FEDERATED LEARNING

LTZ2 A.C. (Anna) Vriend BSc, Wachtsofficier
Zr. Ms. Johan de Witt

De afgelopen jaren is het gebruik van data enorm toegenomen en deze data kan goed gebruikt worden voor *Machine Learning* (ML). Deze grote hoeveelheid data brengt veel mogelijkheden met zich mee, maar datasets zijn vaak niet toegankelijk of privacygevoelig. Hierdoor is de vraag naar een privacy vriendelijke manier van ML toegenomen. *Federated Learning* (FL) is een manier om de geschetste problemen op te lossen. Deze nieuwe ML-techniek zorgt ervoor dat de eigenaar van data controle houdt over zijn ruwe data, doordat het niet gedeeld hoeft te worden met externe partijen. In dit artikel beschrijf ik mijn onderzoek naar de toepassing van FL binnen Defensie om de samenwerking met externe partijen op het gebied van ML mogelijk te maken. →

MACHINE LEARNING

Om het begrip FL uit te leggen is eerst een begrip van Kunstmatige Intelligentie (KI) en ML belangrijk. Het doel van KI is om systemen te creëren die intelligent gedrag vertonen en een advies geven aan gebruikers om menselijke intelligentie te implementeren in machines. Intelligent gedrag houdt in dit geval in dat het systeem leert van ervaringen uit het verleden en zich aanpast aan nieuwe situaties. Binnen KI is automatisch leren, ofwel ML, een breed onderzoeksveld, omdat met ML Kunstmatige Intelligentie bereikt kan worden. De basis van ML is het gebruik van algoritmes om gegevens te analyseren, ervan te leren en daarna een advies of voorspelling te geven. De systemen worden dan in staat gesteld om data-gestuurde beslissingen te nemen in plaats van dat de systemen expliciet geprogrammeerd worden voor het uitvoeren van een bepaalde taak.

Afhankelijk van het doel van de ML wordt hiervoor een type algoritme gekozen. ML algoritmes zijn ontworpen om op basis van historische data na verloop van tijd te leren door training. De uitkomst van dit proces is een ML-model. Tijdens het trainingsproces wordt een voorspelling berekend aan de hand van het gekozen algoritme voor de historisch data. Deze voorspeling wordt vergeleken met test data om te kijken of de output van het model correct is. Het systeem corrigeert zichzelf door deze feedback mee te nemen.

De typen algoritmes die in het onderzoek zijn gebruikt zijn regressiealgoritmes en Neurale Netwerken (NN). Regressie is een techniek waarbij de relatie tussen één afhankelijke en meerdere onafhankelijke variabelen wordt onderzocht. De uitkomst van het algoritme is een continue numerieke waarde. Het voorspellen van huizenprijzen is bijvoorbeeld een regressietaak, omdat het aantal mogelijke uitkomsten continue is en niet in klassen is in te delen.

NN zijn een specifieke set algoritmes binnen ML. Ze onderscheiden zich van andere algoritmes, doordat ze gebruik maken van een architectuur die gebaseerd is op neuronen in het brein. In het menselijk brein ontvangt een neuron een input en gebaseerd op die input wordt er een output afgevuurd die wordt gebruikt door een andere neuron. Een NN bestaat uit op zijn minst drie lagen neuronen: een input laag, één of meerdere verborgen lagen en een output laag. De verborgen laag of lagen bestaan uit veel neuronen die verbonden zijn met elkaar. Met verzamelde data leert het netwerk de wegingen van de verbindingen tussen deze neuronen en zo kan het netwerk accurate voorspellingen doen.

