APLICACIÓ DE TÈCNIQUES DE MACHINE LEARNING A LA SEGURETAT

Aina Moncho Roig

01/2025

Director: Enric Hernández Jiménez
Màster en Ciberseguretat i Privadesa

Universitat Oberta de Catalunya

Treball de Fi de Màster

Aplicació de tècniques de Machine Learning a la Seguretat

Aina Moncho Roig

Màster en Ciberseguretat i Privadesa
Anàlisi de dades

Tutor de TF
Enric Hernández Jiménez
Professor responsable de l'assignatura
Enric Hernández Jiménez

07/01/2025

Universitat Oberta de Catalunya

AGRAÏMENTS

La meva gratitud més especial i sincera al meu tutor Enric Hernández, pel seu suport i per aconseguir guiar-me per fer realitat aquest treball. Gràcies a la seva rigorositat i al seu talent, m'ha ensenyat a aprendre i m'ha transmès la inquietud necessària per superar-me i millorar.

Al meu amic i parella Sergio, per aguantar-me en moments difícils i ajudar-me a mantenir sempre la il·lusió.

A tota la meva família en general i en particular als meus pares, Miquel i Núria, per educar-me en els valors de l'esforç i la superació. Gràcies sobretot a la meva estimada mare, per inculcar-me l'amor a l'estudi i especialment a la matemàtica i a la informàtica. Gràcies als meus germans Eloi i Marina, per haver estat sempre al meu costat recolzant-me i contagiant-me la seva alegria.

Gràcies a tots pel seu suport.

Títol del treball:	*Aplicació de tècniques de Machine Learning a la Seguretat*
Nom de l'autor:	*Aina Moncho Roig*
Nom del consultor/a:	*Enric Hernández Jiménez*
Nom del PRA:	*Enric Hernández Jiménez*
Data de lliurament (mm/aaaa):	*01/2025*
Titulació o programa:	*Màster en Ciberseguretat i Privadesa*
Àrea del Treball Final:	*Anàlisi de dades*
Idioma del treball:	*Català*
Paraules clau	*Machine Learning, Intrusions a la Xarca, Ciberatac*

Resum del Treball

L'objectiu principal d'aquest treball és desenvolupar una solució basada en tècniques de Machine Learning per a la detecció d'activitats sospitoses que podrien comprometre la seguretat informàtica d'una organització. Es volen identificar vulnerabilitats i atacs a la xarxa mitjançant l'anàlisi de dades de tràfic de xarxa.

El projecte segueix una metodologia d'investigació aplicada que inclou la selecció i processament de dades, l'entrenament de models de Machine Learning i l'avaluació del seu rendiment. Per a això, s'utilitza el conjunt de dades UNSW-NB15, que conté mostres de trànsit de xarxa normal i maliciós, i permet identificar atacs com Fuzzers, DoS i Worms, entre d'altres.

Es comparen diferents sistemes de classificació, tant basats en tècniques d'aprenentatge supervisat (Naive Bayes, Random Forest, Logistic Regression, SVM), com en tècniques d'aprenentatge no supervisades (K-Means i DBSCAN). També s'utilitzen models d'aprentatge profund per optimitzar les prediccions. Els resultats s'avaluen mitjançant mètriques com l'*accuracy*, la *precision*, la *recall* i la *f1-score*, per validar la capacitat de detectar amenaces.

En resum, aquest treball pretén oferir una solució eficient capaç de detectar vulnerabilitats i intrusions, aportant noves perspectives en la comparació de models de Machine Learning en ciberseguretat. Això permetrà millorar la seguretat digital, i augmentar la protecció contra intrusions i fugues de dades.

Abstract

The main objective of this work is to develop a solution based on Machine Learning techniques for detecting suspicious activities that could compromise an organization's cybersecurity. The goal is to identify vulnerabilities and network attacks through the analysis of network traffic data.

The project follows an applied research methodology that includes data selection and preprocessing, training of Machine Learning models, and evaluating their performance. For this purpose, the UNSW-NB15 dataset is used, which contains samples of both normal and malicious network traffic and enables the identification of attacks such as Fuzzers, DoS, and Worms, among others.

Different classification systems are compared, including those based on supervised learning techniques (Naive Bayes, Random Forest, Logistic Regression, SVM) and unsupervised learning techniques (K-Means and DBSCAN). Deep learning models are also utilized to optimize predictions. Results are evaluated using metrics such as *accuracy*, *precision*, *recall*, and F1-score to validate the ability to detect threats.

In summary, this work aims to provide an efficient solution capable of detecting vulnerabilities and intrusions, offering new perspectives on the comparison of Machine Learning models in cybersecurity. This will enhance digital security and increase protection against intrusions and data breaches.

TAULA DE CONTINGUT

LLISTA DE FIGURES

LLISTA DE TAULES

ABREVIATURES I ACRÒNIMS

ARGUS	Audit Record Generation and Utilization System	Audit Record Generation and Utilization System
BERT	Bidirectional Encoder Representations from Transformers	Representacions de Codificador per a Transformer Bidireccional
Bro-IDS	Bro Intrusion Detection System	Sistema de Detecció d'Intrusions Basat en Bro
CNN	Convolutional Neural Network	Xarxes neuronals convolucionals
DBSCAN	Density-Based Spatial Clustering of Applications with Noise	Clustering basat en densitat per a aplicacions amb soroll
DBN	Xarxa de Deep Belief Network	Xarxa de Creences Profunda
DoS	Denial of Service	Denegació de Servei
GDPR	General Data Protection Regulation	Reglament General de Protecció de Dades
GPT	Generative Pre-trained Transformer	Transformador generatiu pre-entrenat
IA	Artificial Intelligence	Intel·ligència Artificial
IDS	Intrusion Detection System	Sistema de detecció d'intrusions
KDD	Knowledge Discovery in Databases	Descobriment de Coneixement en Bases de Dades
LLM	Large Language Models	Models de Llenguatge Gran
LR	Logistic Regression	Regressió Logística
LSTM	Long Short-Term Memory	Memòria a llarg i curt termini
ML	Machine Learning	Aprenentatge Automàtic
NB	Classifier Naive Bayes	Classificador de Naive Bayes
PCA	Principal Component Analysis	Anàlisi de Components Principals
R2L	Remote to Local	Remot a Local

RBM	Restricted Boltzmann Machines	Màquina de Boltzmann Restringida
RNN	Recurrent Neural Networks	Xarxes Neuronals Recurrents
SVM	Support Vector Machines	Suport a les Màquines Vectorials
TF	Term Frequency	Freqüència de Termes
TF-IDF	Term Frequency - Inverse Document Frequency	Freqüència de Termes - Freqüència Inversa de Document
TFM		Treball Final de Màster
U2R	User to Root	Usuari a Root

1 INTRODUCCIÓ

Les tècniques desenvolupades en el context de la Intel·ligència Artificial i més concretament en l'aprenentatge automàtic han demostrat la seva efectivitat en diversos dominis. Aquest projecte se centra en l'estudi de les seves aplicacions en el camp de la seguretat.

1.1 Problema a resoldre

La finalitat d'aquest treball és desenvolupar una eina basada en Machine Learning que permeti detectar comportaments anòmals d'usuaris interns o identificar atacs a la xarxa dins d'una organització. El problema a resoldre en aquest context és la detecció d'aquests comportaments sospitosos, així com la identificació de possibles atacs.

Aquest repte pretén ser una aportació tant en la millora de la seguretat informàtica com en el desenvolupament d'eines per a la detecció de vulnerabilitats en entorns tecnològics. La detecció d'aquests incidents pot ajudar a prevenir situacions greus com el robatori de dades, el sabotatge o la fuga d'informació, garantint així una major seguretat en l'entorn tecnològic de l'organització.

1.2 Objectius del Treball

La finalitat d'aquest treball és construir una eina basada en Machine Learning que permeti detectar fugues de seguretat, com ara comportaments anòmals d'usuaris interns o atacs a la xarxa, i classificar-les segons nivell de risc. Aquest repte busca contribuir a la millora de la seguretat informàtica, proporcionant una solució que identifiqui activament possibles vulnerabilitats i amenaces dins d'un context digital.

Els objectius establerts per assolir aquesta fita són:

- Obtenir fonts de dades fiables que siguin rellevants per a l'entrenament dels models. Això inclou la identificació i recopilació de datasets adequats relacionats amb logs d'usuaris, logs de seguretat, tràfic a la xarxa o altres indicadors rellevants en la ciberseguretat.
- Aprofundir en el coneixement de diferents models de Machine Learning adients per analitzar i detectar intrusions.
- Entrenar els models seleccionats utilitzant la font de dades seleccionada.
- Avaluar i analitzar el rendiment dels models, comparant-los a través de mètriques com l'*accuracy*, *precision*, *recall* i *f1-score*, per tal d'assegurar la seva eficàcia en la detecció d'atacs o comportaments suspitosos.
- Comparar els resultats obtinguts tant entre si com amb altres estudis i sistemes existents, per validar la metodologia utilitzada i confirmar la fiabilitat i rendiment de la solució desenvolupada.

En resum, aquest treball pretén oferir una eina capaç de detectar fugues de seguretat de forma eficient, així com aportar noves perspectives en la comparació de models de Machine Learning dins del camp de la ciberseguretat.

1.3 Metodologia i estructura

Aquest treball es divideix en diverses etapes amb l'objectiu de desenvolupar una eina per a la detecció de comportaments anòmals o atacs a la xarxa, utilitzant tècniques de Machine Learning. Per assolir l'objectiu final, el treball s'estructurarà en diferents apartats i seguirà una metodologia clara que inclourà fases de recerca, anàlisi, implementació.

Definició del problema

- Identificar i formular clarament el problema a resoldre.
- Definir els principals objectius.

Revisió de l'estat de l'art

- Investigar estudis i treballs previs relacionats amb detecció i classificació d'intrusions, especialment utilitzant tècniques de Machine Learning i Intel·ligència Artificial.
- Recol·lectar informació sobre mètodes ja aplicats, mètriques utilitzades i tipus de dades emprades.
- Investigar els diferents models de Machine Learning o Aprenentatge profund que es poden utilitzar per a la detecció de comportaments anòmals o atacs, com ara Logistic Regression, Decision Trees, Random Forest, SVM, xarxes neuronals, etc.
- Valorar els avantatges i inconvenients de cadascun en el context del problema a resoldre.

Selecció de dades, definició del conjunt de dades i preprocessament

- Buscar datasets públics que continguin dades de logs d'usuaris, trànsit de xarxa o informes de programari maliciós.
- Un cop identificats els datasets, dur a terme una avaluació preliminar per determinar quins són els més adequats. Realitzar un anàlisi inicial de la mida del dataset, els tipus de característiques disponibles i la seva qualitat.
- Detectar i tractar valors nuls, duplicats, o dades incorrectes. Filtrar les dades irrelevants o sorolloses que no aportin informació útil pel model.
- Convertir les dades categòriques a formats numèrics utilitzant tècniques de vectorització com ara One-Hot Encoding, Label Encoding, TF o TF-IDF. Aplicar tècniques de normalització o escalat de dades per assegurar que les característiques estiguin a la mateixa escala.
- Dividir el conjunt de dades en els conjunts necessaris (entrenament i test o entrenament, validació i test segons el model).

Entrenament de models de Machine Learning

- Basat en la recerca prèvia, entrenar models de Machine Learning com ara la regressió logística, Random Forest, arbres de decisió i Support Vector Machines (SVM). O bé, realitzar un fine-tuning de models preentrenats com BERT o GPT, així com entrenar xarxes LSTM segons les dades disponibles.

- Ajustar els hiperparàmetres, provant diferents configuracions dels models per optimitzar-ne el rendiment i millorar els resultats.
- Utilitzar tècniques de validació creuada (k-fold cross-validation) per assegurar que els models no estan sobreajustant-se a les dades d'entrenament i tenen un bon rendiment generalitzat.

Avaluació dels models

- Avaluar els models utilitzant mètriques com *accuracy*, *precision*, *recall*, *f1-score*, i la matriu de confusió.
- Analitzar si els models són capaços de detectar efectivament comportaments anòmals a d'altres intrusions.
- Revisar els resultats per assegurar que no hi ha biaixos en la detecció de comportaments segons perfils específics d'usuaris (p. ex. determinades funcions dins de l'organització).
- Comparar els resultats entre els diferents models i amb estudis previs.

Redacció de conclusions i suggeriments per a treballs futurs

- Redactar les conclusions a partir dels resultats obtinguts, i identificar quins models han estat més eficients i per què.
- Proposar possibles projecte futurs.

Revisió final i presentació del TFM

- Revisar el contingut final del document, assegurant que manté una coherència adequada.
- Preparar la presentació final del TFM, destacant els resultats més rellevants i les aportacions del treball.

1.4 Planificació del Treball

La planificació del treball es distribuirà en les següents fases, amb els terminis previstos per assegurar l'èxit del projecte dins el temps assignat:

Fase 1 (PAC1): fins al 18/10

- Problema a resoldre: Definir el problema principal i la finalitat del treball.
- Objectius: Establir els objectius del projecte, com la identificació d'anomalies i l'avaluació de models.
- Planificació i Diagrama de Gantt: Crear un pla detallat i visualitzar-lo amb un diagrama de Gantt per organitzar les tasques i terminis del projecte.

Fase 2 (PAC2): del 19/10 al 05/11

- Estat de l'art: Revisar la literatura i estudis previs sobre la detecció d'anomalies i seguretat de la xarxa a partir de ML o IA.
- Dataset: Seleccionar dades adequades i aplicar les tècniques de preprocessament.

- Arquitectura: Definir el llenguatge de programació, llibreries a utilitzar i els models de Machine Learning seleccionats.

Fase 3 (PAC 3): del 06/11 al 03/12

- Entrenament: Implementar el codi per a l'entrenament del diferents models amb les dades seleccionades i preprocessades.
- Avaluació: Avaluar els models utilitzant diferents mètriques com *accuracy*, *precision*, *recall* i *f1-score*.
- Resultats: Analitzar els resultats i redactar les conclusions extretes.

Fase 4 (PAC4): del 01/12 al 07/01

Memòria: Redactar el document assegurant que el format i els aspectes formals compleixin amb les directrius establertes. Revisar, ordenar i aplicar el format estàndard a les referències, per garantir que totes les fonts citades estan correctament referenciades.

