Detecció d'atacs de Ransomware amb Machine Learning

David Sardà Martin

Resum– En els darrers anys, els atacs de malware de la família Ransomware han aumentat molt i han tingut conseqüències molt greus per a individus, organitzacions i institucions. Un clar exemple és el recent cas en la Universitat Autònoma de Barcelona on els alumnes i professorat van viure les greus afectacions. Per això resulta de gran importància buscar alguna solució a aquest tipus d'atacs i el primer pas implica detectar-los. Es buscarà crear models que permetin detectar atacs d'aquesta tipologia basant-se en alguns paràmetres de memòria en els dispositius a través de l'aprenentatge computacional, una eina que s'ha destapat molt útil per predir o detectar valors o estats en molts àmbits de coneixement. Es crearà seguint una metodologia Scrumban: un model weak-learner, un strong-learner i una xarxa neuronal creada amb Pytorch que detectaran atacs de Ransomware.

Paraules clau– Ransomware, detecció, Machine Learning, model, Xarxa Neuronal, PyTorch, sklearn.

Abstract– Last years, malware attacks from the Ransomware family have increased and have had major consequences for individuals, organizations and institutions. A clear example is the recent case at the Universitat Autònoma de Barcelona with the students and professors who suffer the consequences. Because of that it is important to look for a solution of this types of attacks and the first step involves detecting them. This project pretend to create models that allow detect attacks of this typology based on some memory parameters in the device through machine learning, which has been proved to be very useful for predicting or detecting values or stats in many areas of knowledge. We will create this models following a Scrumban methodology: a weak-learner model, a strong-learner model and a neural network created with Pytorch that will detect Ransomware attacks.

Keywords– Ransomware, detection, Machine Learning, model, neural network, PyTorch, sklearn.

1 INTRODUCCIÓ

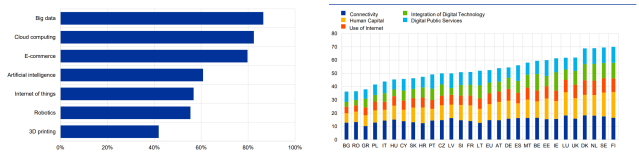
En els últims anys, el Machine Learning s'ha destapat com una eina molt útil per la predicció de valors, accions o tota mena d'aspectes, exemples en són l'ús per identificar possibles criminals o ubicacions de crims, classificar peces de roba, per la presa de decisions en organitzacions...

La informàtica és cada vegada més present a les nostres vides, és per això també totes les dades i registres són digitalitzats de forma que facilita a tots els usuaris treballar amb dades. Les universitats, els bancs, els departaments de policia i moltes altres organitzacions i institucions s'han adaptat a aquesta nova era digitalitzant els serveis i les dades que necessiten. És per això, que els criminals s'han adaptat a les noves tecnologies i han vist noves oportunitats per atacar i treure benefici a través de la informàtica.

A continuació es pot veure el resultat d'una enquesta realitzada a treballadors sobre si les seves empreses han adoptat certs tipus de tecnologia (Figura 1. a), i també índexs que mostren la creixent digitalització del món (Figura 1. b):

Un exemple d'aquesta modernització d'atacs són els atacs de Rasnomware. Les conseqüències d'aquests poden ser molt greus, ja que quan es produeixen, també es poden veure afectats tots els dispositius connectats al sistema, això implica que tant clients com treballadors d'una empresa es poden veure implicats. A més, un cop les dades són capturades per un atac Ransomware són gairebé impossibles de recuperar, només es poden recuperar amb una còpia de seguretat.

• E-mail de contacte: 1492054@uab.cat
• Menció realitzada: Enginyeria de Tecnologies de la Informació
• Treball tutoritzat per: Joan Protasio (deic)
• Curs 2021/22



(a) Adopció de tecnologies digitals a empreses.

(b) Índex d'Economia i Societat Digital per al 2019.

Fig. 1: Creixement de l'ús de tecnologies digitals a les empreses.

El problema resideix en el fet que encara que recuperis la còpia de seguretat, el sistema està infectat i cal eliminar tot el malware del sistema. És per això, que quan es produeixen aquests atacs a organitzacions sovint porta a la paralització d'alguns sistemes per analitzar on s'ha produït l'atac i/o per la restauració del sistema, i aquesta paralització encara que es recuperi l'empresa, no és ràpid i, per tant, pot tenir forts impactes en l'economia de l'empresa.

Recentment, algunes universitats han vist de primera mà aquests atacs informàtics, així com nombroses organitzacions: centres mèdics, tribunals de justícia, clubs de futbol, empreses d'energia... [2]

Farem servir les eines de Machine Learning per identificar atacs de tipus Ransomware que s'estiguin produint en un sistema informàtic. D'aquesta forma, un cop identificats l'equip de resposta podrà intentar evitar-los o protegir el sistema per evitar mals majors.

2 Objectius

En aquest projecte es buscarà trobar un model amb Machine Learning amb una alta capacitat per a detectar ràpidament atacs de tipus Ransomware, de forma que les víctimes de l'atac o l'equip de resposta puguin buscar formes de defensar l'atac o prendre les mesures més adequades per evitar danys significatius a l'organització i als seus clients.

Per a assolir aquest objectiu caldrà marcar unes pautes a seguir que definiran alguns nous objectius o subobjectius necessaris:

- Entendre amb major profunditat que són els atacs de Ransomware.
- Caldrà definir una metodologia de treball, es buscarà simular, en la mesura del possible, que aquest projecte és part d'una empresa, tot i que en aquest cas serà individual. En aquest cas treballarem amb les metodologies Agile i familiaritzar-se amb un framework que ens permet implementar aquesta metodologia de treball.
- Per a realitzar els models caldrà aprendre i entendre les bases del Machine Learning, a més d'alguns conceptes molt utilitzats per la majoria d'algoritmes.
- S'implementarà un model de xarxa neuronal amb tensors. Per poder crear-la caldrà aprendre a treballar amb les llibreries de tensors i teoria sobre les xarxes neuronals.
- Caldrà aprendre tècniques d'optimització d'hiperparàmetres de Machine Learning, per a poder obtenir el millor model possible.
- Caldrà entendre algunes mètriques per mesurar el rendiment dels algoritmes de Machine Learning, com per exemple la precisió, el recall, la matriu de confusió, la corba roc, el temps...
- Creació d'un repositori Github complet amb els models, presentació de resultats i altres fitxers que ajudin a l'usuari a utilitzar o entendre els models.

3 Metodologia

La metodologia per a la realització d'un projecte és molt important, s'ha vist com aplicar metodologies de treball ha incrementat la coordinació de l'equip i el seu rendiment, i d'aquesta forma aconseguir que més projectes siguin exitosos.

