Luces y sombras de la aplicación de técnicas de aprendizaje automático en la monitorización de redes

Jorge E. López de Vergara Méndez

Departamento de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior, Universidad Autónoma de Madrid

jorge.lopez_vergara@uam.es

ESNOG/GORE 29, Madrid, 18 de mayo de 2023

UAM
Universidad Autónoma de Madrid

Escuela Politécnica Superior

Índice

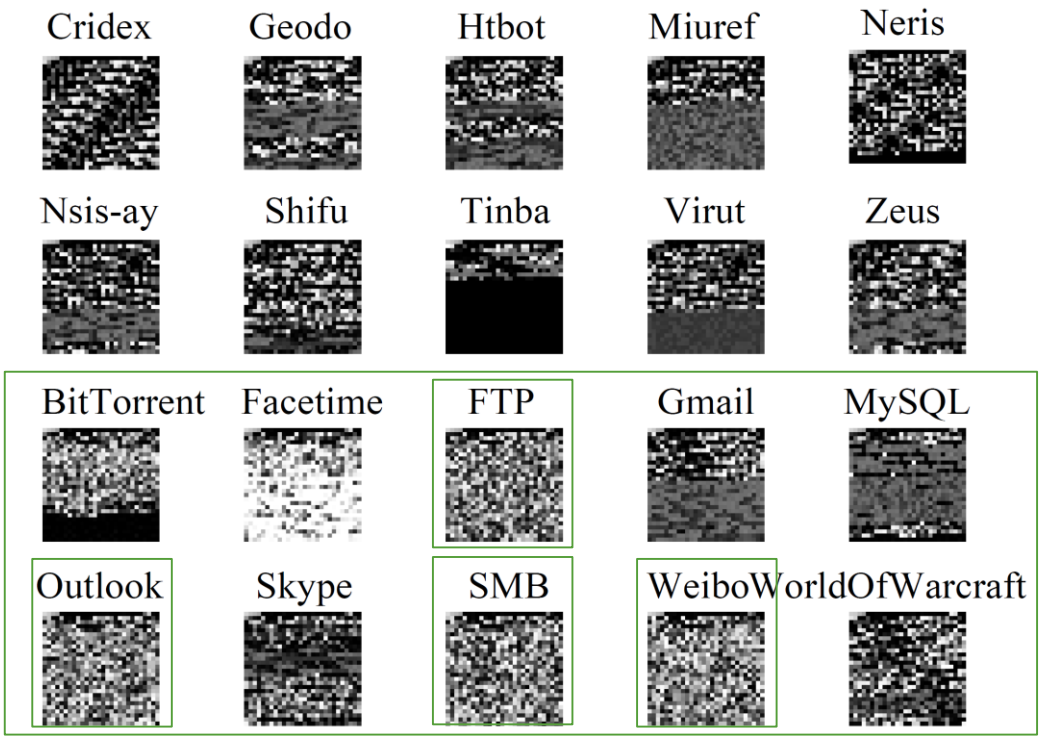
1. Motivación.
2. Problemas detectados.
3. Soluciones aplicables.
4. Casos de éxito.
5. Conclusiones.

Motivación

- Necesidad de clasificar el tráfico de la red.
  - Aplicación de políticas de calidad de servicio.
  - Tarifación.
  - Detección de anomalías y ataques.
- Hoy en día el tráfico está cifrado.
  - ¿Direcciones y puertos? ¿Contenido de paquetes?
  - Clasificación estadística.
  - Flujos extendidos (tamaños mínimo, medio y máximo de los paquetes, etc.)
- Múltiples trabajos basados en técnicas de aprendizaje automático.
  - Desde árboles de decisión a redes neuronales convolucionales.
  - ¡No es oro todo lo que reluce!
    - Muchos trabajos tienen sesgos o son irreproducibles.