Fase 5: del 08/01 al 14/01

Vídeo: Preparació de la presentació, elaboració d'un guió i gravació del vídeo final, assegurant que té un format clar i ben estructurat.

Fase 6: del 15/01 al 24/01

Defensa TFM: Preparar la presentació-resum, estructurant-la en quatre diapositives. Assajar l'exposició i organitzar els punts clau per comunicar els resultats i les conclusions de manera clara, i estar preparat per respondre a possibles preguntes del tribunal.

En la Figura 1, es presenta el diagrama de Gantt associat a aquesta planificació.

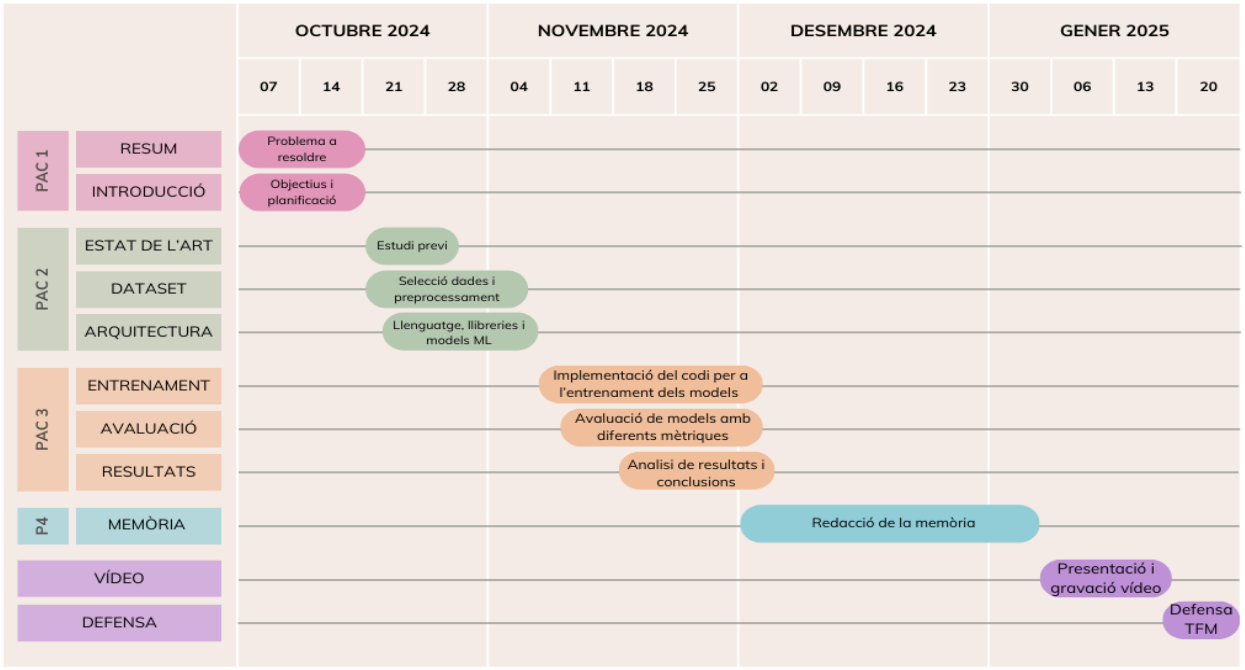

Figura 1: Diagrama de Gantt

1.5 Impacte en sostenibilitat, ètic-social i de diversitat

Pel que fa a l'ètica, és fonamental garantir la protecció de la privacitat de les dades personals en l'ús de tècniques d'Intel·ligència Artificial per a la detecció de comportaments anòmals, assegurant el compliment de normatives com el GDPR. A més, en relació amb l'impacte ambiental, és important considerar el consum energètic associat a l'entrenament de models d'IA, buscant solucions que optimitzin el rendiment i maximitzin l'eficiència i la sostenibilitat. Tenir en compte aquestes qüestions és essencial per garantir un ús responsable i ètic de les noves tecnologies.

2 ESTAT DE L'ART

En els darrers anys, la ciberseguretat ha esdevingut un dels objectius més rellevants per a moltes empreses. Les organitzacions necessiten estar protegides tant de ciberatacs com d'amenaces internes, i, per això, requereixen solucions avançades per detectar i prevenir aquests incidents. Mitjançant tècniques d'intel·ligència artificial i machine learning, és possible processar grans volums de dades i detectar patrons que ajuden a identificar ciberatacs o comportaments sospitosos. Aquest estat de l'art proporciona una visió general dels enfocaments actuals en machine learning per a ciberseguretat i els datasets utilitzats en la investigació.

2.1 Tècniques d'aprenentatge automàtic en ciberseguretat

Diversos algorismes de machine learning han demostrat ser eficaços per a la detecció d'anomalies i la classificació d'atacs en xarxes i sistemes d'informació. A continuació, es descriuen alguns d'aquest algorismes segons el tipus d'aprenentatge:

Aprenentatge supervisat

En l'aprenentatge supervisat, les dades solen tenir etiquetes de classe, fet que resulta útil per a tasques de classificació. Els sistemes de detecció d'anomalies que utilitzen tècniques d'aprenentatge supervisat es basen en mostres tant normals com anòmales per construir els seus models, la qual cosa els fa més precisos i amb menys falsos positius. Aquests sistemes presenten la dificultat d'obtenir mostres etiquetades, especialment de comportaments anòmals. Malgrat això, són àmpliament utilitzats en la detecció d'intrusions i malware en xarxes, així com en la identificació d'amenaces internes. Entre els algorismes d'aprenentatge supervisat més destacats en el context de la ciberseguretat, trobem:

- Naive Bayes: Aquest alorismes es basen en el teorema de Bayes, una tècnica de classificació estadística. Són coneguts com a "Naive" o "Innocents" perquè assumeixen que les variables predictores són independents entre si. Hi ha una gran quantitat d'ús d'aquest algorisme en la ciberseguretat, per exemple en la classificació d'atacse [92].
- Random Forests: Els Random Forests estan formats per un gran nombre de petits arbres de decisió, anomenats estimadors, que produeixen cadascun les seves pròpies prediccions. Aquest algorisme s'ha aplicat, entre d'altres, en sistemes de detecció d'intrusions (IDS) utilitzant els conjunts de dades KDD-99 i NSL-KDD [18].
- Sistemes de Regressió Logística: Els algorismes de regressió logística intenten predir un valor discret, és a dir, són bons per fer seleccions. En ciberseguretat, s'aplica especialment per detectar patrons de comportament que permetin identificar amenaces en temps real i en la detecció de codi maliciós, per exemple en la inicialitzar els pesos en models híbrids, millorant així la classificació de comportaments maliciosos en anàlisis de seguretat en temps real [20].
- Support Vector Machines (SVM): Aquests algorismes es basen en trobar un hiperplà òptim que separi les dades en diferents classes. Aquest hiperplà es determina maximitzant la distància (o marge) entre els

punts de dades més propers de classes oposades. En el context de la ciberseguretat, els SVM poden usar-se per identificar intrusions i atacs de seguretat, mitjançant l'anàlisi de patrons de dades i la detecció de comportaments anòmals. Aquestes tècniques són especialment efectives en la gestió de dades complexes i multidimensionals, millorant així la capacitat de resposta davant de noves amenaces [19].

Aprenentatge no supervisat

En l'aprenentatge no supervisat, els models no solen tenir una variable dependent i es basen en els patrons disponibles dins del conjunt de dades per agrupar les dades en diferents categories. En aquest tipus d'aprenentatge, s'utilitzen diferents algorismes, com ara:

- Clustering: Les tècniques de clustering es basen en l'agrupació de dades similars en clústers o grups basats en les seves característiques. Aquestes tècniques s'utilitzen sovint en ciberseguretat per identificar patrons o comportaments anòmals. Per exemple, poden ajudar a detectar trànsit de xarxa sospitós agrupant connexions similars i identificant aquelles que no s'ajusten als patrons normals. Això és útil per a la detecció d'atacs com poden ser intrusions, comportaments d'usuaris maliciosos o connexions de malware. Algoritmes comuns inclouen K-means, K-medoids, i el clustering jeràrquic [10].
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise): És un algorisme de clustering que identifica clústers basats en densitats de punts, útil per detectar formes arbitràries i ignorar soroll. Es va utilitzar com a mètode identificar comportaments atípics en entorns de xarxa, ressaltant-ne la utilitat en la detecció d'activitats no autoritzades [16].
- Mineria de regles d'associació: En l'aprenentatge de regles d'associació, s'identifiquen associacions entre variables en un conjunt de dades. S'han desenvolupat diferents mètodes de mineria de regles d'associació per a l'aprenentatge automàtic, com els basats en arbres o patrons freqüents. Un exemple és l'algorisme Adaptive-Miner basat en Apache Spark explora un sistema de detecció d'intrusions en temps real analitzant logs de múltiples fonts [17].

Aprenentatge profund

L'aprenentatge profund (deep learning) és una branca de la intel·ligència artificial que se centra en la creació i millora d'algoritmes capaços d'aprendre de grans quantitats de dades mitjançant xarxes neuronals artificials, millorant la capacitat de càlcul i desenvolupant algorismes més eficients. Aquest enfocament cerca emular el funcionament del cervell humà per identificar patrons complexos i relacions amagades en les dades. L'aprenentatge profund s'ha utilitzat per resoldre problemes que, d'altra manera, serien gairebé impossibles de tractar amb algorismes tradicionals, com el reconeixement de veu, imatge i text, així com la predicció de comportaments [1].

Alguns del algorismes d'aprenentatge profund que s'han utilitzat en tasques de ciberseguretat són:

- Xarxa de creences profundes (DBN): és un model gràfic generatiu, o una classe de xarxa neuronal profunda, composta per múltiples capes de variables latents ("unitats ocultes"), amb connexions entre les capes però no entre unitats dins de cada capa. S'ha utilitzat, entre d'altres, per a la detecció de

malware i com a codificador automàtic per extreure els vectors de característiques de les dades d'entrada [11], i per a la detecció d'anomalies de xarxa en temps real [12].

- Xarxes neuronals convolucionals (CNN): Aprèn característiques per si mateixa mitjançant l'optimització de filtres. S'han utilitzat en alguns estudis per a la detecció de programari maliciós per a Android [14].
- CNN-LSTM: és una arquitectura híbrida d'aprenentatge profund que combina els punts forts de les xarxes neuronals convolucionals (CNN) i les xarxes Long Short-Term Memory (LSTM). Aquest algorisme s'ha emprat en diferents estudis per a la detecció d'intrusions i la classificació d'atacs: DoS, probe, R2L i U2R [13].
- Models de llenguatge gran (LLM): Els models LLM són una classe de models d'aprenentatge profund que utilitzen xarxes neuronals per processar i generar text a gran escala. Aquests models han demostrat ser útils en tasques de detecció de ciberamenaces, ja que poden analitzar textos no estructurats per identificar comportaments sospitosos. S'han utilitzat en la detecció d'amenaces de seguretat en dispositius IoT/IIoT [15].

En la Figura 2 es mostren els trets més rellevants dels algoritmes descrits, destacant les característiques clau.

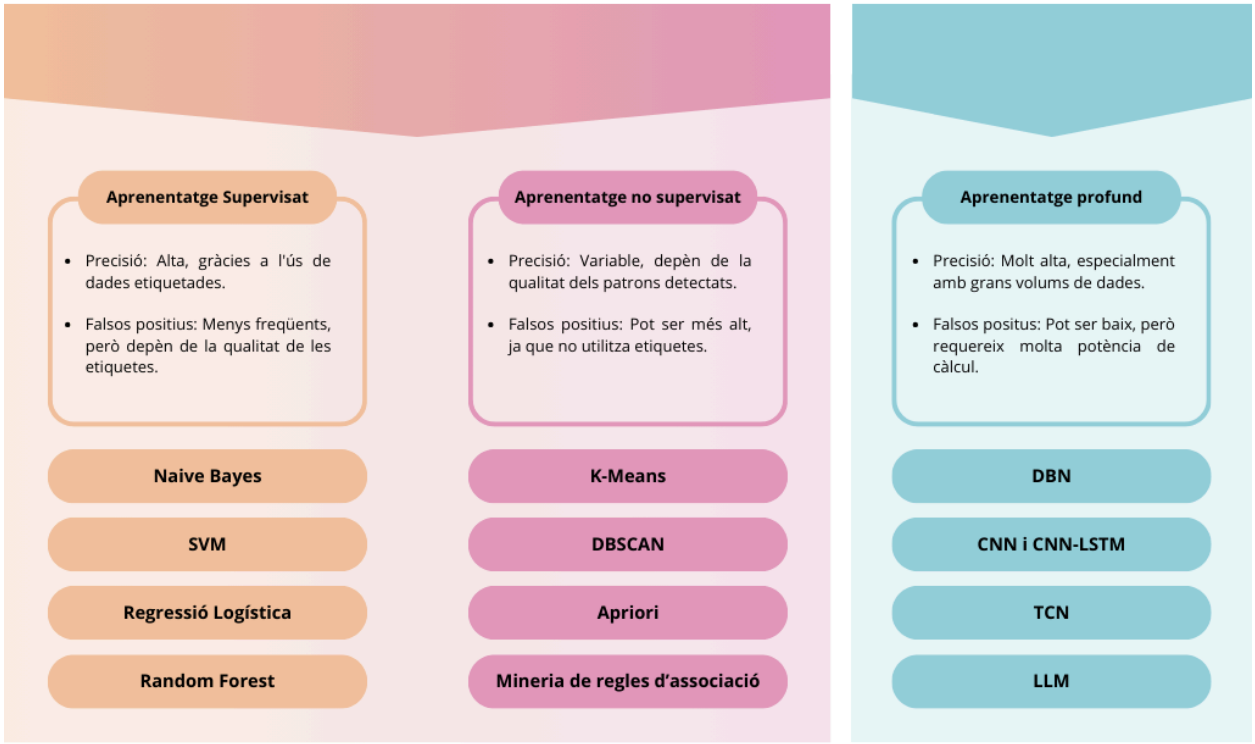

Figura 2: Esquema de tècniques d'aprenentatge de Machine Learning

2.2 Datasets usats en la detecció d'Intrusions o Anomalies

Recopilar dades o trobar un conjunt de dades adequat per a aplicacions de machine learning en el context de la ciberseguretat pot ser un gran repte, especialment en tasques de detecció d'anomalies, on les dades han de representar tant comportaments normals com maliciosos. Tot i això, existeixen conjunts de dades científics àmpliament utilitzats per entrenar models en aquest camp.

