Proyecto Final

Tema 12: Detección de intrusiones en IoT

Asignatura:
Análisis de Datos

Integrantes:
Agustin Castillo
Martín Estefanell

Fecha:
14/10/2025

Descripción del Problema

El crecimiento acelerado del Internet de las Cosas (IoT) ha generado un ecosistema compuesto por millones de dispositivos interconectados que recopilan, procesan y comparten información en tiempo real. Esta interconectividad, aunque ofrece ventajas en automatización y eficiencia, también introduce nuevas vulnerabilidades, ya que muchos dispositivos IoT carecen de recursos y mecanismos de seguridad avanzados.

Como resultado, las redes IoT se han convertido en un objetivo atractivo para los ciberataques, exponiendo tanto a usuarios como a infraestructuras críticas a amenazas que pueden comprometer la integridad, la disponibilidad y la confidencialidad de los datos.

Ante este panorama, los sistemas de detección de intrusiones (IDS) se presentan como una herramienta esencial para monitorear el tráfico de red y reconocer patrones anómalos que indiquen comportamientos maliciosos.

En este proyecto se aborda el análisis del conjunto de datos RT-IoT2022, que recopila registros reales de tráfico normal y ataques en redes IoT, con el propósito de comprender sus características, detectar inconsistencias y, posteriormente, desarrollar modelos predictivos capaces de diferenciar entre actividad legítima y maliciosa.

Este análisis busca aportar conocimiento aplicado a la mejora de la seguridad en entornos IoT y servir de base para futuros sistemas de detección automatizados.

Objetivos

El objetivo principal de este proyecto es predecir si una actividad en la red IoT corresponde a un comportamiento normal o a un posible ataque, a partir del análisis del conjunto de datos RT-IoT2022. Para ello, se aplicarán técnicas de análisis de datos, limpieza, visualización y modelado predictivo que permitan comprender en profundidad las características del tráfico de red y detectar patrones asociados a comportamientos anómalos.

El desarrollo del modelo busca aportar una herramienta capaz de asistir en la detección temprana de intrusiones dentro de entornos IoT, donde la cantidad de dispositivos conectados y la diversidad de protocolos incrementan el riesgo de vulnerabilidades. A través del uso de métodos de aprendizaje automático y evaluación de métricas de desempeño, se pretende identificar las variables más influyentes en la clasificación y contribuir a la mejora de los mecanismos de seguridad en redes inteligentes.

Alcance

El presente proyecto se centra en el análisis y modelado del conjunto de datos RT-IoT2022, que contiene registros de tráfico de red obtenidos en entornos IoT bajo condiciones normales y de ataque. El trabajo abarca desde la exploración y preprocesamiento de los datos hasta la construcción y evaluación de un modelo predictivo, con el fin de clasificar las conexiones de red como normales o maliciosas. Se utilizarán técnicas de análisis exploratorio, visualización y algoritmos de aprendizaje supervisado para identificar patrones y comportamientos relevantes en la detección de intrusiones.

El alcance del estudio se limita al tratamiento y análisis del dataset provisto, sin intervenir en la captura de datos en entornos reales ni en la implementación de un sistema de detección en producción. No obstante, los resultados obtenidos podrán servir como base para el desarrollo futuro de sistemas de detección automatizados (IDS) o la optimización de modelos de seguridad aplicados a redes IoT. De este modo, el proyecto busca generar conocimiento aplicable y contribuir a la comprensión de los desafíos actuales en la protección de dispositivos conectados.

Descripción del conjunto de datos

1. *Identificadores Clave*
    a. id.orig_p: Número de puerto de origen del flujo
    b. id.rest_p: Número de puerto de destino
    c. proto: Protocolo de red utilizado
    d. service: Tipo de servicio o aplicación asociada al flujo
    e. unnamed: Identificador automático de fila generado al exportar el dataset.

2. *Variables Temporales*
    a. flow_duration: Duración total del flujo
    b. active.min, active.max, active.tot, active.avg, active.std: Medidas estadísticas de los períodos activos de comunicación
    c. idle.min, idle.max, idle.tot, idle.avg, idle.std: Medidas estadísticas de los períodos inactivos

3. *Volumen y tamaño de tráfico*
    a. fwd_pkts_tot, bwd_pkts_tot: Total de paquetes enviados (forward) y recibidos (backward)
    b. fwd_data_pkts_tot, bwd_data_pkts_tot:Total de paquetes que contienen datos (sin incluir cabeceras) enviados y recibidos.
    c. fwd_pkts_per_sec, bwd_pkts_per_sec, flow_pkts_per_sec: Tasa de paquetes por segundo en cada dirección y total del flujo.
    d. down_up_ratio: Relación entre paquetes descendentes y ascendentes (indicador de asimetría en el flujo).

4. *Tamaño de Cabecera (Header)*
    a. fwd_header_size_tot, bwd_header_size_tot: Suma total del tamaño de las cabeceras TCP/IP en cada dirección.
    b. fwd_header_size_min, fwd_header_size_max: Tamaño mínimo y máximo de las cabeceras en dirección forward.
    c. bwd_header_size_min, bwd_header_size_max: Tamaño mínimo y máximo de las cabeceras en dirección backward.

5. *Variables Tipo Bandera*
    a. *flow_FIN_flag_count*: Número de paquetes con bandera FIN (finalización de conexión)
    b. *flow_SYN_flag_count*: Número de paquetes con bandera SYN (inicio de conexión TCP).
    c. *flow_RST_flag_count*: Número de paquetes con bandera RST (reinicio de conexión).

d. *fwd_PSH_flag_count, bwd_PSH_flag_count*: Paquetes con bandera PSH (push), enviados y recibidos.
e. *flow_ACK_flag_count*: Número de paquetes con bandera ACK (confirmación de recepción).
f. *fwd_URG_flag_count, bwd_URG_flag_count*: Paquetes con bandera URG (urgente).
g. *flow_CWR_flag_count*: Paquetes con bandera CWR (congestión).
h. *flow_ECE_flag_count*: Paquetes con bandera ECE (notificación de congestión ECN).

6. Estadísticas de Payload
    a. *fwd_pkts_payload.min, fwd_pkts_payload.max, fwd_pkts_payload.tot, fwd_pkts_payload.avg, fwd_pkts_payload.std*: Medidas estadísticas (mínimo, máximo, total, promedio y desviación) del tamaño del payload en dirección forward.
    b. *bwd_pkts_payload.min, bwd_pkts_payload.max, bwd_pkts_payload.tot, bwd_pkts_payload.avg, bwd_pkts_payload.std*: Idem anterior, pero para la dirección backward.
    c. *flow_pkts_payload.min, flow_pkts_payload.max, flow_pkts_payload.tot, flow_pkts_payload.avg, flow_pkts_payload.std*: Medidas combinadas de la carga útil total del flujo (ambas direcciones).
    d. *payload_bytes_per_second*: Tasa de bytes de carga útil transmitidos por segundo.

7. *Tiempo entre Paquetes*
    a. *fwd_iat.min, fwd_iat.max, fwd_iat.tot, fwd_iat.avg, fwd_iat.std*: Intervalos de tiempo entre paquetes enviados (forward).
    b. *bwd_iat.min, bwd_iat.max, bwd_iat.tot, bwd_iat.avg, bwd_iat.std*: Intervalos entre paquetes recibidos (backward).
    c. *flow_iat.min, flow_iat.max, flow_iat.tot, flow_iat.avg, flow_iat.std*: Intervalos entre paquetes considerando ambas direcciones.

8. *Subflujos, Bulk y Ventanas TCP*
    a. *fwd_subflow_pkts, bwd_subflow_pkts*: Número de paquetes en los subflujos (por dirección).
    b. *fwd_subflow_bytes, bwd_subflow_bytes*: Bytes transmitidos en los subflujos (por dirección).
    c. *fwd_bulk_bytes, bwd_bulk_bytes*: Bytes transmitidos en ráfagas (bulk transfers).
    d. *fwd_bulk_packets, bwd_bulk_packets*: Paquetes enviados en ráfagas (bulk).
    e. *fwd_bulk_rate, bwd_bulk_rate*: Tasa de transmisión en ráfagas (bulk rate).
    f. *fwd_init_window_size, bwd_init_window_size*: Tamaño inicial de la ventana TCP por dirección.
    g. *fwd_last_window_size*: Tamaño de la última ventana TCP registrada.

9. Variable Objetivo
    a. *Attack_type*: Indica si el flujo corresponde a tráfico normal o a un ataque específico (por ejemplo, DoS, spoofing, ransomware, etc.). Es la variable a predecir.

Análisis Exploratorio & Protocolo de Limpieza

El análisis exploratorio tuvo como objetivo comprender en profundidad la estructura y composición del conjunto de datos RT-IoT2022, identificar posibles inconsistencias y obtener una primera aproximación al comportamiento del tráfico de red en entornos IoT.

Esta etapa busca detectar patrones, sesgos y relaciones entre variables que justifiquen las decisiones de limpieza, transformación y modelado en etapas posteriores. En particular, se pretende reconocer comportamientos característicos del tráfico normal frente a las distintas categorías de ataques, de modo de sentar las bases para la detección de anomalías.

El dataset cuenta con 85 variables y una cantidad significativa de registros que representan tanto comunicaciones legítimas entre dispositivos IoT como flujos maliciosos generados mediante diversos tipos de ataques.

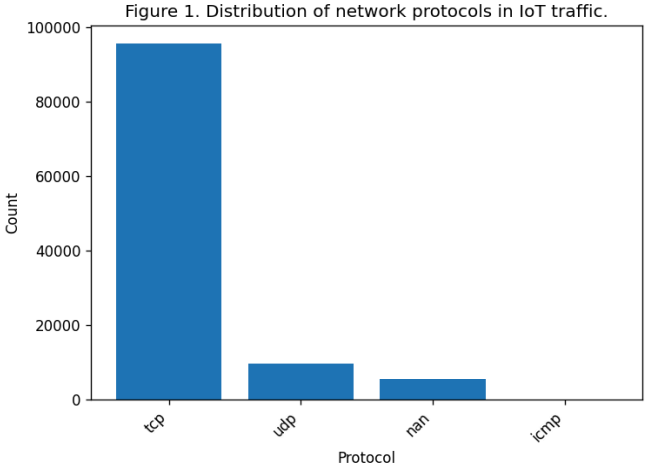
Para facilitar la interpretación, las variables fueron agrupadas según su objetivo y propósito dentro del dataset:

1. Tipo de Comunicación

El análisis inicial se centró en comprender los protocolos y servicios presentes en el tráfico IoT, ya que constituyen la capa más descriptiva del comportamiento de la red y permiten identificar patrones característicos de comunicación normal y de ataque. Examinar esta información antes que las variables numéricas es fundamental para garantizar la integridad del dataset y para descubrir posibles indicadores tempranos de anomalías.

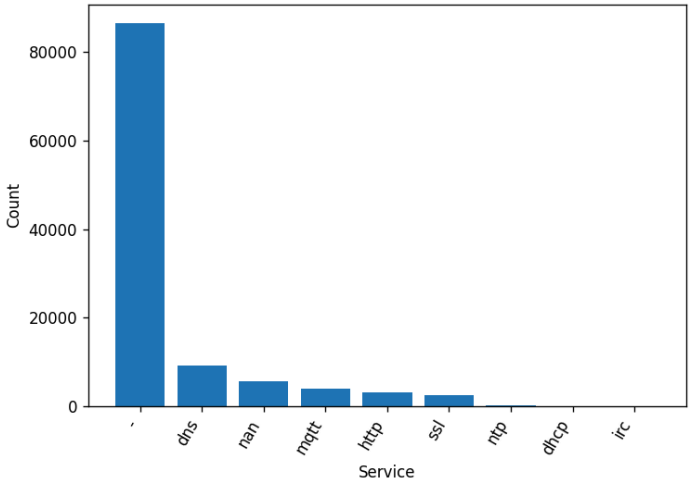
Los resultados muestran una clara predominancia del protocolo TCP, seguido por UDP en una proporción significativamente menor. Este comportamiento es esperable en entornos IoT, donde la mayoría de las aplicaciones de control, transmisión y actualización utilizan conexiones confiables basadas en TCP. Sin embargo, la presencia de registros con valores vacíos o nan indica inconsistencias en la captura o clasificación del tráfico, lo que resalta la necesidad de procesos de limpieza y normalización previos al modelado.

Figure 1. Distribution of network protocols in IoT traffic.



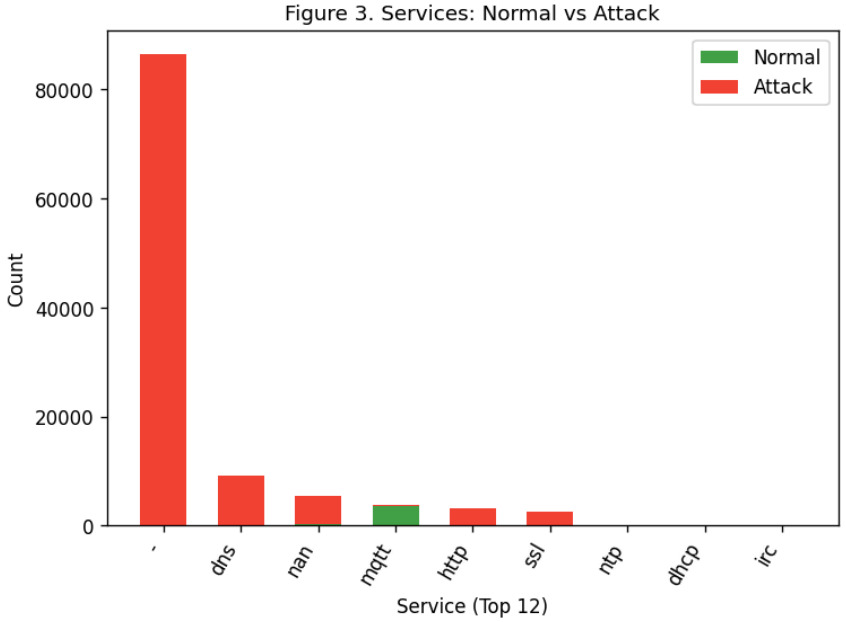
En cuanto a los servicios, se observó que DNS, HTTP, MQTT y SSL son los más frecuentes, reflejando la mezcla típica de comunicaciones de dispositivos IoT (por ejemplo, MQTT en tráfico legítimo de publicación/suscripción). No obstante, también aparecen servicios como DHCP, IRC y NTP, menos comunes en flujos de operación normal, lo que podría estar asociado a intentos de exploración o abuso de servicios para ejecutar ataques distribuidos.

Figure 2. Most frequent services observed in IoT communications.



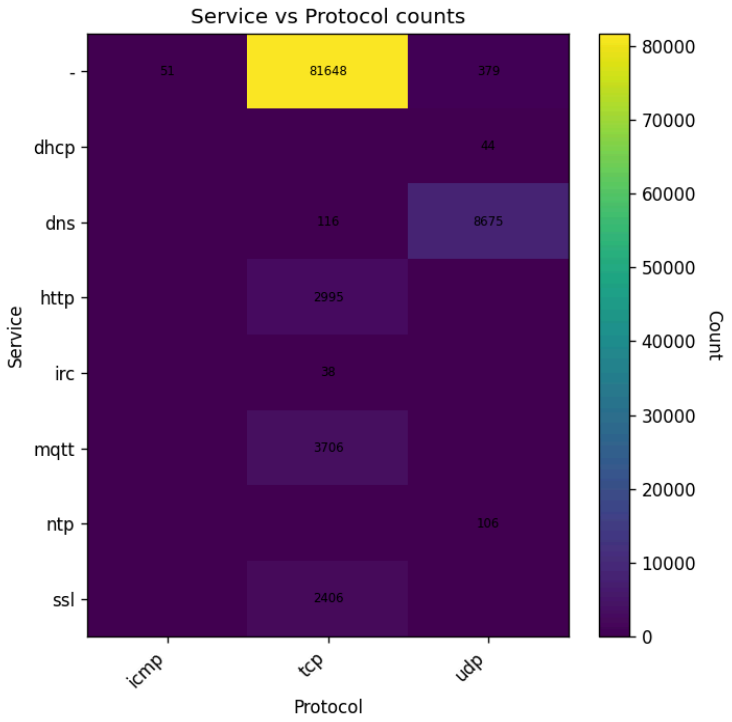
Al analizar la distribución entre tráfico normal y malicioso, se detectó que la mayoría de los ataques se concentran en servicios DNS y sin clasificación específica (nan) mientras que MQTT predomina en flujos normales. Esto confirma que los servicios y protocolos pueden actuar como variables discriminantes entre comportamientos benignos y ataques, aportando un contexto de alto valor para la construcción del modelo de detección.

