2019•2020

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Het gebruik van machine learning voor het detecteren en genereren van DGA-domeinnamen

PROMOTOR :

ing. Frank APPAERTS

PROMOTOR :

Dhr. Andy GERAERTS

Denzel Vanrompay

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

Gezamenlijke opleiding UHasselt en KU Leuven

UHASSELT KU LEUVEN

UHASSELT KU LEUVEN

2019•2020

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Het gebruik van machine learning voor het detecteren en genereren van DGA-domeinnamen

PROMOTOR :

ing. Frank APPAERTS

PROMOTOR :

Dhr. Andy GERAERTS

Denzel Vanrompay

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

UHASSELT

KU LEUVEN

Deze masterproef werd geschreven tijdens de COVID-19 crisis in 2020. Deze wereldwijde gezondheidscrisis heeft mogelijk een impact gehad op de opdracht, de onderzoekshandelingen en de onderzoeksresultaten.

Woord vooraf

Deze scriptie vormt het sluitstuk van mijn opleiding als master in de industriële ingenieurswetenschappen elektronica-ICT aan UHasselt en KU Leuven. Tijdens de realisatie van deze thesis heb ik zeer veel bijgeleerd en mijn kennis verder kunnen verbreden en uitdiepen. Deze thesis zorgde voor een boeiend en uitdagend traject dat ik de afgelopen maanden heb afgelegd. Dit was niet mogelijk geweest zonder bepaalde personen die ik graag wil bedanken.

Eerst en vooral wil ik graag Cegeka bedanken om mij de kans te geven deze masterproef te realiseren. In het bijzonder zou ik hierbij mijn promotor Andy Geraerts alsook Jimmy Cleuren willen bedanken voor de dagelijkse begeleiding van deze masterproef. Hun begeleiding zorgde voor de juiste aanpak van voorgevallen problemen en vraagstukken. Bij uitbreiding wil ik ook het hele NS-team bedanken voor de steun en het werken in een prettige werkomgeving. Voor de juiste praktische regelingen van dit eindwerk het afgelopen jaar wil ik graag interne promotor ing. Frank Appaerts bedanken.

Een speciale dank gaat uit naar mijn familie en vriendin voor hun steun wanneer dit nodig was. Mijn ouders zou ik hierbij ook willen bedanken om mij de kans te geven deze studie aan te vatten en tot een goed einde te brengen.

Inhoudsopgave

Lijst van tabellen

Lijst van figuren

Verklarende woordenlijst

Bias	Negatieve invloed van externe factoren op de uitkomsten van een onderzoek
Botnet	Netwerk van overgenomen computers door een hacker
C&C-server	Command & Control-server, gebruikt door een hacker voor botnet-verbindingen
Cloud orchestration	Automatisch implementeren en beheren van cloud platformen of applicaties
Compiler	Computerprogramma dat een in een brontaal geschreven programma vertaalt in een semantisch equivalent programma in een doeltaal
CPU	Central Processing Unit
CSV	Comma Seperated Values, door komma's gescheiden waardes
CUDA	Compute Unified Device Architecture
cuDNN	CUDA Deep Neural Network
Cybersecurity	Beschermen van computers, servers, netwerken en gegevens tegen aanvallen
DGA	Domain Generation Algorithm
DNS	Domain Name System
DoH	DNS-over-HTTPS
DoT	DNS-over-TLS
GAN	Generative Adversarial Network
GPU	Graphical Processing Unit
HTTPS	HyperText Transfer Protocol Secure
Intel	Bedrijf gespecialiseerd in het ontwerpen en produceren van chips
IP	Internet Protocol
Knn	K-nearest neighbor, machine learning algoritme
Machine learning	Studie van algoritmes die zichzelf verbeteren via ervaringen
Malware	Overkoepelende term voor elk type computersoftware met kwaadaardige bedoelingen
MLTK	Machine Learning ToolKit
MPN	Mobile Private Networks
Multi-cloud	Gebruik van meerdere cloud en opslagservices in een heterogene architectuur
Necurs DGA	Het Necurs-botnet is een van de meest beruchte malware-botnets die tot nu toe bekend is en waarvan wordt aangenomen dat ze miljoenen computers hebben geïnfecteerd voordat Microsoft een gecoördineerde campagne voerde om het in maart 2020 neer te halen.
Neuraal netwerk	Systemen die gebaseerd zijn op de werking van de hersenen van de mens
Outsourcing	Uitbesteding van diensten aan een derde partij

Private cloud	Afgezonderde omgeving waar alleen u toegang toe hebt en mee kunt werken
Public cloud	Model waarbij betaald wordt voor gebruikte diensten en resources
RAM	Random Access Memory
Splunk	Splunk Inc. is een Amerikaans bedrijf dat software produceert voor het zoeken, monitoren en analyseren van door machines gegenereerde data.
Supervised learning	Gecontroleerd leren met behulp van een trainingset
SVM	Support Vector Machine
TLD	Top Level Domain, extensie van een website
TLS	Transport Layer Security
Unsupervised learning	Ongecontroleerd leren zonder de bekende uitgangswaardes
URL	Uniform Resource Locator
Vals positieven	Data waarvoor een positieve klasse werd voorspeld maar in werkelijkheid een negatieve klasse moet zijn
Vals negatieven	Data waarvoor een negatieve klasse werd voorspeld maar in werkelijkheid een positieve klasse moet zijn
VM	Virtual Machine
WGAN	Het Wasserstein GAN (WGAN) is een algoritme geïntroduceerd in een paper en onderzoekt methoden voor het nastreven van resultaten in machine learning-projecten.

Abstract

Het IT-bedrijf Cegeka is een technologische dienstverlener en begeleidt en levert software- en infrastructuuroplossingen, end-to-end IT-oplossingen met bijbehorende kwalitatieve diensten & consultancy aan zijn klanten. Cegeka beheert heel veel netwerkinfrastructuur. Het wordt alsmaar belangrijker deze infrastructuur goed te beveiligen. Een van de methoden is het detecteren van Domain Generation Algorithm (DGA) domeinnamen die via DNS-verzoeken gebeuren. Malware kan deze domeinnamen gebruiken voor botnetcommunicatie.

Deze studie is onderverdeeld in twee delen.

Het eerste deel is het bouwen van een detectiesysteem voor het detecteren van DGA-domeinnamen. Een machine learning-model detecteert DGA-domeinnamen in DNS-logboeken. Het machine learning-model is geïmplementeerd in Splunk. Cegeka gebruikt Splunk voor het verzamelen van netwerkdata van hun systemen.

Het tweede luik van dit project is het genereren van DGA-domeinnamen met behulp van een Generative Adversarial Network (GAN). Deze gegenereerde domeinnamen zijn door het GAN zo gebouwd zodat ze het detectiesysteem zouden omzeilen.

Uit de resultaten van de door machine learning gegenereerde domeinnamen is het detectiesysteem aangepast. De trainingset is onderverdeeld in een reeks DGA-domeinnamen en een reeks legitieme domeinnamen. Het aantal DGA-domeinnamen in de trainingset werd vergroot. Nieuwe testen tonen dat door een GAN gegenereerde domeinnamen deels kunnen gedetecteerd worden.

Abstract in English

The IT company Cegeka is a technological service provider and provides software and infrastructure solutions, end-to-end IT solutions and related qualitative services & consultancy to its clients. Cegeka manages a great deal of network infrastructure. It is becoming increasingly important to secure this infrastructure properly. One of the methods is to detect Domain Generation Algorithm (DGA) domain names that occur via DNS requests. Malware can use these domain names for botnet communication.

This study is divided into two parts.

The first part is building a detection system for detecting DGA domain names. A machine learning model detects DGA domain names in DNS logs. The machine learning model is implemented in Splunk. Cegeka uses Splunk to collect network data from their systems.

The second part of this project is generating DGA domain names using a Generative Adversarial Network (GAN). These generated domain names are built by the GAN in such a way that they would bypass the detection system.

From the detection results of the machine learning generated domain names, the detection system was adapted. The training set is divided into a set of DGA domain names and a set of legitimate domain names. The number of DGA domain names in the training set was increased. New tests show that domain names generated by a GAN can be partially detected.

1. Inleiding

Cegeka is een toonaangevende Europese aanbieder van IT-oplossingen. Het IT-bedrijf is gespecialiseerd in *multi-cloud* oplossingen, *cloud orchestration*, *outsourcing*, *cybersecurity* en de optimalisatie van applicaties, infrastructuur en bedrijfsprocessen. Daarnaast focust Cegeka zich op het opzetten van *mobile private networks* (MPN's) in een businessomgeving.

Het familiebedrijf werd in 1992 opgericht door André Knaepen. Anno 2020 staat CEO Stijn Bijnens aan de leiding van de onderneming vanuit het hoofdkantoor in Hasselt. De afgelopen jaren kende het IT-bedrijf een sterke groei. Cegeka noteerde een jaarlijkse omzetstijging van 15 tot 20%. In 2019 telde de groep bijna 5.000 medewerkers en behaalde het een geconsolideerde omzet van 561 miljoen euro. Vandaag telt Cegeka ruim 6.000 medewerkers, verspreid over heel Europa, met vestigingen in de Benelux, Duitsland, Oostenrijk, Roemenië, Italië, Tsjechië, Slovakije, Frankrijk, Rusland en sinds kort ook Moldavië [1].

Het bedrijf bouwt oplossingen op maat voor klanten en beheert hun softwaresystemen vanuit onder andere hun *private cloud* met eigen datacentra of in de *public cloud.* De data van de klanten wordt vandaag de dag sneller verwerkt en meer opgeslagen dan ooit ervoor. Via *machine learning* zou het mogelijk kunnen zijn om patronen te herkennen in deze data. Hackers zouden op deze manier grote hoeveelheden data over een bepaald systeem kunnen laten analyseren en hiermee een systeem kunnen aanvallen. Bedrijven zouden anderzijds machine learning kunnen inzetten om gaten in hun eigen beveiliging te vinden en proactief te kunnen werken alvorens er misbruik kan van worden gemaakt.

De opdracht voor deze masterproef zal erin bestaan om het domein van *Domain Generation Algorithm* (DGA) domeinnamen te onderzoeken en hoe machine learning hierin kan worden gebruikt zowel als 'schild' tegen hackers of als 'wapen' door hackers.

Het detecteren van DGA-domeinnamen met behulp van machine learning gebeurt in dit project met een app in *Splunk* die de mogelijkheid geeft een set domeinnamen te importeren, te analyseren en eventueel vals positieven aan te passen.

Het genereren van DGA-domeinnamen gebeurt eveneens met behulp van machine learning. Hiervoor wordt een *Generative Adversarial Network* (GAN) gebruikt die door de combinatie van neurale netwerken in een *discriminator* en *generator* op basis van een invoerlijst DGA-domeinnamen zal genereren.

In het tweede hoofdstuk is de literatuurstudie uitgewerkt die een overzicht geeft van de verschillende vormen van machine learning die van belang zijn voor deze masterproef alsook de verschillende machine learning algoritmes die aan bod zijn gekomen tijdens het uitwerken van deze masterproef. In de literatuurstudie is ook uitleg terug te vinden over DNS gezien hackers gebruik maken van DGA-domeinen die door een DNS-server verwerkt worden. De uitleg over het genereren van DGA-domeinnamen met behulp van een GAN is terug te vinden in §2.4.2 van de literatuurstudie.

Het derde hoofdstuk behandelt het materiaal en de methode die gebruikt zijn voor het uitwerken van dit project. In hoofdstuk vier worden de resultaten besproken en de werkingsmethode die voor dit project van toepassing is. Dit gaat over de verschillende stappen die mogelijk zijn voor het detecteren van DGA-domeinen alsook over hoe het genereren van DGA-domeinnamen werd opgezet.

Ten slotte is in het vijfde hoofdstuk de conclusie van dit project terug te vinden die de resultaten en eventueel toekomstig werk bespreekt.

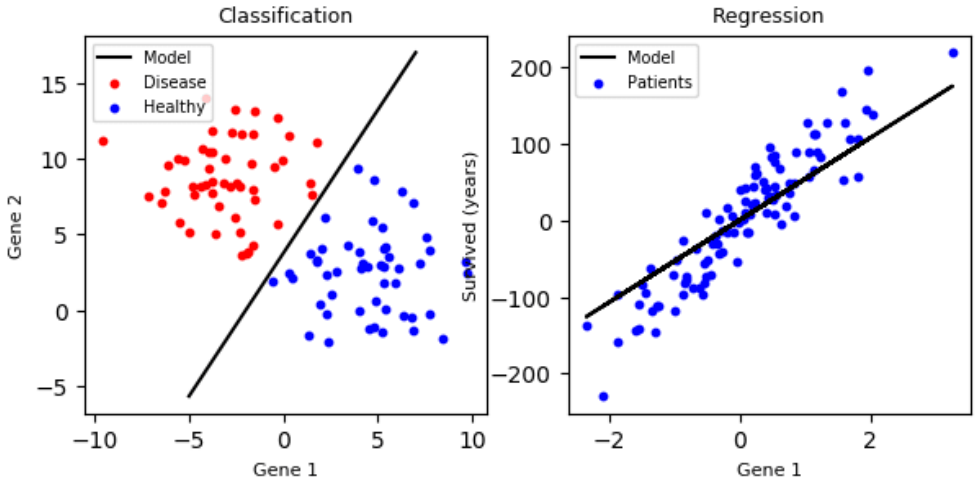
2. Literatuurstudie

2.1 Machine Learning

Machine learning wordt omschreven als toepassing van artificiële intelligentie waarbij systemen de mogelijkheid bieden om te leren en zich automatisch te verbeteren uit ervaringen zonder hiervoor expliciet geprogrammeerd te zijn[2]. Machine learning is opgedeeld in meerdere domeinen met gecontroleerd en ongecontroleerd leren als voornaamste domeinen.

2.1.1 Gecontroleerd leren

Voor gecontroleerd leren (*supervised learning*) wordt het model opgebouwd via een dataset bedoeld voor het trainen van het model. Deze trainingsdata moet op voorhand klaargemaakt worden om het model op de verschillende eigenschappen te trainen en ook te kunnen testen. Tijdens het trainen zal het algoritme zoeken naar patronen in de verschillende eigenschappen (*features*) van de data die overeenstemmen met de gewenste uitgangsdata. De klaargemaakte trainingset wordt meestal opgesplitst in trainingsdata en validatiedata. De verhouding voor het opsplitsen van de trainingset in trainingsdata en validatiedata kan zelf bepaald worden. Meestal wordt er aangeraden om minstens 50% van de data in de trainingset te gebruiken als trainingsdata, anders kan het model onvoldoende worden getraind. Met de validatiedata in de trainingset zal het algoritme controleren hoe accuraat het model is getraind en hoeveel vals positieven of vals negatieve resultaten er zijn [3]. In dit project zal vooral het aantal domeinen dat als legitiem geclassificeerd is, maar toch een DGA-domein is, van belang zijn. Na het trainen van het model zal het gecontroleerd algoritme in staat zijn om de uitgangswaarde te voorspellen voor nieuwe ongeziene ingangsdata. Het algoritme zal dit bepalen aan de hand van de eerdere trainingsdata. Het doel van een gecontroleerd geleerd model is om voor nieuwe ingangsdata de correcte uitgangswaarde te voorspellen. Meerdere technieken zijn mogelijk voor gecontroleerd leren, dit kan zowel via classificatie als via regressie zoals te zien op figuur 1 [4].



Figuur 1: Illustrerende figuur voor classificatie en regressie [5]

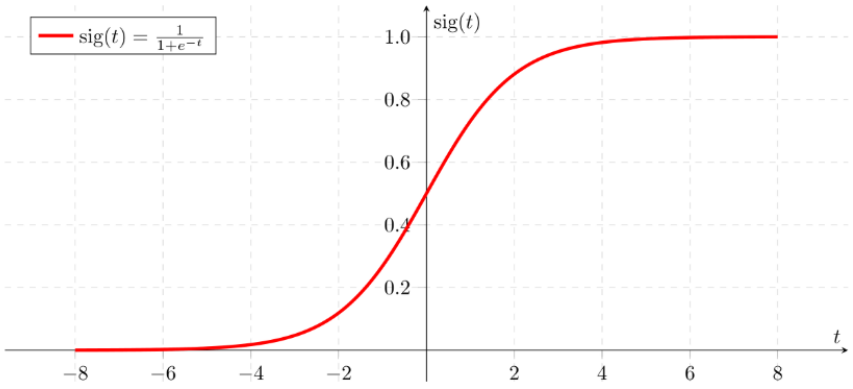
2.1.2 Classificatie in gecontroleerd leren

Bij gebruik van een classificatie algoritme zal het algoritme de ingangsdata proberen op te delen in verschillende categorieën of klassen. Om de classificatie juist te voorspellen is trainingsdata nodig die ervoor zorgt dat het model nieuwe data correct zal herkennen. Binnen classificatie zijn er verschillende types algoritmes zoals: logistische regressie, *support vector machine* (SVM), beslisboom, *random forest* en *k-nearest neighbor* (knn) [6].

2.1.2.1 Logistische Regressie

Logistische regressie wordt in tegenstelling tot wat de naam doet vermoeden niet meteen als regressie gebruikt maar eerder als classificatiealgoritme. Voor dit project zal binaire logistische regressie de gewenste manier van logistische regressie zijn gezien hier het type ofwel DGA ofwel legitiem is.

De voorspelde waarde is afhankelijk van de hypothese die een bepaalde waarde als voorspelling voorstelt. Deze hypothese wordt toegepast in een sigmoid functie om te bepalen of er een 0 of een 1 moet voorspeld worden zoals te zien op onderstaande afbeelding [7].



