

Unidad 3: Técnicas de aprendizaje supervisado y ciberseguridad

Adarsh KUMAR

Universitat Politècnica de Catalunya
Department of Computer Science

Coordinador del proyecto:
Prof. Ilker Demirkol

Proyecto MERiT
3 de septiembre de 2025

Co-funded by the European Union

[Diapositiva 2: Índice]

Modelos de amenazas | Tipos de ataques y defensa | Aplicaciones | Amenazas de día cero | Amenazas avanzadas | Extracción de características | Generalización

Índice

1. Modelos de amenazas

2. Tipos de ataques y defensa

3. Aplicaciones

4. Amenazas de día cero

5. Amenazas avanzadas

6. Extracción de características

7. Generalización

[Diapositiva 3: Aprendizaje automático adversarial (AML) en ciberseguridad]

Aprendizaje automático adversarial (AML) en ciberseguridad

¿Qué es el aprendizaje automático adversarial?

- Estudia las técnicas con las que los atacantes crean entradas para engañar a los modelos de ML.
- Los ejemplos adversariales provocan predicciones incorrectas sin levantar sospechas.
- Es fundamental para comprender las vulnerabilidades del ML en los sistemas de seguridad.

[Diapositiva 4: Modelos de amenazas en el ML adversarial]

Modelos de amenazas en el ML adversarial

Los modelos de amenazas definen el conocimiento y las capacidades del atacante, y condicionan las estrategias de ataque y de defensa.

1. Modelo de amenazas de caja blanca (White-Box)

- El atacante conoce por completo el modelo: arquitectura, parámetros, datos de entrenamiento y extracción de características.
- Permite ataques basados en gradiente muy eficaces (p. ej., FGSM, PGD).
- Ejemplos:
    - Modificar binarios de malware para evadir los clasificadores.

- Alterar las características del tráfico de red para eludir la detección de anomalías.

[Diapositiva 5: Modelos de amenazas en el ML adversarial]

Modelos de amenazas en el ML adversarial

2. Modelo de amenazas de caja negra (Black-Box)

- El atacante solo consulta el modelo y observa sus salidas.
- Utiliza sondeos y modelos sustitutos para inferir las fronteras de decisión.
- Ejemplos:
  - Crear correos de spam por ensayo y error para evadir los filtros.
  - Generar paquetes de red que evitan las alertas mediante el análisis de la respuesta del modelo.

[Diapositiva 6: Por qué importan los modelos de amenazas]

Por qué importan los modelos de amenazas

- **Orientan el diseño de las defensas:** las defensas dependen del nivel de conocimiento del atacante.
- **Influyen en la evaluación de riesgos:** los ataques de caja blanca son más difíciles de defender.
- **Condicionan los métodos de entrenamiento:** el entrenamiento adversarial adapta los modelos para hacerlos más robustos.

[Diapositiva 7: Tipos de ataques adversariales en ML]

Tipos de ataques adversariales en ML

Los modelos de ML se enfrentan a diversos ataques que buscan comprometer su integridad, confidencialidad o disponibilidad.

Ataques de evasión

- Se producen durante el despliegue; las entradas manipuladas evaden la detección o provocan una clasificación errónea.
- Ejemplos: modificar el malware para evitar su detección, alterar las características de los paquetes.
- Defensas: entrenamiento adversarial, características robustas, detección de ejemplos adversariales.

[Diapositiva 8: Tipos de ataques adversariales en ML]

Tipos de ataques adversariales en ML

Ataques de envenenamiento

- Atacan los datos de entrenamiento inyectando muestras maliciosas.
- Ejemplos: registros falsos de tráfico benigno, desplazamiento de las fronteras de un filtro de spam.
- Defensas: saneamiento de datos, entrenamiento robusto, monitorización de los datos de entrenamiento.

Ataques de inversión de modelo

- Extraen datos sensibles de entrenamiento consultando el modelo.
- Ejemplos: recuperar datos biométricos, inferir atributos confidenciales.
- Defensas: limitar la información de salida, privacidad diferencial, limitación de la tasa de consultas.