Figure 3. Services: Normal vs Attack



Finalmente, el heatmap de servicio y protocolo evidencia que la mayoría de los servicios relevantes (HTTP, MQTT, SSL, DNS) operan sobre TCP, mientras que UDP se asocia casi exclusivamente con DNS. Esta correlación entre capa de transporte y aplicación refuerza la coherencia interna del dataset

y brinda una base sólida para los próximos análisis sobre intensidad de flujo y comportamiento temporal.



A partir de los resultados obtenidos, se decidió aplicar un proceso de limpieza específico sobre las variables que describen el tipo de comunicación, con el objetivo de corregir inconsistencias y asegurar una interpretación homogénea de los flujos.

En primer lugar, se normalizaron las variables categóricas proto, service y attack_type, convirtiendo todos sus valores a minúsculas y eliminando espacios o caracteres residuales. Esto permitió unificar categorías duplicadas (por ejemplo, “TCP”, “tcp ” y “Tcp”) y mejorar la consistencia en los análisis posteriores.

Los registros con valores vacíos o nan en los campos de protocolo o servicio fueron revisados. Dado que representan flujos sin clasificación clara, se optó por reemplazarlos por la categoría "desconocido", a fin de preservar la estructura del tráfico sin eliminar datos potencialmente relevantes.

En el caso de las variables id.orig_p y id.resp_p (puertos de origen y destino), se verificó su validez dentro del rango permitido (0–65535). Los valores fuera de rango o no numéricos se eliminaron por considerarse inconsistentes, aunque no se detectaron inconsistencias en el análisis.

Además, se evaluó la coherencia entre los puertos y los servicios declarados (por ejemplo, http → 80) Durante la revisión de coherencia se identificaron registros etiquetados como tráfico normal (mqtt_publish) que presentan combinaciones de protocolo, servicio y puerto incompatibles con el funcionamiento real de dichos servicios. Dado que estas inconsistencias probablemente se deban a errores de captura o etiquetado, se decidió eliminar estos registros del conjunto de datos para evitar introducir ruido en la clase normal.

En cambio, los flujos con incoherencias en etiquetas de ataque o sin etiqueta se conservaron para un análisis más detallado, ya que podrían representar comportamientos anómalos o intentos de evasión.

Se presenta una tabla de criterios considerados como inconsistencias técnicas:

Service	Proto	id.resp_p	Criterio de Coherencia
HTTP	TCP	80	Se considera correcto si proto = tcp y el puerto de destino es 80 o 8080.
HTTPS/SSL	TCP	443	Válido si proto = tcp y id.resp_p = 443 o 8443.
DNS	UDP/TCP	53	Coherente si proto = udp o tcp, y el puerto de destino es 53.
MQTT	TCP	1883/8883	Correcto si proto = tcp y el puerto de destino coincide con 1883 (sin cifrado) o 8883 (con SSL).
SSH	TCP	22	Válido si proto = tcp y id.resp_p = 22.
FTP	TCP	21	Coherente si proto = tcp y id.resp_p = 21.
DHCP	UDP	67/68	Correcto si proto = udp y el puerto de destino está entre 67–68.
NTP	UDP	123	Válido si proto = udp y id.resp_p = 123.
IRC	TCP	6660-6669	Coherente si proto = tcp y el puerto de destino está en ese rango.

Además de la validación entre servicio y puerto, se realizó un control complementario para verificar la coherencia entre el protocolo de transporte (proto) y el rango de puertos utilizados. En condiciones normales, los servicios basados en TCP (como HTTP, SSH o MQTT) no deberían operar sobre puertos reservados a UDP (por ejemplo, 67/68 de DHCP o 123 de NTP), y viceversa.

La presencia de combinaciones de este tipo (ej: proto = udp con id.resp_p = 80 o proto = tcp con id.resp_p = 68) puede responder a errores de captura o clasificación del tráfico, pero también a tácticas de evasión empleadas por ciertos ataques, que buscan camuflar su comunicación bajo protocolos o puertos atípicos.

Por este motivo, se adoptó un criterio diferenciado:

- Los registros etiquetados como tráfico normal y con incoherencias entre protocolo y puerto fueron eliminados, por considerarse ruido o errores de registro.

- En cambio, aquellos etiquetados como ataques o sin etiqueta se mantuvieron para un análisis posterior, ya que podrían representar comportamientos maliciosos o experimentales relevantes para la detección de anomalías.

En la columna Attack_type, se detectaron registros con valores vacíos que no indican explícitamente si corresponden a tráfico normal o malicioso. Dado que la ausencia de etiqueta puede deberse tanto a errores de anotación como a flujos legítimos no clasificados, se decidió crear una categoría adicional denominada “unknown” para mantener estos registros sin asignar falsamente una clase. De esta manera, se preserva la información original del tráfico y se evita introducir sesgos al diferenciar entre actividad normal y ataques identificados.

Se unificaron las distintas representaciones de servicio no identificado (“–”, “none”, “unknown”) bajo la categoría común "unknown", garantizando consistencia en los conteos y visualizaciones, y además se unificaron etiquetas de ataque con diferencias menores de escritura o formato (guiones, mayúsculas o espacios), reduciendo el número de categorías redundantes y mejorando la claridad del conjunto de clases.

2. *Intensidad y tamaño de flujo.*

El propósito de esta sección es caracterizar la magnitud y la carga del tráfico IoT, analizando tanto el volumen de información intercambiada entre los dispositivos como la velocidad con que se produce dicha comunicación. Este análisis busca identificar:

- Qué flujos son más “pesados”, es decir, aquellos con mayor volumen de bytes y paquetes transmitidos.
- Diferencias de comportamiento entre el tráfico normal y el malicioso, evaluando si los ataques se distinguen por su intensidad o tamaño.

La presencia de picos anómalos o valores atípicos (outliers) pueden actuar como indicadores tempranos de intrusión o comportamientos irregulares. Comprender la distribución y el comportamiento de estas variables permite justificar transformaciones posteriores, como la normalización de escalas, el uso de funciones logarítmicas para reducir la asimetría o el recorte de valores extremos. De esta forma, se prepara el dataset para el modelado y la detección de anomalías.

En términos generales, se espera que los flujos normales en entornos IoT (por ejemplo, comunicaciones basadas en MQTT o HTTP keep-alive) presenten bajo volumen, pocos paquetes y duraciones constantes, reflejando transmisiones livianas y periódicas entre dispositivos.

Por el contrario, los ataques DoS o de fuerza bruta suelen exhibir altos volúmenes de tráfico en intervalos muy cortos, evidenciando tasas elevadas de bytes por segundo y un número inusual de paquetes. Finalmente, los ataques de escaneo o reconocimiento se manifiestan como múltiples flujos diminutos y de corta duración, generados para explorar o mapear la red.

Estas diferencias permiten identificar variables con alto poder discriminante entre comportamientos benignos y maliciosos, estableciendo una base sólida para el modelo predictivo posterior.

Con el objetivo de evaluar la redundancia entre las variables de intensidad y tamaño del flujo, se aplicó un Análisis de Componentes Principales (PCA) sobre las variables numéricas previamente estandarizadas.

El PCA permitió identificar qué atributos concentran mayor varianza y cómo se relacionan entre sí, facilitando la detección de métricas redundantes o fuertemente correlacionadas. Los resultados mostraron que unas pocas componentes explican la mayor parte de la variabilidad total, lo que sugiere

que el comportamiento del tráfico IoT puede representarse de forma compacta mediante combinaciones lineales de variables clave como el volumen total, la tasa de bytes por segundo y la duración del flujo. Esta reducción exploratoria contribuye a optimizar el conjunto de características antes del modelado predictivo.

Para este análisis, se consideraron las siguientes familias de variables:

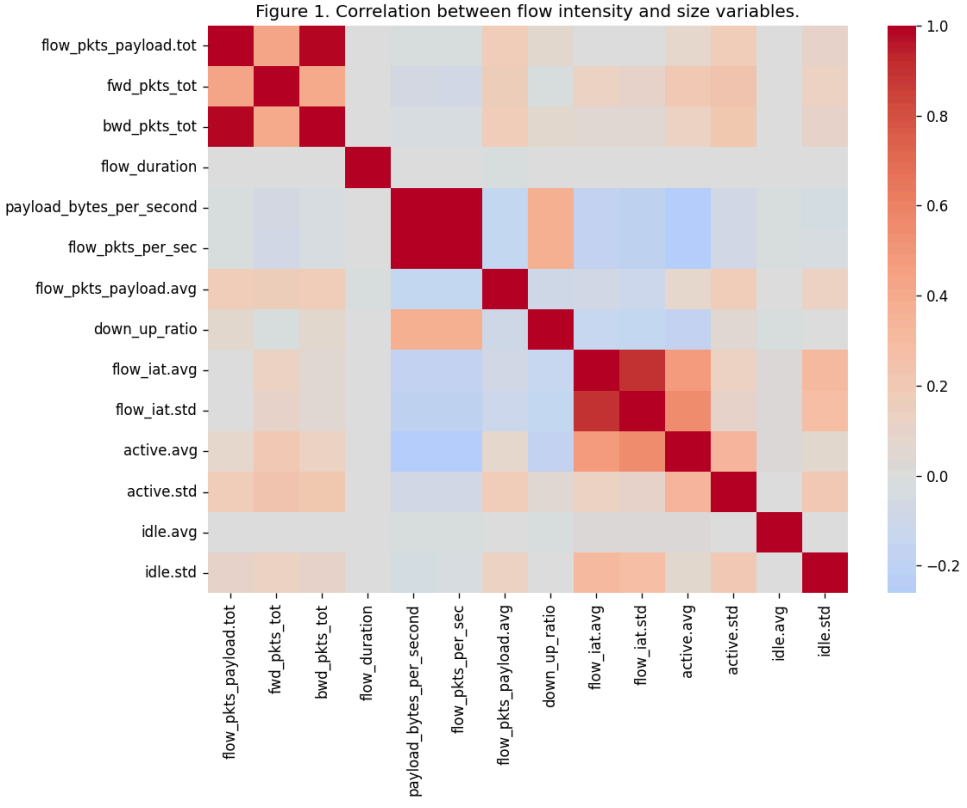
Grupo	Variables Incluidas
Volumen Total y Tamaño de Flujo	flow_pkts_payload.tot, fwd_pkts_tot, bwd_pkts_tot, flow_duration
Ritmo e intensidad del tráfico	payload_bytes_per_second, flow_pkts_per_sec, flow_pkts_payload.avg
Relaciones y proporciones	down_up_ratio
Regularidad temporal y comportamiento dinámico	flow_iat.avg, flow_iat.std, active.avg, active.std, idle.avg, idle.std

El estudio de las métricas asociadas a la magnitud y la velocidad del tráfico IoT permitió identificar patrones claros de diferenciación entre flujos normales y maliciosos. Las variables fueron analizadas tanto de manera individual (distribuciones, medidas descriptivas) como en conjunto (correlaciones, variabilidad y PCA).

La matriz de correlaciones muestra una estructura de dependencias fuerte entre las variables de volumen total (flow_pkts_payload.tot, fwd_pkts_tot, bwd_pkts_tot) y las tasas de transmisión (payload_bytes_per_second, flow_pkts_per_sec).

Esto evidencia que los indicadores de magnitud y ritmo capturan la cantidad total de información intercambiada. En cambio, las variables de regularidad temporal (flow_iat.avg, flow_iat.std, active.avg, idle.avg) presentan correlaciones bajas, lo que confirma que aportan información complementaria sobre el comportamiento temporal de los flujos y deben conservarse en el conjunto final.

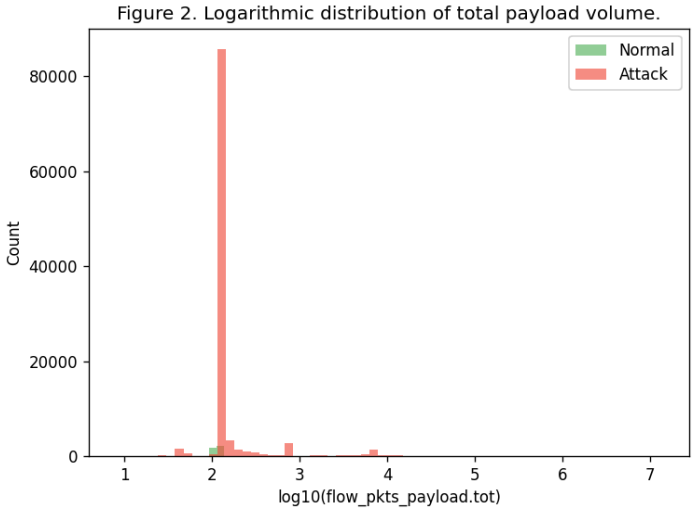
Figure 1. Correlation between flow intensity and size variables.



El histograma logarítmico de flow_pkts_payload.tot revela una distribución fuertemente asimétrica a la derecha, con la mayoría de los flujos concentrados en valores bajos (característico del tráfico IoT regular) y un conjunto reducido de flujos con volúmenes extremadamente altos, asociados principalmente a ataques.

Estos picos aislados confirman la presencia de comportamientos anómalos y justifican el uso de transformaciones logarítmicas o recortes para estabilizar la escala.

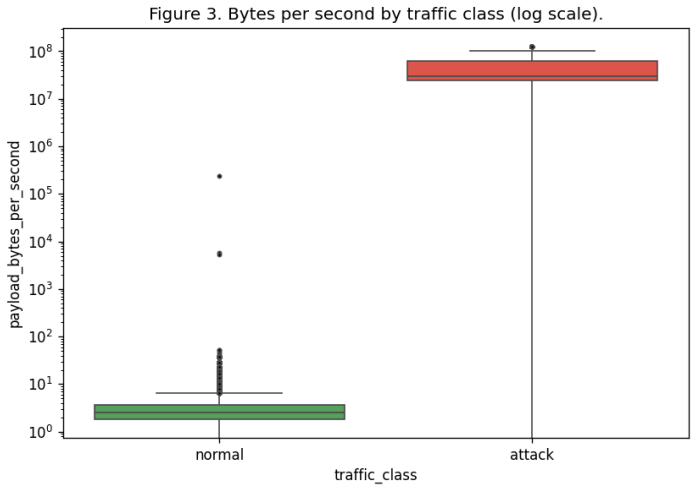
Figure 2. Logarithmic distribution of total payload volume.



El boxplot comparativo de tasas de bytes por segundo evidencia una brecha significativa entre ambas clases, los flujos normales mantienen valores bajos y homogéneos, mientras que los de ataque alcanzan tasas hasta siete órdenes de magnitud mayores.

Este contraste indica que la variable payload_bytes_per_second es una de las más discriminantes del bloque y refleja directamente la agresividad del tráfico malicioso.

Figure 3. Bytes per second by traffic class (log scale).

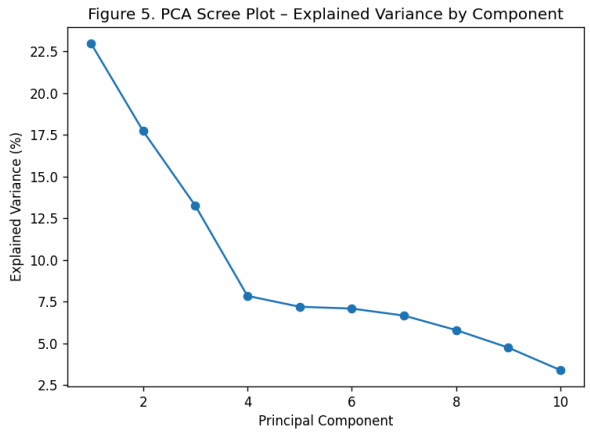


El análisis de dispersión relativo mediante el coeficiente de variación (CV) (eda_intensity_cv.xlsx) muestra que las métricas de tasa (payload_bytes_per_second, flow_pkts_per_sec) son las más inestables, seguidas por las relacionadas con intervalos (flow_iat.std) y períodos activos (active.std).

