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

Megias Barrio, Jaime. *Anàlisi empírica de la indistingibilitat del trànsit P2P v2 de Bitcoin*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334443>

under the terms of the  license.

Anàlisi empírica de la indistingibilitat del trànsit P2P v2 de Bitcoin

Jaime Megías Barrio

30 de juny de 2026

Resum– En aquest treball s'analitza la capacitat d'ofuscació del protocol BIP324 de Bitcoin mitjançant tècniques de classificació automàtica basades en metadades de trànsit. Es desplega un entorn experimental controlat amb Warnet sobre Kubernetes per generar captures reals de trànsit Bitcoin xifrat i comparar-les amb trànsit HTTPS.

Paraules clau– Bitcoin, BIP324, Anàlisi de trànsit xifrat, Aprenentatge automàtic, Warnet, Kubernetes.

Abstract–This work analyzes the obfuscation capability of Bitcoin's BIP324 protocol through automatic traffic classification techniques based on metadata. A controlled experimental environment is deployed using Warnet over Kubernetes to generate real encrypted Bitcoin traffic captures and compare them with HTTPS traffic.

Keywords– Bitcoin, BIP324, Encrypted Traffic Analysis, Machine Learning, Warnet, Kubernetes.

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

EN els darrers anys, la criptomoneda Bitcoin ha evolucionat no només com a sistema de pagaments descentralitzat, sinó també com a infraestructura tecnològica de referència en àmbits com la seguretat, la criptografia i la privadesa de les comunicacions. El seu funcionament es basa en una xarxa global de nodes que intercanvien dades mitjançant un protocol *peer-to-peer* (P2P), a través del qual es propaguen transaccions i blocs entre els participants de la xarxa. [28]. La importància d'aquesta xarxa recau en el fet que qualsevol usuari pot participar en la validació i propagació d'informació sense necessitat d'una entitat centralitzada. Aquesta arquitectura distribuïda proporciona resistència davant censura i punts únics de fallada, però alhora fa que les comunicacions entre nodes siguin especialment sensibles des del punt de vista de la privadesa i la seguretat. La capacitat d'identificar nodes Bitcoin dins d'una xarxa o monitorar els seus patrons de comunicació podria facilitar tècniques de vigilància, censura o anàlisi de tràfic sobre els participants

Per aquest motiu, la comunitat de desenvolupadors de Bit-

coin ha introduït diferents millores orientades a reforçar la privadesa del tràfic P2P. Una de les més recents és la proposta que defineix una nova versió del protocol de transport P2P amb xifratge d'extrem a extrem entre nodes Bitcoin, el BIP324.

L'objectiu principal de BIP324 és dificultar la identificació passiva del tràfic Bitcoin per part d'observadors externs [29]. Tot i que les dades de la cadena de blocs són públiques, les metadades associades a la comunicació entre nodes continuen sent rellevants. Per exemple, identificar quins dispositius executen nodes Bitcoin o detectar quan es produeix activitat dins la xarxa podria permetre aplicar mecanismes de bloqueig, filtratge o monitoratge selectiu.

Gràcies al xifratge introduït per BIP324, el contingut dels paquets queda ocult davant eines tradicionals d'anàlisi de xarxa, com Wireshark. Arran d'això, el protocol incorpora tècniques d'ofuscació per dificultar encara més el reconeixement del tràfic. Com a conseqüència, el tràfic generat pels nodes Bitcoin passa a ser molt més semblant al d'altres protocols xifrats habituals, com HTTPS [30] o SSH [31].

Des del punt de vista de la ciberseguretat, això planteja una qüestió rellevant: fins a quin punt BIP324 aconsegueix ocultar efectivament el tràfic Bitcoin davant tècniques modernes d'anàlisi de tràfic? Més concretament, és possible continuar distingint aquest protocol utilitzant únicament metadades de xarxa, fins i tot quan el contingut dels paquets és inaccessible?

• E-mail de contacte: 1673265@uab.cat

• Menció: Enginyeria de Tecnologies de la Informació

• Treball tutoritzat per: Cristina Perez Sola del departament de Tecnologies de la informació

• Curs 2025/26

Aquestes preguntes constitueixen la motivació principal d'aquest treball. En lloc de buscar mecanismes per monitorar tràfic Bitcoin, l'objectiu és avaluar l'efectivitat real de les proteccions introduïdes per BIP324. Per fer-ho, es combinaran tècniques de *Machine Learning* amb anàlisi estadística de metadades de xarxa per estudiar si encara existeixen patrons residuals detectables en el tràfic xifrat.

L'objectiu d'aquest Treball de Fi de Grau és, doncs, analitzar la capacitat d'ofuscació del protocol BIP324 mitjançant tècniques de classificació automàtica basades en metadades de tràfic.

1.1 Objectius del TFG

1.1.1 Objectiu general

L'objectiu general d'aquest Treball de Fi de Grau és avaluar fins a quin punt el protocol BIP324 de Bitcoin aconsegueix protegir el tràfic P2P davant tècniques de fingerprinting basades en metadades de xarxa. Per fer-ho, es planteja desenvolupar i validar un sistema experimental capaç d'analitzar tràfic xifrat i determinar si, malgrat la protecció del contingut, encara es conserven patrons temporals o estadístics que permetin distingir-lo d'altres protocols xifrats mitjançant tècniques de Machine Learning.

Aquest objectiu implica dissenyar una metodologia d'anàlisi basada en captures de paquets, construir un entorn controlat que reproduïxi condicions properes a una xarxa P2P real i estudiar si les defenses introduïdes per BIP324 redueixen de manera efectiva la identificabilitat del trànsit davant un observador extern.

1.1.2 Objectius específics

1. Dissenyar una infraestructura de proves que permeti simular el comportament d'una xarxa Bitcoin amb nodes interconnectats i generar tràfic xifrat de forma controlada.
2. Construir un conjunt de dades representatiu amb captures de tràfic BIP324 i altres protocols xifrats, especialment HTTPS, per poder comparar comportaments de transport sota condicions similars.
3. Definir i extreure característiques de nivell paquet i flux que siguin rellevants per a l'anàlisi de tràfic xifrat, com la mida dels paquets, els intervals entre arribades, la direccionalitat o la finestra TCP.
4. Avaluar diversos models de Machine Learning, com Random Forest o SVM, per determinar si poden distingir tràfic BIP324 a partir exclusivament de metadades de xarxa.
5. Analitzar quines propietats residuals del transport encara són observables malgrat el xifratge, i valorar fins a quin punt aquestes propietats comprometen l'objectiu de privadesa del protocol.

6. Integrar tots els components del sistema de generació, captura, processament i classificació en un flux de treball automatitzat i reproduïble que permeti ampliar l'experimentació en fases posteriors.

Donats aquests objectius, volem respondre a les següents preguntes: fins a quin punt el trànsit P2P v2 de Bitcoin es manté distingible respecte d'altres connexions TCP xifrades quan l'anàlisi es limita exclusivament a les metadades de xarxa disponibles? En aquest context, es planteja també quines propietats residuals del transport —com ara la mida dels paquets, els intervals temporals entre transmissions o el comportament de la finestra TCP— continuen aportant informació prou rellevant per permetre la classificació del trànsit. Finalment, cal estudiar en quina mesura la implementació de BIP324 redueix la identificabilitat d'aquest tràfic en comparació amb altres protocols xifrats convencionals, i si aquesta millora es tradueix realment en una protecció més gran davant tècniques de fingerprinting passiu.

1.2 Metodologia General

El desenvolupament d'aquest TFG segueix una metodologia àgil [26] adaptada a projectes acadèmics d'enginyeria, estructurada en iteracions setmanals que permeten validar progressivament cada component del sistema. Aquesta aproximació, inspirada en pràctiques Scrum però simplificada per a un desenvolupador únic, facilita l'adaptació a imprevistos tècnics com la complexitat de la negociació BIP324, prioritzant la reproductibilitat i la integració contínua.