[Diapositiva 9: Sin título]

Diseñar sistemas de aprendizaje automático resilientes exige proteger todas las fases (entrenamiento, despliegue y posdespliegue) manteniendo un equilibrio entre rendimiento y privacidad.

La defensa eficaz frente a los ataques adversariales requiere una combinación de estrategias. El entrenamiento adversarial sigue siendo una de las defensas más eficaces, mientras que el enmascaramiento de gradientes (gradient masking) y las técnicas de detección aportan capas adicionales de protección para equilibrar robustez, eficiencia y precisión.

[Diapositiva 10: Aplicaciones en ciberseguridad]

Aplicaciones en ciberseguridad

El endurecimiento de modelos (model hardening) engloba las técnicas y los procesos que mejoran la resiliencia de los modelos de ML frente a ataques adversariales y otras vulnerabilidades.

- **Entrenamiento adversarial:** incorporar ejemplos adversariales durante el entrenamiento para mejorar la robustez del modelo.
- **Técnicas de regularización:** evitan el sobreajuste para mejorar la generalización y reducir la susceptibilidad a los ataques.

Co-funded by the European Union

- **Ingeniería de características robusta:** seleccionar características menos propensas a ser manipuladas por los atacantes.
- **Monitorización y actualización:** evaluar de forma continua el rendimiento del modelo y reentrenarlo para adaptarse a las amenazas emergentes.

[Diapositiva 11: Sin título]

Beneficios:

- Reduce el riesgo de ataques de evasión y de envenenamiento.
- Aumenta la fiabilidad de las herramientas de seguridad basadas en ML.
- Mantiene una alta precisión incluso en condiciones adversariales.
- Integrar modelos de ML endurecidos en los IDS para detectar ataques sofisticados y evasivos.
- Implementar la detección temprana de intentos de manipulación adversarial.
- Adaptar los sistemas a las nuevas tácticas de ataque mediante aprendizaje continuo.

[Diapositiva 12: Sin título]

Los IDS monitorizan la actividad de la red o del sistema para detectar comportamientos maliciosos. Los IDS impulsados por ML aprovechan el reconocimiento de patrones y la detección de anomalías para mejorar la precisión y la adaptabilidad.

Contribuciones del ML a los IDS:

- **Detección basada en anomalías:** uso de modelos como Isolation Forest, Autoencoders o clustering para detectar desviaciones del comportamiento normal.
- **Mejora de la detección basada en firmas:** actualización automática de las firmas a partir de los patrones de ataque aprendidos.
- **Aprendizaje adaptativo:** aprendizaje continuo a partir de nuevos datos de ataques para detectar amenazas de día cero y en evolución.
- **Reducción de falsos positivos:** uso de modelos sofisticados para distinguir las anomalías benignas de los ataques reales.

[Diapositiva 13: Sin título]

Desafíos abordados:

- Detectar ataques novedosos u ofuscados.

Co-funded by the European Union

- Escalar a datos de red de gran volumen y alta velocidad.
- Gestionar la deriva de concepto (concept drift), cuando el comportamiento normal cambia con el tiempo.

[Diapositiva 14: Sin título]

El endurecimiento de los modelos de aprendizaje automático y los IDS robustos son aplicaciones vitales del ML en ciberseguridad. Al reforzar los modelos y mejorar las capacidades de detección, las organizaciones pueden defenderse mejor de las ciberamenazas sofisticadas y reducir los riesgos de seguridad.

[Diapositiva 15: Amenazas de día cero: comprender los exploits de día cero y el comportamiento de las APT]

Amenazas de día cero: comprender los exploits de día cero y el comportamiento de las APT

Las amenazas de día cero representan algunos de los riesgos más difíciles y peligrosos de la ciberseguridad. Implican ataques que explotan vulnerabilidades previamente desconocidas, lo que dificulta su detección y defensa.

Definición: un exploit de día cero es un ciberataque que se dirige a una vulnerabilidad de software desconocida para el fabricante o para la comunidad de seguridad en el momento del ataque.