Els mètodes Agile són un exemple d'aquests sistemes de treball, ofereixen als equips una forma flexible de treballar repartint les tasques ràpidament i de forma dinàmica entre els equips o treballadors que s'encarreguen del projecte. En els darrers anys les metodologies Agile s'han estat utilitzant cada vegada més gràcies als bons resultats que estan donant, empreses com Google, Amazon i Microsoft les utilitzen.

Principalment, es basarà en dues metodologies Agile ben conegudes: Scrum i Kanban, de fet a moltes empreses ja s'ha utilitzat una metodologia aprofitant aspectes d'aquestes dues metodologies i se n'ha anomenat Scrumban, però la definició de com és l'Scrumban no és única, cada empresa pot elegir els aspectes més importants o que consideren que s'ajusten millor a les seves necessitats per crear la metodologia que considerin més adequada pel seu projecte.

La metodologia seguida es pot entendre amb el següent gràfic, cada Sprint durarà dues setmanes, d'aquesta forma es faran revisions de forma bastant continuada que asseguraran que el projecte evolucioni correctament.

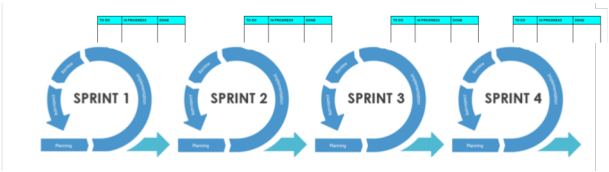


Fig. 2: Metodologia.

Dins de cada sprint, les tasques passaran per les següent etapes:

Fig. 3: Etapes de cada tasca.

Per a treballar amb la metodologia Scrumban, s'utilitzarà el Software de Jira seleccionant la metodologia Scrum, tot i que s'adaptarà per utilitzar Kanban a l'interior de cada Sprint.

4 Estat de l'art

Avui en dia són nombrosos els estudis de Machine Learning i Deep Learning per detectar malware. Decidir quin algo-

ritme fer servir pot ser determinant per tenir uns millors o pitjors resultats de la mateixa manera que els paràmetres utilitzats pels algoritmes poden condicionar a les diferents mètriques que es poden utilitzar [3].

Per la detecció de malware hi ha diferents aproximacions a considerar que es poden utilitzar per analitzar un sistema [5]:

Estàtic:aquest tipus d'anàlisi agafa unes dades que prèviament s'han recopilat i fa una anàlisi. Per realitzar-lo s'han de crear o extreure les dades d'alguna forma, suposant o inventant sovint situacions o aspectes. No serà molt adequat quan el dispositiu respongui de diferent forma de les situacions creades.

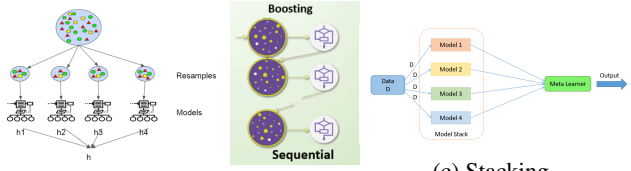
Dinàmic: es tracta d'analitzar amb l'algoritme dades que s'estan produint a temps real. Amb això s'aconsegueix simular una situació més similar a la real, cal tenir en compte que aquests tipus d'anàlisi requereixen més temps. Si el que es vol és detectar l'atac ràpidament per poder fer una resposta com més aviat millor, no és el tipus d'anàlisi més adequat pel nostre problema.

Híbrid: aquest intenta combinar aspectes dels dos anteriors, de forma que intenta aprofitar els avantatges de cada un d'ells.

Entre els diferents algoritmes d'aprenentatge computacional es poden distingir entre 2 tipus [1]:

Weak-learners: són algoritmes que són relativament senzills, no suposen una gran complexitat computacional i que mostren un resultat lleugerament millor que intentar predir de forma aleatòria.

Strong-learners: són algoritmes sovint més complexos que combinen diferents weak-learners per a crear un millor classificador o predictor. Entre ells podem distingir tres tipus diferents:



(a) Bagging. (b) Boosting. (c) Stacking.

Fig. 4: Tipus d'algoritmes strong-learners.

En el nostre cas s'utilitzarà un de cada tipus entre els strong-learners i els weak-learners per poder comparar com funcionen cada un d'ells.

Alguns estudis ja fets demostren que les deteccions es poden fer basades en signatura o també analitzant el tràfic anormal [4], el problema principal de la majoria d'aproximacions d'algoritmes de deteccions de malware són que contínuament aquests atacs es van actualitzant, de forma que un patró que permet detectar aquests atacs, posteriorment és modificat i aquest patró desapareix, el constant canvi provoquen que la detecció d'aquest atac o de la majoria de malware es dificulti [3].

Els atacs de Ransomware consisteixen essencialment a segrestar dades, és a dir, habitualment es tracta d'un malware que accedeix a dades de l'usuari del dispositiu o del sistema de dispositiu i les encripta, que és el cas més habitual, o les roba i posteriorment es demana un rescat o es fa extorsió amb les dades.

Un cas molt proper va ser l'atac a la Universitat Autònoma de Barcelona (UAB), aquest atac que es va produir la matinada de l'onze d'octubre va utilitzar el malware PYSA (Protect Your System Amigo) va deixar molts serveis de la UAB sense disponibilitat: la xarxa interna no es podia utilitzar i per tant no es podien utilitzar els dispositius del centre, el WIFI per evitar la propagació del malware, alguns dispositius personals d'alumnes i professorat van quedar infectats també, el Campus Virtual, que serveix per a la interacció d'alumnes i professors va quedar inhabilitat, els correus associats a la Universitat no s'hi podia accedir i molts serveis de l'administració de la UAB van quedar inhabilitats com per exemple l'assignació de tutors dels treballs de fi de grau, programes de reconeixement de crèdits, sol·licituds d'erasmus, cobrament de matrícules... Es va saturar la xarxa de mòbils, ja que hi havia molts usuaris intentant accedir a Internet a través de les dades dels mòbils, la connexió era més lenta i inclús en alguns punts del campus costava molt accedir a Internet.

Tot el procés de recuperació va tardar un temps a causa de la seva complexitat com va ja preveure i avisar Jordi Serra, professor de la UOC expert el ciberseguretat: "Es muy complejo dejarlo todo limpio y hay que tener esta certeza antes de abrir la puerta de entrada y salida de la red interna" [7], era molt difícil detectar exactament quins sistemes estaven afectats, si alguns routers o còpies de seguretat es podien veure afectats: "Y no se puede formatear todo. Hay que ir aislando ordenadores para ver cómo se comportan". La UAB va avisar via Twitter de l'atac informàtic i va mostrar transparència a l'hora d'elaborar un pla per contrarestar l'atac. Va crear un canal de Telegram, per avisar dels progressos sobre la recuperació de serveis. Els professors també van buscar altres vies per comunicar-se amb l'alumnat, Google Drive, One-drive, Microssoft Teams... La resposta per adaptar-se a la situació va ser relativament ràpida, però algunes eines utilitzades per alumnes i professors van tardar més a estar disponibles.

