
This is the **published version** of the bachelor thesis:

Gotés Atienza, Pol. *Caracterització de les Dinàmiques Temporals del Trànsit de Xarxes TOR : Mateixa web, diferents moments*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334425>

under the terms of the  license.

Caracterització de les Dinàmiques Temporals del Trànsit de Xarxes TOR:

Mateixa web, diferents moments

Pol Gotés Atienza

Resum — La xarxa Tor (*The Onion Router*) és una infraestructura de comunicació fonamental per garantir la privadesa i l'anonimat a Internet. Mitjançant l'encaminament per capes, el trànsit es distribueix a través de múltiples nodes voluntaris (*onion routing*), xifrant la informació pas a pas per impedir que observadors externs en puguin rastrejar l'origen o vincular-lo a un usuari concret. Malgrat aquestes proteccions, les tècniques d'anàlisi de trànsit continuen sent un desafiament per a l'anonimat. En aquest context, el projecte té com a objectiu analitzar i caracteritzar les dinàmiques temporals del trànsit web anonimitzat a la xarxa Tor. A partir de l'estudi de fitxers de captura de xarxa (PCAP), obtinguts en diferents dies i franges horàries per a un mateix conjunt de servidors web, s'avaluen les seves característiques i evolució al llarg del temps. Mitjançant l'aplicació de tècniques d'aprenentatge automàtic, el treball pretén comprendre com muten els patrons de xarxa al llarg del temps (*concept drift*) i identificar com aquesta variació permet identificar el trànsit web.

Paraules clau — TOR, dinàmiques temporals, privadesa, anonimat, trànsit web, Website Fingerprinting

Abstract— The Tor network (*The Onion Router*) is a fundamental communication infrastructure to ensure privacy and anonymity on the Internet. Through layered routing, traffic is distributed across multiple volunteer nodes (*onion routing*), encrypting information step by step to prevent external observers from tracing its origin or linking it to a specific user. Despite these protections, traffic analytics techniques remain a challenge to anonymity. In this context, the project aims to analyze and characterize the temporal dynamics of anonymized web traffic on the Tor network. Based on the study of network capture files (PCAP), obtained on different days and time slots for the same set of web servers, their characteristics and evolution over time are evaluated. Through the application of machine learning techniques, the work aims to understand how network patterns mutate over time (*drift concept*) and identify how this variation allows web traffic to be identified.

Index Terms— TOR, Temporal Dynamics, Privacy, Anonymity, Web Traffic, Website Fingerprinting



1 INTRODUCCIÓ - CONTEXT DEL TREBALL

Internet és avui dia la xarxa de comunicació global per excel·lència, permetent l'intercanvi massiu d'informació a nivell mundial. Aquesta xarxa es fonamenta en la família de protocols TCP/IP, dissenyada principalment per garantir la resiliència i l'entrega de paquets de dades entre múltiples dispositius interconnectats de forma eficient.

No obstant això, el disseny original de l'arquitectura TCP/IP no es va pensar perquè fos anònim ni per protegir la privadesa de les comunicacions per defecte. Les adreces IP d'origen i de destí, així com les metadades bàsiques de les sessions, són inherentment visibles per als *routers* intermedis i els proveïdors de serveis d'Internet (ISP), permetent el rastreig de l'activitat dels usuaris.

Per evitar aquesta falta de privadesa, han aparegut infraestructures de comunicació com la xarxa Tor (*The Onion Router*). Aquest sistema encapsula i xifra el trànsit en múltiples capes i l'encamina a través d'una seqüència de tres nodes (*onion routing*) distribuïts voluntàriament, de manera que cap entitat externa pugui vincular directament l'adreça IP de l'usuari amb el servidor web que està visitant.

Tot i el complex xifratge de les dades, diversos estudis han demostrat que la xarxa Tor continua sent vulnerable a

atacs d'Anàlisi de Trànsit com el *Website Fingerprinting* (WF). Aquestes tècniques permeten a un atacant passiu (que només observa el flux de dades) inferir quina pàgina web s'està visitant mitjançant l'aplicació d'aprenentatge automàtic sobre metadades com la mida, la direcció i la temporalitat dels paquets capturats (PCAPs).

Tradicionalment, els atacs de *Website Fingerprinting* assumeixen que l'entrenament dels models i l'avaluació de les traces es realitzen en el mateix instant de temps. Aquest plantejament, però, xoca amb una realitat que els models clàssics tendeixen a ignorar: les pàgines web canvien constantment al llarg del temps, un fenomen conegut com a concept drift que es detalla més endavant.

1.1 Definició del problema

L'evolució temporal de les traces de xarxa obliga a redefinir com caracteritzem les pàgines web. Si considerem una web W_1 i una web W_2 , cadascuna generarà captures diferents depenent del temps (per exemple, de W_1 :traça₁ a W_1 :traça₂).

Aquest projecte pretén analitzar les captures obtingudes entre l'usuari que inicia la connexió i el primer node TOR, per obtenir noves mètriques (*features*) capaces de des-

criure com varia el trànsit d'una mateixa web durant una setmana. Al finalitzar, aquestes mètriques d'evolució temporal serviran per explicar les variabilitats de cada web i millorar la classificació global d'un conjunt de traces desconegudes.

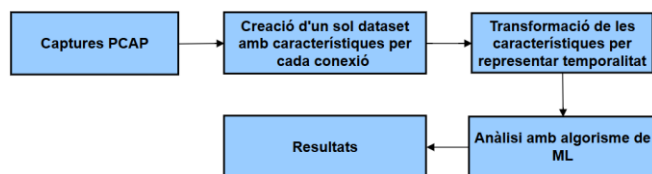
Aquest enfocament crea una representació setmanal robusta de les webs, resistent a aquests canvis explicats.