[Diapositiva 16: Sin título]

Características clave:

- No hay parches ni soluciones disponibles cuando se explota.
- Puede causar daños importantes antes de ser detectado.
- A menudo se vende o se intercambia en mercados clandestinos.

Ejemplos:

- Explotar una nueva vulnerabilidad de desbordamiento de búfer en un software de uso generalizado.
- Atacar fallos no divulgados en sistemas operativos o dispositivos de red.

[Diapositiva 17: Amenazas persistentes avanzadas (APT) y uso de día cero]

Amenazas persistentes avanzadas (APT) y uso de día cero

Las APT son campañas de ciberataque sofisticadas y de largo plazo, normalmente orquestadas por

adversarios con grandes recursos y habilidades, como Estados-nación o grupos de cibercriminales organizados.

Cómo usan las APT los exploits de día cero

- **Acceso inicial:** usar vulnerabilidades de día cero para infiltrarse sigilosamente en las redes objetivo.
- **Movimiento lateral:** explotar vulnerabilidades para desplazarse por las redes sin ser detectado.
- **Persistencia:** establecer una presencia a largo plazo para extraer datos o interrumpir las operaciones.
- **Evasión:** utilizar exploits de día cero para eludir los mecanismos de detección tradicionales.

[Diapositiva 18: Amenazas persistentes avanzadas (APT) y uso de día cero]

Amenazas persistentes avanzadas (APT) y uso de día cero

Desafíos que plantean las amenazas de día cero

- **Dificultad de detección:** los sistemas basados en firmas no pueden detectar exploits desconocidos.
- **Tiempo de respuesta:** las organizaciones deben responder con rapidez sin parches oficiales.
- **Complejidad de la atribución:** es difícil identificar a los atacantes por sus tácticas sigilosas.
- **Alto impacto:** posibilidad de daños generalizados en infraestructuras críticas.

[Diapositiva 19: Estrategias de mitigación]

Estrategias de mitigación

- Emplear la detección de anomalías de comportamiento para identificar actividades inusuales.
- Usar fuentes de inteligencia de amenazas para estar al tanto de los nuevos días cero.
- Implementar una defensa en profundidad con controles de seguridad en capas.
- Actualizar y parchear con regularidad las vulnerabilidades conocidas para reducir la superficie de ataque.
- Invertir en planes de respuesta a incidentes para una contención rápida.

[Diapositiva 20: Extracción de características para la detección de malware]

Extracción de características para la detección de malware

Comprender los exploits de día cero y el comportamiento de las APT es fundamental para la ciberdefensa moderna. La detección proactiva, la respuesta rápida y la monitorización continua constituyen la base para mitigar estas amenazas avanzadas.

La detección eficaz de malware depende en gran medida de extraer características significativas de las muestras de software. La extracción de características puede dividirse en dos grandes categorías:

Características del análisis estático

- Opcodes
- Cadenas de texto (strings)

Características del análisis dinámico

- Comportamientos en el sandbox

[Diapositiva 21: Extracción de características para la detección de malware]

Extracción de características para la detección de malware

La detección eficaz de malware depende en gran medida de extraer características significativas de las muestras de software. La extracción de características puede dividirse en dos grandes categorías: estática y dinámica.

El análisis estático inspecciona el binario del malware sin ejecutarlo. Se centra en las propiedades inherentes que pueden indicar un comportamiento malicioso.

[Diapositiva 22: Extracción de características para la detección de malware]

Extracción de características para la detección de malware

Características estáticas comunes:

- **Opcodes (códigos de operación):** extraer secuencias de instrucciones de máquina para identificar patrones maliciosos. Analizar la frecuencia y el orden de los opcodes para detectar anomalías o firmas maliciosas conocidas.
- **Cadenas de texto (strings):** extraer texto incrustado, como URL, rutas de archivos, claves del registro o comandos sospechosos. La presencia de ciertas palabras clave (p. ej., cmd.exe, powershell) puede indicar una intención maliciosa.
- **Metadatos:** las cabeceras de archivo, las bibliotecas importadas y las características de las secciones aportan pistas contextuales.