1.3 Límits del projecte

Aquest treball se centra en l'anàlisi del tràfic xifrat de Bitcoin en un entorn controlat, i per tant presenta diverses limitacions que cal tenir en compte en la interpretació dels resultats:

- **Entorn local i aïllat:** totes les proves es duen a terme en un clúster local desplegat sobre Kubernetes i Minikube, sense connexió a la mainnet de Bitcoin ni a nodes externs. Això permet controlar les variables experimentals, però redueix la representativitat del tràfic respecte a condicions reals de la xarxa global.
- **Abast del protocol:** l'estudi se centra exclusivament en el tràfic associat al protocol BIP324 en comparació amb altres protocols xifrats. No s'analitza el contingut dels missatges, sinó únicament la informació observable a partir de metadades de transport.
- **Escenari experimental limitat:** el nombre de nodes i de captures és suficient per construir un dataset útil, però no arriba a la complexitat ni a l'escala d'una xarxa Bitcoin real amb gran volum de trànsit simultani.
- **Recursos computacionals:** l'execució i l'entrenament dels models es realitzen amb recursos locals, fet

que condiciona tant la mida del dataset com la complexitat dels algorismes que es poden explorar.

- **Comparació amb tràfic xifrat:** la validació se centra en protocols xifrats comparables, com HTTPS, perquè la pregunta d'investigació principal no és distingir tràfic xifrat de tràfic obert, sinó determinar si BIP324 continua sent identificable davant altres connexions segures.

1.4 Ús d'IA Generativa

Ús mínim declarat: s'ha fet i es farà un ús declarat d'eines d'IA generativa (PerplexityAI/Gemini) sota les següents condicions:

- Àmbit d'aplicació: suport tècnic en el debugging de scripts Bash (syntaxi tcpdump, gestió de rangs de pods {0..23}), suport en la redacció i correcció formal de la memòria en català, i estructuració de l'informe.
- Grau d'ús: estimat en menys del 5% del contingut total. S'utilitza exclusivament com a eina de suport per millorar la claredat de l'expressió i resoldre errors puntuals de programació.
- Autoria: els objectius, el disseny metodològic, l'execució del pipeline i l'obtenció de resultats són 100% originals de l'autor.

2 CONTEXT TEÒRIC: BIP324 I DETECCIÓ DE TRÀFIC

2.1 BIP324: protocol P2P v2 de Bitcoin

BIP324 és una proposta desenvolupada pels desenvolupadors de Bitcoin Core per introduir un nou protocol de transport P2P xifrat entre nodes Bitcoin [29]. Aquesta nova versió del protocol, incorporada a Bitcoin Core v27+, substitueix el model tradicional de comunicació en text pla per un sistema com xifratge i ofuscació integrats.

L'objectiu principal és reduir la capacitat d'identificació passiva del tràfic Bitcoin mitjançant tècniques modernes de *fingerprinting*. Per aconseguir-ho, BIP324 combina diversos mecanismes criptogràfics i d'ofuscació. L'explicació de tots aquests mecanismes es troba a l'apèndix 6.4.

Aquest conjunt de tècniques converteix BIP324 en un dels protocols P2P més avançats pel que fa a la protecció davant tècniques d'anàlisi passiva de tràfic.

2.2 Tècniques generals anti-fingerprinting

BIP324 no és l'únic protocol modern que incorpora mecanismes d'ofuscació de tràfic. Altres tecnologies orientades

a la privadesa també utilitzen tècniques similars per dificultar la identificació del protocol mitjançant inspecció passiva de xarxa. Veure la taula en l'apèndix 6.1.

Aquestes tècniques tenen com a objectiu dificultar les anomenades tècniques d'*Encrypted Traffic Analysis* (ETA), és a dir, mètodes capaços d'identificar protocols xifrats sense necessitat de descodificar-ne el contingut.

Tot i aquestes proteccions, diversos estudis han demostrat que encara poden existir vulnerabilitats associades a les metadades del transport TCP, com ara:

- intervals temporals entre paquets (*Inter-Arrival Time* o IAT),
- variabilitat del tamany dels paquets,[34]
- patrons de ràfegues (*burst patterns*),
- comportament de la finestra TCP (*TCP Window Size*).

La capa TCP és especialment rellevant perquè és la infraestructura real sobre la qual BIP324 i HTTPS acaben operant. TCP introdueix mecanismes de control de flux, confirmació d'entrega, segmentació i retransmissió que afecten directament les metadades observables per un tercer: mida dels segments, IAT, ràfegues, ACK clocking i dinamisme de la finestra. Per això, encara que el payload estigui xifrat, el patró de transport continua depenent de decisions implementatives i de l'estat de la connexió.[35]

En aquest treball, les mètriques més rellevants són l'Inter-Arrival Time i les estadístiques de longitud de paquet, ja que capturen a la freqüència de transmissió i la regularitat volumètrica del flux. La mida de la finestra TCP també pot ser una font de distingibilitat, però alhora un factor de biaix si l'entorn experimental no està homogeni. Això explica el primer problema detectat en Polar: el model no estava aprenent el protocol, sinó la pila TCP i les seves configuracions en el sistema de control. En canvi, la topologia distribuïda de Warnet va reduir aquest efecte i va permetre centrar el model en patrons més propers al comportament real del trànsit. Aquestes propietats poden actuar com una empremta estadística del protocol, cosa que permet tècniques de classificació basades en *Machine Learning* fins i tot quan el payload està completament xifrat.

3 GENERACIÓ DEL DATASET

3.1 Pipeline

3.1.1 Metodologia General

El desenvolupament d'aquest TFG segueix una metodologia àgil [26] adaptada a projectes acadèmics d'enginyeria, estructurada en iteracions setmanals que permeten validar progressivament cada component del sistema. Aquesta aproximació, inspirada en pràctiques Scrum però simplificada per a un desenvolupador únic, facilita l'adaptació

ció a imprevistos tècnics com la complexitat de la negociació BIP324, prioritzant la reproductibilitat i la integració contínua.

3.1.2 Flux de Treball Principal

El pipeline central es divideix en cinc fases iteratives, les quals es repeteixen seqüencialment fins a assolir la convergència en les mètriques de classificació del sistema:

1. **Desplegament (Deploy):** Configuració i llançament d'una xarxa simulada mitjançant l'eina Warnet sobre un clúster local de Kubernetes i Minikube [1, 4, 7]. Aquest entorn permet generar una topologia de 24 nodes Bitcoin amb el suport per al protocol de transport xifrat BIP324 activat al port 28333, a més del protocol tradicional P2Pv1 al port 9333 i un node de control amb tràfic HTTPS al port 443.
2. **Generació de tràfic (TrafficGeneration):** Execució d'escenaris automatitzats per forçar connexions P2P regulades pel nou protocol (processos de *handshake* i propagació de blocs sintètics) de manera conjunta amb tràfic HTTPS simulat [2, 3]. L'objectiu d'aquesta fase és reproduir patrons macroscòpics de comportament reals de la xarxa Bitcoin sense necessitat d'interaccionar amb la xarxa principal (*mainnet*).
3. **Captura (Capture):** Recollida massiva de paquets de xarxa en brut en format PCAP mitjançant la utilització de l'eina de línia de comandes *tcpdump* [18]. El procés s'enfoca exclusivament en l'extracció de metadades del flux, com ara els intervals temporals d'arribada (*Inter-Arrival Time* o IAT), la mida exacta dels paquets i la direccionalitat dels fluxos, evitant deliberadament qualsevol intent de desxiframent del *payload*.
4. **Extracció de característiques (FeatureExtraction):** Transformació dels fitxers de captures PCAP en conjunts de dades estructurats en format CSV a través de scripts propis en Python recolzats en les llibreries Scapy i Pandas. En aquesta fase es seleccionen i calculen les 15 variables clau de nivell de paquet i de flux (com l'entropia del flux, el comportament de les ràfegues TCP i l'estat dels *flags*) fortament validades per la literatura científica en l'àmbit de l'anàlisi de tràfic encriptat (*Encrypted Traffic Analysis* o ETA) [8].
5. **Entrenament ML (MLTraining):** Avaluació empírica de models d'aprenentatge automàtic supervisat, com ara *Random Forest* i màquines de vectors de suport (SVM), fent ús de la suite d'eines de Scikit-learn [9, 11]. Els models s'entrenen per resoldre un problema de classificació binària (BIP324 vs. HTTPS), optimitzant els hiperparàmetres mitjançant tècniques de validació creuada i mesurant la seva robustesa amb mètriques de rendiment estàndard com l'*F1-score* [8].