Per altra banda, aquest TFG no utilitza la temporalitat com una *feature* per si sola, perquè el dataset no reflecteix el context temporal com la hora o dia correctament. Es a dir, com que el procés de captura dels fitxers PCAP és automàtic no trobem valor en que un usuari hagi accedit a les 8:00 cada dia. Amb un altre dataset es podria concloure que aquest usuari té l'hàbit de llegir la web del seu diari preferit cada dia a la mateixa hora, i per tant es podria extreure la web.

Aquest treball, davant la deriva temporal, no busca modelar la mutació de cada web, sinó construir-ne una representació setmanal estable. Agrupa totes les connexions d'una mateixa web durant una setmana i en resumeix el comportament, obtenint una signatura més robusta que una connexió individual.

1.2 Flux de la informació

El procés complet segueix cinc etapes encadenades. Primer, la captura del trànsit, on s'obtenen els fitxers PCAP de cada connexió web a través de la xarxa Tor. A continuació, la creació d'un únic dataset en què cada connexió queda representada per un conjunt de característiques unidimensionals, calculades a partir dels paquets de la captura. La tercera etapa és la transformació temporal, on aquestes característiques s'agrupen per setmana i web per derivar-ne noves mètriques que descriuen com varia el trànsit al llarg del temps. Tot seguit, l'anàlisi amb *Machine Learning*, on l'algorisme XGBoost s'entrena i intenta classificar les webs a partir d'aquestes mètriques. Finalment, l'obtenció i interpretació dels resultats, que permeten avaluar si la dimensió temporal millora la identificació respecte de l'enfocament tradicional.



1.3 Que és TOR?

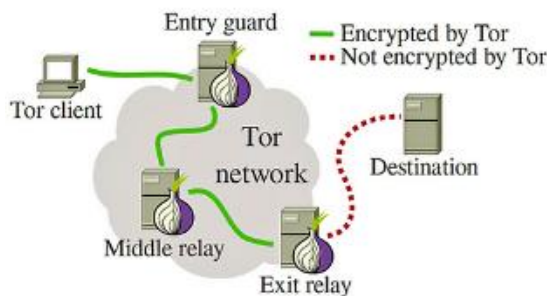
La xarxa Tor és una infraestructura de comunicació anònima que utilitza una xarxa de nodes voluntaris per encaminar el trànsit. La base de l'anonimat de la xarxa Tor és la separació de la identitat de l'usuari amb la seva activitat fragmentant la ruta que pren el trànsit.

Un circuit Tor està format per 3 salts entre *routers* escollits prèviament per l'usuari. D'aquesta manera l'usuari pot establir una connexió xifrada diferent amb cada node mitjançant Diffie-Hellman. A mesura que el missatge avança pel circuit, cada node desxifra la seva part corresponent mostrant només les metadades necessàries per arribar al següent salt. Per tant, durant l'encaminament cap

node Tor coneix tant el destí com l'origen del missatge.

Els tres nodes tenen rols ben diferenciats. El node d'entrada (entry guard) és l'únic que coneix l'adreça IP real de l'usuari, però no sap a quin servidor es connecta. El TFG utilitza el tràfic entre aquest node i l'usuari. El node intermedi (middle relay) actua de pont i no coneix ni l'origen ni el destí. El node de sortida (exit relay) és qui finalment estableix la connexió amb el servidor de destí i, per tant, coneix la destinació, però no l'usuari que hi ha al darrere. És aquesta separació de coneixement el que impedeix que cap entitat individual pugui correlacionar origen i destí.

La informació viatja pel circuit en unitats de mida fixa anomenades cel·les (cells). Aquesta uniformitat és deliberada: en igualar la mida de totes les unitats de transport, Tor dificulta que un observador pugui inferir el contingut a partir de la longitud dels paquets. Aquest detall no és rellevant per al meu treball, ja que les captures analitzades han sigut capturades entre el client i el primer node TOR (entry guard).



Il·lustració 1 Font: Cisco Learning Network [11]

Aquest anonimat fa de Tor una eina de doble ús. D'una banda, és fonamental per a la llibertat d'expressió: periodistes, activistes, dissidents polítics i persones que viuen sota règims censurats l'utilitzen per comunicar-se i accedir a informació sense ser identificats ni perseguits, i organitzacions de defensa dels drets humans en recomanen l'ús per protegir fonts i denunciants. D'altra banda, aquesta mateixa capacitat d'ocultació atreu activitats il·lícites, especialment a través dels serveis ocults (la coneguda dark web), on s'allotgen mercats de béns il·legals, distribució de contingut delictiu i altres usos criminals. Aquesta tensió entre privadesa legítima i abús és precisament el que motiva l'estudi de tècniques d'anàlisi de trànsit com el Website Fingerprinting: comprendre fins a quin punt l'anonimat de Tor és vulnerable té implicacions tant per a la protecció dels usuaris legítims com per a la capacitat de les autoritats de perseguir activitats delictives.

1.4 Concept Drift

El terme concept drift (o deriva de concepte) descriu el fenomen pel qual les propietats estadístiques de les dades que un model intenta predir canvien al llarg del temps. Quan això passa, un model entrenat en un moment determinat veu degradada la seva precisió a mesura que les dades noves s'allunyen de les que va aprendre. És un problema transversal a l'aprenentatge automàtic, però resulta especialment acusat en el Website Fingerprinting.

