Google

Directrices para un aprendizaje automático responsable

Directrices

Sé cauteloso al modificar los flujos de control basados en los resultados de modelos de aprendizaje automático.

Los flujos de trabajo que utilizan los resultados de un algoritmo primitivo de aprendizaje automático pueden modificarse para crear superficies de ataque que son vulnerables. Según el consenso actual, no es aconsejable incorporar datos no previstos explícitamente a un flujo de trabajo de ejecución.

Impide el acceso directo a los modelos y a los metadatos de los modelos.

La interacción directa con los algoritmos primitivos de aprendizaje automático puede generar múltiples superficies de ataque, como exfiltraciones o inversiones, entre otras. Mantén estos objetos alejados de los usuarios. Permite solo interacciones controladas y autorizadas, y asegúrate de que el resultado va asociado con una función establecida, como una base de datos. No proporciones el resultado directamente al usuario.

Asegúrate de que los modelos utilizan sistemas de comprobación de la integridad, tanto durante el entrenamiento como después de la implementación.

El software puede sufrir modificaciones en distintas etapas del proceso de desarrollo e implementación. Con el aprendizaje automático sucede lo mismo. Hay que hacer tantas comprobaciones como sea posible. Ejecuta solo aquello que sepas que está previsto: antes de aplicar, autentica.

La complejidad de las funciones puede generar complejidad en los controles.

Asegúrate de que las interacciones sean breves, concisas y directas. Los flujos de trabajo grandes y complejos representan riesgos de exposición mucho mayores si un ataque consigue su objetivo.

Empieza por la lógica extensional para el entrenamiento y pasa a la lógica intensional para las funciones e inferencias cuando sea necesario.

Utilizar un conjunto verificado y conocido para el entrenamiento es una buena manera de evitar el envenenamiento de datos y otros ataques. Sin embargo, para crear funciones con mayores capacidades de inferencia es necesario incorporar nuevos datos. En estos casos, cotejar detenidamente los resultados del entrenamiento respecto a los datos de salida previstos para datos de entrada seleccionados refuerza la resiliencia de los sistemas.

Asegúrate de que haya pruebas de integración para el uso previsto del algoritmo primitivo.

Como cualquier otro software, los flujos de trabajo de aprendizaje automático requieren que sus unidades se integren y comprueben. La naturaleza heurística del software de aprendizaje automático puede complicar el desarrollo de aserciones fiables. Por esta razón, es recomendable centrar los esfuerzos en los resultados que se quieren obtener del flujo de trabajo a largo plazo.

Busca ataques basados en la síntesis automática, en especial hacia controles que se centran en "lo que eres".

Pregúntate siempre "¿puede una máquina hacer esto ahora?" antes de dar por supuesta la autenticidad de una voz, un vídeo o un texto, en especial cuando se estén utilizando para obtener derechos de acceso.

Usa políticas mutuas para adelantarte al modelo económico y garantizar un modelo de seguridad sólido a un coste bajo.

Si el coste de un ataque es inferior al coste de su posterior resolución, es probable que el control no sea eficaz. Asegúrate de que los costes de seguridad se ajusten automáticamente en función del impacto de los ataques que han tenido éxito para que a los atacantes les resulten cada vez más caros.

Google

Evita poner en manos del usuario, por ejemplo en un dispositivo, modelos entrenados a partir de datos confidenciales.

Reflexiona sobre el caso de uso utilizado para la creación del algoritmo primitivo de aprendizaje automático y supón que, en teoría, los datos de entrenamiento que lo crearon pueden recuperarse. Aplica esto a tu plan de continuidad empresarial y a las evaluaciones de riesgos que se utilizan en el desarrollo de productos.

Limita la velocidad de solicitudes para evitar la extrapolación de los ataques y el consumo innecesario de recursos.

Recuerda que los controles tradicionales siguen siendo muy eficaces: usa diferentes capas de seguridad, como listas de control de accesos, límites de velocidad de solicitudes y procesos de autenticación al diseñar e implementar tecnologías de aprendizaje automático.