Conjunts de dades com el KDD Cup 1999 i el UNSW-NB15 proporcionen una base sòlida per als investigadors, ja que permeten l'entrenament i l'avaluació de models de detecció d'intrusions i anomalies amb atacs coneguts i actualitzats. En concret, el KDD Cup 1999, derivat de les dades de DARPA 1998, inclou quatre categories principals d'atacs (DoS, U2R, R2L, Probing) i ha estat àmpliament utilitzat per avaluar algorismes de machine learning en la detecció d'intrusions [1]. No obstant això, aquest dataset ha estat objecte de crítiques per part d'alguns investigadors a causa de la seva limitació en nous vectors d'atac i del biaix en la representació d'atacs DoS [6].

Per superar aquestes limitacions, es va crear el dataset UNSW-NB15. Aquest conjunt de dades inclou nou categories d'atacs moderns, com ara, Fuzzers, Analysis, Backdoors, DoS, Exploits, i Worms, proporcionant una base més completa i actualitzada per a la detecció d'intrusions [7][8].

Tot i la gran utilitat i reconeixement del conjunt de dades UNSW-NB15, aquest està més orientat a la detecció d'intrusions a les xarxes. En altres contextos, com ara a la detecció d'anomalies a nivell de host, hi ha altres conjunts de dades, com l'ADFA-LD i el Unified Host and Network Data Set de LANL, que se centren específicament en aquests casos. El conjunt ADFA-LD, desenvolupat per a la detecció d'anomalies basades en hosts, conté traces de sistemes Linux i atacs específics contra servidors, i està especialment orientat a proves de seguretat en sistemes operatius moderns [22]. D'altra banda, el conjunt de dades Unified Host and Network Data Set de LANL combina dades d'usuari i de xarxa per permetre la detecció de comportaments interns sospitosos i anomalies de xarxa. Tot i no estar etiquetat, aquest conjunt s'ha utilitzat en estudis d'aprenentatge no supervisat, com el clustering i els autoencoders, per identificar patrons anòmals en activitats internes [9].

2.3 Aplicacions Pràctiques dels Algorismes de Machine Learning en el Sector de la Ciberseguretat

Aquests algorismes no només es fan servir en investigació acadèmica, sinó que també són utilitzats per diverses empreses que han integrat la intel·ligència artificial en les seves estratègies de ciberseguretat.

A continuació, es mostren exemples d'algunes empreses que utilitzen intel·ligència artificial per a la detecció de comportaments anòmals i ciberatacs:

- CrowdStrike és una companyia especialitzada en la protecció al núvol, que disposa de la plataforma CrowdStrike Falcon per oferir solucions d'IA contra amenaces.

- ImpalaSEIDOR ofereix serveis per conèixer el nivell de risc i proposa solucions per aplicar mesures de seguretat.
- Opticks es dedica a la detecció i prevenció del frau en anuncis online, utilitzant machine learning i intel·ligència artificial.
- Evolutio, amb un Centre d'Operacions de Ciberseguretat (SOC) a Barcelona, es dedica a la detecció d'amenaces digitals i a dur a terme investigacions proactives, mitjançant l'ús d'intel·ligència artificial i big data.
- Build38 ha creat un sistema de seguretat per garantir la seguretat de les aplicacions mòbils, detectant amenaces tant a nivell d'usuari com de desenvolupador. [23]

3 METODOLOGIA I ARQUITECTURA

En aquest apartat, es descriuen les diferents etapes del tractament de la informació analitzada en aquest treball, des de l'obtenció de les dades en el seu origen fins a l'obtenció dels resultats finals. La Figura 3 mostra a grans trets els passos realitzats en el procés.

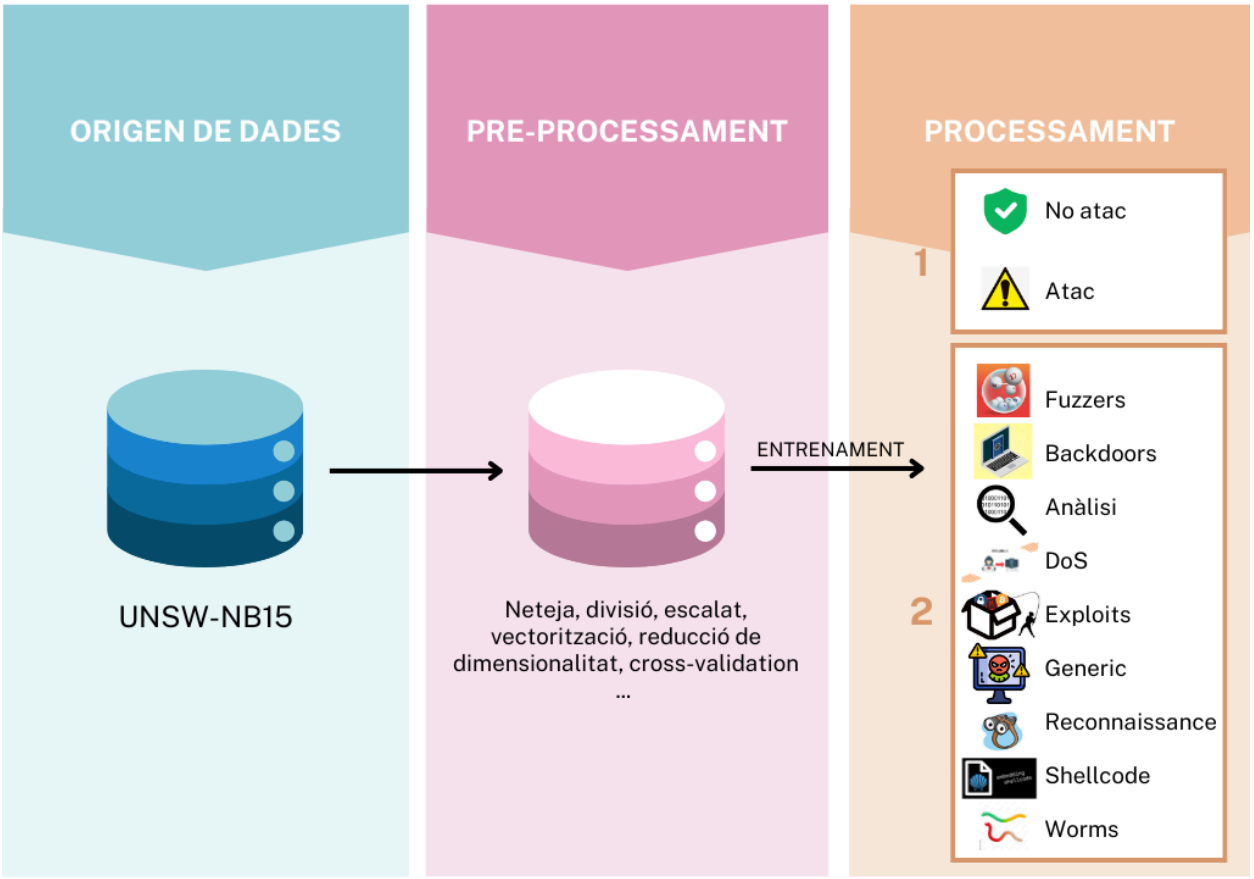

Figura 3: Diagrama de les etapes principals per a la resolució de tarques

3.1 Llenguatge i llibreries

El treball s'implementa principalment en **Python**, un llenguatge molt utilitzat entre els projectes de Machine Learning, ja que disposa d'un gran nombre de llibreries especialitzades. Les llibreries utilitzades en la implementació del treball actual, són:

- **Scikit-learn:** S'utilitza per a la implementació de models d'aprenentatge automàtic, així com per a l'avaluació de rendiment i la validació de models [24].
- **Pandas:** Aquesta llibreria s'utilitza per a la manipulació i preprocessament de dades [25].
- **NumPy:** S'utilitza per a operacions numèriques i manipulació de matrius [26].

- **Matplotlib:** Aquesta llibreria serveix per a la visualització de dades i resultats, permetent una anàlisi més clara dels patrons detectats [27].
- **Scipy:** Proporciona funcions estadístiques avançades per a l'anàlisi de dades, útils per a la modelització estadística [28].
- **Tensorflow:** S'utilitza per a la construcció i entrenament de models de deep learning, incloent xarxes neuronals convolucionals (CNN) i xarxes neuronals recurrents (RNN), com les LSTM [29].
- **Collections (Counter):** Aquesta llibreria de Python es fa servir per a l'anàlisi de la freqüència de valors dins d'una col·lecció o conjunt de dades, facilitant l'extracció de patrons i estadístiques simples [30].
- **Pickle:** S'utilitza per a la serialització i deserialització d'objectes Python, permetent guardar models o preprocessaments per al seu ús posterior [30].
- **Seaborn:** Llibreria de visualització de dades basada en Matplotlib que ofereix gràfics estadístics més detallats i amb un disseny més atractiu. Es fa servir per a analitzar i explorar dades de manera visual [31].
- **Warnings:** Aquesta llibreria permet gestionar i ignorar advertències generades durant l'execució del codi, garantint una sortida més neta [30].

3.2 Descripció i preprocessament de dades

En aquest apartat es descriu l'origen de les dades utilitzades en l'estudi actual junt amb les tasques usades per al seu preprocessament.

3.2.1 Origen de dades

S'ha usat el conjunt de dades UNSW-NB15, un conjunt de dades dissenyat per a la investigació en la detecció d'intrusions a la xarxa. Aquest conjunt de dades es va crear a la Universitat de Nova Gal·les del Sud (UNSW) i inclou una varietat de trànsit de xarxa, tant normal com maliciós, per avaluar el rendiment dels sistemes de detecció d'intrusions.

Aquest conjunt de dades ha estat creat a partir de paquets de xarxa en brut generats per l'eina de generació de trànsit en xarxa IXIA PerfectStorm, utilitzada en el Cyber Range Lab de UNSW Canberra. Aquesta eina permet la generació d'un híbrid d'activitats normals modernes i comportaments d'atac contemporanis sintètics.

El conjunt de dades és etiquetat i inclou nou famílies d'atacs: Fuzzers, Anàlisi, Backdoors, DoS (Denial of Service), Exploits, Generic, Reconnaissance, Shellcode i Worms. S'han utilitzat les eines Argus i Bro-IDS per processar els paquets, i s'han desenvolupat dotze algoritmes que generen un total de 49 característiques amb la seva etiqueta de classe. El conjunt de dades consta de quatre fitxers .csv, amb un total de 2.540.044 registres [7][8].

A continuació, es descriuen breument cadascuna de les famílies d'atac etiquetades en el conjunt de dades utilitzat:

- Fuzzers: Els fuzzers o generadors de dades aleatòries són eines utilitzades per realitzar atacs enviant dades malformades i aleatòries a una aplicació. Aquest procés té com a objectiu identificar i aprofitar les possibles vulnerabilitats del sistema i les falles de seguretat que puguin existir en les aplicacions [32].
- Anàlisi: Inclou tècniques com l'esnifatge de paquets i l'anàlisi de trànsit per obtenir informació sensible o identificar punts febles en una xarxa.
- Backdoors: Els backdoors o atacs de porta del darrere són mecanismes ocults que permeten accedir remotament a un sistema de manera no autoritzada. Sovint s'introdueixen mitjançant codi implantat pels mateixos desenvolupadors o bé són inserits per atacants a través de descàrregues de fitxers o aplicacions malicioses. Aquests mètodes permeten el control del sistema sense el coneixement ni el consentiment de l'usuari legítim [33].
- Denial Of Service: Aquests atacs busquen fer que un servei o xarxa sigui inoperant, saturant-lo amb l'enviament d'una gran quantitat de trànsit o sol·licituds. L'objectiu principal és interrompre el funcionament normal del servei, impedint que els usuaris hi puguin accedir amb normalitat, , afectant així la disponibilitat i l'accessibilitat [34].
- Exploits: Els exploits són una mena d'atacs que aprofiten vulnerabilitats específiques del sistema, en programari o maquinari, abans que el proveïdor o els usuaris siguin conscients de la seva existència. Els atacants exploten aquestes vulnerabilitats per obtenir accés no autoritzat o causar danys al sistema [35].
- Generic: En aquest grup s'inclouen atacs que no es classifiquen fàcilment en altres categories, sovint utilitzant tècniques combinades o noves.
- Reconnaissance: El reconnaissance o reconeixement cibernètic és el procés en què els atacants recopilen informació sobre les seves víctimes per identificar vulnerabilitats i planificar atacs, tot evitant ser detectats. Aquest procés pot incloure activitats com la recopilació de dades, l'escaneig de ports i el mapatge de xarxes, amb l'objectiu de preparar atacs futurs [36].
- Shellcode: El shellcode és un codi maliciós dissenyat per executar tasques perjudicials en un equip víctima. S'injecta a través de vulnerabilitats i sol estar escrit en llenguatge assemblador i en format hexadecimal. Aquest codi s'executa directament en la memòria del sistema, permetent a l'atacant prendre el control del dispositiu o executar accions específiques sense necessitat d'instal·lar programari addicional [37].
- Worms: Els worms o cucs informàtics són programes autònoms capaços de replicar-se i propagar-se a través de xarxes, infectant diversos ordinadors on operen de manera independent. Cada cuc pot generar una gran quantitat de còpies de si mateix, cosa que sovint provoca danys significatius i un consum elevat de recursos del sistema i de la xarxa [38].

A la Taula 1 es llisten les 49 característiques emmagatzemades en el conjunt de dades original, indicant el tipus de dades, així com una definició amb una breu descripció del contingut de cada una d'elles, amb l'objectiu de proporcionar una visió general de les dades que s'han utilitzat en aquest estudi.