[Diapositiva 23: Extracción de características para la detección de malware]

Extracción de características para la detección de malware

Ventajas:

- Rápido y seguro, ya que no requiere ejecutar código.
- Útil para el escaneo preliminar a gran escala.

Limitaciones:

- No puede detectar comportamientos que solo se activan en tiempo de ejecución.
- Se evade fácilmente mediante ofuscación o empaquetado (packing).

[Diapositiva 24: Extracción dinámica de características]

Extracción dinámica de características

El análisis dinámico consiste en ejecutar el malware en un entorno controlado (sandbox) para observar su comportamiento.

Características dinámicas comunes:

- **Llamadas al sistema y a la API:** monitorizar las interacciones con el sistema operativo, como el acceso a archivos, las conexiones de red o las modificaciones del registro.
- **Actividad de procesos y de red:** rastrear la creación de procesos hijo, el uso inusual de recursos o las conexiones salientes a IP sospechosas.
- **Patrones de comportamiento:** detectar patrones como mecanismos de persistencia, intentos de escalada de privilegios o comportamiento de exfiltración de datos.

[Diapositiva 25: Extracción dinámica de características]

Extracción dinámica de características

Ventajas:

- Captura el comportamiento real, incluidas las acciones que se activan de forma condicional.
- Es más difícil que el malware oculte su actividad maliciosa.

Limitaciones:

- Consume muchos recursos y es más lento que el análisis estático.
- Algunos tipos de malware detectan los entornos sandbox y modifican su comportamiento.

[Diapositiva 26: Generalización en el aprendizaje automático]

Generalización en el aprendizaje automático

Combinar la extracción de características estáticas y dinámicas ofrece una visión integral para la detección de malware. Mientras que el análisis estático aporta velocidad y escalabilidad, el análisis dinámico revela información conductual más profunda, lo que permite sistemas de detección más precisos y robustos.

Generalización en el aprendizaje automático

En las aplicaciones de ciberseguridad, en especial en la detección de malware y de intrusiones, los modelos de aprendizaje automático deben generalizar bien frente a amenazas no vistas, en lugar de limitarse a memorizar firmas conocidas. La generalización es la capacidad del modelo para funcionar con precisión con datos nuevos y no vistos, más allá del conjunto de entrenamiento.

[Diapositiva 27: Sobreajuste y generalización en el aprendizaje automático para ciberseguridad]

Sobreajuste y generalización en el aprendizaje automático para ciberseguridad

¿Qué es el sobreajuste (overfitting)? El sobreajuste se produce cuando un modelo aprende demasiado bien el ruido o los patrones específicos de los datos de entrenamiento, incluidas las firmas conocidas, en lugar de las características generales subyacentes. Estos modelos funcionan de maravilla con los datos de entrenamiento, pero mal con amenazas nuevas y no vistas.

¿Por qué es un problema el sobreajuste a las firmas conocidas?

- Las ciberamenazas evolucionan constantemente; con regularidad aparecen nuevas variantes de malware y de ataques.
- Los modelos sobreajustados no detectan los ataques de día cero ni los novedosos que no coinciden con las firmas conocidas.
- Reduce la eficacia práctica de las herramientas de seguridad basadas en ML.

[Diapositiva 28: Técnicas para mejorar la generalización]

Técnicas para mejorar la generalización

- **Regularización:** métodos como la regularización L1/L2 penalizan los modelos demasiado complejos para evitar que memoricen el ruido.
- **Validación cruzada:** el uso de la validación cruzada de k particiones (k-fold) garantiza que el modelo funcione de forma consistente en distintos subconjuntos de datos.
- **Selección e ingeniería de características:** centrarse en las características que capturan comportamientos fundamentales y no firmas superficiales.
- **Aumento de datos (data augmentation):** introducir variaciones en los datos de entrenamiento, incluidos ejemplos sintéticos o adversariales, para mejorar la robustez.

