Voorspellen incidenten in het Optische Transport Netwerk op basis van historische event data

Afstuderen bachelor Toegepaste Wiskunde

Casper Lubbers
Studentnummer: 12076694

Begeleiders:
D. van Huizen
W. Mulder
W. Dekker

KPN B.V.
Wilhelminakade 123
3072 AP Rotterdam

Haagse Hogeschool Delft
Rotterdamseweg 137
2628 AL Delft

Mei 2019

Voorwoord

Voor u ligt mijn scriptie, geschreven ter afsluiting van de bachelor Toegepaste Wiskunde aan de Haagse Hogeschool te Delft. In deze scriptie vindt u de resultaten van mijn afgeronde afstudeerstage bij KPN in de periode februari tot en met mei 2019.

Voor inhoudelijke begeleiding wil ik allereerst Willem-Jan Dekker bedanken voor de uitvoerige hulp en ondersteuning. Op elk moment stond je klaar voor mijn vragen, welke je met plezier uitvoerig en meer dan volledig beantwoordde. Aanvullend wil ik ook Wico Mulder en Dick van Huizen bedanken voor de begeleiding op basis van hun expertise in dit vakgebied.

Verder wil ik Marissa Groenewegen bedanken voor het faciliteren van het kader waarin mijn afstuderen plaatsvond en wil ik Sacha van de Weijer bedanken voor het aanbieden van de opdracht. Tot slot wil ik ook Ruud Vermeij en Karin de Smidt bedanken voor het lezen van mijn scriptie.

Casper Lubbers

Mei 2019

Samenvatting

Om het netwerk van KPN te monitoren en de kwaliteit van de dienstverlening te kunnen borgen, heeft KPN het Service Quality Center in het leven geroepen. Het netwerk is te omschrijven als een reeks verbindingen tussen componenten, waar een component een netwerk dragend onderdeel is en uit meerdere kleinere onderdelen bestaat. Elk component in het netwerk wordt hier gemonitord op prestaties. In het geval een vooraf gestelde grenswaarde overschrijdt, wordt er een event aangemaakt in het event-registratiesysteem, inclusief de *severity*. De severity is indicatief voor de ernst van een event en wordt door KPN gebruikt als startpunt om te bepalen of er al dan niet actie wordt ondernomen. Over het algemeen wordt er enkel op events met een hoge severity gereageerd.

Om minder reactief te sturen op de events heeft KPN verzocht onderzoek te doen naar de mogelijkheid om te kunnen voorspellen welke componenten een verhoogd risico op het veroorzaken van een incident hebben en waarom. De scope van dit onderzoek is beperkt tot het Optische Transport Netwerk. Er is enkel data beschikbaar op het component niveau dus zodoende wordt er enkel gekeken naar dit data niveau. Dit betekent dat onderliggende kleinere onderdelen die een component opmaken als een enkel component worden beschouwd.

Voor dit onderzoek zijn vier databronnen gebruikt, namelijk: historische eventdata; historische incidentdata; topologische netwerkgegevens; de topografische ligging van systemen en de klimatologische omstandigheden. In dit onderzoek zijn meerdere modellen uitgewerkt waarmee het mogelijk is om met een hoge nauwkeurigheid te voorspellen welk component voor een verhoogd risico op het veroorzaken van een incident zorgt. Er zijn een drietal modellen toegepast die los van elkaar diverse inzichten geven, maar gecombineerd de mogelijkheid hebben om incidenten te voorspellen.

Allereerst is er een survival-regressiemodel opgezet dat kan voorspellen welk component een verhoogd risico heeft op falen. Dit model is in staat om dit te kunnen voorspellen met een C-index van 0.85. Het survivalmodel is verder geanalyseerd met behulp van de Shapley Additive Explanations (SHAP) om informatie te krijgen over de mate van invloed van variabelen op een gegeven voorspelling. Ter aanvulling is er een non-parametrisch survivalmodel toegepast om de overlevingsfunctie en hazard-functie te krijgen. Op basis van deze functies is gebleken dat in de eerste dag na herstel van een component de kans op falen het grootste is. Het derde toegepaste model een Markov-keten, welke inzicht gaf aan het type events dat tot incidenten of tot andere type events leidt, inclusief de gemiddelde tijd tussen de events in.

Het is de combinatie van deze drie modellen dat het mogelijk maakt om met grote nauwkeurigheid risico lopende componenten aan te merken, inclusief onderbouwing waarom deze een verhoogd risico hebben en welke events hier waarschijnlijk op zullen volgen. Zodoende is KPN goed geïnformeerd over de oorzaken van de risico's en is er sturing naar de juiste preventieve acties.

English Summary

To monitor the network of KPN and guarantee the quality of services, KPN has the Service Quality Center. The network of KPN can be described as a series of connections between components, where each component in this sense is a carrying member of the network. Meaning there is a flow of data going through the component. These components are called nodes in the network. Each node in the network is monitored for their performance and when a threshold seems to be passed an event is triggered. This event is accompanied with a severity score, with a high severity scoring representing serious issues. In most cases high severity events trigger an incident which means that correcting measures will be taken. Events with low severity are often ignored, despite the possibility that they can lead directly to incidents.

To be able to work more data driven KPN has requested research to be performed to evaluate the possibility of predicting the risk of incidents. In talks with KPN the following research question has been phrased: Is it possible to predict the possibility of an incident on a given node based on historical event data? The research will be scoped on the Optical Transport Network. There is only data available on the node level, thus only this level of data will be considered in this research.

In this research, multiple models are used to perform analysis and predictions. Three different models are used for different purposes and an answer is given to the main research question. The data that is used is based on historical event data, historical incident data, the topological relations of the network, the topographical data and the climatological data.

The first and foremost model is the XGBoost model that is used to predict the Hazard Ratio on a given node with a C-index of 0.85. The XGBoost model and the variable importance's are further analyzed with the use of Shapley Additive Explanations providing insights for each forecast. In addition to the XGBoost model, the non-parametric survival models of Kaplan-Meier and Nelson-Aalen which showed that the first day after maintenance a node is the most likely to cause an incident and must be monitored extra closely. A third model is the Markov chain that provides insights in the sequence of event types that cause an incident, including the mean time between the events.

The combination of these three models enable a high accuracy for predicting the risk of a given node and explain why the model has given these predictions. With the aid of these models KPN is now able to make more informed decisions on what nodes to perform maintenance. Based on the results of this model it is recommended to employ the model and add more descriptive variables to the model for improved accuracy and more insights.

Inhoudsopgave

Afkortingen

SQC	Service Quality Center
OTN	Optische Transport Netwerk
ETN	Ethernet Transport Netwerk
(D)WDM	Dense Wavelength Division Multiplexing
WDM	Wavelength Division Multiplexing
Cox PH	Cox Proportional Hazard
XGBoost	Extreme Gradient Boosting
HMM	Hidden Markov Model
SHAP	Shapley Additivite Explanations

1. Introductie en huidige situatie

In dit hoofdstuk leest u over de achtergrond van de probleemstelling en over het bedrijf KPN als opdrachtgever. Aanvullend zal er een toelichting gegeven worden over de afdeling waarvoor dit onderzoek is gedaan. Tot slot zullen de deelvragen worden toegelicht en zal er een leeswijzer gegeven worden van het verslag als geheel.

1.1. KPN

Koninklijk PTT Nederland (KPN) is een van de grootste en toonaangevende telecomproviders van Nederland en markteleider op het gebied van telecommunicatie en IT. Het is ontstaan uit de privatisering van de PTT in 1989. De PTT was voor de privatisering het grootste staatbedrijf voor posterijen, telefonie en telegrafiediensten wat bijdraagt aan de rijke geschiedenis van KPN. Met de vaste en mobiele netwerken voor telefonie, data en televisie bedient KPN Nederlandse klanten in binnen- en buitenland. KPN richt zich op zowel particuliere klanten als zakelijke gebruikers, van klein tot groot. Sinds de privatisering heeft KPN de kernactiviteiten gemoderniseerd met als slogan: *Voel je vrij.*

1.2. Definities

In dit verslag wordt er veelvoudig gebruik gemaakt van een drietal definities:

1) Events: een event die wordt geproduceerd door het systeem waar het falen optreedt. Een event gaat altijd gepaard met een alertgroup. De typering van het soort fout dat zich heeft ontwikkeld. Er zijn 182 unieke alertgroups voor het OTN gedefinieerd.
2) Tickets: een geregistreerd incident in een centraal gebruikt systeem bij KPN waarop (al dan niet handmatige) handelingen op volgen.
3) Incident: de fout heeft geleid tot een impact op de dienstverlening.

1.3. De afdeling Service Quality Center

Om de dienstverlening van het netwerk te monitoren heeft KPN de afdeling Service Quality Center (SQC) opgezet. Dit is een centrale afdeling die ervoor zorgt dat de klanten een goede beschikbaarheid en kwaliteit van de KPN-diensten ervaren. Het SQC zorgt hiervoor door vanaf een centrale plek het onderhoud aan te sturen. Het is dan ook een vereiste dat het internet voor zowel de consumenten als zakelijke markt zoveel mogelijk beschikbaar is.

KPN heeft duizenden kilometers kabel onder- en bovengronds om Nederland te verbinden. Tussen elk van deze kabels liggen diverse passieve en actieve componenten die ervoor zorgen dat elke gebruiker van het netwerk de afgenomen diensten geleverd krijgt.

De gebruikte actieve netwerkapparatuur genereert grote volumes ‘managementinformatie’, bedoeld voor beheer van drie onderwerpen, namelijk de apparatuur, de netwerken waarvan ze deel uitmaken en het netwerkverkeer. In specifieke gevallen kan deze informatie ook rapporteren over de kwaliteit en beschikbaarheid van de eindgebruikersdiensten.

Generiek gesproken bestaat de informatie in drie verschillende vormen:

- Events – spontane real-time berichten als gevolg van een bepaalde conditie
- Performance data – bestanden met meetgegevens (tellers).
- Logfiles – bestanden met allerlei berichten gerelateerd aan het functioneren en gebruik van de apparatuur.

Bewaking van apparatuur, netwerken en diensten gebeurt vooral op basis van events.

Het netwerk bestaat uit meerdere lagen en netwerktechnologieën. Het Optische Transport Netwerk (OTN) is de laag van het netwerk van KPN waarover al het dataverkeer gaat en bestaat exclusief uit verbindingen met een bandbreedte van minimaal 10 GB/s. Het OTN is gebaseerd op Dense Wavelength Division Multiplexing ((D)WDM) en maakt op zijn beurt gebruik van glasvezels. Dit OTN dient als een substantiële transportlaag voor de overige netwerklagen en vertakt naar kleinere sub-netwerken. Het OTN is zodoende een cruciaal onderdeel van het KPN-netwerk.

Vrijwel elk component in het netwerk van KPN wordt gemonitord, waarvoor verschillende systemen en applicaties gebruikt worden. U2000 is het netwerkmanagementsysteem voor het OTN. U2000 heeft een belangrijke netwerkbewakingsfunctie. Deze applicatie maakt het mogelijk om cruciale prestatiegegevens van het OTN en zijn componenten in de gaten te houden. In het geval dat een grenswaarde van een dergelijk component wordt overschreden, wordt er een event aangemaakt. Een dergelijk event wordt gewaardeerd met een van de vier volgende severities, welke onder te verdelen zijn in twee risicogroepen:

- *Minor* of *Warning* (lage severity): een waarschuwing zonder direct risico
- *Critical* of *Major* (hoge severity): een melding risico op impact op de dienstverlening

Omdat er enorme aantallen events worden gegenereerd, wordt er vooral geacteerd op events met een hoge severity. Deze events worden als belangrijkst beschouwd en hebben in de regel noodzaak tot directe opvolging. Events met een lage severity worden vaak genegeerd. Een aanvullende context wordt gegeven in het volgende hoofdstuk Context.

1.4. Probleemomschrijving

Bij grote dienstverstoringen worden de events, ook wel alarmeringen genoemd, vaak achteraf geanalyseerd met als doel te onderzoeken of de verstoring vooraf gedetecteerd had kunnen worden. Er wordt dan gekeken naar individuele events uit verschillende technische domeinen, maar meestal niet naar patronen en verbanden tussen deze events. Moderne technieken, zoals de recentelijke opkomst van Machine Learning, maken het mogelijk om op een andere manier naar de gegevens te kijken.

Omdat het OTN drager is van veel diensten, lijkt het logisch om de scope in eerste instantie te beperken tot dit netwerk. Een onderliggende theorie is dat een veelvoud van lage severity events zal leiden tot een hoog severity event. De doelstelling om het aantal incidenten te reduceren door middel van een voorspellingsmodel op basis van al bestaande data, leidt tot de volgende onderzoeksvraag:

In hoeverre is het mogelijk om een incident te voorspellen in het OTN op basis van historische gegevens?

Om deze vraag te kunnen beantwoorden is het eerst van belang om inzicht te krijgen in de beschikbare data. Op basis van deze data ontstaat vervolgens de mogelijkheid om een keuze te maken voor een voorspellingsmodel. Om dit model op te zetten is het ook van belang om te weten hoe fouten zich ontwikkelen binnen het netwerk, zodat er gericht gezocht kan worden naar variabelen die als input voor het model (kunnen) dienen. Na het ontwikkelen en fine-tunen van het model is het van belang dat er beschreven wordt hoe de resultaten van dit onderzoek gebruikt kunnen worden door KPN. Zodoende geeft dit onderzoek antwoord op de volgende deelvragen:

- Welke aanwezige data is geschikt voor het voorspellen van incidenten?
- Welke modellen komen in aanmerking voor het maken van voorspellingen?
- Hoe ontwikkelen incidenten zich binnen het netwerk van KPN?

- Hoe kunnen de resultaten gebruikt worden door KPN?

Zodoende worden in het hoofdstuk Context van dit verslag de diverse beschikbaar gestelde databronnen uiteengezet en beschreven. Afsluitend aan dit hoofdstuk wordt er een conclusie gegeven over de mate waarin deze bronnen bruikbaar zijn voor analyse.

Na het hebben beschreven van de context wordt het in het hoofdstuk Theoretische Kader bepaald waarin dit onderzoek plaatsvindt. Naast het geven van een aantal definities wordt ook beschreven in hoeverre incidenten voorspelbaar zijn. Vervolgens worden de onderzoek stappen en resultaten beschreven over de werking van de in aanmerking komende modellen. Na duidelijkheid over de gebruikte modellen zijn de verschillende evaluatiemethodes uiteengezet waarmee correctheid van de modeluitkomsten kan worden vastgesteld.

In het hoofdstuk Methodologie wordt beschreven hoe de data is gekoppeld. Er zijn meerdere databronnen beschikbaar en wordt uitgezocht hoe deze bronnen gecombineerd kunnen worden. Aanvullend welke stappen zijn gezet om dit mogelijk te maken, waarna wordt er uiteengezet hoe verificatie op compleetheid van de data heeft plaatsgevonden.

Vervolgens worden in het hoofdstuk Modelleren de uiteindelijke keuzes en uitwerkingen omschreven op het gebied van model, data en datapreparaties.

Tot slot worden in het hoofdstuk Resultaten en Conclusie de resultaten benoemd en geëvalueerd waarop in het hoofdstuk Aanbevelingen een aanbeveling voor KPN gedaan is. De aanbeveling bevat hoe KPN de resultaten van dit onderzoek kan gebruiken om de dienstverlening te verbeteren.

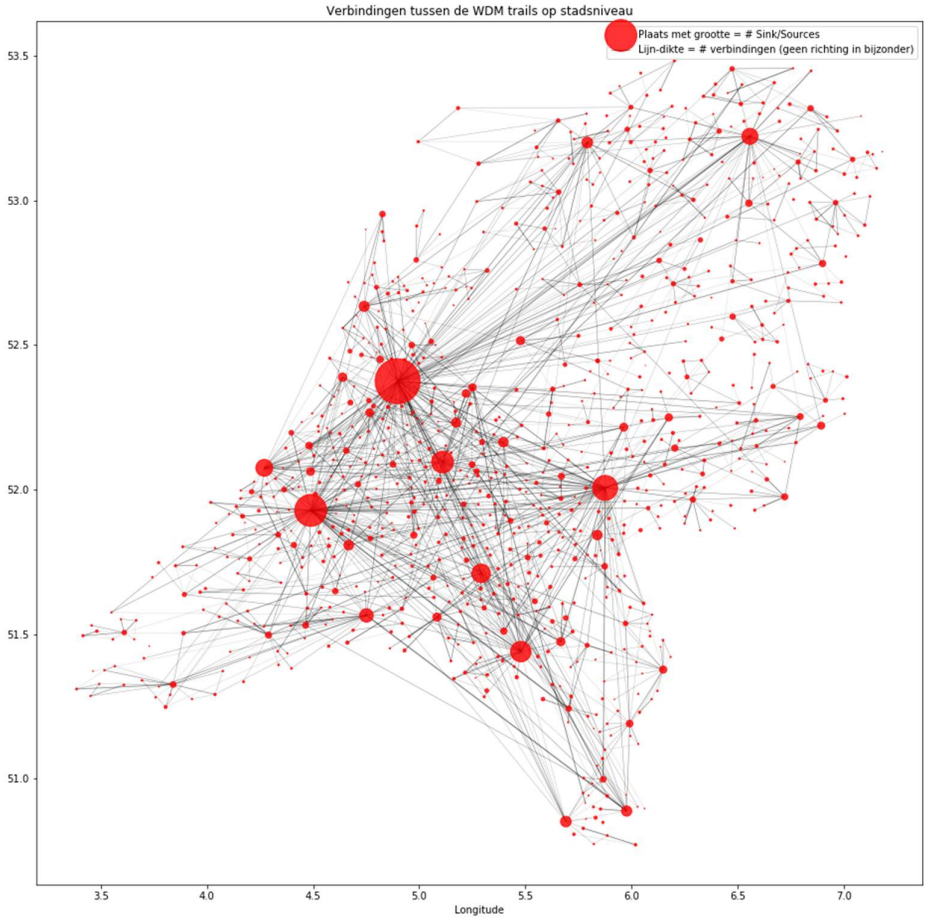
2. Context

In dit hoofdstuk wordt het OTN beschreven en zijn verhouding tot de overige netwerken van KPN. Aanvullend wordt de beschikbare data behandeld en een antwoord geformuleerd op de eerste deelvraag: *Welke aanwezige data is geschikt voor het voorspellen van incidenten?*

2.1. Netwerkbeschrijving

Het over de jaren heen veelvuldig uitgebreide en verbeterde netwerk van KPN bestaat uit diverse lagen waarvan OTN-laag de ruggengraat is. Het OTN faciliteert met de verbinding tussen alle kleinere sub-netwerken binnen het netwerk van KPN dat iedereen in Nederland met elkaar is verbonden.

Het netwerk is opgebouwd uit nodes (een netwerkkaart waar verbinden op gekoppeld worden) en verbindingen. In Figuur 1 is een vereenvoudigde logische weergave te zien van deze nodes en verbindingen in het OTN. De coördinaten van de plaats (stad of dorp) waar deze node zich bevindt zijn gebruikt, waardoor er meerdere locaties in een zelfde plaats zijn weergeven als een enkele node. De onderliggende data van deze weergave is tevens de input van het modelleren en biedt vooralsnog een interessante weergave van de schaal en de plaatsing van het netwerk.



Figuur 1 Vereenvoudigde weergave van verbindingen binnen het OTN, op basis van alle fysieke verbindingen die in de netwerkadministratie bekend zijn. De data die gevisualiseerd is kan gebruikt worden om uit te lezen hoeveel verbindingen een fysieke locatie heeft, geaggregeerd naar plaatsniveau. De afbeelding telt in totaal 874 plaatsen en 61679 verbindingen.

