1 Objetivos

- Investigar ataques de evasión, inferencia, extracción y envenenamiento
- Utilizar el framework Adversarial Robustness ToolBox para atacar y defender modelos de ML y DL

2 Preámbulo

Seguridad en modelos de data science

Los modelos de Machine Learning y Deep Learning son activos que están sujetos a los ciberataques, como cualquier otro activo digital.

Los ataques varían según su propósito y clasificación. En los ataques de caja negra, el adversario no conoce los detalles de implementación del modelo, en tanto que, en los ataques de caja blanca, el adversario si conoce los detalles.

Ataques de extracción

Las empresas que utilizan ML/DL para apoyar sus procesos de negocio invierten una gran cantidad de recursos en la investigación, desarrollo e implementación de sus modelos, y luego ofrecen un servicio pagado de clasificación a través de una API, por ejemplo.

Con esta información, se puede realizar un ataque de caja negra/blanca que consiste en utilizar un dataset y obtener la clasificación y confidencia a través de la API. Aun si utilizar la API tiene un costo, este será mínimo en comparación con los resultados. La idea es obtener las etiquetas y confidencia para cada una de las observaciones, y con ello, ¡entrenar un modelo propio! (Ataque de extracción).

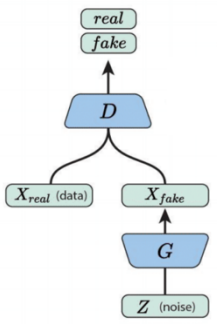
Dado que el nuevo modelo se entrenará en la forma en que el modelo objetivo clasifica, este tendrá resultados muy similares, sin invertir la gran cantidad de recursos que el modelo original.

Ataques de inferencia

Muchos modelos son entrenados con una combinación de datasets públicos y privados. Un atacante puede crear un modelo que permita saber si un registro fue utilizado como parte del entrenamiento (Membership). Un atacante también puede inferir data de un modelo a partir de indicar la clase buscada.

Ataques de evasión

Generative Adversarial Networks (GANs) son una forma de construir un modelo generativo al tener dos redes neuronales compitiendo una contra otra.



Una red toma el papel del generador (**G**), que convierte ruido aleatorio en imitaciones de data, intentando engañar al discriminador.

La otra red toma el papel del discriminador (**D**), que trata de distinguir data real de data falsa creada por el generador. Esto se puede aprovechar para realizar ataques con data que engañen a los modelos de clasificación.

Ataque de envenenamiento

Se aprovecha de la debilidad del entrenamiento federado, pues los nodos locales no siempre toman medidas de seguridad para asegurar la confiabilidad de sus fuentes de datos. La versión más peligrosa de este ataque es un backdoor, pues confunde al modelo únicamente para un patrón específico, y es muy difícil de detectar.

Defensa

La defensa de modelos de ML usa los mismos conceptos utilizados para los ataques. Por ejemplo, los ataques de envenenamiento usan perturbaciones sobre ciertas observaciones. Una técnica de defensa consiste en añadir perturbaciones a un dataset, y comparar los resultados. Si la predicción de ciertas observaciones no cambia a pesar de las perturbaciones añadidas, es probable que dichas observaciones fueron envenenadas con un patrón específico. En el caso de ataques adversariales, una técnica consiste en entrenar a un modelo con observaciones falsas para que aprenda a detectarlas.

3 Desarrollo

El laboratorio consiste en dos partes. En la primera parte, se implementarán dos ataques contra el modelo desarrollado en el laboratorio 8. En la segunda parte, se implementará la defensa contra los ataques anteriores.

Primera parte

Implemente dos ataques (de diferente categoría), utilizando el framework Adversarial Robustness ToolBox, originalmente desarrollador por IBM, y donado recientemente a The Linux Foundation.

https://adversarial-robustness-toolbox.org/

Este framework contiene módulos de ataque y defensa, métricas, etc; y soporta framewroks como TensorFlow, Keras, Scikit-Learn, PyTorch, etc., todo tipo de data (imágenes, tablas, video, etc.) y tareas de machine learning (clasificación, generación, etc.)

El modelo víctima del ataque será el modelo desarrollado en el laboratorio 8.

Sugerencia: instalar el ART framework y probar los ejemplos vistos en clase, antes de realizar los ataques sobre el modelo víctima, para asegurar que la herramienta fue instalada correctamente y que funciona sin problemas.

Segunda parte

Implemente la defensa con los ataques propuestos en la primera parte. Realice una comparación pre y post defensa, para analizar la efectividad de las defensas implementadas.

4 Calificación

- Se debe entregar el link al repositorio en Github del laboratorio que debe incluir:
    - Jupyter Notebook: explicación de los ataques elegidos, evidencia de los pasos realizados y prueba del ataque contra el modelo.
    - Jupyter Notebook: explicación de las técnicas de defensa elegidas, evidencia de los pasos realizados, y **evidencia de la efectividad de la defensa**.
- La fecha de entrega será el lunes **12 de mayo a las 23:59 horas.**
- Plagio parcial o total anula el proyecto, y se elevará el caso a la Dirección para las sanciones administrativas.
- Rúbrica
    - Ataques: 50% (explicación de los ataques, 15%, implementación 35%)
    - Defensas: 50% (explicación de las defensas, 15%, implementación 35%)