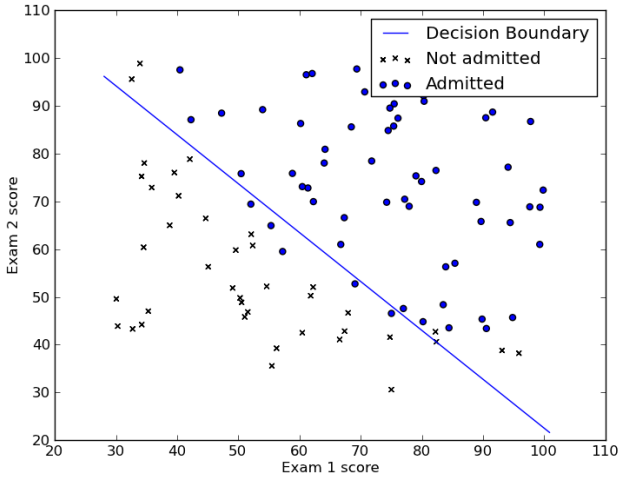
Figuur 2: Sigmoid functie voor logistische regressie [7]

De sigmoid functie ziet er als volgt uit: $g(z) = \frac{1}{1+e^{-z}}$ met $z = \theta^T x$ en de hypothesis $h_\theta(x) = g(\theta^T x)$. De x-waarde geeft de ingangswaarde weer voor het bepaalde voorbeeld. θ wordt vastgelegd bij het minimaliseren van de costfunctie. In een costfunctie wordt berekend hoeveel de voorspelde waardes afwijken van de werkelijke waardes. De costfunctie voor logistische regressie ziet er als volgt uit met m het aantal trainingsvoorbeelden, n het aantal eigenschappen van een trainingsvoorbeeld en λ een regularisatieparameter die wordt ingesteld om te grote invloeden van bepaalde eigenschappen te beperken [8].

$$\min_\theta \frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)}\left(-\log h_\theta(x^{(i)})\right) + (1-y^{(i)})\left((-\log(1-h_\theta(x^{(i)}))\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Figuur 3: Costfunctie logistische regressie

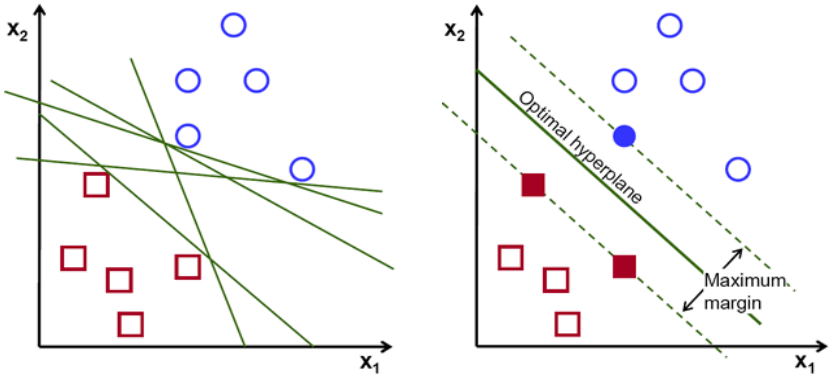
Om te bepalen tot welke klasse bepaalde invoerdata hoort, wordt een bepaalde *threshold* ingesteld. Deze wordt de *decision boundary* genoemd en kan vergeleken worden met een bepaalde grenswaarde vanaf waar het algoritme zal beslissen dat de uitgang voorspeld kan worden als een 1. Deze grenswaarde wordt gewoonlijk bij de start ingesteld op 0,5 zodat het systeem vanaf de waarde 0,5 een 1 zal voorspellen voor de uitgang. Voor situaties waar bepaalde positieve voorspellingen bepaalde consequenties kunnen hebben zoals bij kankervoorspelling kan deze grenswaarde worden ingesteld op 0,9 zodat het systeem minstens 90% zeker is dat de waarde positief zal zijn. Op onderstaande afbeelding is een voorbeeld van de decision boundary terug te vinden. In dit geval wordt een eerstegraads decision boundary gebruikt. Voor andere toepassingen kan het nodig zijn de graad hiervan te verhogen [9].



Figuur 4: Decision boundary bij logistische regressie [10]

2.1.2.2 Support Vector Machine

Een Support Vector Machine of SVM-algoritme zal net zoals bij logistische regressie een grens tussen de positieve en negatieve datapunten proberen te zoeken. Een SVM-algoritme zal ook proberen een zo groot mogelijke marge te bekomen tussen de positieve en negatieve datapunten.



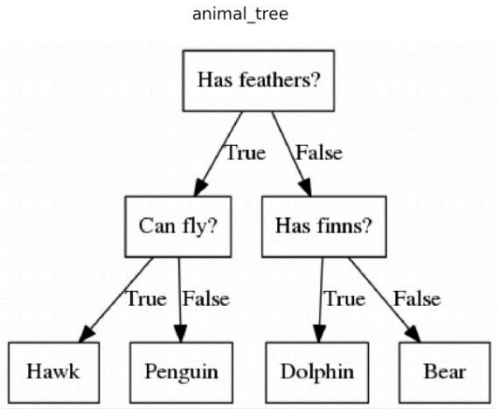
Figuur 5: Bepalen van de decision boundary bij SVM

Zoals op bovenstaande figuur te zien zal SVM verschillende decision boundaries opstellen en trachten de decision boundary met de grootste afstand tussen de twee klassen te zoeken. SVM's hebben ook een C-parameter, die wordt gebruikt voor het bepalen hoe strikt het algoritme de verdeling gaat opstellen. Een grote C zal een lagere *bias* maar een hoge variantie geven en een kleinere C zal een grotere bias maar een kleinere variantie geven [8].

SVM zal voor een positieve uitgangswaarde streven naar $\theta^T x \geq 1$ en voor negatieve uitgangswaarde streven naar $\theta^T x \leq -1$. De waarde van $\theta^T x$ kleiner dan -1 of groter dan 1 wordt gebruikt om een grotere marge te bekomen dan wanneer in beide gevallen respectievelijk kleiner of groter dan 0 gebruikt wordt [8].

2.1.2.3 Beslisboom

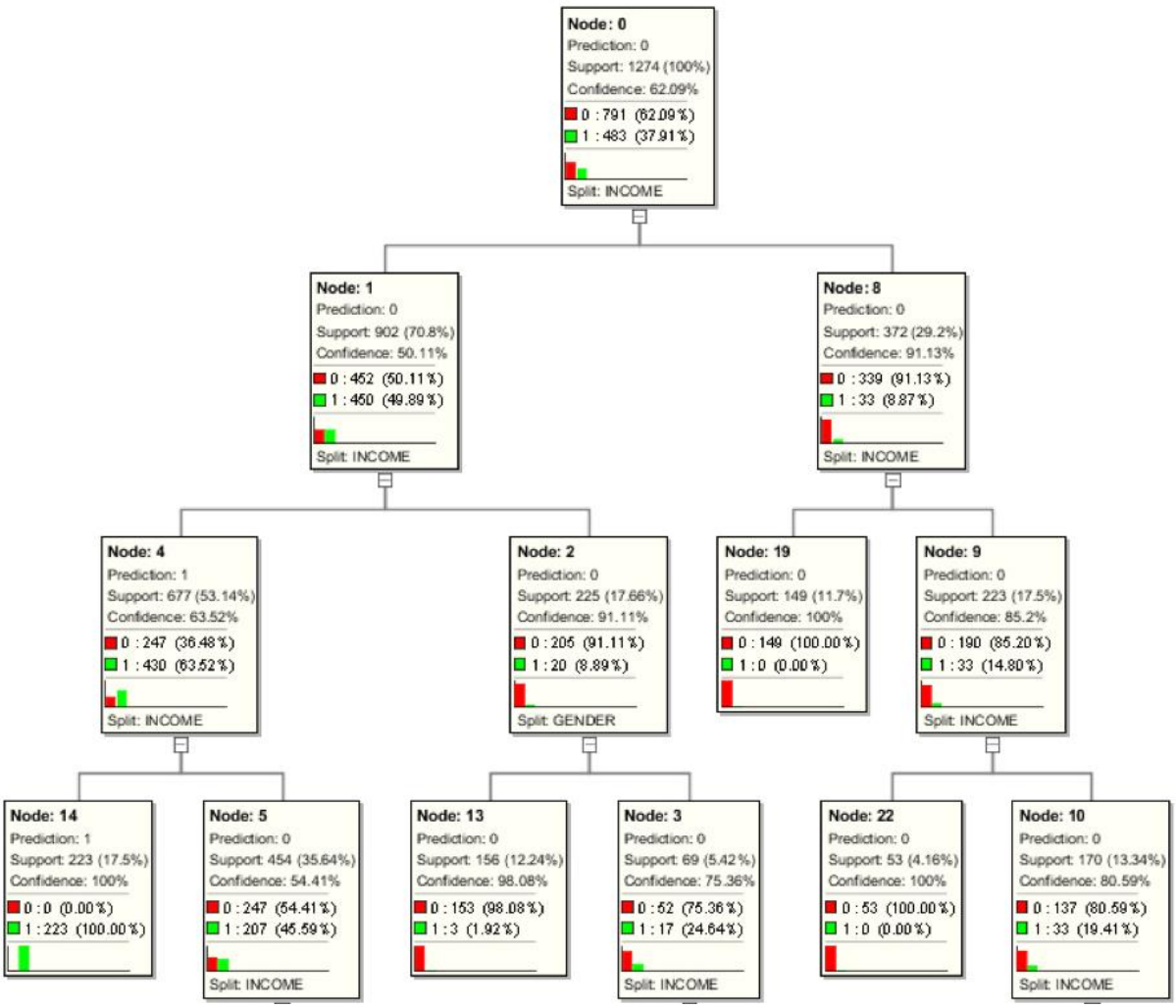
Het volgende algoritme heeft een werkingsprincipe dat begrijpelijker is en is voorgesteld op onderstaande afbeelding met als voorbeeld het voorspellen van dieren.



Figuur 6: Voorstelling beslisboom

Het algoritme zal aan de hand van de trainingsdata een beslisboom opstellen en deze daarna toepassen op de testdata en nieuwe data. De belangrijkste taak van het algoritme zit erin de cost van elke aftakking juist te bepalen zodat een bepaalde groep vergelijkbare data hetzelfde pad zal afleggen.

De Gini-score geeft een zicht hoe goed de split van een bepaalde tak is. Voor gevallen van classificatie wordt deze bepaald via: $G = \sum pk * (1 - pk)$ waar pk de proportie van dezelfde data is. Voor binaire classificatie ligt deze best rond 0,5. Het stoppen met vertakken kan ingesteld worden door bijvoorbeeld het minimum aantal voorbeelden dat aan een bepaalde tak moet worden toegewezen te bepalen of door de maximale diepte van de beslissingsboom in te stellen. De performantie van een beslisboom kan worden verbeterd door vertakkingen waar te weinig training voorbeelden zijn of de takken met te weinig belang te verwijderen, dit wordt *pruning* genoemd [11].

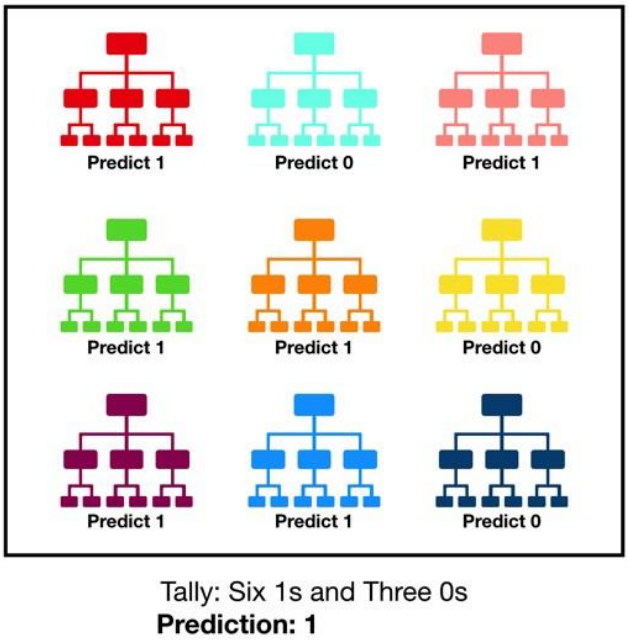


Figuur 7: Beslisboom met eigenschappen [12]

Bovenstaande afbeelding toont een beslisboom met meerdere vertakkingen en meer informatie over het aantal voorbeelden dat op de bepaalde tak van toepassing is alsook de betrouwbaarheid van de split met de groene en rode balken [12].

2.1.2.4 Random Forest

Random Forest is gebaseerd op het beslisboom algoritme maar in plaats van één enkele beslisboom zoals in bovenstaand algoritme zullen er verschillende beslisbomen worden opgebouwd tot er als het ware een “bos” van beslisbomen ontstaat. Voor classificatieproblemen haalt het algoritme op dit moment meestal goede resultaten.

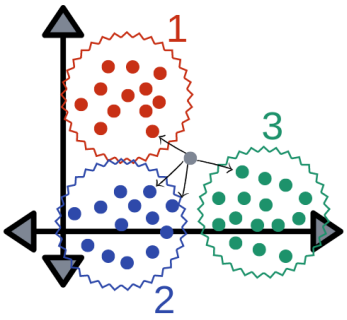


Figuur 8: Voorstelling Random Forest [13]

Bovenstaande afbeelding geeft een voorstelling van de verschillende kleine beslisbomen die gebruikt worden voor het bepalen van de uiteindelijke voorspelde klasse. De voorspelde klasse wordt bepaald door de uitgangen van de verschillende kleine beslisbomen op te tellen. In de bovenstaande voorstelling is er zes keer een 1 voorspeld en drie keer een 0. De uiteindelijke voorspelde klasse is dus een 1. Random Forest heeft een goede performantie omdat de verschillende beslisbomen beschermd zijn tegen hun individuele fouten. Sommige beslisbomen zullen de juiste klasse voorspellen, anderen de verkeerde. Het hele “bos” is wel in staat om met het gezamenlijke resultaat de juiste klasse te voorspellen [13].

2.1.2.5 K-nearest neighbors (knn)

Het knn-algoritme kan zowel op classificatie als op regressieproblemen worden toegepast. Zoals de naam al doet vermoeden zal knn de klassen van omliggende punten in rekening nemen bij het voorspellen van een klasse. Bij het invoeren van een nieuw datapunt zoals het grijze punt in onderstaande figuur zal het algoritme de afstanden bepalen tot de verschillende klassen en het punt aan de klasse toekennen waar deze het dichtst bij ligt [14].



Figuur 9: Knn klasse werkingsprincipe [14]

Het algoritme berekent de euclidische afstand tussen elk datapunt en de testdata en berekent dan de waarschijnlijkheid dat een bepaald datapunt bij een bepaalde klasse hoort. De formule om de euclidisch afstand te berekenen is als volgt [14]:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2}$$

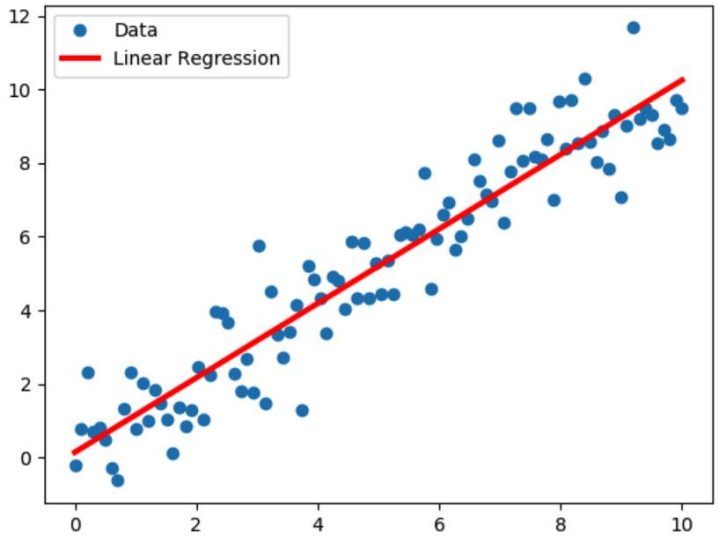
Voordelen van knn zijn dat het algoritme makkelijk te implementeren is, er niet te veel parameters in te stellen zijn en dat het veelzijdig is waardoor het zowel voor classificatie als voor regressie en clustering in ongecontroleerd leren kan toegepast worden. Een nadeel van knn is dat het algoritme snel veel meer rekenkracht zal vragen naarmate het volume aan data toeneemt. Dit zorgt ervoor dat het algoritme opmerkelijk trager begint te werken bij grote volumes data [15].

2.1.3 Regressie in gecontroleerd leren

Modellen die gebruik maken van regressie worden gebruikt voor het bepalen van continue waardes zoals bijvoorbeeld het voorspellen van de prijs van een huis met bepaalde eigenschappen. Net als voor classificatie is voor regressie een set trainingsdata nodig. Voor regressie zijn er verschillende mogelijkheden van algoritmes zoals eenvoudige lineaire regressie maar ook andere algoritmes die reeds hierboven besproken werden voor classificatie kunnen voor regressiedoeleinden gebruikt worden [16].

2.1.3.1 Lineaire regressie

Lineaire regressie wordt in verschillende domeinen gebruikt. Het stelt de relatie voor tussen twee of meer datapunten waarbij er een datapunt wordt voorspeld op basis van de andere gekende datapunten [17]. De datapunten van de trainingset worden uitgezet over de verschillende features, er zal door het algoritme getracht worden een bepaalde hypothese of functie doorheen de datapunten de vinden. Deze functie kan gaan van een eerstegraadsfunctie tot ook hogere graadsfuncties. Voor een eerstegraadshypothese is de vergelijking: $h(\theta) = \theta_0 + \theta_1 x$. Onderstaande figuur geeft een voorbeeld van de hypothese van de eerste graad [18].