Motivación

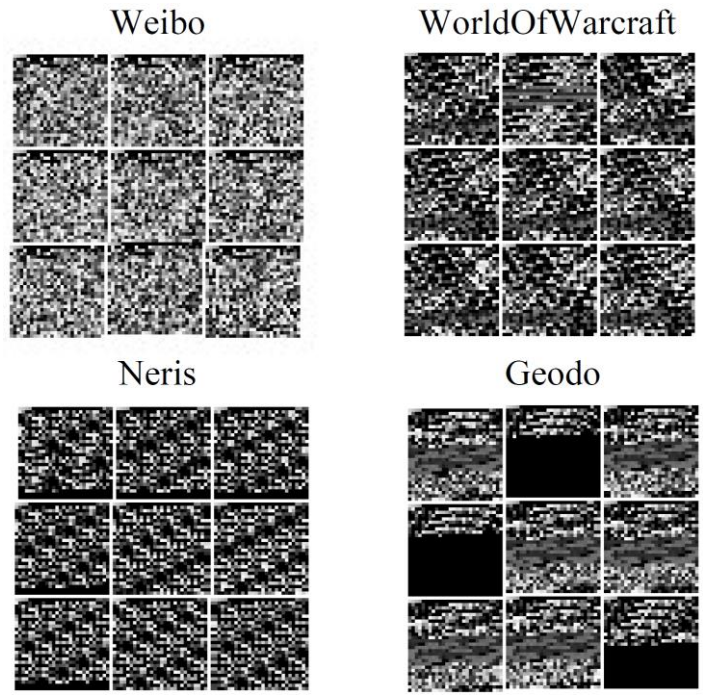
- Si viéramos los paquetes cómo imágenes, ¿qué veríamos?



Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng. "Malware traffic classification using convolutional neural network for representation learning". In *2017 International Conference on Information Networking (ICOIN)*, pp. 712-717. IEEE, 2017.

Motivación

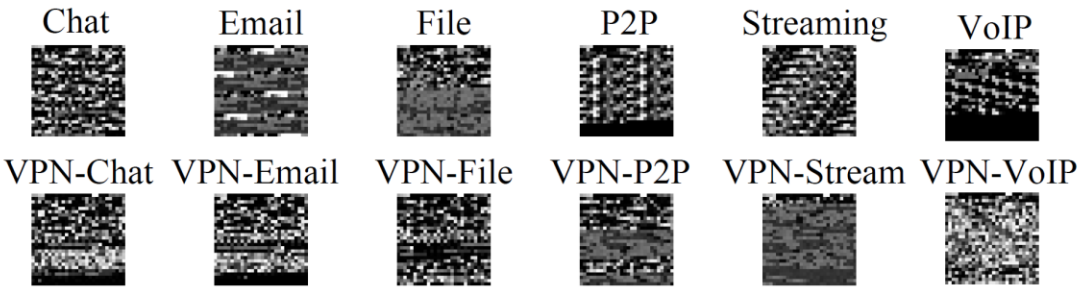
- ¿Se parecen visualmente los paquetes de una misma clase?



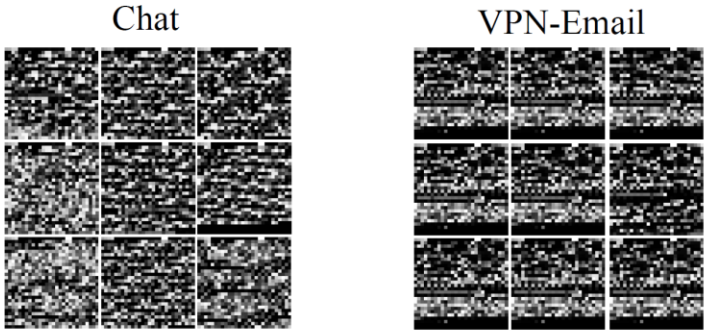
Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye, and Yiqiang Sheng. “Malware traffic classification using convolutional neural network for representation learning”. In *2017 International Conference on Information Networking (ICOIN)*, pp. 712-717. IEEE, 2017.

Motivación

- ¿Y qué pasa si el tráfico está cifrado?



a) Visualization of all classes of traffic

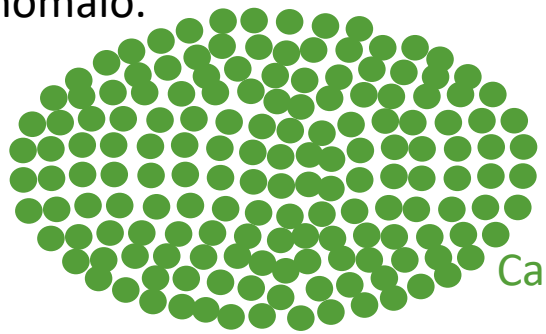


b) Consistency in the same traffic class

Wei Wang, Ming Zhu, Jinlin Wang, Xuewen Zeng, and Zhongzhen Yang. "End-to-end encrypted traffic classification with one-dimensional convolution neural networks". In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pp. 43-48. IEEE, 2017.

Problemas detectados (I)

- La paradoja de la exactitud *(accuracy paradox)*.
  - Un sistema que clasifique siempre el tráfico como normal tendrá una exactitud del 99,9% si 999 de cada 1000 paquetes son normales…
  - … Pero clasificará sistemáticamente como normal el tráfico anómalo.

