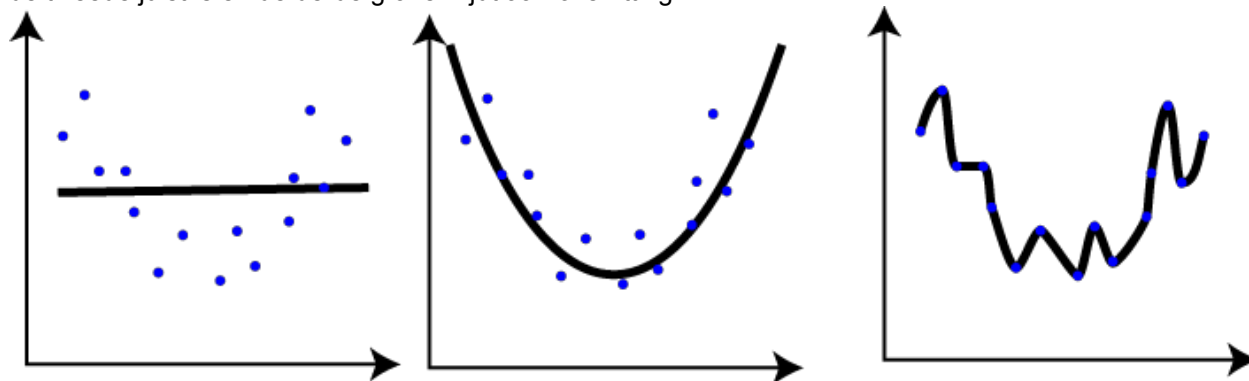


Afbeelding 3.4 - Outlier

Op het moment dat een outlier gevonden is kan ervoor gekozen worden om de outlier niet of wel mee te nemen in de voorspelling. Voordat de outliers genegeerd worden in de voorspelling is het verstandig om goed te controleren of de outlier geen relevante informatie bevat, want het zou kunnen zijn dat de outliers een nieuwe trend of seasonality aantonen die juist wel in de voorspelling opgenomen moet worden. (Wikipedia, Outlier, 2014), (Kriege, Kröger, & Zimek, 2010), (Reference, z.d)

3.4. Under- en overfitting

Als de data klaar zijn om gebruikt te worden in de voorspelling en de cluster grootte en seasonality bepaald zijn, kan de voorspelling gedaan worden. Na het doen van de voorspelling met nieuwe instellingen/voorspellingsmodel is het van groot belang dat de voorspelling gecontroleerd wordt op juistheid. Een probleem dat zich voor kan doen is dat de voorspelling lijdt aan under- of overfitting. In afbeelding 3.5 zijn drie grafieken te zien waarbij in de eerste grafiek de voorspelling lijdt aan underfitting, de tweede juist is en de derde grafiek lijdt aan overfitting

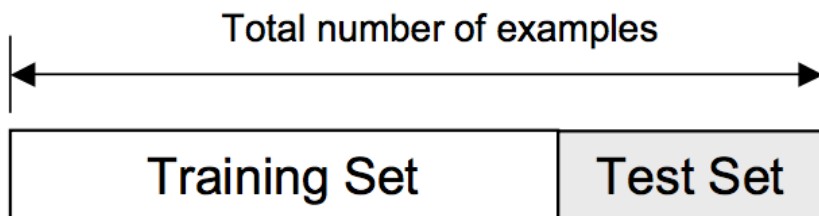


Afbeelding 3.5 - Underfitting en overfitting

Op het moment dat er underfitting plaats vindt dan is de bias hoog, wat betekent dat het model niet de mogelijkheid heeft trends te vinden in de time serie data. Als overfitting plaats vindt dan is de variance hoog wat betekent dat het model veel of alle trends vindt. (data, 2013), (AntiAnti, 2011), (Fortmann-Roe, 2012), (Ye & PhD, 2010)

Er zal een tussenweg gevonden moeten worden tussen de bias en variance, zodat under- en overfitting voorkomen worden. Dit kan gedaan worden door de voorspelling te valideren aan de hand van een test dataset. Deze methode wordt ook wel de 'holdout method' genoemd waarbij de time serie opgesplitst wordt in een trainingsset en een testset. De voorspellingsmethode gebruikt de trainingsset voor het doen van de voorspelling en valideert de voorspelling aan de hand van de testset. In afbeelding 3.6 is te zien

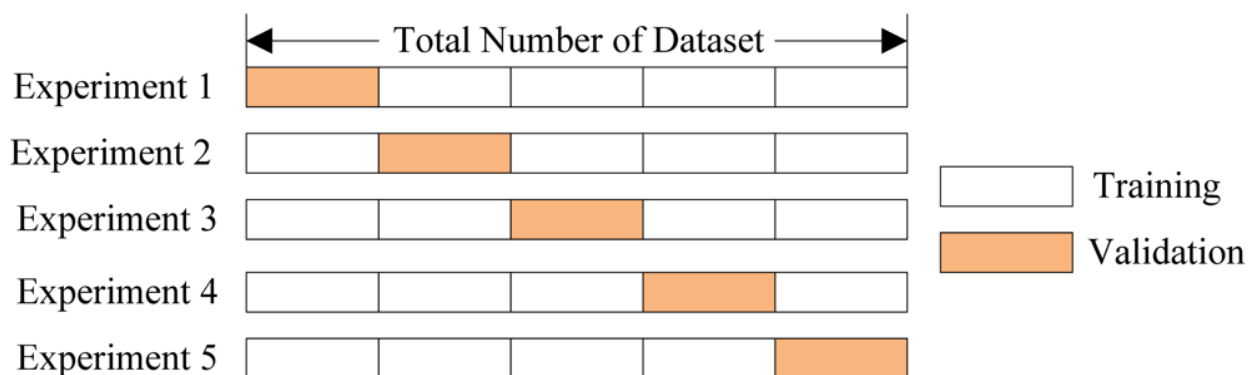
hoe de time serie opgesplitst wordt. Het valideren van de voorspelling aan de hand van testdata waar niet mee getraind is, is belangrijk omdat hiermee gecontroleerd wordt of een goede voorspelling gedaan kan worden in de toekomst op data die nog niet bekend zijn. Door deze manier van valideren kan overfitting gedetecteerd worden omdat bij overfitting het model voor de voorspellingsmethode precies gebaseerd is op de trainingsset in plaats van het onderliggende patroon. Hierdoor zal bij overfitting een voorspelling gedaan worden die precies lijkt op de trainingsset in plaats van een voorspelling die onderliggende patronen zoals een opwaartse trend meeneemt in de voorspelling.



Afbeelding 3.6 - Holdout method

Het valideren met enkel één trainingsset en één test/validatieset heeft helaas wel een nadeel. Dat komt omdat er een gedeelte van de data niet gebruikt kan worden voor het doen van een voorspelling omdat het gebruikt wordt voor het valideren van de voorspelling. Op het moment dat de time serie al niet veel data bevat kan dit betekenen dat er belangrijke data niet gebruikt wordt voor de voorspelling. Dit probleem wordt opgelost door (K-fold) cross validation. Met K-fold cross validation wordt het mogelijk om alle data te gebruiken voor het doen van een voorspelling waardoor er geen relevante data gemist wordt.