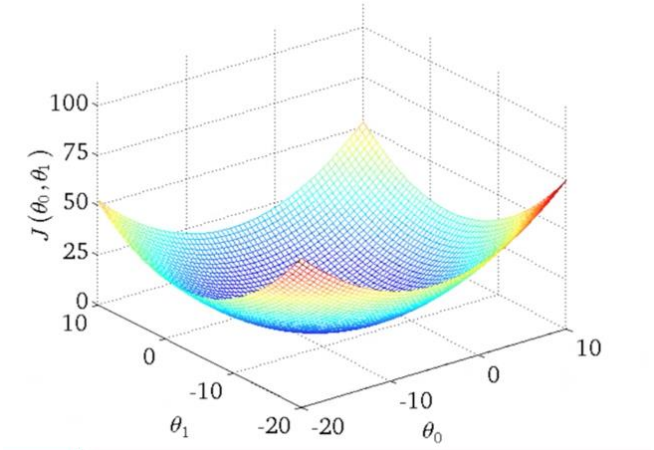


Figuur 10: Hypothese lineaire regressie [19]

Bij het bepalen van de hypothese is het belangrijk de cost zo laag mogelijk te houden. Dit is de afwijking van een voorspelde waarde tot de werkelijke waarde [18]. De costfunctie wordt berekend via onderstaande formule.

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Figuur 11: Costfunctie lineaire regressie [18]



Figuur 12: Voorstelling costfunctie [20]

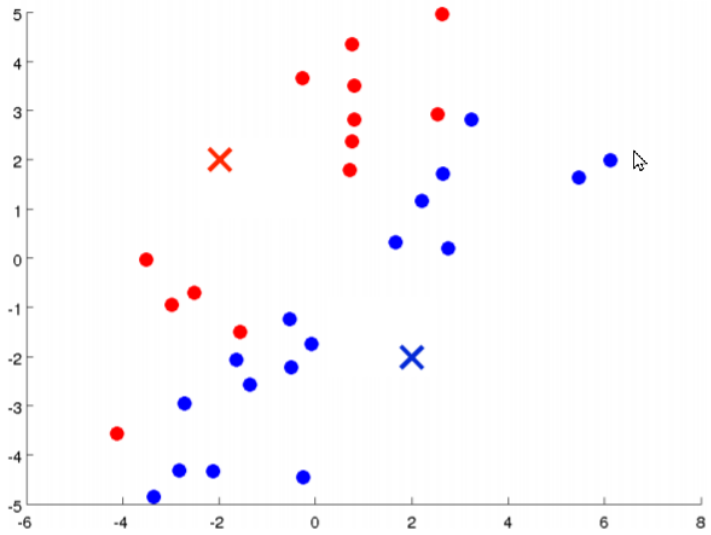
Bovenstaande figuur toont de voorstelling van een costfunctie. Om de juiste waardes θ_0 en θ_1 voor de hypothese te vinden is het belangrijk het minimum van de costfunctie te vinden en hiervan de θ-waarde in te vullen in de vergelijking van de hypothese. Dit zal een goede hypothese en dus goede voorspellingen opleveren [18].

2.1.4 Ongecontroleerd leren

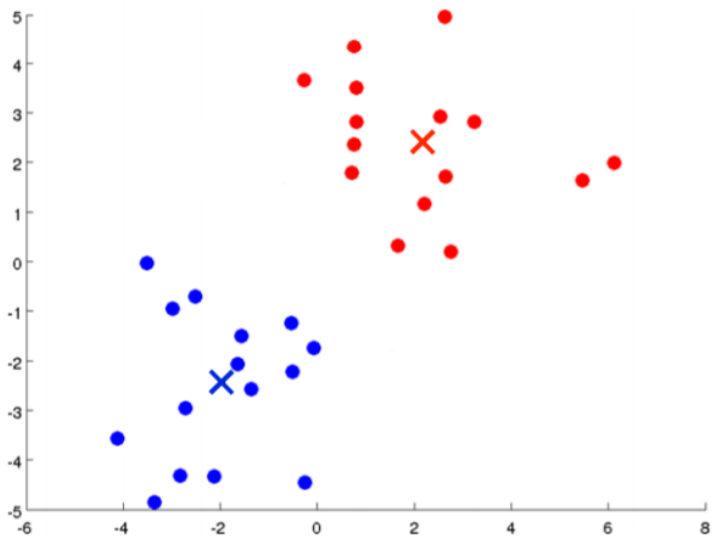
In het domein van ongecontroleerd leren zullen vooral algoritmes gebruikt worden die de data clusteren. Anders dan met gecontroleerd leren heeft het algoritme tijdens het trainen geen uitgangswaarde ter beschikking. Het algoritme zal hierbij bepaalde patronen of groepen in de data proberen te vinden [21].

2.1.4.1 K-means clustering

Een voorbeeld van een clustering algoritme is k-means clustering. Het algoritme start door een aantal punten toe te wijzen als centerpunten voor de cluster. Hierna zal de afstand tot de andere punten berekend worden en het punt aan het dichtstbijzijnde clusterpunt toewijzen. Daarna zal de gemiddelde locatie van de punten uit een bepaalde cluster bepaald worden en zal het centerpunt verplaatst worden naar dit gemiddelde. Het algoritme zal dit proces een aantal keer blijven herhalen tot het centerpunt niet meer verplaatst moet worden tot het gemiddelde en dus samenvalt met de gemiddelde locatie van de cluster. Onderstaande figuren geven een voorbeeld van een start- en eindpositie voor de clusters van de datapunten [22].

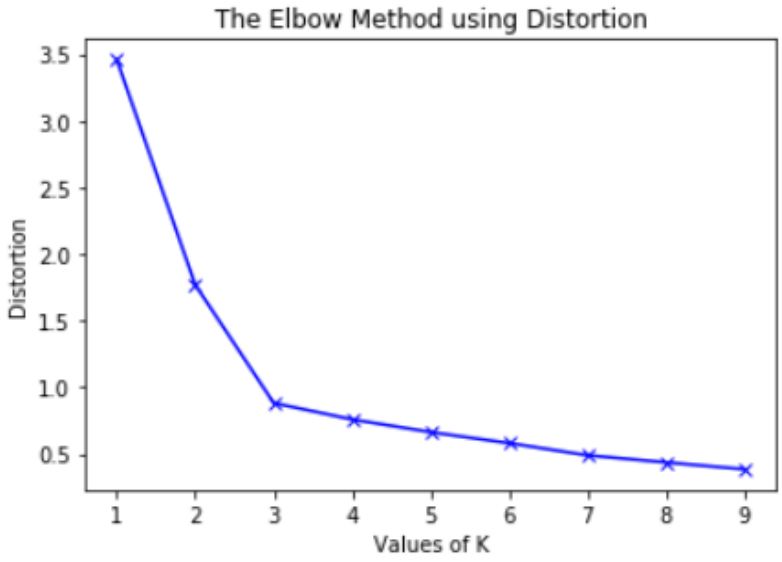


Figuur 13: k-means na een enkele clustering [22]



Figuur 14: k-means eindsituatie [22]

Als startpunten kunnen een aantal willekeurige datapunten gekozen worden. Dit aantal is gelijk aan het aantal gewenste clusters. Het aantal clusters kan bepaald worden via de elleboog methode. Een voorbeeld hiervan is terug te vinden op onderstaande figuur en toont dat de cost niet sterk afneemt bij meer dan drie clusters. Het meest gunstige aantal clusters is in dit geval drie [22].



Figuur 15: Bepalen van het aantal clusters bij k-means clustering [23]

2.1.5 Domeinen in cyberveiligheid voor machine learning

Machine learning kan worden gebruikt in meerdere domeinen van cyberveiligheid. Deze domeinen worden meestal onderverdeeld in volgende domeinen: indringerdetectie, malware analyse en spam en phishing detectie [24].

2.1.5.1 Intrusiedetectie

Deze systemen hebben het doel om onrechtmatige activiteiten binnen een netwerk te ontdekken. Deze systemen zijn gebaseerd op patronen van anomalie detectie, dreigingsdetectie en classificatie op basis van machine learning. Via deze intrusiedetectie kunnen twee specifieke problemen worden geanalyseerd: de detectie van botnets en van DGA-domeinnamen [24].

2.1.5.2 Malware analyse

Moderne malware genereert verschillende soorten malware met hetzelfde kwaadwillige effect en doet dit met verschillende uitvoerbare bestanden. Machine learning kan gebruikt worden om deze malware te analyseren en deze in de juiste categorie malware te plaatsen [24].

2.1.5.3 Spam en phishing detectie

Dit domein gebruikt verschillende technieken voor de detectie van ongewenste e-mails. Machine learning kan hier worden toegepast voor het verbeteren van de detectie van ongewenste e-mails [24]. Phishing is een vorm van ongewenste e-mail, hierbij wordt een e-mail van een vertrouwde organisatie nagebootst om de gebruiker bepaalde software te laten installeren of bepaalde vertrouwelijke informatie door te geven via malafide websites [25].

2.1.5.3.1 Social engineering

Zoals besproken in [26] is *social engineering* een hulp voor phishing. Hierbij worden ontvangers, meestal via e-mail, in de val gelokt door de afzender van het bericht om bepaalde informatie of toegang te geven aan de afzender. Het kan hierbij gaan over het verlenen van wachtwoorden of toegang tot de computer of netwerk.

Een andere trend binnen het versturen van phishing e-mails is *spear phishing*. Dit verschilt met reguliere phishing omdat het gericht is op een bepaalde groep personen zoals een bepaald team in een bedrijf. Deze vorm van phishing wordt vooral gebruikt voor het oplichten van gebruikers en het stelen van gevoelige bedrijfsinformatie zoals intellectuele eigendommen door bedrijfspionnen. Binnen spear phishing spreekt men ook van '*whaling'*. Hierbij worden de phishing e-mails gericht op personeel in hogere rangen van het bedrijf. Deze personeelsleden hebben meestal toegang tot gevoeligere bedrijfsinformatie en hebben hogere rechten op hun account binnen de systemen van het bedrijf [26].

Bij het gebruik van social engineering bij phishing past de afzender meestal een van volgende technieken toe:

- autoriteit: de e-mail lijkt hier te komen van een persoon of instituut met een bepaald gezag;
- social proof: de ontvanger wordt aangemoedigd om de actie uit te voeren omdat vele anderen ook al aan de actie hebben deelgenomen;

- schaarste: de informatie in de e-mail suggereert dat een bepaald aanbod is beperkt en dat de gebruiker maar een bepaalde tijd heeft om te reageren of dat er maar een beperkt aantal plaatsen beschikbaar is.

2.1.6 Phishing en gebruik van machine learning

Phishing is de laatste jaren een veelgebruikte techniek voor het installeren van malware binnen particuliere en bedrijfsnetwerken. Zoals hierboven reeds beschreven kan via phishing malware worden geïnstalleerd op het toestel van de gebruiker of wordt vertrouwelijke informatie zoals codes doorgespeeld aan de hackers via malafide websites [25].

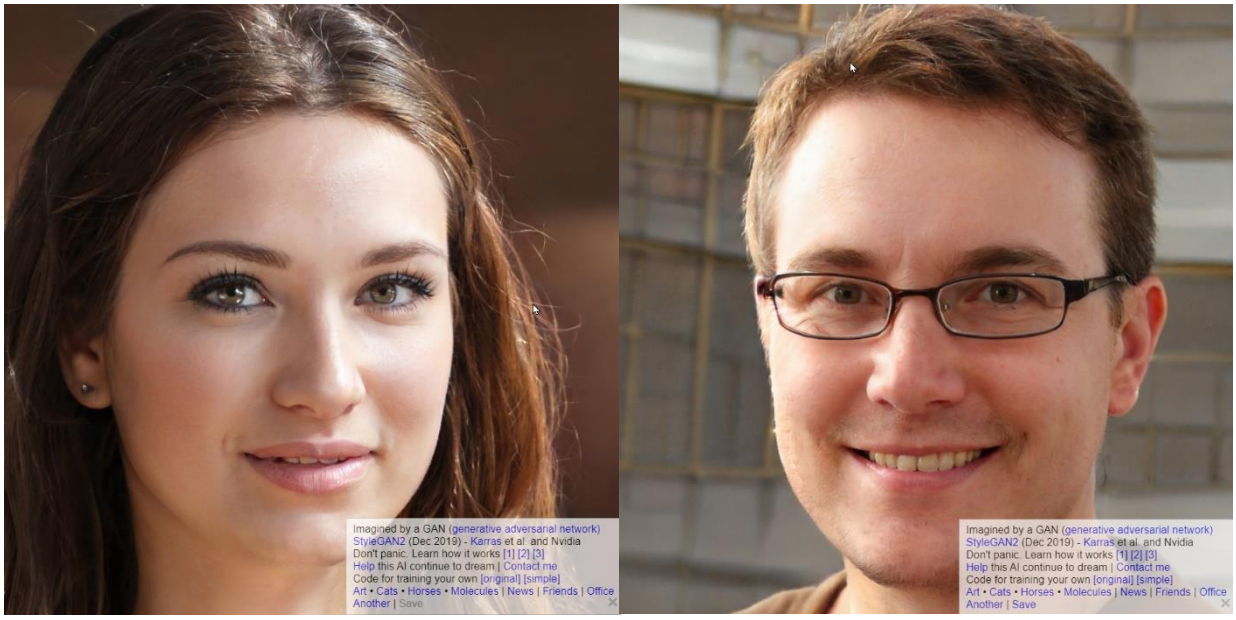
Met machine learning kan een phishing e-mail worden geanalyseerd op verschillende onderdelen. Een trainingsdataset zal het model trainen om daarna nieuwe e-mails te kunnen analyseren op phishing. Het detecteren van phishing e-mails wordt steeds moeilijker gezien de kwaliteit van deze e-mails steeds beter wordt. Een deel van de analyse gebeurt door het analyseren van de URL die de gebruiker zou moeten openen. Machine learning kan meer dan 15 features bepalen voor het detecteren van phishing e-mails [27]. Phishing kan een voorzet zijn voor de toepassingen die in deze masterproef worden besproken. Eens de phishing succesvol is vanuit het standpunt van de hacker zou deze malware kunnen installeren op het toestel van het slachtoffer. Deze malware kan via DGA-domeinnamen communiceren met de *Command and Control* (C&C) server van de hacker voor het aansturen van het botnet [28], hierover is in §2.4 van deze masterproef meer informatie terug te vinden.

2.2 Generative Adversarial Networks

Voor het genereren van domeinnamen met behulp van machine learning wordt in deze masterproef gebruikt gemaakt van een Generative Adversarial Network of GAN. Een GAN is in staat om op basis van bepaalde invoer een soort “verbeeldingskracht” na te bootsen en probeert eigen creaties en ideeën te scheppen [29].

2.2.1 Gebruik van een GAN

Er bestaan zeer uiteenlopende toepassingen voor een GAN. De meest gekende of meest voorkomende toepassing is het nabootsen of creëren van afbeeldingen of foto’s die nauwelijks van echt te onderscheiden zijn. Een voorbeeld hiervan is terug te vinden op de website www.thispersondoesnotexist.com. Op deze website wordt telkens een nieuwe foto weergegeven die werd gebouwd door een GAN. Afgeleiden van deze website tonen ook dat deze technologie kan toegepast worden voor het genereren van tekst zoals bijvoorbeeld scenes uit de reeks Friends. Dit laatste zal voor dit project interessanter zijn gezien er legitieme domeinnamen zullen nagebootst worden als DGA-domeinnamen. Onderstaande foto’s zijn enkele voorbeelden van de website thispersondoesnotexist.com.



Figuur 16: Voorbeelden GAN foto’s van thispersondoesnotexist.com

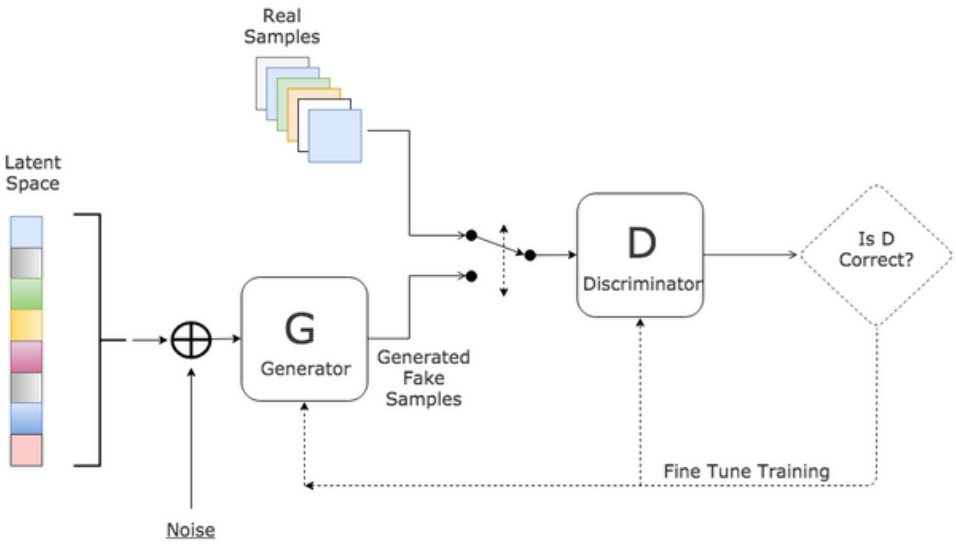
2.2.2 Werkingsprincipe van een GAN

Om met een GAN-systeem een realistische output te genereren gebruikt het twee concurrerende neurale netwerken. Voor het eerste neurale netwerk spreekt men van een discriminator. Aan dit netwerk wordt een grote set trainingsdata gegeven. In dit project zal dit een hele lijst legitieme domeinnamen zijn. Zo kan de discriminator leren hoe een legitieme domeinnaam eruitziet [29].

Het tweede neurale netwerk dat in een GAN-systeem gebruikt wordt noemt men de generator. Deze zal proberen uitgangsdata te maken waarvan de discriminator moet denken dat deze thuishoort in de

legitieme of oorspronkelijke dataset. De generator zal dus valse data aan de discriminator aanbieden en laten lijken alsof dit echte data is door het aan de juiste criteria van echte data te laten voldoen. De discriminator moet proberen de valse data te ontdekken [29].