Tradicionalment, els atacs de Website Fingerprinting assumeixen que l'entrenament dels models i l'avaluació de

les traces es realitzen en el mateix instant de temps. La realitat, però, és que les pàgines web canvien constantment a causa de contingut dinàmic, anuncis o actualitzacions. Aquesta evolució fa que els fingerprints de les webs siguin altament dinàmics i en degrada dràsticament la capacitat d'identificació a mesura que passa el temps. Per il·lustrar-ho, un model entrenat amb captures d'un portal de notícies pot deixar de reconèixer-lo poques setmanes després, simplement perquè la portada ha canviat de titulars, ha incorporat nous bàners o ha modificat la mida de les imatges. Davant d'això, la resposta habitual és reentrenar els models periòdicament o buscar característiques que envelleixin lentament.

En el cas concret de la xarxa Tor, aquest efecte s'agreuja, com que el trànsit ja és difícil d'analitzar pel xifratge i l'encaïment per capes, qualsevol degradació addicional deguda al pas del temps complica encara més la identificació. Entendre i mesurar com es manifesta el concepte drift sobre el trànsit web és, per tant, un pas previ imprescindible per decidir com tractar-lo, que és el que es planteja en aquest treball.

2 OBJECTIUS I METODOLOGIA

2.1 Proposta d'Objectius

Objectiu General: L'objectiu general del treball és analitzar i caracteritzar les dinàmiques temporals del trànsit de la xarxa Tor amb la finalitat d'avaluar l'estabilitat dels patrons de trànsit web al llarg del temps i demostrar que aquesta dimensió temporal es pot explotar per millorar la identificació de webs

Objectius:

- Recopilar i revisar l'estat de l'art sobre la xarxa Tor, atacs d'anàlisi de trànsit i els efectes de l'evolució temporal sobre les traces de xarxa.
- Crear o obtenir un dataset de múltiples connexions web per entrenar i provar com d'efectives són les mètriques aplicades.
- Configurar i preparar un entorn de laboratori capaç de processar múltiples fitxers CSV de gran volum per extreure'n les *features* pertinents.
- Desenvolupar funcions d'extracció de mètriques temporals rellevants (al llarg d'una setmana) de les connexions web.
- Avaluar com varien les característiques de xarxa al pas del temps per definir noves *features* i que permetin caracteritzar el trànsit web sobre xarxes anònimes.

2.2 Metodologia

- **Fase Documental:** Investigació bibliogràfica en publicacions científiques sobre onion routing, atacs de Website Fingerprinting i anàlisi temporal del trànsit. Aquesta fase serveix per situar el projecte respecte de l'estat de l'art, identificar quines característiques de trànsit s'han utilitzat històricament i, sobretot, entendre com la literatura ha tractat el concepte drift, que és el punt de partida sobre el qual el treball construeix la

seva hipòtesi.

- **Fase de Preprocessament:** Es faran servir eines d'anàlisi de paquets com TShark, juntament amb llibreries de Python, per depurar i interpretar els fitxers PCAP subministrats. L'objectiu d'aquesta etapa és convertir les captures en brut, que contenen tota la informació dels paquets, en una seqüència ordenada i neta de metadades rellevants (mida, direcció i marca de temps de cada trama), descartant el soroll i la informació no aprofitable per a la classificació.
- **Fase d'Extracció de Característiques (Feature):** Es dissenyarà un conjunt de scripts en Python per transformar les captures en conjunts de dades estructurats (taules o fitxers CSV). S'extrauran dos tipus de mètriques: les unidimensionals, calculades sobre cada connexió individual i les d'agrupació setmanal, que anomenarem temporals. Aquestes últimes són mètriques de mètriques, ja que es deriven de les unidimensionals i resumeixen el comportament al llarg d'una setmana.
- **Fase d'Anàlisi i Aprenentatge Automàtic (Machine Learning):** S'implementarà i entrenarà un model de classificació basat en l'algorisme XGBoost utilitzant les característiques de xarxa prèviament extretes. S'avaluarà la precisió del model, entrenant-lo amb un 80% de les captures del dataset i comprovant la precisió amb el 20% restant, els resultats de precisió serviran per comparar el rendiment de diferents datasets. A més, s'analitzarà la mètrica d'importància per identificar quines característiques de trànsit són més determinants.
- **Fase de Síntesi:** Redacció dels resultats obtinguts, interpretació de les comparacions entre escenaris i formalització de les conclusions tècniques a la memòria del TFG. En aquesta etapa final es contrasten les xifres de precisió dels diferents datasets per determinar si la hipòtesi inicial, que la temporalitat millora la identificació de webs, queda confirmada.

3 ESTAT DE L'ART

La identificació de pàgines web sobre canals xifrats té una llarga trajectòria de recerca. Abans d'aplicar-se sobre Tor, les primeres tècniques de Website Fingerprinting es van plantejar sobre connexions HTTPS i túnels VPN, on un observador situat al canal pot veure les mides i els temps dels paquets encara que el contingut estigui xifrat. Els treballs clàssics van demostrar que, tot i el xifratge, un atacant passiu pot inferir el lloc visitat a partir de la mida, la direcció i la seqüència temporal dels paquets [1], [4]. Aquests primers enfocaments depenien d'una enginyeria manual de característiques —histogrames de mida de paquet, comptatges per direcció o marques de temps— i de classificadors clàssics, però ja apuntaven que les metadades de trànsit filtren molta més informació de la que es podria esperar. De fet, és precisament d'aquesta família de característiques d'on parteixen les mètriques unidimensionals que utilitzo en aquest treball.