2.2. Databeschrijving

KPN werkt met meerdere met elkaar verbonden en van elkaar afhankelijke systemen om het netwerk te beheren. De data die daaruit voortvloeit, welke is gebruikt voor dit onderzoek, bevat dus ook diezelfde afhankelijkheden. Voor de incidenten en de afhandeling is er sprake van een gelaagde relatie in de data, zichtbaar gemaakt in Figuur 2.

De nodes in het netwerk produceren managementinformatie, welke verzameld en verwerkt wordt door het U2000 Netwerk Management Systeem. Bij waardes die een vooraf bepaalde grenswaarde overstijgen worden events geregistreerd in U2000. Een deel van deze events, de meest relevante voor bewaking, worden doorgegeven aan *Netcool*, de centrale bewakingsapplicatie waarin events vanuit de verschillende netwerken en dienstenplatformen samen komen, waarin ze worden verwerkt en geregistreerd.

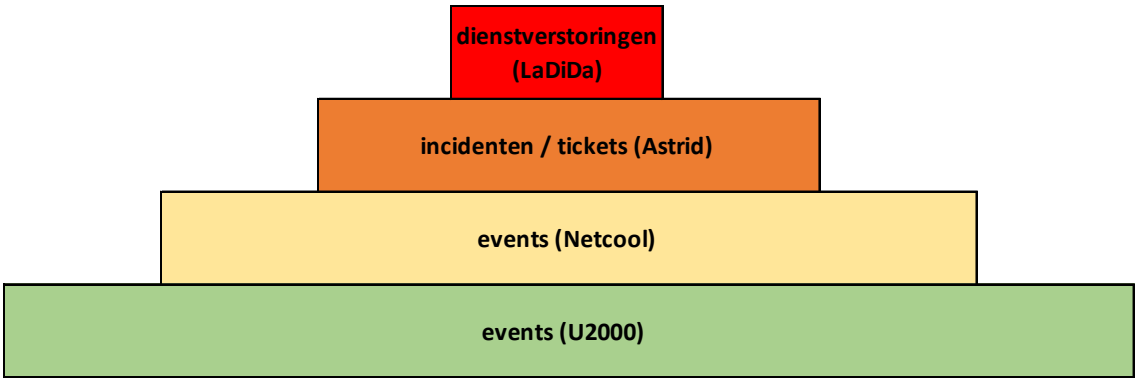
Als engineers van KPN event onderzoeken zal hier een ticket voor worden aangemaakt in het ticketingsysteem *Astrid*. In een ticket worden de probleemoplossende werkzaamheden bijgehouden horend bij een incident. Indien de dienstimpact groot is, dat is als er meer dan 64 gebruikers dupe zijn geworden van het incident, wordt er tevens een *LaDiDa*-event (Landelijke Dienstverstorings Data) geregistreerd. Deze events zijn ten behoeve van KPN's interne communicatie over belangrijke verstoringen. Bij een LaDiDa-event worden nog aanvullende maatregelen getroffen, zoals, indien nodig, een persvoorlichting.

Er is sprake van een constante afname van data (en dus informatie) voor elk niveau dat er omhoog gegaan wordt vanaf de U2000 event data, voor zolang er een koppeling wordt behouden. De visualisatie in Figuur 2 dient dan ook voor de beeldvorming hoe er naarmate er gefilterd wordt op de OTN, er minder overblijft, zo wordt er gefilterd vanaf de Netcool gegevens die betrekking hebben op het OTN. Dit betekent dat een opbouw van data enkel vanaf een kant opgemaakt kan worden.

In de U2000-eventdata is de grootste hoeveelheid data aanwezig, waarvan slechts een fractie zorgt voor events in Netcool, wat ook geldt voor elke keer dat een event wordt doorgegeven naar het volgende systeem. Van de U2000 events eindigt daarmee minder dan 0.1% als LaDiDa-registratie.

Hoewel de door U2000 geregistreerde events vermoedelijk erg waardevol is voor het kunnen voorspellen van een incident, wordt deze data niet altijd opgeslagen voor lange periodes. Opslaan gebeurt bij doelgerichte onderzoeken, om datavolumes te beperken.

In paragrafen 2.2.1 tot en met 2.2.4 zijn de datums en de hoeveelheid records aangegeven die inzicht geven aan het volume van de voor dit onderzoek beschikbare data. Wat er nog overblijft en hoe dit in verhouding staat met de hiërarchische structuur zoals weergeven in Figuur 2 wordt uitgelegd in het hoofdstuk Methodologie. Omdat deze bronnen weergeven worden voordat het koppelen van de data data heeft plaatsgevonden, zijn de aantallen en de periodes niet overeenkomend. Ter aanvulling van de door KPN aangeleverde bronnen wordt er ook gebruik gemaakt van de data van het KNMI.



Figuur 2. Hiërarchie databronnen OTN incidenten

2.2.1. Performance Data (U2000)

Elk component in het OTN produceert prestatiegegevens, zoals bijvoorbeeld de temperatuur van de apparatuur. Op basis van deze prestatiegegevens worden er events aangemaakt als er een bepaalde grenswaarde wordt bereikt, zoals bijvoorbeeld: temperatuur van component X is hoger dan de opgegeven grenswaarde. Op basis van een dergelijk event kan er overgegaan worden tot actie, door bijvoorbeeld direct onderhoud te plegen aan het betreffende component. Van de enorme (meerdere terabytes per dag) hoeveelheden geproduceerde prestatiegegevens wordt er weinig opgeslagen. Zoals al aangegeven in de vorige paragraaf wordt enkel in het geval dat er onderzoek wordt gedaan naar een specifiek aantal nodes prestatiesgegevens opgeslagen van die nodes. Zodoende zijn de beschikbare prestatiesgegevens te beperkt om te gebruiken in dit onderzoek.

2.2.2. Netcool

Netcool is het platform waar een deel van alle events, enkel de events die een grenswaarde passeren, van het KPN-netwerk op binnenkomen, zo ook van andere onderdelen naast OTN. Zodoende is de Netcool database een zeer omvangrijke database met meerdere tabellen met elk miljarden records aan eventregistraties. Er een enkele tabel relevant voor het verkrijgen van inzicht over events, dit betreft de *reporter status*-tabel. De Netcool gegevens zullen centraal staan aan het koppelen en elke beschikbare databron zal gekoppeld worden met de relevante record. Omdat er echter maar een klein deel (minder dan 1%) hiervan betrekking heeft op het OTN, wordt er voor dit onderzoek gefilterd op de voor OTN relevante delen. Deze *reporter status*-tabel heeft de volgende eigenschappen:

- Periode: November 2017 t/m februari 2019
- Een record: Bevat een event over een overschreden grenswaarde in een node.
- Aantal records: meer dan 1B waarvan ~3.1M relevant voor het OTN
- Aantal kolommen: 165

2.2.3. Astrid

Astrid wordt gebruikt door vrijwel alle afdelingen binnen KPN en heeft zodoende ook data voor vrijwel alle afdelingen binnen KPN. Gevolg hiervan is dat er maar een deel (minder dan 1%) van de tickets in het systeem relevant zijn voor het OTN. Ook in dit systeem is er een enkele database-tabel relevant voor het verkrijgen van inzicht over de incidenten, de *incident*-tabel. Deze tickets zijn te koppelen aan de events van Netcool op basis van een Ticket-ID en deze tabel heeft de volgende eigenschappen:

- Periode: Maart 2017 t/m februari 2019

- Een record: Bevat een ticket dat is behandeld met gedetailleerde informatie over het ticket.
- Aantal records: ~4.5M waarvan minder dan 1K voor het OTN
- Aantal kolommen: 146

2.2.4. LaDiDa

Een deel van de LaDiDa events kunnen worden gekoppeld met zowel Netcool als Astrid. Het koppelen gebeurt op basis van het LaDiDa ID-nummer. Het totale LaDiDa bestand heeft de volgende eigenschappen:

- Periode: Januari 2017 t/m februari 2019
- Een record: Bevat een ticket dat is behandeld met aanvullende informatie van een ticket en welke acties zijn genomen en welke gebieden zijn getroffen.
- Aantal records: 6354 waarvan 20 betrekking hebben op het OTN.
- Aantal kolommen: 16

2.2.5. WDM Trails

WDM Trails bevat de fysieke verbindingen en koppelingen, inclusief bijbehorende datastroom richtingen, tussen nodes. Het koppelen van deze databron met de Netcool dataset kan op basis van de datum en Node Alias: de naam van een Node. Deze dataset wordt gebruikt om te onderzoeken hoe met elkaar verbonden nodes elkaar beïnvloeden.

- Periode: 2012 t/m 2019
- Een record: Bevat de oorsprong en doel van een verbinding en of deze in gebruik is.
- Aantal records: ~65K
- Aantal kolommen: 6

2.2.6. KNMI

Om te onderzoeken of weersomstandigheden gecorreleerd zijn aan het ontstaan van incidenten gecorreleerd is een dataset van het KNMI gebruikt met daarin het weer van elke dag vanaf 1990 voor elk KNMI-weerstation in Nederland. Deze data wordt gekoppeld op geografische nabijheid van een node tot een weerstation, waarmee het weer als factor wordt meegenomen in het onderzoek.

- Periode: 1990 t/m 2019
- Een record: Bevat het weer van een weerstation van een dag.
- Aantal records: ~477K
- Aantal kolommen: 41

Een aantal van beschikbare databronnen heeft een zeer grote hoeveelheid records en kolommen wat dataverwerking uitdagend maakt. In het hoofdstuk 4.2. Datamodel wordt dieper ingegaan op de inhoudelijke relatie tussen de bronnen en hoe deze geschoond en gekoppeld zijn. Het doel is om elk van de beschikbare bronnen te verwerken zodat het model zoveel mogelijk informatie heeft om te gebruiken.

Een opmerkelijk detail aan de databronnen is dat voor de meest cruciale databron, Netcool, er alleen data beschikbaar is vanaf November 2017 t/m februari 2019. Dit betekent dat, hoewel er een langere periode aan data in Astrid en LaDiDa aanwezig is, deze nooit gekoppeld kan worden met de originele events in Netcool die de oorzaak vormden voor deze tickets.

Op basis van dit eerste aanzicht lijkt het alsof er voldoende data aanwezig is voor het trainen van een voorspellingsmodel. Er is voor een groot deel van het netwerk meerdere events inzichtelijk en voor Netcool zijn er meer dan 3M regels aan data te gebruiken voor het model. Of de conclusie getrokken kan worden of dit dan ook echt voldoende data is moet blijken uit de prestaties van het model. Nu duidelijk is welke data beschikbaar is, wordt de manier waarop de data is verwerkt verder toegelicht in Hoofdstuk 4: Methodologie.

3. Theoretisch Kader

Dit hoofdstuk beschrijft het onderzoek dat is uitgevoerd naar de mogelijke modellen om incidenten te voorspellen. Er zal dan ook antwoord worden gegeven op de tweede deelvraag: *Welke modellen komen in aanmerking voor het voorspellen van incidenten op basis van historische gegevens?* In de literatuur zijn verschillende modellen beschreven die een vergelijkbare onderzoeksvraag proberen te beantwoorden. Het literatuuronderzoek geeft tevens een beschrijving van de werking van de modellen en hoe deze een mate van voorspelbaar vermogen in zich hebben.

3.1. Voorspelbaarheid incidenten in het netwerk

De incidenten in de netwerken worden geacht voorspelbaar te zijn omdat er in veel gevallen van de incidenten reeds al meerdere, minder urgente (minor / warning), events zijn geregistreerd. Niet elk incident kan voorspeld worden op basis van voorgaande events maar de verwachting is dat er een groot deel wél voorspeld kan worden. Aanvullend zijn eigenschappen zoals het weer en de leeftijd van de apparatuur waarschijnlijk van invloed op de kans van incidenten. De mate van voorspelbaarheid wordt getoetst in het hoofdstuk Modelleren en Resultaten.

3.2. Modellen in de literatuur

In deze paragraaf worden er een aantal modellen uitgelicht die van toepassing kunnen zijn voor het voorspellen van de incidenten, op basis van vergelijkbare onderzoeken in de literatuur. Tevens wordt er per model gekeken naar hoe de voorspelling van een model geëvalueerd kan worden.

3.2.1. Markov-modellen

In (Hossain, 2017) en (Chan, 2005) wordt er met succes gebruik gemaakt van een Markov-model om een vergelijkbaar vraagstuk te beantwoorden. Het doel van het Markov-model in dit literatuuronderzoek is tweeledig, enerzijds om te kunnen voorspellen wanneer een node in het netwerk een incident veroorzaakt; anderzijds om inzicht te geven in de toestand-overgangen. Deze inzichten kunnen gebruikt worden door KPN om beslissingen te nemen over het al dan niet actie ondernemen.

Om meer uitleg te geven over een Markov-model wordt in dit hoofdstuk dieper ingegaan over twee varianten die gebruikt kunnen worden voor dit onderzoek, namelijk de Markov-keten en het Hidden Markov Model. In dit onderzoek wordt er primair dieper ingegaan op de Markov-keten.

Een groot onderscheid tussen elk van de Markov-modellen is of deze *continu* of *discreet* zijn. In het geval dat er sprake is van een continu-Markov-model wordt dit een Markov-proces genoemd. Een Markov-keten is enkel discreet. Het onderscheid van continu en discreet vertaalt zich naar de soorten toestanden die een Markov-keten kan aannemen, als deze toestanden een continue waarde hebben dan is er sprake van een continu-Markov-model. In dit verslag wordt enkel naar de discrete Markov-modellen gekeken om te modelleren welke alertgroups elkaar opvolgen. De alertgroups die er beschikbaar zijn, zijn eindig en bekend, namelijk de 182 alertgroups.

3.2.1.1. Markov-keten

Een Markov-keten is de meest eenvoudige variant van de Markov-modellen. Het is een stochastisch systeem is waarin elke mogelijke toestand van een systeem is vastgelegd. De kans op een volgende toestand kan uitgelezen worden door enkel naar de huidige toestand van het systeem te kijken. Dit betekent dat een Markov-keten een geheugenloos model is, wat betekent dat er met het bepalen van de opvolgende toestand geen rekening gehouden wordt met voorgaande toestanden. In een discrete tijdvariant is een Markov-keten de volgende vergelijking.

Stel een sequentie van willekeurige variabelen $x_1, x_2, \ldots, x_n$ met de Markov-eigenschap, namelijk dat de kans om naar de volgende toestand over te gaan afhankelijk is van enkel de huidige toestand. De afhankelijke kansen zijn als volgt gedefinieerd:

$$\mathrm{P}(X_{n+1} = x \mid X_n = x_n, X_{n-1} = x_{n-1}, \ldots, X_1 = x_1) = \mathrm{P}(X_{n+1} = x \mid X_n = x_n)$$

Met $X_i \in S$, waar S de eindige set van mogelijke waarden van X_i vormt. Deze vergelijking wordt doorgaans weergeven als een verbonden graaf, maar kan ook gepresenteerd worden door een overgangsmatrix van tijd n naar tijd $n + 1$.

De overgangsmatrix kan gebruikt worden voor het verkrijgen van inzichten, zoals gedaan in (Hossain, 2017) en (Reichel, 2014). In (Reichel, 2014) wordt uitgezocht hoe er inzichten gehaald kunnen worden uit Markov-ketens met een grote hoeveelheid toestanden. Omdat er in (Hossain, 2017) opzettelijk beperkt is op het aantal toestanden vanwege de lange rekentijden, wordt het in de methodologie onderzocht en vastgesteld in hoeverre het reduceren van de toestanden noodzakelijk is in dit onderzoek.

Indien het niet noodzakelijk is kan ervoor gekozen worden om gebruik te maken van de methodes die zijn voorgesteld in (Reichel, 2014). Hier wordt onder andere gebruik gemaakt van het kortste pad algoritme van Dijkstra's door de transitiematrix negatief logaritmisch te transformeren en zo het meest waarschijnlijke pad te vinden in de Markov-keten. Dit overzicht aan meest waarschijnlijke paden kan vervolgens gebruikt worden om vast te stellen welke toestanden elkaar opvolgen. Deze paden dienen voornamelijk als inzicht en worden gebruikt de derde deelvraag te beantwoorden: *Hoe ontwikkelt een foutmelding zich binnen het netwerk van KPN.* Deze aanpak wordt verder uitgewerkt in hoofdstuk 5.1. Markov-keten.

3.2.1.2. Hidden Markov Model

Het *Hidden Markov model* is een uitbreiding op de Markov-keten maar met de aanname dat er niet geobserveerde toestanden aanwezig zijn in het systeem. Vanwege de toevoeging van verborgen toestanden komen er een aantal variabelen bij. Ondanks de extra complexiteitslaag is een Hidden Markov Model nog steeds beknopt te omschrijven (Jurafsky, 2018):

T	aantal observaties,
$Q = q_1, q_2, \dots, q_N$	set van N toestanden,
$A = a_{11}, \dots, a_{ij}, \dots, a_{NN}$	overgangskans A van toestand i naar toestand j met $\sum_{j=1}^{N} a_{i,j} = 1 \ \forall i$,
$\mathrm{O} = \mathrm{o}_1, \mathrm{o}_2, \dots, \mathrm{o}_\mathrm{T}$	sequentie van T observaties getrokken uit een collectie van observaties $V = v_1, v_2, \dots, v_n$,
$B = b_i(o_t)$	een sequentie van emissie kansen die de kans van observatie o_t gegenereerd door toestand i,
$\pi = \pi_1, \pi_2, \dots, \pi_N$	Een initiële kansverdeling voor elke toestand waar π_i de kans uitdrukt dat de Markov-keten begint in toestand i. Het is mogelijk dat toestand j een kans van $\pi_j = 0$ heeft, wat betekent dat j geen initiële toestand kan zijn. Er geldt $\sum_{i=1}^{n} \pi_i = 1$.

De initiële distributie π_N is in bovengenoemde vergelijkingsstelsel nog niet gegeven en dient bij het opstellen van het model te worden vastgelegd. Een van de nieuwe eigenschappen aan het Hidden Markov Model is de emissieparameter. Dit is een parameter die de kans van toestand wisseling voor verborgen toestanden weergeeft. Het is hiermee echter nog niet duidelijk wat de precieze waarde moet zijn van de emissieparameters, hoeveel verborgen toestanden er aanwezig moeten zijn en wat de verdere waardes van de parameters van het model moeten zijn.

Het vaststellen van de optimale hoeveelheid verborgen toestanden kan gedaan worden met behulp van het Viterbi-algoritme (Jurafsky, 2018). Dit is een Dynamic Programming-algoritme en zoekt het Viterbi Path: het pad met de meest waarschijnlijke sequentie aan verborgen toestanden.

Met behulp van het Baum-Welch algoritme kan op basis van de sequenties in de data de overgangs- en emissie waardes uitgerekend worden. Het is dus mogelijk om met behulp van zowel het Baum-Welch-algoritme, als het Viterbi-algoritme alle benodigde eigenschappen van het Hidden Markov Model uit te rekenen (Jurafsky, 2018).

Het voorspellen van incidenten kan op verschillende manieren, maar juist Hidden Markov Modellen zijn hiervoor geschikt, gezien verwacht wordt dat een component die meerdere lage severity events produceert waarschijnlijk uiteindelijk een hoog severity event produceert. Ook kan verwacht worden dat een bepaalde sequentie aan laag severity events, ofwel bepaalde alertgroups, vaker zullen leiden tot een alertgroup die weer leidt tot een incident. Vanwege de sequentiële aard van de probleemstelling en de successen die Markov Modellen hebben op dit gebied, zou juist het Markov-model goed in staat moeten zijn om het volgende incident te kunnen voorspellen (Salfner, 2019). Ook is het zo dat de eigenschap van verborgen toestanden in de Hidden Markov Modellen goed aansluiten bij de aard van het voorspellen van incidenten, omdat het eigenlijke falen nooit direct zichtbaar is, maar enkel afleidbaar is uit het event dat het falen veroorzaakt. Faalsequenties zijn bijna nooit identiek waardoor het wenselijk is om dit probabilistisch te benaderen (Salfner, 2007).