K-fold cross validation gebruikt de zelfde manier van valideren als bij de holdout methode maar in plaats van alleen te trainen en te testen op één training of test set aan data worden alle data gebruikt voor training en test. In afbeelding 3.7 wordt een beeld gegeven hoe K-fold cross validation werkt. (Zhang & Wu, 2012)



Afbeelding 3.7 - K-fold cross validation

Zoals in afbeelding 3.7 te zien is wordt de time serie in 5 blokken opgedeeld, wat betekend dat dit 5-fold cross validation is (K staat in dit geval voor 5). Vervolgens wordt de data meerdere keren getraind en getest totdat alle data een keer getraind zijn en één keer getest zijn. Met 5-fold cross validation wordt de time serie data 5 keer doorlopen. Door op deze manier de voorspelling te testen wordt er geen data onbruikbaar gemaakt doordat het alleen als test data gebruikt kan worden. (Gutierrez-Osuna, z.d)

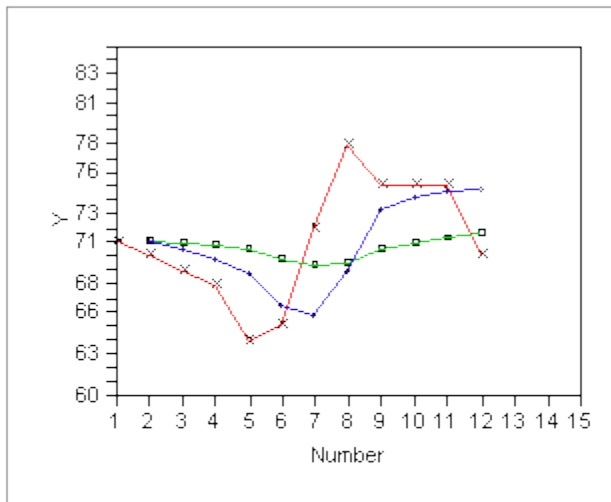
4. Exponential smoothing

Exponential smoothing is een voorspellingsmethode welke bestaat uit een aantal verschillende methodes:

1. Simple exponential smoothing (SES)
2. Double exponential smoothing (DES)
3. Triple exponential smoothing (TES)

Het verschil tussen de drie methodes is, dat bij simple exponential smoothing een voorspelling gedaan wordt op basis van vorige voorspellingen waarbij aangegeven kan worden hoe belangrijk een waarde uit het verleden is voor een voorspelling in de toekomst. Bij SES wordt gebruikt gemaakt van weighted moving average waarbij een waarde uit het verleden exponentieel minder belangrijk wordt voor het doen van een voorspelling. Hoe belangrijk (gewicht) een waarde uit het verleden is, kan aangegeven worden met een getal tussen 0 en 1. Hoe lager het gewicht hoe vloeiender de lijn. In afbeelding 4.1 zijn twee voorspellingen te zien, waarbij de groene lijn een lager gewicht (alpha) heeft en meer vloeiend is en de blauwe lijn meer gewicht heeft en minder vloeiend is.

Exponential Smoothing: Original and Smoothed Values



Y x- Original Y ■ alpha = .1 ◆ alpha = .5

Afbeelding 4.1 - Exponential smoothing

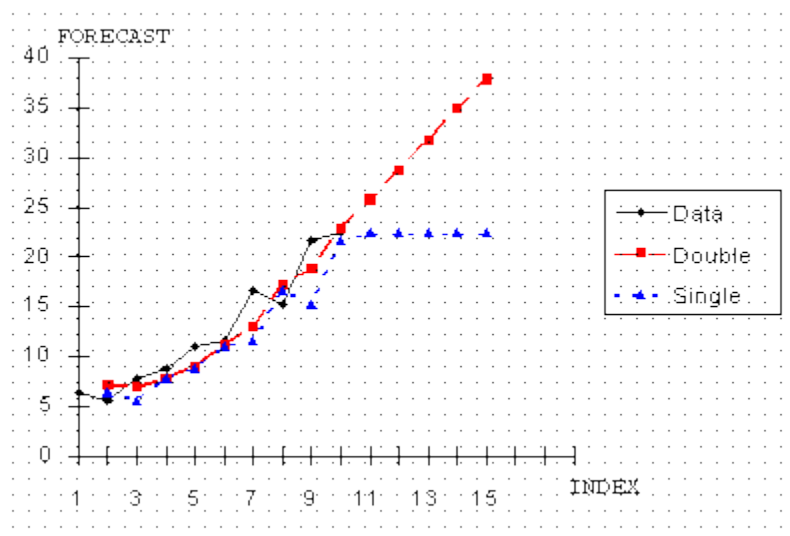
In de Tabel 4.1 - SES voorspelling hieronder is een te zien hoe met SES een voorspelling gedaan wordt waarbij het gewicht 0.2 is.

Days	Sales	Previous prediction	Prediction
1	24	-	23
2	40	23	$(0.2)(40) + (.8)(23) = 26.4$
3	25	26.4	$(0.2)(25) + (.8)(26.4) = 26.12$

Tabel 4.1 - SES voorspelling

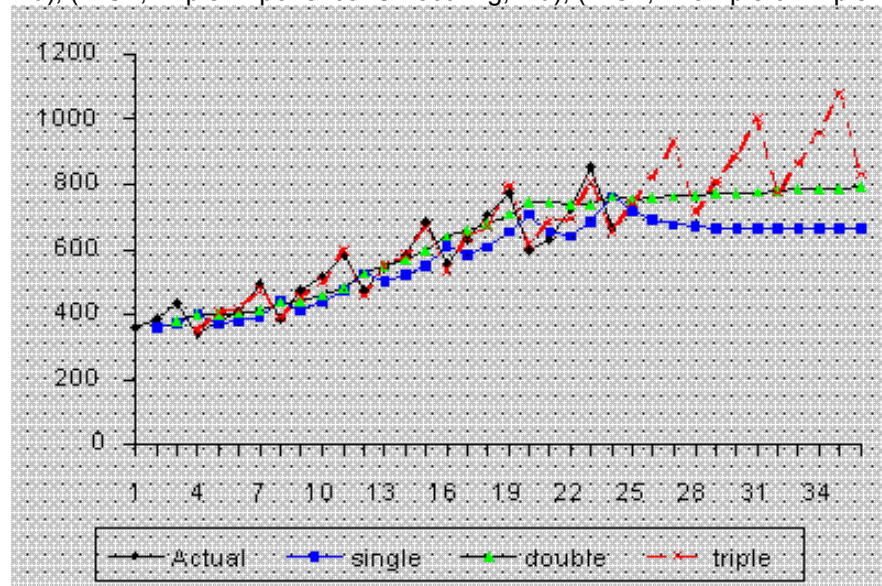
Op het moment dat de time serie een trend of seasonality bevat, kan SES hier niet goed mee overweg. Op het moment dat er een trend is kan double exponential smoothing gebruikt worden om tot een betere voorspelling te komen. (Prof & Toporowski, z.d), (Investopedia, Linearly Weighted Moving Average, z.d), (NIST, Single Exponential Smoothing, z.d), (Hyndman & Athanasopoulos, Simple exponential smoothing, z.d)