Un aspecte transversal a tota aquesta literatura és la distinció entre dos escenaris d'avaluació. En el món tancat

(closed-world) s'assumeix que l'usuari només pot visitar un conjunt limitat i conegut de webs, i el problema es redueix a classificar entre aquestes; en el món obert (open-world), en canvi, l'usuari pot visitar qualsevol lloc i el model ha de distingir les webs vigilades de la resta del trànsit. El món tancat ofereix precisions més altes però és menys realista, mentre que el món obert és més exigent i s'acosta més a un escenari d'atac real. Aquest treball se situa en un escenari de món tancat, amb un conjunt fix de vint webs, que és el punt de partida habitual per validar la utilitat d'unes característiques noves abans de portar-les a condicions més complexes.

L'aparició del deep learning va suposar un salt important en la precisió d'aquests atacs, eliminant la necessitat de dissenyar manualment les característiques i aprenent-les directament de les seqüències de paquets [2]. Aquests models van assolir resultats molt elevats en condicions de laboratori, fet que va desplaçar el focus de la recerca des de "quines característiques extreure" cap a "com fer els models robustos en escenaris reals". Una línia especialment rellevant per a aquest treball és la que estudia el valor de la informació temporal. Tik-Tok [3] va demostrar que els temps d'arribada dels paquets, sovint menystinguts davant de la mida, contenen informació discriminativa significativa. En aquesta mateixa direcció, altres propostes incorporen mecanismes d'atenció sobre les característiques temporals per ponderar-ne les més informatives [6], o exploten les ràfegues (bursts) de trànsit —agrupacions de paquets consecutius en una mateixa direcció— com a tret distintiu d'una web [9]. Aquests treballs comparteixen una intuïció que el meu projecte fa servir: la dimensió temporal del trànsit amaga senyal aprofitable, tot i ser alhora una de les més sensibles al soroll i a les condicions de la xarxa.

El problema central que aborda el meu projecte és el concepte drift: la degradació de la precisió dels models a mesura que les webs evolucionen [5]. Diversos estudis han quantificat aquest efecte i han constatat que un model entrenat en un moment determinat pot perdre una part molt important de la seva precisió en qüestió de setmanes, a mesura que el contingut de les webs canvia [10]. Davant d'això, la literatura ha tractat el fenomen majoritàriament com una amenaça a mitigar, ja sigui reentrenant els models periòdicament o dissenyant característiques que envelleixin lentament i resisteixin la deriva del trànsit [7]. En un context relacionat, altres treballs apliquen aquestes tècniques a la classificació de trànsit a la dark web, on la identificació de serveis ocults té un interès específic per a la seguretat [8].

La meua contribució es diferencia precisament en l'enfocament. Mentre que en el Website Fingerprinting la temporalitat es considera habitualment un enemic a combatre, aquest treball l'afronta de manera diferent: en lloc de mitigar la deriva o de modelar-ne la dinàmica, agrupa les captures d'una mateixa web per setmanes i en construeix una representació agregada i estable. La hipòtesi de partida és que aquesta signatura setmanal és més robusta i discriminativa que una connexió individual, ja que resumeix el comportament típic de la web atenuant el soroll i la variabilitat puntual.

4 DISSENY I IMPLEMENTACIÓ DEL SISTEMA

El processament i l'anàlisi de les dades es duu a terme amb l'execució de 4 scripts. L'objectiu és validar que la creació de mètriques que expressin l'evolució del trànsit d'una web proporciona una precisió major que les mètriques per connexions individuals.

Per al dataset he utilitzat les captures de trànsit web (.pcap) del laboratori Zeus de la UAB. El conjunt original està format per 20 webs amb 1.200 captures cadascuna, és a dir, 24.000 mostres. Les captures es van obtenir entre la workstation que inicia la connexió i el primer node Tor, per tant, l'única informació desconeguda és la IP final dels datagrames.

A continuació s'explica els scripts utilitzats junt amb una breu descripció:

- `processar_dataset.sh` – Processa les 24.000 captures PCAP amb TShark, extreu els camps i els envia per una pipe al script `features.py`.
- `features.py` – Transforma els camps de cada captura en característiques unidimensionals i les guarda com una fila dins un CSV. Cada fila representa les característiques d'una connexió web. Consulteu la Taula 1.
- `temporal.py` – A partir del CSV anterior, agrupa les connexions per web, setmana i dia de la setmana, es calcula la mitjana diària per evitar l'esbiaixament i es calculen els *features* temporals de cada setmana. Com a resultat obtenim un nou CSV de 160 mostres, on cada fila representa les característiques temporals d'una setmana. Consulteu la Taula 2.
- `xtree.py` – Analitza les característiques temporals utilitzant un 80% del dataset per l'entrenament i el 20% restant per a les proves. Com a resultat mostra la precisió d'encerts i el percentatge d'importància que ha tingut cada *feature* durant l'anàlisi.
- `FeaturesNormals160.py` – Transforma el dataset creat per `features.py` en un de 160 mostres, agafant una mostra diària d'una sola setmana sumat a una mostra aleatòria de la setmana vinent. Així amb un total de 20 webs y 8 mostres per web obtenim 160 mostres.
- `FeaturesHibrids.py` – Transforma el dataset creat per `features.py` per afegir nous *features* que aportin temporalitat, però mantenint el volum de mostres (24.000)

Aquesta arquitectura encadenada té un avantatge pràctic important: en lloc de generar primer un CSV amb els camps extrets de cada captura PCAP i després processar cada CSV individual per construir el CSV final de *features*, els camps que TShark extreu de les captures es passen directament al `features.py`. D'aquesta manera s'evita la creació de fitxers intermedis, ja que les dades flueixen en streaming des de TShark cap a l'script de Python.

4.1 Disseny de mètriques temporals

La idea central del treball és que les connexions individuals d'una web poden variar segons el dia en què es realitzen, agrupades per setmanes i resumides amb mètriques

setmanals, donen lloc a una representació estable i robusta de cada web. Per capturar aquest comportament, he decidit agrupar les connexions d'una mateixa web per setmanes i estudiar com es comporten les seves *features* al llarg d'aquella finestra.