Por el contrario, variables como idle.avg o flow_iat.avg exhiben menor variabilidad, lo que indica que el tráfico benigno tiende a mantener ritmos regulares y pausas predecibles.

Los registros de outliers (eda_intensity_outliers.xlsx) refuerzan esta observación, destacando un porcentaje reducido pero significativo de flujos con valores extremos en volumen y tasa, todos pertenecientes a tráfico de ataque.

Por último, el Análisis de Componentes Principales (PCA) confirmó la existencia de estructura que muestra causalidad entre las variables del bloque. El scree plot (eda_intensity_pca_scree.png) muestra que las tres primeras componentes explican alrededor del 55 % de la varianza total, concentrando la mayor información en variables de magnitud y tasa (payload_bytes_per_second, flow_pkts_per_sec, flow_pkts_payload.tot).



La proyección bidimensional PC1–PC2 revela una separación visible entre tráfico normal y de ataque, donde los ataques se ubican en zonas de alta varianza y magnitud, mientras que los flujos benignos se concentran en regiones estables de baja intensidad.

La tabla de cargas (eda_intensity_pca_loadings.xlsx) respalda este hallazgo, indicando que las variables más influyentes son las asociadas al volumen total y la velocidad de transmisión.

Con respecto a la limpieza, aquellos registros de flujo que tengan valores nulos o iguales a cero en variables como flow_pkts_payload.tot, payload_bytes_per_second y flow_duration. Dado que la presencia de un flujo sin duración o sin bytes transmitidos no tiene sentido físico en el contexto de comunicación IoT, estos registros se eliminarán.

Asimismo, La matriz de correlaciones (eda_intensity_corr_heatmap.png) y el PCA mostraron alta redundancia entre flow_pkts_payload.tot, fwd_pkts_tot y bwd_pkts_tot. Por tanto, se conservará solo flow_pkts_payload.tot como variable representativa del volumen total. De manera análoga, se eliminará flow_pkts_per_sec ya que su correlación con payload_bytes_per_second supera 0.9, quedando esta última como indicador más estable del ritmo de transmisión.

Por otro lado, el análisis IQR identificó un conjunto pequeño pero significativo de flujos con valores fuera de rango (payload_bytes_per_second, flow_pkts_per_sec, flow_pkts_payload.tot), en estos casos, si los flujos pertenecen a tráfico normal, se considerarán ruido y se recortan en el percentil 99 de cada variable y se imputarán con la mediana. Si pertenece a tráfico de ataque, se mantendrán, con

el objetivo de mantener outliers relevantes para la clase maliciosa, pero evitar que los extremos benignos dañen el modelo.

Además, se normalizarán las variables flow_pkts_payload.tot y tasa payload_bytes_per_second, flow_pkts_per_sec. De esta forma se reduce la asimetría y mejora la comparabilidad entre magnitudes.

Se verificará la consistencia de los valores de flow_iat.avg y flow_duration. Casos donde flow_iat.avg * flow_pkts_payload.tot sea significativamente menor o mayor que flow_duration (>3 desviaciones estándar) se eliminarán, ya que probablemente corresponden a capturas corruptas o registros truncados.

Además, se añadirá una verificación de coherencia cruzada entre:

- payload_bytes_per_second ≈ flow_pkts_payload.tot / flow_duration
- flow_pkts_per_sec ≈ tot_pkts / flow_duration

Casos donde la diferencia supere el 15 % se marcarán como inconsistentes y se eliminarán.

Tras los pasos anteriores, se realizará una nueva evaluación de correlaciones y PCA sobre las variables restantes. Solo se conservarán aquellas que:

- Sumatoria de varianza significativa > 85%
- Coherencia física verificada.

El conjunto final esperado incluiría variables representativas de cada dimensión:

- Magnitud: flow_pkts_payload.tot
- Ritmo: payload_bytes_per_second
- Regularidad temporal: flow_iat.avg, active.avg, idle.avg
- Variabilidad: flow_iat.std, active.std, idle.std
- Relación direccional: down_up_ratio

3. *Tamaño de los Datos Transmitidos.*

El propósito de esta sección es analizar la estructura interna de los flujos IoT, centrando la atención en el tamaño, la variabilidad y la simetría de los datos transmitidos entre origen y destino.

A diferencia del bloque anterior,que examinó la intensidad y el ritmo del tráfico, este análisis busca comprender cómo se distribuye la carga útil (payload) dentro de cada flujo, tanto en dirección forward (envío) como backward (respuesta).

El dataset RT-IoT2022 ya contiene métricas preagregadas para cada flujo, tales como el tamaño promedio, mínimo, máximo, total y desviación estándar del payload en ambas direcciones. Estas variables resumen el comportamiento de los paquetes individuales y permiten evaluar cuán homogéneos o irregulares son los datos intercambiados en cada comunicación, sin necesidad de acceder al tráfico crudo.

Este análisis no pretende recalcular las estadísticas del tráfico, sino examinar cómo estas métricas se comportan a nivel de conjunto de flujos, buscando diferencias significativas entre el tráfico normal y el malicioso.

De este modo, se obtiene una visión más precisa sobre la naturaleza del contenido transmitido, complementando los hallazgos previos sobre ritmo e intensidad, y fortaleciendo la base para el modelado predictivo posterior.

Analizar estas métricas es fundamental para:

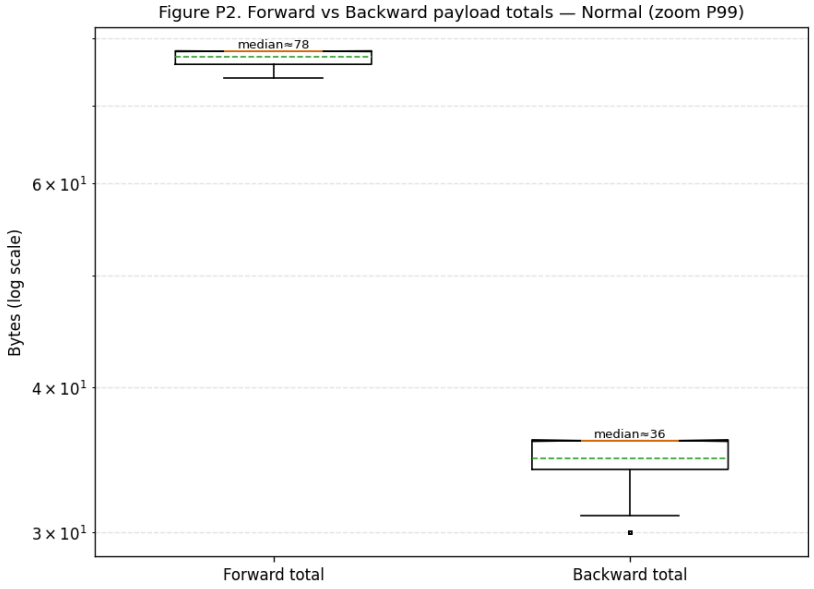
- Identificar patrones de asimetría entre envío y recepción, característicos de ataques unidireccionales.
- Detectar fragmentaciones o tamaños atípicos en los paquetes, indicadores posibles de manipulación o evasión.
- Comparar la homogeneidad del tamaño de los paquetes, donde flujos regulares suelen mostrar valores estables y ataques presentan tamaños constantes o picos extremos.
- Evaluar la coherencia interna entre los distintos niveles de agregación (fwd, bwd, flow), lo que ayuda a validar la integridad de los datos y detectar errores de captura.

Se propone analizar la siguiente familia de variables:

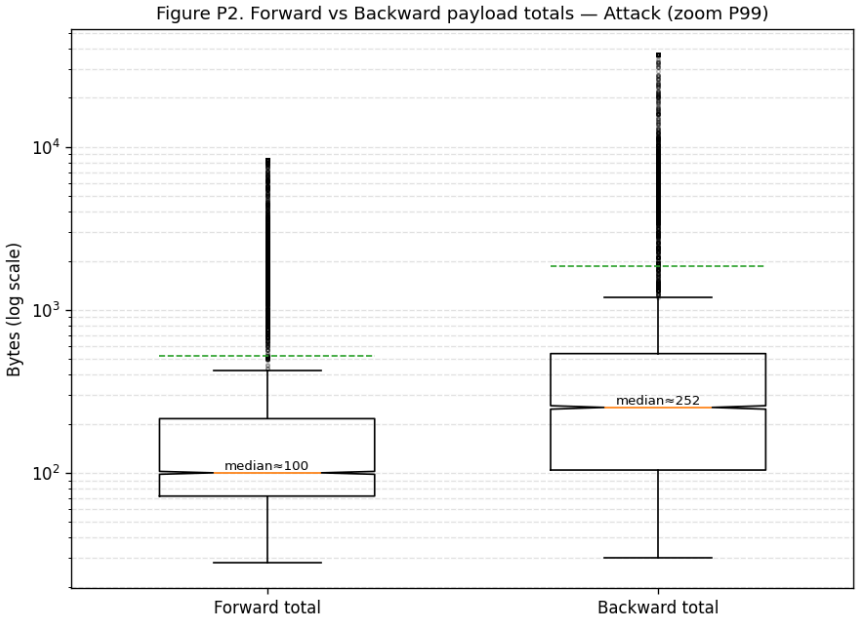
Grupo	Variables	Resultados Esperados
Payload Forward	fwd_pkts_payload.min, fwd_pkts_payload.max, fwd_pkts_payload.avg, fwd_pkts_payload.std, fwd_pkts_payload.tot	Ataques DoS, payloads grandes y poco variables, por otro lado en tráfico normal se esperan datos pequeños y estables.
Payload Backward	bwd_pkts_payload.min, bwd_pkts_payload.max, bwd_pkts_payload.avg, bwd_pkts_payload.std, bwd_pkts_payload.tot	Ataques de escaneo tienen respuestas cortas, y tráfico legítimo tiene simetría con forward.
Payload Total	flow_pkts_payload.min, flow_pkts_payload.max, flow_pkts_payload.avg, flow_pkts_payload.std, flow_pkts_payload.tot	Elevada desviación podría indicar mezcla de tamaños heterogéneos o fragmentación.
Cabeceras	fwd_header_size_tot, bwd_header_size_tot	Proporción alta de cabeceras indica tráfico de control (ej. SYN/ACK).
Simetría direccional	(fwd_pkts_payload.tot / (bwd_pkts_payload.tot + 1))	Ratios muy altos o bajos indican flujos anómalos o unidireccionales.

El análisis de payload confirma que los ataques se caracterizan por volúmenes y tamaños medios por paquete sensiblemente mayores y por una fuerte asimetría direccional entre envío y respuesta.

En primer lugar, los boxplots forward vs. backwards muestran diferencias por clase. En tráfico normal, la mediana de fwd_pkts_payload.tot se ubica alrededor de ≈78 bytes y la de bwd_pkts_payload.tot en ≈36 bytes, con rangos acotados.



En ataques, en cambio, las medianas suben a ≈100 bytes (fwd) y ≈252 bytes (bwd) y aparece una cola muy extensa que alcanza varios órdenes de magnitud. Esta asimetría, con respuesta (bwd) más voluminosa que la solicitud (fwd), es consistente con ataques de reflexión/amplificación (p. ej., DNS) y con exfiltraciones o respuestas abusivas.



La dispersión de los tamaños medios por paquete refuerza ese patrón. El scatter log–log entre fwd_pkts_payload.avg y bwd_pkts_payload.avg exhibe una nube compacta y pequeña para tráfico normal (valores ~5–10 bytes), mientras que los ataques se expanden desde decenas hasta miles de

bytes por paquete en ambas direcciones, con acumulaciones que sugieren modos operativos distintos (por ejemplo, cargas medias altas sólo en bwd, o elevadas en ambas direcciones).

La asimetría direccional queda cuantificada por el histograma de la razón de payload: $payload\_ratio\_log = log_{10}(\frac{fwd\_pkts\_payload.tot}{bwd\_pkts\_payload.tot + 1})$, es claro que si ≈ 0 el flujo es balanceado, si es > 0, domina forward, y si es < 0 domina backward.

Figure P1. Distribution of payload direction ratio (log10)

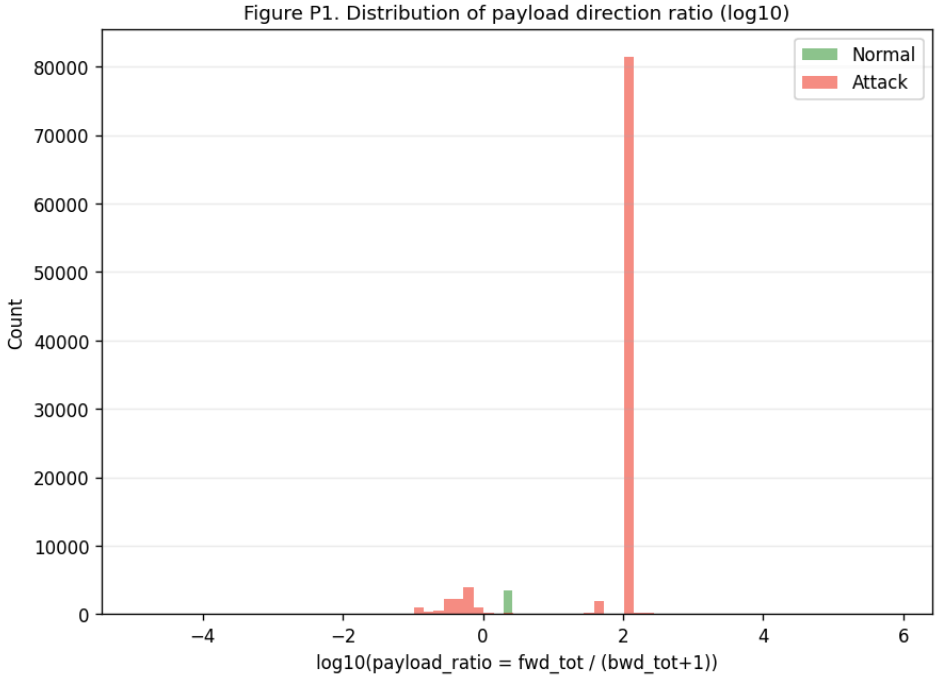
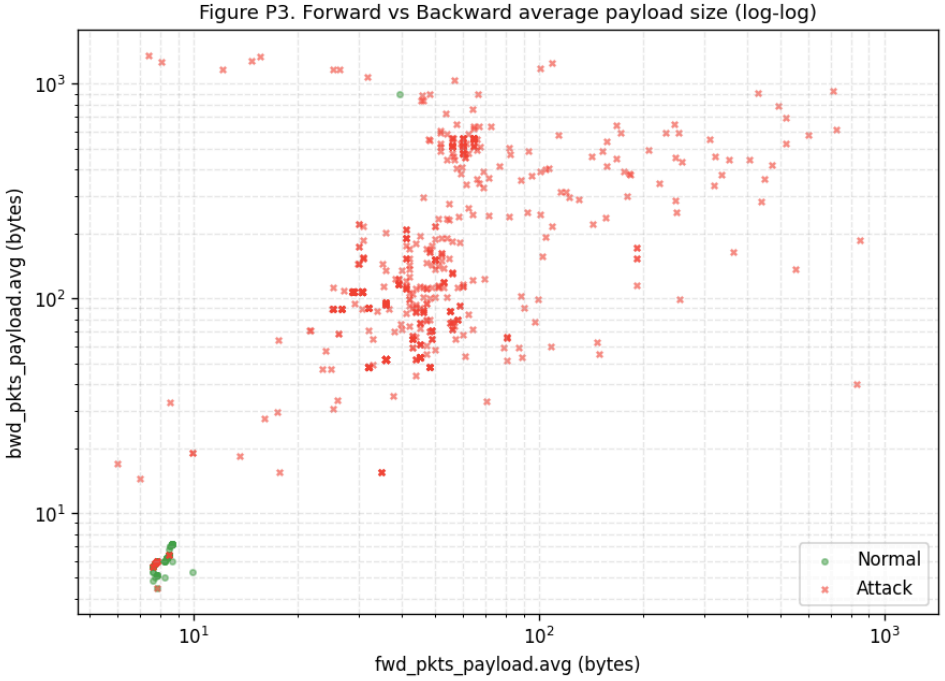


Figure P3. Forward vs Backward average payload size (log-log)



Desde la estructura estadística, la matriz de correlación revela correlaciones muy altas entre métricas de la misma familia: flow_pkts_payload.tot, fwd_pkts_payload.tot y bwd_pkts_payload.tot

(≥0.95–0.99), y relaciones intensas entre max/avg/std dentro de cada dirección. Esto indica redundancia y justifica una reducción de dimensionalidad.