Encara no hi ha coneixement total de les conseqüències de l'atac, ja que no es pot descartar que els atacants puguin tenir dades sensibles i que pròximament aquestes puguin ser publicades o utilitzades per l'extorsió. Jordi Hernández ha explicat que el risc és residual, ja que afirma que els servidors atacats eren: "archivos de ofimática del día a día" i per tant els atacants no poden disposar de informació sensible rellevant. [8]

Un altre exemple d'atac de Ransomware que ens permet veure les afectacions tan greus que poden tenir aquesta família d'atacs és el Wannacry, un dels atacs de Ransomware més conegut i que va tenir més impacte. Va ser un atac que va afectar mundialment que va començar el 12 de maig de 2017, aquest atac va afectar més de 230.000 dispositius a 150 països en un sol dia que aprofita un error de Windows, coneguda com a MS17-010, per fer vulnerable el sistema. Amb aquest atac es van veure afectades moltes empreses, universitats i agències governamentals.

A continuació podem veure un mapa que mostra els atacs que es van produir de WannaCry:

L'atac afectava a més de 10.000 usuaris diàriament. Per sort, aquest atac es va poder frenar gràcies al fet que l'investigador de ciberseguretat Marcus Hutchins va detectar que cada cop que el malware entrava en un sistema intentava accedir a una determinada URL, si no la trobava alesho-

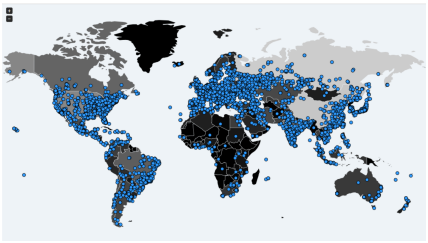


Fig. 5: Víctimes de l'atac massiu de Ransomware WannaCry el 12 de Maig de 2017.

res infectava el sistema. Hutchins va ser capaç de registrar un domini que va servir com Kill Switch per desactivar el WannaCry, els atacants van intentar atacar aquest domini per anul·lar-lo, però van aconseguir mantenir-lo i desactivar així el WannaCry.

De forma resumida, es pot observar que hi ha molts estudis i moltes alternatives per a detectar malware, el que s'aplica també a atacs de Ransomware que s'ha vist que poden afectar molt greument a les organitzacions, s'han fet estudis utilitzant diferents algoritmes i s'han obtingut resultats molts bons, per exemple amb KNN [6], Random Forest [5]... També s'ha pogut veure que l'anàlisi que sol donar millor resultats, en general, per la detecció de malware és el dinàmic [6]. Tot i així, a la realitat moltes vegades no és senzill, ja que es modifiquen aspectes del patró d'atac per evitar ser detectats. També cal tenir en compte que en un atac de Ransomware el temps serà un paràmetre important, cal respondre ràpidament per evitar qualsevol dany que pugui causar, però també serà important poder detectar els atacs amb una alta taxa d'èxit, per això caldrà tenir en compte ambdós aspectes.

5 RANSOMWARE

Si analitzem la pròpia paraula en si, ja podem tenir una idea que és un atac de Ransomware. La paraula Ransomware ve de la combinació de dues paraules: ransom, que en anglès significa rescat, i ware que prové de Software. Com es pot imaginar es tracta d'un tipus de malware que es particularitza per segrestar dades i demanar-ne posteriorment un rescat.

En general, els atacs de Ransomware per segrestar les dades, encripten les dades, aquest segrest de dades pot tenir diferents repercussions. També una altra forma d'extorsió que s'utilitza és amenaçar en publicar dades trobades en els dispositius que poden ser humiliants o comprometedores. Un cop les dades són segrestades deixen un missatge a l'usuari avisant de l'atac i demanant-ne un rescat, en els darrers anys, la pràctica més habitual és demanar un rescat en bitcoins o monedes virtuals, ja que són més difícils de rastrejar i, per tant, és més difícil descobrir el culpable.

En els atacs de Ransomware, el malware sol mantenir-se ocult fins que finalment quan l'atacant ja ha aconseguit accedir a dades sensibles o pot bloquejar el dispositiu i d'aquesta forma, els atacants ja poden fer l'extorsió. És per això que no és un atac senzill de detectar, ja que durant la majoria de l'atac el malware no genera cap efecte en el dispositiu.

Hi ha molts tipus de malware que s'utilitzen per introduir el malware per fer un atac Ransomware al dispositiu, per exemple troians o cucs són els més comuns, que per introduir-se un mètode molt comú és el phishing.

El cost d'un atac de Ransomware és difícil d'estimar ja que no s'ha de considerar només els diners que es puguin perdre de forma directa, sinó també les pèrdues de diners que comporta la paralització d'un sistema. Considerant això, el 2020 es va fer un estudi per predir els costos que el 2021 comportarien els atacs de Ransomware, que mostra els costos que en els darrers està comportant els atacs de Ransomware, així com el seu increment cada any:

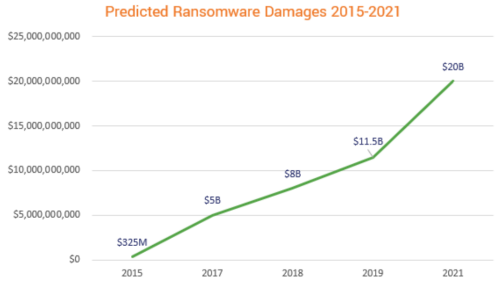


Fig. 6: Cost predit dels atacs de Ransomware el 2021 basat en els costs des del 2015.

5.1 METODOLOGIA

Els atacs de Ransomware solen seguir sempre els mateixos passos:

- L'atacant utilitza algun mètode per infectar el dispositiu, per exemple envia un correu perquè es descarreguin un fitxer o phishing són els més comuns.
- La víctima cau en la trampa i permet l'accés de malware que infecta el seu dispositiu, però no se n'adona.
- S'instal·la el Ransomware i es generen claus d'encriptació per encriptar les dades de l'usuari.
- El malware ataca el dispositiu sense que se n'adoni la víctima, fins a accedir a tots els fitxers o informació que vol.
- El malware mostra un avís per pantalla a la víctima que el seu dispositiu ha sigut hackejat i les seves dades han estat encriptades i posa una forma de rescatar les dades, pagant uns diners a alguna compte o amb bitcoins.
- En aquest punt, la víctima pot decidir fer cas i pagar el rescat o no pagar el rescat, però l'atac ja s'ha produït i el seu dispositiu està infectat.

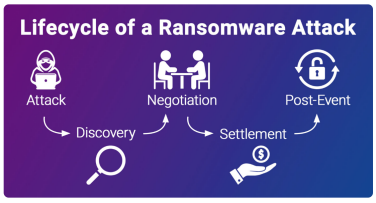


Fig. 7: Fases d'un atac de Ransomware.