El punt clau del disseny és que les mètriques temporals són, en realitat, mètriques de mètriques. És a dir, no es calculen directament sobre els paquets, sinó sobre les *features* unidimensionals que ja he extret de cada connexió i que es poden veure a continuació.

Característica	Nom del feature	Definició
Sumatori de Longitud	sumframelen	Indica la mida total de la connexió en bytes
Volum per Direcció	sum_up, sum_down	Separa el trànsit d'anada del de tornada
Màxims de Trama	max_up, max_down	Identifica la presència de fitxers grans
Rati de Paquets	ratio	Proporció entre càrrega i descàrrega
Activitat post-inicial	num_frames_after_one_sec	Captura l'activitat post-càrrega inicial

Taula 1: Taula de Característiques unidimensionals

Per a cada feature normal d'una web, per exemple, el volum total descarregat, primer en calculo la mitjana de cada dia de la setmana (mitjana interdiària) i, a partir d'aquests valors diaris, en derivo mètriques que en resumeixen el comportament setmanal: principalment el seu nivell típic, mitjançant la mitjana, i la seva dispersió, mitjançant el rang interquartílic. D'aquesta manera, cada feature unidimensional original genera diverses *features* temporals que la caracteritzen al llarg del temps. A continuació es mostren els *features* temporals calculats.

Mètrica temporal	Definició
Desviació Estàndard	Variació absoluta de les mitjanes diàries respecte a la mitjana setmanal.
Rang (max - min)	Indica la diferència entre el valor més gran i més petit d'un <i>feature</i> .
Rang Interquartílic	Mesura la dispersió del 50% central de les dades.
Coeficient de Variació	Mesura la dispersió de les dades de cada dia amb la mitjana de la setmana
Mitjana	La mitjana dels valors d'un <i>feature</i> durant la setmana

Taula 2: Taula de característiques temporals

Un cop obtinc totes aquestes mètriques per a totes les setmanes i totes les webs, genero un únic document que recull aquesta variabilitat, que posteriorment analitza l'algorisme XGBoost.

Un punt important del disseny és evitar l'esbiaixament per sobrerepresentació. Si una web té, per exemple, 100 captures del dilluns i només 1 per a la resta de dies, el patró

del dilluns tindria un pes desproporcionat sobre el càlcul setmanal. Per resoldre-ho, agrupo prèviament les connexions per dia de la setmana i en faig la mitjana; només després calculo la variació sobre la setmana sencera. Així, cada dia contribueix de manera equitativa al resultat final, independentment de quantes captures s'hagin obtingut aquell dia.

4.2 Selecció de característiques i hiperparàmetres

Cal tenir en compte que els *features* no són l'únic factor a l'hora d'entrenar l'algorisme XGBoost. Els hiperparàmetres són variables de configuració externa que l'algorisme no pot aprendre de les dades per si sol. Aquests paràmetres (com la profunditat de l'arbre o la velocitat d'aprenentatge) s'han de definir prèviament per controlar la complexitat del model i evitar que memoritzi el conjunt d'entrenament, garantint així que pugui generalitzar correctament en escenaris amb poques dades. A la Taula 3 es mostren els hiperparàmetres utilitzats durant les anàlisis.

Hiperparàmetre	Valor
max_depth	4
learning_rate	0.05
n_estimators	300
subsample	0.7
colsample_bytree	0.7
min_child_weight	3
random_state	42
eval_metric	mlogloss
use_label_encoder	False

Taula 3: : Hiperparàmetres utilitzats

A més, hi ha *features* temporals que poden "confondre" a l'algorisme i empitjorar els encerts. Això provoca que el model intenti trobar patrons on només hi ha soroll aleatori produït per errors puntuals de connexió. També hi ha *features* com el *max_up* que calcula la mida màxima de les trames durant la setmana que no és gens útil degut al MTU.

Per resoldre-ho he creat un script *reducció_features.py* basat en el *xtree.py*, que executa un mateix anàlisi múltiples cops traient *features* seqüencialment per detectar si la seva absència millora la precisió i per tant signfica que no és realment útil.

El procés de selecció seqüencial va deixar com a supervivents, gairebé en exclusiva, mètriques basades en la mitjana setmanal (5 de les 6 *features* finals), acompanyades del rang interquartílic.

L'anàlisi del pes que XGBoost assigna a cada feature confirma aquesta selecció i hi aporta un matís important. En el dataset temporal, la importància no recau en una sola mètrica, sinó que es reparteix de manera molt equilibrada entre les sis *features* finals, totes amb un valor entre 0,146 i 0,191 (vegeu la Taula 8 de l'Annex). Encapçala la llista la mitjana de la mida total de trama (*sumframelen_interday_mean*, 0,1906), seguida de les mitjanes de l'activitat posterior a la càrrega, dels paquets de pujada i del volum de baixada i de pujada; tanca el conjunt el rang interquartílic de la mida total (*sumframelen_interday_iqr*, 0,1459). Aquest repartiment és un punt fort de l'enfocament: en no

dependre d'una única característica, la signatura setmanal és més robusta davant de variacions puntuals

Feature	Descripció
Mitjana Setmanal	La mitjana dels tots els valor d'un mateix feature dins una mateixa setmana
Rang interquartílic (IQR)	La diferència entre el valor més gran i més petit dins el rang interquartílic 50