De generator heeft geen aanknopingspunt en weet dus niet waar het moet beginnen voor het genereren van data, voor dit project domeinnamen. Het zal daarom beginnen met ruisdata of maar enkele karakters. De discriminator zal vervolgens deze data beoordelen en bepalen of de gegenereerde data realistisch genoeg is vergeleken met de oorspronkelijke dataset. De discriminator zal in het begin daarom zeer veel invoer vanuit de generator afkeuren [29]. Een voorbeeld van de werking van het systeem is ook te zien in onderstaande figuur.



Figuur 17: Voorbeeld werking GAN [30]

Zoals te zien op de figuur met het werkingsprincipe van een GAN vindt er ook een terugkoppeling plaats van de discriminator naar de generator. Deze terugkoppeling zorgt ervoor dat de generator data kan genereren van een hogere kwaliteit die steeds dichter aanleunt bij de oorspronkelijke data [29].

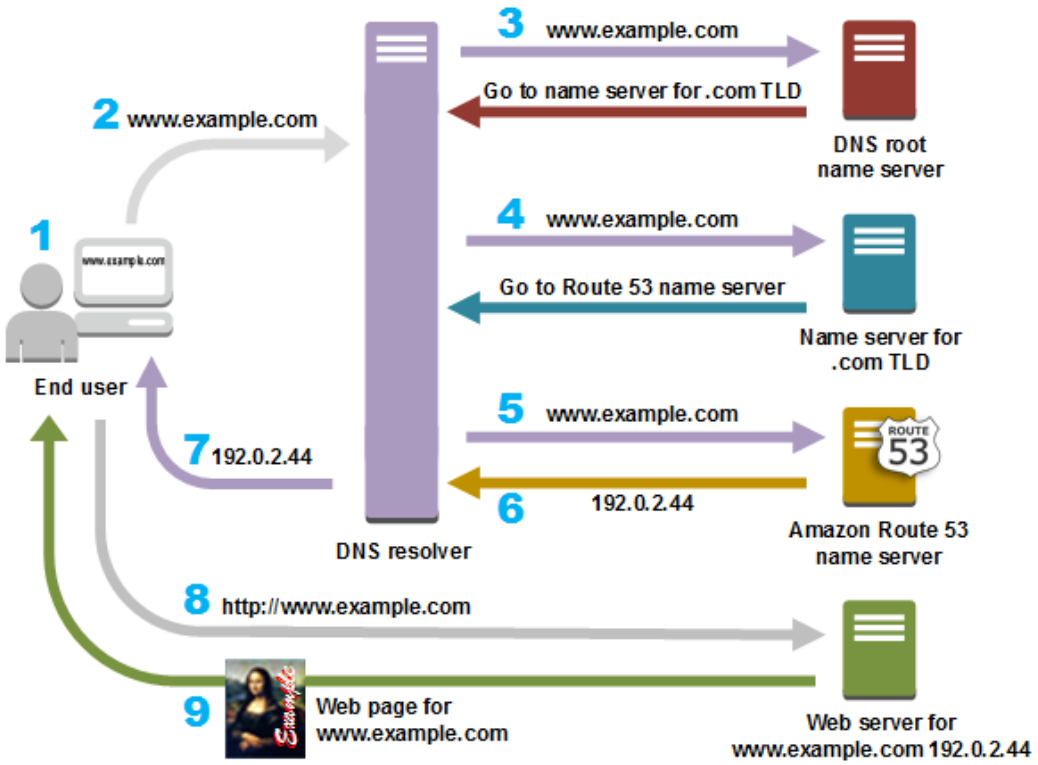
Uiteindelijk zal de data zo realistisch overkomen dat ook mensen niet meer in staat zijn het onderscheid te kunnen maken tussen de oorspronkelijke data en de valse gegenereerde data. Het systeem is afgeleid van wat mensen doen wanneer men zijn verbeeldingskracht gebruikt. Mensen bedenken ook nieuwe beelden en nieuwe ideeën op basis van gekende beelden en gebeurtenissen. Een GAN-systeem komt op deze manier dichter bij de werking van menselijke verbeeldingskracht [29].

2.2.2.1 Wasserstein GAN

Het GAN-type dat in dit project gebruikt wordt is Wasserstein GAN of WGAN. Het GAN-type werd ontworpen door verschillende onderzoekers onder andere uit de Facebook AI Research groep en lost een aantal problemen op die kunnen voorkomen bij het trainen van een GAN. Het WGAN heeft een opmerkelijke verbetering in de stabiliteit van het leerproces en beschikt over verschillende leercurves die nuttig kunnen zijn voor het debuggen en het instellen van de hyperparameters om zo op een eenvoudigere manier tot betere resultaten te komen [31].

2.3 Domain Name System

Een Domain Name System (DNS) fungeert als een soort telefoonboek voor het internet. DNS zorgt ervoor dat gebruikers geen ingewikkelde IP-adressen moeten onthouden. In het DNS worden domeinnamen gekoppeld aan de IP-adressen. Zo wijst de domeinnaam “www.uhasselt.be” naar 193.190.2.30. Het domein “www.cegeka.com“ wijst naar 212.113.88.235 [1]. De IP-adressen zijn makkelijk voor een computer om het internetverkeer naar de juiste webserver te sturen [32].



Figuur 18: Werkingsprincipe DNS [33]

Bovenstaande figuur toont de werking van DNS en de verschillende stappen om een gebruiker naar de juiste website of toepassing te sturen. Hieronder worden de verschillende stappen beschreven [33].

1. De gebruiker opent een webbrowser en geeft hierin bijvoorbeeld www.example.com in.
2. Het verzoek voor www.example.com wordt gestuurd naar een DNS-resolver. Deze wordt typisch beheert door de internetprovider of het bedrijfsnetwerk van de gebruiker.
3. De DNS-resolver stuurt het verzoek naar een DNS root name server. Deze server antwoord dat het verzoek in dit geval moet doorgestuurd worden naar de naamserver voor .com domeinen.
4. De DNS-resolver stuurt het verzoek opnieuw door maar dit keer naar de naamserver voor de .com domeinen. De naamserver zal in dit geval antwoorden met de vier naamservers die zijn toegewezen aan het example.com domein. In het voorbeeld Amazon Route 53 naamservers.
5. De DNS-resolver zal dan een naamserver voor het domein kiezen en het verzoek voor example.com hiernaar doorsturen.

[1] DNS Lookup via https://www.whatismyip.com/dns-lookup/ op 11 mei 2020.

6. De naamserver voor example.com kijkt na of het domein voorkomt in zijn zone en zal dan de toegewezen waarde in dit geval het IP-adres voor de webserver 192.0.2.44 antwoorden aan de DNS-resolver
7. De DNS-resolver heeft nu eindelijk het IP-adres gevonden dat de gebruiker nodig heeft voor het bezoeken van de website. De DNS-resolver zal de gegevens tijdelijk bewaren om de volgende keer sneller te kunnen antwoorden op het verzoek.
8. De webbrowser van de gebruiker zal dan een verzoek sturen voor example.com naar het IP-adres dat het kreeg van de DNS-resolver.
9. De webserver of andere bron op 192.0.2.44 zal de webpagina terugsturen voor example.com naar de webbrowser van de gebruiker waarbij de webbrowser zal zorgen dat de webpagina wordt weergegeven.

2.3.1 DNS-over-HTTPS

DNS-verkeer is bijna het enige verkeer dat nog zonder encryptie over het netwerk wordt verstuurd. De meeste websites gebruiken immers vandaag de dag HTTPS om de verbinding tussen de gebruiker en de webserver te versleutelen. Om ook dit stuk verkeer te beveiligen wordt er door o.a. Google en Cloudfare voorgesteld om over te schakelen op DNS-over-HTTPS (DoH). Hierbij wordt het DNS-verkeer ook verstuurd via HTTPS waardoor het ook versleuteld is.

Dit heeft als voordeel voor de gebruiker dat de privacy meer gegarandeerd wordt en het moeilijker is voor hackers om een man-in-the-middle aanval uit te voeren. Bedrijven als Google, Mozilla en Cloudfare die als voortrekkers van DoH worden gezien zetten stevig in op het garanderen van de privacy van de gebruikers.

Bij de invoering van DoH spelen ook verschillen in internetcultuur tussen Amerika en Europa mee. In de meeste Europese landen zijn internetproviders vrij neutraal en sturen ze het verkeer door van A naar B. Voor Amerikaanse internetproviders zit het anders. Zij verdienen veel geld aan het verkopen van de browsegeschiedenis van hun klant voor advertentiedoeleinden. Vooral Amerikaanse internetgebruikers zullen dus baat hebben bij DoH.

Nadelen van DoH zijn dat het voor overheden moeilijker wordt om bepaalde website te blokkeren zoals The Pirate Bay maar ook websites over terrorisme en kindermisbruik. Het gerechtelijk bevel moet dan overgemaakt worden aan de DNS-provider zoals Google of Cloudfare.

Ook kan het nadelig zijn voor bedrijfsnetwerken die hun eigen DNS-resolver gebruiken om bijvoorbeeld bepaalde inhoud zelf te kunnen blokkeren. DNS-verzoeken worden ook vaak gebruikt voor het scannen op malware zoals in dit project. Systeembeveiligers zullen hiermee ook een probleem hebben wanneer ook dit verkeer versleuteld is. Google voorziet ondertussen een functie die DoH uitschakelt wanneer het detecteert dat een bedrijf een eigen DNS-resolver gebruikt.

Ook is er kritiek dat wanneer DoH de standaard zou worden dat DNS, wat op dit moment een grotendeels gedecentraliseerd protocol is, in de handen van slechts een beperkt aantal bedrijven komt die alle DNS-data beheren en hierdoor een monopoliepositie kunnen uitspelen.

DNS-verkeer versleutelen kan niet enkel via HTTPS. Steeds meer experts zijn voorstander om het verkeer niet te versleutelen met HTTPS, maar hiervoor gebruik te maken van TLS. Dit komt als een laag bovenop standaard-DNS. Met DNS-over-TLS of DoT is het ook makkelijker om pakketten te inspecteren en op deze manier malware te scannen of niet gewenste inhoud op een netwerk te blokkeren. DoH werkt met poort 443 die gebruikt wordt voor HTTPS verkeer. DoT werkt met de

unieke poort 853 en is makkelijker te blokkeren zonder dat hiermee een groot deel van het internet geblokkeerd wordt. Experts lijken op dit moment verdeeld te zijn over de beste methode en proberen een afweging te maken van de voor- en nadelen van elke methode [34].

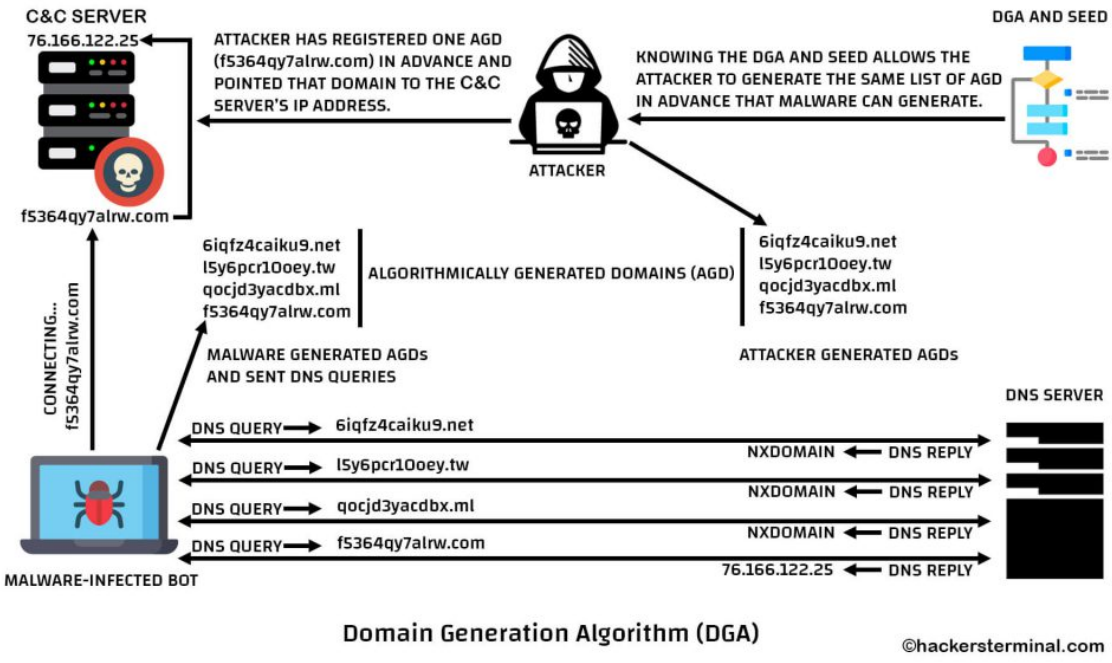
2.4 Domain Generation Algorithms

Een Domain Generation Algorithm of DGA wordt gebruikt door malware en genereert een groot aantal willekeurige domeinnamen. Dit kan gaan tot 1000 domeinen per uur waarvan de aanvaller slechts één domein koppelt aan het IP-adres van de Command and Control (C&C) server die door de aanvaller is opgezet. Alle bots binnen het botnet zullen dan verbinden met deze C&C-server. DGA werd voor het eerst gebruikt door de Kraken malware in 2008 en won sindsdien steeds meer aan populariteit onder hackers [35].

DGA's maken gebruik van een seed. Dit kunnen bepaalde woorden zijn of bijvoorbeeld de datum die de malware uitleest uit de systeemklok. Op basis van deze seed zal de DGA een lijst met DGA-domeinnamen genereren en deze lijst opslaan. Een voorbeeld van een DGA-script is terug te vinden in de bijlagen. In dit project werden 30 verschillende DGA-types getest. Deze gekende DGA's zijn beschikbaar op https://github.com/baderj/domain_generation_algorithms. De werking van een DGA is voor het ene type al complexer dan voor het andere type. Eenvoudigere types zullen enkel voor de eerste karakters willekeurige waardes kiezen, anderen zullen een volledig willekeurige naam ontwikkelen gebaseerd op de seed.

De seed is nodig zodat de aanvaller door gebruik te maken van dezelfde seed ook dezelfde domeinnamen zal verkrijgen. Wanneer de aanvaller dan een domeinnaam registreert weet deze dat deze domeinnaam ook bij het slachtoffer gegenereerd is en deze kan gebruiken voor communicatie met de C&C-server. Wanneer een domeinnaam gedetecteerd wordt en op een zwarte lijst terechtkomt, kan de aanvaller eenvoudig een ander domein uit de DGA-lijst registreren en zo de verbinding met het botnet verder in stand houden.

Een overzicht over de opstelling van het gebruik van DGA-domeinen is terug te vinden op onderstaande figuur.



Figuur 19: Overzichtsfiguur van een DGA-opstelling [35]

De verschillende stappen die de aanvaller gebruikt om een aanval met behulp van DGA-domeinen uit te voeren zijn hieronder beschreven.

1. De aanvaller bouwt het DGA in in de malware en zorgt dat deze malware verspreid geraakt.

2. Doordat de aanvaller de seed kent, weet deze welke domeinnamen de DGA zal genereren en dus voor welke domeinnamen DNS-verzoeken zullen uitgevoerd worden.

3. De aanvaller registreert een domeinnaam uit de lijst gegenereerd door de DGA en wijst deze toe aan het IP-adres van de C&C-server

4. De malware zal ondertussen DNS-verzoeken uitvoeren voor de domeinnamen die gegenereerd werden door de DGA.

5. Voor de geregistreerde domeinnaam van de aanvaller zal de DNS-server het IP-adres antwoorden.

6. De malware maakt verbinding met de C&C-server.

7. De aanvaller heeft nu controle over het systeem van het slachtoffer en kan dit gebruiken voor zijn botnet.

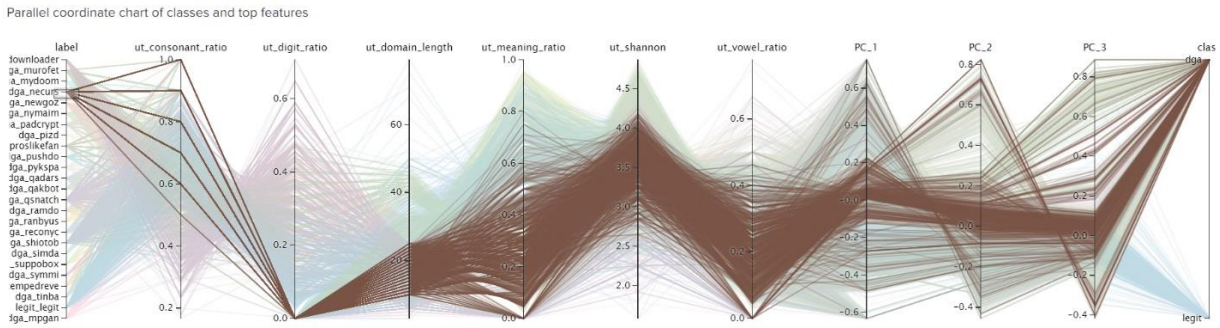
2.4.1 DGA-detectie met machine learning

Wanneer een DGA-domein wordt gedetecteerd, werd dit eerst via reverse engineering ontleed waarna de verschillende domeinen werden geblokkeerd. Doordat er zeer veel domeinen door een DGA worden gegenereerd, is dit vaak een tijdrovende taak.

Machine learning kan hierin een ondersteuning bieden door automatisch de domeinen die via DNS worden opgevraagd te analyseren en hierbij automatisch de gedetecteerde DGA-domeinen te blokkeren. Doordat er een foutmarge zit op de DGA-detectie met machine learning is het belangrijk dat deze methode als ondersteuning wordt gebruikt voor beveiligingspersoneel, maar hun werk kan hiermee wel worden verlicht.