5.2 TIPUS

En general es distingeixen entre tres tipus d'atacs de Ransomware:

- **De bloqueig**: aquest tipus es caracteritzen per bloquejar parts o tot el sistema informàtic, fent que l'usuari no pugui treballar amb el dispositiu. En ells sol aparèixer un missatge a la pantalla de l'usuari on se l'avisa de l'atac i es demana un rescat.
- **De xifratge**: es tracta de xifrar arxius de l'usuari del dispositiu, com les fotos, documents o altres de l'usuari. També en cas que continguin bases de dades poden xifrar-les afecten als clients, en cas que en tinguin. Habitualment s'utilitzen criptografia asimètrica avui en dia.
- **Híbrids**: aquest tipus combina característiques dels dos anteriors.

5.3 MESURES

En els atacs de Ransomware, un cop el teu dispositiu es troba infectat és molt complicat d'arreglar l'atac, en cap cas es recomana el pagament del rescat que demanen els atacants no hi ha cap garantia que les dades seran retornades. A més, també es recomana desconnectar el dispositiu de la xarxa per evitar que altres dispositius puguin ser atacats. Les principals mesures davant d'atacs de Ransomware són de prevenció, algunes recomanacions són:

- Realitzar còpies de seguretat periòdiques, d'aquesta forma la majoria de dades no es perdran amb l'atac.
- Mantindre el Software actualitzat, quan es troba alguna vulnerabilitat es sol fer alguna actualització per protegir la vulnerabilitat, d'aquesta forma el nostre dispositiu serà més segur.
- Evitar els correus de remitents desconeguts, s'ha de tenir especial cura amb aquells que tenen arxius adjunts.
- Evitar les webs no segures, així com els que puguin contenir anuncis o banners que no siguin segurs.
- Repassar les opcions de privacitat i seguretat dels navegadors pot ajudar a navegar de forma més segura.

5.4 INDICADORS D'UN ATAC DE RANSOMWARE

A la llarga s'ha vist que hi ha algunes alteracions del sistema que indiquen que ens podem trobar sota un atac de Ransomware:

- Quan s'obren molts fitxers.
- Estructures streams d'entrada i sortida diferents.
- Moltes operacions d'escriure i de sobreescriure.
- Un procés que crida APIs d'encriptació.
- Processos de lectura, escriptura o suprimir freqüent en un període curt de temps.
- Comunicació amb comandes i control d'un servidor.
- Modificar les contrasenyes dels usuaris registrats.

6 DATASET

L'elecció d'un dataset en Machine Learning és molt important, per a predir un comportament o qualsevol aspecte cal tenir les dades adequades, cal seleccionar les dades adequades per a que posteriorment amb Machine Learning es puguin identificar patrons o tendències que siguin reals, no casualitats.

El dataset escollit és un dataset publicat per la Universitat de New Brunswick, anomenat CIC-MalMem-2022 [12]. El dataset és un recull de dades sobre la memòria per detectar malware ofuscat, aquest tipus de Malware es caracteritza per amagar-se per evitar la detecció o la seva eliminació. El dataset s'ha creat per representar dades realistes sobre els Malwares més comuns com són: Spyware, Ransomware i Trojan Horse, en el nostre cas només ens fixarem amb els de Ransomware.

Per a crear el dataset s'han triat diferents famílies de Ransomware entre les famílies més comuns, a continuació, en la Figura 8, es pot veure una distribució de les famílies de Ransomware:

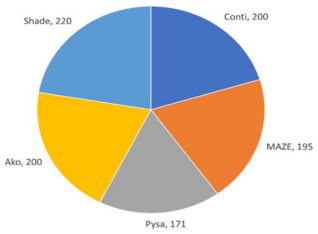


Fig. 8: Famílies de Ransomware del dataset.

Aquest conjunt de dades utilitza el mode de debug per al procés d'abocament de memòria per evitar que el procés d'abocament es mostri als abocaments de memòria. Això funciona per representar més precisament el que un usuari mitjà executaria mentres es produeix un atac de malware.

Es pot veure en més detall en l'apèndix A.1 més detall sobre com es va crear el Dataset que utilitzarem per la predicció de Ransomware.

7 ANÀLISI I PREPROCESSAT DEL DATASET

El dataset té diferents atacs a part de Ransomware, la primera transformació que es farà serà eliminar aquelles instàncies o mostres que no són benignes o resulten d'un atac de Ransomware.

Seguidament, podem veure que la classe 'Category' i 'Class' pel nostre dataset representaran el mateix, s'ha triat eliminar l'atribut 'Category'. Posteriorment, com per treballar amb algoritmes de Machine Learning ens interessa treballar amb variables numèriques s'han convertit els valors en numèrics que representaran etiquetes, sent la conversió: 0 per les mostres 'Benigne', 1 per les mostres 'Malware' que són les mostres que representen un atac de Ransomware.

Un cop creat el dataset pel cas particular que volem estudiar iniciarem mirant les dimensionalitats del dataset, el dataset té 58596 files (instàncies o mostres) i 57 columnes (atributs). Cal tenir en compte que aquestes dimensi-

ons són sense fer el preprocessament de dades per adaptar als interessos del projecte, treballant només amb aquelles mostres que són per Ransomware o benignes, en aquest cas tindrem 39089 files.

Posteriorment, un pas important és identificar la quantitat de mostres que són nul·les, és a dir que no hi ha dada, en aquest cas no hi ha atributs nuls, per tant no cal fer cap adaptació.

Ara passarem a veure les distribucions i correlacions dels atributs, la correlació és una mesura que compara les tendències de dos conjunts d'atributs. En la Figura 9 podem veure les correlacions d'aquest dataset en forma d'una matriu de correlacions abans de fer les transformacions. Es pot

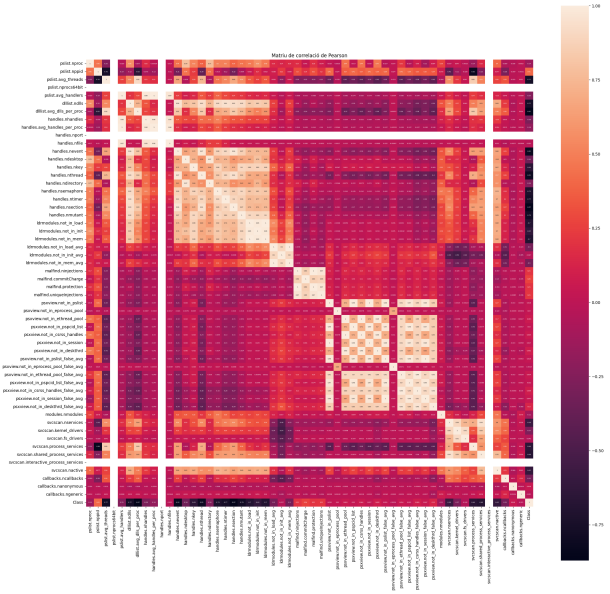


Fig. 9: Matriu de correlació del dataset.

observar els següents aspectes 'pslist.nprocs64bit', 'handles.nport' i 'svcscan.interactive_process_services' no tenen correlació, ja que són aquells que sempre prenen els mateixos valors. També observant amb més detall els atributs i els possibles valors s'ha observat que els atributs 'callbacks.ngeneric' i 'callbacks.nanonymous' no aporten informació, ja que gairebé sempre tenen el mateix valor, això s'ha provat creant models amb i sense ells i s'ha observat el mateix resultat.