Double exponential smoothing (DES) doet het zelfde als SES maar houdt bij de voorspelling ook rekening met trends. In afbeelding 4.2 is te zien hoe DES (rode lijn) de opwaartse trend voorspelt en SES (blauwe lijn) niet kan voorspellen dat er een opwaartse trend is doordat SES alleen kijkt naar recente waarden. (Sourceforge, z.d)



Afbeelding 4.2 - Double exponential smoothing

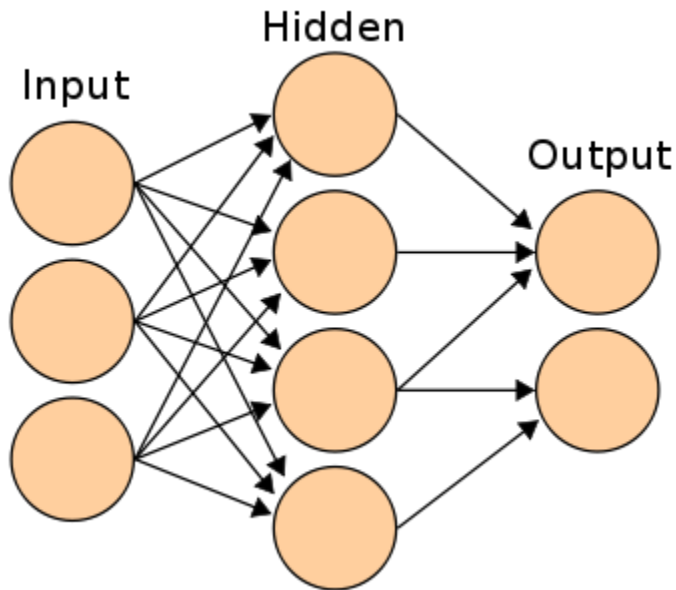
In het geval dat de time serie ook een seasonality bevat zullen SES of DES geen juiste voorspelling kunnen geven. In dit geval zal triple exponential smoothing gebruikt moeten worden. TES houdt naast de trend en recente waarden uit het verleden ook rekening met seasonality door extra gewicht te geven aan data dat zich één of meer seasonalities terug bevindt. In afbeelding 4.3 is te zien dat TES (rode lijn) de pieken en dalen (seasonality) van de actuele data volgt (zwarte lijn). (NIST, Forecasting with Double Exponential Smoothing(LASP), z.d), (Wikipedia, Exponential smoothing, 2014), (Hyndman & Athanasopoulos, Holt's linear trend method, z.d), (Hyndman & Athanasopoulos, Holt-Winters seasonal method, z.d), (NIST, Triple Exponential Smoothing, z.d), (NIST, Example of Triple Exponential Smoothing, z.d)



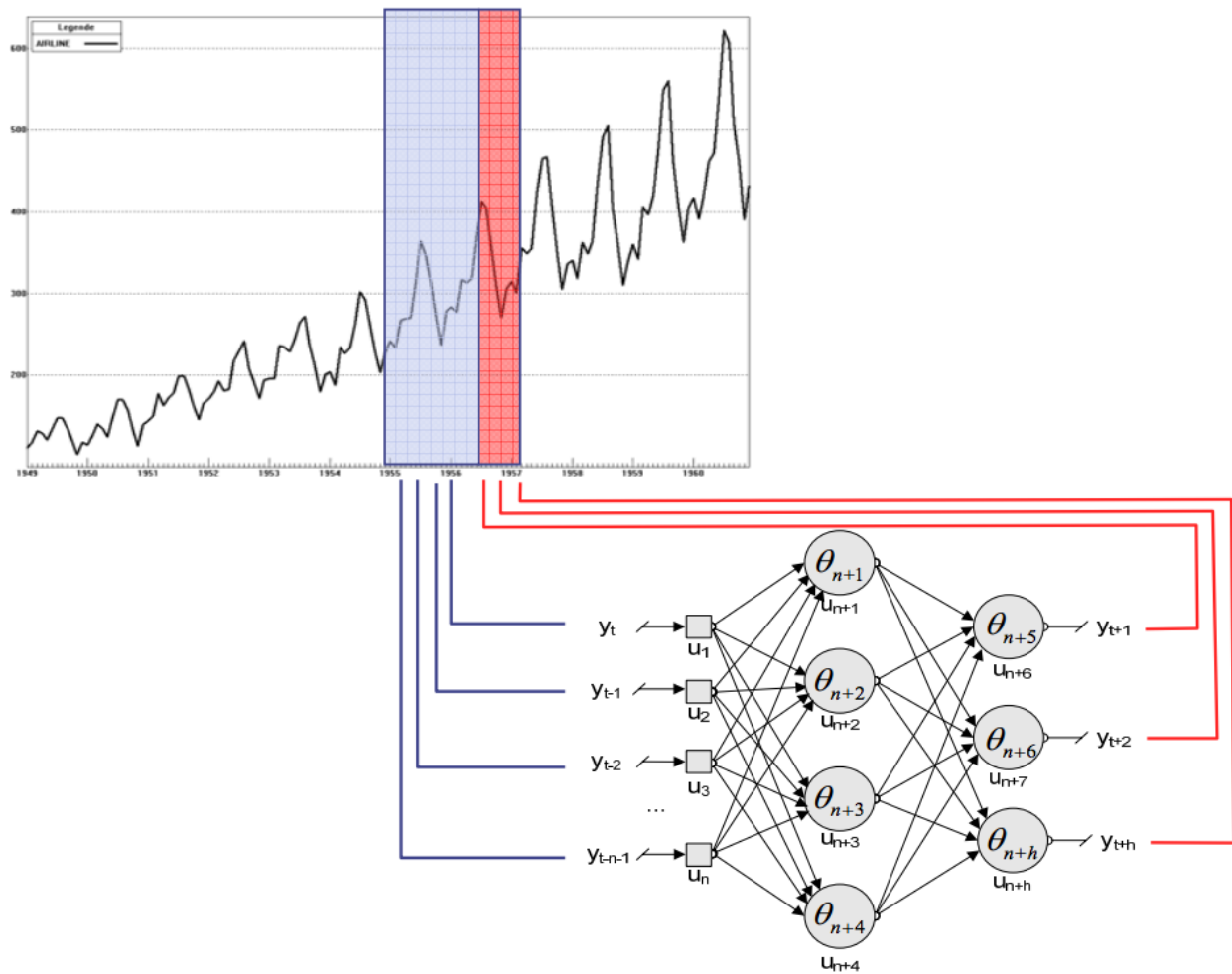
Afbeelding 4.3 - Triple exponential smoothing

5. Artificial neural networks

Artificial neural networks (ANN) is een zelf lerende voorspellingsmethode. De werking van een neural network kan gezien worden als de werking van neurons in onze hersenen. De neurons in hersenen en ANN ontvangen input die ze vervolgens verwerken wat vervolgens resulteert in output. In afbeelding 5.1 is te zien hoe dat in zijn werk gaat. De cirkels onder hidden zijn de neurons. afbeelding 5.2 toont hoe er met ANN een voorspelling gedaan wordt met time serie data, waarbij blauw de input is en rood de output.