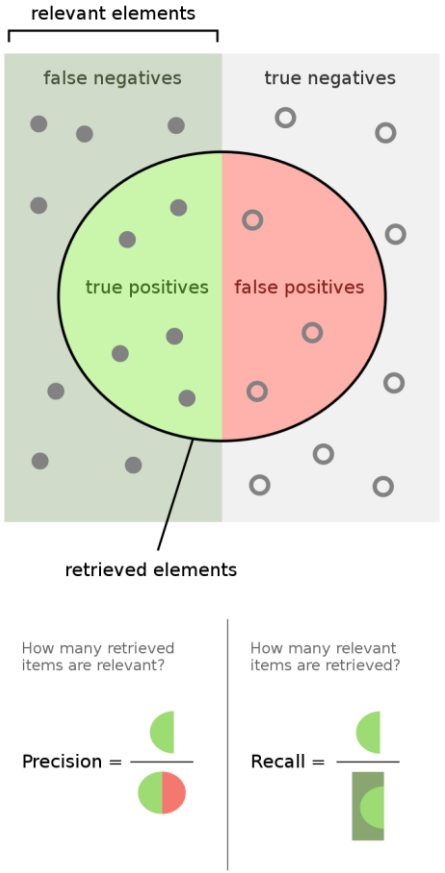


$$Acc = \frac{Casos\ correctos}{Casos\ totales} \cdot 100$$

- Los modelos de aprendizaje automático aprenden mal cuando hay pocas muestras de ataques o anomalías en comparación con el tráfico normal.
- Necesidad de disponer de conjuntos de datos balanceados.

Soluciones aplicables (I)

- Utilizar métricas distintas a la exactitud (*accuracy*)
  - Precisión
  - *Recall* (sensibilidad o exhaustividad)
  - Puntuación F1 (media armónica de precisión y *recall*)
- Especialmente si no es posible balancear bien el conjunto de datos.
- Las matrices de confusión suelen ser muy útiles para comprender mejor qué clases se clasifican mal y con cuáles se están confundiendo.



https://en.wikipedia.org/wiki/Precision_and_recall

Problemas detectados (II)

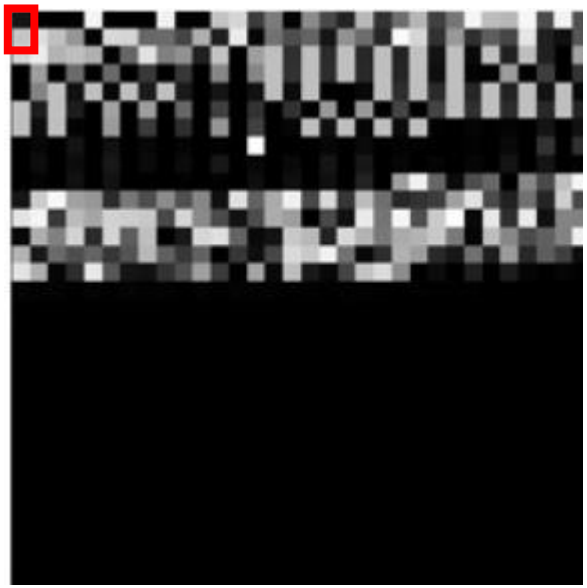
- Conjuntos de datos sesgados y filtración de datos
  - A priori, todos los *datasets* parecen bien generados, pero es posible identificar esos sesgos o filtraciones de datos a posteriori.
  - Ejemplos reales:
    - Se incluye en el conjunto de datos la dirección IP del equipo atacante que se ha usado para generar el *dataset*, y el modelo la utiliza para distinguir del tráfico normal.
    - Para la generación de los ataques se usa la pila TCP/IP de Kali Linux, y para el tráfico de usuarios, una pila TCP/IP de Windows, por lo que el modelo usa esos parámetros de la pila (MSS, ventana…) para identificar un atacante.
    - Variable para indicar el tiempo entre llegadas de paquetes que por error de implementación presenta una marca temporal absoluta, y el modelo usa esta marca temporal para distinguir los ataques del tráfico normal, generados en dos instantes de tiempo distintos.
    - Al repartir entre entrenamiento y test, paquetes de un mismo flujo acaban en ambos conjuntos, lo que hace que los resultados del modelo aparentan ser más altos de lo que son realmente.