També es pot observar que la relació de correlació entre les següents parelles d'atributs és 1: 'handles.avg_handles_per_proc' i 'pslist.avg_handlers', 'ldrmodules.not_in_load' i 'ldrmodules.not_in_mem', 'ldrmodules.not_in_mem_avg' i 'ldrmodules.not_in_load_avg', 'malfind.ninjections' i 'malfind.protection', 'psxview.not_in_session' i 'psxview.not_in_pslist', 'psxview.not_in_pslist_false_avg' i 'psxview.not_in_session_false_avg' i 'psxview.not_in_ethread_pool' i 'psxview.not_in_csrss_handles'.

Això ens indica que els dos atributs aporten la mateixa informació, a més si ens fixem les relacions amb la resta d'atributs d'aquestes parelles d'atributs són les mateixes. De forma que de cada una de les parelles s'ha eliminat el segon element.

Ara passarem a observar la distribució de la variable objectiu, per fer prediccions ens interessarà que les classes de la variable objectiu estiguin balancejades i si no és així caldrà tindre en compte com de balancejades estan. Es pot observar en la Figura 10, que el percentatge de mostres que són benignes representa un 74.95% del total de dades, mentre que el percentatge de mostres que són ransomware en representa un 25.05%.

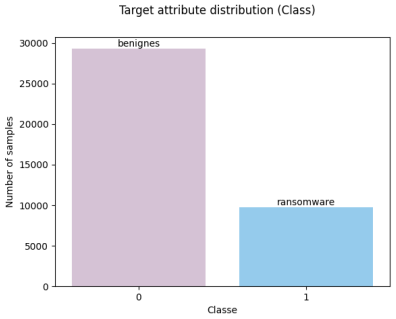


Fig. 10: Distribució de la variable objectiu del dataset.

Finalment per a crear els models les dades s'estandarditzaran amb la funció MinMaxScaler(), amb l'estandardització les dades es situaran amb valors entre 0 i 1 proporcionalment a l'atribut, provocant que un atribut no tingui una influència major en la predicció.

8 SELECCIÓ DE MODEL

S'han triat un algoritme de cada tipus entre els algoritmes simples i els algoritmes complexos, que combinen diferents classificadors simples, d'acord als criteris esmentats.

Els algoritmes triats són:

- **SVM polinomi**: l'elecció de weak-learners. Dona un bon rendiment quant a les mètriques de rendiment, tant l'accuracy com f1_score, la roc i el recall, i a més el temps de test és dels més baixos.
- **HistGradientBoosting**: És un dels algoritmes amb millors mètriques de rendiment entre els strong-learners i amb que menor temps de test, quasi la meitat que la resta.

Per seleccionar els models s'han seguit per tan els següents criteris, es pot veure en l'apèndix A.2 explicacions sobre les mètriques que s'han utilitzats per triar els millors models:

- Que els paràmetres de rendiment accuracy, f1_score, la roc i el recall siguin el més gran possible.
- El temps de test sigui el més petit possible.
- Un equilibri entre el punt 1 i el punt 2, és a dir, s'ha sacrificat una mica de rendiment per tenir un millor temps de predicció, el que ens permetrà una resposta més ràpida a l'atac de Ransomware si fos el cas.
- El temps de convergència no sigui molt gran.

Mencionar també al Decision Tree, que ofereix uns resultats molt bons i el temps de realitzar els tests és el més ràpid de tots. Això ajuda que molts algoritmes que es basen en el Decision Tree, com són el HistGradientBoosting,

ExtraTrees, RandomForest... Funcionen també molt bé i aportin molt bons resultats.

Es pot veure en més detall els resultats dels diferents algoritmes provats en l'Apèndix A6.

9 OPTIMITZACIÓ

En Machine Learning la majoria d'algoritmes tenen almenys un hiperparàmetre, els hiperparàmetres són paràmetres que no es pot definir una forma per predir el valor que ens donarà millor rendiment i que normalment el programador sol definir d'acord amb l'experiència o proves. Tot i això, si s'han definit alguns algoritmes, els quals apareixen a la llibreria sklearn, que ens ajuden a trobar els millors hiperparàmetres, en aquesta secció passarem a veure quins són aquests algoritmes i com funcionen:

Random Search: per cada paràmetre que es vol optimitzar es dona una distribució de possibles valors i s'especifica també el nombre de proves que és l'algoritme farà, l'algoritme farà proves aleatòries dels possibles valors amb diferents combinacions i retornarà el model que dona millors resultats.

Per trobar els millors hiperparàmetres, seleccionarem l'algoritme de Random Search, ja que permetrà per cada un dels algoritmes provar entre diferents valors, més dispersos per apropar-nos a l'òptim el més possible. El Grid Search provar tots els possibles valors és molt complex, ja que hi ha una limitació computacional i seria necessari molt de temps.

9.1 PARÀMETRES OPTIMITZATS

A continuació veurem els hiperparàmetres que s'han optimitzat per cada un dels algoritmes seleccionats i el rang de valors o valors que s'han provat per cada un d'ells:

SVM polinomi:

'C': "loguniform(1e-2, 1e4)" que indica que pot prendre valors entre 1e-2 i 1e4 seguint una distribució logarítmica. És un paràmetre de regularització per evitar overfitting.

'gamma': loguniform(1e-2, 1e4) seguint una distribució logarítmica. Un coeficient per l'algoritme de SVM polinomial.

'class_weight': ['balanced', None]

HistGradientBoosting:

'l2_regularization': loguniform(1e-6, 1e3) seguint una distribució logarítmica. És un paràmetre de regularització que ajuda a evitar l'overfitting en el model.

'learning_rate': loguniform(0.001, 10), seguint una distribució logarítmica. Aquest paràmetre especifica la rapidesa amb el que l'algoritme convergeix per arribar al seu òptim.

'max_leaf_nodes': range(2, 256). Aquest paràmetre especifica el nombre màxim de fulles que tindrà l'arbre.

'min_samples_leaf': range(1, 100). Aquest paràmetre especifica el nombre mínim de mostres en cada fulla.