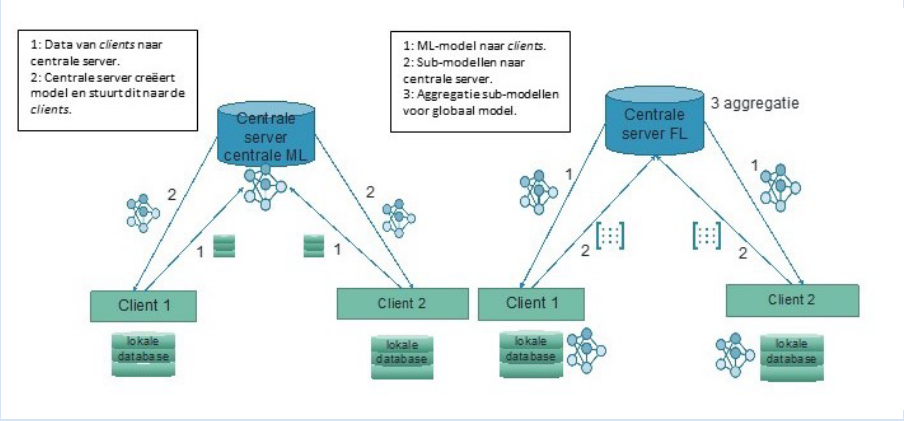
Bij het type *Recurrent Neural Networks* (RNN) wordt belangrijke informatie uit de input data van een neuron onthouden en hierdoor kunnen RNNs goed gebruikt worden om voorspellingen te doen. Deze flexibiliteit maakt een RNN geschikt voor toepassingen zoals sentiment analyse of *predictive maintenance* met bijvoorbeeld *Long Short Term Memory* (LSTM) netwerken. Bij LSTM-netwerken bestaan de verborgen lagen van het netwerk uit LSTM-lagen. Deze LSTM-lagen zorgen ervoor dat patronen in data kunnen worden herkend en onthouden en zo kan worden geleerd wat een normaal patroon in een dataset is en wat hiervan afwijkt. Zodra het model is getraind kan het model toekomstige waardes voorspellen.

FEDERATED LEARNING

Zoals eerder benoemd worden tegenwoordig veel succesvolle ML-modellen gebruikt in KI-applicaties. De twee belangrijke uitdagingen bij de implementatie hiervan zijn: dat de meeste industrieën data gescheiden van elkaar opslaan en dat er een toenemende vraag is naar KI waarbij privacy gewaarborgd wordt. Conventionele ML is gebaseerd op een gecentraliseerde data verzameling en kan deze uitdagingen niet aan. FL is fundamenteel anders, omdat het algoritme naar de data van de gebruikers wordt gebracht. In 2017 heeft Google de term FL bedacht en sindsdien heeft het een stijgend aantal toepassingen.

FL is niet afhankelijk van het verzamelen van data op een centrale plek waar training plaats vindt en maakt het mogelijk om modellen te trainen met gedecentraliseerde data. In figuur 1 is dit verschil schematisch weergegeven.

Bij FL worden sub-modellen getraind bij verschillende partijen, genaamd clients, met alleen lokale data. Een client kan een bedrijf zijn, maar ook een mobiel apparaat. De verschillende partijen delen vervolgens hun sub-modellen en daaruit komt een globaal model voort. Dit proces wordt gecoördineerd door een centrale server. In figuur 1 is het FL-proces te zien met twee voorbeeld clients. De ruwe data van clients wordt lokaal opgeslagen en niet gedeeld om privacy van gebruikers en vertrouwelijkheid van gegevens te garanderen. Het proces moet zorgvuldig worden ontworpen, zodat geen partij de vertrouwelijke data van andere partijen kan herleiden.



Voorbeeld met aan de linkerhand het proces volgens centrale ML en aan de rechterhand het FL-proces.]