Co-funded by the European Union

- **Parada temprana (early stopping):** detener el entrenamiento cuando el rendimiento con los datos de validación deja de mejorar, evitando el sobreentrenamiento con los datos de entrenamiento.
- **Métodos de conjunto (ensemble):** combinar varios modelos reduce el riesgo de sobreajuste a patrones de datos concretos.

[Diapositiva 29: Implicaciones prácticas]

Implicaciones prácticas

- Los modelos entrenados con una buena generalización pueden detectar variantes de ataque no vistas previamente.
- Reducen los falsos negativos al evitar la dependencia de firmas de ataque estáticas o conocidas.
- La generalización permite sistemas de seguridad adaptativos que evolucionan con las amenazas emergentes.

[Diapositiva 30: El aprendizaje automático y los pipelines de detección de malware]

El aprendizaje automático y los pipelines de detección de malware

Garantizar una buena generalización es fundamental para los modelos de aprendizaje automático en ciberseguridad. Evitar el sobreajuste a las firmas conocidas permite a los modelos detectar con eficacia amenazas novedosas y en evolución, y mantener defensas sólidas a lo largo del tiempo.

El aprendizaje automático en los pipelines de detección de malware

El aprendizaje automático desempeña un papel crucial en la automatización y la mejora de la detección de malware. A lo largo de los pipelines de detección se aplican distintos algoritmos y arquitecturas de modelos, según los datos y la tarea.

[Diapositiva 31: 1. Random Forests]

1. Random Forests

Los Random Forests son métodos de aprendizaje por conjuntos basados en árboles de decisión, muy utilizados por su interpretabilidad y su sólido rendimiento con datos estructurados.

Aplicación:

- Se usan sobre todo con características estáticas, como frecuencias de opcodes, metadatos de archivos y cadenas de texto extraídas.
- Manejan bien los espacios de características de alta dimensionalidad.

- Son robustos frente al ruido y menos propensos al sobreajuste que los árboles de decisión individuales.

[Diapositiva 32: 2. Redes neuronales convolucionales (CNN)]

2. Redes neuronales convolucionales (CNN)

Beneficios:

- Entrenamiento e inferencia eficientes.
- Pueden ofrecer la importancia de las características para facilitar la interpretabilidad.

Las CNN destacan en la extracción de características y patrones espaciales; se usan tradicionalmente en el procesamiento de imágenes y se aplican cada vez más a la detección de malware.

Aplicación:

- Tratan los binarios de malware o las secuencias de opcodes como imágenes o cuadrículas.
- Extraen patrones jerárquicos a partir del bytecode sin procesar o de representaciones de características transformadas.
- Son útiles para detectar variantes de malware ofuscadas o polimórficas.

[Diapositiva 33: 3. Modelos de secuencias (RNN, LSTM)]

3. Modelos de secuencias (RNN, LSTM)

Beneficios:

- Aprenden automáticamente representaciones complejas de las características.
- Reducen la necesidad de ingeniería manual de características.

Los modelos de secuencias, como las redes neuronales recurrentes (RNN) y las redes de memoria a corto y largo plazo (LSTM), capturan dependencias temporales y patrones secuenciales.

Aplicación:

- Analizan registros de comportamiento dinámico o secuencias de llamadas a la API y al sistema.
- Modelan la evolución temporal del comportamiento del malware durante su ejecución.
- Detectan patrones sospechosos en secuencias de eventos indicativos de actividad de malware.

[Diapositiva 34: 3. Modelos de secuencias (RNN, LSTM)]

3. Modelos de secuencias (RNN, LSTM)

Beneficios:

- Capturan el contexto y las dependencias de largo alcance.
- Son eficaces para el análisis dinámico de malware y el modelado del comportamiento.

[Diapositiva 35: Integración de Random Forests, CNN y modelos de secuencias]

Integración de Random Forests, CNN y modelos de secuencias

Integrar Random Forests, CNN y modelos de secuencias en los pipelines de detección de malware permite aprovechar sus puntos fuertes complementarios. Los Random Forests proporcionan una clasificación robusta con datos estáticos, las CNN destacan en el aprendizaje de patrones complejos en datos sin procesar y los modelos de secuencias capturan la dinámica del comportamiento; en conjunto, permiten una detección de malware más precisa y resiliente.