Het detectiesysteem houdt rekening met verschillende eigenschappen van de domeinnaam zoals: de lengte van de domeinnaam, het aantal cijfers in de domeinnaam, het aantal klinkers in de domeinnaam en de Shannon entropie in de domeinnaam [36].

Verschillende machine learning algoritmes die gebruikt kunnen worden voor classificatie kunnen ook gebruikt worden voor het detecteren van DGA-domeinnamen op basis van de verschillende eigenschappen van de domeinnaam. Onderstaande figuur geeft een voorbeeld voor de parallelle coördinaten van Necurs DGA. Op de figuur is per eigenschap van de domeinnaam te zien welke waardes er aangenomen kunnen worden voor dit type DGA. Op basis van de waardes van deze eigenschappen en het pad dat ze afleggen zal het machine learning algoritme, dat instaat voor de detectie van DGA-domeinnamen, voorspellen of het domein een legitiem of een DGA-domein is.



Figuur 20; Parallelle coördinaten voor necurs DGA

2.4.2 Ontwijken van DGA-detectie met machine learning

Wanneer machine learning zou kunnen gebruikt worden voor het detecteren van DGA-domeinen, zou dit ook kunnen gebruikt worden voor het genereren van domeinen die niet kunnen gedetecteerd worden door de machine learning-modellen voor detectie van deze domeinen.

Voor het vermijden van detectie door, machine learning ondersteunde, detectiesystemen bestaan o.a. *Deception DGA* dat gebruikt kan worden voor het genereren van domeinen die niet kunnen gedetecteerd worden als DGA-domeinen door machine learning modellen. Deze methode zou het opnieuw mogelijk maken om DGA-domeinen te gebruiken voor het opzetten van botnets.

Deze methode is gebaseerd op het GAN-systeem dat eerder in deze masterproef werd uitgelegd en als doel heeft uitvoerdata te genereren die niet kan gedetecteerd worden door detectiesystemen en gebaseerd is op legitieme invoerdata.

Er zijn verschillende parameters die kunnen worden ingesteld bij het opzetten van een GAN-systeem. De invoerdata is in dit geval een lijst die dagelijks beschikbaar wordt gesteld door Cisco Umbrella. Deze lijst bevat de 1 miljoen populairste domeinen gebaseerd op meer dan 100 miljard verzoeken per dag met 65 miljoen unieke actieve gebruikers, in meer dan 165 landen [37]. Uit deze lijst met 1 miljoen legitieme domeinnamen worden willekeurig 100.000 domeinnamen gekozen waarop het systeem zich zal baseren om na te bootsen. Dit aantal kan worden ingesteld als parameter. Een grotere waarde kan voor een hogere kwaliteit valse domeinnamen zorgen maar vergroot ook de rekenkracht die nodig is voor het genereren van de domeinnamen.

Andere parameters die kunnen worden ingesteld voor het GAN-systeem zijn de batch grootte, deze is hier ingesteld op 64 en stelt het aantal datapunten voor dat het algoritme verwerkt alvorens een

verbetering van de domeinnaam te geven. Ook het aantal iteraties kan ingesteld worden, dit is het aantal iteraties dat het systeem uitvoert tot het de uiteindelijke uitvoerdata geeft. Hoe hoger dit getal is, hoe hoger de kwaliteit van de gegenereerde domeinnamen. Hier staat tegenover dat het langer zal duren alvorens het systeem alle iteraties verwerkt heeft.

De lambda parameter kan worden ingesteld om te bepalen hoe hard bepaalde afwijkingen moeten worden afgestraft en heeft ook een invloed op de kwaliteit van de uitvoerdata. Een kleinere lambda zal meer variatie in de uitvoerdata veroorzaken [38].

3. Materiaal en methode

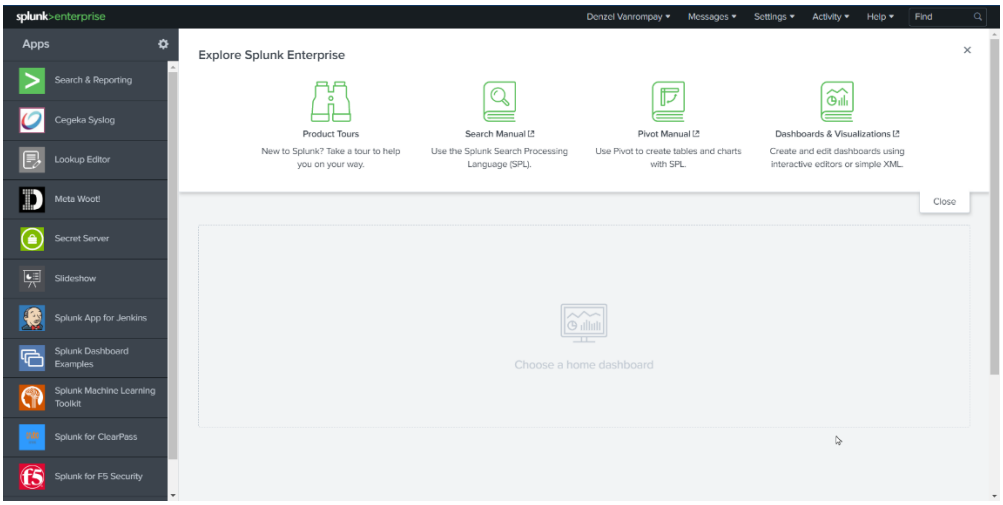
Voor het realiseren van deze masterproef zijn verschillende tools gebruikt. Zo werd Splunk Enterprise gebruikt voor het opzetten van het detectiesysteem van de DGA-domeinnamen. Splunk werd uitgevoerd op een *Virtual Machine* (VM) op een Cegeka server. De DGA-domeinnamen worden gegenereerd met behulp van Anaconda Spyder als programmeeromgeving en Python als programmeertaal. De Tensorflow-bibliotheek voor Python wordt gebruikt bij het genereren van machine learning gebaseerde domeinnamen.

3.1 Virtual Machine

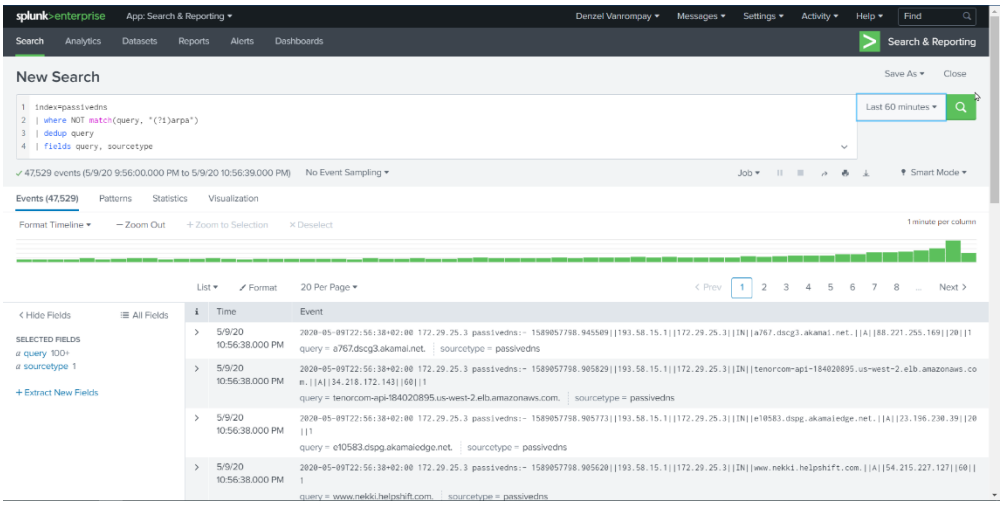
Om het trainen van het model te versnellen werd door Cegeka een VM opgezet om hierop een eigen Splunk omgeving te installeren gescheiden van de operationele Splunk van Cegeka. Als processor voor het VM wordt een Intel Xeon E5-2650 op 2,30GHz gebruikt met 16GB RAM-geheugen. Als harde schijf werd er via VMWare 150GB voorzien wat zeker voldoende is voor deze masterproef.

3.2 Splunk

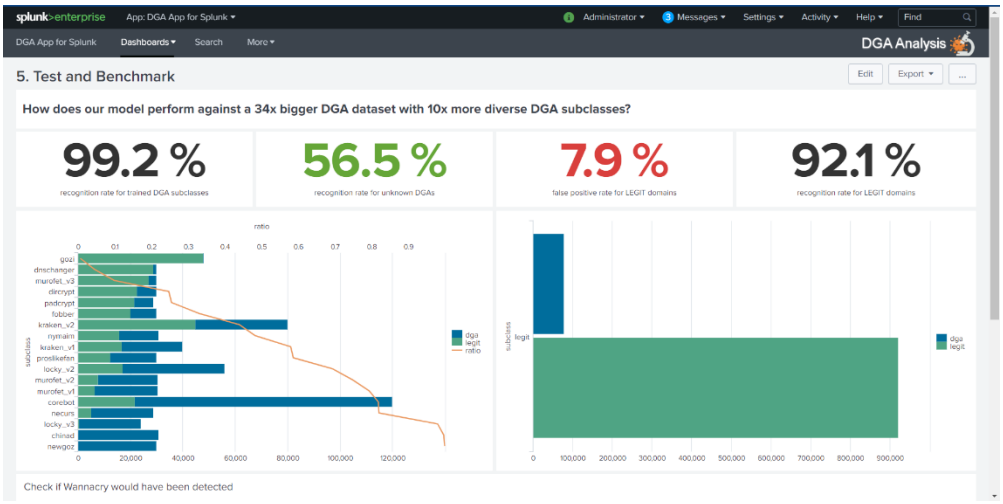
In deze masterproef wordt Splunk Enterprise gebruikt voor het opzoeken en analyseren van data. Splunk Enterprise is een product van Splunk Inc. en wordt door bedrijven vooral gebruikt als datacollector. Het voordeel van Splunk is dat data van veel verschillende bronnen kan worden samengevoegd en in real-time kan geanalyseerd worden [39]. De data die uit de verschillende bronnen wordt geanalyseerd en weergegeven kan gebruikt worden om meer inzicht in de data te krijgen en hierop bepaalde voorspellingen te maken. De data kan eenvoudig inzichtelijk gemaakt worden via eenvoudige visualisaties [40]. Inzicht krijgen in de data gebeurt via het uitvoeren van de benodigde "*searches*". In deze zoekopdrachten kan de index worden meegegeven waar de data moet gevonden worden alsook bepaalde voorwaarden worden opgelegd aan de opzoeking om alleen de gewenste zoekresultaten weer te geven op het dashboard. Onderstaand zijn enkele voorbeelden terug te vinden voor de startpagina, een search en een dashboardvisualisatie.



Figuur 21: Splunk startscherm



Figuur 22: Voorbeeld van een Splunk search



Figuur 23: Voorbeeld van een Splunk visualisatie in een dashboard

3.3 Anaconda

De scripts voor het genereren van DGA-domeinnamen zijn geschreven in de programmeertaal Python. Anaconda Spyder is de programmeeromgeving waarin de verschillende scripts uitgevoerd en bewerkt worden. Ook het GAN dat de DGA-domeinnamen met behulp van machine learning genereert is geschreven met Anaconda Spyder als programmeeromgeving en Python als programmeertaal.

Python is een programmeertaal gemaakt door Guido van Rossum en vrijgegeven in 1991. Het wordt vooral gebruikt bij web- en softwareontwikkeling, data-analyse en scripting. Python werkt op verschillende platformen en heeft een vrij eenvoudige syntax om met een klein aantal lijnen code een programma te schrijven. Het gebruikt ook nieuwe lijnen en een inspringingsstijl voor het voltooien van een bepaalde functie waar andere programmeertalen een puntkomma of haakjes gebruiken [41].

Anaconda is een gratis en open-source programmeeromgeving voor Python en R. Anaconda komt met verschillende pakketten gerelateerd aan machine learning en data analyse. Voor machine learning

worden pakketten zoals TensorFlow, scikit-learn en Theano gebruikt. De data analyse bibliotheken die het populairste zijn, zijn onder andere pandas, NumPy en Dask [42].

3.4 TensorFlow

TensorFlow is een open-sourcebibliotheek voor Python die als doel heeft machine learning met Python makkelijker en sneller te maken. Het werd gecreëerd door het Google Brain team en bundelt functies voor zowel machine learning als deep learning. TensorFlow heeft verschillende functies voor onder andere handschriftherkenning, beeldherkenning, recurrente neurale netwerken en natural language processing. Een van de grootste voordelen van TensorFlow is dat het voor een bepaalde abstractie zorgt die maakt dat de details die nodig zijn voor het implementeren van een machine learning algoritme door TensorFlow worden afgehandeld. Zo kan de ontwikkelaar zich focussen op de algemene logica en implementatie van het algoritme terwijl TensorFlow de berekeningen achter de schermen afhandelt [43]. Voor deze masterproef zal TensorFlow gebruikt worden voor het opzetten van een GAN die het genereren van ML-gebaseerde domeinnamen voor zijn rekening zal nemen.

Er zijn verschillende versies van Tensorflow beschikbaar. Zowel een CPU als een GPU versie. Voor het GAN dat in dit project gebruikt wordt, is de GPU versie aangeraden. Deze versie wordt ondersteund door NVIDIA CUDA en cuDNN.

3.5 NVIDIA CUDA met cuDNN

Zoals hierboven besproken maakt de gebruikte Tensorflow-gpu versie gebruik van NVIDIA CUDA en cuDNN. NVIDIA CUDA werd ontwikkeld om berekeningen van bepaalde soorten applicaties te versnellen door de kracht van een grafische kaart te benutten. Waar een CPU maar enkele kernen kan gebruiken voor de berekeningen is het met de CUDA toolkit mogelijk om de duizenden kernen van de GPU parallel te gebruiken. Met CUDA kunnen ontwikkelaars applicaties ontwikkelen in C, C++, Fortran, Python en MATLAB. De CUDA-toolkit bevat de benodigde GPU-bibliotheke, een *compiler*, ontwikkelingstools en de CUDA *runtime* [44].

CUDA kan worden uitgebreid met bibliotheken zoals de cuDNN-bibliotheek voor het gebruik van *Deep Neural Networks* in een GPU versnelde omgeving. De bibliotheek bevat zeer goed afgestelde implementaties die nodig zijn voor het opzetten van een deep neuraal netwerk. Er kunnen verschillende frameworks gebruikt worden met de bibliotheek waaronder Keras, MATLAB en TensorFlow. Deze laatste wordt in dit project gebruikt.

4. Resultaten

Bij het uitwerken van het detectiesysteem alsook voor het systeem voor het genereren van DGA-domeinnamen zijn verschillende algoritmes onderzocht en met elkaar vergeleken om zo tot het beste resultaat te komen.

4.1 Detecteren van DGA-domeinnamen

Gezien Splunk gebruikt wordt als datacollector voor onder andere DNS-verzoeken werd beslist dat een app binnen Splunk de meest gebruiksvriendelijke en best te integreren optie zou zijn voor het uitwerken van het detectiesysteem.

4.1.1 Opbouw van de trainingset

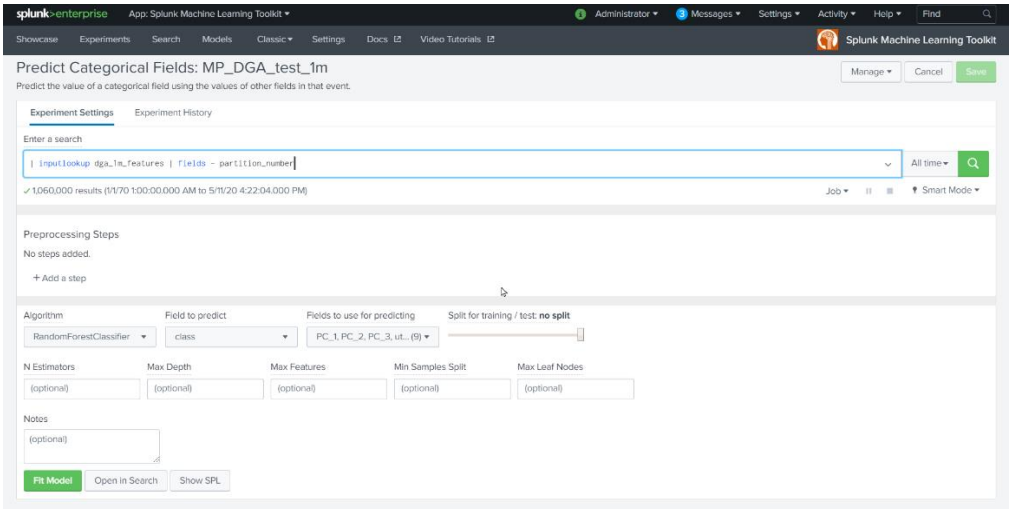
Voor het trainen van het machine learning-model werden verschillende trainingsets aangemaakt. In deze sets werden een aantal DGA-domeinnamen samengevoegd met een aantal legitieme domeinnamen. De legitieme domeinnamen zijn afkomstig van Cisco Umbrella. Voor het genereren van de DGA-domeinnamen werd gekozen om 30 verschillende DGA-types te genereren. Voor de tweede set werd gekozen om een evenredig aantal domeinnamen per type DGA te genereren om zo de resultaten van het model te verbeteren. Een overzicht van het aantal DGA-domeinnamen is terug te vinden in onderstaande figuur. Voor elke hoeveelheid DGA-domeinnamen werden een gelijk aantal legitieme domeinnamen uit de Cisco Umbrella set toegevoegd.