ONDERZOEKSDOEL

FL kan ook gebruikt worden door Defensie voor bijvoorbeeld *predictive maintenance*. De opdrachtgever van het onderzoek, Data voor Onderhoud (DvO), is een team binnen Directie Materiële Instandhouding (DMI) van de Koninklijke Marine (KM). Het doel van DvO is om DMI te ondersteunen bij *Maintenance Decision Making* door (sensor)data van schepen te gebruiken om onderhoud te voorspellen. Voor die transitie is samenwerking met kennisinstellingen, de industrie en andere externe partijen noodzakelijk. Deze samenwerking kan mogelijk worden gemaakt door het gebruik van FL, zodat de KM geen ruwe (geheime) data hoeft te delen met externen.
DvO wilde erachter komen of FL gebruikt kan worden voor toekomstige projecten en hoe dit in de praktijk geïmplementeerd kan worden. Dit is een algemene vraag, dus heeft het onderzoek zich gericht op een specifiek voorbeeld van het delen van een ML-model van een externe partner van DvO, Koninklijke Van Oord. Koninklijke Van Oord is een van de grootste baggermaatschappijen ter wereld en heeft veel sensordata van schepen beschikbaar. Als meerdere partijen met vergelijkbare werktuigen samenwerken is een grotere hoeveelheid data beschikbaar (van voldoende kwaliteit) en dit levert nauwkeurigere voorspellingen op. Daarnaast kan FL gebruikt worden om de communicatie tussen schepen en de wal, of eenheden en Nederland, efficiënter te maken.

ONTWERP FL-ARCHITECTUUR

FL framework

De eerste stap bij het ontwerpen van een FL-systeem is het kiezen van een framework. Een framework is een interface voor de ontwikkelaar om het systeem makkelijker en sneller te ontwikkelen. Het bestaat uit bouwstenen om ML-modellen te kunnen trainen op de systemen, ook wel nodes, van externe clients. Elk framework heeft andere eigenschappen, waardoor ze geschikt zijn voor verschillende toepassingen.

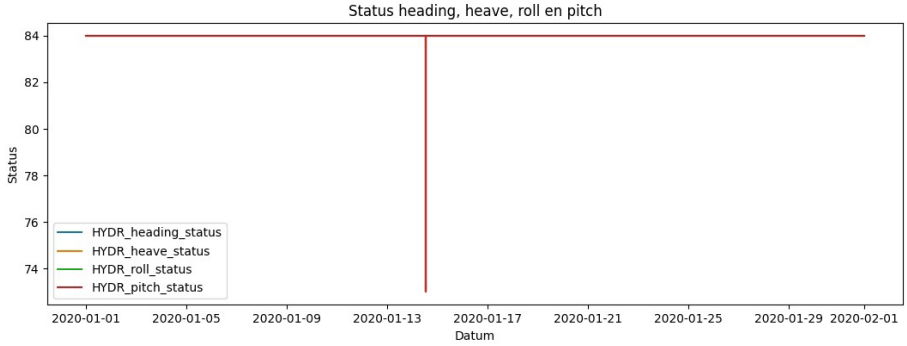
In dit onderzoek is na het vergelijken van verschillende frameworks gekozen om gebruik te maken van Flower. De eigenschappen van de frameworks die nader zijn onderzocht, zijn gekozen aan de hand van de eisen vanuit DvO. Daarnaast is gekeken naar hoeveel voorbeelden van implementaties van het framework beschikbaar waren. DvO vond het belangrijk om een framework te hebben dat met tijdreeksen kan werken, omdat scheepsdata bestaat uit tijdreeksen. Daarnaast was het bij de keuze voor een FL framework belangrijk dat het op het besturingssysteem Linux uitgevoerd zou kunnen worden en in *federated* modus zou kunnen werken. Dit houdt in dat het systeem gebruikt kan worden op meerdere nodes, bijvoorbeeld servers op verschillende plekken. Deze uitvoering vereist netwerkcommunicatie tussen de centrale server en clients.
Flower 0.17.0 bezit de juiste eigenschappen en is het meest flexibele FL framework. Het framework kan in combinatie met elk gangbaar ML framework gebruikt worden, werkt op de meeste soorten besturingssystemen en er is de meeste literatuur over beschikbaar.