Een nadeel aan Hidden Markov Modellen is dat zij zich niet eenvoudig lenen tot het uitbreiden van aanvullende informatie. Zo kan bijvoorbeeld het weer ten tijde van een incident niet meegenomen worden zonder het model substantieel uit te breiden (Wang, 2014).

3.2.2. Survival modellen

Een alternatieve aanpak naast Markov-modellen is om de probleemstelling te formuleren als een survivalanalysevraagstuk, zoals in (Matz, 2002). Dit betekent dat er gebruik gemaakt wordt van de modellen en verdelingen die bekend zijn in het survivalanalyse-domein. Het survivalanalyse-domein houdt zich bezig met het bepalen van de verwachte tijdsduur en kans dat een gebeurtenis plaatsvindt. Deze gebeurtenis is bijvoorbeeld het ontstaan van een incident of, veelal gebruikt in het medische domein, dat er een persoon ziek wordt. Een groot voordeel van survivalmodellen, ten opzichte van de Hidden Markov Modellen, is dat deze modellen het mogelijk maken om meerdere variabelen te verwerken in het model (Pölsterl, 2015, 2016).

De eigenschappen van survivalanalyse geeft enerzijds een goed model de mogelijkheid om bijvoorbeeld een onderhoudsplanning af te stemmen, anderzijds ook een indicatie van faalrisico.

Een ander voordeel van survivalmodellen is dat er veelal rekening gehouden wordt met *censored data*. Er is sprake van censored data als de verwachte gebeurtenis niet heeft plaatsgevonden ten tijde van de observatie van de proef, waar de proef in de context van KPN de gehele observatietijd van een node betreft.

In de dataset zoals gegeven door KPN is er sprake van *right-censored data*. Dit betekent dat de observatietijd eindig en korter is geweest dan de overlevingstermijn van alle deelnemers van de proef. Ofwel, in de dataset van KPN (die een tijdspan van november 2017 tot en met februari 2019 betreft) zijn er nodes die niet gefaald hebben. Deze nodes zijn volgens deze definitie right-censored.

In de survivalanalyse worden meerdere functies veel gebruikt. In onderstaande secties worden deze uiteengezet.

3.2.2.1. Survivalfunctie

De kern van de survival analyse is de Survivalfunctie, welke op basis van een gegeven tijd t de kans $S(t)$ berekent dat een event plaatsvindt na tijdstip t.

De overlevingsfunctie wordt geformuleerd aan de hand van de vergelijking (Rodriguez, 2007):

$$S(t) = \Pr(T \geq t)$$

Een eis van deze vergelijking is dat de functie niet-stijgend is, namelijk: $S(u) \leq S(t)$ waar $u \geq t$. In veel overlevingsfuncties geldt dat $S(t) \to 0$ met $t \to \infty$. Het eenvoudigste model dat een survivalfunctie kan geven op gecensureerde data is de *Kaplan-Meier estimator* (Kaplan & Meier, 1958) Deze non-parametrische estimator maakt het mogelijk om op basis van enkel de overlevingstijden en de eigenschap of deze gecensored zijn een survival functie op te stellen. Dit is tevens het nadeel van deze estimator. Mogelijke andere factoren die invloed kunnen hebben op de overlevingstijden, zoals bijvoorbeeld het weer, kunnen niet meegenomen worden in het model door de restrictie op deze enkele variabele. Om ervoor te zorgen dat deze variabelen wel gebruikt kunnen worden voor het verkrijgen van een survival functie kan het Cox Proportional Hazard (Cox PH) model gebruikt worden.

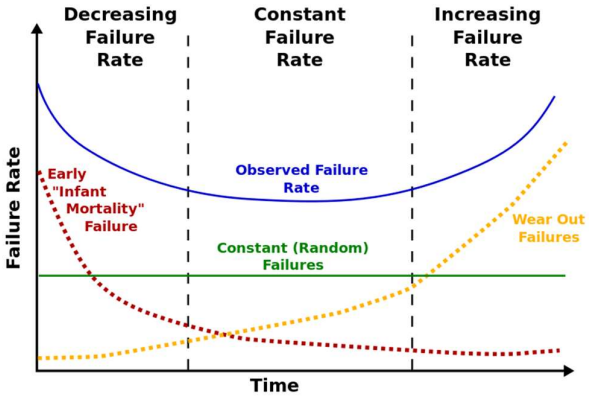
3.2.2.2. Hazard Function

De hazard function geeft aan wat de kans is dat een node het niet overleeft voor tijd $t + dt$, gegeven dat deze het voor tijd t heeft overleeft.

De hazard function kan als volgt worden geformuleerd (Rodriguez, 2007):

$$h(t) = \lim_{dt \to 0} \frac{\Pr(t \leq T < t + dt \mid T \geq t)}{dt * S(t)} = \frac{f(t)}{S(t)} = \frac{S'(t)}{S(t)}$$

Om te berekenen hoe de hazard rate zich ontwikkelt over tijd wordt de Nelson-Aalen estimator gebruikt. Deze krommes zijn goed uit te lezen in de context van de Bathtub curve (Klutke, 2003) en weergeven in figuur 3. Een kromme die snel daalt geeft aan dat er sprake is van vroegtijdig sterven, ofwel "Infant Mortality". Dit betekent in de context van dit onderzoek dat apparaten aan het begin van hun levensduur veel incidenten produceren en dit afneemt naarmate deze levensduur hoger wordt.



Figuur 3 Bathtube curve varianten. Bron: wikipedia.org

Het aantal dagen dat een node zonder incident heeft gewerkt wordt beschouwd als het leven van een node. De verwachting is dat een node een kromme als de "Infant Mortality" volgt. Een node start na herstel weer aan een nieuw 'leven', en dan is het waarschijnlijk dat er nog enkele foutjes zitten. Zodoende wordt het verwacht dat een node een kromme als de "Infant Mortality" volgt.

3.2.2.3. Survival Decision Trees en regression

In deze paragraaf worden de survival modellen toegelicht die gebaseerd zijn op regressie en gebruik maken van decision trees. Een decision tree is een model waar meerdere beslissingen achtereen zorgen voor de uiteindelijk afgegeven waarde. Een gedetailleerde uitleg volgt in de paragraaf Classificatie & Regressie trees.

3.2.2.3.1. Cox Proportional Hazard Regressie analyse

In 1972 heeft (Cox, 1972) het Cox Proportional Hazard (Cox PH) model voorgesteld. Dit model heeft de assumptie dat bij elke numerieke wijziging van een "covariate", de hazard, proportioneel toeneemt, of daalt. Een covariate in deze context is een variabele die invloed uitoefent op de doelvariabele – de overlevingstijd.

Het Cox PH model is als volgt te formuleren: laat $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ de gerealiseerde waardes zijn van de covariaten voor onderwerp i. De hazard functie voor het Cox PH model heeft dan de volgende vorm:

$$h(t) = h_0(t) * e^{(b_1 x_1 + b_2 x_2 + \cdots + b_n x_p)}$$

waar

t = de overlevingstijd vertegenwoordigd

$h(t)$ = de hazard functie is die bepaalt wordt door een set van n covariaten $(x_1, x_2, \dots, x_n)$.

b_n = de coëfficiënten de impact van de covariaten bepalen

$h_0(t)$ = de baseline hazard is die verkregen kan worden door alle covariaten gelijk aan 0 te stellen.

Deze vergelijking geeft de hazard functie op tijdstip t voor voorwerp i met covariate vector X_i. De coëfficiënten van de Cox Proportional Hazard worden verkregen door middel van Gradient Descent wat wordt verder toegelicht wordt in het hoofdstuk Ensemble Forest.

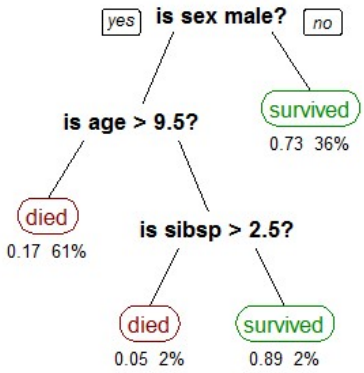
Cox Proportional Hazard wordt vaak uitgebreid met decision trees, waarvan de werking verder wordt uitgelegd in de paragraaf Classificatie en Regressie Trees. Dit leunt op de assumptie dat het mogelijk is om de data op zo'n wijze te scheiden dat een verzameling van lijnen, krommen of oppervlaktes voldoet. Dit is in de werkelijkheid vaak niet het geval en daarom is het ook interessant om te kijken naar andere technieken, zoals classificatie en regressie bomen – waar er gebruik gemaakt wordt van meerdere decision trees.

3.2.2.3.2. Hazard Ratio

De Hazard Ratio is een maatstaf die de waarde van de hazard rate afzet tegen de baseline hazard rate. De baseline Hazard Rate is het gevaar waar alle covariaten gelijk aan het totale gemiddelde van de respectievelijke covariate zijn. In de Cox PH vergelijking wordt de Hazard Ratio uitgedrukt door de term $e^{(b_1x_1+b_2x_2+\cdots+b_nx_p)}$.

3.2.2.3.3. Classificatie & Regressie Trees

Het klassieke Classificatie en Regressie Trees model, ook wel C&RT genoemd, is gepopulariseerd door (Breiman, 1984). Deze classificatie en regressie modellen werken op basis van een decision tree, welke bestaat uit nodes, taken en bladeren en wordt opgebouwd door het maximaliseren van de informatie toename.. Het blad zit altijd aan het einde van een reeks takken in de decision tree.



Figuur 4. Voorbeeld illustratie van een decision tree. Deze begint bovenin bij de node waarin de variabele 'sex' wordt geëvalueerd op de waarde 'yes' of 'no'. In het geval dat de variabele de waarde 'No' heeft zal er bij het blad 'Survived' geëindigd worden. Bij de waarde 'Yes' zal de boom verder afgereisd worden via de taken tot het datapunt bij een ander blad eindigt.

Als een decision tree gevormd is heeft elke node een bepaalde grenswaarde voor een variabele. Deze grenswaarde bepaalt de verdere route door de boom, door de waarde van het datapunt te vergelijken met deze grenswaarde. In het geval dat de waarde van het datapunt groter of gelijk is aan de waarde in de node, wordt rechtsaf geslagen, anders linksaf. Onder een vertakking zit of een node, of een blad. Bij een node zal het proces zich herhalen op dezelfde of een andere variabele en zal er wederom naar links of rechts uitgeweken worden. In het geval dat het datapunt bij een blad uitkomt, wordt de uiteindelijk voorspelde waarde teruggegeven.

Bij het maken van een decision tree, moet vastgesteld worden waar de vertakkingen komen. Het berekenen hoe goed een split is wordt vaak met de *Gini-Index* en *Information Gain* gedaan. Het is gebruikelijk dat deze afwisselend gebruikt worden (Ramawadh, 2018).

De Gini-Index en Information Gain worden als volgt gedefinieerd:

Gini-Index: $G(\varphi) = 1 - \sum_{i=1}^{k} p_i^2$

Information Gain: $H(\varphi) = -\sum_{i=1}^{k} p_i \log_2 p_i$

waar

φ = een verzameling objecten is die elk over een attribuut A beschikken en $A \in \mathbb{R}$.

p_i = een fractie van alle objecten φ is waarvan attribuut A de waarde i heeft en k het aantal unieke waardes is die een attribuut A kan aannemen.

Om vervolgens de boom te maken wordt de data opgedeeld in meerdere partities. Voor de Gini-Index en Information Gain van een partitie ziet de vergelijking er als volgt uit:

Ter aanvulling van de voorgaande definitie kan hier aangenomen worden dat $\mathbb{P} = \{P_1, \dots, P_n\}$van φ, waar P een partitie, n het aantal waarnemingen en m het aantal variabelen is.

Gini Index: $$G(\mathbb{P}) = \frac{1}{|\varphi|}\sum_{i=1}^{m}|P_i| * G(P_i)$$

Information Gain: $$H(\mathbb{P}) = \frac{1}{|\varphi|}\sum_{i=1}^{m}|P_i| * H(P_i)$$

Door iteratief voor een partitie de Gini-index te maximaliseren kan er besloten worden voor welke variabele, op welke waarde een tak wordt aangemaakt. Vervolgens wordt er of de toename van de Gini-Index voldoende is om nog verder af te splitsen in meer takken. Grofweg ziet het proces er als volgt uit (Ramawadh, 2018):

$$\mathbb{P}_{i,z} = \{\{X_{i,\cdot} \in C \ : X_{i,j} < z\}, \{X_{i,\cdot} \in C \ : X_{i,j} \geq z\}\}$$

Waar $\mathbb{P} = \{P_1, \dots, P_n\}$ een partitie voorstelt en C de cel is die opgedeeld dient te worden, en $X_{\cdot,j}$,een variabele. De volgende splitsing wordt bepaald door de partitie te selecteren met de maximale Gini-index of Information Gain. Wat een partitie precies oplevert kan bepaald worden met de volgende formule:

$$\Delta G(C) = G(C) - G_{X_{\cdot,j}}(C)$$

waarbij $G_{X_{\cdot,j}}(C)$ de Gini index van de partitie van de cel C op basis van $X_{\cdot,j}$ is, of bij het gebruik van entropie waarbij $H_{X_{\cdot,j}}(C)$ de Entropie van de partities van de cel C en $\Delta H(C)$ de Information Gain.

$$\Delta H(C) = H(C) - H_{X_{\cdot,j}}(C)$$

Door deze splits te maken is het mogelijk om een classificatie of regressie model te fitten op de data. Dit blijft echter beperkt tot een enkele boom en vereist dan ook dat de data goed gescheiden kan worden met lijnen. Omdat dit niet altijd mogelijk is, worden ensembles gebruikt: samenvoegingen van meerdere modellen. Het idee achter een ensemble is dat meerdere gespecialiseerde modellen gemiddeld genomen beter presteren dan een enkel algemeen model. Een voorbeeld hiervan is de Ensemble Forest dat in het volgende hoofdstuk wordt uitgelegd.

3.2.2.3.4. Ensemble Forest

Een ensemble forest is een combinatie van meerdere unieke bomen die elk op een andere manier gefit zijn op de data. Decision Forests presteren goed, en zijn doorontwikkeld tot vele varianten (Li, 2010). Een van de bekendste modellen in dit domein zijn de Random Forests (James et al. 2013) en Gradient Boosting. Vooral Gradient Boosting heeft in de literatuur goed gepresteerd op een aantal benchmarks (Li, 2010) en zal daarom centraal staan in de rest van deze sectie.

Gradient Boosting is een techniek die is voorgesteld door (Kearns, 1988) en (Valiant et al. 1989) en is ontwikkeld door (Schapire, 1990). Gradient Boosting bestaat uit het principe van meerdere zwakke modellen combineren tot een sterk model. Een zwak model in deze context is een model dat iets beter kan voorspellen dan willekeurig gokken. Door de nauwkeurigheid van elk zwak model te evalueren over de gehele dataset en dit gewicht mee te nemen is het mogelijk om de modellen te combineren tot een sterk model.

Om het proces van het trainen van de zwakke modellen zo snel mogelijk te maken is het algoritme Extreme Gradient Boosting (XGBoost) ontwikkeld (Tianqi et al. 2016). Dit is een implementatie van het Gradient Boosting Algoritme met een aantal optimalisaties waardoor deze parallel en out-of-core, buiten het werkgeheugen, gefit kan worden. Door deze implementatie is het mogelijk om met zeer grote datasets en met beperkte middelen alsnog de gehele dataset te gebruiken voor het trainen van een model. De keuze is dan ook op dit model gevallen omdat deze goed presteert en schaalbaar is naar de grote datasets zoals aangeleverd door KPN.

Bij het uitvoeren van Gradient Boosting wordt er een doelfunctie geoptimaliseerd die in de meeste gevallen een Mean Square Error is:

$$L\big(y, F(x)\big) = \frac{1}{n}\sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

waar

$\hat{y} = F(x)$, de voorspelde waarde

F = een model die een waarde voor x probeert te voorspellen

n = het aantal voorspellingen

y_i = de werkelijke waarde

Door elke werkelijke y_i van de voorspelde waarde $\hat{y}_i$ af te trekken en dit verschil te kwadrateren kan de afwijking tussen het model F en de werkelijkheid berekend worden.

Omdat er uit het enkele model $F(x)$ een verschil overblijft tussen de werkelijke en voorspelde waarde die gebruikt wordt om de Mean Square Error te berekenen, kan er vervolgens een nieuwe regressieboom $h(x)$ worden toegevoegd aan het model: $F(x) + h(x)$.. Mocht het zo zijn dat het nieuwe model $F(x) + h(x)$ nog een te hoge Mean Square Error produceert, dan kan er nog een nieuwe decision tree worden toegevoegd en dit wordt herhaald tot de Mean Square Error laag genoeg is.

Hoewel dit een veelgebruikte doelfunctie is in het Gradient Boosting algoritme, is dit niet de enige mogelijke doelfunctie. Zo is het ook mogelijk om de Cox Proportional Hazard als doelfunctie te gebruiken. Het optimaliseren van de doelfunctie wordt gedaan met behulp van Gradient Descent, waar een functie geminimaliseerd wordt door in de tegengestelde richting van de helling te bewegen. De standaard vergelijking van Gradient Descent ziet er als volgt uit:

$$\theta_i := \theta_i - p\frac{\partial J}{\partial \theta_I}$$

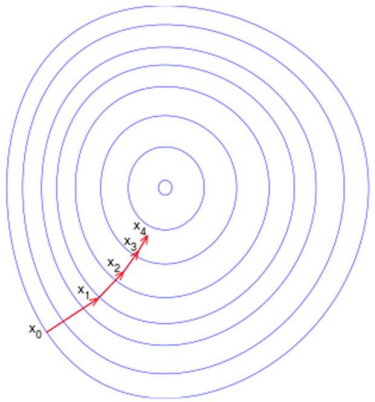
Waar θ en J twee variabelen zijn en p een coëfficiënt.

De Mean Square Error kan geminimaliseerd worden door de gradiënt te berekenen van de afgeleide van $h(x_i)$, waar $h(x_i)$ een decision tree is. Dit kan als volgt afgeleid worden Cheng, L. (2016):

$$F(x_i) := F(x_i) + h(x_i)$$

$$F(x_i) := F(x_i) + y_i - F(x_i)$$

$$F(x_i) := F(x_i) - \frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}$$



Figuur 5. Illustratie Gradient Descent (bron: wikipedia.org)

Op basis van bovenstaande vergelijkingen kan Gradient Boosting ervoor zorgen dat de foutmarge zo klein mogelijk blijft en het model dus optimaal gefit is op de data. In Figuur 5 is een voorbeeld illustratie van een aantal iteraties Gradient Descent weergeven. De doel functie zoals opgenomen in de bovenstaande vergelijking kan dan ook uitgewisseld worden met een andere doel functie zoals de Cox PH Regressor.

Het XGBoost model geeft bij een voorspelling een waarde in het interval $[0, \infty)$ waarvan de voorspelde waarde de hazard ratio vertegenwoordigd voor die set aan covariaten. Omdat de implementatie van de Cox Proportional Hazard regressor in het XGBoost model geen cumulatieve survival functie heeft, is het helaas niet mogelijk om op basis van het XGBoost algoritme een verwachte levensduur van een datapunt te verkrijgen.