Cada iteració culmina amb un sprint review: anàlisi de logs Kubernetes, visualització de PCAPs (Wireshark) i proves de model, ajustant hiperparàmetres o topologies de xarxa

segons resultats [6,17]. Veure la taula de les capes amb la seva definició a l'apèndix 6.2

3.2 Arquitectura de l'entorn de simulació

Per avaluar la resiliència de BIP324 davant l'anàlisi de trànsit, l'entorn experimental es va dividir en dues fases amb diferents nivells d'abstracció. La primera va consistir en una prova de concepte en un entorn controlat amb dues instàncies de Bitcoin Core en una xarxa privada, gestionades via RPC, que va permetre validar el funcionament bàsic del protocol, les connexions xifrades i els mecanismes inicials de captura de trànsit. Tot i ser útil per establir una base funcional, aquesta configuració presentava limitacions importants en termes d'escala, variabilitat i representativitat del comportament real de la xarxa.

Per superar aquestes restriccions, la segona fase es va implementar amb una arquitectura distribuïda formada per 12 instàncies independents de Bitcoin Core interconnectades en una topologia P2P més realista. Aquest desplegament va permetre augmentar significativament el nombre de connexions simultànies, la diversitat dels fluxos de dades i el volum total d'informació generada. En aquest context, es van dur a terme proves de connectivitat, sincronització entre nodes, verificació de connexions entrants i sortints, així com la validació del procés de *handshake* i l'anàlisi del comportament de BIP324 en escenaris amb múltiples participants. Gràcies a aquesta aproximació, es van obtenir resultats més robustos i representatius, cosa que permet avaluar amb més precisió l'impacte del xifratge de transport en els patrons observables i la resistència del protocol en condicions properes a un entorn real.

3.3 Generació de tràfic Bitcoin xifrat

Per tal de generar una captura representativa de trànsit Bitcoin real sobre la infraestructura desplegada, es va desenvolupar un script d'automatització en Bash que interactuava directament amb els nodes Bitcoin Core executats dins del clúster Kubernetes. L'arquitectura utilitzada estava formada per dotze nodes Warnet connectats mitjançant el protocol P2P de Bitcoin v2 amb xifrat BIP324 habilitat. Dins d'aquesta topologia, el node `tank-0000` es va designar com a emissor principal de les transaccions, mentre que els nodes `tank-0001` fins a `tank-0011` actuaven com a receptors dels pagaments.

L'automatització de l'activitat es va estructurar en tres fases operatives:

1. **Mineria inicial de 101 blocs:** El protocol de consens de Bitcoin estableix que les recompenses de mineria (*coinbase transactions*) no es poden gastar fins que acumulen 100 confirmacions. Per aquest motiu, abans de generar qualsevol activitat econòmica, el node emissor va minar 101 blocs consecutius mitjançant la comanda RPC `generatetoaddress`. Aquesta operació va permetre desbloquejar les recompenses

obtingudes al primer bloc i disposar de prou saldo suficient per finançar totes les transferències posteriors.

2. **Generació continuada de transaccions:** Un cop disponibles els fons, el node `tank-0000` va iniciar un procés de transferències periòdiques cap a la resta de nodes de la xarxa. Per a cada enviament, el node receptor generava una nova adreça Bitcoin mitjançant la comanda RPC `getnewaddress`, garantint que cada transacció utilitzava una destinació única. Posteriorment, el node emissor executava la comanda `sendtoaddress` enviant exactament 0,001 BTC a aquesta adreça.

El procés es va repetir durant 50 rondes consecutives. En cada ronda es realitzaven 11 transferències (una per cada node receptor), separades per una pausa de 0,5 segons entre operacions. Això va generar un total de:

$$50\text{rondes} \times 11\text{trans/ronda} = 550\text{trans}$$

Aquest patró va produir un flux constant de missatges Bitcoin sobre la xarxa P2P, simulant l'activitat d'usuaris reals que creen noves adreces i efectuen pagaments de manera continuada.

3. **Mineria de confirmació i propagació:** En finalitzar cadascuna de les 50 rondes, el node emissor minava un bloc addicional. Aquesta operació tenia un doble objectiu: confirmar les 11 transaccions generades durant la ronda i provocar la propagació dels nous blocs a través de tota la topologia distribuïda. Com a resultat, a més de les 550 transaccions generades, es van crear 50 blocs addicionals de confirmació que van desencadenar l'intercanvi de missatges de validació, sincronització i anunci de blocs entre tots els nodes Warnet.

En conjunt, l'escenari experimental va generar una càrrega sostinguda de trànsit Bitcoin composta per 550 transaccions distribuïdes durant aproximadament 30 minuts d'execució, juntament amb la propagació de 151 blocs (101 blocs inicials per desbloquejar fons i 50 blocs de confirmació). Aquesta activitat va permetre capturar tant missatges associats a la difusió de transaccions com al procés complet de mineria, validació i sincronització de blocs sota el protocol P2P v2 xifrat pel port 28333 [2, 3].

3.4 Generació d'altres tipus de tràfic xifrat

Per a la construcció de la classe de referència (Classe 0 / HTTPS), es va desplegar un script d'automatització encarregat de generar trànsit xifrat no relacionat amb Bitcoin. L'objectiu era obtenir una línia base de comunicacions TLS habituals a Internet que permetés comparar-les posteriorment amb el trànsit produït pel protocol Bitcoin P2P xifrat. D'aquesta manera, els algorismes de classificació podien entrenar-se amb exemples de trànsit xifrat heterogeni i aprendre a distingir patrons específics de Bitcoin respecte a altres protocols també protegits criptogràficament.

El trànsit generat es va basar principalment en connexions HTTPS establertes mitjançant l'eina `curl`. A di-

ferència d'un navegador web convencional, `curl` no renderitza pàgines ni executa codi JavaScript, però sí que estableix connexions TLS completes i intercanvia dades reals amb els servidors remots, fet suficient per reproduir fluxos xifrats representatius a nivell de xarxa.

L'script accedia repetidament a diversos serveis web amb perfils diferents. A l'apèndix 6.5 es troben les pàgines on s'ha accedit per la generació de tràfic.

La càrrega es va executar durant 300 iteracions consecutives. En cada iteració es realitzaven 19 peticions HTTPS distribuïdes entre els diferents serveis, seguides d'una pausa d'un segon abans d'iniciar el següent cicle. En conseqüència, es van generar aproximadament:

$$[300 \times 19 = 5700]$$

connexions HTTPS independents al llarg de l'experiment.

A més del trànsit web, es van incorporar altres comunicacions xifrades i de suport per incrementar la diversitat dels fluxos capturats. Concretament, cada iteració incloïa tres consultes DNS mitjançant `nslookup` sobre el domini `example.com` i un intent de connexió SSH local executat amb la comanda `ssh localhost`. Aquestes operacions representen mecanismes habituals de control i administració de xarxa: les consultes DNS permeten resoldre noms de domini, mentre que les connexions SSH generen un procés de negociació criptogràfica propi del protocol Secure Shell.

Com a resultat, l'escenari experimental va produir aproximadament 5.700 connexions HTTPS, 900 consultes DNS i 300 intents de connexió SSH. Aquesta combinació va permetre construir una classe de trànsit xifrat heterogènia, formada per navegació web, accés a APIs, serveis multimèdia i protocols de gestió de xarxa, proporcionant una base de comparació adequada per avaluar la capacitat dels models de machine learning de diferenciar el trànsit Bitcoin xifrat de la resta de comunicacions protegides.