Onderzoeksmethode

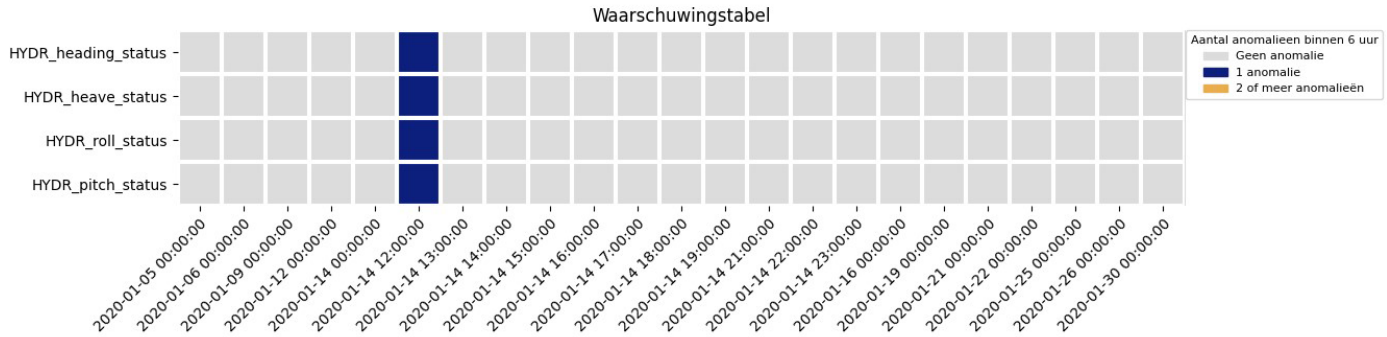
Na het literatuuronderzoek en de keuze voor het Flower framework, is code geschreven voor een Flower server en client. Om deze code te testen en te gebruiken, waren een dataset en ML-model benodigd. Van Oord heeft voor het onderzoek een dataset beschikbaar gesteld van *Flexible Fallpipe Vessel* (FFPV) *Stornes*. Deze dataset bestaat uit NetCDF (NC) bestanden die elk data over het schip bevatten voor een tijdsperiode van een dag.

De ML-modellen waarmee gewerkt is, zijn een Lineair Regressie (LR) model en een *Long Short Term Memory* (LSTM) netwerk. Het LR-model is gebruikt om de werking van de Flower server en client te testen. De inputs voor het LR model waren de breedte- en lengtegraden van de positie van het schip. De output van het model is een lijn door deze punten heen. Het doel van het LR-model trainen, was controleren of de gradiënten van de clients (in dit geval de hellingshoeken van de lijnen) overeenkwamen met de gradiënten die de server ontving en de communicatie tussen de server en clients dus goed verliep.

Nadat het LR-model succesvol getraind kon worden met FL is een LSTM-netwerk eerst met centrale ML getest. Er is gekozen om een bestaand LSTM-netwerk van DvO te gebruiken dat anomalieën in sensordata van schepen detecteert. Om dit model te trainen kunnen verschillende typen sensordata als input dienen en is er niet een specifieke dataset van sensoren benodigd. In het LSTM-netwerk wordt een regressiemodel gebruikt. Om bij regressieproblemen de afstand tussen voorspelde waardes van een model en echte waardes van een dataset te berekenen wordt vaak de *Root Mean Squared Error* (RMSE) gebruikt. Een lagere RMSE betekent dat het model beter waardes voorspeld.
Eerst is het LSTM-netwerk getest met een kleine dataset, waarbij op één tijdstip een duidelijke afwijking plaatsvond in vier signalen. Deze anomalie is te zien in figuur 2.



Grafiek met status parameters per dag van FFPV Stornes.



Waarschuwingstabel status parameters per dag van FFPV Stornes.

Het resultaat van de training met het LSTM-netwerk is de waarschuwingstabel die te zien is in figuur 3. In dit figuur is een duidelijke anomalie te zien voor alle systemen en dit komt overeen met het tijdstip van de afwijking in figuur 2.