Columna	Tipus	Definició
id	int64	Identificador
dur	float64	Durada de la connexió
proto	object	Protocol utilitzat
service	object	Servei utilitzat
state	object	Estat de la connexió
spkts	int64	Nombre de paquets enviats des de l'origen
dpkts	int64	Nombre de paquets rebuts per la destinació
sbytes	int64	Bytes enviats des de l'origen
dbytes	int64	Bytes rebuts per la destinació
rate	float64	Taxa de paquets o bytes per segon en una connexió específica
sttl	int64	Time to Live (TTL) de l'origen
dttl	int64	Time to Live (TTL) de la destinació
sload	float64	Càrrega de l'origen
dload	float64	Càrrega de la destinació
sloss	int64	Paquets perduts des de l'origen
dloss	int64	Paquets perduts per la destinació
sinpkt	float64	Nombre de paquets enviats des de l'origen
dinpkt	float64	Nombre de paquets rebuts per la destinació
sjit	float64	Jitter de l'origen. El jitter és la variació en el temps de retard dels paquets de dades enviats des de l'origen
djit	float64	Jitter de la destinació. És la variació en el temps de retard dels paquets de dades rebuts per la destinació

swin	int64	Finestra de l'origen
stcpb	int64	Seqüència TCP de l'origen
dtcpb	int64	Seqüència TCP de la destinació
dwin	int64	Finestra de la destinació
tcprtt	float64	Round Trip Time (RTT) de TCP, temps que triga un paquet TCP a viatjar des de l'origen fins a la destinació i tornar a l'origen
synack	float64	Temps entre SYN i ACK
ackdat	float64	Temps entre ACK i dades
smean	int64	Mida mitjana dels paquets enviats des de l'origen
dmean	int64	Mida mitjana dels paquets rebuts per la destinació
trans_depth	int64	Profunditat de la transacció
response_body_len	int64	Longitud del cos de la resposta
ct_srv_src	int64	Nombre de connexions al mateix servei des de la mateixa IP d'origen
ct_state_ttl	int64	Nombre de paquets amb el mateix estat i TTL
ct_dst_ltm	int64	Nombre de connexions a la mateixa IP de destinació
ct_src_dport_ltm	int64	Nombre de connexions des de la mateixa IP d'origen al mateix port de destinació
ct_dst_sport_ltm	int64	Nombre de connexions a la mateixa IP de destinació des del mateix port d'origen
ct_dst_src_ltm	int64	Nombre de connexions entre la mateixa IP d'origen i la mateixa IP de destinació
is_ftp_login	int64	Indicador de si és un login FTP

ct_ftp_cmd	int64	Nombre de comandes FTP
ct_flw_http_mthd	int64	Nombre de fluxos amb el mateix mètode HTTP
ct_src_ltm	int64	Nombre de connexions des de la mateixa IP d'origen
ct_srv_dst	int64	Nombre de connexions al mateix servei a la mateixa IP de destinació.
is_sm_ips_ports	int64	Indicador de si els ports i IPs són petits
attack_cat	object	Categoria de l'atac
label	int64	Etiqueta de la classe (normal o atac)

Taula 1: Definició del conjunt de dades UNSW-NB15

3.2.2 Preprocessament de dades

La tasca de preprocessament s'encarrega de la preparació de les dades per tal que puguin passar a ser dades numèriques preparades per ser analitzades i categoritzades pel sistema de detecció i categorització d'atacs a la xarxa. Dins del preprocessament s'han portat a terme diferents tasques, descrites a continuació.

Divisió del dataset

Com ja s'ha comentat prèviament, el dataset UNSW-NB15 es proporciona organitzat inicialment en quatre fitxers (*UNSW-NB15_1.csv, UNSW-NB15_2.csv, UNSW-NB15_3.csv, UNSW-NB15_4.csv*), acompanyats d'un fitxer addicional que inclou la descripció de les característiques emmagatzemades i el número de columna corresponent a cada característica (*UNSW-NB15_features.csv*).

Per treballar amb aquestes dades, s'ha utilitzat la llibreria pandas per carregar els quatre fitxers i emmagatzemar-los en un sol DataFrame [25]. A continuació s'ha dividit el conjunt de dades en els conjunts necessaris (un conjunt d'entrenament amb un 30% de les dades i un de prova amb el 70% restant).

El gràfic de la Figura 4 mostra la proporció de dades normals (valor 0) i dades d'atac (valor 1) tant en el conjunt d'entrenament com en el conjunt de prova. Es pot observar que les dades no estan balancejades, ja que hi ha moltes més dades normals que dades d'atac. També es pot veure que la distribució és similar en els dos conjunts de dades. El gràfic de la Figura 5 mostra la proporció de dades normals i de les diferents categories d'atac tant en el conjunt d'entrenament com en el de prova. Com en el cas anterior, es pot veure que la distribució és similar en els dos conjunts de dades.

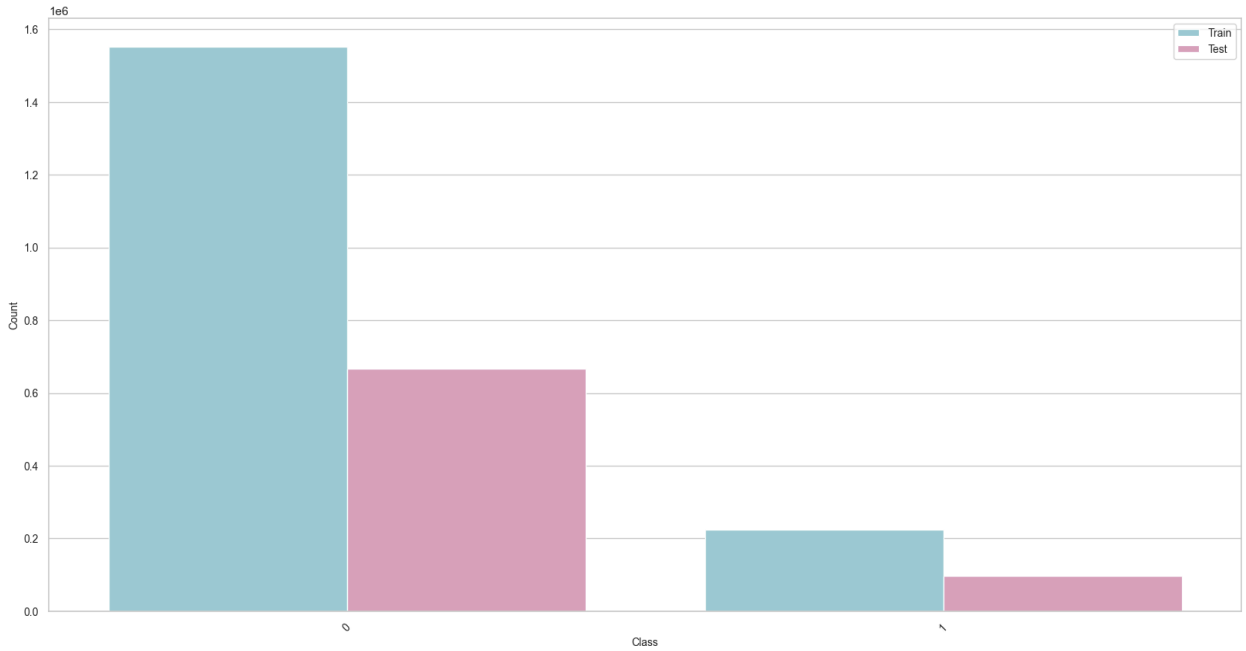

Figura 4: Gràfic de distribució de classes

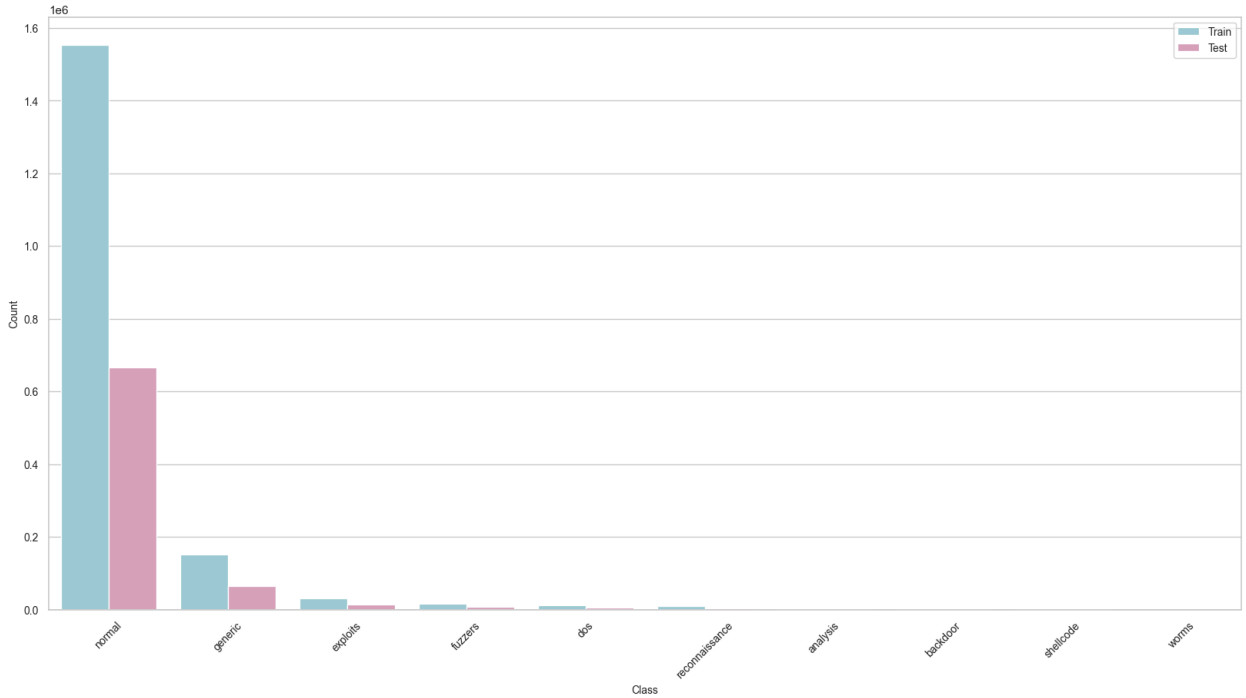

Figura 5: Gràfic de distribució de categories d'atac

A partir del DataFrame obtingut amb les dades dels fitxers inicials, s'han creat els següents conjunts per facilitar el procés de classificació:

- *X_train* i *X_test:* Característiques utilitzades pels models per a entrenar i avaluar.
- *y_train_binary* i *y_test_binary*: Etiquetes per a la classificació binària, que identifica si un registre és un atac (1) o no (0).
- *y_train_category* i *y_test_category*: Etiquetes per a la classificació multiclasse, que identifica si un registre és normal o el tipus d'atac específic.

Aquesta divisió ens permet abordar les dues tasques de classificació de manera independent:

1. Classificació binària (atac / no atac).
2. Classificació multiclasse (normal / tipus específic d'atac).

A la Taula 2 es mostren els conjunts de dades generats i el seu ús.

Conjunt	Descripció	Tipus de classificació
X_train	Característiques per a l'entrenament.	Entrenament.
X_test	Característiques per a l'avaluació.	Test.
y_train_binary	Etiquetes binàries (atac / no atac) per entrenar.	Classificació binària (entrenament).
y_test_binary	Etiquetes binàries (atac / no atac) per test.	Classificació binària (test).
y_train_category	Etiquetes multiclasse per entrenar.	Classificació multiclasse (entrenament).
y_test_category	Etiquetes multiclasse per test.	Classificació multiclasse (test).

Taula 2: Conjunts de dades utilitzats

Tractament de valors nuls i Consistència de tipus de dades

S'han identificat valors nuls únicament en les columnes que es llisten en la Figura 6.

```
Name
ct_flw_http_mthd    1348145
is_ftp_login        1429879
attack_cat          2218764
dtype: int64
```

Figura 6: Característiques amb valors nuls

A la columna *attack_cat* s'ha observat que tots els registres amb valors nuls tenien un valor de 0 a la columna *label* (és a dir, normal). Per tant, s'ha substituït el valor nul de la columna *attack_cat* pel text "normal".

A les columnes *ct_flw_http_mthd* i *is_ftp_login,* que són característiques numèriques, s'han substituït els valors nuls per 0s.

A més, s'ha comprovat que el tipus de totes les columnes coincidís amb el tipus definit al fitxer de característiques *UNSW-NB15_features.csv*. S'ha observat que coincidien en tots els casos, excepte en les següents columnes:

```
Columns with incorrect types:
['sport', 'dsport', 'ct_flw_http_mthd', 'is_ftp_login', 'ct_ftp_cmd']
```

Figura 7: Característiques amb tipus incorrectes

ct_ftp_cmd hauria de ser un enter, però es llegia com a un objecte. Aquest fet es devia a l'existència de registres amb valor " ". Per solucionar-ho, s'han substituït aquests valors per 0 i s'ha pogut convertir la columna a tipus enter.

is_ftp_login hauria de ser una característica binària, però, a part de tenir 0s i 1s, també contenia alguns 2s i 4s. S'han substituït tots els valors més grans que 1 per 1s.

ct_flw_http_mthd hauria de ser un enter, però s'estava llegint com a *float*. Per resoldre-ho, s'ha forçat que la columna sigui de tipus *integer*.

Les columnes *sport* i *dsport* haurien de ser de tipus enter, però es llegien com a objectes. S'ha observat que hi havia alguns valors en format hexadecimal que es percebien com a strings, així com també alguns amb un "-". S'han convertit els valors hexadecimals a decimals i els "-" a 0s per poder interpretar totes les dades com a enters.

Eliminació de característiques no significatives

En primer lloc, s'han eliminat les característiques relacionades amb les adreces IP i els ports, ja que no aporten informació rellevant per a la classificació actual.

A més, s'han descartat les característiques que presenten una correlació superior a 0,95 amb altres variables, ja que la seva contribució informativa és mínima i consumeixen recursos innecessàriament.

Aplicació del logaritme natural de 1+x

S'ha aplicat una transformació logarítmica $\log(1+x)$ a determinades variables contínues del conjunt de dades amb l'objectiu de reduir l'asimetria (skewness) i tractar els valors extrems. Aquesta transformació resulta especialment útil en variables amb distribucions altament esbiaixades, ja que comprimeix els valors grans i expandeix els petits, aconseguint una distribució més equilibrada. Abans de realitzar la transformació, s'han analitzat les correlacions entre les variables i la variable *label*, tant en la seva forma original com transformada, seleccionant aquelles en què la correlació millora després de l'aplicació del logaritme. A més, s'ha utilitzat el coeficient d'asimetria (skewness) com a criteri addicional per identificar les variables més beneficiades per aquest ajustament, prioritzant aquelles amb alta asimetria ($|skewness|>1$). Aquesta estratègia ha permès millorar la qualitat de les dades en el context del treball actual.