Type	# set1	# set 2
banjori	1000	2000
chinad	256	2000
corebot	1000	2000
fobber (v1+v2)	600	2000
gozi	1000	2000
kraken (v1+v2)	4000	2000
locky (v2+v3)	284	2000
monerodownloader	2500	2000
murofet (v1+v2+v3)	3040	2000
mydoom	999	2000
necurs	4096	2000
newgoz	1000	2000
nymaim (v1+v2)	1728	2000
padcrypt	768	2000
pizd	1000	2000
proslikefan	1000	2000
pushdo	2700	2000
pykspa	1000	2000
qadars	1000	2000
qakbot	1000	2000
qsnatch (v1+v2)	2760	2000
ramdo	1000	2000
ranbyus	1000	2000
reconyc	1000	2000
shiotob	1000	2000
simda	1000	2000
suppobox	900	2000
symmi	1000	2000
tempedreve	1809	2000
tinba	2211	2000

Figuur 24: Overzicht met aantal gegenereerde DGA-domeinnamen met 30 DGA-types

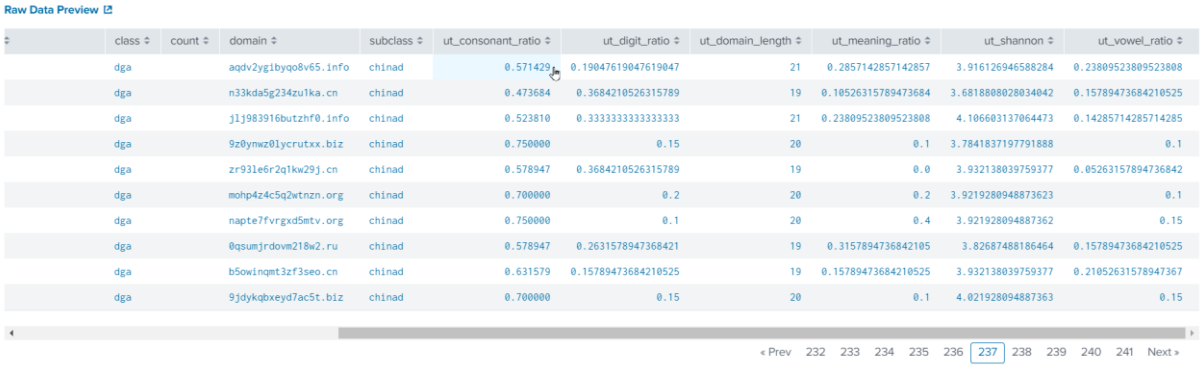
4.1.2 Splunk Machine Learning ToolKit

De machine learning algoritmes zelf worden toegepast met behulp van de Splunk Machine Learning ToolKit (MLTK). Voor de MLTK moet een experiment in een bepaalde categorie worden opgezet. Voor dit project is dit het voorspellen van een bepaalde klasse namelijk *legit* wat staat voor legitiem of DGA. De instellingen van het experiment zijn terug te vinden op onderstaande figuur.



Figuur 25: Instellingen MLTK-experiment

Het machine learning algoritme wordt uitgevoerd op de resultaten van een bepaalde search, in dit geval | *inputlookup dga_1m_features* | *fields – partition_number*. Het uitvoeren van de search geeft onderstaande resultaten.



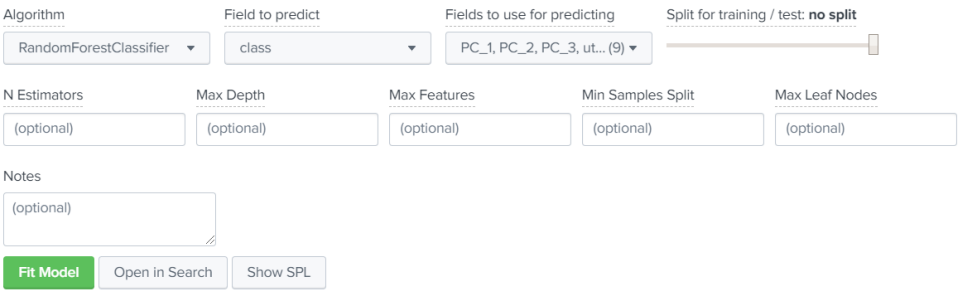
Figuur 26: Resultaten search MLTK

Voor elk domein zijn verschillende features berekend. Op basis van de resultaten van deze features zal het machine learning-model voorspellen of een ongeziene domeinnaam een legitieme of DGA-domeinnaam is. Het berekeningen van de features gebeuren onder andere via de URL Toolbox voor Splunk. De URL Toolbox-functies worden gebruikt binnen een search die de tabel met de features genereert. De invoer voor deze search in geval van training is het domeinnaam met de klasse en in het geval van een DGA-domeinnaam ook de subklasse. Onderstaand is ook de volledige search terug te vinden.

```
| inputlookup "mp_training2.csv"
| apply "dga_ngram"
| apply "dga_pca"
| fields - "domain_tfidf*"
| lookup ut_shannon_lookup word as domain
| lookup ut_meaning_lookup word as domain
| eval ut_digit_ratio=0.0, ut_vowel_ratio=0.0, ut_domain_length=max(1,len(domain))
| rex field=domain max_match=0 "(?<digits>\\d)"
| rex field=domain max_match=0 "(?<vowels>[aeiou])"
| eval ut_digit_ratio=if(isnull(digits),0.0,(mvcount(digits) / ut_domain_length)), ut_vowel_ratio=if
    (isnull(vowels),0.0,(mvcount(vowels) / ut_domain_length)), ut_consonant_ratio=max(0.0,((1.000000 -
    ut_digit_ratio) - ut_vowel_ratio))
| fields - digits, "-", vowels
| sample partitions=2
| outputlookup dga_mp_features
| stats count
| eval setup_action="events processed and written to lookup \"dga_mp_features\""
```

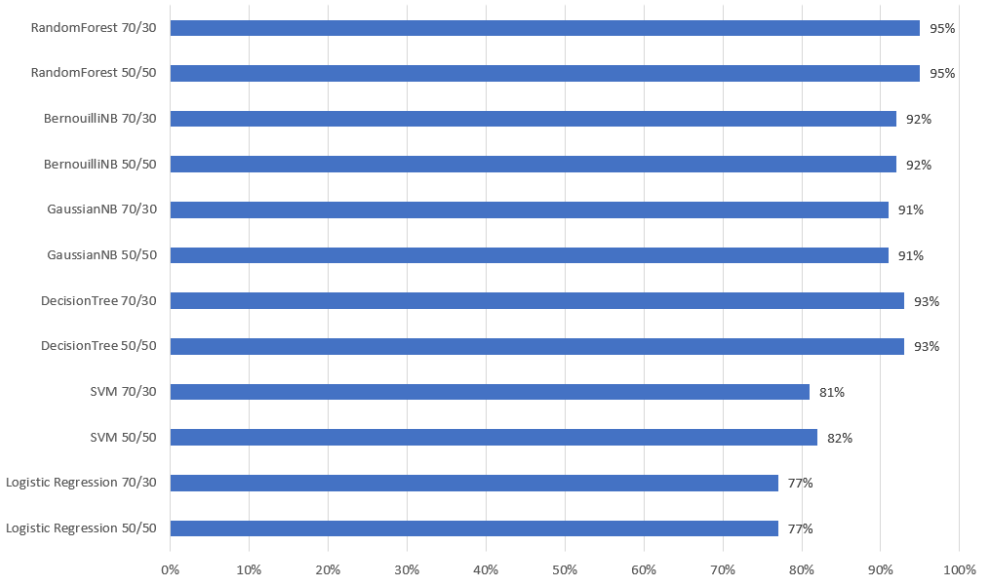
Figuur 27: Search voor genereren features trainingsdata

Wanneer de resultaten van de search met de features van de trainingsdata verwerkt is kunnen hierop de nodige instellingen gemaakt worden voor het trainen van het detectiemodel. Hieronder zijn de instellingen terug te vinden die van toepassing zijn op het model dat voor dit project getraind werd bij gebruik in het detectiesysteem.



Figuur 28: Instellingen training model detectiesysteem

Bij “Algorithm” kan het gewenste machine learning algoritme gekozen. Er zijn zes algoritmes beschikbaar waarbij na verschillende testen bleek dat Random Forest-classificatie de beste resultaten gaf. Het “*Field to predict*” is voor dit project de klasse van de domeinnaam en de “*Fields to use for prediction*” zijn hier de features die eerder per domeinnaam werden berekend. Trainingsdata wordt voor het testen van algoritmes steeds opgesplitst in een set waarop de training effectief gebeurt en een set waarop na training het model wordt getest om te bepalen hoe goed het model presteert. Voor de vergelijkende test werd voor elk algoritme een training uitgevoerd met een split op 50% trainingsdata en 50% testdata als ook op 70% trainingsdata en 30% testdata. Het wordt aangeraden steeds minstens 50% als trainingsdata te gebruiken. Wanneer de gewenste resultaten en de instellingen van het algoritme bepaald zijn, kan voor de uiteindelijke training de split op “*no split*” gezet worden om zoveel mogelijk data te gebruiken als trainingsdata en een zo goed mogelijk presterend model te ontwikkelen. Op onderstaande afbeelding is een vergelijking terug te vinden met de verschillende algoritmes en een split op 50/50 en 70/30.



Figuur 29: Vergelijking ML-algoritmes en verschillende splits

Elk algoritme heeft bepaalde prestatiescores zoals *precision*, *recall*, nauwkeurigheid en de F1-score. De verschillende prestatiescores met de bijbehorende *confusion matrices* zijn terug te vinden als bijlage aan deze masterproef. Voor het model met Random Forest classificatie dat gebruikt wordt als detectiesysteem in dit project zijn de resultaten in onderstaande tabel terug te vinden.

Tabel 1: Resultaat Random Forest-classificatie

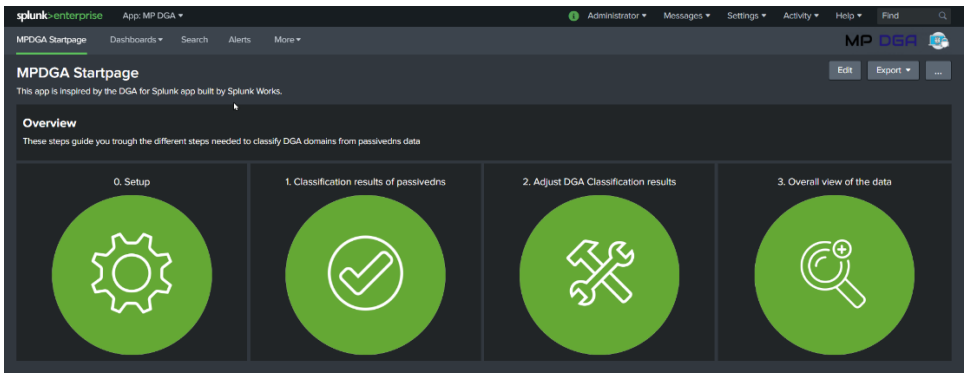
	Predicted DGA	Predicted legit
DGA	56575 (94,3%)	3425 (5,7%)
legit	3385 (5,6%)	56615 (94,4%)

Tabel 2: Prestatiescores Random Forest-classificatie

Precision	Recall	Accuracy	F1-score
95%	95%	95%	95%

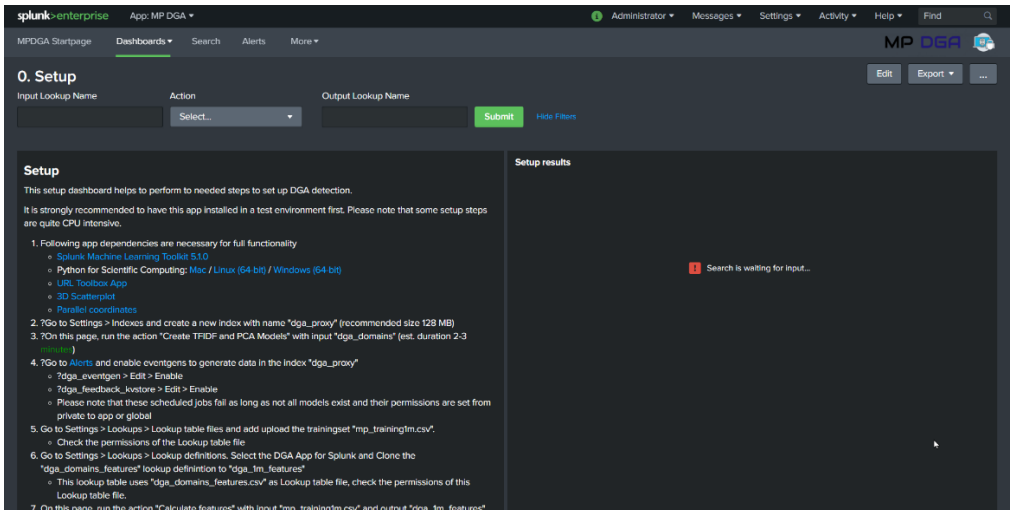
4.1.3 MP DGA

MP DGA oftewel MasterProef DGA is de app in Splunk die ontwikkeld is voor deze masterproef. De app bestaat uit verschillende dashboards of visualisaties voor het instellen, analyseren en overzichtelijk maken van de data.



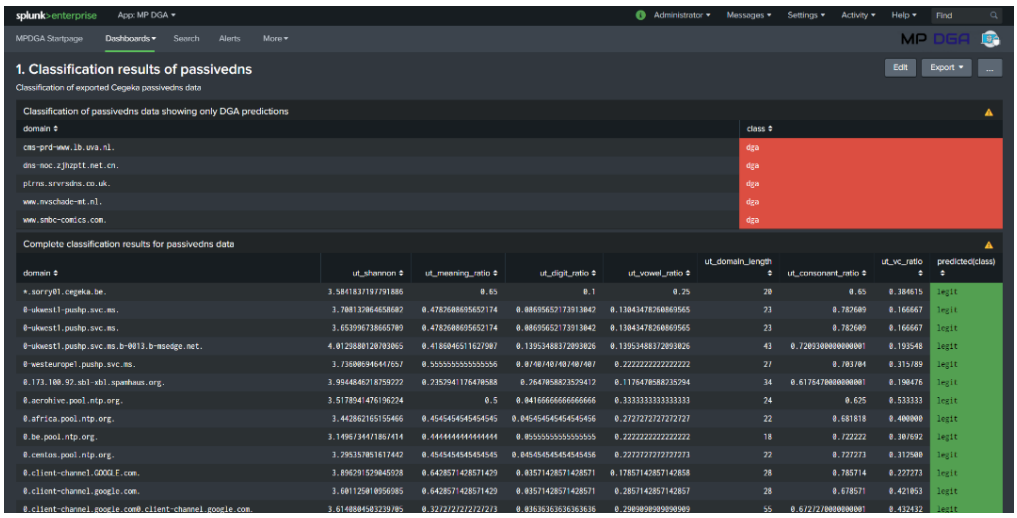
Figuur 30: Startscherm MP DGA-app

Bovenstaande figuur geeft het startscherm weer van de MP DGA-app in Splunk. Het geeft toegang tot de verschillende stappen in de app. De eerste stap is de *setup* stap en geeft toegang tot de verschillende stappen die nodig zijn voor het opzetten van de DGA-detectie. Ook het importeren van een csv-bestand van de passivedns index uit de Cegeka Splunk kan via hier worden ingeladen. Het *dashboard* van de setup is weergegeven in onderstaande afbeelding.



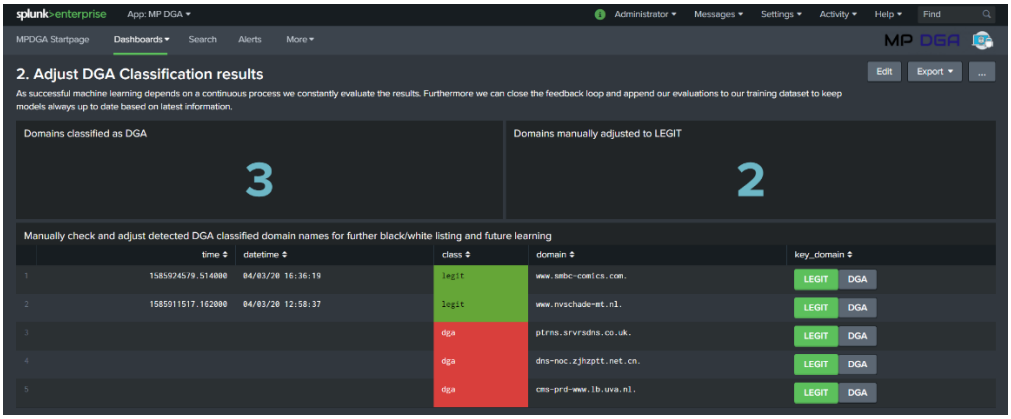
Figuur 31: MP DGA Setup dashboard

Nadat alle gegevens zijn ingeladen en de features hiervan zijn berekend kan de volgende stap uitgevoerd worden. Deze stap zal de domeinnamen afkomstig uit de passivedns analyseren en hiervoor een klasse voorspellen. De bovenste tabel geeft enkel de domeinnamen weer die voorspeld zijn als DGA-domeinnaam. De tweede tabel geeft de resultaten van alle domeinnamen die werden ingeladen in de app. Onderstaande afbeelding geeft een weergave van deze classificatie.



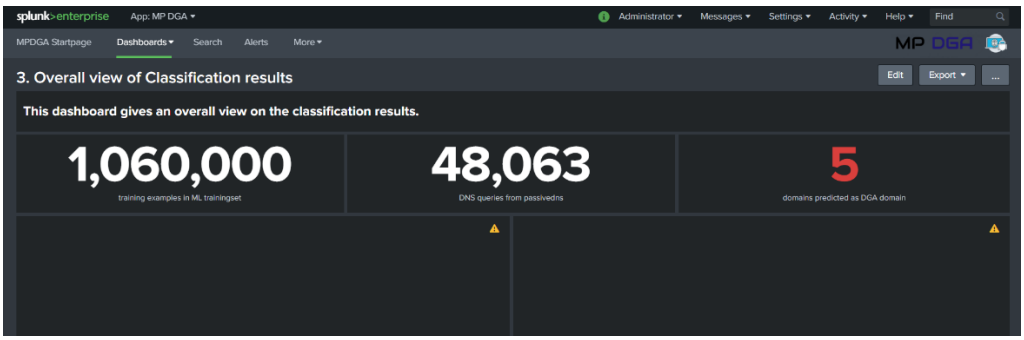
Figuur 32: MP DGA Classificatieresultaten

De voorspellingen door het detectiesysteem zijn niet 100% correct. Om dit te verbeteren kan de voorspelde klasse van een geïmporteerd domein worden aangepast om het model te verbeteren. Nadat een klasse van een domein manueel is aangepast tot de juiste klasse wordt dit opgenomen in de trainingset. Het model is ingesteld met een geplande training zodat dit één keer per week opnieuw een training doorloopt rekening houdend met de manueel aangepaste domeinen. De weergave van het dashboard voor het manueel aanpassen van geïmporteerde domeinnamen is op onderstaande afbeelding terug te vinden.