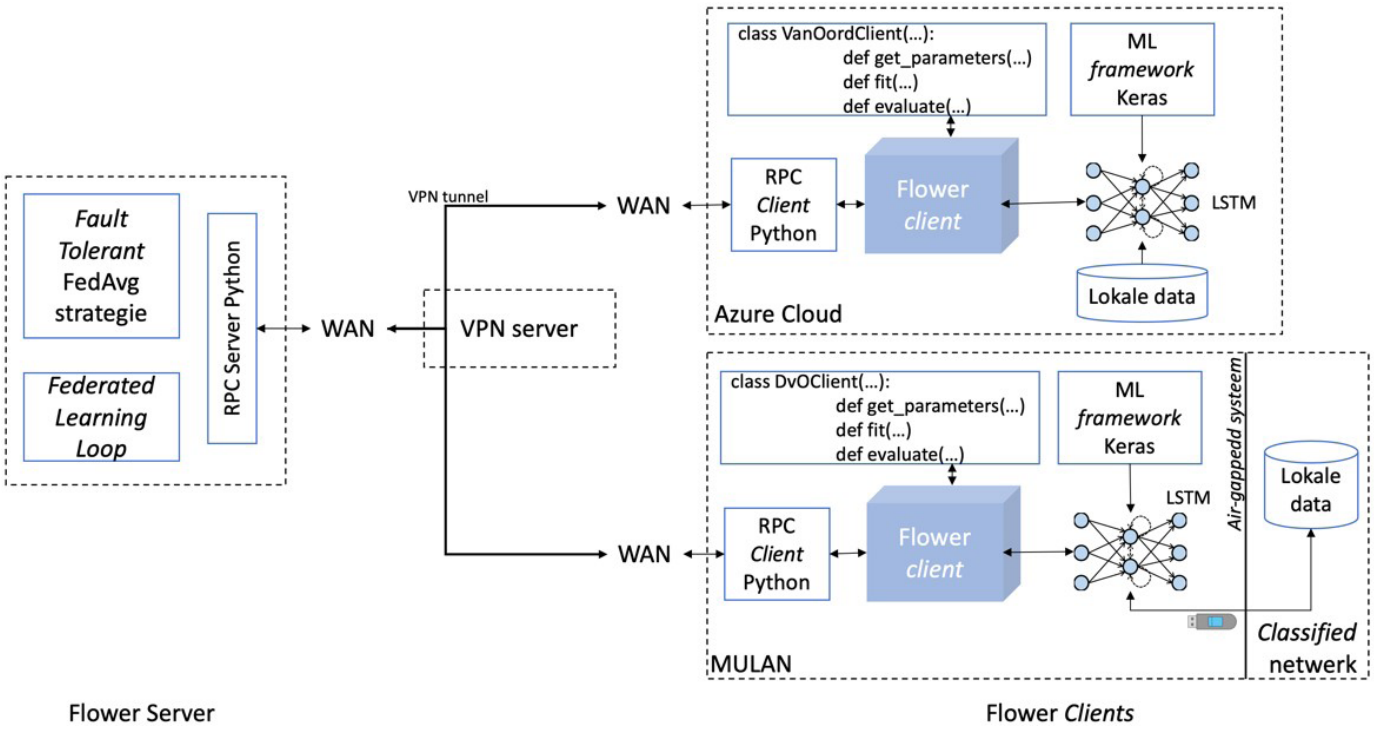
Na het verifiëren van de correcte werking van het FL-framework en het centraal getrainde ML-model is de code van het ML-model aan het FL-framework toegevoegd. Het resultaat hiervan is getest met behulp van meerdere laptops die dienden als server en clients. Tijdens het FL-proces zijn verschillende evaluatie parameters van het gebruikte ML-model verzameld en gebruikt als resultaten.
Om de laptops met elkaar te laten communiceren is een Wireguard *Virtual Private Network* (VPN) server opgezet waarmee de centrale server en clients konden verbinden. Vervolgens is de client code in combinatie met een eigen dataset op een aantal laptops gezet en is de server code op een andere laptop gezet. Hiermee is getest of de uitgewerkte FL-architectuur werkt en wat er eventueel nog nodig zou zijn om FL in de praktijk toe te kunnen passen.