El PCA aplicado al bloque de payload muestra: PC1 ≈ 46–47%, PC2 ≈ 14–15%, PC3 ≈ 13%; con 3–4 componentes se captura aproximadamente 75–85% de la varianza (eda_payload_pca_variance.xlsx). La proyección 2D refleja nubes de ataque mucho más extensas que las normales, lo que sugiere que las PCs capturan adecuadamente los modos volumétricos y de asimetría observados. Los loadings (eda_payload_pca_loadings.xlsx) confirman que PC1 está dominada por volúmenes totales (flow/fwd/bwd), PC2 por tamaños medios y/o asimetrías direccionales, y PC3 por variabilidad (std/max), proporcionando una base interpretable para la compresión del bloque.

Figure P4. Correlation matrix for payload metrics

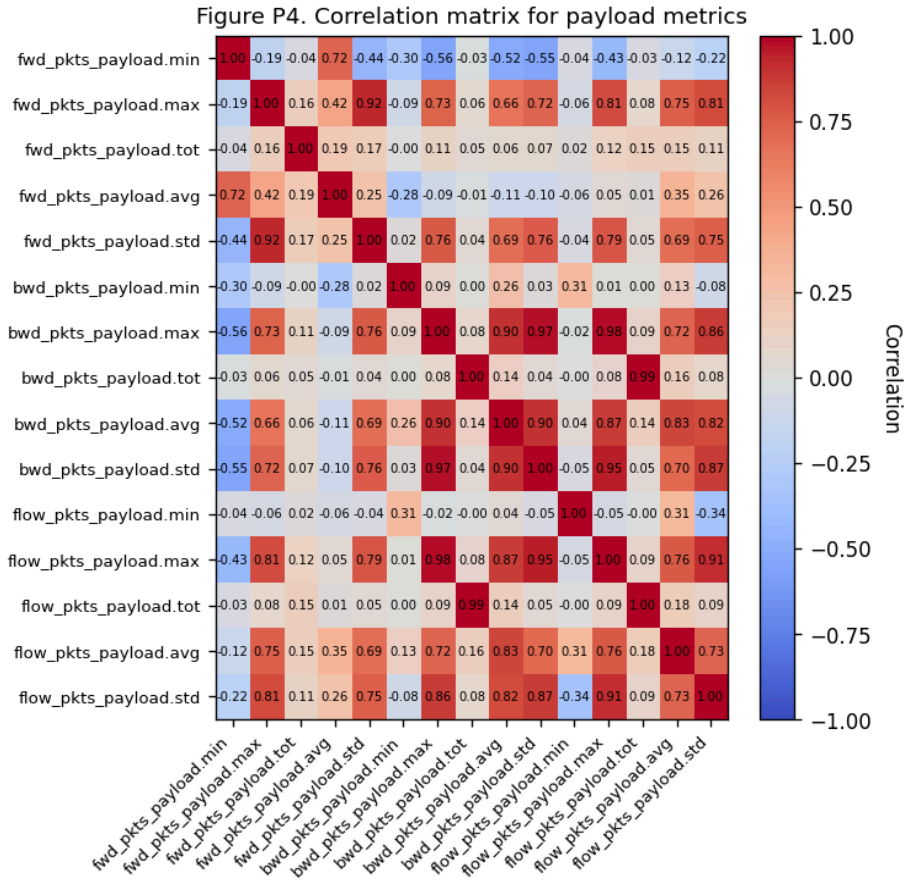


Figure P5. PCA Scree Plot – Payload metrics

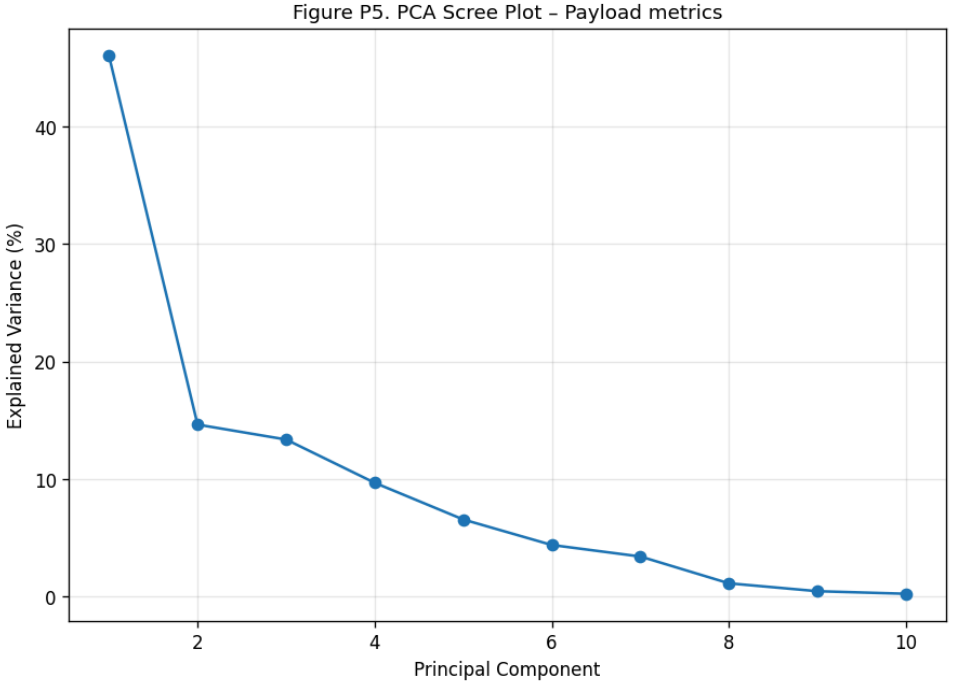
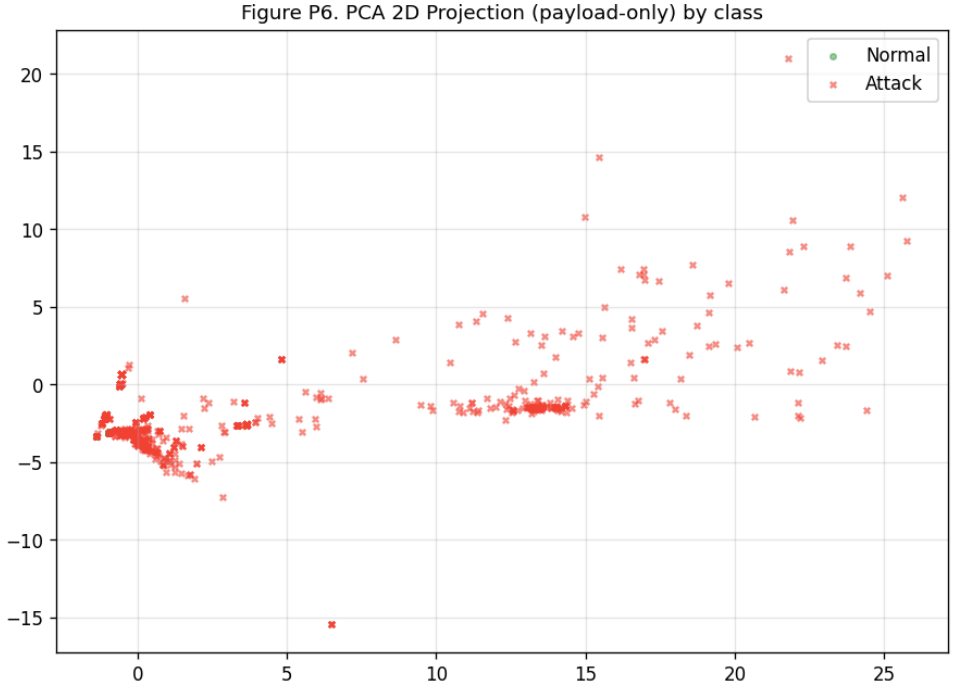


Figure P6. PCA 2D Projection (payload-only) by class



Finalmente, los outliers reportados (eda_payload_outliers.xlsx) se concentran mayoritariamente en la clase ataque, lo que concuerda con la naturaleza intrusiva (volúmenes/tamaños extremos). En la clase normal se observaron pocos extremos; cuando aparecen sin justificación protocolar, conviene tratarlos como ruido potencial de captura.

Para el proceso de limpieza, primero se unifican las columnas a tipo numérico, convirtiendo valores no finitos en faltantes. Dado que el PCA no admite NaN, se aplica una imputación por mediana calculada únicamente sobre el conjunto de entrenamiento, evitando sesgos y preservando la asimetría observada en las distribuciones. A continuación, se emplea transformación logarítmica sobre las

métricas con colas largas (…payload.tot, …payload.avg, …payload.std y payload_bytes_per_second) para estabilizar la varianza; y luego una estandarización z-score (centrado en 0 y desviación estándar 1), ajustada en entrenamiento y reutilizada en validación/prueba, de modo que todas las variables contribuyan en la misma escala al PCA.

Se incorpora un control de coherencia direccional verificando que el total del flujo sea consistente con la suma de las direcciones: flow_pkts_payload.tot ≈ fwd_pkts_payload.tot + bwd_pkts_payload.tot. Cuando el error relativo excede el 15% en filas etiquetadas como normales, se considera ruido de captura y se eliminan esos registros. Si la fila está etiquetada como ataque, se conserva, porque la inconsistencia puede ser parte del propio comportamiento intrusivo. Respecto de los valores extremos, no se eliminan outliers de la clase ataque; en la clase normal, solo si existen valores extremos se sustituyen por el percentil 99 en lugar de eliminarlos, y solo en entrenamiento, para proteger la estabilidad numérica sin alterar la distribución de validación/prueba.

La reducción de dimensionalidad se realiza exclusivamente con PCA sobre todas las métricas de payload (min/max/tot/avg/std en ambas direcciones y a nivel de flujo). Se retienen 3–4 componentes, o las necesarias hasta alcanzar ≥80% de varianza explicada, y se sustituyen las variables originales del bloque por PC1..PCk, documentando la varianza explicada y las cargas (loadings) para mantener trazabilidad e interpretabilidad del subespacio.

Por último, se conserva fuera del PCA una variable de asimetría direccional (payload_ratio en su versión logarítmica), dado su alto poder discriminante evidenciado en los histogramas y boxplots. Con estas decisiones, el bloque queda compacto, coherente y estable, sin multicolinealidad y con la señal intrusiva (volumen elevado, tamaños medios grandes y desbalance fwd/bwd) preservada para el modelado.

4. *Ritmo del tráfico*

El objetivo de esta sección es analizar la velocidad, estabilidad y comportamiento temporal de los flujos de comunicación IoT, observando cómo se distribuyen los intervalos y tasas de transmisión a lo largo del tiempo.

Mientras que los análisis anteriores se centraron en cuánto se transmite (volumen, tamaño y duración de los flujos), esta parte se focaliza en cómo se transmite la información es decir, en la dinámica temporal y la regularidad del tráfico.

Este estudio permite identificar patrones característicos de dispositivos IoT legítimos, que tienden a enviar datos de forma periódica y estable, frente a ataques que suelen mostrar ráfagas de tráfico, variaciones abruptas o intervalos anómalos entre paquetes.

La comprensión de estas métricas es esencial para detectar irregularidades sutiles que no dependen del tamaño del flujo, sino del ritmo temporal de las comunicaciones.

A pesar de que algunas de estas variables podrían agruparse junto con las de “Intensidad y Tamaño de Flujo”, se decidió analizarlas de forma separada para mantener la claridad conceptual y analítica del análisis exploratorio.

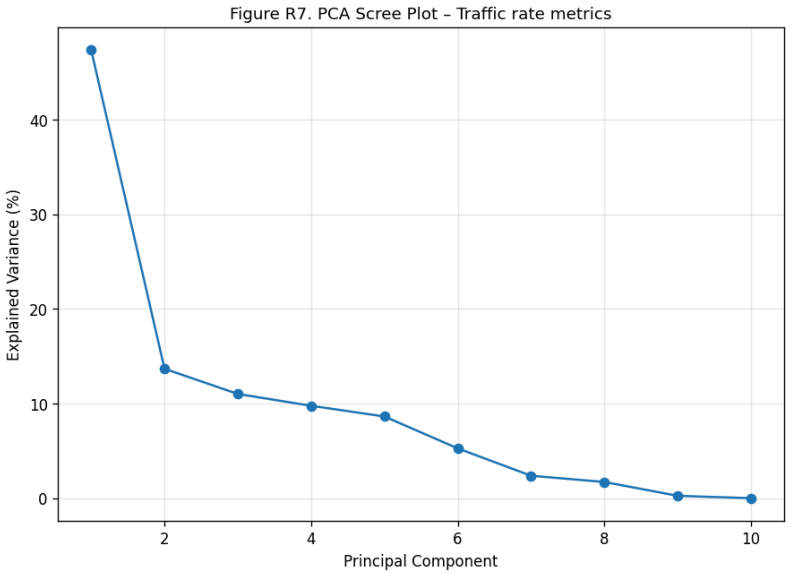
Las variables de intensidad reflejan magnitudes acumuladas (por ejemplo, cantidad total de bytes), mientras que las de ritmo describen la evolución temporal de esas magnitudes.

Esta separación permite aplicar procesos de limpieza y normalización distintos para cada grupo y facilita la interpretación de los resultados.

Para ello, se consideraron las siguientes variables:

Categoría	Variables	Descripción / Interpretación
Tasa de transmisión	flow_bytes_s, flow_pkts_s	Los ataques DoS suelen mostrar valores muy altos
Intervalos entre paquetes	flow_iat.min, flow_iat.max, flow_iat.tot	Permiten observar la variabilidad dentro del flujo
Ritmo direccional	fwd_iat.avg, bwd_iat.avg	Desbalances pueden indicar congestión o ataques dirigidos.
Periodos activos	active.min, active.max	En tráfico maligno, se presentan patrones regulares
Períodos inactivos	idle.min, idle.max	Tráfico IoT legítimo muestra patrones regulares
Ritmo de carga útil	payload_bytes_per_second	Sirve como tasa de referencia complementaria a las métricas principales

En el análisis del Ritmo de Tráfico se detectó que la variabilidad temporal del comportamiento IoT puede resumirse en pocas dimensiones. El scree plot mostró que la primera componente principal explica aproximadamente el 47% de la varianza, la segunda el 14% y la tercera el 11%; en conjunto, tres a cuatro componentes capturan en torno al 70–80% de la variabilidad total. Este resultado sugiere que es viable reducir la dimensionalidad o agrupar métricas temporales sin perder información relevante para la caracterización del tráfico.



Asimismo, la matriz de correlación evidenció redundancias marcadas entre las métricas de intervalos y de pausa: flow_iat.min/max/tot y idle.min/max presentaron correlaciones muy altas (≈0.95–0.99).

En términos prácticos, esto respalda la selección de uno o dos representantes de cada familia, por ejemplo, flow_iat.max e idle.max, o alternativamente, la aplicación de un PCA local sobre este subgrupo para sintetizar la información.