3.3. Evaluatie voorspellingsnauwkeurigheid

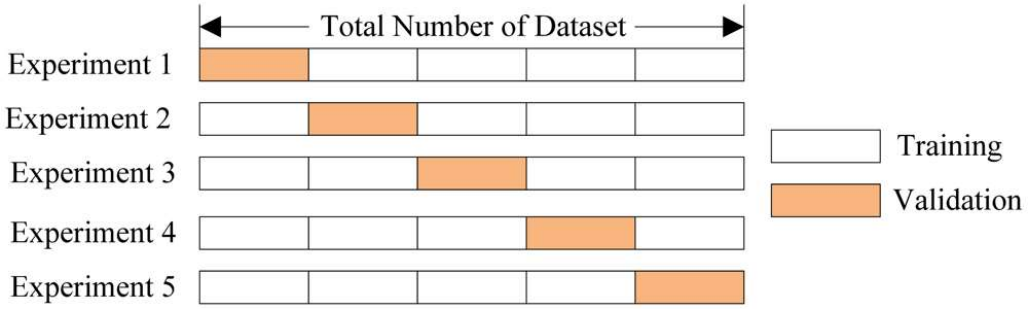
In dit hoofdstuk wordt ingegaan op de manier waarop de nauwkeurigheid van het model geëvalueerd kan worden. Omdat de survival modellen en Markov modellen verschillen qua output wordt er onderscheid gemaakt tussen de twee verschillende methodes.

3.3.1. Kruisvalidatie

Om te verifiëren of het model goed generaliseert, is het zaak om dit te evalueren. Een veel gebruikte methode is kruisvalidatie. Bij kruisvalidatie wordt de gehele dataset opgebroken in een aantal partities en wordt elke partitie uiteindelijk gebruikt ter evaluatie van het model. Dit gebeurt doorgaans kruislings.

In Figuur 6 is geïllustreerd hoe een vijfvoudige kruisvalidatie toegepast wordt op een dataset:

1) De totale dataset wordt in 5 delen geknipt
2) Voor het eerste experiment worden de laatste vier delen samengevoegd tot een training dataset, en het eerste deel wordt gebruikt als validatie set
3) Het model wordt getraind op de trainingsdata en doet een voorspelling over de validatie set
4) De voorspellingen op de validatie set worden afgezet tegen de werkelijke waarde om een score te berekenen.
5) In de volgende iteratie wordt een andere partitie gekozen als validatie set. Daarna herhaalt dit proces zich tot alle vijf de delen zijn getoetst.



Figuur 6. Vijfvoudige Kruisvalidatie

Een voorwaarde is dat de gehele dataset zuiver vertegenwoordigd wordt in elke partitie. Door de partities te stratificeren voldoe je aan deze voorwaarde.

3.3.2. Evaluatie Hidden Markov Model

De output van het Hidden Markov Model kan op verschillende manieren geëvalueerd (Salfner, 2007) worden. De uitkomst van het model bestaat uit een kans voor elke mogelijke toestand. De som van deze kansen is 1 en de hoogste kans van alle toestanden kan worden beschouwd als voorspelde waarde. De prestaties van de uitkomst van het model kan uitgedrukt worden in vier termen, hieronder weergegeven in de Confusion Matrix (Tabel 1):

Tabel 1 Confusion Matrix

	Waarheid		
Voorspeld		Positief gerealiseerd	Negatief gerealiseerd
	Positief voorspeld	True Positive (TP)	False Negative (FN)
	Negatief voorspeld	False Positive (FP)	True Negative (TN)

Een ideaal model bevat enkel de True Positives en True Negatives. In de praktijk komt dit bijna nooit voor, dit is namelijk vrijwel onmogelijk te realiseren, er zullen vrijwel altijd False Positives en False Negatives aanwezig zijn. Op basis van deze vier verschillende resultaten is het mogelijk om een aantal verschillende maatstaven, beschreven in de onderstaande alinea's, te berekenen die inzicht in de algemene prestaties van het model.

3.3.2.1. Nauwkeurigheid

De bekendste maatstaf is de nauwkeurigheid. De nauwkeurigheid is te berekenen met de formule:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

De nauwkeurigheid geeft een goede indicatie hoeveel datapunten goed zijn voorspeld, mits de data redelijk gelijk verdeeld is. In een situatie waar de data scheef verdeeld is kunnen er misleidende waarden uitkomen, zoals bijvoorbeeld in een data set waar 90% de waarde 0 heeft en de resterende 10% de waarde 1. Hier kan het voorspellen van enkel en alleen waarde 0 voor elk datapunt leiden tot een nauwkeurigheid van 90%. Dit is in het algemeen niet wenselijk omdat je zowel waarde 0 als waarde 1 goed wilt voorspellen.

3.3.2.2. Precision & Recall

De precision is een maatstaf die aangeeft wat de fractie van goed voorspelde waarde 1 is (True Positive) in vergelijking met alle voorspelde waarden. De recall geeft aan wat de fractie van goed voorspelde incidenten is ten opzichte van alle incidenten.

$$Precision = \frac{TP}{TP+F},\ Recall = \frac{TP}{TP+FN}$$

Over het algemeen is er sprake van een inverse relatie van de precision en recall. Waar het regelmatig voor kan komen dat de precision daalt als de recall toeneemt, en vice versa, is het ook mogelijk dat deze gezamenlijk hoog zijn, bijvoorbeeld in het geval van een zeer nauwkeurig model. Een goede combinatie tussen de twee scores is dan ook de F1-score.

3.3.2.3. F1-Score

Een meer gebruikte maatstaf is de F1-Score (Salfner, 2007). De F1-score is een combinatie van de Precision en Recall en wordt veel gebruikt voor optimalisatie, met name in asymmetrische datasets (Nan, 2012). Hiermee kun je de perfecte balans tussen precision en recall vinden. De F1-score is als volgt te berekenen:

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

3.3.2.4. Matthew Correlation Coefficient

Een andere maatstaf, de Matthew Correlation Coëfficiënt (MCC), is een maatstaf die in het domein van Machine Learning minder gebruikt wordt dan de F_1-score. Het is echter volgens (Boughorbel, S. 2017) een veel gebruikte maatstaf in de bio-informatica. Het waardevolle aan de MCC is dat deze een goede weergave geeft van de algemene correctheid, ongeacht de verdeling van de data. De MCC wordt gebruikt in de situatie waar een foutief positieve net zo kwalijk is als een foutief negatieve. De formule voor de MCC is als volgt:

$$Matthew\ Correlation\ Coefficient = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

3.3.3. Evaluatie Survival Model

In dit hoofdstuk wordt toegelicht hoe een continue event time voorspellend survival model kan worden geëvalueerd.

3.3.3.1. Harrell's Concordance index (C-Index)

Voor het evalueren van een Survival Model dat een continue event time voorspelt is een veel gebruikte maatstaf de Harrell's Concordance Index (Uno, 2011). Een paar waarnemingsparen (X_i, X_j) en (Y_i, Y_j) van stochastische variabelen wordt concordant genoemd als geldt $X_i < X_j$ en $Y_i < Y_j$ of $X_i > X_j$ en $Y_i > Y_j$ en waar i de node is. Dit betekent dat de Harrell's Concordance Index een score in het interval $[0, 1]$ opgeeft waar 1.0 een perfecte score is, 0.0 een absoluut foutieve score en 0.5 een score die gelijk staat aan willekeurige voorspelling. Harrell's Concordance Index wordt ook wel de C-Index genoemd en wordt vanaf nu op deze manier omschreven.

Een perfecte score betekent dat het relatieve gevaar dat is toegekend aan elke individuele entiteit, qua ordering perfect overeenkomt met de gerealiseerde overlevingstijden. Vanwege de inverse relatie tussen gevaar en overlevingstijd wordt er gesproken van concordance in de situatie dat de waardes zijn omgedraaid. Dus een hoog gevaar is toegekend aan een korte overlevingstijd, en een laag gevaar is toegekend aan een lange overlevingstijd. In onderstaande tabel is vastgelegd wanneer er sprake is van concordance. Waar X_n overlevingstijd (gecensureerd of ongecensureerd) is van een entiteit, en Y_n het voorspelde gevaar. In onderstaande vergelijking is de formule vastgelegd voor het verkrijgen van de C-index.

Tabel 2 Concordance tabel.

Concordant	Disconcordant
$X_i < X_j$ en $Y_j < Y_i$	$X_i < X_j$ en $Y_j > Y_i$
$X_i > X_j$ en $Y_j > Y_i$.	$X_i > X_j$ en $Y_j < Y_i$

$$C_{index} = \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\mathbf{1}\{0 < X_i < X_j \wedge \mathrm{Y_i} > Y_j\}}{\mathbf{1}\{0 < X_i < X_j\}}$$

Waar **1** de indicatorfunctie is, wat betekent dat als de stelling waar is binnen de {} er een waarde 1 wordt vastgelegd en als deze niet waar is, een waarde 0 wordt vastgelegd.

4. Methodologie

In de methodologie worden de gebruikte technieken, keuzes en aannames beschreven. Allereerst wordt er ingegaan op de verificatie van de data compleetheid. Vervolgens wordt er gekeken naar een manier om de databronnen met elkaar te koppelen en waar nodig te reinigen. De opzet van het model wordt na de analyse beschreven met daarbij een onderbouwing over de model keuzes.

4.1. Data verificatie

Het is van belang dat de data waar mee gewerkt wordt compleet is en zodoende geen vertekend beeld kan geven van de werkelijkheid. De databronnen gebruikt voor dit vraagstuk garanderen dit echter niet. Bijvoorbeeld de data van Netcool. Deze data komt uit het systeem Netcool Reporter, welke maar een opslagperiode van maximaal 100 dagen heeft. Om toch alle data te kunnen behouden, heeft KPN een koppeling gemaakt met het interne data platform: Universal Data Exchange Platform. Omdat deze opslag dagelijks gebeurt kan het gebeuren dat iets niet compleet of correct wordt opgeslagen of overgedragen. Hoewel de kans klein is dat dit gebeurt, is het alsnog van belang hiervoor te waken en dit te verifiëren.

Om de compleetheid van de data te verifiëren is er gekozen voor een tweetal aanpakken:

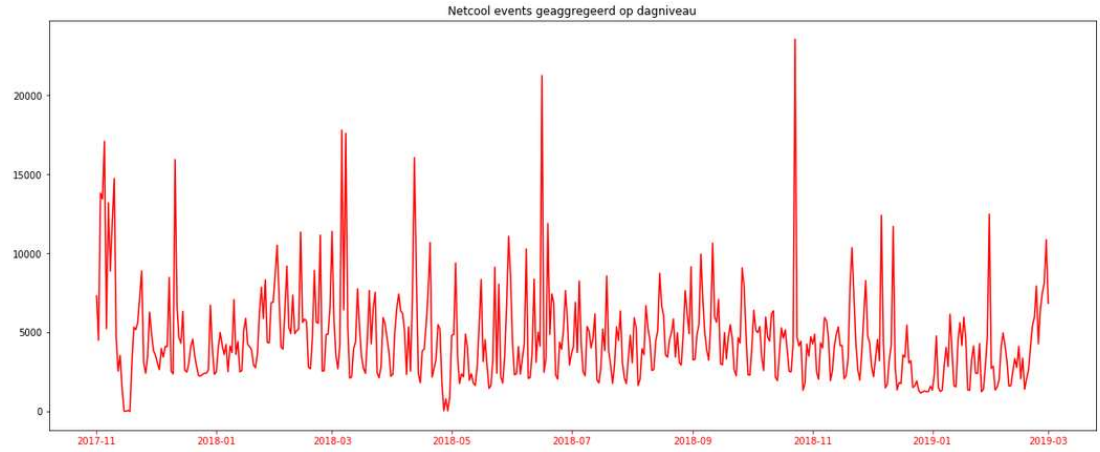
1) Er wordt gekeken naar rapporten in het verleden in hoeverre de hoeveelheden overeenkomen.
2) Er wordt er een visualisatie gemaakt van het aantal regels per dagdeel. Dit dient ter aanvullende controle op de eerste aanpak.

4.1.1. Verificatie met behulp van bestaande rapporten

KPN heeft in het verleden een reeks rapporten geproduceerd bestaande uit geaggregeerde data overzichten van het aantal events én tickets per node in een gegeven maand. Dit rapport kan gebruikt worden om te verifiëren of de overlap tussen de datasets kloppend is. Dit bleek de meest waardevolle verificatie methode omdat er uit deze controles naar voren kwam dat er kleine delen ontbreken. Na meerdere checks zijn de getallen overeenkomend in de gebruikte datasets en de rapporten van KPN.

4.1.2. Visuele verificatie op basis van detectie ontbrekende delen

Een visuele inspectie is een goede leidraad om te verifiëren of er geen grote delen ontbreken in de data. Door een aggregatie te maken op dagniveau van de modificatiedatum van een event kan een tijdreeks verkregen worden van het aantal events per dag per dataset.

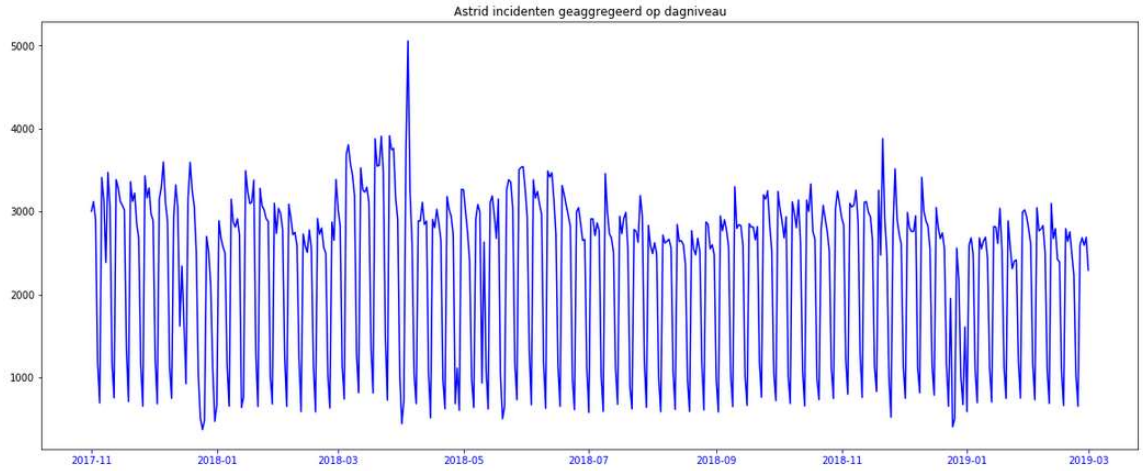


Figuur 7. Op dagniveau geaggregeerde events van Netcool voor de klasse 4401, 4406 en 4431. De Alarmklasse die het OTN dekken.

In Figuur 7 is een visualisatie te zien van alle Netcool events in de klasse 4401, 4406 en 4431, geaggregeer op dagniveau. In deze visualisatie is goed zichtbaar dat er geen dalen aanwezig zijn waar de lijn onderbroken wordt. Wel is het duidelijk dat de data een grillig karakter heeft, met sterke pieken en dalen. Een groot deel van deze dalen vindt plaats in het weekend. Omdat bijvoorbeeld het diepe dal rond de jaarwisseling van 2018 naar 2019 opvalt omdat het de 0 lijkt te benaderen, is hierop ingezoomd in figuur 8. Dit dal wordt verklaard door de “freezes” die KPN in de vakantieperiodes uitvoert om de kans op fouten te minimaliseren. Bij een freeze worden geen grote werkzaamheden gedaan aan het netwerk, en neemt de kans aanzienlijk af op fouten. Dit neemt echter niet het risico weg dat er alsnog fouten optreden. De periodische dalen kunnen verklaard worden doordat aanvullend geen werkzaamheden in het weekend verricht worden. De grote pieken blijken kabelwerkzaamheden in de nachtelijke uren, welke zich vertalen naar grote kabelstoringen.

Figuur 8. Netcool events op dagniveau tussen de periode van 1 december 2018 tot 10 januari 2019.

In Figuur 9 is een visualisatie te zien van alle tickets in Astrid geaggregeerd op dagniveau. Hoewel maar een fractie hiervan te koppelen is met de Netcool events voor het OTN, is het alsnog van belang om zeker te weten dat de gehele dataset compleet is alvorens deze verder te verwerken. Op basis van de visuele inspectie is er geen gat aanwezig. Het periodieke karakter van de data kan verklaard worden door de weekenden. In de weekenden zijn er namelijk aanzienlijk minder mensen aan het werk die tickets kunnen aanmaken.



Figuur 9. Alle Astrid tickets van november 2017 t/m februari 2019 voor elk domein.

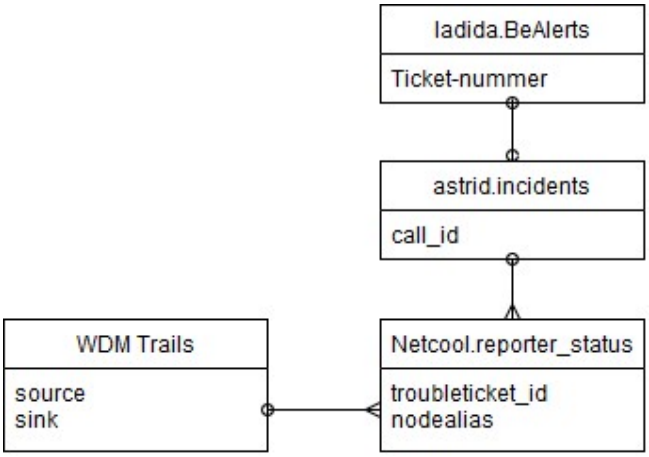
De WDM Trails behoeven geen verdere verificatie omdat deze direct uit het originele bronbestand komen die actueel bijgehouden wordt naar de daadwerkelijke staat van het netwerk.

4.2. Data model

In dit hoofdstuk wordt het gebruikte data model, de gebruikte data preparaties en de primaire dataset beschreven. De preparatie van de data voor het survival model en voor het Markov model zijn te vinden in de respectievelijke paragraven onder het hoofdstuk Modelleren omschreven.

4.2.1. Data koppelen

Om te kunnen onderzoeken welke tickets geproduceerd zijn uit welke events is het van belang dat deze gekoppeld kunnen worden. Omdat elk van de tabellen meer dan 100 kolommen telt, is niet wenselijk elke kolom op te nemen in het Entity Relationship Diagram. Het ER diagram in Figuur 10 is overzichtelijk omdat er van elke databron een enkele tabel gebruikt wordt, en de relationele sleutels gedeeld zijn. Doordat elke kolom een veelvoud van type ID's bevat, elk benodigd voor een koppeling met een andere database. Om dit af te vangen is er nog enige data preparatie nodig. Dit wordt toegelicht bij de relaties.



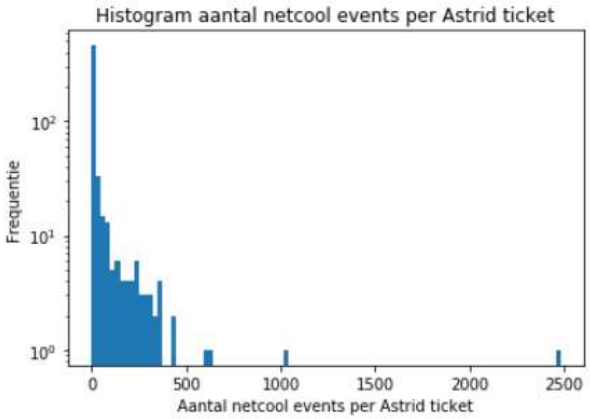
Figuur 10. Entity Relationship Diagram

4.2.1.1. Koppelen LaDiDa en Astrid

De koppeling tussen LaDiDa en Astrid gebeurt op basis van de kolom $Ticket\_nummer$ uit LaDiDa en de kolom $call\_id$ uit Astrid. Dit is een mogelijke one-to-one relatie, omdat er weinig LaDiDa events zijn en deze events altijd een koppeling hebben met een Astrid ticket. Er zijn in totaal 16 LaDiDa events gekoppeld aan de Astrid data in de termijn van november 2017 t/m februari 2019. De overige 4 LaDiDa events die aanwezig zijn in de dataset zijn niet bruikbaar voor dit onderzoek omdat deze voor november 2017 hebben plaatsgevonden. Deze data wordt dan ook enkel gebruikt om aan te geven of een incident gekoppeld is met een LaDiDa event.