RESULTATEN ONDERZOEK

Architectuur

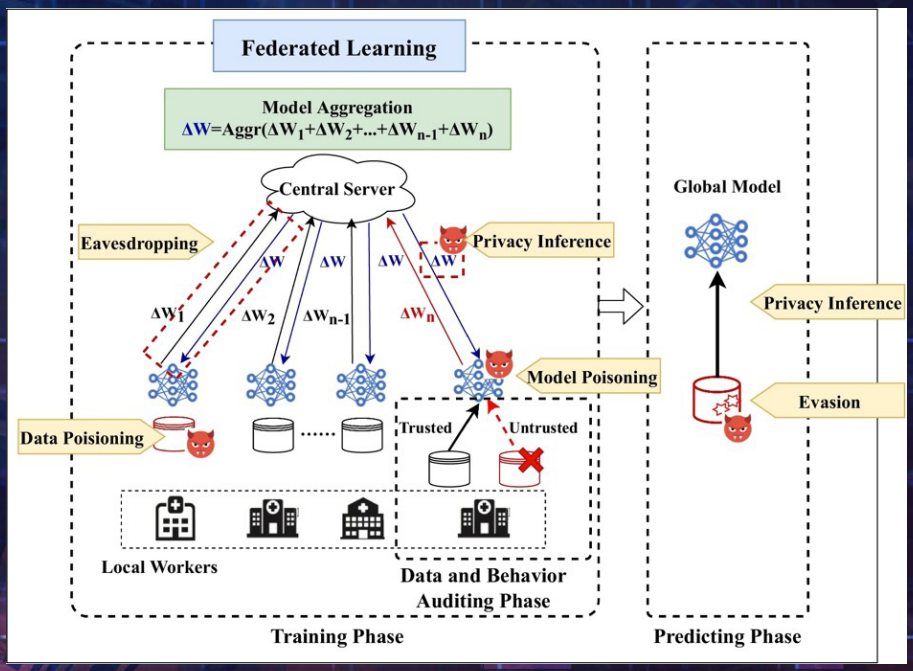
Het doel van het onderzoek was uitvinden of en hoe FL binnen Defensie, in het specifiek bij DvO, toegepast kan worden. De uitkomst van dit onderzoek is de FL-architectuur zoals te zien in figuur 4. De architectuur van het FL framework is zo veel mogelijk ontworpen om het gebruik in de toekomst door partijen, zoals DvO en Van Oord, na te bootsen.

Aan de Flower server kant is te zien, dat is gekozen voor een *Fault Tolerant* FedAvg strategie. De gekozen strategie bepaalt hoe clients geselecteerd worden, de training configuratie, het type parameter aggregatie en de evaluatie van het model. FedAvg is het standaard FL-algoritme. Het is een communicatie efficiënt algoritme waarbij de client eerst lokaal een aantal updates uitvoert en eenmaal per ronde de server het gemiddelde neemt van de model updates, ofwel aggregeert. *Fault Tolerant* FedAvg is een variant die om kan gaan met clients die de verbinding verbreken of achterblijven.



Architectuur Federated Learning met Flowers en twee clients.

Via het internet zijn de server en clients verbonden met een VPN server. De VPN zorgt voor een beveiligde verbinding, waardoor het voor aanvallers moeilijker wordt om het FL-proces af te luisteren. De architectuur van het framework in dit onderzoek komt overeen met de Van Oord client uit figuur 5. FL is in dit onderzoek namelijk niet getest op een defensienetwerk en de data is niet afkomstig vanuit een gerubriceerd netwerk. Ook is in het figuur te zien dat een LSTM-model wordt gebruikt. Het model detecteert anomalieën in de datasets van Van Oord. Om het te evalueren zijn *Mean Square Error* (MSE), RMSE en *accuracy* gebruikt. Om de validiteit van het model te testen, is gekeken of de evaluatie parameters van het model bij de server en clients overeenkwamen en of de gedetecteerde anomalieën overeenkwamen met anomalieën in de sensordata.



