Federated learning (Aprendizaje Distribuido)

Muchas veces los datasets que se quieren procesar con Machine Learning son tan grandes que no pueden ser procesados con una sola máquina. En la computación distribuida clásica tenemos una plataforma central (Servidor) que almacena datos de manera distribuida en varios servidores esclavos. El problema que tenemos es que se debe realizar un envío de datos a un servidor central, estando el problema de que en caso de un ataque, un atacante puede quedarse escuchando para tomar los datos que se transportan al servidor. Otro problema es la latencia que hay de por medio, contando tanto el tiempo de transporte como el de procesado por parte del servidor.

Técnicas de protección de modelos de datos en Machine Learning:

- **Anonimización** (De las peores para machine learning al enmascarar los datos): Eliminación de datos, K-Anonimidad, etc… Vulnerable a ataques de enlazado, también puede generar datasets inútiles
- **Privacidad diferencial**: Consiste en añadir ruido a los datos, puede generar datasets inútiles.
- **Computación seguras entre múltiples partes**: Es la mejor manera de proteger machine learning, pero es excesivamente lenta. Hace que el calculo no dependa de nadie, pensado para entornos P2P. Sirve para solucionar problemas donde hay una función objetivo que se puede computar de forma colaborativa entre diferentes módulos sin necesidad de revelar información.

Que es Federated Learning

La idea de Federated Learning es un proceso colaborativo y distribuido para entrenar modelos de Machine Learning sin revelar información de entrenamiento. La idea es que las computaciones se realizan en cada dispositivo que recopila información. Cada dispositivo debe tener bastantes recursos computacionales. Los datos nunca se deben transferir al servidor o a ningún medio de almacenamiento centralizado. El aprendizaje Federado es similar a la computación distribuida con la diferencia de que los datasets son independientes e idénticamente distribuidos.

El algoritmo está descentralizado pero hay un nodo coordinador. El conjunto esta dividido en diferentes trozos que operan de forma independiente. Cada trozo tiene un número de muestras diferente. Los datos no se solapan, una muestra de un conjunto K nunca aparece en un conjunto E. En vez de calcular una función objetivo a nivel global, se calcula en trozos. El coordinados se encarga de juntar todos los resultados para obtener los datos. Existen varias soluciones:

- FSVRG (Federated Stochastic Variable Reducec Gradient)
- Federated SDG and Federate Averaging Algorithm (Usado por el teclado de google para predicción.)

Ejemplo: Funcionamiento del teclado de Google

El servidor central crea un modelo con unos parámetros y se lo envía a todos los clientes, estos empiezan a utilizarlo en local. Al principio se usa un modelo base que pueda servir para todos los clientes. La idea es que según se va usando se va refinando el modelo con los inputs del usuario. Los resultados finales se envían al servidor central, que crea una relación, crea un nuevo modelo y pasa a la siguiente iteración.

Problemas del Federated Learning

No hay comunicación con el servidor central, lo que significa que no se sabe si se usan bien o mal los datos. Los modelos contienen información sensible, por lo que aunque se entrenen por separado, los datos siguen ahí y pueden ser recuperados. Si hay un atacante malicioso entre el servidor y los clientes o el servidor está corrupto, un atacante puede manipular los datos como quiera. Un atacante Man In The Middle también puede hacerse con la información al estar en el medio de las comunicaciones. Existen ataques como Model Inversión que pueden sacar datos. Existen 2 tipos de ataques:

- Black Box (Pasivo): Su objetivo es el modelo ya entrenado final.
- White Box (Activa): Monitoriza cambios en el modelo en cada ronda de entrenamiento.

También hay 2 tipos de atacantes:

- Cliente
- Servidor

Poisoning

Otro tipo de ataque que se puede realizar es el de poisoning, enviando un modelo corrupto al servidor para que o el modelo no funcione (Ataque Random) o un ataque de reemplazo, que cambia el modelo original por uno distinto y que conviva con el anterior, la idea es realizar una predicción colateral sin que el servidor se entere. Estos ataques suelen ser de tipo cliente.

Prevención de ataques

Frente ataques de inteferencia:

- Se puede aplicar Segure Agregation Through Secure Multiparty Computation (SMC) o Cifrado homomórfico.

Frente ataque de poisoning:

- Se exploran los datos de los clientes (No tiene sentido ya que la idea de este sistema es que no se pueda hacer eso)

Nodos individuales:

- SMC
- Cifrado homomorfico
- Mecanismos de privacidad diferencial