3.5 Captura de dades

La recol·lecció del trànsit de xarxa en brut es va fer mitjançant la interceptació passiva dels paquets a les interfícies de xarxa virtuals dels contenidors, generant fitxers de captures en format PCAP mitjançant la utilització de l'eina de línia de comandes `tcpdump` [18].

En l'etapa inicial amb Polar, la captura es va dur a terme sobre el contenidor aïllat del *Backend2* fent ús d'un contenidor de diagnòstic de xarxa adjunt (*netshoot*) mitjançant la següent instrucció específica: **`docker run --rm --network container:polar-n3-backend2/netshoot tcpdump -i eth0 -w - --tee bip324.pcap`**

Aquesta captura pilot va registrar una mostra petita però controlada de 106 paquets de tipus RPC i ZMQ vinculats al node.

Posteriorment, en la fase definitiva amb Warnet, es va desenvolupar un sistema de captura massiva automatitzat

instal·lat directament sobre els nodes de Kubernetes. El sistema es va mantenir actiu i recopilant dades de manera ininterrompuda durant aproximadament 30 minuts continus d'execució. La metodologia va consistir a enregistrar tot el tràfic TCP dels nodes executant de manera simultània processos de mineria i propagació de blocs, emmagatzemant els paquets en fitxers `.pcap` independents i segregant clarament el tràfic de Bitcoin d'altres fluxos de la xarxa.

4 CONSTRUCCION DEL MODEL DE ML

4.1 Pipeline

Avaluació empírica de models d'aprenentatge automàtic supervisat, com ara *Random Forest* i màquines de vectors de suport (SVM), fent ús de la suite d'eines de Scikit-learn [9, 11]. Els models s'entrenen per resoldre un problema de classificació binària (BIP324 vs. HTTPS), optimitzant els hiperparàmetres mitjançant tècniques de validació creuada i mesurant la seva robustesa amb mètriques de rendiment estàndard com l'*F1-score* [8].

Cada iteració culmina amb un sprint review: anàlisi de logs Kubernetes, visualització de PCAPs (Wireshark) i proves de model, ajustant hiperparàmetres o topologies de xarxa segons resultats [6,17].

4.2 Preprocessament i extracció de característiques

Un cop recollits els fitxers de captures en brut, es va dissenyar un pipeline de processament de dades en Python utilitzant les llibreries Scapy [36] i Pandas [37] per transformar els arxius PCAP en datasets estructurats aptes per a l'entrenament computacional. Les dades es van tractar de manera agnòstica al contingut dels paquets, centrant l'atenció únicament en les relacions estadístiques residuals. El procediment es va dividir en quatre etapes conceptuals:

1. **Lectura i filtratge:** Es realitza una lectura seqüencial paquet per paquet validant la presència i integritat de les capes de xarxa IP i de transport TCP. Es descarten de manera sistemàtica tots els protocols analitzats com a soroll irrelevant per a la recerca (com ara UDP, ICMP o ARP).
2. **Extracció de variables primàries:** Per a cada paquet vàlid s'extreuen els camps base estructurals: la longitud total del paquet (`len`), el temps absolut d'arribada (`timestamp`) utilitzat com a fonament per al càlcul de temps posteriors, els flags de control TCP i la mida de la finestra de recepció (`window`).
3. **Càlcul d'interval temporal:** Es calcula l'interval de temps d'arribada entre paquets consecutius (*Inter-Arrival Time* o IAT), una variable altament discriminant en l'anàlisi de comportament de xarxa.

4. **Enginyeria de fluxos avançada:** En lloc d'avaluar paquets de manera aïllada, les mostres es van agrupar per identificadors de flux TCP (*flow id*) per calcular mètriques estadístiques mitjançant finestres mòbils [8]. Per a la selecció final de característiques es va utilitzar l'algorisme d'eliminació recursiva amb validació creuada (*RFECV*), reduint la dimensionalitat a 4 variables clau d'alta rellevància temporal i de volum:

- `iat_rolling_mean`: Mitjana mòbil de l'IAT, que mesura la cadència del flux.
- `iat_rolling_std`: Desviació estàndard de l'IAT, que mesura la regularitat temporal.
- `len_rolling_mean`: Mitjana mòbil de la longitud en bytes de la ràfega de paquets.
- `len_rolling_std`: Desviació estàndard de les longituds, indicant la variabilitat de les mides.

4.2.1 Escenari experimental

L'escenari s'ha dissenyat per emular l'activitat d'un node dins d'un entorn on un observador intermedi passiu (com un proveïdor de serveis d'Internet o una entitat de vigilància) aplica tècniques d'inspecció profunda de paquets (*DPI*) per identificar quins fluxos xifrats de la xarxa corresponen a l'ecosistema Bitcoin P2P v2. Per fer-ho, el problema s'enfronta com una classificació binària on s'avalua el tràfic de l'aplicació Bitcoin enfront d'un tràfic de referència d'ús comú.

Mètrica de flux	Tràfic Bitcoin (BIP324)	Tràfic de referència (HTTPS/SSH)
Paquets totals	30.754	64.503
Fluxos TCP únics	1.677	5.706
Entorn de generació	Bitcoin Core / Wintern	Navegació web i administració
Durada de la captura	30 minuts	15 hores

TAULA 1: VOLUM DE DADES ESTADÍSTIQUES EXTRETES EN L'EXPERIMENTACIÓ AMB WINTERN.

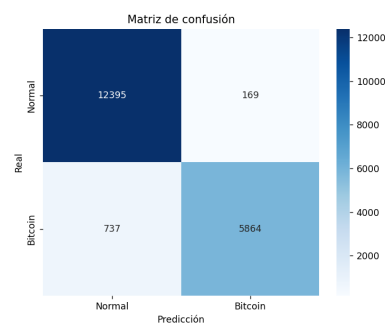


Fig. 1: Matriu de confusió del model de classificació binària.

4.3 Entrenament i validació del model

La fase de modelització es va dur a terme mitjançant un algorisme de classificació supervisada basat en **Random Forest** (*Bosc Aleatori*), implementat amb la llibreria Scikit-learn [11]. Aquest algorisme construeix un conjunt d'arbres de decisió independents que aprenen diferents particions de l'espai de característiques i combinen posteriorment

les seves prediccions mitjançant votació majoritària. La seva elecció es va fonamentar en la seva capacitat per modelar relacions no lineals, la seva robustesa davant dades sorolloses i la seva resistència al sobreajustament (*overfitting*), característiques especialment adequades per a l'anàlisi de trànsit de xarxa xifrat [9].

El conjunt de dades resultant de les captures Bitcoin i HTTPS es va unificar en un únic repositori etiquetat i es va dividir en dues particions independents: un 80% dels fluxos es va destinar a l'entrenament i el 20% restant a la validació final. A diferència d'una partició aleatòria convencional, la divisió es va realitzar mitjançant l'estratègia *GroupShuffleSplit*, utilitzant l'identificador de flux TCP (*flow id*) com a criteri d'agrupació. Aquesta decisió metodològica evita el fenomen conegut com a *data leakage*, ja que impedeix que paquets pertanyents a una mateixa connexió apareguin simultàniament en els conjunts d'entrenament i prova. D'aquesta manera, el model és avaluat sobre fluxos completament nous i desconeguts, reproduint un escenari més proper a les condicions reals d'operació.

Sobre el conjunt d'entrenament es va aplicar un procés de selecció automàtica de característiques mitjançant l'algorisme **Recursive Feature Elimination with Cross Validation** (RFECV). Aquest mètode elimina iterativament les variables menys rellevants i avalua l'impacte de cada reducció sobre el rendiment del model. Per garantir la independència entre mostres durant aquest procés, la validació creuada es va implementar mitjançant **GroupKFold**, mantenint tots els paquets d'un mateix flux dins del mateix grup de validació. Aquesta estratègia permet obtenir una estimació més fiable de la capacitat de generalització del model i minimitza el risc d'obtenir resultats excessivament optimistes [8].