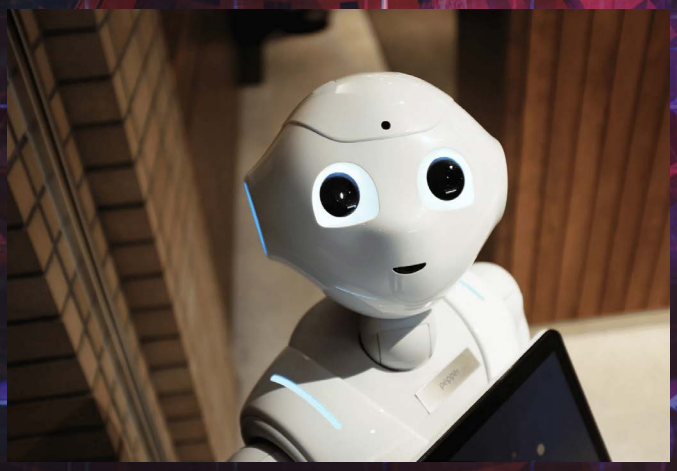
Mogelijkheden voor aanvallers in het FL-proces.

Beveiliging en privacy

FL wordt vaak toegepast in situaties waarbij de beveiliging van data en privacy van clients zeer belangrijk is. Alhoewel FL problemen van klassieke ML oplost, zijn er nog steeds verschillende mogelijkheden voor aanvallers. Binnen FL zijn *poisoning* en *inference* aanvallen bekende risico's. In figuur 5 staan voorbeelden van deze aanvallen weergegeven.

Poisoning kan plaatsvinden voordat een client de dataset en model gradiënten aanpast of doordat de centrale server het globale model aanpast. *Inference* kan plaatsvinden als buitenstaanders mee kunnen luisteren met het dataverkeer tussen de server en clients.

Model of data *poisoning* is mogelijk bij FL doordat de datasets van clients niet zichtbaar zijn en clients willekeurig kunnen worden geselecteerd. De kwaliteit van de datasets en historie van een client zijn dan niet te controleren. Als FL door de DvO en Van Oord gebruikt gaat worden, zal dit initieel zijn met een kleine groep grote partijen en een *Trusted Third Party* (TTP) als centrale server. Door de kleine groep partijen kan de server goed het gedragspatroon van clients evalueren en controleren op afwijkingen. Ook kan het risico op *poisoning* verminderd worden door de betrouwbaarheid van de clients regelmatig te controleren middels bijvoorbeeld authenticatie via het VPN.



Naast *poisoning* kan het FL-proces afgeluisterd worden door clients, de centrale server of buitenstaanders en kunnen vervolgens lokale of globale model gradiënten achterhaald worden. Met deze gradiënten kunnen aanvallers de dataset reconstrueren middels bijvoorbeeld een *Model Inversion Attack* (MIA). De mogelijkheid tot *inference* wordt verkleind door het gebruik van een VPN. Daarnaast zijn gradiënt inversie methodes in de praktijk vaak niet succesvol. Als een aanvaller toch een succesvolle MIA uitvoert kan *Differential Privacy* (DP) gebruikt worden. Door DP te implementeren in de code wordt aan de gradiënten van elke client willekeurige ruis toegevoegd voordat ze naar de server verzonden worden. Deze ruis maakt het moeilijk voor aanvallers om een dataset te herleiden aan de hand van lokale gradiënten. DP gaat wel ten koste van de prestatie van het model, dus moet een balans gevonden worden tussen prestatie en het risico op *inference* bij het realiseren van FL-systemen.

AFSLUITING

De ontwikkeling van FL biedt kansen voor de koninklijke marine in het bijzonder en defensie in het algemeen, om op een nieuwe en veiligere manier samen te werken met externe partijen. Voor het volledige onderzoek en de bronvermelding kunt u terecht bij: https://bibliotheeknlda.contentdm.oclc.org/digital/collection/p21075coll3/id/1234.