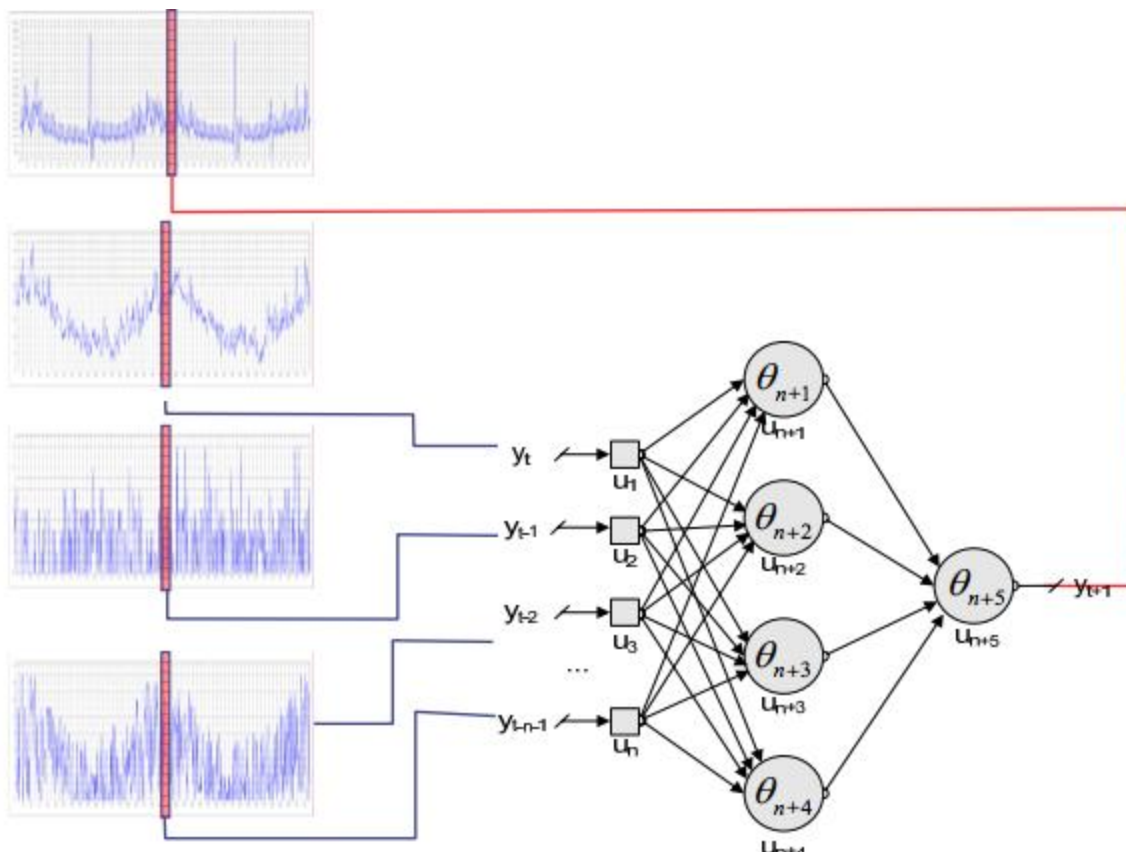


Afbeelding 5.1 - Artificial neural network



Afbeelding 5.2 - Time serie voorspelling met een artificial neural network

In de afbeeldingen is te zien dat er verschillende input gegeven wordt aan de hidden layers. In het geval van time serie analyse kan dit zoals in afbeelding 5.2 te zien is verschillende delen uit één time serie zijn of delen uit verschillende time series, zoals in afbeelding 5.3 te zien is. (Crone, 2005)



Afbeelding 5.3 - Artificial neural network voorspelling met meerdere time series

De verschillende input kan te vergelijken zijn met de zintuigen van een mens zoals, zien, horen en ruiken. Door deze drie zintuigen te combineren kan een mens een logische beredenering maken over wat die persoon ziet, hoort en ruikt. Een persoon kan hierdoor leren hoe een hond eruit ziet, wat voor geluid een hond maakt en hoe een hond ruikt. Als de persoon vervolgens na het leren van de kenmerken van een hond iets ziet dat lijkt op een hond, het geluid maakt van een hond en ruikt naar een hond kan de persoon datgene identificeren als een hond. (Wisconsin, z.d)

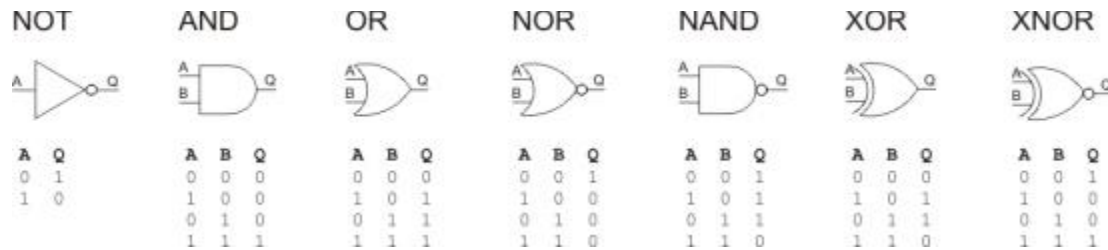
Op dezelfde manier kan een mens ook voorspellen welk getal er volgt na het bekijken van een reeks getallen. Als een persoon geleerd heeft wat de volgorde van getallen is, 1,2,3,4,...,101,102, etc, kan die persoon voorspellen wat het volgende getal is bij de reeks getallen 1,2,3,4,5, namelijk 6. Op het moment dat er een patroon in de reeks getallen is en de persoon ziet en leert van het patroon kan dit meegenomen worden in het doen van de voorspelling. Zo'n patroon kan zijn +2, -1, .., +2, -1, wat resulteert in 0,2,1,3,2,4,3,... Door het patroon te leren zal de persoon in staat zijn om te voorspellen dat het volgende getal 5 zal zijn.