4.2.1.2. Koppelen Astrid en Netcool

Het koppelen van de Astrid incident tabel en Netcool reporter status tabel kan gedaan worden op de kolom $call\_id$ van Astrid en de kolom $troubleticket\_id$ van Netcool. Na de koppeling blijven er echter nog maar 659 Astrid tickets en 24157 Netcool events over. De relatie is mogelijk one-to-many; er zijn dus meerdere Netcool events per Astrid ticket, waarvan de mediaan 3 events per ticket is met een maximum van 2478 events op één ticket. In Figuur 11 is de verdeling van events per ticket te zien. Er is dus in totaal van de grofweg 3.1M events van Netcool data op 24157 events gehandeld.



Figuur 11. In totaal hebben 177 Astrid tickets maar een enkele Netcool events gekoppeld. Het Astrid ticket met de grootste hoeveelheid Netcool events telt er in totaal 2478

Ondanks deze reductie in data hoeveelheid blijft het resterende deel van Netcool wel relevant omdat vermoed wordt dat de events waar niet op gehandeld is wellicht een voorbode zijn voor een event dat wel tot een incident heeft geleid.

4.2.1.3. Koppelen Netcool en WDM Trails

De WDM Trails geven aan hoe de nodes in Netcool verbonden zijn met elkaar. De relatie van WDM Trails tot Netcool is een one-to-many: Er kunnen meerdere events gekoppeld kunnen worden met een enkele node.

Het is echter zo dat de notatie binnen de WDM Trails vele malen uitgebreider is de notatie in Netcool. In de WDM Trails is een node bijvoorbeeld als volgt geformuleerd

$$Asd2 - WDM8W - 14 - Asd2 - WDM8W - 14 - shelf1 - shelf0 - 6 - MD8 - 9(OUT)$$

Waar in Netcool deze betreffende node als volgt is opgenomen:

$$Asd2\ /\ WDM8W\ /14$$

De notatie van de node aliassen in Netcool is als volgt:

Plaats / Apparaat type / volgnummer

Door de scheidingstekens gelijk te maken kunnen deze alsnog gekoppeld worden met elkaar. Desondanks zijn er een totaal van 467 nodes in het Netcool bestand die te complexe correcties vereisen om te matchen met de data in de WDM Trails. Aan deze nodes zijn een klein aantal Netcool events gekoppeld, waardoor het verlies van Netcool events beperkt blijft. Omdat er minder Netcool events overblijven, en de relatie zich hoger in de hiërarchie bevindt dan de overige databronnen, betekent dit dat er ook minder Astrid incidenten over blijven, en dus ook minder LaDiDa events.

Er blijft na het weglaten van deze nodes een totaal van 5511 nodes over die een totaal van 2119006 events hebben geproduceerd tussen de periode november 2017 t/m februari 2019. Van deze 2119006 events zijn er in totaal 21021 events die ook daadwerkelijk geleid hebben tot 566 incidenten. In overleg met KPN is dit voor het doel van het onderzoek een acceptabel verlies. Met de gevalideerde en gekoppelde data is het mogelijk om de modellen op te stellen. Dit wordt gedaan in het volgende hoofdstuk Modelleren.

5. Modelleren

Om inzicht te krijgen in de sequenties van *alertgroups* die leiden tot een incident, wordt in de paragraaf Markov-keten gekeken naar de meest waarschijnlijke paden. Vervolgens wordt in de volgende paragraaf de formule vastgesteld voor het bereken van de Time to Failure. Deze wordt in de daar op volgende paragraaf vervolgens gebruikt om vast te stellen wat de Mean Time to Failure is en om het survival model op te stellen.

5.1. Markov-keten

Om inzichtelijk te krijgen hoe bepaalde alertgroups overspringen naar ander soort alertgroups en incidenten wordt er gebruik gemaakt van een Markov-keten. Om de Markov-keten op te stellen is er gebruik gemaakt van de Netcool data, zoals beschreven in het data model. In deze data is de kolom $Alertgroups$ de kolom die gebruikt is voor het verkrijgen van de toestanden. Deze aanpak is gebaseerd op (Hossain, 2017), waar een Markov-keten gebruikt wordt om de transitiekansen en de Mean Time to Event (MTTE) te berekenen. Dit is gedaan omdat de alarmen in het totale bestand niet altijd een relatie met elkaar hebben. Aanvullend wordt er ook gebruik gemaakt van alle toestanden, waar (Hossain, 2017) enkel gebruik maakt van 80% van de meest voorkomende toestanden.

nodealias	alertgroup	lastmodified
Db-P87/TRSP/1	OTU2_SSF	2017-11-28 03:59:30
Db-P87/TRSP/1	R_LOS	2017-11-28 03:59:30
Db-P87/TRSP/1	R_LOS	2017-11-28 04:16:07
Db-P87/TRSP/1	OTU2_SSF	2017-11-28 04:16:07
Db-P87/TRSP/1	REM_SF	2017-12-08 08:31:15

Figuur 12. Voorbeeld Netcool alertgroup data van de node Db-P87/TRSP/1.

Een voorbeeld van deze data is te zien in Figuur 12. Op basis van deze reeks kan een lijst van 2-tuples verkregen worden. Een 2-tuple heeft de volgende vorm:

$$(OTU2\_SSF, R\_LOS)$$

Met deze lijst is het mogelijk om de overgangsmatrix van de toestanden te berekenen door in een matrix op de bijbehorende kolom en rij te tellen hoe vaak deze zijn voorgekomen. Vervolgens wordt er gedeeld door de som van de rij en worden op deze manier de overgangskansen verkregen. Tevens wordt de Mean Time to Failure matrix bijgehouden op basis van de tijd die het kostte om van een toestand naar de andere toestand te gaan.

De toestanden, ofwel alertgroups, die een incident hebben veroorzaakt worden overschreven met de naam $incident$ met als doel om te achterhalen welke alertgroups hebben geleid tot de alertgroup die een incident heeft veroorzaakt. Een nadeel hiervan is dat de kans van de toestand overgang $P_{incident \rightarrow incident}$ niet representatief voor de daadwerkelijke kans. Het is mogelijk dat een alertgroup direct wordt omgezet naar een incident omdat deze direct impact heeft op de dienstverlening en er dus geen alertgroup aan vooraf ging die als waarschuwing geldt. Indien dit voor een enkele node meerdere keren achter elkaar gebeurt, vertaalt dit zich naar een 2-tuple met daarin $(incident, incident)$ waar er feitelijk een incident al is verholpen. Zodoende dient de kans $P_{Incident \rightarrow Incident}$ bij de toestand overgang met zorg genomen te worden. De toestandenmatrix voldoet aan de volgende voorwaarde, met een totaal van 183 toestanden:

$$\sum_{k=1}^{183} p_{ik} = \sum P(X_{m+1} = k \,|X_m = i) = 1.0$$

In de toestandenmatrix is er voor elke toestand een meest grote kans om over te gaan naar een andere toestand maar zijn er ook toestanden waar de kans 0 is om over te gaan naar een andere toestand. In het geval dat er een kans van 0 is om over te gaan naar een andere toestand, zal deze toestand dus ook nooit overgaan. Dit kan ook gebeuren voor incidenten, zo kan het zijn dat er in de

data toestanden aanwezig zijn die nooit direct zullen leiden tot een incident. Desondanks is het van waarde om te weten naar welke toestand deze dan wel zal leiden en welke van deze toestand vervolgens de grootste kans heeft om een incident te veroorzaken. Dit pad van toestand naar toestand wordt het meest waarschijnlijke pad genoemd. Met behulp van Dijkstra's algoritme kan er gekeken worden welk pad hier dan het meest waarschijnlijk toe leidt, voor elke toestand die er aanwezig is.

Om inzichten uit een dusdanige grote toestandenmatrix te halen en antwoord te kunnen geven op de derde deelvraag: *hoe ontwikkelen fouten zich binnen het netwerk van KPN*, wordt de methode van (Reichel, 2014) gebruikt. Hier wordt de methode voorgesteld om met behulp van Dijkstra's algoritme het meest waarschijnlijke pad te berekenen vanuit een gegeven toestand. Het antwoord op deze deelvraag wordt gegeven door het overzicht van de sequenties en de meest waarschijnlijke paden. Hieruit kan vastgesteld worden welke alertgroups zullen leiden tot incidenten en welke niet.

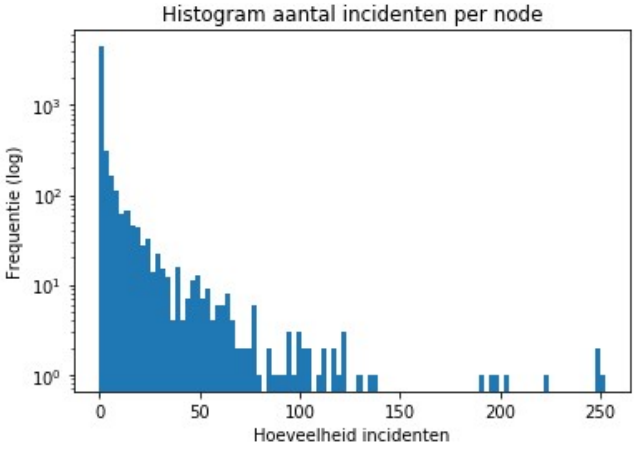
Omdat er met kansen gewerkt wordt en deze waardes niet bij elkaar opgeteld kunnen worden kan er het negatieve logaritme genomen worden van de waardes in de overgangsmatrix, negatief om te corrigeren voor de negatieve waardes die ontstaan uit het nemen van het logaritme van waardes onder de 1.

$$\mathrm{P}_{A\to B} = -\,log\,(\mathrm{P}(X_{m+1} = B \mid X_m = A))$$

Op basis van het meest waarschijnlijke pad vanuit een gegeven toestand naar de anderen kan er worden gekeken welk alarmen direct kunnen leiden tot een incident, en welke niet. Daarbij is het waardevol om te kijken naar de Mean Time to Event (MTTE) die inzicht kan geven in de hoeveelheid tijd die een alarm gemiddeld nodig heeft om een incident te worden. Hoewel de Markov-keten antwoord geeft op de derde deelvraag, geeft dit nog geen antwoord op de hoofdvraag. Hiervoor is eerst de Time to Failure nodig als input voor het survival model.

5.2. Time to Failure

Om een survival model op te zetten is het van belang dat er een levensduur bekend is die gebruikt kan worden door een survival model. De opzet van het Survival model wordt vergelijkbaar gedaan zoals in (Mats, 2002) en (Frisk, 2014). In (Mats, 2002) wordt gebruik gemaakt van een enkel Cox Proportional Hazard model om een voorspelling te doen van de Mean Time To Failure (MTTF). In (Frisk, 2014) wordt echter de remaining useful lifetime voorspeld van een datapunt. De aanpakken van deze onderzoeken worden voor een deel gecombineerd, omdat er in dit onderzoek gebruik gemaakt wordt van het XGboost model met de Cox Proportional Hazard als doelfunctie. Om dit model op te stellen is echter de Time to Failure benodigd.



Figuur 13. De verdeling van hoeveelheid incidenten per Node in de tijdspan November 2017 t/m Februari 2019. De node met de grootste hoeveelheid incidenten heef 252 incidenten gehad sinds het begin van de meting.

Om de Time To Failure te berekenen kan de gehele Netcool dataset gebruikt worden. Het gebeurt vaker dat een node meerdere keren stuk gaat in deze tijdspan, maar het gebeurt ook dat een node helemaal niet, of maar een enkele keer stuk gaat in dezelfde tijdspan. Zodoende is er onderscheid tussen right-

censored incidenten en uncensored incidenten, right-censored betekent dat er nodes zijn geweest die niet een incident hebben ondervonden gedurende de onderzochte periode. Er is in de data sprake van een incident als er een Astrid ticket van is aangemaakt. De berekening van de Time To Failure word als volgt gedaan:

$Time\ to\ Failure\ (uncensored) = t_n - t_{n-1}$ waar $start \leq (t_n, t_{n+1}) \leq end$ met $n \geq 2$

$Time\ to\ Failure(censored_{start}) = t_0 - start$

$Time\ to\ Failure(censored_{end}) = end - t_n$

Waar $start = 2019 - 11 - 01$, $end = 2019 - 02 - 28$,t_n het tijdstip van incident is voor een gegeven node en n de hoeveelheid incidenten van een node binnen de tijdspan $(start, end)$. Zo is bijvoorbeeld t_0 het eerste incident dat voorkomt in die tijdreeks. De periode van $start$ tot end betreft een tijdspan van 484 dagen.

Deze Time to Failure data wordt berekend voor elke node. Dit leidt tot 15083 unieke Time to Failures waarvan er 6516 right-censored zijn.

5.3. Survival functie & Hazard Rate

Op basis van de verkregen reeks met Time To Failures is het mogelijk om de Survival Functie op te stellen zoals beschreven in hoofdstuk 3.3.2.1. Door de Kaplan-Meier estimator te fitten op de reeks van Time To Failure wordt de Survival kromme verkregen. Deze kromme geeft vervolgens antwoord over het type faalgedrag die de nodes vertonen, dit kan bijvoorbeeld “Infant Morality” of “Wear Out” failures zijn. De periode waar de data op gefit is betreft de periode van $start$ tot end.

Aanvullend wordt er gekeken naar de Hazard Rate en deze wordt verkregen met behulp van de Nelson-Aalen estimator. De te verwachten krommes en gebruikte formules zijn omschreven in het hoofdstuk Hazard Function.

5.4. Extreme Gradient Boosting

In het theoretische kader is het Extreme Gradient Boosting algoritme toegelicht met als doelfunctie het Cox Proportional Hazard, zoals beschreven in het hoofdstuk Survival modellen. Dit model wordt gebruikt met als doelfunctie een Cox Proporional Hazard survival model. Voordat de parameters van het model kunnen worden berekend, moet eerst de data in het juiste format zijn verwerkt. Hiervoor wordt de gewenste output en de bijbehorende data rij voor rij verkregen. De data die het model helpt onderscheid te maken tussen de verschillende outputs worden *features* genoemd. Door kruisvalidatie toe te passen in combinatie met Harrell’s Concordance Index wordt het model geëvalueerd. Tot slot worden de invloeden van de variabelen op de voorspellingen onderzocht.

5.4.1. Model opzet

Om het model goed te leren welke omstandigheden hebben geleid tot het incident van deze node wordt de data verkregen in de tijdspan van $(t_n,\ t_{n-1})$ en afgezet tegen de Time to Failure. Door het model vervolgens data te geven vanaf t_n kan een voorspelling worden verkregen voor de verwachte levensduur op basis van de historische gegevens.

5.4.2. Doelfunctie

Laat $x_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}$ de gerealiseerde waardes zijn van de covariaten voor onderwerp i. De hazard functie voor het Cox PH model heeft dan de volgende vorm:

$$h(t) = h_0(t) * e^{(b_1x_1+b_2x_2+\cdots+b_nx_p)}$$

waar

t = de overlevingstijd vertegenwoordigd

$h(t)$ = is de hazard functie die bepaald wordt door een set van n covariaten $(x_1, x_2, \dots, x_n)$.

b_n = De coëfficiënten bepalen de impact van de covariaten

h_0 = de baseline hazard die kan worden verkregen door alle covariaten gelijk aan 0 te stellen in $e^{(b_1x_1+b_2x_2+\cdots+b_nx_p)}$.

Omdat het XGBoost model uit een combinatie van duizenden verschillende Cox PH modellen bestaat, met elk een eigen waarde van de b_n coëfficiënten, en elk een eigen waarde van $h_0(t)$, is het niet mogelijk om het voorbeeld uit te werken. De voorspelde waarde van het XGBoost model met Cox PH als doelfunctie is echter enkel de Hazard Ratio en betreft dus enkel de waarde die voorkomt uit $b_1x_1 + b_2x_2 + \cdots + b_nx_p$.

5.4.3. Feature engineering.

Voor het XGBoost model zijn diverse covariaten, ofwel features, gebruikt. Deze features zorgen ervoor dat het model een onderscheid kan maken tussen de verschillende verwachte levensduren. Voor het model is er gebruik gemaakt van zoveel mogelijk databronnen om het model zo goed mogelijk onderscheid te kunnen laten maken.

In de data is een aantal categorische variabelen aanwezig die niet direct gebruikt kunnen worden voor het model. Dit betreft namelijk datapunten van het type String en het model kan enkel overweg met numerieke datatypen. Om dit probleem op te lossen is het mogelijk om een aantal "dummy" kolommen toe te voegen. Het aantal kolommen dat wordt toegevoegd staat gelijk aan de hoeveelheid unieke events. Elke kolom krijgt vervolgens de waarde 1 indien de waarde overeenkomstig met de kolom aanwezig is in die rij, en 0 als dit niet aanwezig is.

De features die zijn gebruikt in het model, komen uit elke beschikbare databron en hebben als doel om het model onderscheid te laten maken tussen situaties waar een node een incident veroorzaakt heeft en waar niet. Allereerst wordt er begonnen met de data uit de Netcool gegevens die het dichtst bij de Node staan. De informatie die hier uit gehaald wordt zou goed moeten kunnen omschrijven hoe de historie van een node er uit ziet. Zo wordt er gekeken naar het cumulatief van de type alerts, het type apparaat waar het feitelijk om gaat en hoe veel van dit soort apparaten aanwezig zijn bij KPN. Wellicht namelijk dat een type apparaat eerder geneigd is te falen dan een ander soort type apparaat. Verder word er ook gekeken naar de hoeveelheid nodes die voorkomen in dezelfde locatie. Verder is het ook waardevol om te weten wat het historische gemiddelde interval is van events. Nodes die frequent falen zullen in de toekomst waarschijnlijk ook falen. Er zijn nog meer features gebruikt voor het model die vergelijkbare informatie bieden die worden toegelicht in de onderstaande hoofdstukken.

5.4.3.1. Historische Netcool gegevens

De Netcool gegevens zijn -als vermoeden- of vermoedelijk het meest invloedrijk op het kunnen voorspellen of er een event gaat gebeuren. Er wordt verwacht dat nodes die vaak problemen geven, deze blijven geven. Dit kan zijn omdat deze nodes zwaarder belast worden, gecompliceerd geconfigureerd zijn, of in een omgeving aanwezig zijn die aan verandering onderhevig is. Alternatief is het ook zo dat een deel van de Netcool events wel ontstaan zijn in het OTN, maar de oorzaak ligt in een netwerk bovenop het OTN. Op basis van een grote dataset wordt deze kans zoveel mogelijk geminimaliseerd.

De Netcool gegevens die gebruikt worden bestaan uit:

Tabel 3 Overzicht en omschrijving van Netcool features.

Feature naam	Type feature	Aantal	Toelichting
Total_alert_row_N	Categorisch	182	De som van alerts sinds $start$ voor die gegeven Node. Elke kolom is een unieke alert.
Total_services_row_N	Categorisch	7	Een aparte classificatie voor het soort service dat verleend moet worden voor het type fout
Node_device_type_N	Categorisch	7	Het soort hardware type van de node.

Node_had_ladida	Numeriek	1	Of een node ooit een LaDiDa event heeft gehad.
Node_device_count	Numeriek	1	Het aantal apparaten van dit type dat voorkomt in de gehele dataset.
Node_place_count	Numeriek	1	De hoeveelheid nodes die voorkomen in dezelfde stad.
Last_event_original_severity	Numeriek	1	De ernst van de laatste event zoals deze origineel is geproduceerd.
Last_event_severity	Numeriek	1	De ernst van de laatste event zoals deze handmatig is gecorrigeerd.
Last_event_tally	Numeriek	1	De hoeveelheid identieke events dat is geregistreerd op hetzelfde event ID.
Total_errors_so_far	Numeriek	1	De hoeveelheid events op een gegeven node, in totaal.
Total_mean_error_interval_hours	Numeriek	1	De gemiddelde tijd tussen de events, per node.
Total_severity_mean	Numeriek	1	De gemiddelde severity van alle events op een Node.
Total_tally_mean	Numeriek	1	De gemiddelde tally van alle events op een Node.
Total_ticket_count_on_node	Numeriek	1	De hoeveelheid incidenten op een gegeven node.