Escalat de característiques

Per preparar les columnes numèriques del dataset, s'han aplicat diferents tècniques d'escalat:

- StandardScaler: Centra les dades a una mitjana de 0 amb una desviació estàndard de 1. És útil per models que assumeixen una distribució normal de les característiques.
- RobustScaler: Escala les dades en funció dels seus quartils, fent-lo menys sensible als valors extrems (outliers).
- MinMaxScaler: Transforma els valors a un rang fix (entre 0 i 1), preservant la distribució relativa de les dades.

La selecció del millor tipus d'escalat per a cada model s'ha realitzat a partir de diverses execucions i anàlisi dels resultats obtinguts.

Vectorització de les dades

Per convertir les columnes categòriques en representacions numèriques aptes per als models, s'han aplicat les tècniques següents:

- One-Hot Encoding: S'ha utilitzat per a les columnes categòriques *proto*, *service* i *state*, ja que aquestes no tenen un ordre inherent. Aquesta tècnica crea columnes binàries per a cada categoria, assegurant una representació equitativa de les característiques.
- Label Encoding: Només s'ha aplicat a la columna *attack_cat*, que conté les classes multiclasse dels atacs, per simplificar-ne la representació numèrica. Normalment, aquest tipus d'encoding s'utilitza en característiques ordenades. En aquest cas, tot i no tenir cap ordre específic, ens ha semblat la millor solució, ja que aquesta columna s'utilitza com a etiqueta.

Reducció de dimensionalitat

Després de l'augment de dimensionalitat provocat pel One-Hot Encoding, s'ha aplicat PCA (Principal

Component Analysis) per reduir el nombre de dimensions, mantenint el 95% de la variància original. Aquesta reducció millora l'eficiència computacional dels models i elimina característiques redundants.

Cross-validation

S'ha utilitzat GridSearchCV com a metodologia per al tuning dels hiperparàmetres dels models. Aquesta tècnica realitza una cerca exhaustiva de les combinacions possibles de paràmetres per a cada model, aplicant validació creuada (cross-validation) per avaluar el rendiment de cada combinació. La validació creuada s'ha utilitzat durant el procés d'ajustament dels hiperparàmetres, la qual cosa ha permès optimitzar el rendiment global del sistema sense comprometre la seva capacitat de generalització. Aquest enfocament ha facilitat la identificació de la combinació òptima de paràmetres per a cada model, millorant els resultats obtinguts.

3.3 Implementació dels sistemes de classificació

Tal com s'ha comentat anteriorment, s'han construït diferents models per resoldre les tasques de detecció i classificació, a partir dels conjunts de dades d'entrada. S'han implementat tres solucions amb models de ML d'aprenentatge supervisat, dues solucions amd models d'aprenentatge no supervisat i dues solucions amb models d'aprenentatge profund.

S'han seguit els passos de cada algorisme, primer aplicats a la Tasca 1, és a dir, a la classificació de trànsit de xarxa en normal o atac, i després aplicat a la Tasca 2, on l'algorisme és el mateix, amb la diferència que en lloc de dues classes de categorització, n'hi ha deu. Per tant, els càlculs fets per a la classe normal i atac, s'han fet en la segona tasca per a cadascuna de les nou classes d'atacs (Fuzzers, Anàlisi, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode i Worms).

3.3.1 Models d'aprenentatge supervisat

En aquesta secció, es descriuen els models d'aprenentatge supervisat utilitzats en aquest treball.

Naive Bayes (NB)

Els models de NB són una classe d'algoritmes d'aprenentatge automàtic que es basen en el teorema de Bayes, una tècnica de classificació estadística. Aquests algoritmes són coneguts com a "Naive" o "Innocents" perquè assumeixen que les variables predictores són independents entre si, és a dir, la presència d'una característica no té cap relació amb la presència d'altres característiques.

Els models de Naive Bayes (NB) ofereixen una manera senzilla de construir models amb un bon rendiment gràcies a la seva simplicitat. En el nostre cas, s'ha entrenat el model gaussià de Naive Bayes, realitzant diverses execucions i ajustant el paràmetre de variància *var_smoothing*. Aquest paràmetre afegeix una porció de la major variància de totes les característiques a les variàncies per garantir l'estabilitat dels càlculs i evitar errors numèrics en la distribució gaussiana.

Finalment, després d'aplicar la validació creuada, s'ha determinat que el valor més òptim per al paràmetre *var_smoothing* és 1e-06, el qual ha mostrat els millors resultats en termes de rendiment i estabilitat del model.

Random Forest Classifier

Els models de Random Forest permeten treballar amb dades complexes evitant el sobreajustament. L'algorisme Random Forest Classifier funciona creant múltiples arbres de decisió durant l'entrenament i combinant els resultats d'aquests arbres per millorar la precisió i la robustesa del model. Cada arbre de decisió es construeix utilitzant un subconjunt aleatori de les dades i de les característiques. En el nostre cas, s'han fet diferents proves ajustant al nombre d'arbres utilitzat (*n_estimators*) i la profunditat màxima d'arbre de (*max_depth*).

A la Figura 8, es mostra l'*accuracy* obtingut per a la classificació binària en funció d'alguns dels valors provats per als paràmetres *n_estimators* i *max_depth*. Els millors resultats s'han obtingut amb *n_estimators* = 300 i *max_depth* = 20.

A la Figura 9, es mostra l'*accuracy* obtingut per a la classificació multiclasse en funció d'alguns dels valors provats per als paràmetres *n_estimators* i *max_depth*. Els millors resultats s'han obtingut amb *n_estimators* = 200 i *max_depth* = 20.

Figura 8: Heatmap segons *n_estimators* i *max_depth* per a Random Forest en la classificació binària

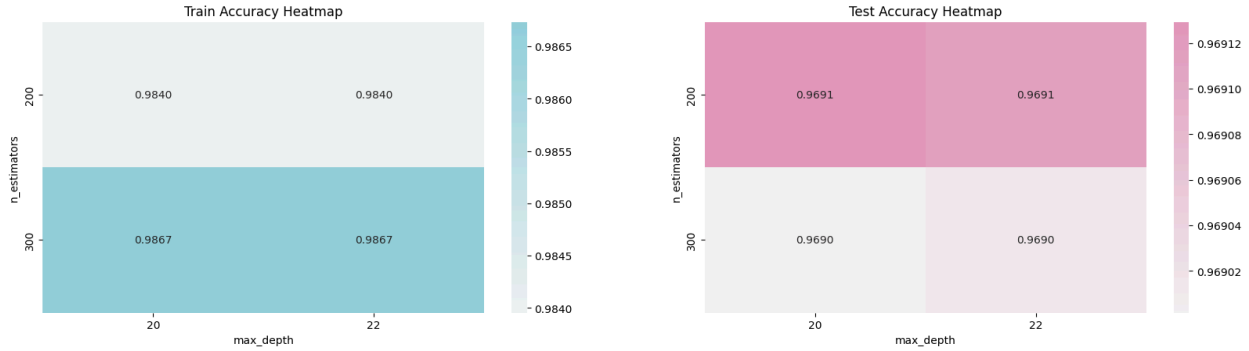

Figura 9: Heatmap segons *n_estimators* i *max_depth* per a Random Forest en la classificació categòrica

Logistic Regression

L'algorisme de regressió logística funciona modelant la probabilitat que una instància pertanyi a una classe determinada mitjançant una funció logística (sigmoide). En cas, s'ha utilitzat l'algoritme Logistic Regression ajustant els paràmetres de penalització entre *L1* i *L2*, específicament el paràmetre *penalty*, que controla la regularització del model. La *L*1 (regularització Lasso) afegeix una penalització basada en la suma absoluta dels coeficients del model, la qual pot portar a que alguns coeficients es facin zero, contribuint així a l'eliminació de característiques irrellevants, un procés conegut com a selecció de característiques. D'altra banda, la *L2* (regularització Ridge) afegeix una penalització basada en la suma dels quadrats dels coeficients, la qual evita que els coeficients siguin massa grans, millorant la capacitat de generalització del model sense eliminar característiques. Així mateix, s'ha utilitzat el paràmetre *alpha* com a factor de multiplicació de la penalització, el qual controla la intensitat de la regularització. Per a l'ajustament d'aquests paràmetres, s'ha realitzat una cerca de hiperparàmetres mitjançant GridSearchCV, optimitzant la combinació d'*alpha* i *penalty* per aconseguir el millor rendiment en el model.

A la Figura 8, es mostra l'a*ccuracy* obtingut per a la classificació binària en funció dels valors provats per als paràmetres alpha i penalty. Els millors resultats s'han obtingut amb *alpha* = 0,1 i *penalty* = L2.

A la Figura 9, es mostra l'*accuracy* obtingut per a la classificació multiclasse en funció dels valors provats per als paràmetres alpha i penalty. Els millors resultats s'han obtingut amb *alpha* = $1e^{-06}$ i *penalty* = L2.

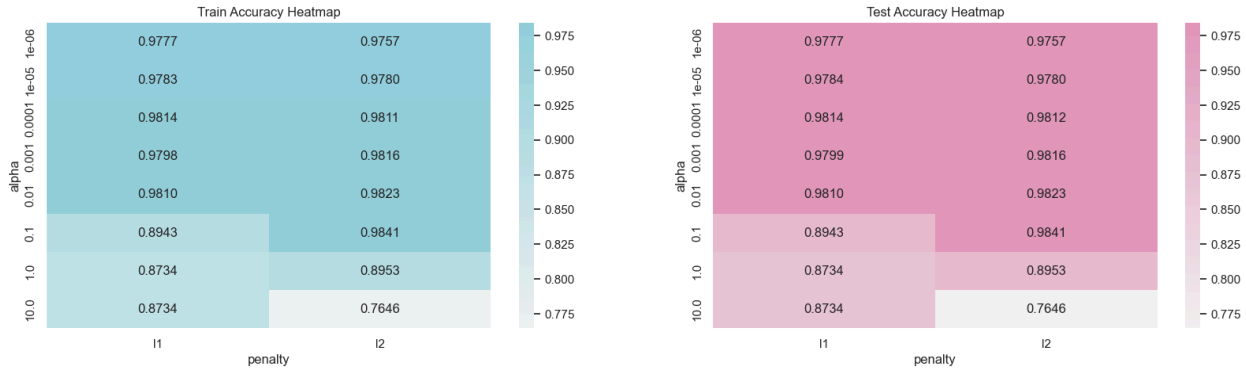

Figura 10: Heatmap d'*accurary* segons *alpha* i *penalty* per a Logistic Regression en la classificació binària

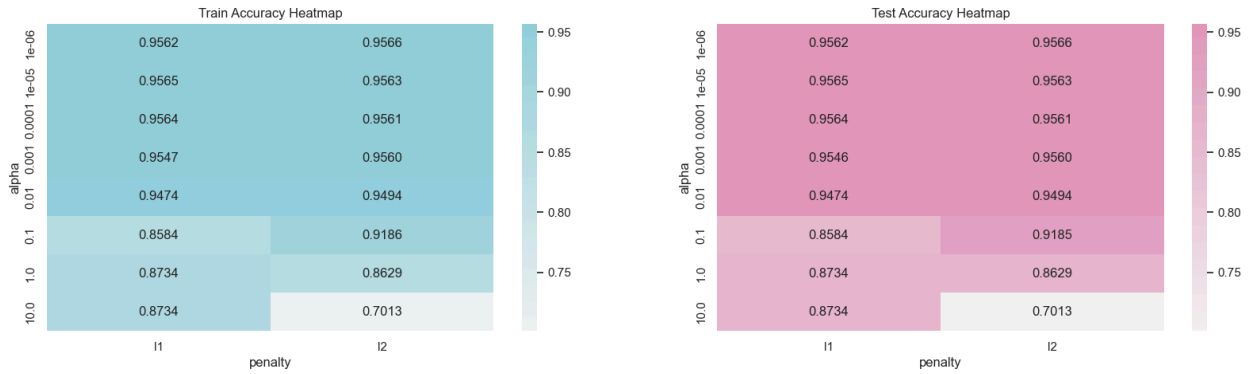

Figura 11: Heatmap d'*accurary* segons *alpha* i *penalty* per a Logistic Regression en la classificació categòrica

Support Vector Machines

Aquests algorismes es basen a trobar un hiperplà òptim que separi les dades en diferents classes. En aquest treball, s'ha utilitzat l'algoritme SVM ajustant els paràmetres de penalització (*penalty*), entre L1 i L2, per controlar la regularització del model, i *alpha* com a factor de multiplicació de la penalització, per regular la intensitat de la regularització. És a dir, en aquest cas s'han ajustat els mateixos paràmetres que s'han utilitzat per a la regressió logística, explicats més detalladament a l'apartat anterior.

A la Figura 10, es mostra l'*accuracy* obtingut per a la classificació binària en funció dels valors provats per als paràmetres alpha i penalty. Els millors resultats s'han obtingut amb *alpha* = 0,1 i *penalty* = L2.

A la Figura 11, es mostra l'*accuracy* obtingut per a la classificació multiclasse en funció dels valors provats per als paràmetres alpha i penalty. Els millors resultats s'han obtingut amb *alpha* = 0,0001 i *penalty* = L1.