Voor het doen van een accurate voorspelling is het van belang dat er bepaald wordt hoeveel neurons de hidden layers zal bevatten. Als er te weinig neurons gebruikt worden kan dit zorgen voor underfitting en als er te veel neurons gebruikt worden kan dit zorgen voor overfitting en een te lange trainingstijd waardoor het gebruik van ANN onbruikbaar wordt. Voor het bepalen van het aantal neurons zijn 3 vuistregels:

- Het aantal hidden neurons zitten tussen het formaat van de input layer en het formaat van de output layer.
- Het aantal hidden neurons moet 2 tot 3 keer het formaat zijn van de input layer plus het formaat van de output layer.
- Het aantal hidden neurons moet minder zijn dan twee keer het formaat van de input layer.

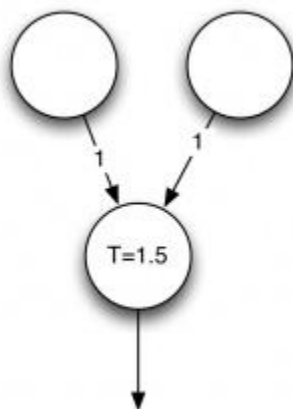
Naast het gebruik van de vuistregel blijft het ook een kwestie van uitproberen en kijken welke aantal neurons het beste resultaat geeft. (jeffheaton, 2008)

Een neuron kan gezien worden als een logische poort, zoals te zien is in afbeelding 5.4.



Afbeelding 5.4 - Logische poorten

De eigenschap van een AND poort is dat de output(Q) alleen 1 is als de twee inputs ook 1 zijn. Een neuron in ANN werkt op de zelfde manier maar bij ANN moet de totale input aan een bepaald gewicht voldoen. In afbeelding 5.5 is een neuron te zien die werkt als een AND poort.



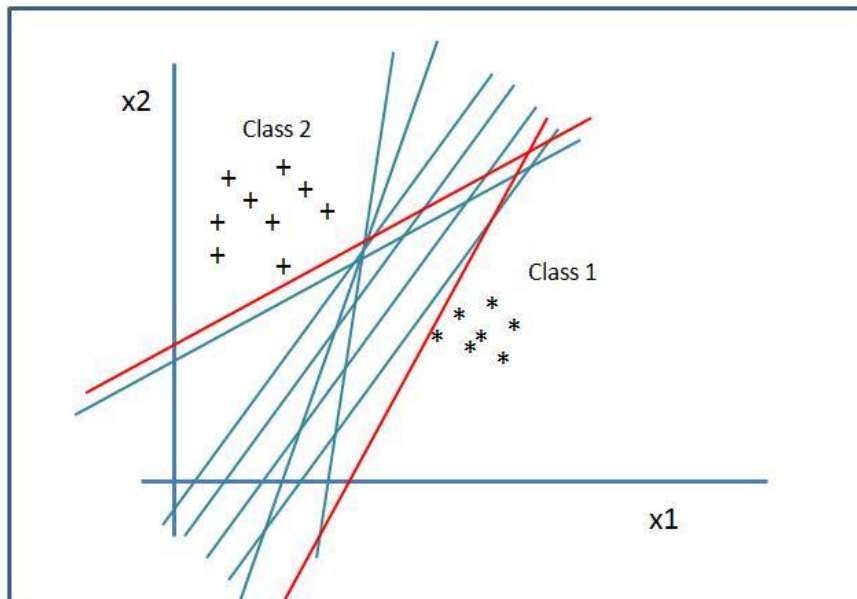
Afbeelding 5.5 - Neuron als AND poort -

T (1.5) is het gewicht waaraan de totale input moet voldoen en de twee op 1en op de pijlen zijn de gewichten van de input. De input bij elkaar is 2 dus dit betekent dat de neuron geactiveerd wordt. In ANN wordt tijdens de training bepaald wat het gewicht van een neuron is door herhaaldelijk de input te analyseren. (Crijns, 2014), (Heaton, 2012)

Nadat het ANN getraind heeft kan het ANN gebruikt worden voor het doen van voorspelling met nieuwe data, waarbij de nieuwe data overeenkomsten moet hebben met de data waarop getraind is. Dit is bijvoorbeeld een nieuwe time serie van de zelfde bron waarop getraind is. Doordat na het trainen bekend is welke en in welke volgorde de neurons geactiveerd zijn wordt er een voorspelling gedaan door de nieuwe data door de geactiveerde neurons te sturen. (Leverington, z.d), (Gershenson, z.d)

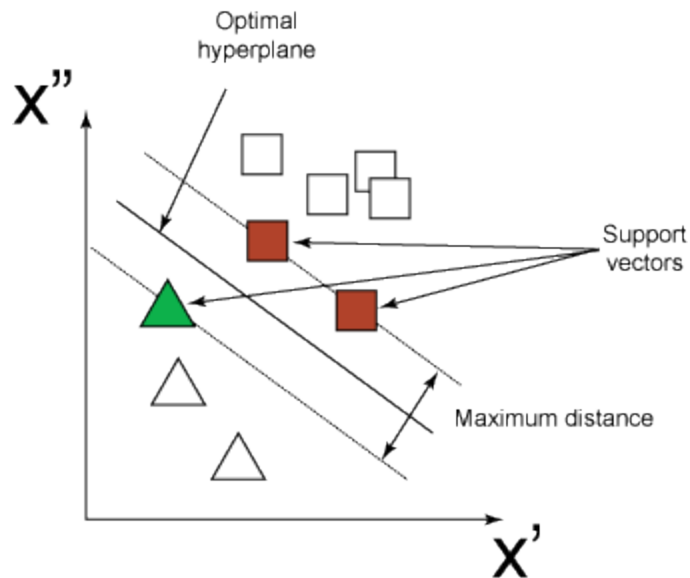
6. Support vector machine

Support vector machine (SVM) is een voorspellingsmethode die net zoals bij ANN traint aan de hand van trainings-data. Daarbij heeft SVM wel een set aan test-data nodig om te verifiëren of de voorspelling die gedaan wordt na het trainen accuraat is. De manier waarop SVM traint, is anders dan bij ANN. SVM is namelijk een lineair classifier. Bij lineaire classificatie worden datapunten onderscheiden aan de hand van een optimale hyperplane. De hyperplane is de lijn die de datapunten onderscheidt. In afbeelding 6.1 is te zien dat er meerdere hyperplanes mogelijk zijn voor het onderscheiden van datapunten. Van deze hyperplanes moet de optimale hyperplane gevonden.