Soluciones aplicables (II)

- Separar correctamente el conjunto de datos, evitando que paquetes o registros relacionados aparezcan a la vez en el conjunto de entrenamiento y el de validación.
  - Separar por flujos bidireccionales.
- Eliminar variables que estén sesgando los resultados, o filtrando información al modelo, pues de otra forma, no va a valer en un caso real.
- Analizar bien los conjuntos de datos antes de su procesado, o bien tras el mismo, identificando las variables que resultan más relevantes para el modelo.
- Muchas veces son necesarias las técnicas de Inteligencia Artificial Explicable (XAI) para poder identificar estas variables, tales como SHAP o GradCAM.
  - Evitar siempre ver el modelo como una caja negra, pues puede dar valores altos de rendimiento y no ser útiles en un caso real.
  - Seleccionando las variables más útiles y descartando las variables que puedan estar sesgando los resultados permite buscar modelos más simples que permitan una identificación igual de precisa.
    - Por ejemplo, usar un árbol de decisión en vez de una red convolucional.

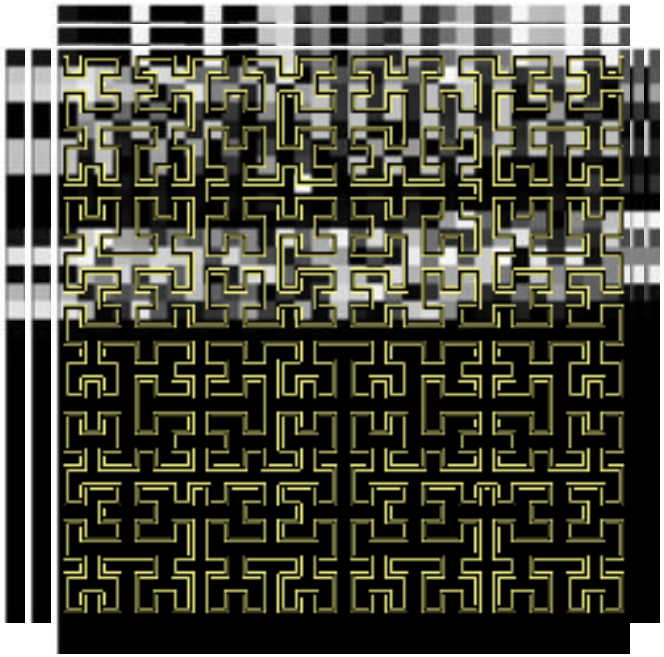
Problemas detectados (III)

- Tratamiento de paquetes como imágenes bidimensionales y convolución con kernels 3x3
- A diferencia de la imagen de una cara, en el caso de los paquetes de red, los bordes de la imagen son especialmente importantes.
- La conversión en una imagen 2D de un paquete puede ser muy arbitraria:
  - Imágenes de 32x32 píxeles:
    - Los paquetes pequeños hay que rellenarlos con píxeles negros.
    - Los paquetes grandes hay que truncarlos a los primeros 1024 bytes.
    - ¿Cabeceras o solo carga útil?
  - La relación del pixel 0 con el pixel 32 probablemente sea mucho más baja que lo que ocurriría en una imagen real.



Soluciones aplicables (III)

- Tratamiento de paquetes como imágenes bidimensionales y convolución con kernels 3x3
  - Necesario ampliar los bordes de la imagen para evitar que se pierda información.
  - Necesario buscar otros métodos de representación para mantener la coherencia espacial de los píxeles respecto a una imagen 1D.
    - Curvas de Hilbert o Peano.
  - Usar imágenes 1D.
    - Son más lentas de procesar en GPUs, al no poder paralelizar las operaciones.
  - Si la imagen es de tráfico cifrado, probablemente lo que la red convolucional identifique y use para clasificar sea el tamaño del paquete.



Problemas detectados (IV)

- Lentitud de los modelos basados en *Deep Learning*
  - Suponiendo que cada paquete se convierte en una imagen de 32x32 píxeles, una GPU moderna puede procesar con un modelo convolucional del orden de miles de paquetes por segundo.
  - Se obtienen resultados parecidos con una FPGA.
  - No aplicable en tiempo real en redes de alta tasa (e.g. ≥10Gbps), con millones de paquetes por segundo. ☹

Soluciones aplicables (IV)

- No siempre es necesario usar un modelo basado en Deep Learning
  - Un modelo basado en un árbol de decisión puede alcanzar, dependiendo del caso, valores de exactitud, precisión y *recall* similares, funcionando a velocidades mucho más altas (dos órdenes de magnitud).
  - Es necesario reducir la dimensionalidad para su aplicación actual. En redes de alta tasa se puede trabajar sobre registros de flujo enriquecidos, más ligeros que procesar cada byte de un paquete.
    - Cuidado, el registro no se exporta hasta que se cierra el flujo.