Figure R4. Correlation matrix – Traffic rate & temporal metrics

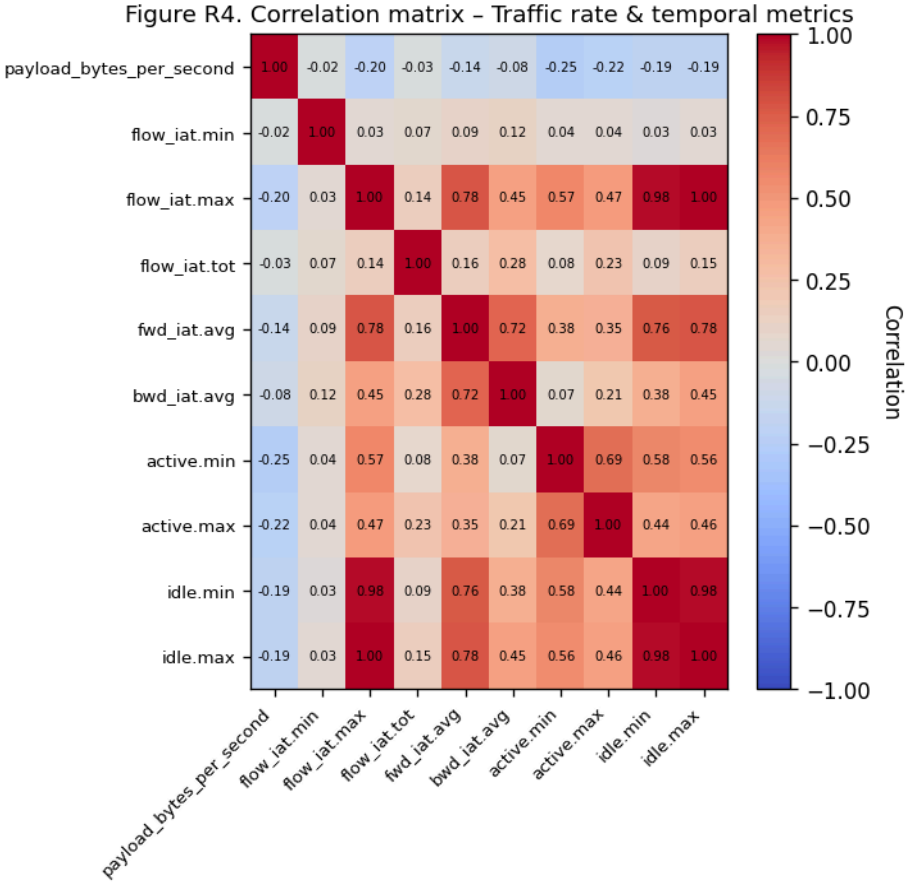
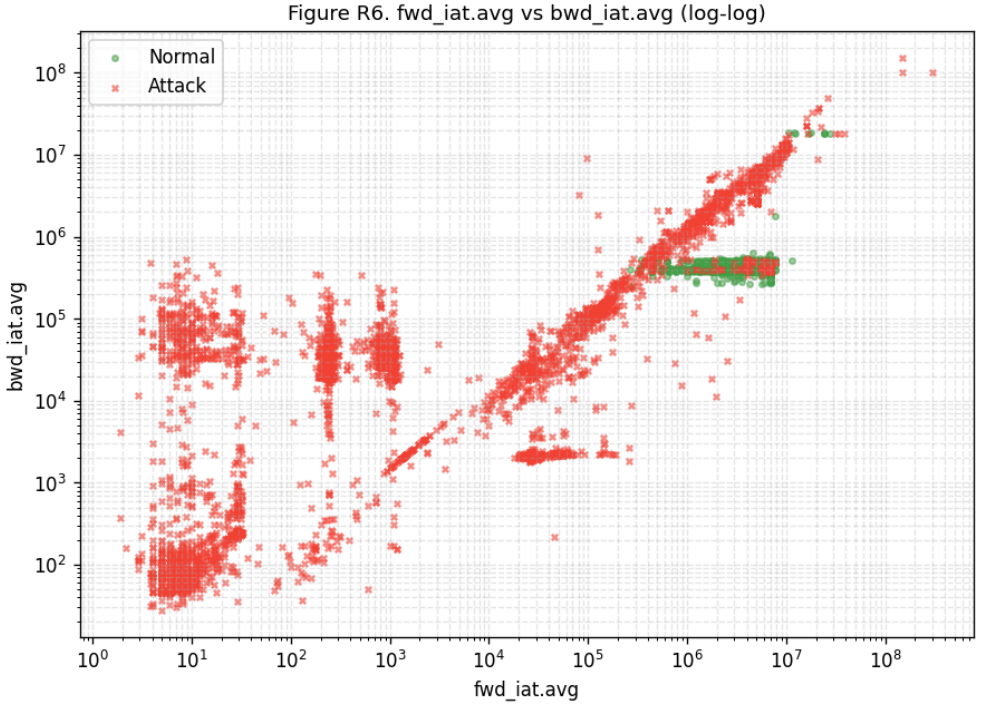
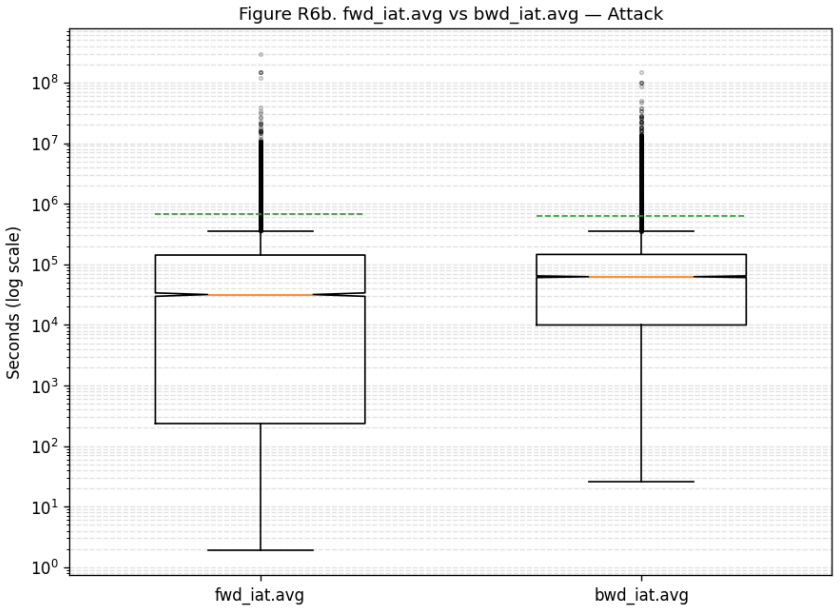


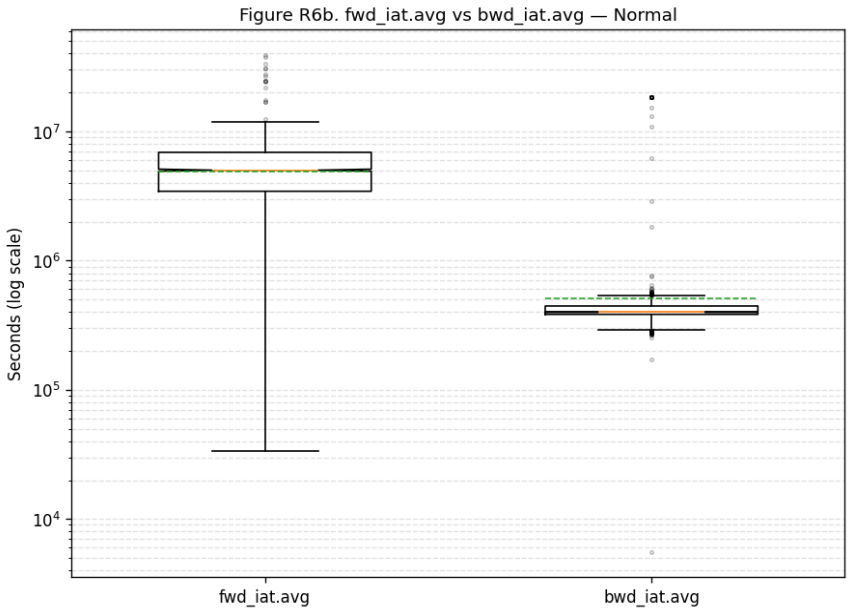
Figure R6. fwd_iat.avg vs bwd_iat.avg (log-log)



En cuanto a la dinámica direccional, el diagrama de dispersión log–log entre fwd_iat.avg y bwd_iat.avg reveló patrones claramente diferenciados por clase. En el tráfico normal se observó una nube compacta en torno a la diagonal, que indica un ritmo equilibrado entre envío y respuesta. En cambio, en el tráfico malicioso se visualiza múltiples regímenes con gran dispersión, presencia de intervalos muy pequeños (ráfagas) y asimetrías pronunciadas entre direcciones.

Gracias al boxplot direccional se obtuvo que, en condiciones normales, bwd_iat.avg presenta rangos más acotados y una mediana estable, mientras que en escenarios de ataque las medianas tienden a ser menores (intervalos más cortos) y las distribuciones exhiben colas muy largas, reflejando picos y pausas irregulares. Esta evidencia es coherente con una variabilidad relativa mayor en ataques, lo que también se verifica en los coeficientes de variación reportados en eda_rate_cv.xlsx, y refuerza el valor de la variabilidad como señal discriminante.





Antes de la reducción de dimensionalidad, se aplicó un preprocesamiento mínimo orientado a estabilizar escalas y garantizar la validez numérica del PCA, sin introducir combinaciones ni eliminar información de clase.

En primer lugar, todas las variables del bloque de ritmo (tasas, intervalos y períodos activo/inactivo) se forzaron a tipo numérico, reemplazando valores no finitos por faltantes. Dado que el PCA no admite valores ausentes, se implementó una imputación conservadora por mediana (estimada exclusivamente sobre el conjunto de entrenamiento), opción robusta frente a distribuciones sesgadas con colas largas, como evidenciaron los histogramas y boxplots.

A continuación, se aplicó una transformación logarítmica a las variables no negativas con fuerte asimetría (por ejemplo, flow_byts_s, flow_pkts_s, métricas de IAT y períodos), con el objetivo de reducir el peso desproporcionado de valores extremos y estabilizar la varianza.

Finalmente, se realizó una estandarización z-score (centrado en cero y desviación estándar) aprendida sobre el entrenamiento y reutilizada en validación/test, de modo que el PCA opere sobre variables comparables en escala y sensibilidad, y que las componentes reflejan estructura informativa y no magnitudes arbitrarias.

La reducción de variables se llevó a cabo exclusivamente mediante Análisis de Componentes Principales (PCA) sobre el conjunto completo de métricas de ritmo, ya transformadas y estandarizadas. El PCA se ajustó en el conjunto de entrenamiento y se seleccionó el número de componentes según dos criterios convergentes: alcanzar al menos el 80% de varianza explicada acumulada, y lo observado en el scree plot (en nuestros datos, 3–4 componentes capturaron ~70–80%).

Tras estos pasos, el bloque de Ritmo de Tráfico queda representado por 3–4 componentes principales que concentran la mayor parte de la variabilidad temporal observada, reemplazando un conjunto altamente redundante de métricas de intervalos y períodos. El dataset resultante es más compacto, mejor condicionado (escalas estables y varianza homogénea) y libre de multicolinealidad, sin sacrificar la señal discriminante identificada en el EDA (asimetrías direccionales, variabilidad de ritmos y tasas efectivas). Esta representación reduce el riesgo de sobreajuste, simplifica la selección de características y sienta una base más robusta para los modelos posteriores.

5. *Control de conexión TCP*

El propósito de este bloque es caracterizar el plano de control de TCP en los flujos IoT: cómo se inician, mantienen y cierran las conexiones, qué banderas intervienen y cómo evoluciona el tamaño de ventana. Este enfoque permite detectar handshakes incompletos (p. ej., SYN flood), terminaciones anómalas (RST storms), uso inusual de PSH/URG, presencia atípica de ECN (ECE/CWR) y colapsos de ventana, señales típicas de ataques o de errores de captura/etiquetado. El análisis se restringe a proto = tcp y se compara normal vs. ataque, priorizando métricas de control (no de volumen), para luego reducir redundancia con PCA dentro de este bloque.

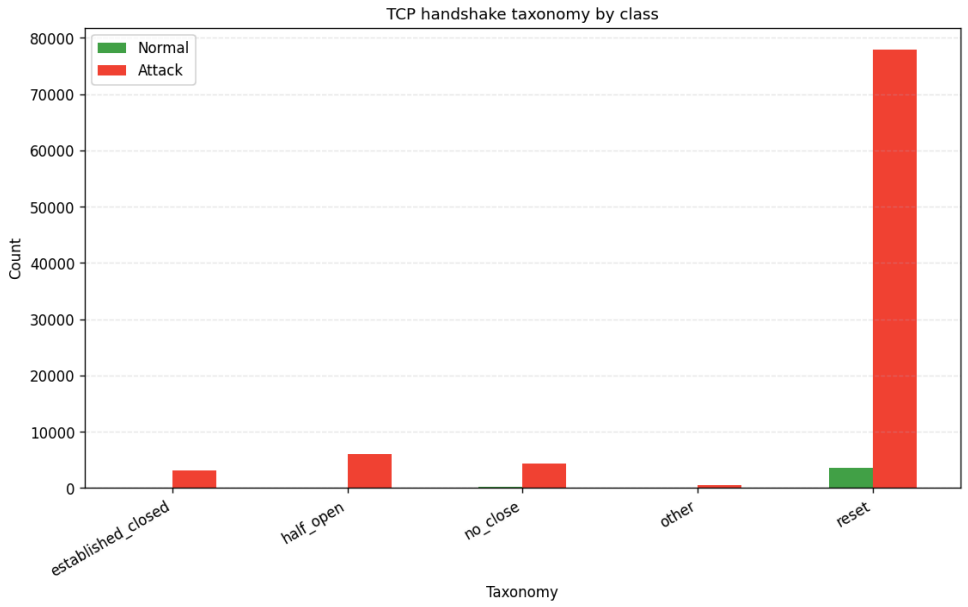
Para ello, se tomó las siguientes variables:

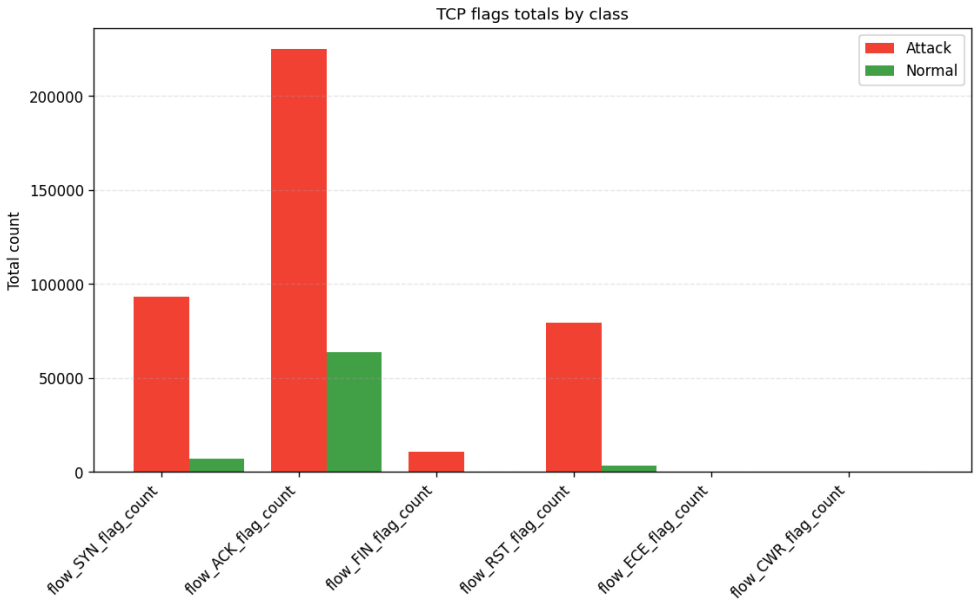
Categoría	Variables
Banderas	flow_SYN_flag_count, flow_ACK_flag_count, flow_FIN_flag_count,

	flow_RST_flag_count, flow_ECE_flag_count, flow_CWR_flag_count
Banderas direccionales	fwd_PSH_flag_count / bwd_PSH_flag_count, fwd_URG_flag_count / bwd_URG_flag_count,
Ventana TCP	fwd_init_window_size, bwd_init_window_size, fwd_last_window_size

Se detectó que la taxonomía de handshake está fuertemente sesgada hacia situaciones de reset en el tráfico malicioso. El gráfico de barras por clase muestra que la categoría reset domina abrumadoramente los flujos marcados como ataque, mientras que en tráfico normal casi no aparece.

En línea con esto, el gráfico de totales de flags por clase confirma órdenes de magnitud más alta de ACK y, especialmente, RST en ataques, con SYN también elevado. Este panorama sugiere abusos de SYN-flood/Reset-abuse y cierres abruptos como mecanismos característicos del tráfico ofensivo.