Afbeelding 6.1 - Hyperplanes

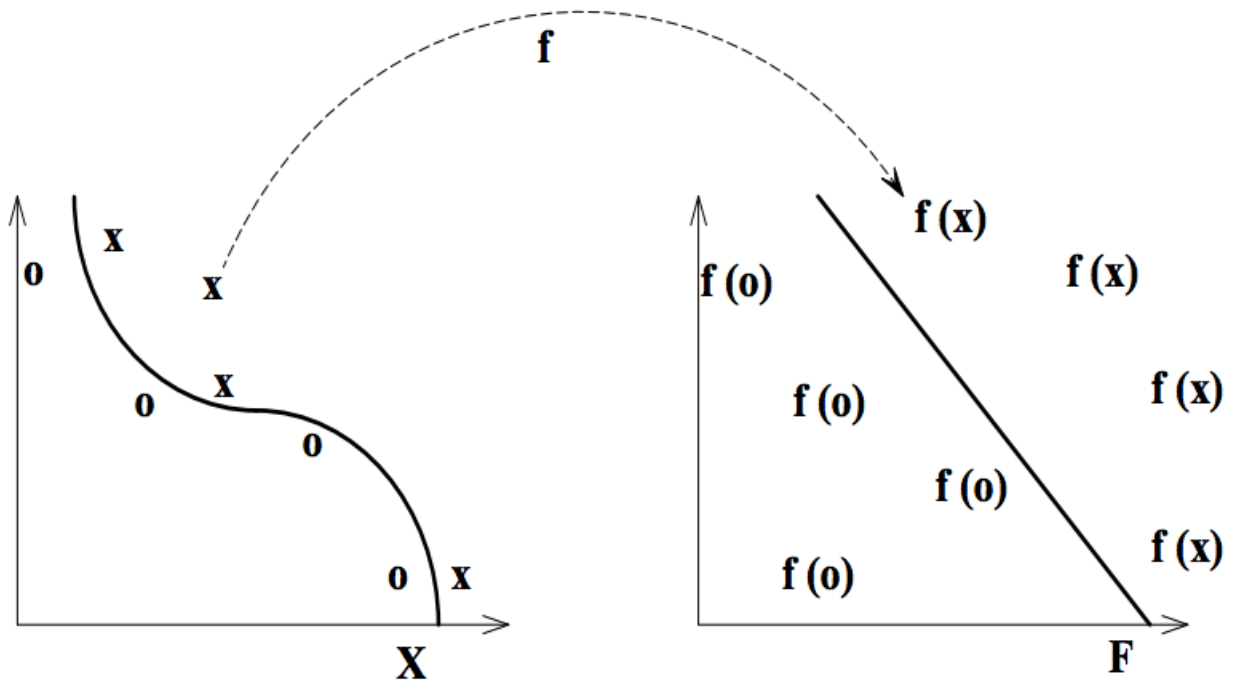
In afbeelding 6.2 is te zien hoe de optimale hyperplane gevonden wordt en de vierkanten en driehoeken onderscheidt. Het bepalen van de optimale hyperplane wordt gedaan door de maximale afstand te zoeken tussen de twee parallelle hyperplanes (de twee lijnen die parallel lopen aan de optimale hyperplane). De grootste afstand van de twee parallelle hyperplanes wordt bepaald door datapunten die op de parallelle hyperplanes liggen. Deze datapunten worden support vectors genoemd. (Support Vector Machines (SVM) Fundamentals Part-I, 2014)



Afbeelding 6.2 - Support vectoren

De support vectors zijn dus cruciaal voor het bepalen van de optimale hyperplane. Doordat de support vectors de optimale hyperplane bepalen, wordt de rest van de data overbodig en zijn alleen de support vectors nodig voor het trainen van de support vector machine.

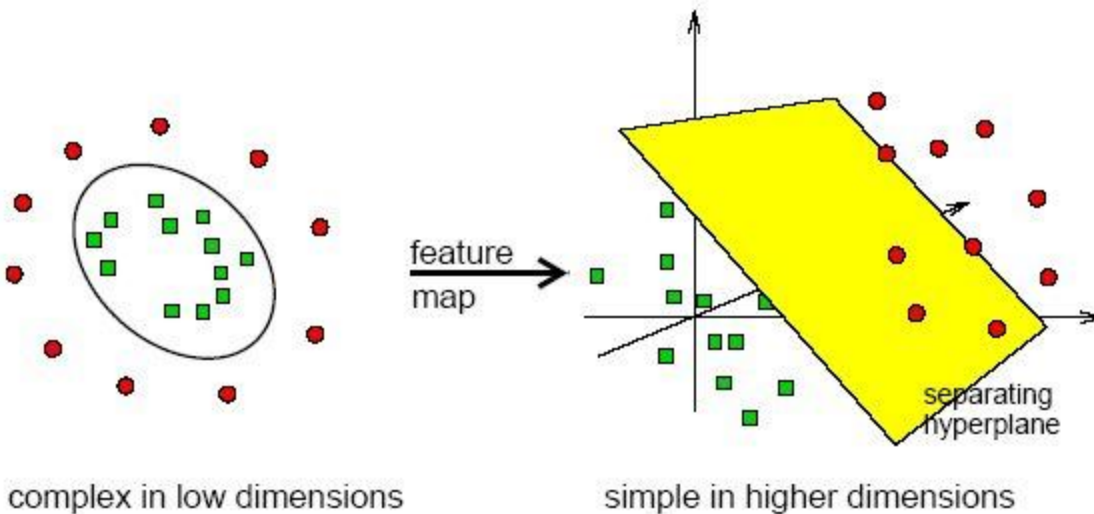
Voordat de vectors lineair onderscheiden kunnen worden met een optimale hyperplane kan het zijn dat de verschillende datapunten eerst uit elkaar gehaald moeten worden in een feature space. In afbeelding 6.3 is in de linker afbeelding te zien dat de datapunten door elkaar heen staan en niet lineair te onderscheiden zijn, er kan namelijk geen rechte lijn doorheen getrokken worden. Om ze toch lineair te onderscheiden worden ze in een feature space van elkaar gescheiden wat in de rechter afbeelding te zien is. (Guazzelli, 2012), (Settles, 2003)