Figuur 33: MP DGA voor manueel aanpassen domeinnamen

Het laatste dashboard van de MP DGA-app in Splunk geeft een algemeen overzicht van een aantal waardes die in de app gebruikt worden. Dit dashboard kan verder uitgebreid worden met de gewenste visualisaties. Voor dit project werden volgende getallen gekozen: het aantal domeinen in de trainingset, het aantal geïmporteerde domeinen uit de passivedns en het aantal domeinen waarvoor het detectiesysteem voorspelde dat dit een DGA-domein zou zijn. Deze drie waardes en een weergave van dit dashboard zijn terug te vinden op onderstaande afbeelding.



Figuur 34: MP DGA dashboard algemeen overzicht

4.2 Genereren van DGA-domeinnamen

Het genereren van DGA-domeinnamen met behulp van machine learning werd opgezet met behulp van een GAN. In dit project werd een Wasserstein GAN gebruikt met de Cisco Umbrella populariteitslijst met de 1 miljoen populairste domeinnamen van de dag. De parameterinstellingen van het GAN zijn op onderstaande figuur terug te vinden.

```
BATCH_SIZE = 64 # Batch size
# How many iterations to train for, min value is 1000, Please increase the number of
iteration in 1000 units
ITERS = 2000
SEQ_LEN = 32 # Sequence length in characters
DIM = 512 # Model dimensionality. This is fairly slow and overfits, even on
          # Billion Word. Consider decreasing for smaller datasets.
CRITIC_ITERS = 10 # How many critic iterations per generator iteration. We
                  # use 10 for the results in the paper, but 5 should work fine
                  # as well.
LAMBDA = 10 # Gradient penalty lambda hyperparameter.
MAX_N_EXAMPLES = 100000 # Max number of data examples to load. If data loading
                        # is too slow or takes too much RAM, you can decrease
                        # this (at the expense of having less training data). default
                        value is 10000000
```

Figuur 35: Parameterinstellingen GAN

Als eerste werden de extensies van de invoerdomeinen ook meegenomen als invoer bij het GAN. Het aantal iteraties werd hierbij ingesteld op 1000. Dit leverde onderstaande resultaten op. De figuur geeft een deel van het verloop weer van de iteraties van het GAN.

After 10 iterations	After 50 iterations	After 100 iterations	After 150 iterations	After 250 iterations	After 500 iterations	After 1000 iterations
		tsaammammmm	sra.cocc.ooo	danandontd	faoraraha.cpm	sharonolaceas.com.com
	a	aaaaaammmmm	seesssssseesc.ccm	tandndnd.toi	rrrrfrana.com	findanshaliedeaiy.com
	tr	aamamammm	soomo.coc	nnandndndnndna.toto	sdofolrera.com	ehla.com
		aaaammmm	soooeeees.ccom	dotcaoti	inal.com.co	shygomacietliltc.com
	k.c	aaaaamammmmm	seeea.ccccom	dntot.ds	ssplfdra.ag	mal.co
		mammmmmammmmm	seeeeeeear.coco	nndndandoti	eracankco.n	ihaytaladds.com
		aaammmmmmm	eesees.ccccoo	dandi.cotdom	sollier.com	tescfard.co
		taaemaaamammmm	sesssooooooc.coom	nddntr.cimt	arasdae.com	sagalplanys.com
	a	aammammaaaaammmm	eaa.comooom	tandndes.com	rirarrorararaaga.com	farls.com
	a	samammmmamammmm	eeeea.cc	tondindrtaom	caneaena.com	cilato.yk
	a	aaaammmmmmmm	eesarcccoo	nnndeta.cii	iefarna.com	dlishavhmagis.com
		aaaaaaammm	eess.c.ccom	nndea.com	eraofarr.com	lapkepir.com
		aammmmmammmmmm	sasrc.ccom	nnndaanndndototi	dinangl	aheat.he
	a	amtmmamaaammmm	sassc.ccoco	ndsta.com	oeasrael.cagn	cpinde.ik
	r	aaamammmm	eeearc.com	ndnnndndrt.c.mi	hgdntsn	assyvrcferoglinyisiai.d
4		aaammmmmmammm	aac.ooomoooo	dnnnnnnndndrra.mi	elincailincy.p	cpiichorld.o.co
	mr	aaaammm	aasooooossscc.m	ndnnds.aocom	aoraeadosorrea.com	syplanto.com
	u	aaaaamammmm	tseeeeesrc.om	dindos.aot	cida.cn	calrsho0.com
		aaammmaaaammmmm	taosseesrc.co	tondnndnts.com	mlirrrrrrrfrarar.ca.gl.c	areesyterasphi0er
	r	aaammmmmm	siseeeeeeerrccccoooo	ndnnnnnd.totim	aersca.com	lilacilo.com
		aammiiaaaaammmmm	aocosess.ccoo	dnearcandnddndrt.com	horsnd.t	gifamdsipard.cy
	a	taaaammmm	eaeeeeessrcccco	ninddtaadodot	care.dnd	adle.com
4		aimmmmammmmm	seea.c.com	dnnnindnta.com	rena.can.hl	acolplecyllitc.com
		aaaammmmm	sreseeeec.ccom	tannndnea.ci.com	laras.comi.com	tifvplrne.istaardepi.com
w		amammmmmmm	seesscoc.cococom	donds.commdd.aom	lrtrdrt.com	chosiandetirerdat.i
		amammmmmmmmm	sess.c.ccmo	nnded.comi	lanalraeasead.co	splamenser.in
	m	taammmmmmamm	aasosooossscccccoo	nnnnandntr.dom	meraofrrsdreoar.com	soriethedpo.cy
		aaammmmmm	asssscccom	nnannndnta.com	araoolorarc.com	davierlin.com
		aaammmmmmmmm	eeescc.coom	ndnndntdes.com	orasogerd.com	trefandplane.com
		aaamaaamammm	seeeeeeerrcc.cm	ndnnea.com	comsdanthrg	shrm.com
		taammmmmmmmm	tsssccc.com	dndnnndndrtam	raeadnaarangn	crlifanlatapre.com
		eaaeaaaammmmmm	aesosssses.c.co	dnaondi.coto	aracomes.co	copsfare0or.ne
		aeamamaaammm	eeesrs.c.om	nednnddntrrandoti.t	aerteraeafrrscogo	mosieaeisaplen.co

Figuur 36: Verloop iteraties van het GAN

Zoals te zien in de meest rechtse kolom kloppen de extensies niet altijd voor de gegenereerde domeinnamen. Hierdoor werd er voor de tweede test gekozen om de extensies bij de invoerdomeinen te verwijderen en na het genereren een willekeurige extensie aan de domeinnaam toe te voegen. Het script hiervoor is terug te vinden in de bijlagen.

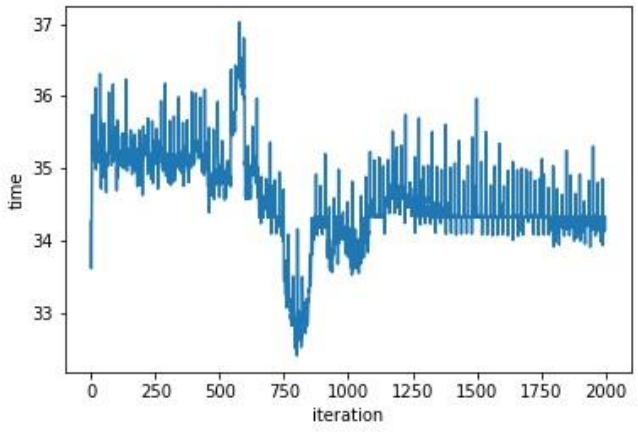
Als extensie werd er gekozen om generieke *Top Level Domains* (TLD) te gebruiken alsook *country code TLD* (ccTLD). Dit zijn extensies zoals onder andere .com en .net alsook de extensies gekoppeld aan landen zoals .be of .eu. Een volledige lijst hiervan is terug te vinden in de bijlagen.

Voor de tweede test werden 2000 iteraties uitgevoerd en achteraf een willekeurige extensie toegevoegd aan de domeinnaam. Dit gaf domeinnamen die er realistischer uitzagen en een aanvaller ook de mogelijkheid geven om deze domeinnamen te registreren en te gebruiken als connectie met de C&C-server. Onderstaande lijst geeft enkele domeinnamen weer die werden gegenereerd door het GAN.

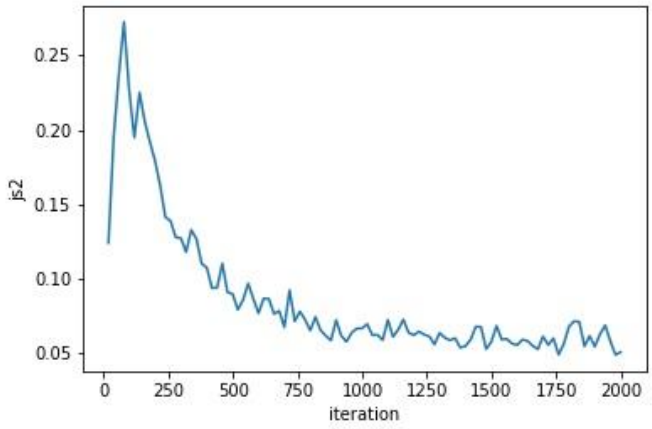
Voorbeelden GAN domeinnamen
curetgiso.be
mrastege.do
ne.ca
truvertic.lc
stin.su
toutinc.am
sybefenge.ee
sga.yt
fepcegpre.tc
bheatorv.uz
cpefsumurenkimr.us
tommemasserecg.cv
tpetshupaoc.vn
movoc.iq
sunheapprmteainerocmcsper.dk
cu.me
at2ip.su
racpineng.cm
pa.ni
bast.gw
fepcony.mk

Figuur 37: Voorbeelden GAN-domeinnamen met extensie

Het GAN verwerkt ongeveer 100 iteraties per uur. Voor het verwerken van de 2000 iteraties had het GAN ongeveer 20 uur nodig voor het genereren van de domeinnamen. De tijd per iteratie is te zien op onderstaande figuur.



Figuur 38: Tijd in seconden per iteratie voor het GAN



Figuur 39: Cost per iteratie voor het GAN

Bovenstaande figuur toont de cost per iteratie van het GAN. Dit kan in dit geval gezien worden als de afwijking tot een echte domeinnaam. Zoals te zien op de figuur is er een dalende trend voor de cost en is een minimum van 1000 iteraties aan te raden. Vanaf hier daalt de cost minder snel.

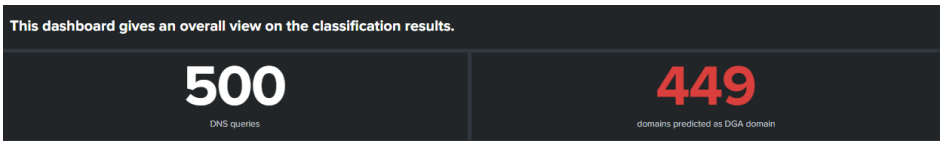
4.3 Detecteren van door GAN gegenereerde domeinnamen

Vervolgens werden de DGA-domeinnamen gegenereerd door het GAN ingeladen in de Splunk app die werd gebouwd voor het detecteren van DGA-domeinnamen. Er werd telkens getest met 500 gegenereerde DGA-domeinnamen door het GAN. Onderstaande figuur laat zien dat het detectiesysteem niet in staat was de domeinnamen van het GAN te detecteren. Uit ander reeds gevoerd onderzoek [45] blijkt dat dit type domeinnamen amper kan gedetecteerd worden. Dit onderzoek behaalt een detectie van ongeveer 1%.

Figuur 40: Resultaat GAN-domeinnamen zonder training

Na de eerste test voor detectie van de GAN-domeinnamen werd het detectiesysteem ook getraind op dit type DGA. Er werden net als voor de eerdere types 2000 DGA-domeinen toegevoegd aan de trainingset. Na het trainen van het model detecteerde het detectiesysteem 103 van de 500 DGA-domeinnamen. Om de detectieresultaten verder te verbeteren werden er vijf keer meer trainingsvoorbeelden toegevoegd aan de trainingset voor dit type DGA. Hierna kon het detectiesysteem 398 DGA-domeinnamen van de 500 detecteren wat ongeveer overeenkomt met 80%.

Om de resultaten verder te verbeteren werden eigenschappen op basis van *Principle Component Analysis* (PCA) toegevoegd voor het verbeteren van de detectieresultaten. Bij de derde test detecteerde het systeem 449 van de 500 DGA-domeinen, dit komt overeen met ongeveer 90% zoals ook te zien op onderstaande figuur.



Figuur 41: Detectie GAN-domeinen na training met PCA

De resultaten tonen dat dit een succesvolle methode is voor het genereren van DGA-domeinnamen. Hier dienen echter enkele bemerkingen gemaakt te worden. Zo is het wel zeer rekenintensief om de DGA-domeinen te genereren met het GAN en moest er een willekeurige extensie worden toegevoegd om de domeinnamen realistischer te maken zodat een aanvaller ook de mogelijkheid heeft om ze effectief te registreren.

Wanneer de hacker zou beslissen de domeinnamen op voorhand te genereren en deze lijst in te bouwen in de malware vallen de berekeningen op het toestel van het slachtoffer weg maar dit maakt de malware ook vatbaarder voor beveiligingssystemen. Eens een beveiligingssysteem de lijst kan vinden kunnen alle domeinnamen in een keer worden geblokkeerd op het systeem.

5. Besluit

In deze masterproef werd zowel het detecteren als het genereren van DGA-domeinnamen behandeld.

De app in Splunk die fungeert als detectiesysteem behaalt vrij goede resultaten op de gegevens afkomstig uit de passivedns index van Cegeka. De app is momenteel voorzien om geëxporteerde gegevens uit de Cegeka Splunk in te laden in de app. Een toekomstige aanpassing kan zijn om deze gegevens in real-time in te lezen en te analyseren. Op passivedns data van een kwartier werden ongeveer vijf vals positieven gemeld. Verdere aanpassingen en training van het model kan ervoor zorgen dat de accuraatheid van het systeem verhoogd wordt. Om problemen te vermijden kan dit type detectiesysteem best als ondersteuning van mensen gebruikt worden. Een volledig automatisch system dat ook automatische zwarte lijsten zou doorvoeren zou meer domeinen blokkeren dan nodig en hierdoor te veel manuele aanpassingen vragen.

Voor het detecteren van DGA-domeinnamen valt op uit de resultaten dat er een opmerkelijk verschil is tussen gekende en onbekende DGA-types. Het systeem heeft het veel moeilijker om ongeziene DGA-types te detecteren. De DGA-domeinnamen gegenereerd door het GAN zijn veel moeilijker te detecteren dan DGA-types die geen gebruik maken van machine learning. Daardoor zijn er ook veel meer trainingsvoorbeelden nodig om hiervoor degelijke detectieresultaten te verkrijgen. Een nadeel van door GAN gegenereerde DGA-domeinnamen is dat het veel reken intensiever is wanneer dit systeem ingebouwd zit in malware en het slachtoffer vermoedelijk zal merken dat er een probleem is.

De kwaliteit van de door GAN gegenereerde DGA-domeinen kan ook verhoogd worden door het aantal iteraties te verhogen en de batch-grootte te vergroten alsook door het aantal invoerdomeinnamen te verhogen. Langs de detectiezijde kan het aantal gedetecteerde DGA-domeinnamen ook verhoogd worden door het toevoegen van eigenschappen zoals het IP-adres en NXDOMAIN eigenschappen voor het verbeteren van de detectie van DGA-domeinnamen.

Literatuurlijst

[1] Cegeka NV, „2020_Boilerplate," 2020. [Online]. Available: https://www.cegeka.com/hubfs/Belgium%20and%20Netherlands/06%20-%20News/2020_Boilerplate.pdf. [Geopend 01 06 2020].

[2] expertsystem, „What is Machine Learning? A definition," expertsystem, 7 maart 2017. [Online]. Available: https://expertsystem.com/machine-learning-definition/. [Geopend 25 februari 2020].

[3] B. L. Hongyu Liu, „Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey," State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China, 2019.

[4] A. Wilson, „A Brief Introduction to Supervised Learning," Towards Data Science, 29 09 2019. [Online]. Available: https://towardsdatascience.com/a-brief-introduction-to-supervised-learning-54a3e3932590. [Geopend 27 02 2020].

[5] P. Cour, „Machine Learning Classification vs Regression," 19 07 2019. [Online]. Available: https://dev.to/petercour/machine-learning-classification-vs-regression-1gn. [Geopend 27 02 2020].

[6] S. Asiri, „Machine Learning Classifiers," Towards Data Science, 11 06 2018. [Online]. Available: https://towardsdatascience.com/machine-learning-classifiers-a5cc4e1b0623. [Geopend 2020 03 01].

[7] S. Swaminathan, „Logistic Regression — Detailed Overview," Towards Data Science, 15 03 2018. [Online]. Available: https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc. [Geopend 10 05 2020].

[8] A. Ng, „Lecture12," Coursera, [Online]. Available: https://www.coursera.org/learn/machine-learning/supplement/pSe2X/lecture-slides. [Geopend 01 06 2020].

[9] A. Ng, „Decision Boundary," Coursera, [Online]. Available: https://www.coursera.org/learn/machine-learning/supplement/N8qsm/decision-boundary. [Geopend 28 05 2020].