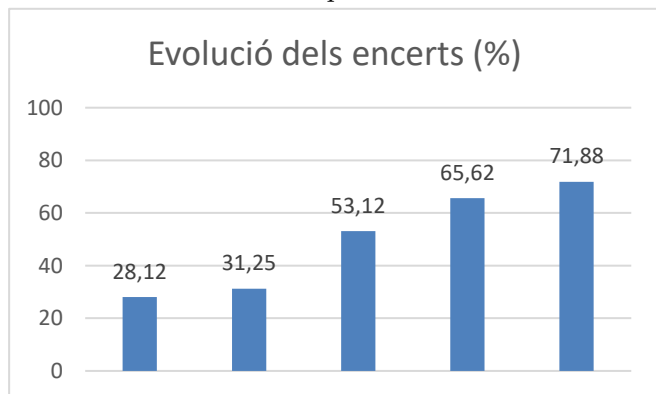
Taula 4: Features seleccionats

Aquest predomini té una explicació raonable. La mitjana setmanal captura el nivell de trànsit típic d'una web (quant volum mou, en quina proporció entre pujada i baixada) i al calcular-se de manera interdiària no queda esbiaixada pel nombre de captures de cada dia. És, per tant, una signatura estable i poc sensible a fluctuacions puntuals. El rang interquartílic, per la seva banda, descriu la dispersió central del comportament sense deixar-se arrossegar pels valors extrems, a diferència del rang absolut o de la desviació estàndard, més vulnerables a anomalies.

En canvi, les mètriques basades en màxims van quedar sistemàticament descartades. Com s'ha explicat, el MTU de la xarxa limita molt el valor d'aquest feature, de manera que el màxim observat aporta més soroll que senyal. Aquesta coherència entre el comportament del protocol i el resultat empíric de la selecció reforça la validesa de l'enfocament.

D'aquesta manera s'ha optimitzat l'anàlisi amb els *features* correctes i s'ha mantingut uns hiperparàmetres comuns que mantenen l'algorisme XGboost determinista.

Aquests paràmetres i seleccions ha millorat la precisió del dataset temporal (160 mostres) substancialment. A continuació podem veure aquesta millora, però cal tenir en compte que aquest gràfic només té valor per demostrar la millora dels resultats obtinguts a mida que s'intentava trencar els baixos valors de precisió inicials.



Il·lustració 2: Gràfica de millora de la precisió

4.3 Validació del procés

Per validar els resultats obtinguts i poder realitzar una comparació qualitativa i justa s'han comparat dos escenaris. L'objectiu es evitar comparar resultats de datasets amb

volums diferents que puguin suposar una desavantatge durant l'anàlisi de l'algorisme XGboost.

Per fer-nos una idea, el CSV de *features* temporals és un dataset de 160 mostres, on cada web aporta 8 setmanes. Per altra banda, tenim el dataset original amb 24.000 captures i per tant, 1200 mostres per web.

La primera validació és comparar el dataset de *features* temporals de 160 mostres amb un dataset del mateix volum, obtingut al reduir el dataset original (de 24.000) amb l'script *FeaturesNormals.py*.

Per aconseguir-ho es va agafar una captura per dia d'una mateixa setmana junt amb 1 captura addicional aleatòria de la següent setmana. Això ens deixa amb 8 captures que multiplicades per les 20 webs 160 captures. D'aquesta manera igualement el volum i evitem la reducció de la precisió de les captures individuals donada per comparar captures de diferents setmanes.

La segona validació es comparar dos datasets de 24.000 mostres. Per fer-ho es va utilitzar exactament el mateix dataset original de *features* normals, però afegint-ne nous *features* temporals a un d'ells amb l'script *FeaturesHíbrids.py*. Per afegir aquesta temporalitat es va calcular la diferència de cada *feature* unidimensional amb la mediana setmanal. Es a dir, el dataset es va agrupar per setmanes i dins de cada setmana es va calcular la mediana de cada *feature*, llavors es calcula la diferència de cada mostra amb aquesta mediana. Amb aquest *feature* mesurem si una captura individual varia molt del valor més comú de la setmana.

Per tant, es comparen els següents datasets:

- **Dataset normal.** CSV de 24.000 mostres amb *features* unidimensionals calculats a partir de les captures proporcionades per la universitat.
- **Dataset normal reduït.** CSV de 160 mostres resultat de reduir el dataset normal de 24.000 mostres (1 mostra diària de la mateixa setmana més 1 mostra d'un dia aleatori de la setmana següent).
- **Dataset híbrid.** Combinació dels *features* normals amb una mètrica temporal afegida, la diferència de cada *feature* respecte de la mediana de la seva setmana. Es va escollir la mediana perquè evita que valors anòmals puntuals distorsionin el valor representatiu del trànsit. Aquest enfocament manté el gran volum de dades i hi incorpora la variació temporal
- **Dataset Temporal:** Dataset de 160 mostres resultat de l'agrupació per setmanes del dataset normal, calculant *features* temporals per cada setmana. Aquest dataset representa només la temporalitat setmanal de les web.

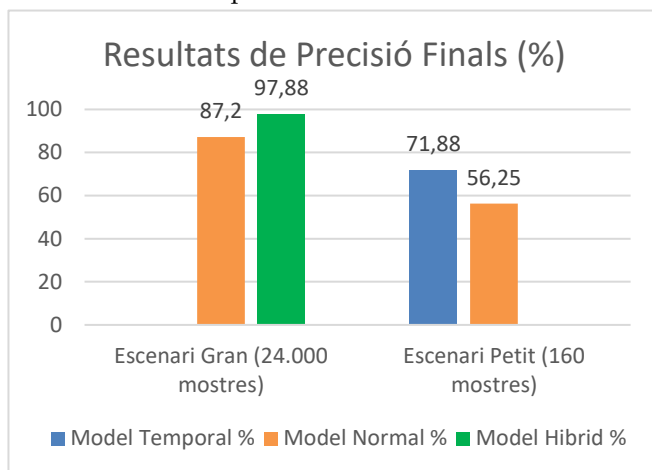
5 RESULTATS

Finalment amb uns hiperparàmetres fixos decidits per a XGBoost, la nova selecció de *features* realment útils, la igualtat de volums i amb una relació d'entrenament i test de 80/20, s'han realitzat les diferents proves.