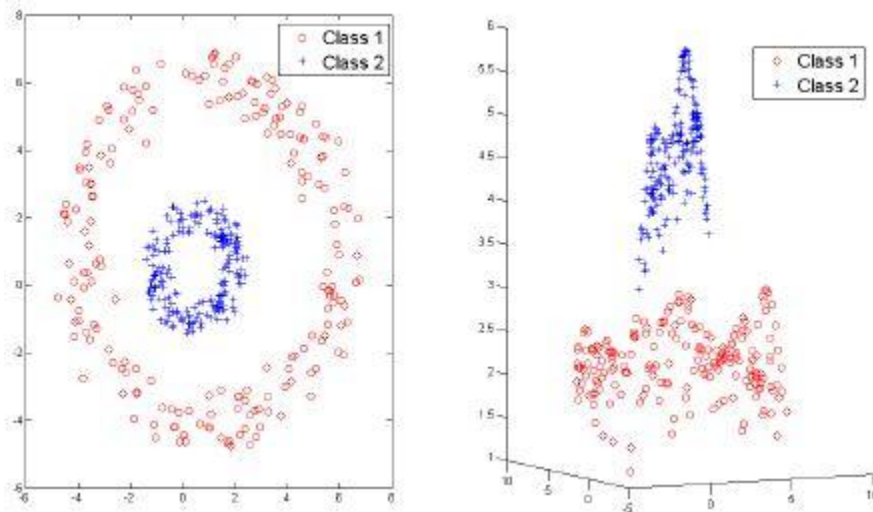
Afbeelding 6.3 - Feature space

In sommige gevallen is het lastig om data te onderscheiden voor het vinden van de optimale hyperplane waardoor de data onderscheiden moeten worden in een feature space in een hogere dimensie. In afbeelding 6.3 worden de datapunten onderscheiden door een 2-dimensionale hyperplane en in afbeelding 6.4 is te zien dat de data worden onderscheiden door een hogere dimensionale hyperplane, waarbij de waarde van de datapunten wordt gebruikt om ze te onderscheiden. Het is nodig om ze te onderscheiden in een hogere dimensie, omdat de groene vierkanten omringd worden door de rode cirkels waardoor ze niet onderscheiden kunnen worden met een hyperplane. In afbeelding 6.5 is duidelijk te zien hoe de rode en blauwe datapunten van elkaar onderscheiden worden door gebruik te maken van hun waarde (in een hogere dimensie). (Cristianini, 2001)

Separation may be easier in higher dimensions



Afbeelding 6.4 - Feature space in hogere dimensie 1



Afbeelding 6.5 - Feature space in hogere dimensie 2

Doordat er verschillende soorten data sets zijn waarbij data onderscheiden moeten worden, zoals te zien is in afbeelding 6.3, afbeelding 6.4 en afbeelding 6.5, maakt SVM gebruik van kernels. Deze kernels maken gebruik van kernel functies waarmee de data onderscheiden worden. Populaire kernels waar tussen gekozen kan worden bij het gebruik van SVM zijn:

1. Linear kernel
2. Polynomial Kernel
3. Radial Basis Function (RBF) Kernel
4. Neural Network Kernel

(DTREG, z.d), (Chen, z.d), (Support Vector Machines (SVM) Fundamentals Part-II, 2013), (Adhikari & Agrawal, z.d)

7. Bibliografie

- Support Vector Machines (SVM) Fundamentals Part-II.* (2013, Juli 30). Opgeroepen op Oktober 8, 2014, van Artificial Intelligence & Machine Learning: <http://panthimanshu17.wordpress.com/2013/07/30/support-vector-machines-svm-fundamentals-part-ii/>
- Support Vector Machines (SVM) Fundamentals Part-I.* (2014, Juli 18). Opgeroepen op Oktober 8, 2014, van Artificial Intelligence & Machine Learning: <http://panthimanshu17.wordpress.com/2013/07/28/svm-fundamentals-part-1/>
- Adhikari, R., & Agrawal, R. K. (z.d). *An Introductory Study on Time Series Modeling and Forecasting.* Opgeroepen op Oktober 8, 2014, van arVix: <http://arxiv.org/ftp/arxiv/papers/1302/1302.6613.pdf>
- AntiAnti. (2011, Januari 7). *How to diagnose problems with your statistical learning algorithm.* Opgeroepen op Oktober 8, 2014, van AntiAnti: <http://antianti.org/?p=175>
- Chen, B. (z.d). *Nonlinear Support Vector Machine.* Opgeroepen op Oktober 8, 2014, van Bindi Chen: <http://www.bindichen.co.uk/post/Al/Nonlinear-Support-Vector-Machines.html>
- Crijns, K. (2014, Augustus 4). *Achtergrond: hoe werkt een processor?* Opgeroepen op Oktober 8, 2014, van Hardware: <http://nl.hardware.info/reviews/1772/3/achtergrond-hoe-werkt-een-processor-booleaanse-logica>
- Cristianini, N. (2001). *Support Vector and Kernel Machines.* Opgeroepen op Oktober 8, 2014, van Support Vector Machines: <http://www.support-vector.net/icml-tutorial.pdf>
- Crone, S. F. (2005, December 15). *Forecasting with Artificial Neural Networks.* Opgeroepen op Oktober 8, 2014, van Neural Forecasting: http://www.neural-forecasting.com/Downloads/EVIC05_tutorial/EVIC'05%20Slides%20-%20Forecasting%20with%20Neural%20Networks%20Tutorial%20SFCrone.pdf
- Data, M. I. (z.d). *Model Identification for Southern Oscillations Data.* Opgeroepen op Oktober 8, 2014, van NIST: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc4461.htm>
- data, T. s. (2013, Maart 26). *General regression and over fitting.* Opgeroepen op Oktober 8, 2014, van The shape of data: <http://shapeofdata.wordpress.com/2013/03/26/general-regression-and-over-fitting/>
- DTREG. (z.d). *SVM - Support Vector Machines.* Opgeroepen op Oktober 8, 2014, van DTREG: <http://www.dtreg.com/svm.htm>
- Duke. (z.d). *Stationarity and differencing.* Opgeroepen op Oktober 8, 2014, van Duke: <http://people.duke.edu/~rnau/411diff.htm>

- Eurostat. (z.d). *Lag*. Opgeroepen op Oktober 8, 2014, van Eurostat:
http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Glossary:Lag
- Eurostat. (z.d). *Time lag*. Opgeroepen op Oktober 8, 2014, van Eurostat:
http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Glossary:Time_lag
- Eurostat. (z.d). *Trend*. Opgeroepen op Oktober 8, 2014, van Eurostat:
http://epp.eurostat.ec.europa.eu/statistics_explained/index.php/Glossary:Trend
- Fortmann-Roe, S. (2012, Juni). *Understanding the Bias-Variance Tradeoff*. Opgeroepen op Oktober 8, 2014, van Scott Fortmann-Roe: <http://scott.fortmann-roe.com/docs/BiasVariance.html>
- Gershenson, C. (z.d). *Artificial Neural Networks for Beginners*. Opgeroepen op Oktober 8, 2014, van arVix: <http://arxiv.org/ftp/cs/papers/0308/0308031.pdf>
- Guazzelli, A. (2012, Juni 19). *Predicting the future, Part 2: Predictive modeling techniques*. Opgeroepen op Oktober 8, 2014, van IBM:
<https://www.ibm.com/developerworks/library/ba-predictive-analytics2/>
- Gutierrez-Osuna, R. (z.d). *Lecture 13: Validation*. Opgeroepen op Oktober 8, 2014, van Perception sensing instrumentation:
http://research.cs.tamu.edu/prism/lectures/iss/iss_l13.pdf
- Heaton, J. (2012, Maart 31). *NN Session 1: What is a Neural Network? Weights & Thresholds*. Opgeroepen op Oktober 8, 2014, van Heaton research:
http://www.heatonresearch.com/wiki/NN_Session_1:_What_is_a_Neural_Network%3F_Weights_%26_Thresholds
- Hyndman, R. J., & Athanasopoulos, G. (z.d). *Holt's linear trend method*. Opgeroepen op Oktober 8, 2014, van OText: <https://www.otexts.org/fpp/7/2>
- Hyndman, R. J., & Athanasopoulos, G. (z.d). *Holt-Winters seasonal method*. Opgeroepen op Oktober 8, 2014, van OText: <https://www.otexts.org/fpp/7/5>
- Hyndman, R. J., & Athanasopoulos, G. (z.d). *Simple exponential smoothing*. Opgeroepen op Oktober 8, 2014, van OText: <https://www.otexts.org/fpp/7/1>
- Investopedia. (z.d). *Autoregressive*. Opgeroepen op Oktober 8, 2014, van Investopedia:
<http://www.investopedia.com/terms/a/autoregressive.asp>
- Investopedia. (z.d). *Linearly Weighted Moving Average*. Opgeroepen op Oktober 8, 2014, van Investopedia: <http://www.investopedia.com/terms/l/linearlyweightedmovingaverage.asp>
- Investopedia. (z.d). *Moving Average - MA*. Opgeroepen op Oktober 8, 2014, van Investopedia:
<http://www.investopedia.com/terms/m/movingaverage.asp>