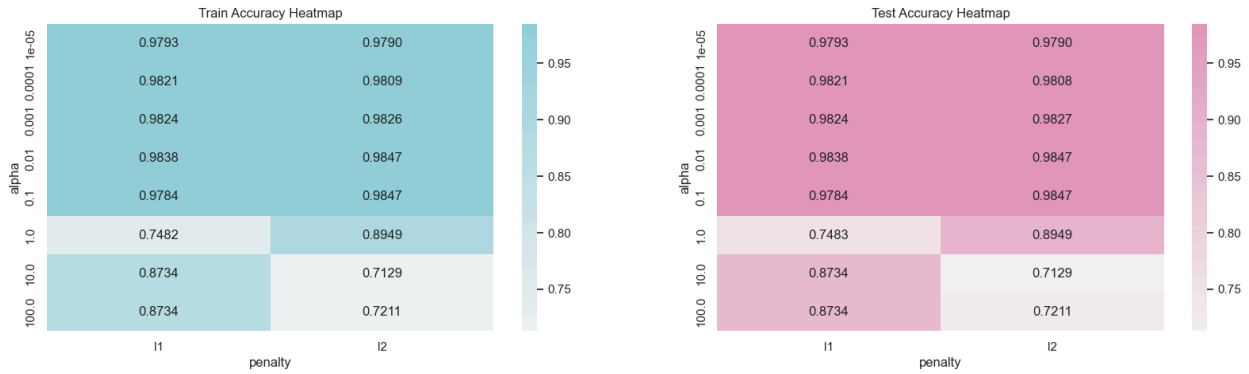

Figura 12: Heatmap d'*accurary* segons *alpha* i *penalty* per a SVM en la classificació binària

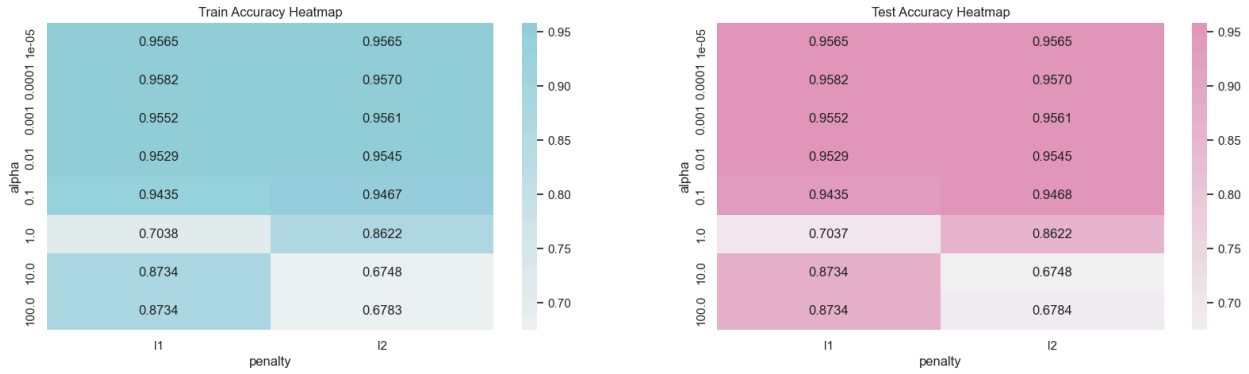

Figura 13: Heatmap d'*accurary* segons *alpha* i *penalty* per a SVM en la classificació categòrica

3.3.2 Models d'aprenentatge no supervisat

En aquesta secció, es descriuen els models d'aprenentatge no supervisat utilitzats en aquest treball. Per al cas dels models d'aprenentatge no supervisat, com que agrupen les dades en diferents clústers sense etiqueta, cada clúster s'ha mapejat a l'etiqueta més probable. Això s'ha aconseguit trobant l'etiqueta més freqüent dins de cada clúster. Posteriorment, s'han assignat aquestes etiquetes als clústers predits, permetent així avaluar el rendiment del models.

K-Means

L'algorisme k-Means és una tècnica d'agrupament que organitza les dades en k clústers, basant-se en la proximitat dels punts de dades als centroides. Inicialment, es seleccionen k centroides i cada punt de dades s'assigna al clúster amb el centroid més proper. Els centroides es recalculen de manera iterativa fins que els clústers convergeixen. En el nostre treball, s'ha utilitzat k-Means amb k=2 per agrupar el trànsit de xarxa en funció dels seus comportaments similars, identificant trànsit normal vs atacs. També s'ha utilitzat k-Means amb k=10 per agrupar el trànsit de xarxa en cadascuna de les tipologies d'atac analitzades. En aquest cas, el preprocessament s'ha dut a terme amb StandardScaler.

DBSCAN

L'algorisme DBSCAN (Density-Based Spatial Clustering of Applications with Noise) és una tècnica de clustering que agrupa punts de dades basant-se en la densitat de les seves àrees. A diferència de k-Means, DBSCAN no requereix especificar el nombre de clústers per endavant. En lloc d'això, identifica regions d'alta densitat de punts i les considera com a clústers, mentre que els punts en regions de baixa densitat es consideren soroll. En el nostre treball, s'ha utilitzat DBSCAN per identificar clústers de trànsit de xarxa que presenten comportaments similars, permetent detectar regions de dades d'alta densitat associades a activitats específiques, com ara atacs.

Per a la implementació de DBSCAN, s'ha controlat la distància màxima entre punts veïns (*eps*=0.7) i el nombre mínim de punts necessaris per formar un clúster (*min_samples*=100).

Degut a les limitacions de recursos, no ha estat possible executar aquest model amb el conjunt de dades complet. En lloc d'això, s'ha utilitzat una mostra reduïda (concretament un 10% del conjunt de dades de prova, amb un total de 76.201 registres). Com a conseqüència, els resultats obtinguts es basen en aquesta mostra inferior i no en la totalitat de la base de dades utilizada en els altres models.

3.3.3 Models d'aprenentatge profund

En aquesta secció, es descriuen els models d'aprenentatge profund utilitzats en aquest treball.

Xarxa DBN

Una Xarxa de Deep Belief Network (DBN) és un tipus de xarxa neuronal profunda composta per múltiples capes de Restricted Boltzmann Machines (RBM) i una capa final de classificació. Les RBM són models generatius capaços d'aprendre una distribució probabilística sobre els seus conjunts d'entrada. En el nostre cas, s'ha implementat una xarxa DBN amb una primera capa RBM de 256 components i una segona capa de 128 components. Aquestes capes RBM s'entrenen amb una taxa d'aprenentatge de 0,01 i 20 iteracions per capa. Un cop les característiques han estat extretes, la capa de regressió logística es fa càrrec de la classificació final, permetent al model aprendre a distingir entre les diferents classes d'atacs. Això permet al model aprofitar les capacitats de les RBM per capturar relacions complexes en les dades, mentre que la regressió logística proporciona una solució de classificació eficient.

CNN-LSTM

Aquest model combina les capacitats de les xarxes neuronals convolucionals (CNN) per a l'extracció de característiques i les xarxes neuronals recurrents (LSTM) per a la modelització de seqüències temporals. En aquest cas, els models utilitzats per a les taques 1 i 2 són una mica diferents.

En la primera tasca, el model es construeix de la següent manera:

1. Capa Convolucional (Conv1D): Aquesta capa aplica convolucions a les dades d'entrada per extreure característiques rellevants. S'han utilitzat 64 filtres amb una mida de nucli de 3 i la funció d'activació ReLU.
2. Capa de Max-Pooling (MaxPooling1D): Redueix la dimensionalitat de les característiques extretes, mantenint les més importants.
3. Capa LSTM: Aquesta capa processa les seqüències temporals de les característiques extretes per capturar les dependències temporals. S'han utilitzat 50 unitats LSTM amb la funció d'activació ReLU.
4. Capa Densa (Dense): La capa final és una capa densa amb una unitat de sortida i la funció d'activació sigmoide per a la classificació binària (atac o no atac).

El model es compila utilitzant l'optimitzador Adam i la funció de pèrdua *binary_crossentropy*, i es fa servir la mètrica *accuracy* per monitoritzar el rendiment durant l'entrenament. L'entrenament es duu a terme amb les dades preprocessades durant 5 èpoques, amb un *batch_size* de 64 i un *validation_split* del 20%. Finalment, el model que es guarda no és necessàriament el corresponent a l'última època (època 5), sinó aquell que ha obtingut els millors valors de *loss* i *accuracy* en el conjunt de validació al llarg de tot l'entrenament.

A la Figura 12 es mostra l'evolució de la funció de pèrdua (*loss*) i de l'*accuracy* tant per al conjunt d'entrenament com per al conjunt de validació al llarg de les èpoques.

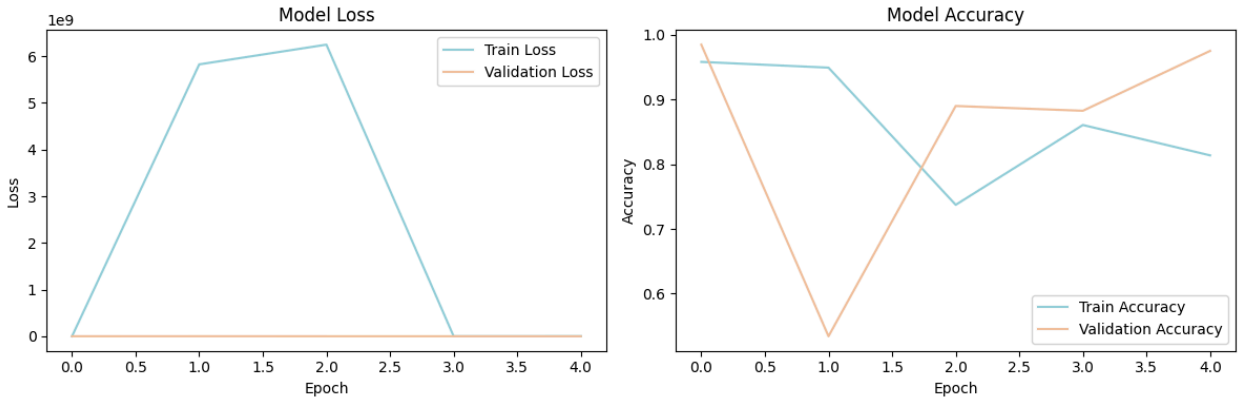

Figura 14: Evolució de *loss* i *accuracy* durant l'entrenament de CNN-LSTM en la classificació binària

En la segona tasca, el model es construeix seguint els següents passos:

1. Conversió de les etiquetes a format One-Hot: Les etiquetes de les classes d'atac es converteixen en vectors one-hot, que són necessaris per a la classificació multiclasse.
2. Capa Convolucional (Conv1D): Aquesta capa aplica convolucions a les dades d'entrada per extreure característiques rellevants. S'han utilitzat 64 filtres amb una mida de nucli de 3 i la funció d'activació ReLU.

3. Capa de Max-Pooling (MaxPooling1D): Redueix la dimensionalitat de les característiques extretes, mantenint les més importants.
4. Capa LSTM: Aquesta capa processa les seqüències temporals de les característiques extretes per capturar les dependències temporals. S'han utilitzat 50 unitats LSTM amb la funció d'activació ReLU.
5. Capa Densa (Dense): La capa final és una capa densa amb 10 unitats de sortida i la funció d'activació softmax per a la classificació multiclasse.

El model es compila utilitzant l'optimitzador Adam i la funció de pèrdua *categorical_crossentropy*, i es fa servir la mètrica *accuracy* per monitoritzar el rendiment durant l'entrenament. L'entrenament es duu a terme amb les dades preprocessades durant 5 èpoques, amb un *batch_size* de 64 i un *validation_split* del 20%. Finalment, el model que es guarda no és necessàriament el corresponent a l'última època (època 5), sinó aquell que ha obtingut els millors valors de *loss* i *accuracy* en el conjunt de validació al llarg de tot l'entrenament.

A la Figura 13 es mostra l'evolució de la funció de pèrdua (*loss*) i de l'*accuracy* tant per al conjunt d'entrenament com per al conjunt de validació al llarg de les èpoques.

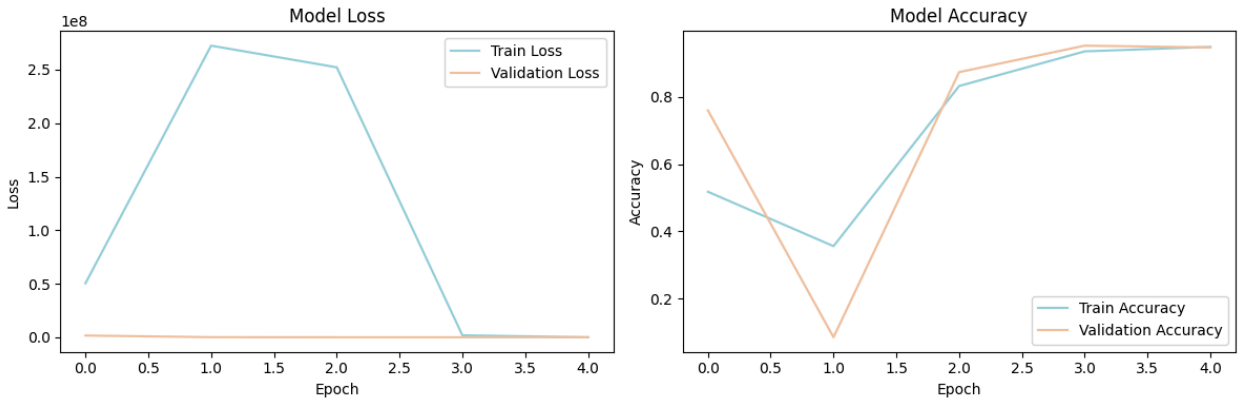

Figura 15: Evolució de *loss* i *accuracy* durant l'entrenament de CNN-LSTM en la classificació categòrica

4 RESULTATS

En aquest apartat, recollim els resultats obtinguts per a les tasques de classificació 1 i 2, amb els diferents models usats en aquest treball. S'han avaluat les tasques 1 i 2 de manera independent.

Les mètriques que s'han analitzat són les següents:

- Accuracy: indica quants paquets s'han classificat correctament respecte al total de paquets .
- Precision: representa quants paquets d'una classe (atac, no atac a la Tasca 1, i tipologia d'atac a la Tasca 2) han estat classificats correctament pel model respecte al total de paquets que el model ha classificat com a pertanyents a aquesta classe. A partir de la precision obtinguda per a cada classe d'una tasca, es calcula la mitjana.
- Recall: representa quants paquets d'una classe han estat classificats correctament pel model, respecte al total de paquets d'aquesta classe en el conjunt de dades. A partir de la recall obtinguda per a cada classe d'una tasca, es calcula la mitja.
- F1-Score: Combinant la precision i la recall, ens proporciona informació referent al rendiment del model.

En tots els models, es mostra la mitjana ponderada (*weighted_average*) de totes les classes per a la *precision*, la *recall* i el *f1-score*.

Primerament, es mostren els resultats dels models d'aprenentatge supervisat, Naive Bayes,, Random Forest i Regressió logística. En les taules següents es poden veure els resultats de cada un dels classificadors explicats anteriorment (veure secció 3.3.1).

La Taula 3 mosra els resultats de la Tasca 1 de classificació binària. La Taula 4 mostra els resultats de la Tasca 2 de classificació categòrica.