'max_bins': range(2, 255). Màxim nombre de contenidors, que permeten un entrenament més ràpid.

'max_iter': [10,50,100,200, 500,1000]. Aquest paràmetre el màxim nombre d'arbres per la classificació.

El resultat obtingut per cada un dels algoritmes que hem optimitzat ha sigut:

- SVC (C = 257.445884056174, class_weight = 'balanced', gamma = 0.18325543501264982, kernel='poly', probability=True)

- HistGradientBoostingClassifier (l2_regularization= 1.0407479858562003e-05, learning_rate= 0.08115041228011703, max_bins= 84, max_iter= 200, max_leaf_nodes= 83, min_samples_leaf= 95)

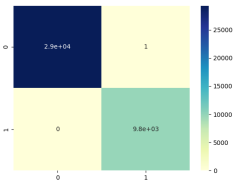
Si veiem els resultats que obtenim en executar aquests models, amb els hiperparàmetres òptims, obtenim:

TAULA 1: RESULTATS MODELS SELECCIONATS DE SKLEARN.

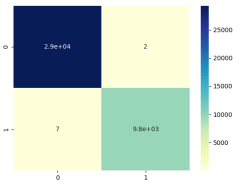
Model	accuracy	f1 score	recall	roc_auc	temps convergir	temps test
HistGradientBoosting	0.999983	0.999974	0.999974	0.999999	0.692764	0.047505
SVM polinomi	0.999932	0.999949	0.999949	1.000000	1.459770	0.024470

Podem observar que no hi ha gaire millora respecte als models creats anteriorment, això és en part perquè el marge de millora és molt baix, aconseguir resultats del tot perfectes es pot interpretar com arribar a l'overfitting, fet que provocaria que el nostre model no respongués correctament a noves instàncies a predir que fossin diferents de les ja predites en l'entrenament. El temps també es manté aproximadament, el qual permetria una ràpida predicció.

Per més detall ens fixarem en la matriu de confusió en la Figura 11:



(a) Matriu confusió model HistGradientBoosting.



(b) Matriu confusió model SVM polynomial.

Fig. 11: Matrius confusió models seleccionats.

L'anterior mètrica de matriu de confusió dona suport al supòsit anterior, ja s'ha aconseguit models amb molt bon rendiment i que les diferents mètriques s'aproximen molt a la perfecció, però no arriben a l'overfitting. També es poden veure més resultats i altres mètriques en l'apèndix A.6.

10 XARXA NEURONAL

En aquesta secció es passarà a parlar de la xarxa neuronal que s'ha creat amb Tensors. Per crear la xarxa neuronal primer s'ha procedit a entendre com funcionen els Tensors i com es poden utilitzar per crear una xarxa neuronal, posteriorment s'ha provat el seu funcionament amb un dataset molt conegut i que ofereix la pròpia llibreria de sklearn amb la que hem provat altres models, la llibreria d'iris. Finalment, s'ha creat una xarxa neuronal pel nostre dataset en concret.

Podem veure explicacions més en detall sobre les xarxes neuronal en l'apèndix A.5.

10.1 CONSIDERACIONS DE LA XARXA NEURONAL

Per a crear la xarxa neuronal s'han hagut de triar alguns aspectes que a continuació veurem quins han sigut:

- **Número d'Epochs**: que defineixen les iteracions que donem al model per convergir amb el conjunt d'entrenament, a major número d'EPOCHS més temps tardarà a convergir l'algoritme, però podem aconseguir un model amb millor rendiment.
- **Nombre de capes ocultes de la xarxa neuronal**: a major número d'EPOCHS més temps trigarà a fer prediccions i convergir l'algoritme, però podem obtenir un model amb millor rendiment.
- **Optimització**: Per a l'optimització de paràmetres s'ha triat el model SGD (Stochastic gradient descent) que intenta buscar el mínim global, en lloc de conformarse amb un mínim local i per aconseguir-ho fa alguns càlculs redundants.
- **Funció de pèrdues**: Cal definir una funció de pèrdues per definir com penalitzen les mostres que és predeixin malament en l'entrenament, per tal de decidir si un model és millor o pitjor que un altre. Pel nostre model hem seleccionat l'entropia amb un balanceig de pesos, que permetrà tindre en compte les distribucions de les variables de sortida, es donarà més pes als errors i encerts en les que la mostra a predir sigui menys freqüent.

L'entropia es pot calcular com:
$H(X) = -\Sigma p_i * log_2(p_i)$

11 XARXA NEURONAL AMB EL DATASET RANSOMWARE

En aquesta secció veurem el model de xarxa neuronal creat per resoldre el problema de detecció d'atacs de Ransomware. Els paràmetres utilitzats per crear aquesta xarxa neuronal han sigut els especificats anteriorment i similarment a l'exemple anterior: utilitzant la funció d'activació ReLu, separant els conjunts d'entrenament i test amb un 80% de les dades per les dades d'entrenament i un 20% de mostres per les de test, utilitzant com a funció de pèrdues l'entropia amb uns pesos i optimitzador SGD.

Per la creació de la xarxa neuronal s'ha iniciat amb cap capa oculta i aumentar poc tan els nodes com el nombre de capes ocultes. S'han fet diferents proves i la que ha obtingut millors resultats tenint en compte la precisió i el temps ha sigut amb la xarxa tenint la següent forma:

- Capa entrada: 43 entrades d'entrada (que corresponen a les 43 característiques del dataset) i 50 sortides.
- 1a capa oculta: 50 entrades i 100 sortides.
- 2a copa oculta: 100 entrades i 200 sortides.
- Capa sortida: 200 entrades i 2 sortides (que corresponen a les 2 possibles classificacions de cada instància).

S'han definit perquè l'algoritme arribi a convergir EPOCHS=1000, s'aconsegueix amb aquest algoritme una precisió del 99,136%, a continuació es pot veure l'evolució de l'accuracy al llarg de les iteracions dels diferents conjunts d'entrenament i test, en la Figura 12:

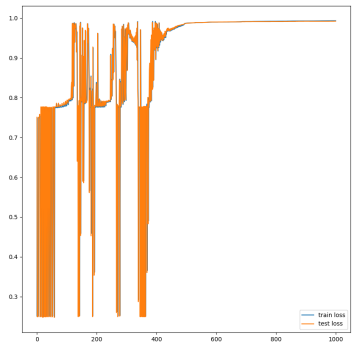


Fig. 12: Evolució de l'accuracy en relació el número d'EPOCHS (1000) dataset de Ransomware.

També, per veure en més detall els resultats i analitzar-lo podem veure en la Figura 13 la matriu de confusió, on es pot veure que per ambdues categories s'estan aconseguint percentualment una gran quantitat d'encerts tot i que amb més errors que amb els models de sklearn seleccionats que hem vist anteriorment.