La funció d'avaluació emprada durant el procés RFECV va ser la mètrica F1 macro, seleccionada per la seva capacitat per equilibrar precisió i exhaustivitat entre les dues classes analitzades. Com a resultat del procés de selecció es van conservar únicament les característiques amb major capacitat discriminant, reduint la dimensionalitat del problema i eliminant informació redundant o poc rellevant.

Finalment, es va entrenar un model *Random Forest* format per 200 arbres de decisió sobre les característiques seleccionades. El rendiment es va avaluar sobre el conjunt de prova independent mitjançant diverses mètriques complementàries: exactitud global (*accuracy*), precisió, exhaustivitat (*recall*), puntuació F1, matriu de confusió i àrea sota la corba ROC (ROC-AUC). Aquesta combinació de mesures permet analitzar tant la capacitat global de classificació com el comportament del sistema davant errors de falsos positius i falsos negatius, proporcionant una validació robusta de la capacitat del model per distingir trànsit Bitcoin xifrat de trànsit HTTPS convencional.

4.4 Discussió dels resultats i importància de les variables

Els assajos experimentals completats aporten conclusions d'alt valor per al TFG, revelant que la suite d'intel·ligència

és capaç de segregar els fluxos amb una precisió molt elevada malgrat la presència de xifratge. Les mètriques de rendiment recollides en l'experimentació final amb la xarxa massiva de Warnet es desglossen a la següent taula:

Mètrica de rendiment	Classe 0 (HTTPS)	Classe 1 (BIP324)	Rendiment Global
Precision	100%	97%	—
Recall	—	89%	—
F1-Score	—	93%	—
Accuracy	—	—	95,3%
ROC-AUC	—	—	0,980

TAULA 2: MÈTRIQES DE CLASSIFICACIÓ FINALS DEL MODEL RANDOM FOREST (FASE WARNET).

A l'apèndix 6.3 es troba una imatge de la corba ROC i mètrica AUC.

L'obtenció d'una *Accuracy* del 95,3% i un valor de l'àrea sota la corba *ROC-AUC* de 0,980 demostren empíricament una capacitat excel·lent de separació de classes independentment del llindar de decisió aplicat. L'anàlisi detallada de la matriu de confusió confirma aquest escenari, registrant 12.395 paquets HTTPS catalogats correctament i 5.864 paquets Bitcoin validats de manera exacta. L'algorisme es mostra moderat a l'hora d'etiquetar tràfic com a Bitcoin, reportant només 169 falsos positius i 737 falsos negatius; el fet d'assolir una *precision* del 97% per al protocol BIP324 indica que quan el model afirma que un corrent de dades pertany a Bitcoin, la probabilitat d'encert és gairebé absoluta.

La investigació de la importància de les variables va exposar dos comportaments estructurals altament rellevants que representen la troballa central de la memòria:

- **El biaix de l'entorn de control a la fase pilot de Polar:** Durant les primeres proves realitzades en l'entorn simplificat amb Polar, el model va donar uns resultats extraordinaris del 100% de rendiment a causa del paràmetre *TCPWindowSize*. L'algorisme va identificar immediatament que el node de Bitcoin Core, en executar-se sota els contenidors de Docker en entorn Regtest, mantenia un valor de finestra de recepció fix o confinat a rangs estàtics molt estrets (habitualment 64 KB, valor optimitzat al nucli de Linux per a missatges petits i ràpids). En canvi, els navegadors web del sistema amfitrió feien servir mecanismes d'escalat dinàmic (*Window Scaling*) amb valors molt més alts i variables, dissenyats per a la baixada massiva de dades. Aquest fenomen demostra que el model no estava discriminant el xifratge de l'aplicació, sinó fent un *fingerprinting* de la pila TCP del sistema operatiu amfitrió (*TCP stack fingerprinting*) [12].
- **Predomini de la dinàmica de ràfegues en la fase massiva de Warnet:** En corregir el biaix anterior gràcies a l'entorn distribuït de Warnet i en aplicar la selecció de variables mitjançant *RFECV*, l'algorisme va prioritzar les característiques estadístiques temporals i de volum sobre els paquets individuals. Les dues variables més determinants van ser *len_rolling_std* (36,70%) i *len_rolling_mean* (26,98%). Aquesta priorització respon a la naturalesa arquitectònica del

programari: Bitcoin Core manté una periodicitat i estabilitat estadística molt marcada durant la propagació i sincronització de blocs entre *peers*, mentre que les connexions HTTPS exhibeixen un perfil temporal molt més caòtic i asimètric condicionat per la interacció humana i la descàrrega d'elements web.

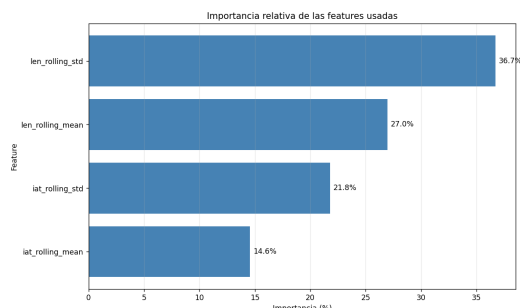


Fig. 2: Importància rellevant de les característiques en la classificació del tràfic.

En resum, els resultats indiquen que, tot i que BIP324 compleix el seu objectiu d'ocultar el contingut del payload, amagar les longituds inicials i eliminar signatures clàssiques de xarxa per evitar fingerprinting immediat, el trànsit continua depenent del transport TCP. Aquesta dependència genera metadades, asimetries de volum i patrons temporals que permeten als models de machine learning identificar el trànsit Bitcoin amb alta precisió.

4.4.1 Anàlisi tècnic del resultat

La Figura 2 mostra la importància de les característiques utilitzades pel model Random Forest per identificar el trànsit generat amb BIP324. Els resultats indiquen que les variables més rellevants són les relacionades amb la grandària dels paquets. En particular, la desviació estàndard mòbil de la longitud dels paquets (*len_rolling_std*) aporta un 36,7 % de la capacitat de decisió del model, mentre que la mitjana mòbil (*len_rolling_mean*) representa un 27,0 %. En conjunt, aquestes dues variables concentren més del 63 % de la importància total.

Les característiques temporals tenen un pes menor, però continuen sent significatives. La desviació estàndard dels temps entre arribades (*iat_rolling_std*) aporta un 21,8 % i la mitjana (*iat_rolling_mean*) un 14,6 %.

Aquests resultats indiquen que, tot i que BIP324 xifra el contingut dels missatges i dificulta la identificació del protocol mitjançant signatures estàtiques, encara existeixen patrons estadístics observables en el trànsit. Les mides dels paquets continuen presentant determinades regularitats derivades del comportament de la xarxa Bitcoin, com la propagació de transaccions o de blocs, i aquestes poden ser capturades pel model d'aprenentatge automàtic.

De manera similar, BIP324 no modifica significativament el comportament temporal de les comunicacions. Per això, els temps entre arribades dels paquets continuen reflectint

patrons propis de la xarxa Bitcoin que poden ser utilitzats com a criteri de classificació.

Els mecanismes de dissimulació addicionals del protocol, com els garbage bytes i els decoy packets, tampoc eliminen completament aquestes característiques. Els primers només s'utilitzen durant l'establiment de la connexió, i els segons no introdueixen prou soroll per ocultar els patrons estadístics presents en les captures analitzades.

En conjunt, els resultats mostren que BIP324 representa una millora important en termes de privacitat i protecció davant la inspecció del contingut dels paquets, però no impedeix completament les tècniques de fingerprinting basades en metadades. El fet que el model sigui capaç de classificar el trànsit utilitzant únicament la grandària dels paquets i els temps entre arribades demostra que encara persisteixen característiques observables del comportament de la xarxa Bitcoin.