Model	Accuracy	Precision	Recall	F1-score
Naive Bayes	0,96430	0,96843	0,96430	0,96551
Random Forest	0,98799	0,98855	0,98799	0,98815
Regressió logística	0,98441	0,98611	0,98441	0,98479
Support Vector M	0,98488	0,98650	0,98488	0,98524

Taula 3: Resultats dels models d'aprenentatge supervisat classificació binària.

Model	Accuracy	Precision	Recall	F1-score
Naive Bayes	0,92527	0,95745	0,92527	0,94027
Random Forest	0,97019	0,96752	0,97019	0,96750
Regressió logística	0,97209	0,97037	0,97209	0,97039
Support Vector M	0,95822	0,94709	0,95822	0,95165

Taula 4: Resultats dels models d'aprenentatge supervisat classificació multiclasse.

Seguidament, es mostren els resultats dels models d'aprenentatge no supervisat, K-Means i DBSCAN, a partir dels resultats obtinguts per als classificadors descrits (veure secció 3.3.2). La Taula 5 reflecteix els resultats de la Tasca 1 i la Taula 6 els de la Tasca 2.

Model	Accuracy	Precision	Recall	F1-score
K-Means	0,88500	0,86375	0,88500	0,85508
DBSCAN	0,99637	0,99647	0,99637	0,99639

Taula 5: Resultats dels models d'aprenentatge no supervisat classificació binaria.

Model	Accuracy	Precision	Recall	F1-score
K-Means	0,87387	0,76365	0,87387	0,81505
DBSCAN	0,97707	0,96233	0,97707	0,96865

Taula 6: Resultats dels models d'aprenentatge no supervisat classificació multiclasse.

Finalment, es mostren els resultats dels models d'aprenentatge profund, DBN i CNN-LSTM, a partir dels resultats obtinguts per als classificadors descrits (veure secció 3.3.3).

La Taula 7 reflecteix els resultats de la Tasca 1 i la Taula 8 els de la Tasca 2.

Model	Accuracy	Precision	Recall	F1-score
DBN	0,98367	0,98547	0,98367	0,98408
CNN-LSTM	0,98472	0,98620	0,98472	0,98506

Taula 7: Resultats dels models d'aprenentatge profund classificació binaria.

Model	Accuracy	Precision	Recall	F1-score
DBN	0,87387	0,76365	0,87387	0,81505
CNN-LSTM	0,94718	0,93960	0,94718	0,94234

Taula 8: Resultats dels models d'aprenentatge profund classificació multiclasse.

A continuació, es proporcionen unes gràfiques per tal d'oferir una comparativa entre els resultats obtinguts en cadascun dels models d'aprenentatge supervisat, no supervisat i profund. Cada figura reflecteix els resultats dels diferents models per a una de les tasques. La Figura 14 correspon als resultats obtinguts en la tasca de classificació binària (Tasca 1) i la Figura 15 als obtinguts en la tasca de classificació multiclasse (Tasca 2). Com s'ha mencionat anteriorment, aquests resultats corresponen a les mitjanes ponderades de totes les classes.

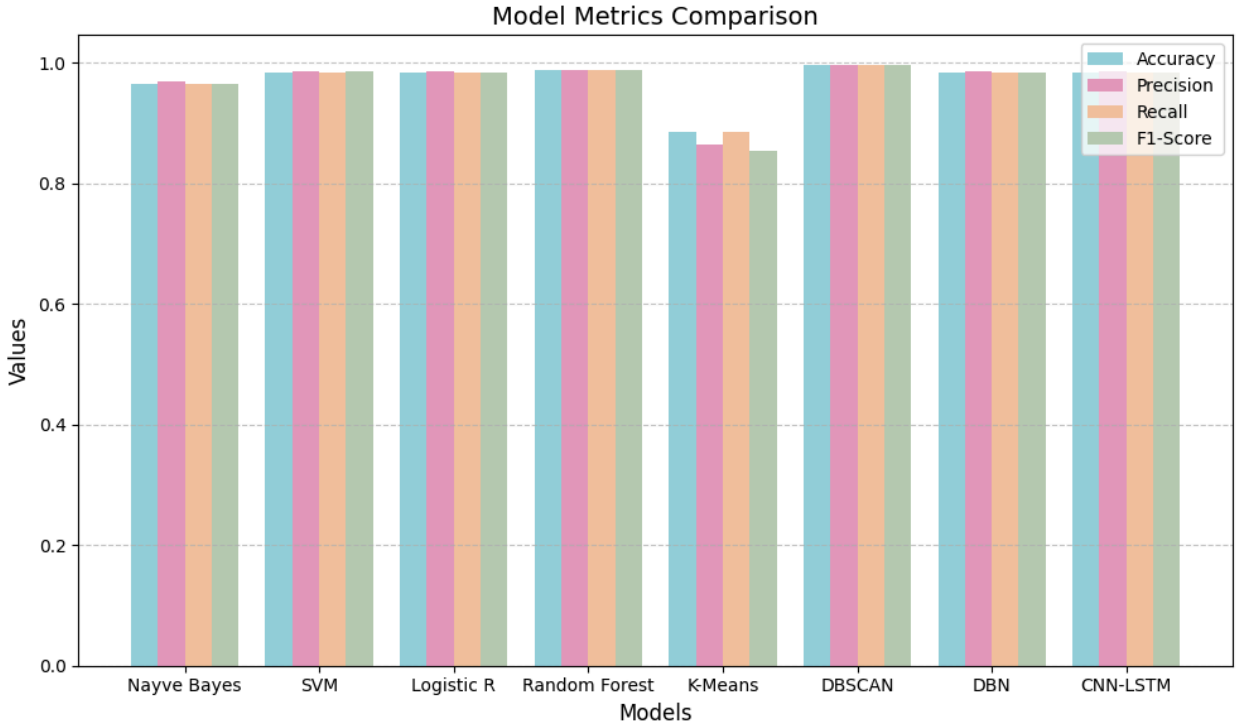

Figura 14: Comparativa resultats Tasca 1 (classificació binària)

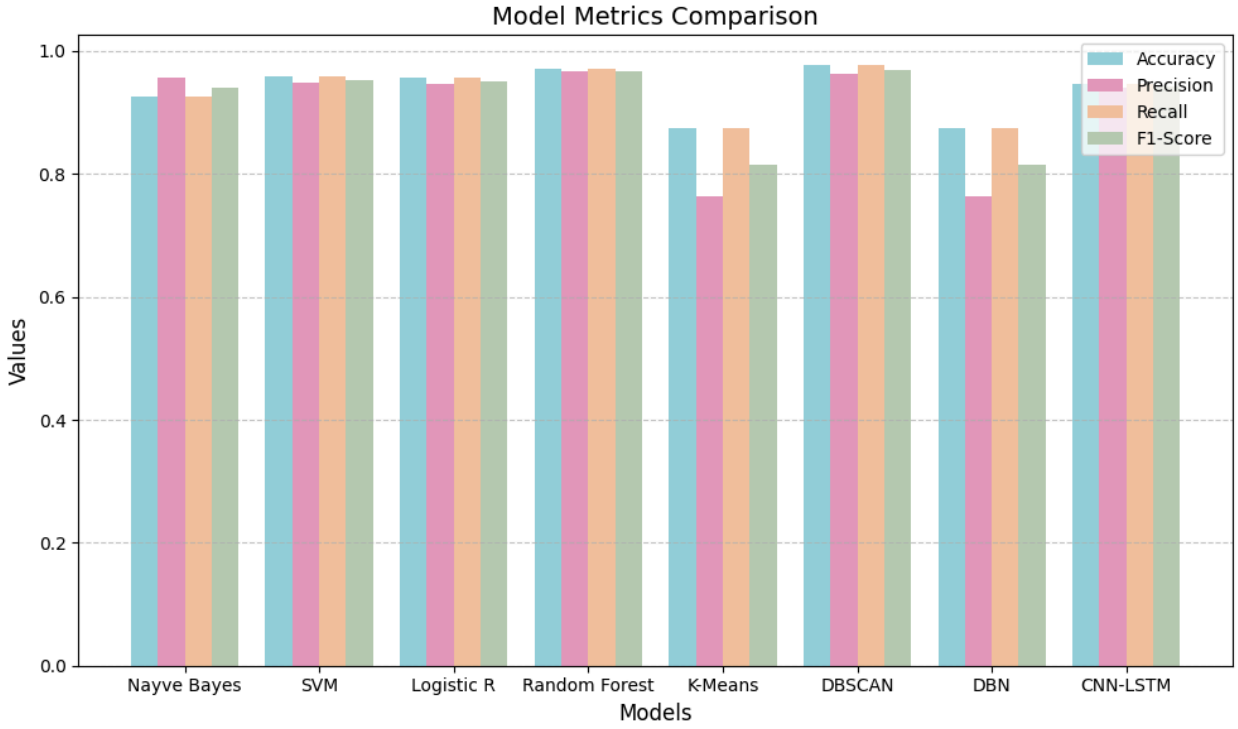

Figura 15: Comparativa resultats Tasca 2 (classificació categòrica)

A la Figura 16 i a la Figura 17 es mostren les matrius de confusió (confusion matrix) de la Tasca 1 i 2, respectivament, utilitzant l'algorisme Random Forest Classifier. Es poden observar els resultats de la classificació comparant les prediccions del model amb les etiquetes reals per a cada classe. En ambdues figures, les cel·les amb un nombre més alt de prediccions es destaquen en color taronja.

A la Figura 16, es pot veure que predominen els encerts positius (True Positive) per a la classe "normal" o "no atac". A més, també es destaca un valor elevat d'encerts negatius (True Negative) per a la classe "atac". Els falsos positius i falsos negatius són relativament baixos, indicant que les prediccions són força encertades per a ambdues classes.

A la Figura 17, predominen els encerts per a la classe "normal". Tot i que la classe "generic" no està destacada en taronja, s'observa clarament que la majoria de les dades classificades com a "generic" realment pertanyien a aquesta classe, amb pocs casos erronis. En canvi, per a les altres classes, les prediccions no són tan consistents i es poden veure algunes discrepàncies més evidents entre les prediccions i les etiquetes reals.

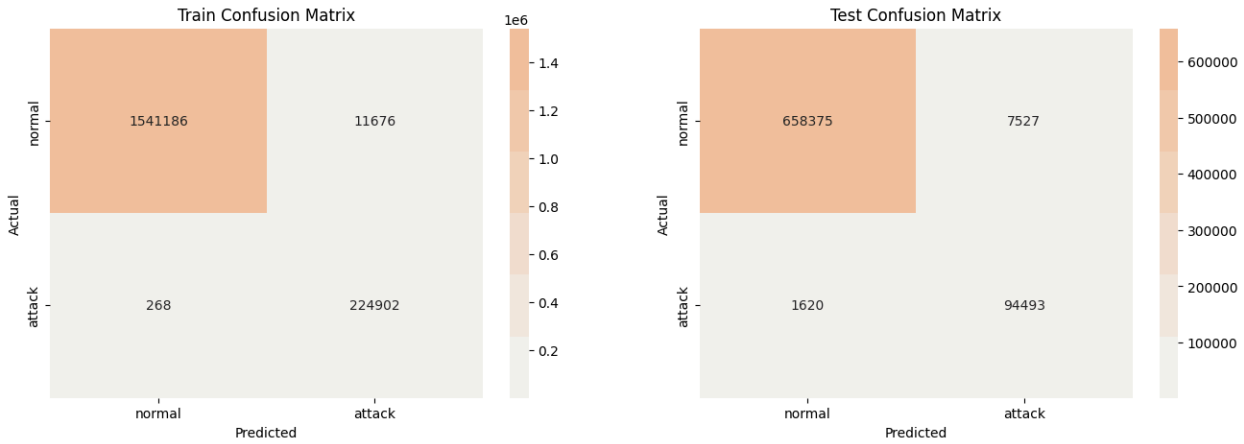

Figura 16: Confusion matrix Tasca 1 (classificació binària)

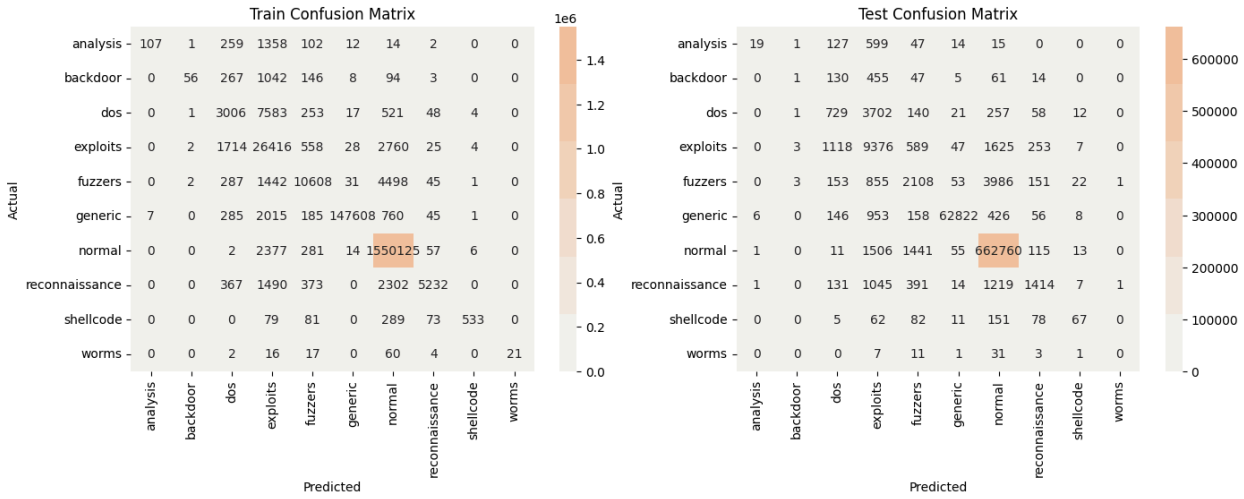

Figura 17: Confusion matrix Tasca 2 (classificació categòrica)

5 CONCLUSIONS

El present treball ha desenvolupat una solució basada en tècniques de Machine Learning per a la detecció d'activitats a la xarxa que podrien comprometre la seguretat informàtica d'una organització. S'ha aconseguit classificar comportaments sospitosos i identificar possibles atacs mitjançant diversos models d'aprenentatge supervisat, no supervisat i profund. A través de l'anàlisi de les dades del conjunt UNSW-NB15, s'ha creat un sistema capaç de detectar comportaments anòmals i associar-los a diferents tipus d'atacs cibernètics actuals. En total, s'han establert nou categories d'atac: Fuzzers, Analysis, Backdoors, DoS, Exploits, Reconnaissance, Shellcode, Generic i Worms.