[Diapositiva 36: Ejemplos numéricos]

Ejemplos numéricos

Un sistema de monitorización de red utiliza aprendizaje automático para detectar anomalías (posibles ciberataques) a partir de las características del tráfico de red. El sistema se entrena con datos históricos mediante una One-Class SVM, que se emplea habitualmente en tareas de detección de anomalías.

Las características de entrada utilizadas son:

- X1: número de conexiones por segundo
- X2: tamaño medio de los paquetes
- X3: porcentaje de intentos de conexión fallidos

[Diapositiva 37: Datos de entrenamiento (tráfico normal)]

Datos de entrenamiento (tráfico normal)

El modelo se entrena únicamente con datos de tráfico normal para aprender el comportamiento típico y distinguirlo de las anomalías.

Muestra	X1 (conex./s)	X2 (tamaño de paquete)	X3 (% de fallos)
1	5.0	300	1.0
2	7.0	280	0.5
3	6.0	310	0.8

Co-funded by the European Union

4	5.5	295	1.2

[Diapositiva 38: One-Class SVM]

One-Class SVM

La One-Class SVM aprende que el tráfico normal se encuentra dentro de esta región del espacio de características.

XA = [6.5, 290, 1.0]

Predicción: Normal

Esta entrada está cerca de la región normal aprendida por el modelo, por lo que no se detecta ninguna anomalía.

XB = [20, 100, 15]

[Diapositiva 39: One-Class SVM]

One-Class SVM

Predicción: ¡Anomalía detectada!

- Alta tasa de conexiones
- Tamaño de paquete inusualmente pequeño
- Alta tasa de fallos

Estos valores se desvían significativamente de los patrones normales, lo que sugiere un probable ataque de denegación de servicio (DoS).

El sistema puede responder a XB de las siguientes maneras:

- Activando una alerta
- Limitando o bloqueando la IP de origen
- Notificando al equipo de operaciones de seguridad

[Diapositiva 40: One-Class SVM]

One-Class SVM

Contexto: supongamos que tenemos un modelo de detección de malware que clasifica los archivos como maliciosos (1) o benignos (0) a partir de tres características:

- X1: frecuencia del opcode MOV

- X2: número de llamadas de red
- X3: uso de cadenas de texto sospechosas

Modelo base (regresión logística simplificada):

$\hat{y} = \text{sigmoid}(w1\ X1 + w2\ X2 + w3\ X3 + b)$

[Diapositiva 41: One-Class SVM]

One-Class SVM

Parámetros:

$w = [2.0, 1.5, 2.5], \quad b = -5$

$X = [3, 4, 2] \Rightarrow \hat{y} = \text{sigmoid}(2 \cdot 3 + 1.5 \cdot 4 + 2.5 \cdot 2 - 5) = \text{sigmoid}(6 + 6 + 5 - 5) = \text{sigmoid}(12) \approx 0.9999 \Rightarrow$ Detectado como malicioso

El atacante calcula los gradientes y aplica una pequeña perturbación adversarial:

$X' = [2.5, 2, 1.5]$ (reduce ligeramente los valores clave)

$\Rightarrow \hat{y} = \text{sigmoid}(2 \cdot 2.5 + 1.5 \cdot 2 + 2.5 \cdot 1.5 - 5) = \text{sigmoid}(5 + 3 + 3.75 - 5) = \text{sigmoid}(6.75) \approx 0.9988$

[Diapositiva 42: One-Class SVM]

One-Class SVM

Sigue siendo detectado como malicioso, pero más cerca del umbral.

$X'' = [1.5, 1, 1.0] \Rightarrow \hat{y} = \text{sigmoid}(2 \cdot 1.5 + 1.5 \cdot 1 + 2.5 \cdot 1.0 - 5) = \text{sigmoid}(3 + 1.5 + 2.5 - 5) = \text{sigmoid}(2) \approx 0.88$

Con un umbral de 0,9, esta entrada podría clasificarse erróneamente como benigna, evadiendo con éxito el sistema.