5 CONCLUSIONS

Aquest treball ha demostrat la viabilitat d'identificar trànsit Bitcoin xifrat a partir de característiques estadístiques de nivell de xarxa, fins i tot quan el contingut de les comunicacions no és accessible. Mitjançant la construcció d'un entorn experimental controlat basat en Warnet i Bitcoin Core, s'ha pogut generar i capturar trànsit representatiu tant del protocol Bitcoin P2P v2 amb xifrat BIP324 com d'altres protocols xifrats habituals a Internet, principalment HTTPS/TLS.

La metodologia desenvolupada ha permès transformar captures de trànsit en brut en un conjunt estructurat de característiques estadístiques relacionades amb la temporalitat i la mida dels paquets. Els resultats obtinguts confirmen que aquestes propietats continuen contenint informació suficient per diferenciar comportaments de xarxa, fins i tot quan el contingut de les comunicacions està completament protegit mitjançant mecanismes criptogràfics.

L'ús de tècniques de selecció automàtica de característiques ha permès identificar que les variables més rellevants per a la classificació estan associades principalment als patrons temporals d'arribada de paquets i a la variabilitat de les seves longituds. Aquest resultat és coherent amb la naturalesa del protocol Bitcoin, que presenta dinàmiques de propagació i sincronització diferents de les observades en serveis web convencionals.

Pel que fa a la fase de classificació, el model Random Forest ha mostrat una capacitat notable per distingir entre trànsit Bitcoin i trànsit HTTPS, obtenint resultats satisfactoris en les mètriques d'avaluació utilitzades. L'ús de particions basades en fluxos TCP i de mecanismes de validació agrupada ha permès evitar situacions de *data leakage*, garantint que les mètriques obtingudes reflecteixin de manera més fidel la capacitat real de generalització del sistema.

Des d'una perspectiva més àmplia, els resultats evidencien una limitació inherent dels sistemes de xifrat orientats exclusivament a protegir el contingut de les comunicacions.

Encara que protocols com HTTPS o BIP324 impedeixen la inspecció directa de les dades transmeses, continuen existint metadades observables que poden ser explotades mitjançant tècniques modernes d'anàlisi estadística i aprenentatge automàtic.

Finalment, aquest treball obre diverses línies de recerca futura. Entre les més rellevants destaquen l'ampliació del conjunt de protocols analitzats, la incorporació de tècniques de *Deep Learning*, l'estudi de trànsit Bitcoin en entorns més heterogenis i distribuïts, així com l'avaluació de possibles contramesures orientades a reduir la capacitat de classificació basada en metadades. Aquestes línies podrien contribuir tant a millorar els sistemes de monitoratge de xarxa com a reforçar els mecanismes de privacitat dels protocols descentralitzats del futur.

AGRAÏMENTS

M'agradaria dedicar unes paraules d'agraïment a totes les persones que m'han acompanyat durant aquest camí i que, d'una manera o altra, han contribuït a fer possible la realització d'aquest treball.

En primer lloc, a la meva tutora, per la seva orientació, disponibilitat i suport al llarg del desenvolupament d'aquest TFG. Gràcies pels seus consells, les correccions i per ajudar-me a enfocar el treball amb rigor i criteri acadèmic.

Als meus pares, pel seu suport incondicional durant tota la meva formació i per haver-me transmès el valor de la constància, el compromís i l'esforç. Sense la seva confiança, aquest objectiu no hauria estat possible.

Als meus germans, Nacho i Cris, per ser sempre al meu costat, pels consells i pel suport que m'han ofert al llarg d'aquest recorregut.

A la meva parella, Ari, per la seva paciència, comprensió i suport durant els moments més exigents d'aquest projecte. Gràcies per animar-me, acompanyar-me i compartir cada pas d'aquest camí.

Finalment, a totes les persones que han contribuït, directament o indirectament, al meu creixement personal i professional durant aquests anys.

A tots vosaltres, moltes gràcies.

6 REFERÈNCIES

REFERÈNCIES

- [1] Bitcoin Dev Project. *Warnet: Bitcoin Network Simulation on Kubernetes*. Disponible a: <https://github.com/bitcoin-dev-project/warnet>. Últim accés: 18 febrer 2026.
- [2] Phoura, J. i Bitcoin Core Developers. *BIP-0324: Version 2 P2P Encrypted Transport Protocol*. Disponible a: <https://github.com/bitcoin/bips/blob/master/bip-0324.mediawiki>. Últim accés: 20 febrer 2026.
- [3] CriptoNoticias. *Què és la BIP-324 i com soluciona els riscos d'atac a Bitcoin*. Disponible a: <https://www.criptonoticias.com/tecnologia/version-2-protocolo-p2p-bitcoin-encripta-mens>. Últim accés: 22 febrer 2026.
- [4] The Kubernetes Authors. *Kubernetes Documentation*. Disponible a: <https://kubernetes.io/docs/home/>. Últim accés: 24 febrer 2026.
- [5] Bitcoin StackExchange. *Is Bitcoin BIP-324 v2 P2P transport distinguishable from random traffic?*. Disponible a: <https://bitcoin.stackexchange.com/questions/130500/is-bitcoin-bip-324-v2-p2p-transport-distinguishable>. Últim accés: 27 febrer 2026.
- [6] Bitcoin Dev Project. *Warnet Testing Tutorial*. YouTube. Disponible a: <https://www.youtube.com/watch?v=PTkJTYIK3Bc>. Últim accés: 2 març 2026.
- [7] Bitcoin Dev Project. *Warnet DESIGN.md: Kubernetes-based Bitcoin network simulation*. Disponible a: <https://github.com/bitcoin-dev-project/warnet/blob/main/DESIGN.md>. Últim accés: 4 març 2026.
- [8] Aceto, G., Ciuonzo, D., Montieri, A. i Pescapé, A. (2024). "Encrypted Network Traffic Analysis and Classification: A Survey". *IEEE Communications Surveys & Tutorials*. Últim accés: 14 abril 2026.
- [9] Breiman, L. (2001). "Random Forests". *Machine Learning*, 45(1), pp. 5–32.
- [10] Hastie, T., Tibshirani, R. i Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.
- [11] Scikit-learn Developers. *RandomForestClassifier Documentation*. Disponible a: <https://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees>. Últim accés: 22 abril 2026.
- [12] Postel, J. (1981). *Transmission Control Protocol*. RFC 793. Internet Engineering Task Force (IETF). Disponible a: <https://datatracker.ietf.org/doc/html/rfc793>. Últim accés: 11 març 2026.
- [13] Wireshark Foundation. *TCP Window Size*. Disponible a: <https://wiki.wireshark.org/TCP%2FWindow%2FSize>. Últim accés: 19 abril 2026.
- [14] Bitcoin StackExchange. *Differences in Bitcoin Network Traffic*. Disponible a: <https://bitcoin.stackexchange.com/>. Últim accés: 16 abril 2026.