Casos de éxito (I)

- Clasificación de la navegación de usuarios a partir de los nombres de dominio.
  - Se obtienen a partir de las consultas de DNS o del SNI de TLS.
  - El conjunto de servidores solicitados permiten identificar el sitio accedido.
    - Necesario generar el *dataset* con un cliente sintético.
  - Uso de técnicas de procesado de lenguaje natural, identificando cada nombre de dominio con un término del "lenguaje de navegación".
  - Permite distinguir la navegación de *banners* publicitarios o servicios de terceros.

D. Perdices, J. Ramos, J.L. García-Dorado, I. González, J.E. López de Vergara, ***Natural Language Processing for Web Browsing Analytics: Challenges, Lessons Learned, and Opportunities***, Computer Networks, Vol. 198, October 2021, article no. 108357, Elsevier, ISSN: 1389-1286, doi:10.1016/j.comnet.2021.108357

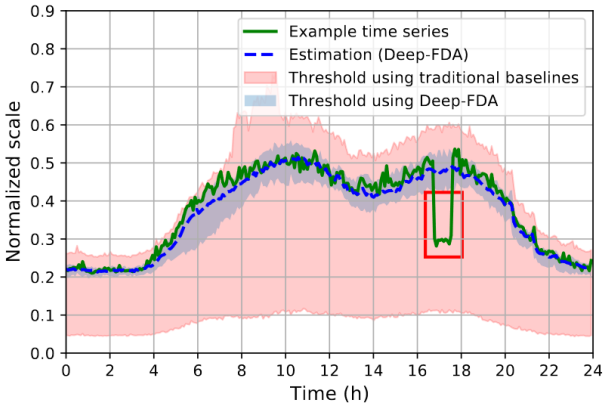
Casos de éxito (II)

- Clasificación de la navegación de usuarios a partir de las direcciones IP accedidas.
  - Sin realizar consultas inversas al DNS.
    - Pueden responder con un servidor de una nube o una CDN.
  - Comparando con las direcciones a las que se accede para los principales dominios más visitados de Internet.
    - Necesario generar el *dataset* con un cliente sintético.
  - Uso de un perceptrón multicapa para clasificar la navegación para cada dominio más visitado.
  - Depende del servidor de DNS que resuelva las consultas del cliente.
    - Entrenar con varios servidores independientes para mejorar los resultados.
    - Necesario reentrenar el modelo según varíen las direcciones de los servidores.

D. Perdices, J.E. López de Vergara, L. de Pedro, I. González, ***Web browsing privacy in the deep learning era: beyond VPNs and encryption***, Computer Networks, Volume 220, January 2023, Elsevier, ISSN: 1389-1286 doi:10.1016/j.comnet.2022.109471

Casos de éxito (III)

- Identificación de anomalías en la red
  - Tratamiento de las series temporales como una familia de funciones ortonormales, cuyos parámetros permiten regenerar la serie.
  - El tráfico futuro es predecible a partir del tráfico pasado, obteniendo los parámetros de la familia de funciones.
  - Útil para anomalías que no son identificables fácilmente utilizando umbrales habituales de tráfico.



D. Perdices, J.E. López de Vergara, J. Ramos, ***Deep-FDA: Using Functional Data Analysis and Neural Networks to Characterize Network Services Time Series***, IEEE Transactions on Network and Service Management, Vol. 18, Issue 1, pp. 986-999, ISSN 1932-4537, doi:10.1109/TNSM.2021.3053835

Conclusiones

- Las técnicas de aprendizaje automático aplicadas correctamente pueden ayudar en la monitorización de red.
- Es necesario prestar especial atención a su funcionamiento, pues un modelo entrenado a partir de un conjunto de datos mal escogido puede generar resultados inútiles en un entorno real.
- Hay que analizar los resultados más allá de la exactitud y la precisión, sin que el algoritmo de aprendizaje automático se vea como una caja negra, usando las técnicas conocidas como XAI.
- La aplicación cuidadosa de estas cuestiones puede permitir alcanzar soluciones exitosas, pero hay que tener siempre claro que el modelo que se genere va a depender del conjunto de datos con que se haya entrenado.

Luces y sombras de la inteligencia artificial en la monitorización de redes

Jorge E. López de Vergara Méndez

Departamento de Tecnología Electrónica y de las Comunicaciones

Escuela Politécnica Superior, Universidad Autónoma de Madrid

jorge.lopez_vergara@uam.es

ESNOG/GORE 29, Madrid, 18 de mayo de 2023