[10] Stackoverflow, „Plot Decision Boundary for Scikit Logistic Regression with 7 Features," Stackoverflow, 24 05 2015. [Online]. Available: https://stackoverflow.com/questions/30427819/plot-decision-boundary-for-scikit-logistic-regression-with-7-features. [Geopend 30 05 2020].

[11] P. Gupta, „Decision Trees in Machine Learning," Towards Data Science, 17 05 2017. [Online]. Available: https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052. [Geopend 11 05 2020].

[12] Oracle Big Data, „Decision Trees in Machine Learning, Simplified," Medium, 04 04 2018. [Online]. Available: https://medium.com/oracledevs/decision-trees-in-machine-learning-simplified-abc916c8b22b. [Geopend 30 05 2020].

[13] T. Yiu, „Understanding Random Forest," Towards Data Science, 12 06 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2. [Geopend 10 05 2020].

[14] M. Schott, „K-Nearest Neighbors (KNN) Algorithm for Machine Learning," Capital One Touch, 22 04 2019. [Online]. Available: https://medium.com/capital-one-tech/k-nearest-neighbors-knn-algorithm-for-machine-learning-e883219c8f26. [Geopend 25 04 2020].

[15] O. Harrison, „Machine Learning Basics with the K-Nearest Neighbors Algorithm," Towards Data Science, 10 09 2018. [Online]. Available: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761. [Geopend 20 04 2020].

[16] A. Dave, „Regression in Machine Learning," Data Driven Investor, 04 12 2018. [Online]. Available: https://medium.com/datadriveninvestor/regression-in-machine-learning-296caae933ec. [Geopend 01 03 2020].

[17] B. R. A. S. Stefan Michiels, „Enkelvoudinge Lineaire Regressie," Universitair Centrum voor Statistiek, KU Leuven, 1999. [Online]. Available: https://lstat.kuleuven.be/java/newbeta/presentation/dutch.htm. [Geopend 01 06 2020].

[18] A. Ng, „Linear regression with one variable," [Online]. Available: https://www.coursera.org/learn/machine-learning/supplement/ExY6Z/lecture-slides. [Geopend 18 02 2020].

[19] H. Tran, „Linear Regression model sample illustration," Researchgate, 09 2019. [Online]. Available: https://www.researchgate.net/figure/Linear-Regression-model-sample-illustration_fig3_333457161. [Geopend 01 06 2020].

[20] W. Ngaw, „Linear Regression," 07 04 2019. [Online]. Available: https://wngaw.github.io/linear-regression/. [Geopend 01 06 2020].

[21] MathWorks, „Unsupervised Learning," MathWorks, [Online]. Available: https://www.mathworks.com/discovery/unsupervised-learning.html#:~:text=Unsupervised%20learning%20is%20a%20type,patterns%20or%20grouping%20in%20data.. [Geopend 01 06 2020].

[22] N. Andrew, „Unsupervised Learning," Coursera, [Online]. Available: https://www.coursera.org/learn/machine-learning/supplement/hFF7A/lecture-slides. [Geopend 05 04 2020].

[23] AlindGupta, „Elbow Method for optimal value of k in KMeans," GeeksForGeeks, [Online]. Available: https://www.geeksforgeeks.org/elbow-method-for-optimal-value-of-k-in-kmeans/. [Geopend 01 06 2020].

[24] G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido en M. Marchetti, „On the Effectiveness of Machine and Deep Learning for Cyber Security," *2018 10th International Conference on Cyber Conflict,* vol. 1, p. 20, 2018.

[25] Cisco, „What is Phishing?," Cisco, 2019. [Online]. Available: https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html. [Geopend 01 03 2020].

[26] M. Butavicius, K. Parsons, M. Pattison en A. McCormac, „Breaching the Human Firewall: Social engineering in Phishing and Spear-Phishing Emails," in *Australasian Conference on Information Systems* , Adelaide, 2015.

[27] V. Santhana Lakshmi en M. Vijaya, „Efficient prediction of phishing websites using supervised learning algorithms," *International Conference on Communication Technology and System Design 2011,* vol. 1, p. 8, 2011.

[28] Unit 42, „Threat Brief: Understanding Domain Generation Algorithms (DGA)," Palo Alto Networks, 04 02 2019. [Online]. Available: https://unit42.paloaltonetworks.com/threat-brief-understanding-domain-generation-algorithms-dga/. [Geopend 20 04 2020].

[29] J. Duursma, „Wat is een Generative Adversarial Network?," Studio Overmorgen, [Online]. Available: https://www.jarnoduursma.nl/wat-is-een-generative-adversarial-network/. [Geopend 28 04 2020].

[30] A. DUBEY, „GANs in Real World - Can Bad Actors Use GANs to Beat AI?," Bolster.ai, 11 04 2020. [Online]. Available: https://bolster.ai/blog/gans-in-real-world-can-bad-actors-use-gans-to-beat-ai/. [Geopend 01 06 2020].

[31] M. Arjovsky, S. Chintala en L. Bottou, „Wasserstein Generative Adversarial Networks," 06 08 2017. [Online]. Available: https://research.fb.com/publications/wasserstein-generative-adversarial-networks/. [Geopend 03 05 2020].

[32] Cloudfare, „What is DNS? | How DNS Works," Cloudfare, [Online]. Available: https://www.cloudflare.com/learning/dns/what-is-dns/. [Geopend 01 03 2020].

[33] Amazon Web Services, „What is DNS?," Amazon Inc., [Online]. Available: https://aws.amazon.com/route53/what-is-dns/. [Geopend 01 06 2020].

[34] T. Hofmans, „Dns-over-https: vloek of zegen?," Tweakers, 14 10 2019. [Online]. Available: https://tweakers.net/reviews/7406/all/dns-over-https-meningen-verdeeld-over-encryptie-dns-queries.html. [Geopend 20 03 2020].

[35] S. A. Chowdhury, „DOMAIN GENERATION ALGORITHM – DGA IN MALWARE," Hackers Terminal_, 30 08 2019. [Online]. Available: https://hackersterminal.com/domain-generation-algorithm-dga-in-malware/. [Geopend 01 03 2020].

[36] V. R, P. S. K, P. Prabaharan, A. S en E. Mohamed, „Improved DGA Domain Names Detection and Categorization Using Deep Learning Architectures with Classical Machine Learning Algorithms," in *Cybersecurity and Secure Information Systems*, Cham, Switzerland, Springer Nature Switzerland AG, 2019, pp. 161-192.

[37] Cisco, „Umbrella Popularity List," Cisco Inc, 2016. [Online]. Available: http://s3-us-west-1.amazonaws.com/umbrella-static/index.html. [Geopend 20 02 2020].

[38] A. Ng, „Logistic Regression Model," Coursera, [Online]. Available: https://www.coursera.org/learn/machine-learning/supplement/bgEt4/cost-function. [Geopend 15 03 2020].

[39] Splunk Inc., „About Splunk," Splunk Inc, [Online]. Available: https://www.splunk.com/en_us/about-splunk.html. [Geopend 02 05 2020].

[40] SMTWare, „Producten," SMTWare, [Online]. Available: https://www.smtware.com/product/splunk/. [Geopend 02 05 2020].

[41] W3Schools, „Python Introduction," W3Schools, [Online]. Available: https://www.w3schools.com/python/python_intro.asp. [Geopend 02 05 2020].

[42] T. Osterbuhr, „What is Anaconda and how does it relate to Python?," Venture Lessons, [Online]. Available: https://www.venturelessons.com/what-is-anaconda/. [Geopend 02 05 2020].

[43] S. Yegulalp, „What is TensorFlow? The machine learning library explained," InfoWorld, 18 06 2019. [Online]. Available: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html. [Geopend 03 05 2020].

[44] NVIDIA, „TRAIN MODELS FASTER," NVIDIA, [Online]. Available: https://developer.nvidia.com/cuda-zone. [Geopend 01 06 2020].

[45] J. Spooren, D. Preuveneers, L. Desmet, P. Janssen en W. Joosen, „On the Use of DGAs in Malware: An Everlasting Competition of Detection and Evasion," ACM Symposium on Applied Computing, Leuven, 2019.

[46] States News Service, „DNS OVER HTTPS: WHY WE'RE SAYING DOH COULD BE CATASTROPHIC," *States News Service*, vol. 1, p. 2, 2019.

Bijlagen

1. Vergelijking detectiealgoritmes

Logistic Regression 50/50

Precision	Recall	Accuracy	F1
92%	69%	69%	77%

predicted actual	predicted DGA		predicted legit	
dga	19841	66,3%	10074	33,7%
legit	156524	31,1%	343875	68,7%

Logistic Regression 70/30

Precision	Recall	Accuracy	F1
92%	69%	69%	77%

predicted actual	predicted DGA		predicted legit	
dga	11977	66,2%	6107	33,8%
legit	92646	30,9%	206936	69,1%

SVM 50/50

Precision	Recall	Accuracy	F1
93%	76%	76%	82%

predicted actual	predicted DGA		predicted legit	
dga	22670	75,4%	7377	24,6%
legit	118786	23,8%	381277	76,2%

SVM 70/30

Precision	Recall	Accuracy	F1
93%	74%	74%	81%

predicted actual	predicted DGA		predicted legit	
dga	13870	76,6%	4234	23,4%
legit	77886	26,0%	221921	74,0%

DecisionTree 50/50

Precision	Recall	Accuracy	F1
95%	92%	92%	93%

predicted actual	predicted DGA		predicted legit	
dga	21825	73,0%	8066	27,0%
legit	33536	6,7%	467171	93,3%

DecisionTree 70/30

Precision	Recall	Accuracy	F1
95%	92%	92%	93%

predicted actual	predicted DGA		predicted legit	
dga	12907	72,0%	5027	28,0%
legit	20563	6,9%	279362	93,1%

GaussianNB 50/50

Precision	Recall	Accuracy	F1
91%	90%	90%	91%

predicted actual	predicted DGA		predicted legit	
dga	7199	23,9%	22969	76,1%
legit	30186	6,0%	469447	94,0%

GaussianNB 70/30

Precision	Recall	Accuracy	F1
91%	90%	90%	91%

predicted actual	predicted DGA		predicted legit	
dga	4225	23,5%	13718	76,5%
legit	18065	6,0%	281703	94,0%

BernouilliNB 50/50

Precision	Recall	Accuracy	F1
89%	94%	94%	92%

predicted actual	predicted DGA		predicted legit	
dga	0	0,0%	29836	100,0%
legit	33	0,0%	499189	100,0%

BernouilliNB 70/30

Precision	Recall	Accuracy	F1
89%	94%	94%	92%

predicted actual	predicted DGA		predicted legit	
dga	0	0,0%	17966	100,0%
legit	0	0,0%	299207	100,0%

RandomForest 50/50

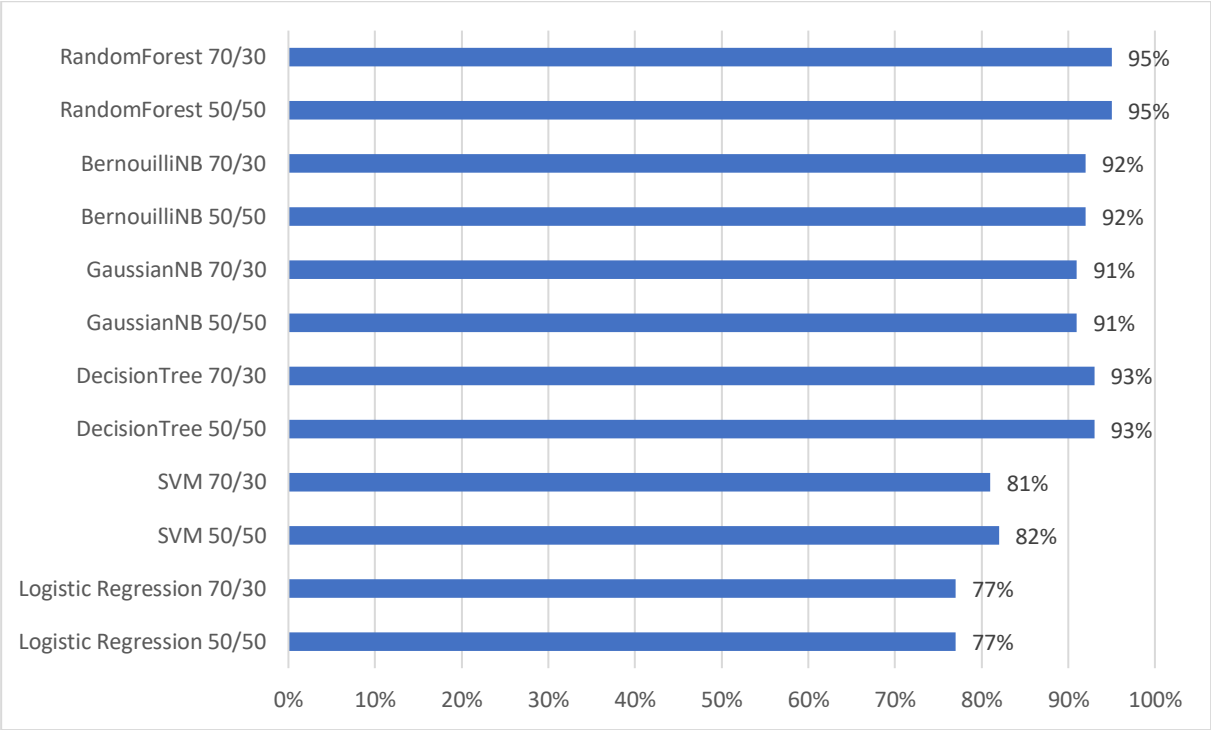
Precision	Recall	Accuracy	F1
95%	95%	95%	95%

predicted actual	predicted DGA		predicted legit	
dga	18401	61,8%	11395	38,2%
legit	14629	2,9%	484824	97,1%

RandomForest 70/30

Precision	Recall	Accuracy	F1
95%	95%	95%	95%

predicted actual	predicted DGA		predicted legit	
dga	11226	63,2%	6544	36,8%
legit	8886	3,0%	291053	97,0%


RandomForest 70/30
95%
RandomForest 50/50
95%
BernouilliNB 70/30
92%
BernouilliNB 50/50
92%
GaussianNB 70/30
91%
GaussianNB 50/50
91%
DecisionTree 70/30
93%
DecisionTree 50/50
93%
SVM 70/30
81%
SVM 50/50
82%
Logistic Regression 70/30
77%
Logistic Regression 50/50
77%
0%
10%
20%
30%
40%
50%
60%
70%
80%
90%
100%

2. Script willekeurige extensie

```
# -*- coding: utf-8 -*-
"""
Created on Wed May 20 11:50:29 2020

@author: denzelv
"""

import csv, random


rows = csv.reader(open("deception2_dga.csv", "r", encoding='utf-8'))
tlds = list(csv.reader(open("tld-list-ok.csv", "r", encoding='utf-8')))

outputfilename = "deception2_data_with_extension.csv"
with open(outputfilename, mode='w', newline='') as output_file:
    output_writer = csv.writer(output_file, delimiter=',', quotechar='"', quoting=csv.
    QUOTE_MINIMAL)
    for row in rows:
        print(row)
        output_with_extension = row[0].split(" ")[0] + str(*tlds[random.randint(0, 267)])
        output_writer.writerow([output_with_extension.ljust(32)])
        print(output_with_extension)
```

3. Extensies GAN domeinnamen

.aero	.cr	.kh	.pw
.biz	.cu	.ki	.py
.cat	.cv	.km	.qa
.com	.cx	.kn	.re
.coop	.cy	.kp	.ro
.edu	.cz	.kr	.rs
.gov	.de	.kw	.ru
.info	.dj	.ky	.rw
.int	.dk	.kz	.sa
.jobs	.dm	.la	.sb
.mil	.do	.lb	.sc
.mobi	.dz	.lc	.sd
.museum	.ec	.li	.se
.name	.ee	.lk	.sg
.net	.eg	.lr	.sh
.org	.eh	.ls	.si
.pro	.er	.lt	.sj
.travel	.es	.lu	.sk
.ac	.et	.lv	.sl
.ad	.eu	.ly	.sm
.ae	.fi	.ma	.sn
.af	.fj	.mc	.so
.ag	.fk	.md	.sr
.ai	.fm	.me	.st
.al	.fo	.mg	.su
.am	.fr	.mh	.sv
.an	.ga	.mk	.sy
.ao	.gb	.ml	.sz
.aq	.gd	.mm	.tc
.ar	.ge	.mn	.td
.as	.gf	.mo	.tf
.at	.gg	.mp	.tg
.au	.gh	.mq	.th
.aw	.gi	.mr	.tj
.ax	.gl	.ms	.tk
.az	.gm	.mt	.tl
.ba	.gn	.mu	.tm
.bb	.gp	.mv	.tn
.bd	.gq	.mw	.to
.be	.gr	.mx	.tp
.bf	.gs	.my	.tr
.bg	.gt	.mz	.tt
.bh	.gu	.na	.tv
.bi	.gw	.nc	.tw

.bj	.gy	.ne	.tz
.bm	.hk	.nf	.ua
.bn	.hm	.ng	.ug
.bo	.hn	.ni	.uk
.br	.hr	.nl	.um
.bs	.ht	.no	.us
.bt	.hu	.np	.uy
.bv	.id	.nr	.uz
.bw	.ie	.nu	.va
.by	.il	.nz	.vc
.bz	.im	.om	.ve
.ca	.in	.pa	.vg
.cc	.io	.pe	.vi
.cd	.iq	.pf	.vn
.cf	.ir	.pg	.vu
.cg	.is	.ph	.wf
.ch	.it	.pk	.ws
.ci	.je	.pl	.ye
.ck	.jm	.pm	.yt
.cl	.jo	.pn	.yu
.cm	.jp	.pr	.za
.cn	.ke	.ps	.zm
.co	.kg	.pt	.zw