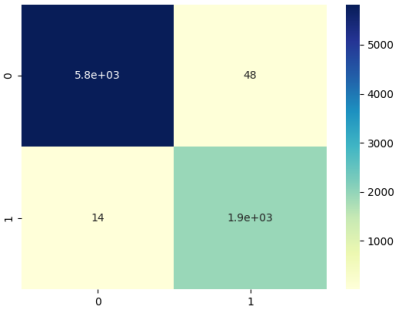


Fig. 13: Matriu de confusió de la xarxa neuronal pel dataset Ransomware.

Observant els resultats podem veure que hem creat una xarxa neuronal amb bon rendiment pel que fa a la precisió i el recall, per tant és un model que ens pot servir per detectar atacs de Ransomware, a més el temps per predir és de 0.0114 segons de forma, que hem aconseguit un model amb un millor rendiment quan el temps i una precisió molt similar a les anteriors. Tot i que el temps de convergir l'algoritme és molt major als anteriors (81.1464 s).

12 COMPARACIÓ DE RESULTATS

Si comparem amb els resultats obtinguts amb els dels creadors del Dataset [13], podem veure que els resultats dels algoritmes és lleugerament superior pel nostre cas de predir Ransomware que en el seu cas per predir malware ofuscat que arriben fins el 98%, superen lleugerament els models d'aquest estudi.

Al llarg del temps s'han fet nombrosos estudis per predir atacs de Ransomware amb diferents datasets, un altre exemple és l'estudi realitzat [14] en el que es crea un dataset per predir atacs de Ransomware amb anàlisi dinàmic, en aquest projecte els resultats experimentals mostren el mètode proposat pot detectar atacs de Ransomware utilitzant només característiques de comportament de baix nivell. De forma que si els atacants poden comprometre la primera capa de protecció, la capa de seguretat addicional seria útil.

Brengel va presentar un mètode de detecció de virtualització basat en el temps mitjançant la sobrecàrrega de sortida de VM i la memòria intermèdia de traducció assolint un 95.95% de tassa d'encert.

Altres exemples d'estudis realitzats són:

"Ransomware Detection using Random Forest Technique", on es mostra que amb el Random Forest es va aconseguir una accuracy de fins a 97.74% d'encert. També es mostra diversos estudis de predicció de Ransomware a partir de trucades a l'API que es representen a partir de vectors q-gram, on els resultats amb models SVM mostren precisió de 97.48%. Vinayakumar va proposar mètodes per la detecció d'atacs de Ransomware a partir de recollir les seqüències de l'API amb anàlisi dinàmic, amb un model multicapa de perceptró (MLP) van aconseguir una precisió del 98%. Homayoun va introduir un sistema de detecció de Ransomware basat en la mineria de patrons seqüencials com a característiques candidates per utilitzar-les com a entrada a les tècniques d'aprenentatge computacional (MLP, Bagging, Random Forest i J48) amb finalitats de classificació. Els resultats van mostrar una precisió de fins al 99% per a la detecció de Ransomware.[5]

"API-Based Ransomware Detection Using Machine Learning-Based Threat Detection Models". Aquesta anàlisi va obtenir una alta precisió de detecció de Ransomware del 99,18% per a plataformes basades en Windows i mostra el potencial d'aconseguir capacitats de detecció d'alta precisió quan s'utilitza una combinació de trucades d'API i un model ML. [6]

13 CONCLUSIONS

Els objectius inicials del projecte de forma general eren: entendre amb més profunditat els algoritmes de Machine Learning, els atacs de Ransomware, crea dos models per detectar-los i crear una xarxa neuronal amb Pytorch per detectar-los.

Al final del projecte hem vist que els diferents objectius marcats s'han assolit satisfactòriament. S'han creat un model SVM i un model HistoGradientBoosting per predir atacs de Ransomware amb un percentatge d'encert superior al 99%, a més també s'ha creat una xarxa neuronal amb dues capes ocultes que ens donava també una tassa d'encert similar, i a més ens oferia millores pel que fa al temps de predicció. Per realitzar el treball, s'ha entès amb més aprofundiment el funcionament, els algorismes i paràmetres del Machine Learning i com treballar amb ells.

Per altra banda, també al llarg del treball s'ha seguit una metodologia pròpia, que es tractava d'una metodologia Scrumban amb algunes modificacions, un tipus de metodologia Agile, amb algunes petites modificacions, que inclou elements de Scrum i Kanban per a treballar de forma continuada, estructurada i eficient.

Els atacs de Ransomware poden tenir conseqüències molt greus per les víctimes, poden bloquejar l'accés a dispositius, robar dades o bloquejar tots els serveis informàtics d'una organització. El que pot comportar grans pèrdues econòmiques o del dret a la privacitat. Els atacs de Ransomware són del tipus malware ofuscat, és a dir que es mantenen ocults en el sistema sense que l'usuari se n'adoni, addicionalment això són atacs que al llarg del temps evolucionen i segueixen nous patrons de forma que són molt complicats de detectar. A causa d'això, van aparèixer nombrosos estudis per detectar aquest tipus d'atac per poder posteriorment respondre'n. En aquest projecte s'han creat models per detectar atacs de Ransomware.

Analitzant rendiment dels models creats i comparant-lo amb altres models que s'han definit per detectar atacs de Ransomware, podem dir que hem obtingut resultats amb un alt nivell d'encert, tant en termes de precisió com de recall, és molt difícil millorar els resultats, ja que si es fes passaria a tenir models amb overfitting, que per futures mostres no funcionarien correctament. A més la predicció es fa en un temps bastant baix, de forma que la detecció per posteriorment actuar pot ser molt ràpida. D'aquesta forma podem afirmar que els Ransomware es pot detectar a partir de l'anàlisi de la memòria, ja que el dataset s'ha creat a partir de dades extretes de la memòria. Cal tenir en compte que aquest sistema sempre hi quan l'extracció d'aquestes característiques en temps real no afecti la memòria. Stuttgen and Cohen van proposar un framework que ens pot ajudar en aquest procés.

Amb tot el procediment de crear els diferents models, també s'ha vist la importància dels paràmetres per triar un model o pensar raonadament per definir les mètriques adequades pel problema que volem resoldre, en el nostre cas la importància de tenir en compte les distribucions de les classes i el temps per predir. També hem vist com crear una xarxa neuronal a partir de PyTorch i com especificar els diferents paràmetres per definir una xarxa neuronal i l'efecte que poden tenir i com poc a poc modificar els paràmetres per millorar el rendiment d'aquesta.

Resumint els models seleccionats per detectar Rasomware hem aconseguit els següents resultats:

TAULA 2: RESULTATS FINALS.