5.4.3.2. Historische Astrid gegevens

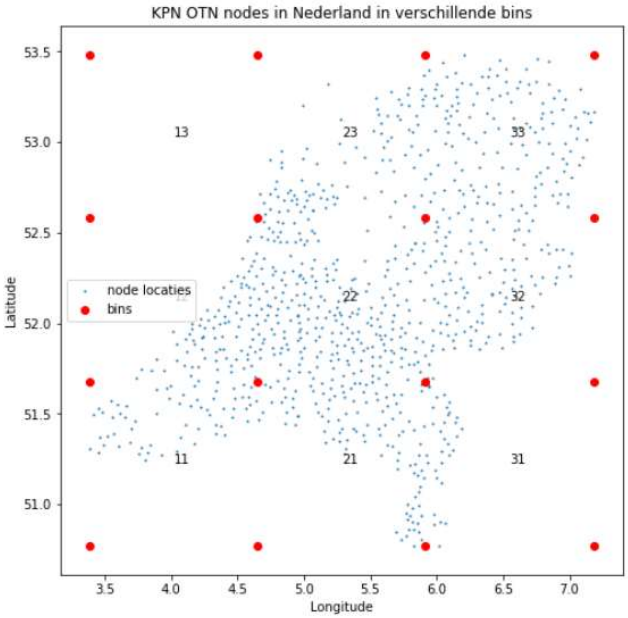
In de Astrid databron wordt de historie van een incident bijgehouden en hoe deze wordt opgelost in tekstvorm. In deze tekst zijn de diverse handelingen van diverse personen omschreven en hoe zij het probleem hebben opgelost. In sommige gevallen zijn deze teksten extreem lang en duiden op een complex probleem. In andere gevallen zijn ze bijzonder kort en was het eenvoudig om het probleem op te lossen. Door gebruik te maken van de lengte van een bericht is het mogelijk om de hoeveelheid complexe problemen te verwerken in de features.

Tabel 4. Overzicht en omschrijving van Astrid features.

Feature naam	Type feature	Aantal	Toelichting
Progress_note_length_last	Numeriek	1	De lengte van het bericht in de progressie kolom bij een incident. Grofweg, hoe langer, hoe complexer het incident om op te lossen.
Progress_node_length_mean	Numeriek	1	Het gemiddelde van de lengte van de incident berichten. Hoe hoger, hoe waarschijnlijker het is dat er vaak complexe problemen aanwezig zijn op deze node.

5.4.3.3. Geografische regio's

Omdat het vermoeden aanwezig is dat de geografische ligging van een node in het netwerk van invloed is op de betrouwbaarheid, is ervoor gekozen om deze mee te nemen als dummy variabele. Om de hoeveelheid dimensies te beperken is er voor gekozen om Nederland op te delen in 9 verschillende vlakken op basis van de extrema die zijn gevonden in de data. Aanvullend kunnen deze regio's het koppelen van de KNMI data vereenvoudigen. Hoewel het overgrote binnen Label 22 valt is er voor gekozen om de label opbouw niet aan te passen omdat het koppelen van de labels aanzienlijk compliceert en omdat er andere variabelen aanzienlijk meer invloed hebben op de uiteindelijke voorspelling, zoals wordt beschreven in de resultaten.



Figuur 13. Indexering van Nederland in 9 regio's.

Tabel 5 Overzicht en omschrijving van geografische features.

Feature naam	Type feature	Aantal	Toelichting
Node_place_label_dum_n	Categorisch	10	Nederland opgebroken in 9 verschillende regio's met elk een eigen label. Sommige nodes konden niet gematcht worden vanwege incomplete vertaal tabellen, deze kregen het label "00".

5.4.3.4. KNMI

Op de dag van een incident kan het warm of koud zijn. Hoewel het OTN uitsluitend is verwerkt in gekoelde kamers was het vermoeden dat het weer een kleine invloed heeft op de kans op een incident van Nodes. Anderzijds was het vermoeden wel sterk aanwezig dat een groot deel van de events binnen het OTN veroorzaakt wordt door de lagere niveaus van het netwerk, bijvoorbeeld de regionale wijkkastens, die wel vatbaar zijn voor temperatuurschommelingen. Door de temperatuur op de dag van een incident op te nemen als feature kan het model onderscheid maken in het geval er een defect optreedt met warm of koud weer. Het weer is gekoppeld op basis van de regio's die zijn verkregen uit de indexering van de geografische regio's.

Tabel 6 Overzicht en omschrijving van klimatologische features.

Type feature	Feature naam	Aantal	Toelichting
Numeriek	Label_temp_day	1	De gemiddelde temperatuur in het regio label op de dag dat de node een incident produceerde uitgedrukt in een getal met 1 decimaal

5.4.3.5. WDM Trails

Er is een deel data gebruikt uit de WDM Trails ter ondersteuning van het model. Zo zou er een onderscheid zitten tussen de kans op een incident op basis van de hoeveelheid verbindingen die een node heeft. Het vermoeden is dat een node die weinig verbindingen met zichzelf en/ of anderen, minder relevant is, en dus een kleinere kans op falen heeft. Door de mate van verbondenheid mee te nemen in het model kan het model ook onderscheid maken op de mate van verbondenheid.

Tabel 7 Overzicht en omschrijving van WDM Trail features.

Type feature	Feature naam	Aantal	Toelichting
Numeriek	Amount_connected_with_others	1	De hoeveelheid verbindingen die een node met andere nodes heeft.
Numeriek	Amount_connected_with_self	1	De hoeveelheid verbindingen die een node met zichzelf heeft.

5.4.3.6. Samengevoegde data

Om de verschillende features bij elkaar te voegen zijn deze allen gecombineerd tot een enkele grote tabel die als input kan dienen voor het model. Dit levert vervolgens een tabel van 15083 rijen met 221 kolommen aan trainingsdata.

5.4.4. Hyperparameters vaststellen

Om het model op te stellen en zo goed mogelijk te laten presteren, is gekozen om de hyperparameters op te stellen op basis van een Grid Search techniek. De hyperparameters zijn de parameters van het machine learning model zelf. Zo kan bijvoorbeeld de *"Learning Rate"* aangepast worden, die invloed heeft op de stapgrootte van de Gradient Descent. In deze techniek wordt een parameter zoekruimte opgegeven waarin elke combinatie geprobeerd wordt. Om het proces snel te laten verlopen is er gebruik gemaakt van een klein aantal hyperparameters.

Om dit te evalueren wordt er gebruik gemaakt van de C-index zoals omschreven in het theoretische kader in hoofdstuk 3.4.3. De hyperparameters met de hoogste C-index worden gekozen als het waardes voor de hyperparameters van het model. Bij het XGBoost model zijn veel verschillende hyperparameters aanwezig, het is echter onhaalbaar om elke combinatie aan hyperparameters te proberen. In de documentatie van het XGBoost model zijn een aantal richtlijnen gevonden over welke stappen handig zijn. Vervolgens is er met experimenteren een adequate aanpak gevonden.

Om de optimalisatie tijd enigszins te beperken is er gekozen voor een klein aantal hyperparameters om te verfijnen. Zo is er met de hyperparameter num_boost_round geprobeerd een betere score te geven. Deze hyperparameter maakt bij elke "Boosting" ronde een nieuwe estimator, in dit geval Cox PH model aan. Op basis van kruisvalidatie is gebleken dat een hoge num_boost_round leidt tot een hogere score. Deze kan echter ook te hoog zijn waardoor de score terugloopt. Deze zelfde situatie gaat ook op voor de hyperparameters eta ofwel "Learning Rate". De learning rate zorgt ervoor dat elke boosting ronde meer behoudend blijft. De max_depth zorgt ervoor dat het model complexer wordt en eerder leidt tot overfitting. Het gevaar van overfitten wordt afgevangen door de testset voldoende groot te maken en daarin te variëren.

Op basis van dit begin is er met behulp van een brute-force scan een hyperparameter zoekruimte opgezocht. De hyperparameters die zijn geoptimaliseerd zijn als volgt:

Tabel 8 Overzicht van de hyperparameters en de getoetste waardes

Num_boost_round	$1000, 3000, 5000, 8000, 10000, 15000$
Eta	$0.5, 0.1, 0.05, 0.01, 0.005, 0.001$
Max_depth	$3, 4, 5, 6, 7$

Dit maakt een totaal van $6 * 6 * 5 = 180$ modellen mogelijk die elk apart geëvalueerd worden. De set van hyperparameters die verantwoordelijk is voor de hoogste C-index wordt de definitieve set hyperparameters. Dit zijn de hyperparameters $num\_boost\_round = 10000, eta = 0.005,$ $max\_depth = 5$ geworden. Op basis van dit model zijn de resultaten verkregen

5.4.5. Mate van invloed van features verkrijgen

Het is ook van belang om te weten waarom een voorspelling is zoals die is, zodat juist hetgeen wat van invloed is op die voorspelling kan worden gelimiteerd. Zo kan het bijvoorbeeld zijn dat een node een grote kans op falen heeft omdat er een bepaalde alarmtype recentelijk is afgegaan die normaal gesproken niet direct tot een incident zou leiden. Daarbij is het ook zo dat sommige nodes een grote kans op falen hebben doordat de temperatuur van de dag erg hoog of laag is.

Met behulp van de analyse techniek Shapley Additivite Explanations (SHAP) die is voorgesteld in (Lundberg, 2018) kan er nader ingezoomd worden op deze voorspelde waardes, en welke variabelen van invloed zijn geweest voor het maken van de voorspelling van het model. De techniek van SHAP werkt kort gezegd door iteratief de waarde van een variabele te veranderen en te kijken wat er gebeurt met de uiteindelijke voorspelling, tegelijkertijd rekening te houden met de aard van een Ensemble Tree. Namelijk dat het wijzigen van een enkele feature niet een direct verband heeft met de uiteindelijke voorspelling, omdat de totale set van features van invloed is. SHAP vangt deze complexiteit op door gebruik te maken van spel theorie. De onderliggende theorie van SHAP is verder terug te lezen in het artikel van (Lundberg, 2018). In het volgende hoofdstuk Resultaten worden deze waardes berekend van het model.

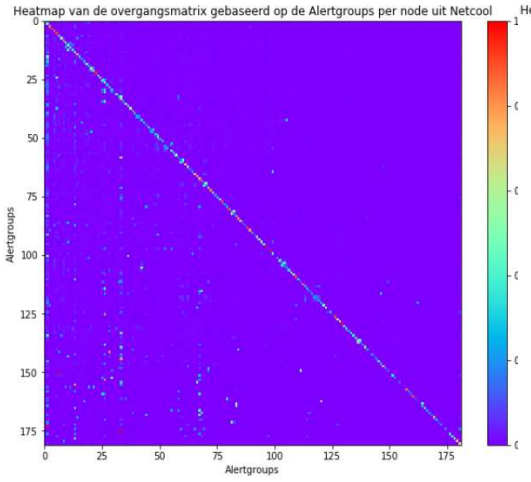
6. Resultaten

In Hoofdstuk 5 zijn twee modellen geïntroduceerd: Het Markov model en het Survival model. Deze modellen gebruiken de data van zowel KPN als van het KNMI, als beschreven in Paragraaf 2.2, om events in het OTN te voorspellen. Het vermogen om dit correct te voorspellen is voor het survival model geëvalueerd door gebruik van kruisvalidatie zoals uitgelegd in paragraaf 3.3.1. Vanwege de betere prestaties van het survival model zijn hier extra tests mee uitgevoerd om tot meer inzichten te komen, met betrekking tot de invloed van variabelen. In dit hoofdstuk worden eerst de resultaten van het Markov model beschouwd, en vervolgens het survival model.

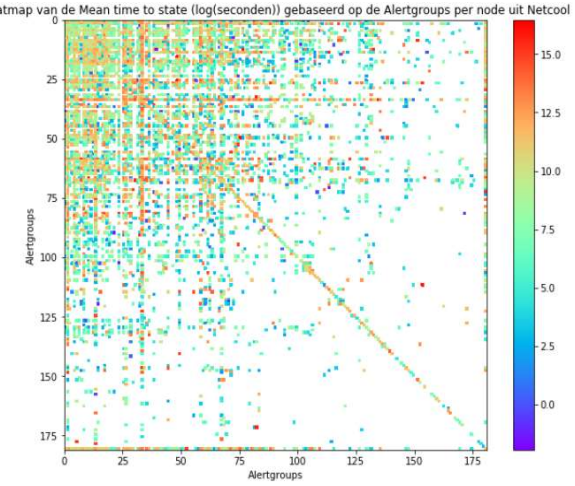
6.1. Resultaten Markov-keten

Allereerst worden in deze paragraaf de resultaten van de Markov-keten toegelicht. Hierbij wordt er gesproken over *toestanden* en *alertgroups*. Een alertgroup is gedefinieerd als een toestand waarin een node kan verkeren en wordt geproduceerd door een event, zoals aangegeven in het hoofdstuk Modelleren. De betekenis van elke alertgroup is vertrouwelijke informatie van Huawei en kan zodoende niet als bijlage worden bijgevoegd.

De dimensies van de toestandenmatrix zijn 183 bij 183 en te groot waardoor deze niet zinvol als grafiek gevisualiseerd kan worden. Er is in (Reichel, 2014) voorgesteld om de matrix te visualiseren als een heat map en zodoende is er gekozen om deze aanpak over te nemen. In Figuur 14A is de heatmap zichtbaar en sterk duidelijk is de diagonale lijn die in het midden oplicht. Dit betekent dat het overgrote deel van de toestanden het meest waarschijnlijk is om in diezelfde toestand te blijven. Verder is het ook opvallend dat het overgrote deel van de heatmap leeg lijkt te zijn. De waardes zijn alleen dusdanig laag dat deze nagenoeg niet oplichten. Er zijn een aantal uitzondering te vinden in de linker onder hoek maar de conclusie die getrokken kan worden uit dit figuur is dat de toestanden voornamelijk blijven in de toestand waar ze in zitten. Verder is uit de overgangsmatrix gebleken dat een tweetal alertgroups nooit een andere alertgroup veroorzaken, namelijk $10G\ service\ outage$ en $DBMS\_ABNORMAL$. Deze zijn niet meegenomen in de visualisatie en worden vanaf hier buiten beschouwing gelaten.



Figuur 14A. Overgangsmatrix van Markov-keten op basis van Alertgroups uit Netcool



Figuur 14B. Mean Time to Event uitgedrukt in Log(seconden)

De top 10 alertgroups die het meest waarschijnlijk een incident veroorzaken zijn te zien in Figuur 15. Zoals beschreven in de Methodologie Markov-keten dient de interpretatie $P_{incident \rightarrow incident}$ met zorg gedaan te worden. Hier is te zien dat de alertgroup ODU_LCK een kans van 0.167 heeft om binnen gemiddeld 0.018 uur tot een incident te leiden en de alertgroup $SERVICE\_OUTAGE$ een kans van 0.071 heeft binnen gemiddeld 0 uur. Dit betekent feitelijk dat de alertgroups $SERVICE\_OUTAGE$ in 7.1% van de gevallen direct tot een alertgroup leidt die een incident veroorzaakt. Een alertgroup zoals $TEMP\_ALARM$ heeft maar een kans van 0.019 met een gemiddelde van bijna 180 uur om tot een incident te leiden, dit is een betrekkelijk kleine kans om tot een incident te leiden maar behoeft alsnog meer aandacht.

alertgroup	Kans	MTTE (uur)
INCIDENT	0.6820	22.375
ODU_LCK	0.1667	0.018
SERVICE_OUTAGE	0.0714	0.000
ETH_LINK_DOWN	0.0435	284.822
FEC_OOF	0.0426	0.166
MODULE_TEMP_OVER	0.0357	0.000
OPU2_PLM	0.0282	0.061
TP_LOC	0.0244	0.000
OSC_RDI	0.0192	0.000
TEMP_ALARM	0.0189	179.679

Figuur 15. De top 10 alertgroups die het meest waarschijnlijk een Incident veroorzaken

Dit zijn dan goede alerts om in de gaten te houden aangezien deze een voorbode zijn voor het ontstaan van incidenten, bij de overige incidenten dient er nog een extra zorg getroffen te worden omdat deze direct kunnen leiden tot een incident.

Op basis van Dijkstra's algoritme is het mogelijk om de meest waarschijnlijke routes te krijgen, namelijk de het pad van toestand naar toestand met de grootste kansen die uiteindelijk tot een incident leiden. De resultaten van Dijkstra's algoritme geven aan dat 102 alertgroups het meest waarschijnlijke pad direct naar een incident nemen, zonder via een andere alertgroup te gaan. Verder zijn er 77 alertgroups die het meest waarschijnlijk via een andere alertgroup een alarm worden. De langste route bestaat uit 3 unieke alertgroups voordat deze tot een incident leidt, namelijk de alertgroup OMS_SSF_O zoals weergeven in Figuur 16. De kans dat deze alertgroup via deze route überhaupt leidt tot een incident is $0.75 * 0.0392 * 0.004 = 0.00012$, ofwel vrijwel verwaarloosbaar klein. Zelfs de alertgroup R_LOS heeft nog steeds maar een kans van 0.004 om een incident te veroorzaken binnen grofweg 53 uur.

Start toestand	eind toestand	kans	MTTE
OMS_SSF_O	OOS_LOST	0.7500	0.225
OOS_LOST	R_LOS	0.0392	163.206
R_LOS	INCIDENT	0.0040	52.647

Figuur 16. Een optimale route met de meeste alertgroups begint vanuit alertgroup OMS_SSF_O. Deze alertgroup kan dus beschouwd worden als slechte voorspeller voor het veroorzaken van een incident.

Voor de routes met meer dan 1 unieke alertgroup is het interessant om te kijken naar de routes met totaal de meeste kans om een incident te veroorzaken. Hoewel deze historisch gezien nooit direct geleid hebben tot een incident kan er op deze manier alsnog gekeken worden naar welke alerts het meest waarschijnlijk zijn om tot een volgend alert te leiden die wél een incident kan veroorzaken. Zodoende kan er naar deze alertgroups extra zorgvuldig gekeken worden.

Start toestand	Totaal kans
OMS_BDI	0.010668
SM_BEI	0.006050
PM_BEI	0.006050
MUT_TLOS	0.005440
SUM_OUTPWR_LOW	0.005440

Figuur 16 Routes met meer dan 1 unieke alertgroup de totale kans berekend en gesorteerd op de grootste kans.

In Figuur 17 is een overzicht te zien met de routes met meer dan 1 unieke alertgroup, gesorteerd op de grootste totale kans. Hier is in te zien dat de alertgroup OMS_BDI de grootste kans heeft om een andere alert te veroorzaken, namelijk een totale kans van 0.01. Dit is alsnog betrekkelijk laag, maar geeft wel een goede indicatie. Opmerkelijk is wel dat de overgang $P_{OMS\_BDI \rightarrow OSC\_RDI}$ een MTTE heeft van 0.010 en dus met een grote kans zeer snel overgaat naar de andere alertgroup. Dit maakt het lastig om direct op te handelen. De route van OMS_BDI is te zien in Figuur 18.

Vanwege de grote hoeveelheid routes met één unieke alertgroup die direct overspringen naar de alertgroup incident en de kleine kans van de routes met meer dan 1 unieke alertgroup, en het succes van het survival model, is er voor gekozen om geen verder onderzoek meer te doen naar het Hidden Markov Model. Dit model kan echter nog steeds waardevol zijn voor nader onderzoek.

		kans	MTTE
Start toestand	**eind toestand**		
OMS_BDI	**OSC_RDI**	0.5556	0.010
OSC_RDI	**INCIDENT**	0.0192	3.321

Figuur 18. Route met meer dan 1 unieke alertgroup met de grootste totale kans.