Per a la prova dels dataset de 160 mostres, el dataset temporal ha resultat guanyador amb una precisió de

71.88% comparat amb un 56.25% pel dataset unidimensional.

Per altra banda, la comparativa dels datasets híbrid i normal ha resultat en un 87.20% de precisió per dataset normal i un 97.88% pel híbrid.



Il·lustració 3: Comparativa resultats finals

Interpretant aquests valors, la incorporació de la temporalitat aporta una millora absoluta de 10,68 punts percentuals en l'escenari de gran volum (del 87,20% al 97,88%) i de 15,63 punts en l'escenari petit (del 56,25% al 71,88%). Tots dos experiments confirmen, doncs, la hipòtesi de partida, i és especialment destacable que el guany relatiu sigui més gran justament en l'escenari amb menys dades, on la informació temporal compensa l'escassetat de mostres.

Una lectura addicional d'aquests resultats és que els dos enfocaments temporals responen a necessitats diferents. El model temporal pur és útil quan només es disposa d'un conjunt reduït de captures agrupades per setmana, ja que extreu el màxim de senyal d'aquesta estructura. El model híbrid, en canvi, és l'opció preferent quan es disposa de gran volum de dades, perquè no renuncia a la riquesa de les *features* clàssiques i només hi suma la variació temporal com a senyal complementari.

Per validar aquestes conclusions, és útil comparar el pes de les *features* entre els quatre datasets (Taules 5 a 8 de l'Annex). El primer fet remarcable és que la mida total de la connexió (sumframelen) és la característica més determinant en tots els escenaris sense excepció: encapçala el dataset normal (0,3326), l'híbrid (0,2497), el normal reduït (0,1960) i, en la seva versió de mitjana setmanal, també el temporal (0,1906). Això la confirma com el tret identificador més estable d'una web, independentment de l'enfocament.

El segon fet valida la decisió de descartar els màxims. Al dataset normal, max_up amb prou feines arriba a 0,0289 i max_down a 0,0233, situant-se a la cua de la classificació; per això la selecció seqüencial els elimina i cap mètrica de màxim sobreviu al dataset temporal. Aquest resultat empíric és coherent amb el fet que la mida de trama estigui fitada per l'MTU de la xarxa.

Finalment, la comparativa confirma que la temporalitat afegida al dataset híbrid aporta senyal real. Les *features* de diferència respecte de la mediana setmanal, que no existeixen

als altres datasets, apareixen entre les deu més determinants de l'híbrid: sum_down_diff_median (0,0664), sumframelen_diff_median (0,0618) i sum_up_diff_median (0,0390), entre altres. La seva presència explica, almenys en part, el salt de precisió del 87,20% al 97,88% i demostra que la variació temporal complementa les *features* clàssiques en lloc de substituir-les.

6 CONCLUSIONS

Els resultats obtinguts em permeten concloure que la temporalitat del trànsit aporta informació discriminativa real per a la identificació de webs sobre la xarxa Tor. Quan es comparen models sota les mateixes condicions (mateixos hiperparàmetres, mateix percentatge d'entrenament i mateix volum de dades), les característiques temporals milloren la classificació respecte de les característiques convencionals.

La conclusió principal és que el concept drift, tradicionalment considerat un obstacle per als atacs de Website Fingerprinting, es pot afrontar de manera diferent. En lloc de combatre la variabilitat temporal, agrupar les captures per setmanes i resumir-ne el comportament permet obtenir una signatura estable i robusta de cada web, que millora la identificació respecte d'una sola captura. L'enfocament híbrid és especialment prometedor, ja que combina el volum i la riquesa de les *features* clàssiques amb el valor afegit de la variació temporal, assolint la millor precisió de tot el treball (97,88%).

Com a línies de treball futures, seria interessant ampliar el dataset a un nombre més gran de webs i de setmanes per confirmar l'estabilitat dels patrons a llarg termini. També seria valuós que el dataset tingués captures reals, per així poder aprofitar l'hora i el dia en que es va realitzar. Actualment com que el dataset va ser creat de forma automàtica, 1 captura cada hora, no te cap sentit pensar que un usuari consultaria el diari pel matí i plataformes de streaming o altres per la tarda.

Pel que fa als objectius plantejats a l'inici, en faig un balanç positiu. He revisat l'estat de l'art sobre Tor, el Website Fingerprinting i el concept drift. He obtingut i processat un dataset real de 24.000 captures provinent del laboratori Zeus de la UAB. He preparat un entorn de laboratori capaç de tractar aquest volum de dades de manera eficient. He desenvolupat noves funcions d'extracció de mètriques temporals i un mecanisme de selecció automàtica de *features*. I, finalment, he avaluat com varien aquestes característiques amb el temps, demostrant que la variació temporal és explotable per millorar la identificació del trànsit. Tots els objectius específics s'han assolit.

7 ÚS DE LA IA GENERATIVA

En l'elaboració del meu Treball de Final de Grau (TFG), s'han utilitzat eines d'Intel·ligència Artificial (IA) generativa, concretament ChatGPT i Gemini, amb l'objectiu principal de millorar en la claredat del text, informació sobre l'algorisme XGBoost davant a diferents situacions i com a suport a la creació i modificació dels scripts de creació de *features*.