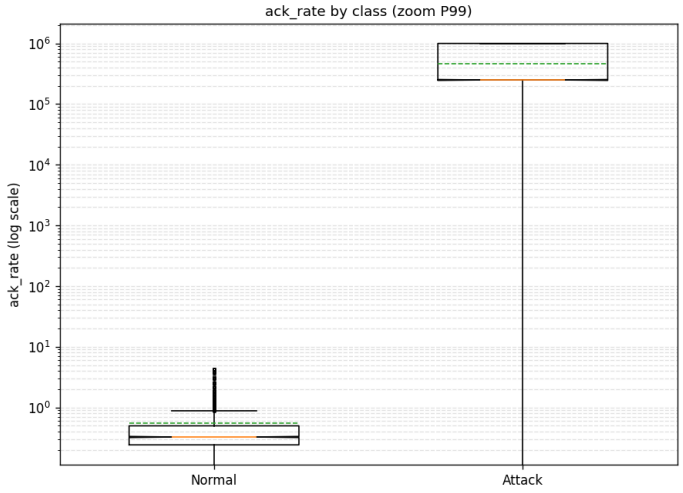


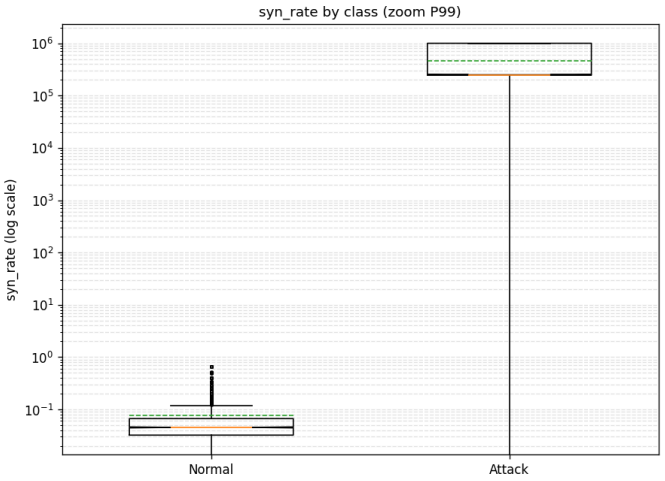


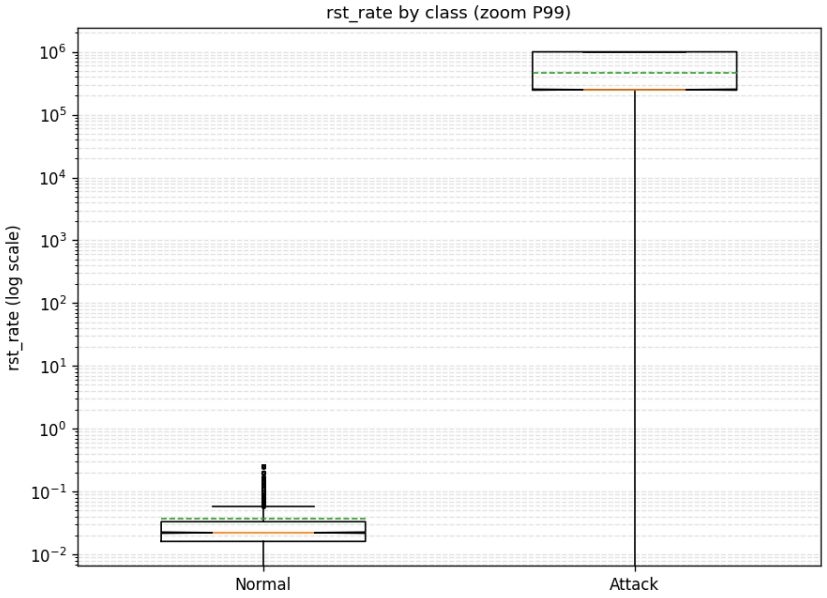
Gracias a los boxplots a escala logarítmica, se obtuvo que syn_rate, ack_rate y rst_rate son varias potencias de 10 mayores en ataques que en tráfico normal. Se detectó, en particular, que:

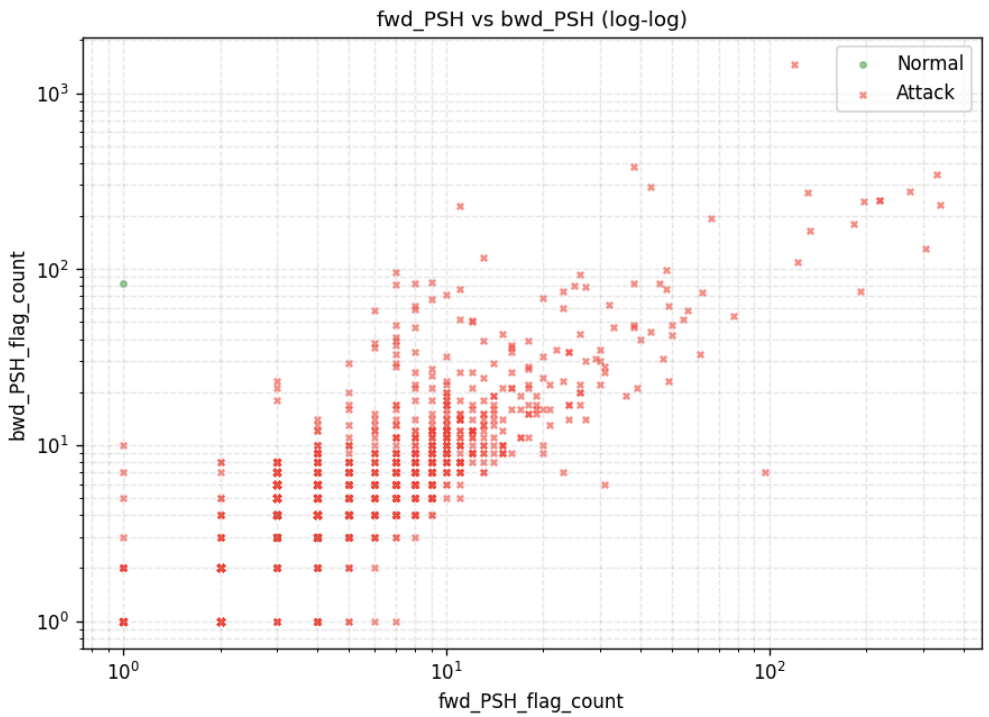
- syn_rate y rst_rate exhiben colas extremadamente largas en ataques, coherentes con ráfagas y cierres inoportunos propios de DoS o exploraciones agresivas.
- En normal, las medianas se mantienen cerca de valores con dispersión acotada, reflejando ritmos de señalización regulares.

Estas separaciones sugieren que las tasas normalizadas por tiempo son fuertemente discriminantes para el modelado.

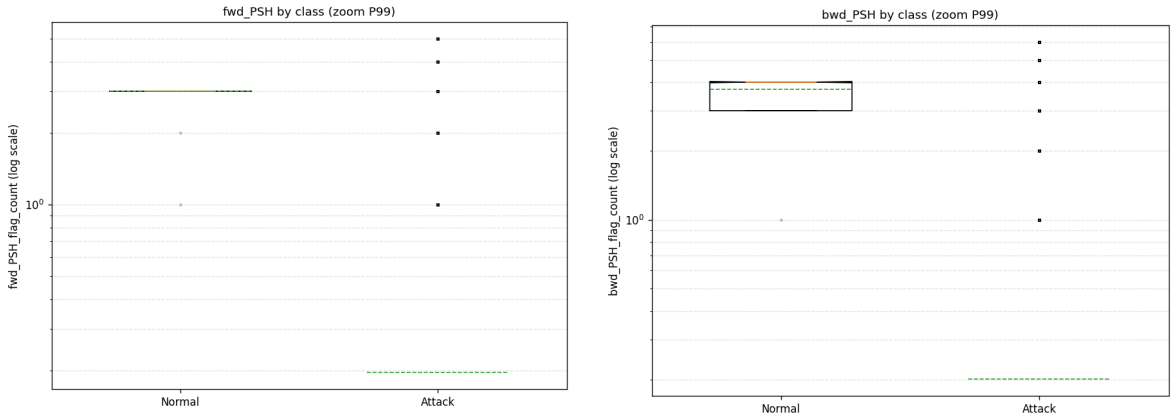








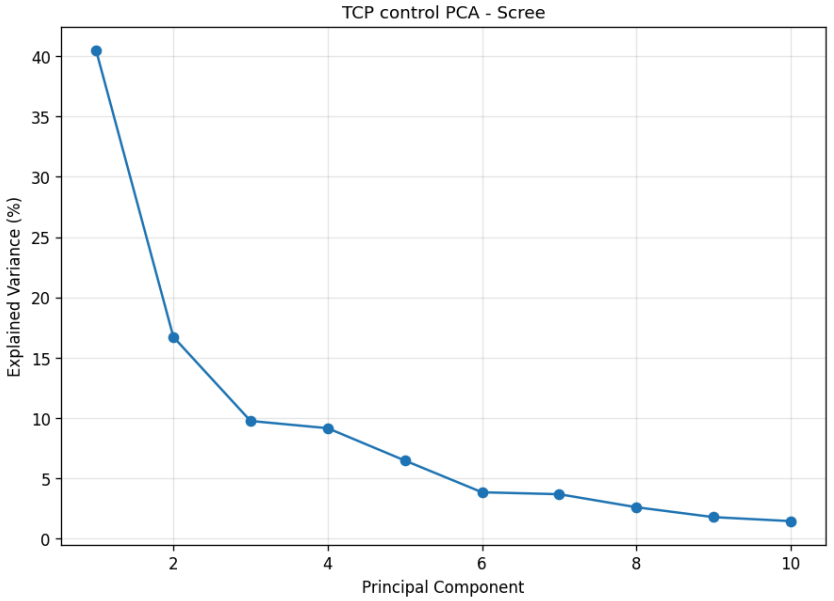
El scatter fwd_PSH vs bwd_PSH (log-log) evidencia que los ataques presentan amplia dispersión y múltiples regímenes de intensidad, con correlación positiva entre direcciones y valores muy por encima del rango normal. Complementariamente, los boxplots direccionales muestran que en tráfico normal PSH es casi nulo (próximo a 0), mientras que en ataques aparecen numerosos outliers y medianas más elevadas Gracias a este análisis direccional se obtuvo que la asimetría de PSH y su presencia sostenida son marcadores útiles de comportamiento anómalo orientado a empuje de datos.

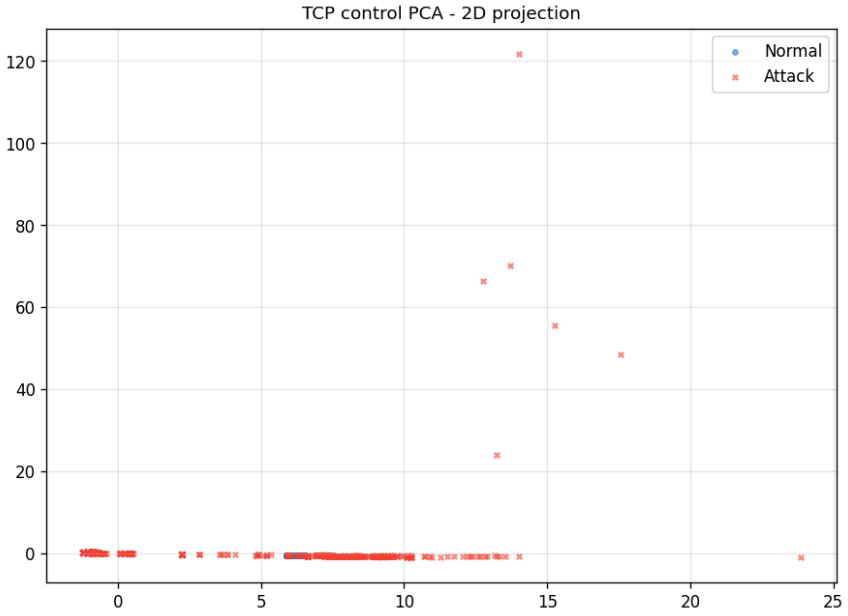


Se observó que URG permanece en cero prácticamente para ambas clases, por lo que aporta poca señal discriminante. Algo similar ocurre con ECN: el resumen (tcp_ecn_summary.xlsx) indica uso muy marginal de ECE/CWR, sin patrones diferenciales claros. Estas variables pueden conservarse para trazabilidad, pero su valor predictivo directo parece limitado.

El scree plot de PCA sugiere que PC1 ≈ 40%, PC2 ≈ 17% y PC3 ≈ 10%; con 3–4 componentes se captura alrededor de 65–75% de la varianza total. La proyección 2D separa visiblemente a la clase ataque en regiones de mayor actividad de flags y tasas, aunque con algo de solapamiento. Las cargas (tcp_pca_loadings.xlsx) muestran que syn_rate/rst_rate, contadores de RST/ACK y métricas de

ventana son los principales contribuidores de varianza, mientras que URG/ECN aportan marginalmente, confirmando la posibilidad de reducir dimensionalidad sin perder capacidad explicativa.





En conjunto, los resultados apoyan que syn_rate, rst_rate, ack_rate, los contadores de RST/ACK, y la evolución de ventana (init vs last) capturan la esencia del abuso de TCP en ataques. PSH direccional añade una señal complementaria de empuje anómalo de datos. Por el contrario, URG y ECN presentan baja utilidad práctica en este dataset. Finalmente, el PCA habilita compactar el grupo TCP a unas pocas combinaciones lineales manteniendo la mayor parte de la información, lo que puede simplificar el modelo sin degradar su poder discriminante.

A fin de depurar el bloque Control de conexión TCP, se implementará un protocolo de limpieza. En una primera instancia se validará la coherencia semántica de las variables: los contadores de flags (flow_SYN/ACK/FIN/RST/PSH/URG_flag_count) y los tamaños de ventana (fwd_init_window_size,

fwd_last_window_size) deberán ser enteros no negativos. Cualquier valor negativo o no numérico se tratará como dato inválido.

Sobre esos casos se actuará con reglas determinísticas: si el resto de los metadatos del flujo es consistente (por ejemplo, protocolo TCP declarado, duración y payload plausibles) y el problema se limita a redondeos o conversiones, se corregirá al entero válido más cercano. Si, en cambio, el registro entra en contradicción con la semántica de TCP (p. ej., proto ≠ tcp pero aparecen flags TCP, o combinaciones RST/ACK imposibles para un mismo tramo temporal), el flujo se excluirá del conjunto de entrenamiento.

Dado el carácter fuertemente asimétrico de las distribuciones de tasas y ventanas, se aplicará una normalización centrando en la mediana y escalando por el IQR, con el objetivo de estabilizar la variabilidad. Complementariamente, se realizará un recorte por percentiles diferenciado por clase (Normal/Ataque): los valores por debajo del P1 y por encima del P99 se acotará a dichos umbrales. Esta estrategia reduce el impacto de colas patológicas atribuibles a errores de captura.

Para evitar redundancias y condensar la información relevante, se ejecutará un PCA específico del subespacio TCP una vez aplicadas las transformaciones anteriores. Se conservarán las tres o cuatro primeras componentes, que acumulan la mayor parte de la varianza observada.

Este conjunto de acciones deja un bloque TCP coherente con la semántica del protocolo, numéricamente estable, sin variables redundantes ni casi constantes, y listo para integrarse al pipeline de modelado.

6. *Eficiencia y Balance entre tráfico de subida y bajada*

El propósito de esta sección es cuantificar qué tan eficiente es cada flujo (proporción de carga útil frente a sobrecarga de encabezados) y qué tan balanceado está entre uplink (forward) y downlink (backward). La hipótesis operativa es que el tráfico IoT legítimo presenta balances cercanos a cero (sube y baja en proporciones estables según el servicio) y eficiencias moderadas, mientras que distintos ataques introducen asimetrías marcadas (p. ej., muchos bytes hacia un único sentido) o ineficiencias (exceso de control/overhead). Estas señales complementan las vistas de intensidad y ritmo y ayudan a separar comportamientos benignos de maliciosos.

Se utilizarán las siguientes variables, algunas ya analizadas, sin embargo, es enriquecedor analizar estas variables desde la perspectiva direccional y eficiencia.

Variable	Significado
fwd_pkts_tot, bwd_pkts_tot	Cantidad total de paquetes por dirección.
fwd_pkts_payload.tot, bwd_pkts_payload.tot, flow_pkts_payload.tot	Carga útil (payload) por dirección y total del flujo.
fwd_pkts_per_sec, bwd_pkts_per_sec, flow_pkts_per_sec	Ritmo de paquetes por dirección y total.
fwd_header_size_tot, bwd_header_size_tot	Tamaño total de encabezados por dirección.