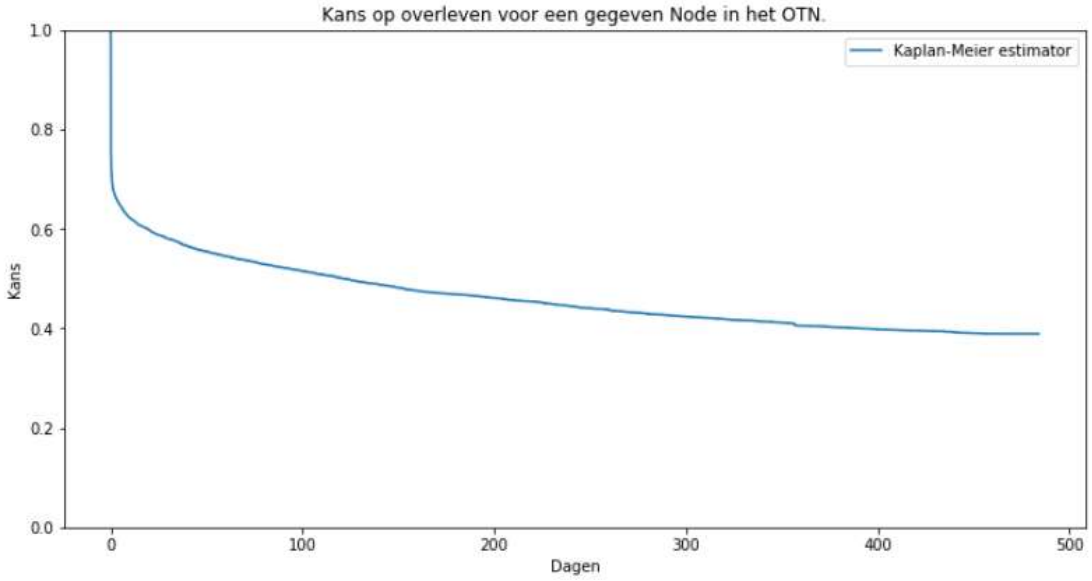
Tevens is er vanwege de prestaties van het Survival model gekozen om enkel te blijven tot het verkrijgen van inzichten van de ontwikkeling van alertgroups bij de Markov-keten en om geen voorspellingen proberen te doen met dit model. Dit is echter alsnog mogelijk.

6.2. Resultaten Survival Model

De Markov-keten heeft inzicht gegeven in welke sequenties leidend zijn voor het veroorzaken van een incident. Dit model is echter niet gebruikt voor het maken van voorspellingen, hier is het survival model voorgebruikt en wordt toegelicht in dit hoofdstuk.

6.2.1. Survival functie en Hazard Rate resultaten

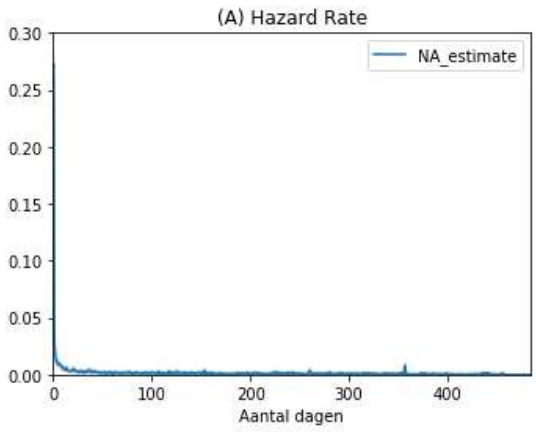
De survival kromme die geproduceerd wordt door een Kaplan-Meier estimator geeft de kans op overleven. De kromme houdt op bij dag 484 met een kans van 0.389, ofwel de kans voor een node om voor 484 dagen lang geen incident te veroorzaken is 0.389. De kans daalt niet verder naar nul omdat er een groot aantal nodes in het netwerk zijn die gedurende de gehele periode van 484 dagen geen incident hebben, ofwel right-censored zijn. Verder is het opvallend dat op dag 0 de kans op overleven 0.995 is en op dag 1 de kans enorm gedaald is naar 0.684.



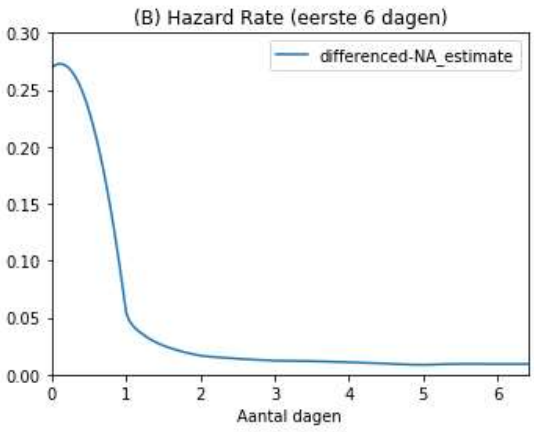
Figuur 19. Survival kromme van de nodes in het OTN.

Het verloop van deze kromme duidt op een "Infant Mortality' kromme, zoals beschreven in het hoofdstuk Hazard Function, die betekent dat een node de grootste kans op falen heeft direct na een reparatie. Dit komt overeen met het vermoeden van KPN, dat een groot aantal incidenten resulteert uit een foutieve configuratie of door complicaties door onderhoud. Eenmaal dat een node de eerste paar dagen overleeft heeft is de kans bijzonder klein dat er nog iets misgaat. Het advies dat volgt uit deze kromme is dan ook om extra scherp op te letten op nodes die recentelijk onderhoud gehad hebben. Op basis van de Kaplan-Meier estimator is het mogelijk om de Median Time to Failure uit te rekenen. Dit is het aantal dagen dat een node in 50% van de gevallen tenminste zonder incident overbrugt, ofwel oplossen voor $P(T \geq t) = 0.5$ geeft de Median Time to Failure. De Median Time to Failure komt uit op $t = 121.0$ dagen.

Om het directe gevaar, ofwel de Hazard Rate uit te lezen is het mogelijk om met gebruik van de Nelson-Aalen estimator de Hazard Rate te plotten. In de Hazard functie is het directe risico op een gegeven tijdstip uit te lezen. De gebruikte formules zijn terug te vinden in het hoofdstuk Hazard Function.



Figuur 20A Hazard Ratio over de gehele tijdperiode. Deze is bijna 0 voor het grootste deel.



In Figuur 20B is te zien hoe de failure rate zeer sterk daalt op de eerste dag

6.2.2. C-index

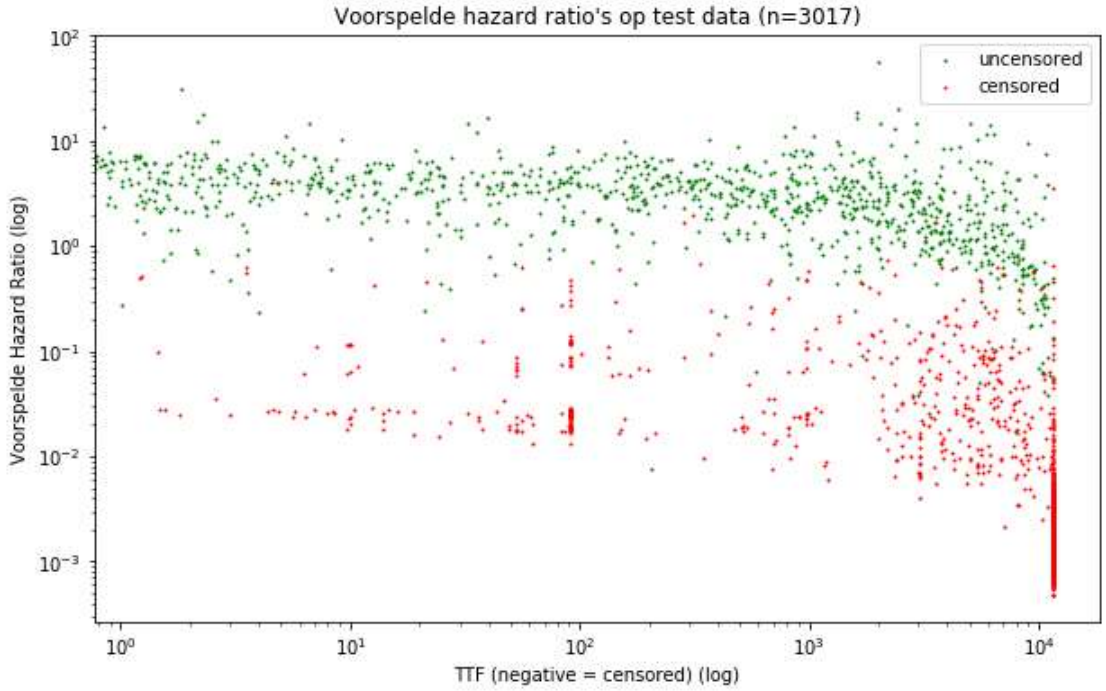
Zoals toegelicht in het hoofdstuk Theoretische Kader worden de prestaties van het survival model geëvalueerd met behulp van Harrel's Concordance Index, ofwel de C-index. Het model heeft op basis van kruisvalidatie met een split van 80/20 een C-index gehaald van 0.856. Waar 0 een zeer slechte score is, 0.5 een score is die gelijk aan gokken staat, 1 een perfecte score, dan is 0.856 een score die aangeeft dat het model goed in staat is om de verwachte hazard ratio in te schatten.

In Figuur 21 is te zien wat de verdeling is van de voorspelde Hazard Ratio ten opzichte van de gerealiseerde Time To Failure (TTF), indien gerealiseerd. Hoe hoger de voorspelde Hazard Ratio, hoe waarschijnlijker het is dat een node op het punt staat een incident te creëren.

Indien een datapunt een hoog voorspelde Hazard Ratio heeft, dan zal deze hoog geplaatst worden in de plot, bij een lage Hazard Ratio wordt deze laag afgezet. In dit figuur is te zien dat naarmate de TTF stijgt, en er dus verder naar rechts bewogen wordt, de voorspelde Hazard Ratio relatief daalt. Een groot deel gecensureerde data punten, waarvan er nooit een incident is geregistreerd, heeft zodoende een zeer lage Hazard Ratio. Het model voorspelt dan ook goed dat de kans op falen voor deze datapunten klein is, aangezien deze nooit gefaald hebben in de waargenomen tijd. Voor de datapunten die groen gekleurd zijn, en dus een waargenomen moment van falen hebben, is goed te zien dat deze bijna geheel boven de rode punten staan – deze hebben namelijk een moment van falen gehad en het model voorspelt hier dan ook een vergroot risico voor.

Het is verder ook opmerkelijk hoe er een rechte lijn van censored data punten rond de TTF van 90 aanwezig is. De reden hiervan is terug te vinden in de data van deze datapunten, zoals omschreven in

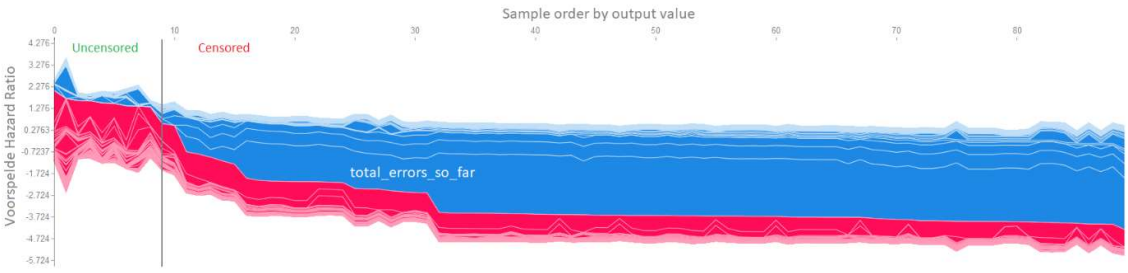
de volgende alinea met Figuur 22 als ondersteuning.



Figuur 21. Spreiding van voorspelde waardes op de test data (N=3017), inclusief censored data.

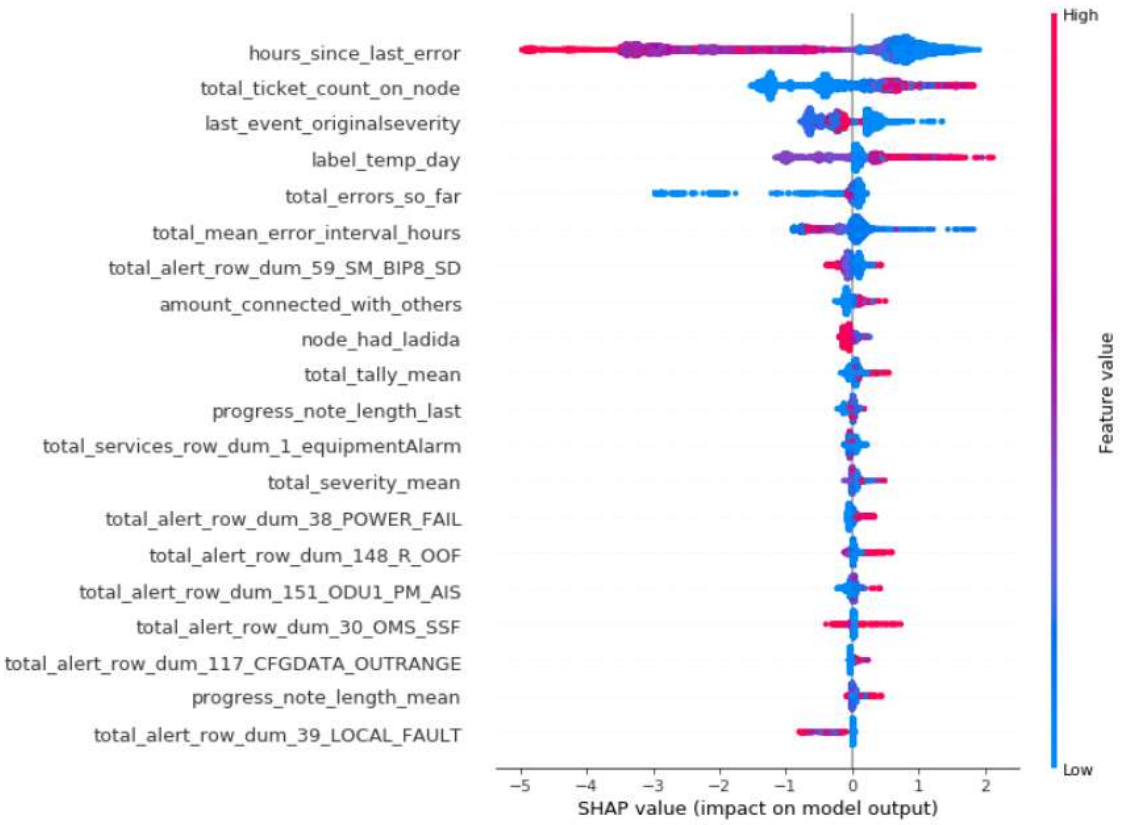
6.2.3. Mate van invloed van variabelen

Om inzicht te krijgen waar KPN moet zoeken voor het oplossen van de problemen is het waardevol om te weten wat de mate van invloed is van de variabelen. In Figuur 21 is een visualisatie van de bijdrages van elke feature voor elke voorspelling die een gerealiseerde TTF van rond de 90 dagen hebben. Deze grafiek is af te lezen door elke blauwe waarde die de Hazard Ratio naar beneden drukt, en de rode waardes de Hazard Ratio's omhoog duwen. Zo heeft bijvoorbeeld de lage waarde in de feature $total\_errors\_so\_far$ er voor gezorgd dat de Hazard Ratio laag komt te liggen. Omdat deze waarde voor elk van de voorspelde waardes hetzelfde is, blijft deze ook vrijwel hetzelfde. Er is een daling ter hoogte van de output value van 30, die is ook terug te zien in de tweede lijn in Figuur 21 van rode punten.



Figuur 22. Ingezoomde aggregatie Shapley force plot van de voorspelde waardes met een TTF van ongeveer 90 dagen. Deze visualisatie biedt inzicht over waarom er een lijn van punten aanwezig is rond de TTF van ongeveer 90 dagen in Figuur 21.

Ter aanvulling is het XGBoost ook nog zeer waardevol omdat het in combinatie met SHAP inzicht kan geven over de bijdrage van elke feature in het model en het op deze manier duidelijk kan maken welke variabelen van invloed zijn over het geheel.



Figuur 23. Mate van invloed van variabelen. Hoe roder de waarde is, hoe hoger deze is. Hoe verder deze aan de linker kant staat, hoe verder deze er verantwoordelijk voor is dat de Hazard Ratio laag wordt ingeschat, en vice versa.

In Figuur 23 is de bijdrage van de meest invloedrijke features uit te lezen, hoe hoger een feature in deze lijst staat, hoe invloedrijker deze is voor het model. Vanwege de technieken die zijn gebruikt bij de Shaply Additive Explanations is het ook nog uit te lezen welke waardes de voorspelling welke kant op sturen.

Zo is de $hours\_since\_last\_error$ vrijwel altijd bepalend voor de voorspelling; hoe meer uur sinds de laatste fout (high feature value, kleur rood), hoe lager de SHAP value (links op de as). De SHAP value kan direct vertaald worden als: $SHAP\ value = \log(Hazard\ Ratio)$. Zodoende kan er gesteld worden op basis van de visualisatie dat, hoe hoger de $hours\_since\_last\_error$, hoe lager de Hazard Ratio. Dit komt overeen met de survival curve zoals verkregen uit de Kaplan-Meier estimator, daar was te zien dat als een node de eerste paar dagen geen event geeft, er een kleinere kans ontstaat dat deze in de toekomst stuk gaat.

Opvallend is dat de temperatuur van de dag een zeer grote invloed heeft. De apparatuur van het OTN ligt in temperatuur geregelde gebouwen waar de temperatuur vrijwel altijd hetzelfde blijft. De vermoedelijke reden dat de temperatuur zo invloedrijk is, is dat incidenten haar oorsprong vinden in een hoger gelegen netwerk dan het OTN, maar dat enkel het OTN een event afgeeft op de dag dat de temperatuur hoog is. Het is namelijk zo dat het OTN vertakt naar lagere niveaus in het netwerk, die wel vatbaar zijn voor hoge temperaturen en zodoende kunnen zorgen voor events in het OTN.

Verder is het waardevol voor KPN om te weten of de events met een lage $originalseverity$ genegeerd kunnen worden, zoals nu doorgaans het geval is. Het antwoord hierop wordt gegeven in

Figuur 23. Hier is in af te lezen dat de variabele $last\_event\_originalseverity$ een zeer grote invloed op het model heeft als derde meest invloedrijke variabele. Zo is het mogelijk om op basis van het model te stellen dat events met een lage $originalseverity$ niet genegeerd moeten worden. Idealiter wordt het model geraadpleegd op mate van invloed op de voorspelde hazard ratio bij een gegeven event. Het model geeft namelijk aan dat een lage, blauw gekleurde, $originalseverity$ zowel de hazard ratio omhoog als omlaag kan duwen.