- jeffheaton. (2008, September 14). *The Number of Hidden Layers*. Opgeroepen op Oktober 8, 2014, van Heaton research: <http://www.heatonresearch.com/node/707>
- Kriege, H.-P., Kröger, P., & Zimek, A. (2010). *Outlier Detection Techniques*. Opgeroepen op Oktober 8, 2014, van Siam: <http://www.siam.org/meetings/sdm10/tutorial3.pdf>
- Leverington, D. (z.d). *A Basic Introduction to Feedforward Backpropagation Neural Networks*. Opgeroepen op Oktober 8, 2014, van Texas Tech University: http://www.webpages.ttu.edu/dleverin/neural_network/neural_networks.html
- MathWorks. (z.d). *Autocorrelation and Partial Autocorrelation*. Opgeroepen op Oktober 8, 2014, van MathWorks: <http://www.mathworks.nl/help/econ/autocorrelation-and-partial-autocorrelation.html#bs87k2k>
- NCSS. (z.d). *The Box-Jenkins Method*. Opgeroepen op Oktober 8, 2014, van NCSS: http://www.ncss.com/wp-content/themes/ncss/pdf/Procedures/NCSS/The_Box-Jenkins_Method.pdf
- NIST. (z.d). *Box-Jenkins Model Identification*. Opgeroepen op Oktober 8, 2014, van NIST: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc446.htm>
- NIST. (z.d). *Example of Triple Exponential Smoothing*. Opgeroepen op Oktober 8, 2014, van NIST: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc436.htm>
- NIST. (z.d). *Forecasting with Double Exponential Smoothing(LASP)*. Opgeroepen op Oktober 8, 2014, van NIST: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc434.htm>
- NIST. (z.d). *Seasonality*. Opgeroepen op Oktober 8, 2014, van NIST: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc443.htm>
- NIST. (z.d). *Single Exponential Smoothing*. Opgeroepen op Oktober 8, 2014, van NIST: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc431.htm>
- NIST. (z.d). *Triple Exponential Smoothing*. Opgeroepen op Oktober 8, 2014, van NIST: <http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc435.htm>
- Prof, P. D., & Toporowski, D. W. (z.d). *Smoothing methods Smoothing methods*. Opgeroepen op Oktober 8, 2014, van http://www.statok.wiso.uni-goettingen.de/veranstaltungen/graduateseminar/SmoothingMethods_Narodzonek-Karpowska.pdf
- Reference, M. O. (z.d). *Outlier*. Opgeroepen op Oktober 8, 2014, van Math Open Reference: <http://www.mathopenref.com/outlier.html>
- Settles, B. H. (2003). *Feature space*. Opgeroepen op Oktober 8, 2014, van The university of Wisconsin: http://pages.cs.wisc.edu/~bsettles/cs540/lectures/16_feature_spaces.pdf

- Sourceforge. (z.d). *Class DoubleExponentialSmoothingModel*. Opgeroepen op Oktober 8, 2014, van Sourceforge:
<http://openforecast.sourceforge.net/docs/net/sourceforge/openforecast/models/DoubleExponentialSmoothingModel.html>
- Wikipedia. (2013, Augustus 24). *Extrapolatie*. Opgeroepen op Oktober 8, 2014, van Wikipedia:
<http://nl.wikipedia.org/wiki/Extrapolatie>
- Wikipedia. (2014, September 14). *Autocorrelation*. Opgeroepen op Oktober 8, 2014, van Wikipedia: <http://en.wikipedia.org/wiki/Autocorrelation>
- Wikipedia. (2014, Oktober 1). *Autoregressive integrated moving average*. Opgeroepen op Oktober 8, 2014, van Wikipedia:
http://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average
- Wikipedia. (2014, September 15). *Dickey–Fuller test*. Opgeroepen op Oktober 8, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Dickey%E2%80%93Fuller_test
- Wikipedia. (2014, Septembr 8). *Exponential smoothing*. Opgeroepen op Oktober 8, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Exponential_smoothing
- Wikipedia. (2014, Oktober 7). *Linear regression*. Opgeroepen op Oktober 8, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Linear_regression
- Wikipedia. (2014, Oktober 1). *Outlier*. Opgeroepen op Oktober 8, 2014, van Wikipedia:
<http://en.wikipedia.org/wiki/Outlier>
- Wikipedia. (2014, Juni 25). *Partial autocorrelation function*. Opgeroepen op Oktober 8, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Partial_autocorrelation_function
- Wikipedia. (2014, September 24). *Stationary process*. Opgeroepen op Oktober 8, 2014, van Wikipedia: http://en.wikipedia.org/wiki/Stationary_process
- Wisconsin, T. u. (z.d). *A Basic Introduction To Neural Networks*. Opgeroepen op Oktober 8, 2014, van The university of Wisconsin:
<http://pages.cs.wisc.edu/~bolo/shipyard/neural/local.html>
- Ye, F., & PhD. (2010, April 16). *What You See May Not be What You Get – A Brief Introduction to Overfitting*. Opgeroepen op Oktober 8, 2014, van Department of Biostatistics:
http://biostat.mc.vanderbilt.edu/wiki/pub/Main/AlexZhao/Overfitting_Cancer_workshop_04162010.pdf
- Yuan, D., Joshi, N., Jacobson, D., & Oberai, P. (2014, December 4). *Scryer: Netflix's Predictive Auto Scaling Engine - Part 2*. Opgeroepen op Oktober 8, 2014, van Netflix:
<http://techblog.netflix.com/2013/12/scryer-netflixs-predictive-auto-scaling.html>

Zhang, Y., & Wu, L. (2012, September 13). *Classification of Fruits Using Computer Vision and a Multiclass Support Vector Machine*. Opgeroepen op Oktober 8, 2014, van MDPI:
http://www.mdpi.com/1424-8220/12/9/12489/html#fig_body_display_f7-sensors-12-12489