- [15] Polar. *Lightning Network Visualization and Management*. Disponible a: <https://lightningpolar.com/>. Últim accés: 9 març 2026.
- [16] Wireshark Foundation. *Wireshark Network Protocol Analyzer*. Disponible a: <https://www.wireshark.org/>. Últim accés: 12 març 2026.
- [17] Auvik Networks. *Understanding TCP Window Size*. Disponible a: <https://www.auvik.com/franklyit/blog/tcp-window-size/>. Últim accés: 20 abril 2026.
- [18] Tcpdump Group. *Tcpdump and Libpcap*. Disponible a: <https://www.tcpdump.org/>. Últim accés: 15 març 2026.
- [19] River Financial. *What is a Bitcoin Block Reward?*. Disponible a: <https://river.com/learn/terms/b/block-reward/>. Últim accés: 18 març 2026.
- [20] IBM. *Random Forest*. Disponible a: <https://www.ibm.com/mx-es/think/topics/random-forest>. Últim accés: 25 abril 2026.
- [21] Bitcoin Core Developers. *Bitcoin P2P v2 Transport - BIP324 Overview*. YouTube, BitcoinCoreDevTalks. Últim accés: 8 maig 2026.
- [22] Antonopoulos, A. M. (2023). *Bitcoin Improvement Proposals (BIPs) Explained*. YouTube. Últim accés: 10 maig 2026.
- [23] Bitcoin Magazine. (2024). *Bitcoin Privacy Upgrades: BIP324 + Taproot*. YouTube. Últim accés: 11 maig 2026.
- [24] Bitcoin Core Team. (2024). *Bitcoin Core v27 - What's New (BIP324)*. YouTube. Últim accés: 12 maig 2026.
- [25] Antonopoulos, A. M. (2023). *"Bitcoin Improvement Proposals (BIPs) Explained"*. [Video]. YouTube. https://www.youtube.com/watch?v=K_vXSetu-Mw.
- [26] Wikipedia. *Agile software development*. Disponible a: https://en.wikipedia.org/wiki/Agile_software_development. Últim accés: 22 maig 2026.
- [27] Bitcoin Developer Reference. *P2P Network Protocol Overview - Message structures*. Disponible a: https://developer.bitcoin.org/reference/p2p_networking.html#p2p-network. Vegeu la secció: *"P2P Network"*.
- [28] Introduction. *Introduction*. Disponible a: <https://bitcoindevelopers.org/docs/developer-guides/p2p-network/>. Vegeu la secció: *Introduction*.
- [29] Corallo, M. i Fields, D. *Bitcoin Improvement Proposal 324: Peer-to-Peer Transport v2*. Disponible a: <https://github.com/bitcoin/bips/blob/master/bip-0324.mediawiki>. Vegeu les seccions: *"Motivation"*, *"Handshake Pseudorandomness (ElligatorSwift)"* i *"Packet Structure"*.
- [30] Internet Engineering Task Force (IETF). *RFC 9110: HTTP Semantics and Encryption over TLS*. Disponible a: <https://datatracker.ietf.org/doc/html/rfc9110>. Vegeu la secció: *Secció 4: "Message Routing and Connection Management"*.
- [31] Internet Engineering Task Force (IETF). *RFC 4253: The Secure Shell (SSH) Transport Layer Protocol*. Disponible a: <https://datatracker.ietf.org/doc/html/rfc4253>. Vegeu la secció: *Secció 6: "Binary Packet Protocol"*.
- [32] Bitcoin Dev Project (GitHub). *Warnet Architecture: Simulating P2P Networks in Kubernetes*. Disponible a: <https://github.com/bitcoin-dev-project/warnet/blob/main/docs/architecture.md>. Vegeu la secció: *Secció 3: "Network Topology and Scenario Injection"*.
- [33] Informació ChaCha20-poly1305 de Wikipedia Disponible a: <https://en.wikipedia.org/wiki/ChaCha20-Poly1305>. Última consulta dia 25 de Juny 2026
- [34] Cloudflare. *What is MSS (maximum segment size)?* Disponible en: <https://www.cloudflare.com/learning/network-layer/what-is-mss/> Secció: *What is MSS (maximum segment size)?*
- [35] CS 168 Textbook. *TCP Implementation* Disponible en: <https://textbook.cs168.io/transport/tcp-implementation.html> Secció: *TCP Implementation*
- [36] Scapy Project. *Scapy Documentation*. Disponible a: <https://scapy.readthedocs.io/>. Últim accés: 22 maig 2026.
- [37] The Pandas Development Team. *Pandas Documentation*. Disponible a: <https://pandas.pydata.org/docs/>. Últim accés: 22 maig 2026.
- [38] Disponible a: <https://paritytech.github.io/try-runtime-cli/secp256k1/ellswift/struct.ElligatorSwift.html>. Últim accés: 25 juny 2026
- [39] Disponible a: <https://en.bitcoin.it/wiki/BIP%0324?#Transport%20layer%20specification>. Últim accés: 25 juny 2026
- [40] Disponible a: <https://unprotect.it/technique/inserting-garbage-bytes/> Últim accés: 25 de Juny 2026
- [41] Disponible a: <https://lacryptoguia.com/glossary/decoy-transaction/> Últim accés: 25 de Juny 2026

6.1 Secció d'Apèndix

Protocol	Tècnica principal
BIP324 (Bitcoin)	ElligatorSwift i bytes aleatoris
Tor (Obfs4)	Padding i ofuscació de paquets
Shadowsocks	Xifratge AEAD i reorganització de paquets
WireGuard	Protocol criptogràfic Noise

TAULA 3: EXEMPLES DE TÈCNiques ANTI-FINGERPRINTING EN PROTOCOLS MODERNS

6.2 Secció d'Apèndix

Capa	Eina
Infraestructura	Docker/Kubernetes/Minikube
Bitcoin	Bitcoin Core v27+ (BIP324)
Captura	tcpdump/Wireshark
ML	Python 3 (pandas/scikit-learn)

TAULA 4: EINES I TECNOLOGIES UTILITZADES

Capa d'Infraestructura: La simulació precisa d'una xarxa de nodes distribuïts requereix un grau elevat d'aïllament, control estricte de recursos i capacitat de reproducció experimental.

- **Docker:** S'utilitza com a motor de contenització per a empaquetar de forma aïllada les imatges de Bitcoin Core. Això garanteix que cada node s'executi en el seu propi espai d'usuari, compartint el nucli del sistema operatiu però evitant conflictes de dependències o configuració.
- **Kubernetes i Minikube:** Com que la gestió manual de desenes de contenidors Docker és inviable, s'empra Kubernetes per a l'orquestració automatitzada. Aquest sistema s'encarrega de desplegar, escalar i gestionar el cicle de vida dels nodes, així com de definir la topologia de xarxa i els canals de comunicació interns. **Minikube** actua com a l'entorn local que permet executar aquest clúster de Kubernetes en una única màquina física de desenvolupament.

Aquesta elecció arquitectònica respon a la necessitat d'establir un entorn de simulació escalable, aïllat i reproduïble basat en microserveis [4], seguint de prop la línia de projectes de referència en el sector com Warnet [1].

Capa Bitcoin: El nucli de l'experiment es fonamenta en la branca oficial de **Bitcoin Core v27+**, la versió de referència de la xarxa que incorpora de manera nativa el suport per al transport P2P v2 definit a la proposta **BIP324**.

- **Bitcoin Core (v27+):** Executa la lògica de consens, valida les transaccions i els blocs, i manté el llibre de

comptabilitat (*ledger*). En aquest projecte, actua com el generador del comportament real de la xarxa.

- **BIP324 (P2P v2):** És el protocol de transport xifrat objecte d'estudi. Aquesta implementació permet que els nodes negociïn connexions xifrades de manera totalment transparent per a l'operador del sistema, protegint la topologia de la xarxa contra observadors passius [2]. L'objectiu d'incloure aquesta capa és generar un entorn idèntic al de la xarxa principal (*mainnet*) per a poder analitzar si el xifratge és vulnerable a l'anàlisi de trànsit.

Capa de Captura: Dissenyada per a la recollida del trànsit de xarxa de manera no intrusiva, garantint que el comportament i la latència dels nodes no es vegin alterats durant l'experiment.

- **tcpdump:** És una eina de línia de comandes que s'executa directament dins de l'entorn dels nodes de Kubernetes. S'encarrega d'interceptar i registrar els paquets de dades de les interfícies de xarxa virtuals en temps real. S'ha triat per la seva lleugeresa, el seu baix consum de recursos i la seva integració nativa amb el nucli de Linux [18], fet que minimitza l'impacte sobre el rendiment de la simulació.
- **Wireshark (i l'ecosistema tshark):** S'utilitza en la fase posterior a la captura per a la inspecció en profunditat i la validació estructural dels fitxers de traça generats (PCAP) [16]. Permet desglossar els paquets a nivell de bit i verificar de forma analítica que les signatures d'ofuscació i xifratge de la capa BIP324 s'estan aplicant correctament abans d'enviar les dades als models de classificació.