Een individueel datapunt kan ook in detail bekeken worden om verder onderzoek te doen naar welke factoren geleid hebben bij een gegeven voorspelling. In Figuur 24A is een datapunt met een zeer hoge Hazard Ratio uiteengezet. Zo is af te lezen dat de hazard ratio dusdanig hoog uitvalt doordat de $hours\_since\_last\_error$ relatief laag is, en de $total\_ticket\_count\_on\_node$ hoog. In de figuur reeks 24A, 24B, 24C zijn respectievelijk hoog, neutraal en laag voorspelde Hazard Ratio's uiteengezet en de mate van invloed van de bijbehorende data. In rood gekleurd zijn de invloeden weergeven die aan een positieve, dus hogere Hazard Ratio bijdragen en in blauw gekleurd zijn de invloeden weergeven die aan een lagere Hazard Ratio bijdragen. Zo is te zien dat in Figuur 24A een $hours\_since\_last\_error = 1155$ verantwoordelijk is voor het verhogen van de Hazard Ratio, en de veel hogere $hours\_since\_last\_error = 6232$ in Figuur 24C verantwoordelijk is voor het verlagen van de Hazard Ratio.

Vergelijkbaar is dat de $label\_temp\_day = 198$, een relatief hoge gemiddelde temperatuur van 19.8 graden celsius voor de gehele dag, verantwoordelijk is voor het omhoog drijven ten opzichte van de relatief lage temperatuur in Figuur 24C $label\_temp\_day = 75$ voof het omlaag drijven. Deze relatie gaat echter niet altijd op, zo is de koude temperatuur van $label\_temp\_day = 25$ van Figuur 24B verantwoordelijk voor het verhogen van de Hazard Ratio. De reeks Figuren in 24A, 24B, 24C geven zodoende een goed beeld bij de invloeden van de data op de voorspelde Hazard Ratio en kunnen gebruikt worden door KPN om te herleiden waarom het model doet wat het doet, gegeven dat er naar het geheel van de invloeden gekeken wordt.

Figuur 24A. Shaply Force plot op een zeer hoog voorspelde Hazard Ratio

Figuur 24B. Shaply Force plot op neutraal voorspelde Hazard Ratio

Figuur 24C. Shaply Force plot op zeer laag voorspelde Hazard Ratio.

De categorische variabelen worden weergeven als individuele features in SHAP. Het is echter van toegevoegde waarde om het totaal van de dummy variabelen te zien om in te schatten wat de waarde is van het type feature als geheel, en of hier wellicht meer varianten van toegevoegd moeten worden. Dit kan gedaan worden door het aggregeren van de feature scores. Door de individuele feature invloedrijkheid van alle categorische variabelen bij elkaar op te tellen kan dit inzicht alsnog verkregen worden.

De relatieve feature invloedrijkheid wordt berekend door te berekenen wat de kans is dat een sample een bepaalde node bereikt op basis van die gegeven feature. Deze kansen vertalen zich naar een invloedrijkheid score.

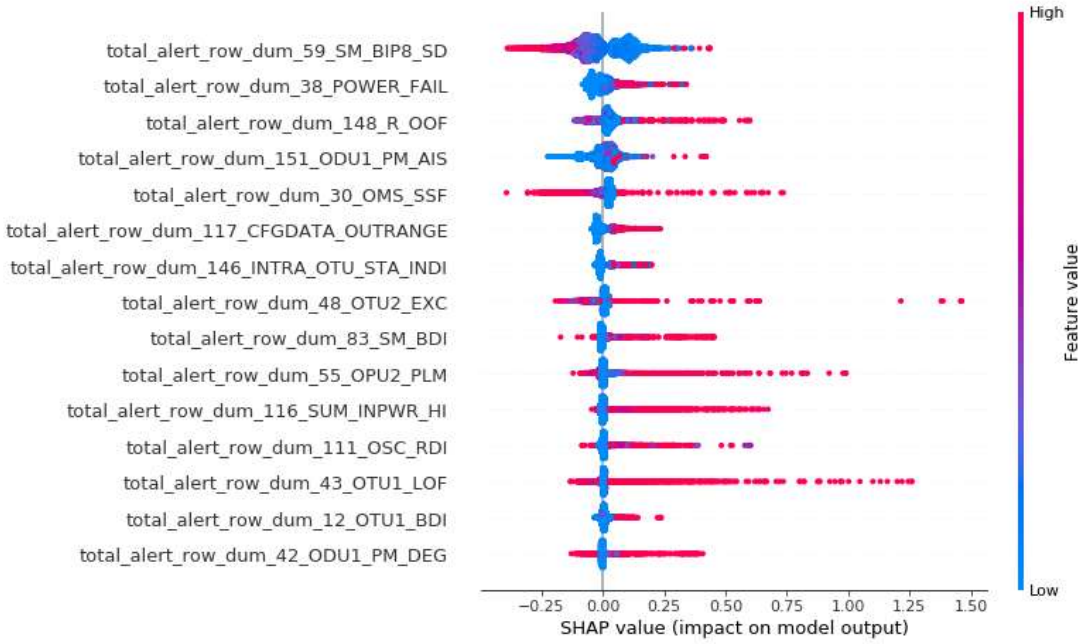
Variabelen	Mate van invloed
hours_since_last_error	27117
total_mean_error_interval_hours	16385
label_temp_day	14669
total_severity_mean	11606
total_tally_mean	11178
total_errors_so_far	8433

Figuur 25. Relatieve score mate van invloed per feature

Op basis van dit overzicht is het mogelijk om de categorische variabelen, die zijn opgebroken als dummy features, bij elkaar op te tellen. Uit deze optelling blijkt dat de historische Alertgroups types een zeer grote invloed hebben gehad op de voorspelling van het model. Hieruit is op te maken dat de historie van Alertgroups een zeer goede voorspeller is. Zodoende is het waardevol om te kijken naar de uiteenzetting van de waardes van Alertgroups. De 15 meest invloedrijke Alertgroups zijn uiteengezet in Figuur 26, met de meest invloedrijke bovenaan. Hier is in af te lezen dat hoe vaker een bepaald alarm type is afgegaan, hoe waarschijnlijker het is dat een Node binnenkort een incident veroorzaakt.

Dummy variabelen	min	max	sum
Alertgroups	1	5479	92004
Service	242	4335	10951
Plaats labels	74	3234	5406

Figuur 26. Min, Max en Cumulatief van dummy features



Figuur 27. Shap summary plot van de top 15 meest invloedrijke alertgroups.

Het overzicht uit Figuur 27 kan leiden tot verder verfijning van het model door meer vergelijkbare features toe te voegen, of deze op een temporale manier uit te breiden. Waar er nu enkel gekeken is naar de totale tijdlijn van een node is het mogelijk om de categorische features uit te breiden op een manier dat deze voor een meer recentelijke tijdspan betrekking hebben, bijvoorbeeld de aggregatie van het type alarmen van de afgelopen 30 dagen.

7. Conclusies

In dit hoofdstuk wordt nogmaals kort antwoord gegeven op de onderzoeksvragen zoals uitgeschreven in de Probleemomschrijving en worden de belangrijkste bevindingen en vervolgacties genoemd.

De eerste onderzoeksvraag is "*Welke aanwezige data is geschikt voor het voorspellen van incidenten*". Op deze vraag is in delen antwoord gegeven. Allereerst is de aanwezige data omschreven in sectie 2.2 en is de data vervolgens gekoppeld tot een enkele tabel in sectie 4.2.1. Op basis van deze tabel is er in sectie 5.4.3. een dataset geprepareerd met omschrijvende variabelen voor een voorspellingsmodel.

De voorspellingsmodellen die zijn bestudeerd zijn uitvoerig omschreven in het theoretische kader. Hier zijn zowel modellen als Markov modellen als Survival modellen beschreven waarvan beide categorieën in aanmerking komen. Op basis van het theoretische kader is er gekozen voor een XGBoost model met als doelfunctie de Cox Proportional Hazard. Dit model is in staat geweest om goede evaluatie score te halen op de toets dataset en de behaalde een C-index van 0.85. Dit betekent dat het model met goede nauwkeurigheid kan voorspellen wat de verhoogde Hazard Ratio is van een gegeven node.

Er is tevens gebruik gemaakt van een Kaplan-Meier en Nelson-Aalen model om een inzicht te krijgen in de algemene Survival functie en Hazard Rate. Hier was naar voren gekomen dat de Hazard Rate zeer hoog is op de eerste dag en daarna snel afneemt om vervolgens relatief laag te blijven. Dit duidt op een Early Infant rate en geeft aan dat de nodes binnen het OTN eerder vroegtijdig een incident ondervinden, dan dat dat op de lange termijn een incident ondervinden. Dit betekent dat een node waar onderhoud aan gepleegd is extra zorgvuldig in de gaten gehouden dient te worden in de eerste twee dagen na het onderhoud.

Als laatste model is er gewerkt met een Markov-keten, die gebruikt is om onderzoek te doen naar de sequentie van alertgroups. Hieruit is gebleken dat 105 van de 181 alertgroups direct een incident kunnen veroorzaken en dat de overige nooit direct een incident hebben veroorzaakt in het verleden. Wel hebben sommige van deze alertgroups een zeer grote kans om direct over te gaan naar een alertgroup die wél in staat is om een incident te veroorzaken. Daarom is het dus mogelijk om minder aandacht te besteden aan de overige 76 alertgroups.

Het Markov model geeft deels antwoord op de onderzoeksvraag "*Hoe ontwikkelen incidenten zich binnen het netwerk van KPN?*". De vervolganalyse op het survival model geeft voor het overige deel antwoord op deze onderzoeksvraag. Er zijn een groot aantal variabelen gebruikt voor het opstellen van het survivalmodel. Zo is gebleken dat hoe meer uur een node al geen event heeft veroorzaakt, hoe waarschijnlijker het is dat deze ook geen event doet veroorzaken. Verder is er gebleken dat de temperatuur van de dag een bijzonder grote invloed heeft op de kans op falen van een node, tegen de verwachtingen in. Aansluitend blijkt het ook dat de originele severity van een event niet simpelweg genegeerd kan worden als deze een lage originele severity heeft. Een niet te negeren aandeel van de nodes heeft een verhoogde kans op falen, ondanks dat het meest recente event een lage originele severity heeft gehad. Verder is er ook gebleken dat er over het algemeen gesteld kan worden hoe meer alerts een node heeft gehad, hoe waarschijnlijker het is te stellen dat deze node een event veroorzaakt dat leidt tot een incident.

De resultaten van het Markovmodel en het Survival model geven tevens deels antwoord op de laatste onderzoeksvraag: "*Hoe kunnen deze resultaten gebruikt worden door KPN?*". Door het model in gebruik te nemen en de mate van invloed voor een voorspelling te gebruiken in de besluitvorming kan er proactief gehandeld worden op de invloedrijke variabelen.

Het antwoord op de onderzoeksvraag: "*Hoe ontwikkelen incidenten zich binnen het netwerk van KPN?*" is deels beantwoord door de analyse van de Markov-keten en deels door de vervolg analyse van het survival model en is zo zijn de belangrijkste bevindingen dat er 81 alertgroups nooit direct tot een incident zullen leiden maar eerder tot een van de 102 alertgroups die het meest waarschijnlijk overspringen naar een incident, dan over naar een andere alertgroup die weer naar een incident doorspringt. Verder is uit het survival model gebleken dat de drie meest invloedrijke variabelen t.w. het aantal uur sinds de laatste fout, de temperatuur van de dag op een incident en het gemiddelde aantal uur sinds de laatste fout, de resterende variabelen en de bijbehorende invloedrijkheid antwoord geven op de vraag hoe fouten zich ontwikkelen door het netwerk. Er is niet één enkele reden, maar een veelvoud aan oorzaken die samenhangen.

Concluderend, het is mogelijk om met een XGBoost model met een Cox PH model als doelfunctie, een C-index van 0.85 te kunnen voorspellen of een node een vergrote kans op een incident heeft op basis van historische gegevens.

8. Aanbevelingen

Tijdens het onderzoek zijn meerdere mogelijke vervolgstappen vastgelegd. In dit hoofdstuk worden de meest impactvolle vervolgstappen opgesomd en toegelicht.

8.1. Meer variabelen toevoegen aan het Survival model

Ondanks de behaalde nauwkeurigheid van het XGBoost model is het alsnog wenselijk om deze nog nauwkeuriger te maken. Maar naast de nauwkeurigheid is het wellicht eerder gewenst om meer informatie toe te voegen aan het model en vast te stellen hoeveel invloed deze hebben op het veroorzaken van incidenten. Zodoende kunnen onderbuikgevoelens getoetst worden en kan er nieuwe stuurinformatie verkregen worden.

8.2. In gebruik nemen van het model

De modellen zoals deze zijn opgezet kunnen een grote meerwaarde toevoegen aan de huidige operatie vanwege de getoetste nauwkeurigheid. Door het model op te nemen in de infrastructuur van KPN kan er een email naar Netcool worden verzonden die dan wordt opgenomen als een apart event in het event systeem en wordt zodoende weergeven bij het SQC aan de medewerkers die de Netcool events verwerken. Op deze manier zou KPN dan direct kunnen overgaan naar actie.

In deze email zou dan de voorspelde Hazard Ratio staan, de mate van invloed van features die verantwoordelijk zijn voor de verhoogde Hazard Ratio, de verwachte levensduur als per het Kaplan-Meier model en de waarschijnlijke kansen op vervolg incidenten voor die gegeven node op basis van het Markov model. Vanwege de grote hoeveelheid informatie is het belangrijk dat iemand die dezelfde inhoudelijke kennis bezit ook kan ondersteunen in de informatie voorziening.

8.3. Vervolg onderzoek doen naar de invloedrijke variabelen

Er zijn een groot aantal variabelen gebruikt in het XGBoost model, hier uit is gebleken dat een deel van deze variabelen een invloed heeft. De reden waarom deze variabelen een dusdanige invloed hebben is nog niet geheel vastgesteld. Om hier zekerheid over te krijgen dient er vervolgonderzoek uitgevoerd te worden naar de oorzaak van de invloed van deze variabelen.

8.4. Model opschalen naar andere domeinen

Op dit moment is de informatie die de modellen hebben verwerkt beperkt gebleven tot de informatie die beschikbaar is in het OTN. Er is echter gebleken dat de temperatuur van de dag bijzonder veel invloed heeft op de kans op veroorzaken van een incident van een node, iets waarvan getwijfeld werd of dit een invloed zou hebben. De oorzaak ligt hiervan vermoedelijk in een ander netwerkdomein. Vanwege het succes van dit model, en de schijnbare invloeden van buitenaf geeft dit een goede reden om zowel meer informatie toe te voegen aan dit model, als ook om dit type model te toetsen op bruikbaarheid in de andere domeinen.

8.5. Voorgestelde acties herleiden uit Astrid en meenemen in email

In de Astrid tickets wordt zorgvuldig bijgehouden welke acties zijn ondernomen om een incident te verhelpen. Op basis van deze historische acties kan een incident gekoppeld worden met een oplossing. Dit model kan gebruikt worden om een suggestie te doen aan de operator hoe om te gaan met een verhoogd risico, nog voor er een incident is geweest. Zo hoeft de operator minder interpretatie te verrichten aan de voorspelde waardes en variabele invloeden, en kan deze eerder direct overgaan naar actie.

9. Bronnenlijst

Breiman, L., J. Friedman, R. Olshen, C. Stone. (1984). *Classification and regression trees.* Wadsworth Books, 358.

Boughorbel, S., Jarray, F., & El-Anbari, M. (2017). *Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric.* PLOS ONE, 12(6)

Chan, G.K., Asgarpoor, S. (2005). *Optimum Maintenance Policy with Markov Processes*. Elsevier.

Cheng, L. (2016). *A gentle introduction to Gradient Boosting.* College of Computer and Information Science North eastern University

Cox, D. R. (1972). *Regression Models and Life-Tables.* Journal of the Royal Statistical Society. Series B.

Documentatie XGBoost Model. (z.d.). *XGBoost Parameters, Learning Task Parameters.* Geraagdpleegd op 01/05/2019, van https://xgboost.readthedocs.io/en/latest/parameter.html

Frisk, E., Krysander, M. & Larsson, E. (2014). *Data-Driven Lead-Acid Battery Prognostics Using Random Survival Forests*. Annual Conference of the Prognostics and Health Management Society 2014.

Hossain, M. K., Shahrir, M. S., Yusof, M. I. M., Yusof, Z., & Asraf, N. M. (2017). *Predictive maintenance of network elements using Markov model to reduce customer trouble tickets*. 2017 IEEE Conference on Big Data and Analytics (ICBDA).

Kaplan, E. L., & Meier, P. (1958). *Nonparametric Estimation from Incomplete Observations. Journal of the American Statistical Association, 53(282), 457–481.*

Kearn, M.. Valiant, L. (1989). *Crytographic limitations on learning Boolean formulae and finite automata.* Symposium on Theory of Computing.

Kearns, M. (1988). *Thought on Hypothesis Boosting.* University of Pennsylvania

Klutke, G., Kiessler, P. C., & Wortman, M. A. (2003). *A critical look at the bathtub curve. IEEE Transactions on Reliability, 52(1), 125–129.*

Li, P. (2010). *Robust Logitboost and adaptive base class (ABC) Logitboost.* In Proceedings of the Twenty-Sixth Conference. Annual Conference on Uncertainty in Artificial Intelligence(UAI'10). 302–311.

Liao, H., Zhao, W., & Guo, H. (2006). *Predicting remaining useful life of an individual unit using proportional hazards model and logistic regression model.* RAMS '06. Annual Reliability and Maintainability Symposium.

Lundberg, S.M. Erion, G.G., Lee, S. (2018). *Consistent Individualized Feature Attribution for Tree Ensembles.* Follow-up to 2017 ICML Workshop.

Matz, S. M., Votta, L. G., & Malkawi, M. (2002). *Analysis of failure and recovery rates in a wireless telecommunications system*. Proceedings International Conference on Dependable Systems and Networks.

Nan, Y., Ming Chai, K. , Sun Lee, W., and Leong Chieu. H. (2012). *Optimizing f-measure: A tale of two approaches*. arXiv preprint arXiv:1206.4625.

Pölsterl, S., Navab, N., & Katouzian, A. (2015). Fast Training of Support Vector Machines for Survival Analysis. Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD, Porto, Portugal, Lecture Notes in Computer Science, vol. 9285, pp. 243-259

Pölsterl, S., Navab, N., & Katouzian, A. (2016). An Efficient Training Algorithm for Kernel Survival Support Vector Machines. 4th Workshop on Machine Learning in Life Sciences, Riva del Garda, Italy

Pölsterl, S., Gupta, P., Wang, L., Conjeti, S., Katouzian, A., & Navab, N., (2016) Heterogeneous ensembles for predicting survival of metastatic, castrate-resistant prostate cancer patients. F1000Research, vol. 5, no. 2676

Reichel, K., Bahier, V., Midoux, C., Masson, J., Stoeckel, S. (2014). *Interpretation and approxmiation tools for big, dense Markov Chain transition matrices in ecology and evolution.* Elsevier

Rodríguez, G. (2007). *Lecture Notes on Generalized Linear Models*.

Salfner, F. (2019). *Predicting Failures with Hidden Markov Models*. Humboldt University Berlin

Salfner, F. Malek, M. (2007). *Using Hidden Semi-Markov Models for Effective Online Failure Prediction*. 26th IEEE International Symposium on Reliable Distributed Systems

Schapire, R. E. (1990). *The Strength of weak learners.* Machine Learning. **5** (2): 197–227.

Statistical Tools for high-throughput data analysis (STHDA*). (z.d.). Cox Proportional-Hazards model.* Geraadpleegd op 01/05/2019 van http://www.sthda.com/english/wiki/cox-proportional-hazards-model

Tianqi, C., Guestrin, C. (2016). *XGboost: A scalable Tree Boosting System.* KDD '16 Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 785-794

Uno, H., Cai, T., Pencina, M. J., D'Agostino, R. B., & Wei, L. J. (2011). On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. Statistics in Medicine.

Wang, C., Huang, T.-Z., & Ching, W.-K. (2014). *A New Multivariate Markov Chain Model for Adding a New Categorical Data Sequence. Mathematical Problems in Engineering,*

Witten, J. Tibshirani, H., (2013), Springer. *An Introduction to Statistical Learning with Applications in R*, pp. 37-39, 231, 303-321