Durant el procés, s'ha implementat una estratègia exhaustiva de preprocessament de dades, que ha inclòs l'escalat de característiques, la vectorització de dades categòriques i la reducció de dimensionalitat, la qual ha estat fonamental per millorar tant l'eficiència computacional com la precisió dels models. A més, l'ús de tècniques com la validació creuada ha permès optimitzar els models i seleccionar els millors paràmetres per a cada algorisme.

Els resultats obtinguts en tots els models d'aprenentatge són satisfactoris, ja que, d'acord amb la mètrica *accuracy*, 6 dels 8 models analitzats han aconseguit un percentatge d'encerts superior al 98% en la tasca de classificació binària. Pel que fa a la tasca de classificació multiclasse, 6 dels 8 models han obtingut resultats superiors al 92%.

En la comparativa de resultats que es presenta en les conclusions finals, no s'inclourà l'algorisme DBSCAN, ja que la seva execució ha requerit una quantitat considerable de recursos, fet que ha obligar a realitzar la prova amb una mostra reduïda del conjunt de dades original. Per aquest motiu no seria gaire equitatiu comprar aquest resultats amb el resta. No obstant això, es pot concloure que DBSCAN ofereix molt bons resultats en la seva aplicació.

En la Tasca 1 de classificació entre atac i no atac, el millor resultat ha estat l'obtingut pel model d'aprenentatge supervisat Random Forest, amb una *accuracy* propera al 99%.

En la Tasca 2 de classificació multiclasse, que inclou 10 categories (9 corresponents a atacs i 1 a comportament normal o no atac), el millor resultat ha estat obtingut mitjançant el model d'aprenentatge supervisat Regressió Logística, amb una *accuracy* propera al 97,21%. El resultat més baix, tot i ser un bon resultat amb un 87,39%, es correspon a l'algorisme K-Means, aplicat en l'aprenentatge no supervisat.

L'algorisme K-Means, utilitzat per identificar patrons en el trànsit de xarxa i detectar comportaments anòmals sense la necessitat d'etiquetar prèviament les dades, ha obtingut resultats inferiors en comparació amb la resta d'algorismes en ambdues tasques de classificació.

D'altra banda, malgrat la qualitat del conjunt de dades utilitzat, cal destacar que la manca d'equilibri entre les classes d'atac i les de comportament normal pot afectar negativament els resultats obtinguts, especialment en les

mètriques de *precision*, *recall* i *f1-score* per a cada categoria. Així, tot i que les mitjanes ponderades de les mètriques per a la Tasca 2 han estat satisfactòries, cal subratllar que els resultats no han estat igual de positius per a totes les categories. Els millors resultats s'han obtingut en les dues classes amb una major presència al conjunt de dades (categoria *normal*, amb un 87,3% de presència, i *generic*, amb un 8,7%, tal com es mostra a la Figura 5). En canvi, en les altres categories, amb una presència inferior al 2%, els resultats han estat menys favorables. És possible que, amb un conjunt de dades més equilibrat, els resultats fossin millors en tots els casos. Aquest aspecte podria ser abordat mitjançant tècniques de balanceig de dades.

En resum, aquest treball aporta una contribució significativa al camp de la seguretat informàtica, ja que proporciona una solució que millora la detecció de comportaments sospitosos i atacs a la xarxa mitjançant l'ús de models de Machine Learning. La solució desenvolupada podria ser integrada en sistemes de seguretat informàtica per ajudar a prevenir situacions crítiques, com el robatori de dades o el sabotatge de la infraestructura tecnològica d'una organització.

Els resultats obtinguts amb els models implementats mostren una detecció precisa dels atacs a la xarxa. En general, aquest treball demostra que els models de Machine Learning són una eina eficaç per millorar la seguretat informàtica, permetent la detecció proactiva de vulnerabilitats i amenaces.

6 PROJECTES FUTURS

Partint dels sistemes de detecció i classificació d'atacs a la xarxa, identificats en aquest treball, plantegem diverses millores, així com propostes per a projectes futurs.

Entre les possibles millores, contemplem les següents:

- Disposar d'un conjunt de dades més equilibrat. El conjunt de dades actual, tot i ser de qualitat i extens, pot ser ampliat amb més exemples d'atacs, així com amb més dades generades per a equilibrar les classes, millorant així la capacitat del model per detectar tant atacs comuns com rareses o nous tipus d'atacs.

- Recollir dades en entorns de xarxes en temps real. Seria beneficiós utilitzar dades de xarxes en temps real per a simular escenaris actuals i canvis en els patrons de trànsit associats a nous tipus d'atacs. A més, una avaluació contínua i actualització dels models seria essencial per garantir la seva eficàcia a llarg termini.

- Realitzar una parametrització més extensa dels models utilitzats. Experimentar una combinatòria més àmplia en la configuració dels models utilitzats podria millorar el seu rendiment.

- Ampliar l'estudi actual amb més sistemes de classificació. Existeix un ventall molt ampli d'algorismes de ML i d'aprenentatge profund. Un estudi més exhaustiu i complet permetria explorar i comparar diferents mètodes i possiblement millorar les mètriques obtingudes en alguns casos.

- Ampliar l'estudi actual per incloure altres tipus de comportaments que puguin comprometre la ciberseguretat, com ara la detecció d'abusos per part d'usuaris interns o la identificació de fugues de dades. Això permetria reforçar i diversificar les capacitats de l'actual sistema de detecció d'atacs més enllà dels incidents exclusivament relacionats amb la xarxa.

- Fomentar la col·laboració amb altres investigadors i organitzacions. Crear fòrums de col·laboració per compartir coneixement i desenvolupar conjunts de dades més representatius.

- Integració amb eines existents de seguretat. Els sistemes desenvolupats podrien integrar-se amb solucions ja existents de seguretat com firewalls o IDS/IPS (Sistemes de detecció/previsió d'intrusions), millorant la capacitat de resposta de les xarxes davant d'atacs.

- Implementar la detecció en temps real. Implementar els sistemes de manera que siguin capaços de detectar i respondre als atacs en temps real, minimitzant els danys en les infraestructures.

- Desenvolupar eines de visualització i interpretabilitat: Crear eines visuals per millorar la comprensió de resultats i facilitar la presa de decisions dels administradors de seguretat.

7 REFERÈNCIES

[1] Yavanoglu, O., & Aydos, M. (2017). A review on cyber security datasets for machine learning algorithms. Proceedings of the IEEE International Conference on Big Data, 825-8167. https://doi.org/10.1109/BigData.2017.8258167

[2] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., & Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers & Security, 86, 147-167. https://doi.org/10.1016/j.cose.2019.06.005

[3] Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Communications Surveys & Tutorials, 18(2), 1153-1176. https://doi.org/10.1109/COMST.2015.2494502

[4] Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In 2010 IEEE symposium on security and privacy (pp. 305-316). IEEE.

[5] Kent, T., & Matarazzo, C. (2020). Applying Unsupervised Machine Learning Techniques to Detect Insider Threats. Journal of Cyber Security and Information Systems, 8(1).

[6] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 dataset. In 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (pp. 1-6). https://doi.org/10.1109/CISDA.2009.5356528

[7] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)."Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015. https://doi.org/10.1109/MilCIS.2015.7348942

[8] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set." Information Security Journal: A Global Perspective (2016): 1-14

[9] M. Turcotte, A. Kent and C. Hash, “Unified Host and Network Data Set”, in Data Science for Cyber-Security. November 2018, 1-22

[10] Ahsan M, Nygard KE, Gomes R, Chowdhury MM, Rifat N, Connolly JF. Cybersecurity Threats and Their Mitigation Approaches Using Machine Learning—A Review. Journal of Cybersecurity and Privacy. 2022; 2(3):527-555. https://doi.org/10.3390/jcp2030027

[11] Gao, N.; Gao, L.; Gao, Q.; Wang, H. An intrusion detection model based on deep belief networks. In Proceedings of the 2014 Second International Conference on Advanced Cloud and Big Data, Huangshan,

China, 20–22 November 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 247–252. https://ieeexplore.ieee.org/abstract/document/7727705

[12] Alrawashdeh, K.; Goldsmith, S. Optimizing Deep Learning Based Intrusion Detection Systems Defense Against White-Box and Backdoor Adversarial Attacks Through a Genetic Algorithm. In Proceedings of the 2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington, DC, USA, 13–15 October 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8. https://ieeexplore.ieee.org/abstract/document/9425293

[13] Ahsan, M.; Nygard, K.E. Convolutional Neural Networks with LSTM for Intrusion Detection. CATA 2020, 69, 69–79. https://d1wqtxts1xzle7.cloudfront.net/102656347/cXbs-libre.pdf

[14] Millar, S.; McLaughlin, N.; del Rincon, J.M.; Miller, P. Multi-view deep learning for zero-day Android malware detection. J. Inf. Secur. Appl. 2021, 58, 102718. https://www.sciencedirect.com/science/article/abs/pii/S2214212620308577

[15] B. F. Hamouda, M. L. Sabry, M. B. A. Friha, i R. Khuri. "Revolutionizing Cyber Threat Detection with Large Language Models: A privacy-preserving BERT-based Lightweight Model for IoT/IIoT Devices." ArXiv, 2023. https://arxiv.org/abs/2306.14263

[16] X. Yang, X. Zhang, W. Wang, et al. "Research on Anomaly Detection Method Based on DBSCAN Clustering Algorithm." 2021 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), 2021. IEEE Xplore, https://ieeexplore.ieee.org/document/9363764.

[17] P. Lou, Q. Yu, X. Yang, J. Zhang, y M. Li. "Cyber intrusion detection through association rule mining on multi-source logs." Journal of Ambient Intelligence and Humanized Computing, 2022. https://link.springer.com/article/10.1007/s12652-022-03760-6

[18] Ravipati, R.D.; Abualkibash, M. Intrusion detection system classification using different machine learning algorithms on KDD-99 and NSL-KDD datasets—A review paper. Int. J. Comput. Sci. Inf. Technol. 2019, 11, 65–80. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3428211

[19] Kumar, A., & Singh, G. (2020). Improving security using SVM-based anomaly detection: Issues and challenges. Soft Computing, 24(15), 11391-11403. https://doi.org/10.1007/s00500-020-05373-x

[20] Almaleh A, Almushabb R, Ogran R. Malware API Calls Detection Using Hybrid Logistic Regression and RNN Model. Applied Sciences. 2023; 13(9):5439. https://doi.org/10.3390/app13095439

[21] Panda, M.; Patra, M.R. Network intrusion detection using naive bayes. Int. J. Comput. Sci. Netw. Secur. 2007, 7, 258–263. https://d1wqtxts1xzle7.cloudfront.net/33414102/20071238-libre.pdf

[22] Creech, G., & Smith, J. (2013). ADFA Intrusion Detection Datasets. Australian Centre for Cyber Security, Australian Defence Force Academy. https://learn.saylor.org/mod/book/view.php?id=29755&chapterid=5445

[23] Agència de Ciberseguretat de Catalunya. Ciberseguretat a Catalunya: Informe tecnològic 2022. Generalitat de Catalunya, 2022. https://ciberseguretat.gencat.cat/web/.content/PDF/Ciberseguretat_Informe-tecnologic-2022.pdf

[24] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830. https://scikit-learn.org

[25] The pandas development team. (2023). pandas (Version 2.1.1) [Software]. https://pandas.pydata.org/

[26] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., ... Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[27] Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

[28] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ... SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods, 17(3), 261–272. https://doi.org/10.1038/s41592-019-0686-2

[29] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). TensorFlow: A System for Large-Scale Machine Learning. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), 265–283. https://www.tensorflow.org

[30] Python Software Foundation. (2001). *Python Standard Library Documentation*. Available at https://docs.python.org/3/

[31] Waskom, M. L. (2021). Seaborn: statistical data visualization. *Journal of Open Source Software, 6*(60), 3021.

[32] Tarlogic. (n.d.). *Fuzzing: Qué es y cómo se usa en ciberseguridad*. Tarlogic. Retrieved November 20, 2024, from https://www.tarlogic.com/es/glosario-ciberseguridad/fuzzing/

[33] Delta Protect. (n.d.). *Backdoor o puerta trasera*. Delta Protect. Retrieved November 20, 2024, from https://www.deltaprotect.com/blog/backdoor-o-puerta-trasera

[34] Cloudflare. (n.d.). *What is a denial-of-service (DoS) attack?*. Cloudflare. Retrieved November 20, 2024, from https://www.cloudflare.com/learning/ddos/glossary/denial-of-service/

[35] EasyDMARC. (n.d.). *Los 10 tipos de ataques cibernéticos más comunes*. EasyDMARC. Retrieved November 20, 2024, from https://easydmarc.com/blog/es/los-10-tipos-de-ataques-ciberneticos-mas-comunes/

[36] eSecurity Planet. (n.d.). *How Hackers Use Reconnaissance*. eSecurity Planet. Retrieved November 20, 2024, from https://www.esecurityplanet.com/threats/how-hackers-use-reconnaissance/

[37] KeepCoding. (n.d.). *Qué es el Shellcode*. KeepCoding. Retrieved November 20, 2024, from https://keepcoding.io/blog/que-es-el-shellcode/

[38] CrowdStrike. (n.d.). *What is a Computer Worm?*. CrowdStrike. Retrieved November 20, 2024, from https://www.crowdstrike.com/en-us/cybersecurity-101/malware/computer-worm/

[39] Maji, S. (n.d.). Building an intrusion detection system on UNSW-NB15 dataset based on machine learning algorithm. Medium. Retrieved January 7, 2025, from https://medium.com/@subrata.maji16/building-an-intrusion-detection-system-on-unsw-nb15-dataset-based-on-machine-learning-algorithm-16b1600996f5

8 ANNEXES

Per obtenir el codi i les dades emprades en aquest treball feu clic a l'enllaç següent:
https://github.com/ainamonch/network-intrusion-detection-TFM