L'ús d'aquestes IA s'ha dut a terme de manera ètica i exclusivament com a eina de suport. Tota la informació i els suggeriments que he rebut de la IA han estat revisats, contrastats i validats de forma crítica.

BIBLIOGRAFIA

- [1] M. Juárez, S. Afroz, G. Acar, C. Díaz i R. Greenstadt, "[A Critical Evaluation of Website Fingerprinting Attacks](#)," a *Proc. 2014 ACM SIGSAC Conf. on Computer and Communications Security (CCS '14)*, Scottsdale, AZ, EUA, 2014, pp. 263–274, doi: 10.1145/2660267.2660368.
- [2] V. Rimmer, D. Preuveneers, M. Juárez, T. Van Goethem i W. Joosen, "[Automated Website Fingerprinting through Deep Learning](#)," a *Proc. 2018 Network and Distributed System Security Symp. (NDSS)*, 2018, doi: 10.14722/ndss.2018.23105.Fdf.
- [3] M. S. Rahman, P. Sirinam, N. Mathews, K. G. Gangadhara i M. Wright, "[Tik-Tok: The Utility of Packet Timing in Website Fingerprinting Attacks](#)," *Proceedings on Privacy Enhancing Technologies*, vol. 2020, no. 3, pp. 5–24, jul. 2020, doi: 10.2478/popets-2020-0043.
- [4] A. Panchenko, L. Niessen, A. Zinnen i T. Engel, "[Website Fingerprinting in Onion Routing Based Anonymization Networks](#)," a *Proc. 10th Annual ACM Workshop on Privacy in the Electronic Society (WPES '11)*, Chicago, IL, EUA, 2011, pp. 103–114, doi: 10.1145/2046556.2046570.
- [5] G. I. Webb, R. Hyde, H. Cao, H. L. Nguyen i F. Petitjean, "[Characterizing Concept Drift](#)," *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 964–994, abr. 2016, doi: 10.1007/s10618-015-0448-4.
- [6] J. Gong et al., "[WFCAT: Augmenting Website Fingerprinting With Channel-Wise Attention on Timing Features](#)," *IEEE Transactions on Dependable and Secure Computing*, vol. 23, no. 1, pp. 149–163, gen.–feb. 2026, doi: 10.1109/TDSC.2025.3605197.
- [7] "[Enhancing Website Fingerprinting Attacks against Traffic Drift](#)," a *Proc. 2026 Network and Distributed System Security Symp. (NDSS)*, San Diego, CA, EUA, feb. 2026, doi: 10.14722/ndss.2026.230059.
- [8] D. Xu, J. Zhang, F. Huang, Y. Chen i B. Zheng, "[CRS: A Website Fingerprinting Technique for Dark Web Tor Traffic](#)," preprint, mar. 2025, doi: 10.21203/rs.3.rs-6169549/v1.
- [9] D. Li, Y. Pan i Y. Xu, "[Tor Traffic Classification Based on Burst Features](#)," a *Proc. 21st Int. Conf. on Intelligent Computing (ICIC 2025)*, Ningbo, Xina, jul. 2025, pp. 771–783, doi: 10.65286/icic.v21i1.86615.
- [10] A. Shusterman, R. David i Y. Oren, "[Understanding and Addressing Concept Drift in Website Fingerprinting](#)," *Computer Networks*, vol. 275, art. 111811, feb. 2026, doi: 10.1016/j.com-net.2025.111811.
- [11] "The Onion Router (Tor)," Cisco Learning Network. [En línia]. Disponible: <https://learningnetwork.cisco.com/s/question/0D56e0000EBumUTCQZ/the-onion-router-tor> [Accés: 11 abril 2026].
- [12] "XGBoost Parameters," XGBoost Documentation. [En línia]. Disponible: <https://xgboost.readthedocs.io/en/stable/parameter.html> [Accés: 11 abril 2026].
- [13] "XGBoost Hyperparameter Optimization," XGBoost Documentation. [En línia]. Disponible: https://xgboost.readthedocs.io/en/stable/tutorials/param_tuning.html [Accés: 11 abril 2026].

8 ANNEX

Dataset normal (24.000 mostres) - precisió 87,20%

Feature	Pes (importància)
sumframelen	0,3326
sum_up	0,2041
sum_down	0,1666
num_frames_after_one_sec	0,1018
count_up	0,0774
count_down	0,0395
max_up	0,0289
ratio	0,0258
max_down	0,0233

Taula 5: Importància de les features del dataset normal de 24.000 mos-tres.

Dataset normal reduït (160 mostres) - precisió 56,25%

Feature	Pes (importància)
sumframelen	0,1960
sum_up	0,1546
count_up	0,1397
sum_down	0,1362
num_frames_after_one_sec	0,1357
count_down	0,1177
ratio	0,0689
max_down	0,0511

Taula 6: Importància de les features del dataset normal reduït a 160 mostres.

Dataset híbrid (24.000 mostres + diferència amb la mediana setmanal) - precisió 97,88%

Feature	Pes (importància)
sumframelen	0,2497
sum_up	0,1502
sum_down	0,1242
num_frames_after_one_sec	0,0952
count_up	0,0760
sum_down_diff_median	0,0664
sumframelen_diff_median	0,0618
sum_up_diff_median	0,0390
count_down	0,0376
count_up_diff_median	0,0327

Taula 7: Importància de les features del dataset híbrid.

Dataset temporal (160 mostres) - precisió 71,88%

Feature	Pes (importància)
sumframelen_interday_mean	0,1906
num_frames_after_one_sec_interday_mean	0,1753
count_up_interday_mean	0,1716
sum_down_interday_mean	0,1654
sum_up_interday_mean	0,1512
sumframelen_interday_iqr	0,1459

Taula 8: Importància de les sis features del dataset temporal.