Capa d'Aprenentatge Automàtic (ML): Aquesta capa processa les dades en brut obtingudes de la captura de xarxa per a extreure-ne patrons i entrenar els models de classificació. Tot l'ecosistema s'articula al voltant de **Python 3** a causa del seu estàndard de facto en la ciència de dades.

- **Pandas:** És la llibreria encarregada de la manipulació de dades. Transforma els fitxers PCAP ja processats en estructures tabulars (*DataFrames*). S'utilitza per a les tasques de neteja, tractament de valors nuls, normalització a escala i enginyeria de característiques (*feature engineering*) (com càlcul de mides de paquet, temps inter-arribada, etc.), seguint els estàndards de la literatura d'*Encrypted Traffic Analysis* (ETA) [8].
- **Scikit-learn:** És la biblioteca utilitzada per a la fase de modelització. Proporciona les eines per a dividir les dades en conjunts d'entrenament i prova, aplicar algorismes de classificació supervisada (com *Random Forest* o *Gradient Boosting*) i extreure les mètriques d'avaluació (precisió, recall, F1-score i la corba ROC) [11]. Es tria per la seva robustesa i eficiència en la implementació d'algorismes clàssics basats en arbres de decisió.

6.3 Secció d'Apèndix

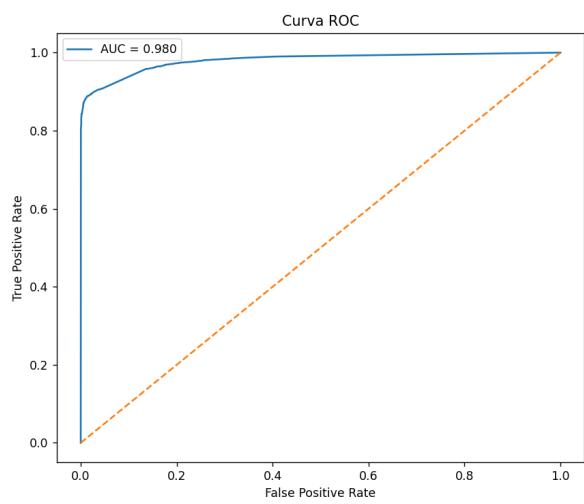


Fig. 3: Corba ROC i mètrica AUC assolida en l'entorn experimental.

6.4 Secció d'Apèndix

- **ElligatorSwift:** ElligatorSwift és un mètode criptogràfic que permet representar claus públiques de corbes el·líptiques com si fossin seqüències de bytes completament aleatòries.

Normalment, les claus públiques de Bitcoin tenen una estructura reconeixible. Un sistema de monitorització de xarxa pot detectar fàcilment aquestes estructures i deduir que hi ha una connexió Bitcoin. ElligatorSwift evita aquest problema fent que les claus intercanviades durant el handshake semblin dades aleatòries sense cap patró identificable. [29].

El funcionament consisteix en que un node genera la clau pública normal de la corba el·líptica secp256k1. En lloc d'enviar-la en el format habitual (que té una estructura reconeixible), la codifica en 64 bytes que semblen completament aleatoris. L'altre node rep aquests 64 bytes i pot recuperar la clau pública original. Com que les dades semblen aleatòries, un sistema de vigilància o un ISP no pot reconèixer-ho fàcilment. [38].

- **ChaCha20-Poly1305:** ChaCha20-Poly1305 és un esquema de xifratge autènticat amb dades associades (AEAD, Authenticated Encryption with Associated Data) que combina l'algorisme de xifratge de flux ChaCha20, desenvolupat per Daniel J. Bernstein, amb el codi d'autenticació de missatges Poly1305. El seu principal objectiu és proporcionar simultàniament confidencialitat, integritat i autenticitat en les comunicacions.

Una vegada que el procés de handshake s'ha completat amb èxit mitjançant ElligatorSwift, tots dos nodes han establert una clau simètrica secreta compartida. A partir d'aquest moment, tot el trànsit intercanviat entre ells, incloent-hi missatges relacionats amb transaccions (tx) i blocs (block), es xifra utilitzant ChaCha20.

D'una banda, ChaCha20 garanteix la confidencialitat de la comunicació, ja que impedeix que tercers puguin llegir el contingut dels missatges transmesos. D'altra banda, Poly1305 s'encarrega de garantir la integritat i autenticitat de les dades mitjançant la generació d'una etiqueta criptogràfica associada a cada paquet enviat. D'aquesta manera, si un atacant intentés modificar qualsevol part del missatge durant la transmissió, encara que fos un únic bit, el receptor detectaria immediatament l'alteració perquè l'etiqueta d'autenticació deixaria de ser vàlida, descartant el missatge o fins i tot la connexió. [33].

- **Ocultació de longituds:** L'ocultació de longituds és un mecanisme introduït per BIP324 amb l'objectiu de dificultar l'anàlisi del trànsit de la xarxa Bitcoin. En la versió anterior del protocol P2P, la longitud de cada missatge era visible, la qual cosa permetia a un observador extern identificar fàcilment on començava i acabava cada paquet. A partir d'aquesta informació, era possible inferir el tipus de missatges intercanviats i fins i tot reconstruir parcialment la comunicació entre els nodes.

Per a evitar aquest problema, BIP324 incorpora un sistema que oculta les fronteres dels missatges, de manera que un tercer que capturi el trànsit de xarxa no pot determinar amb facilitat la grandària real de cada missatge ni separar clarament uns paquets d'uns altres. Encara que el contingut dels missatges ja es troba xifratge mitjançant ChaCha20-Poly1305, l'ocultació de longituds afegeix una capa addicional de protecció enfront de les tècniques d'anàlisi de trànsit.

Aquest mecanisme resulta especialment important perquè, en molts casos, el simple fet de conèixer la grandària i la seqüència dels paquets pot revelar informació significativa sobre el protocol utilitzat o sobre les operacions que s'estan duent a terme. En ocultar aquesta informació, BIP324 dificulta la identificació passiva del trànsit de Bitcoin, millora la privacitat dels nodes i augmenta la resistència de la xarxa enfront de sistemes de monitoratge i censura basats en l'anàlisi de patrons de comunicació. [39]

- **Garbage bytes:** Els garbage bytes és una tècnica d'ofuscació de codi que consisteix a introduir dades aleatòries o buides de significat dins del flux d'un programa. L'objectiu principal és dificultar les tasques d'enginyeria inversa i l'anàlisi estàtica del programari.

Aquesta pràctica actua com una mesura de distorsió de codi. En rebre aquests bytes binaris inesperats, els desassembladors lineals poden malinterpretar on comença o acaba una instrucció real, cosa que els porta a sincronitzar-se malament i a generar un codi d'assemblador completament erroni o deformat que no es correspon amb l'execució real. [40]

- **Decoy packets:** Els Decoy Packets són paquets cimbells introduïts per BIP324 que no contenen informació útil i són descartats pel receptor. El seu objectiu és alterar artificialment els patrons de trànsit de la connexió, dificultant que un observador extern pugui identificar el protocol utilitzat o inferir informació a partir de la

grandària, la freqüència o la seqüència dels paquets intercanviats. D'aquesta manera, contribueixen a millorar la privacitat i la resistència de la xarxa Bitcoin enfront de tècniques d'anàlisis de trànsit. Exactament funciona afegint trànsit que no correspon a missatges reals de Bitcoin. Des del punt de vista d'un observador extern, tots els paquets semblen similars i no pot saber quins contenen informació útil i quina és falsa. [41]

6.5 Secció d'Apèndix

- <https://www.google.com>: simulació d'accés a motors de cerca i serveis web generalistes.
- <https://api.github.com>: generació de peticions HTTPS cap a APIs REST utilitzades habitualment per aplicacions i eines de desenvolupament.
- <https://www.marca.com>: accés a contingut informatiu i pàgines de notícies.
- <https://www.youtube.com>: connexions dirigides a plataformes multimèdia i distribució de contingut audiovisual.
- <https://example.com>: domini de referència emprat per generar trànsit estable i controlat.