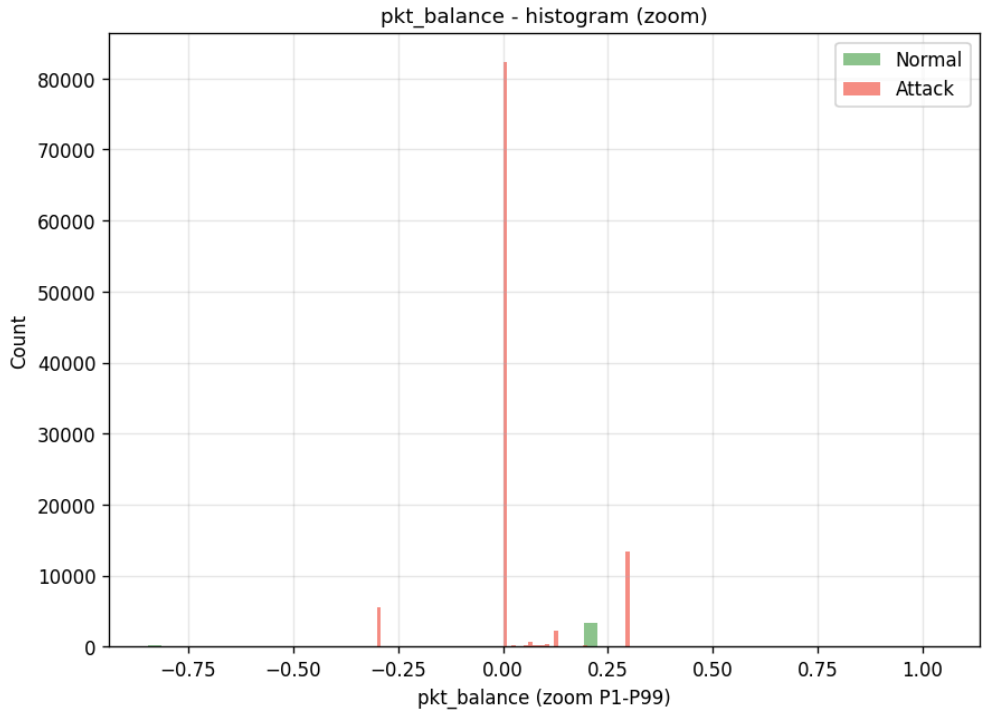
Además de las métricas originales, se definieron indicadores que normalizan y sintetizan el comportamiento del flujo, transformando volúmenes y conteos absolutos en relaciones interpretables.

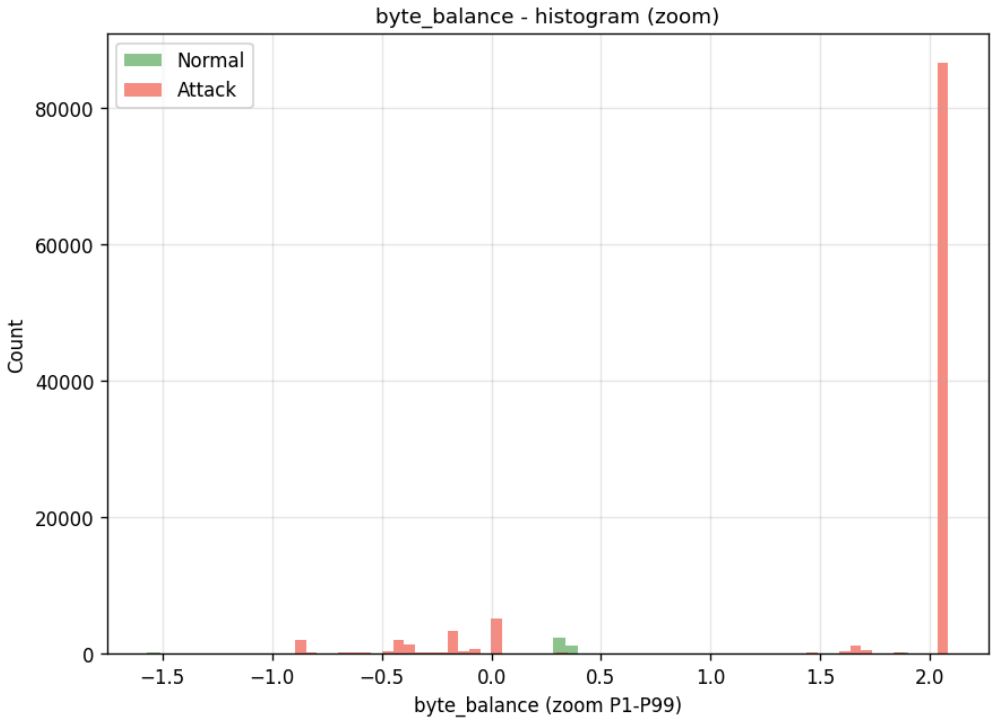
En particular, los balances (pkt_balance, byte_balance, rate_balance) capturan la diferencia direccional entre subida y bajada de forma comparable entre servicios y dispositivos, mientras que las eficiencias (fwd_eff, bwd_eff, flow_eff) miden la proporción de carga útil frente al overhead. Estas variables son menos sensibles a la escala y a picos extremos, refuerzan la comparabilidad entre flujos heterogéneos y reducen colinealidades presentes en las medidas crudas, aportando señales más nítidas para diferenciar tráfico legítimo de malicioso y preparando el terreno para técnicas de reducción (p. ej., PCA) y modelado.

Variable derivada	Fórmula (robusta)	Qué captura / interpretación
pkt_balance	log10((fwd_pkts_tot + 1) / (bwd_pkts_tot + 1))	Asimetría en número de paquetes.
byte_balance	log10((fwd_pkts_payload.tot + 1) / (bwd_pkts_payload.tot + 1))	Asimetría en volumen de payload.
rate_balance	log10((fwd_pkts_per_sec + 1e-9) / (bwd_pkts_per_sec + 1e-9))	Asimetría en ritmo (paquetes por segundo)
abs_pkt_balance, abs_byte_balance, abs_rate_balance	abs(...) de las anteriores	Magnitud de la asimetría (sin signo), cuán desbalanceado está el flujo
fwd_eff	fwd_pkts_payload.tot / (fwd_pkts_payload.tot + fwd_header_size_tot + 1e-9)	Eficiencia de uplink: fracción de bytes que son payload (no encabezados).
bwd_eff	bwd_pkts_payload.tot / (bwd_pkts_payload.tot + bwd_header_size_tot + 1e-9)	Eficiencia de downlink.
flow_eff	flow_pkts_payload.tot / (flow_pkts_payload.tot + fwd_header_size_tot + bwd_header_size_tot + 1e-9)	Eficiencia global del flujo.

Se detectó que las métricas de balance (diferencias/ratios entre direcciones) y las métricas de eficiencia (proporción de payload frente a overhead) captaron patrones muy distintos del tráfico. En particular, los ataques muestran asimetrías marcadas entre “forward” y “backward” y, al mismo tiempo, eficiencias mucho más altas en la dirección ofensiva. Esta combinación ofrece un poder discriminante claro frente al tráfico normal.

Gracias a los histogramas de byte_balance se observó una concentración masiva de flujos de ataque con valores positivos elevados (pico en torno a ~2 en la escala usada), lo que implica que el volumen de bytes en la dirección “forward” supera ampliamente al “backward”. En contraste, el tráfico normal se concentró cerca de valores moderados y próximos a 0, compatibles con intercambios más simétricos.

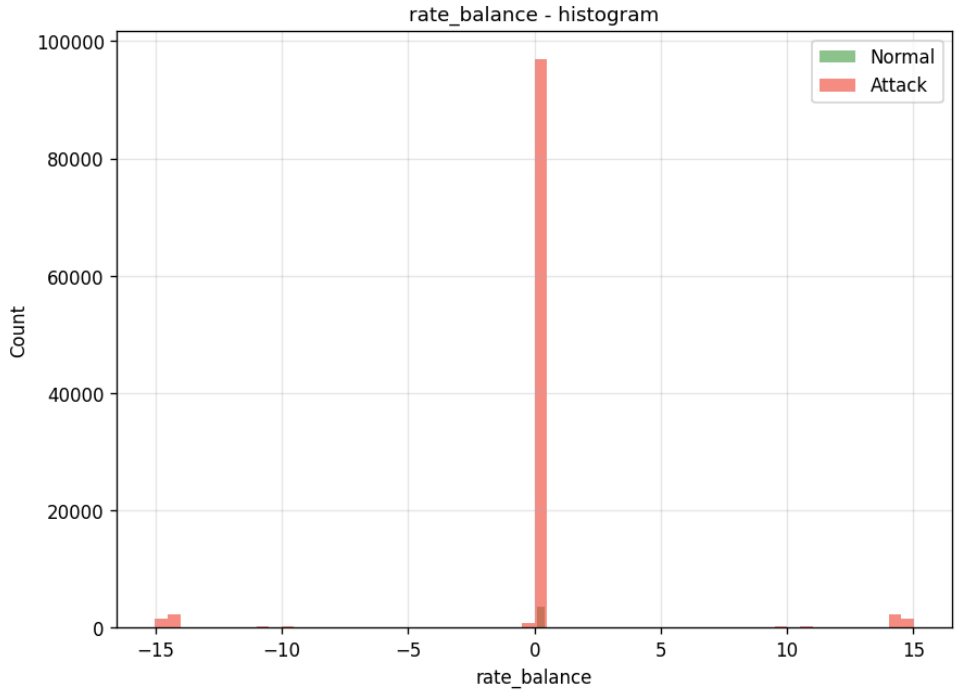




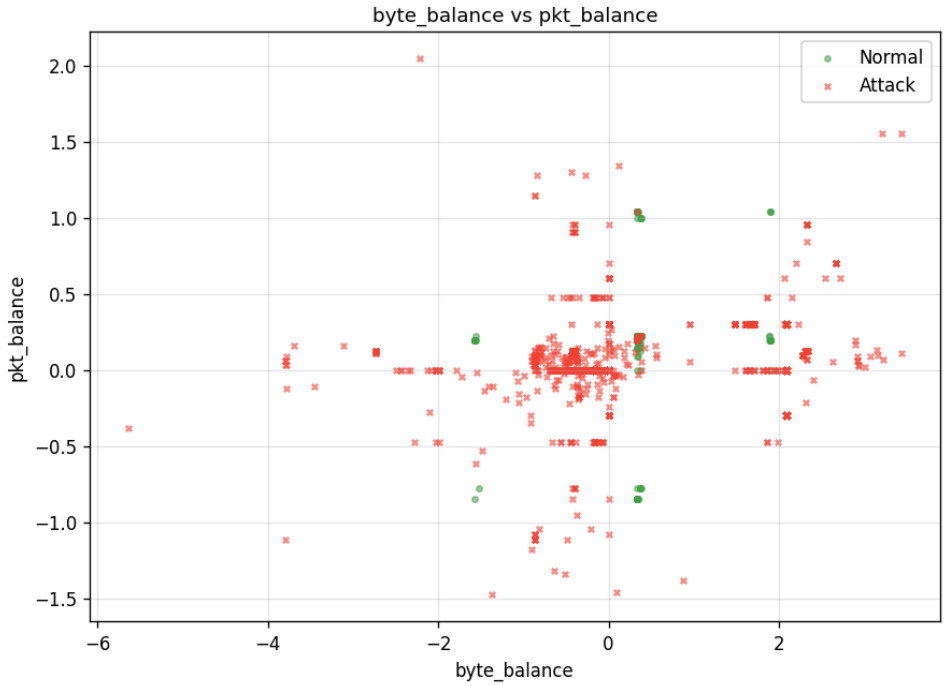
El comportamiento por paquetes replica la idea, aunque con menos contraste. En los histogramas de pkt_balance se detectó que los ataques se amontonan muy cerca de 0 (muchos intercambios con un número parecido de paquetes a ambas direcciones o ráfagas cortas que no rompen la simetría en "conteo"), mientras que el normal tiende a ligeras asimetrías positivas.

En rate_balance se observó una nube central alrededor de 0 con colas muy largas en ataques (valores extremos positivos y negativos, ±15 aprox.). Esto indica que, aun cuando el conteo de paquetes pueda

parecer equilibrado, la tasa efectiva (bytes/seg o pkts/seg por dirección) se desbalancea de forma pronunciada en escenarios maliciosos, coherente con picos de envío unidireccionales.



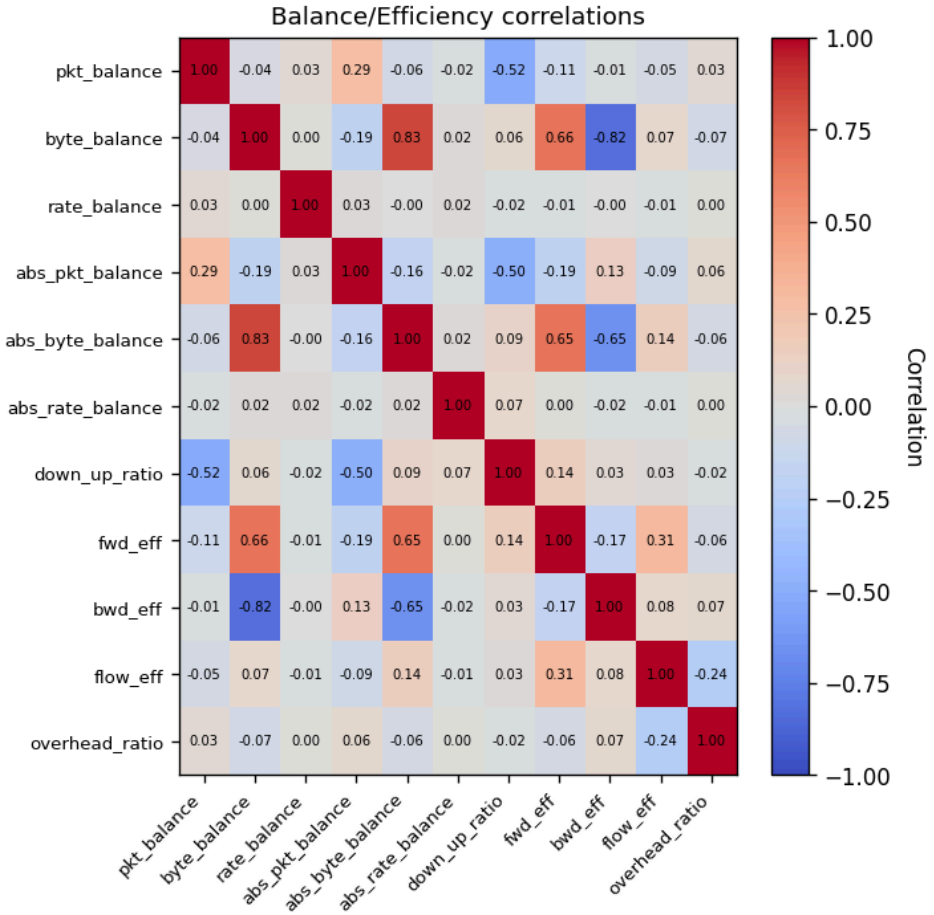
El scatterplot byte vs pkt balance refuerza la lectura anterior: se detectó una dispersión amplia en ataques cubriendo los cuatro cuadrantes (a veces mucho byte pero pocos paquetes, o viceversa), mientras que los normales se agrupan en una zona compacta con balances bajos y consistentes. Operativamente, conviene retener ambos tipos de balance (por bytes y por paquetes), ya que aportan señales complementarias.



Los boxplots por clase mostraron diferencias muy claras. Gracias al análisis de mediana en fwd_eff y flow_eff se detectó que la mediana de eficiencia en ataques es sustancialmente mayor (≈0.75 en el flujo total y ≈0.85 en "forward"), frente a valores bajos y estables en tráfico normal (≈0.18–0.22).

Esto sugiere que, en ataques, una fracción mucho mayor del tráfico son datos útiles (payload) y no encabezados/overhead, consistente con floods de alto caudal o transferencias masivas.

A la inversa, bwd_eff apareció deprimida casi a cero en la mayoría de ataques (con algunos outliers altos), mientras que en normal se mantuvo baja pero estable (~0.15). Se interpreta como poca "carga útil" en respuestas (el extremo atacado responde poco o corta conexiones), lo que incrementa la asimetría direccional.



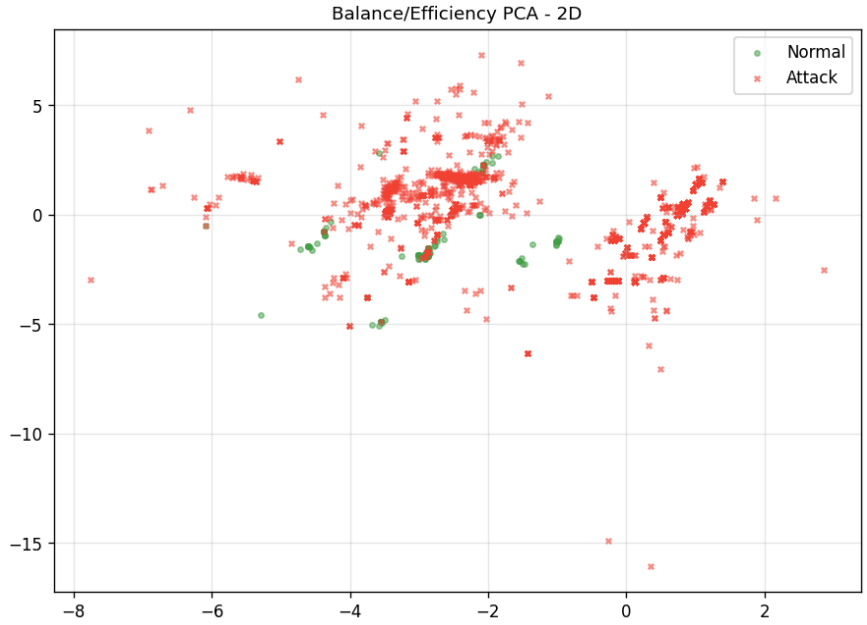
La matriz de correlación (balance_correlaciones.xlsx) confirmó estas intuiciones:

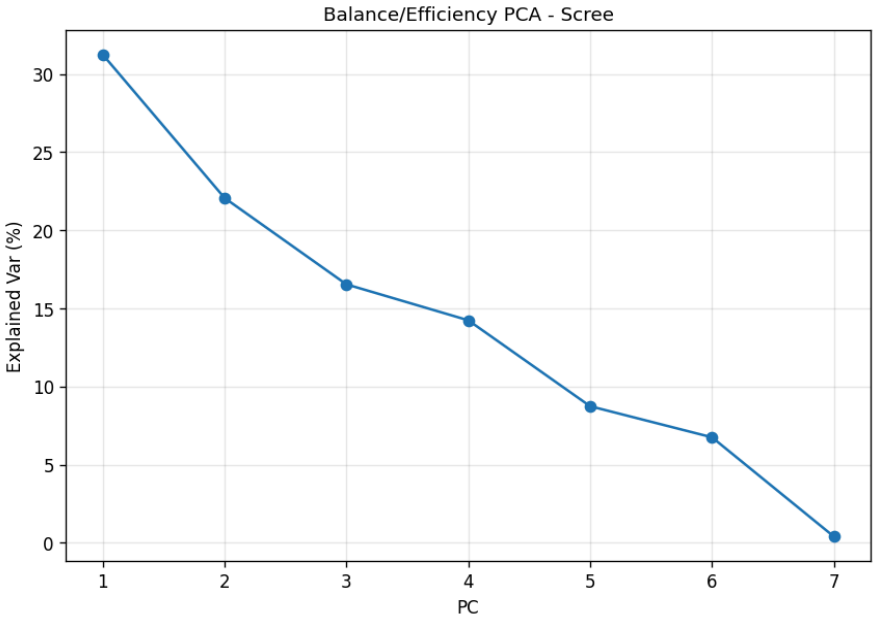
- byte_balance correlaciona positivamente con fwd_eff y negativamente con bwd_eff y overhead_ratio.
- down_up_ratio (del dataset original) se alinea con los balances por bytes/tasa.
- flow_eff comparte señal con fwd_eff, pero no es redundante total; aporta estabilidad a nivel de flujo completo.

El PCA sobre este subespacio arrojó una estructura compacta. El scree plot mostró que PC1≈31%, PC2≈22%, PC3≈16% y PC4≈14%; con 3–4 componentes se captura ~80–85% de la varianza. La proyección 2D evidenció agrupaciones separables: normales quedan más compactas y cercanos al

origen, mientras que ataques se dispersan y forman ramas asociadas a distintos modos de desbalance/eficiencia.

Según los loadings (balance_pca_loadings.xlsx), PC1 está dominada por eficiencias (flow_eff, fwd_eff vs overhead_ratio), y PC2 por balances (byte_balance, rate_balance), lo que legitima mantener ambas familias en el set de features y se concluye que no son intercambiables.





En conclusión, se detectó un desbalance direccional muy marcado en ataques, y ausente o moderado en tráfico normal. Gracias a los boxplots de eficiencia se obtuvo que los ataques presentan payload dominante en “forward” y respuesta con poco payload.

El PCA mostró que la eficiencia y balance son ejes ortogonales de variación, y se conservarán ambas familias para el modelado. Este bloque aporta señales muy fuertes y complementarias a las de intensidad y ritmo vistas en secciones anteriores.

En esta etapa no se repiten limpiezas ya realizadas sobre variables base en bloques previos. En cambio, se procede a verificar y normalizar únicamente las variables no tratadas previamente y a dejar consistentes las variables derivadas de balance y eficiencia que se emplearán en el modelado.

Se computará de forma determinística el conjunto de derivadas (pkt_balance, byte_balance, rate_balance, fwd_eff, bwd_eff, flow_eff) a partir de los contadores crudos ya validados en etapas anteriores. Las razones logarítmicas se calculan con correcciones numéricas mínimas (evitando divisiones por cero) y las eficiencias se acotan explícitamente al intervalo [0,1].

Cualquier infinito o valor no numérico que pueda emerger en el proceso se convierte en NaN y se documenta su frecuencia para asegurar trazabilidad, sin introducir decisiones adicionales de eliminación en este bloque. El objetivo es homogeneizar estas métricas compuestas y garantizar que su escala y dominio sean compatibles con los pasos estadísticos y de modelado posteriores.

Dado que los balances, en especial rate_balance, presentan colas largas, se contempla un recorte suave por percentiles (p. ej., P0.5–P99.5) muy útil ya que el algoritmo de aprendizaje es sensible a valores atípico. El recorte se aplica exclusivamente al conjunto de entrenamiento.

Para evitar multicolinealidad y mejorar estabilidad, se evalúa la redundancia entre balances y eficiencias. Sobre el conjunto final de variables derivadas se aplican estandarización (media 0, desvío 1) y, cuando se busque compacidad, un PCA local del bloque que permita representar la señal en 2–3 componentes con alta varianza explicada. Este procedimiento no reemplaza a las variables originales, las complementa.

7. *Control de flujo y comportamiento temporal*

El objetivo de esta sección es caracterizar cómo se envían los datos en ráfagas (subflujos) y en bloques (bulk) para distinguir tráfico IoT legítimo, típicamente periódico y en lotes pequeños, de patrones maliciosos, ráfagas unidireccionales, bloques grandes y tasas sostenidas. Este bloque se centra en magnitudes direccionales nuevas (subflow/bulk) y en variaciones de overhead que no fueron analizadas en intensidad/ritmo/eficiencia previos. La hipótesis: ataques DoS, floods y exfiltraciones suelen manifestarse como picos de subflujos y bloques voluminosos con asimetría marcada.

Se tomó la siguiente familia de variables:

Grupo	Variables	Señal Esperada
Subflujos (ráfagas) Forward	fwd_subflow_pkts, fwd_subflow_bytes	Ataques: valores altos, picos de ráfagas
Subflujos (ráfagas) Backward	bwd_subflow_pkts, bwd_subflow_bytes	Ataques: bajos/irregulares
Bloques (bulk) Forward	fwd_bulk_bytes, fwd_bulk_packets, fwd_bulk_rate	Ataques; muy altos

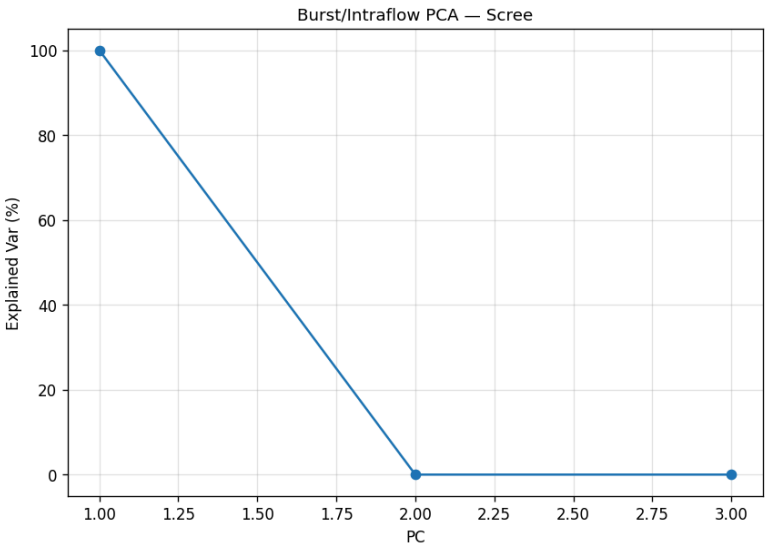
Bloques (bulk) Backward	bwd_bulk_bytes, bwd_bulk_packets, bwd_bulk_rate	Ataques: bajos, con picos aislados
Overhead direccional: Forward	fwd_header_size_min, fwd_header_size_max	Dispersión alta puede indicar manipulación
Overhead direccional: Backward	bwd_header_size_min, bwd_header_size_max	Dispersión alta, respuestas truncadas

Además, se decidió crear un nuevo conjunto de variables que modelan mejor el comportamiento, son más compactas y brindan mayor información.

Grupo	Variable Derivada	Fórmula	Interpretación
Asimetría de subflujos	subflow_balance_bytes	log10((fwd_subflow_bytes + 1) / (bwd_subflow_bytes + 1))	>0: ráfagas dominan en forward; <0: en backward
Asimetría de bloques (bytes)	bulk_balance_bytes	log10((fwd_bulk_bytes + 1) / (bwd_bulk_bytes + 1))	>0: bloques dominan en forward
Asimetría de tasa en bloques	bulk_rate_balance	log10((fwd_bulk_rate + 1e-9) / (bwd_bulk_rate + 1e-9)))	Picos unidireccionales en floods
Intensidad local de ráfagas	subflow_bytes_sum	fwd_subflow_bytes + bwd_subflow_bytes	Magnitud total de ráfagas del flujo
Intensidad local de bloques	bulk_bytes_sum	fwd_bulk_bytes + bwd_bulk_bytes	Magnitud total de bloques del flujo

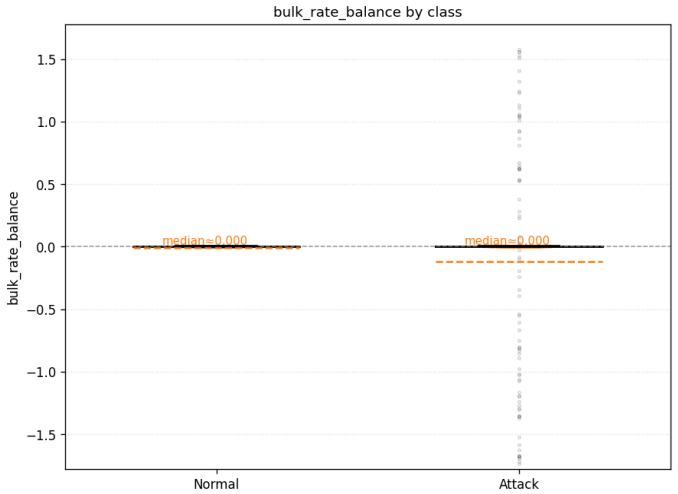
Se generaron boxplots de subflow_balance_bytes y bulk_balance_bytes muestran medianas≈0 y rangos ínfimos en ambas clases, la serie es prácticamente constante y las variables casi degeneradas.

Además, el PCA indica que PC1 ≈ 100% de la varianza y PC2–PC3 ≈ 0%. En términos prácticos, el sub-bloque vive en 1 sola dimensión porque dos de las tres variables aportan varianza nula o despreciable.

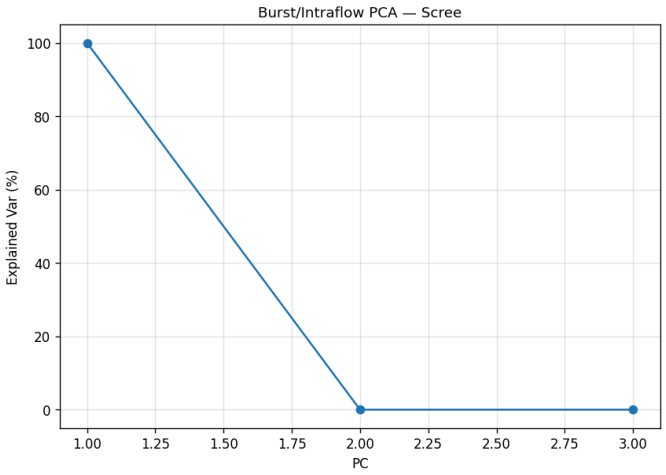


Debido a los resultados obtenidos en al análisis de las variables derivadas, se decidió utilizar estas variables para el entrenamiento, pero dado el resultado atípico con respecto a los anteriores, se decidió realizar el análisis de las variables originales, para reducir el margen de error y concluir si el resultado del análisis de variables derivadas proviene de la elección de la fórmula de cada variable o de los datos en el Dataset.

Para las variables bulk_balance_bytes y subflow_balance_bytes, ambos boxplots están prácticamente planos en 0 para Normal y Attack. Las medianas anotadas son ≈0 y la dispersión visible es mínima. Esto sugiere que, a nivel de bytes, los bursts están balanceados en la mayoría de los casos, sin poder separarlos entre clases.



PCA: el scree muestra 100% de varianza en PC1 y ~0% en las siguientes componentes; en el scatter 2D todos los puntos quedan alineados sobre el eje X (PC2≈0). Esto confirma colinealidad/constancia en las originales: efectivamente hay una sola dimensión informativa y es débil para separar clases.



Los resultados con variables originales confirman que el dataset, para este bloque, es mayormente "cero-céntrico" y con señal concentrada en outliers: los balances en bytes son casi constantes y el desbalance en tasa sólo se manifiesta en las colas de la clase Attack. El PCA respalda esta lectura al colapsar toda la varianza en una única componente.

Por ello, la decisión de entrenar con variables derivadas se mantiene: esas transformaciones captan mejor las asimetrías operativas (eficiencia, razón de direcciones, balances normalizados) que las originales no reflejan. Repetir el análisis con originales sugiere que el "resultado atípico" de las derivadas no proviene de un defecto de fórmula, sino de que las derivadas exponen patrones reales que las originales esconden.

Explicación División de Data Set

Una vez finalizada la etapa de limpieza, se procedió a dividir el conjunto de datos en tres subconjuntos con el objetivo de disponer de información separada para entrenamiento, ajuste y evaluación del modelo predictivo.

La división se realizó de manera estratificada sobre la variable binaria is_attack, que distingue entre tráfico benigno (0) y malicioso (1). Esto asegura que cada subconjunto mantenga la misma proporción de clases que el conjunto original, evitando sesgos durante el entrenamiento o evaluación del modelo.

Proporciones utilizadas

- 70 % Entrenamiento (train): utilizado para ajustar los parámetros del modelo.
- 15 % Validación (valid): empleado para ajustar hiperparámetros y prevenir sobreajuste.
- 15 % Prueba (test): reservado exclusivamente para evaluar el desempeño final del modelo.

Estas proporciones permiten disponer de un volumen adecuado de datos para el entrenamiento sin comprometer la representatividad de los conjuntos de validación y prueba.

Durante la limpieza se identificaron registros cuya columna Attack_type no especificaba una etiqueta clara (por ejemplo "unknown", "none", "-", o vacía). Estos casos no fueron incluidos en ninguno de

los tres conjuntos principales, ya que su clase real es incierta y su incorporación podría introducir ruido o sesgo en el modelo supervisado. En lugar de eliminarlos, se almacenaron en un archivo independiente (unknown.csv) para permitir su análisis posterior.

Esto posibilita, realizar un análisis exploratorio de sus características comparándolos con flujos normales y de ataque. Utilizarlos en tareas no supervisadas o de detección de anomalías, donde la ausencia de etiqueta no impide su aprovechamiento.

Análisis de Datos Mediante Visualizaciones sobre el Conjunto de Entrenamiento