Model	accuracy	temps convergir (s)	temps test (s)
HistGradientBoosting optimitzat	0.999983	0.692764	0.047505
SVM polinomi optimitzat	0.999932	1.459770	0.024470
Xarxa Neuronal pròpia	0.99136	81.1464	0.0114

Els diferents models ofereixen una precisió bastant similar entre ells, sent el millor d'ells el "HistGradientBoosting optimitzat", per altra banda, la Xarxa Neuronal creada és la que ofereix un millor temps per predir. El temps per convergir l'algoritme no és tan rellevant per escollir l'algoritme en aquest cas, ja que és un temps que servirà per crear un nou model degut a la constant evolució dels atacs de Ransomware provoca que si s'ha de tenir en compte que no sigui molt elevat, s'utilitzarà per definir un nou model i mantenir-se actualitzats a les evolucions dels atacs de Ransomware, però podem veure que en aquest cas no és exageradament elevat.

De cara a futurs treballs podria utilitzar-se els models que s'han creat per a crear un script que detecti en un disposi-

tiu si hi ha algun atac de Ransomware periòdicament, obtenint els diferents paràmetres de memòria que s'han especificat prèviament i es faci alguna resposta. A més, també es podrien ampliar el dataset i utilitzar el mateix procediment perquè l'script detecti diferents atacs de malware i faci diferents respostes per evitar-los. Un altre treball futur és provar el rendiment creant alguna xarxa convolucional o altres tipus de xarxes de models.

Finalment, en aquest projecte s'ha seguit una metodologia de treball Agile, que ha ajudat a avançar continuadament en el projecte, per tant s'ha pogut experimentar com les metodologies Agile ajuden a treballar i acabar els treballs a temps, sense tenir grans càrregues de treball en alguns instants de temps, sinó que treballant de forma continuada per aconseguir els objectius marcats en cada Sprint la sensació de càrrega general de treball és menor.

Podem veure el Github realitzat a: https://github.com-/DavidSardaM/Machine-Learning-to-detect-Ransomware.

AGRAÏMENTS

Agraïr al meu tutor del TFG Joan Protasio per les recomenacions i ajudar-me en el plantejament del projecte. Agraïr també a la meva família i la meva nòvia pel suport i consells donats al llarg del projecte.

REFERÈNCIES

[1] A. Chamorro Fernández, "Malware Detection with Machine Learning", Dipòsit Digital de Documents de la UAB, 2022. [Online]. Available: https://ddd.uab.cat/record/231499?ln=ca. [Accessed: 20- Feb- 2022].

[2] Kaspersky]L·Los principales ataques de ransomware", www.kaspersky.es, 2022. [Online]. Available: https://www.kaspersky.es/resource-center/threats/top-ransomware-2020. [Accessed: 24- Feb- 2022].

[3] U. Urooj , B. Ali Saleh Al-rimy , A. Zainal , F. A. Ghaleb and M. A. Rassam, "Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions". Appl. Sci. 2022, 12, 172. [Online]. Available: https://doi.org/10.3390/app12010172. [Accessed: 28- Feb- 2022].

[4] D. Dmitry, "Ransomware Detection Using Machine Learning — SpinOne", SpinOne, 2019. [Online]. Available: https://spinbackup.com/blog/ransomware-detection-using-machine-learning/. [Accessed: 25- Feb- 2022].

[5] B. Mohammed Khammas, "Ransomware Detection using Random Forest Technique", Science Direct, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240-5959520304756. [Accessed: 27- Feb- 2022]

[6] M. Almousa, S. Basavaraju and M. Anwar, "API-Based Ransomware Detection Using Machine Learning-Based Threat Detection Models,"2021 18th International Conference on Privacy, Security and Trust (PST), 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9647816. [Accessed: 1- Mar- 2022].

[7] https://elpais.com/espana/catalunya/2021-10-26/pysa-el-virus-que-lastra-otro-semestre-en-la-universidad-autonoma-de-barcelona.html

[8] https://elpais.com/espana/catalunya/2021-11-11/la-uab-ve-un-riesgo-residual-en-que-los-ciberatacantes-publiquen-datos-personales.html

[9] J. González Sabaté, "Aprenentatge Computacional", class notes for Aprenentatge Computacional (102787) - Grau en Enginyeria Informàtica MO47367, Escola D'enginyeria, Universitat Autònoma de Barcelona, 2021. [Accessed: 02- Apr- 2022]

[10] D. Jorge Matich, Redes Neuronales: Conceptos Básicos y Aplicaciones.. Rosario: Universidad Tecnológica Nacional – Facultad Regional Rosario Departamento de Ingeniería Química Grupo de Investigación Aplicada a la Ingeniería Química (GIAIQ), 2001. [Online]. Available: https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf [Accessed: 04- Apr- 2022].

[11] Arash Habibi Lashkari, Andi Fitriah A. Kadir, Laya Taheri, and Ali A. Ghorbani, "Toward Developing a Systematic Approach to Generate Benchmark Android Malware Datasets and Classification", In the proceedings of the 52nd IEEE International Carnahan Conference on Security Technology (ICCST), Montreal, Quebec, Canada, 2018.Available: https://www.unb.ca/cic/datasets/andmal2017.html. [Accessed: 10- Apr- 2022].

[12] Tristan Carrier, Princy Victor, Ali Tekeoglu, Arash Habibi Lashkari," Detecting Obfuscated Malware using Memory Feature Engineering", The 8th International Conference on Information Systems Security and Privacy (ICISSP), 2022. Available: https://www.unb.ca/cic/datasets/malmem-2022.html [Accessed: 10- Apr- 2022].

[13] Carrier T., Victor P., Tekeoglu A. and Lashkari A. (2022). Detecting Obfuscated Malware using Memory Feature Engineering. In Proceedings of the 8th International Conference on Information Systems Security and Privacy - Volume 1: ICISSP, ISBN 978-989-758-553-1, pages 177-188. DOI: 10.5220/0010908200003120 Available: https://www.scitepress.org/Papers/2022/109082/109-082.pdf [Accessed: 30- Apr- 2022]

[14] Manabu Hirano, Ryo Hodota, Ryotaro Kobayashi, RanSAP: An open dataset of ransomware storage access patterns for training machine learning models, Forensic Science International: Digital Investigation, Volume 40, 2022, 301314, ISSN 2666-2817, https://doi.org/10.1016/j.fsidi.2021.301314. [Online]. Available: https://www.sciencedirect.com/science/article/pii/-S2666281721002390. [Accessed: 28- Apr- 2022].

APÈNDIX

A.1 CREACIÓ DEL DATASET

El dataset CIC-MalMem-2022 publicat per la Universitat de New Brunswick recull paràmetres de memòria per detectar malware ofuscat. El dataset original s'ha creat s'ha fet servir un framework amb els següents components:

Memory Dump File: Les seccions de memòria tirades poden ser obtingudes utilitzant programes com MAGNET RAM o controladors de màquines virtual amb la característica de capturar memòria. Aquest és un snapshot que mostra l'activitat que té lloc en la memòria del sistema.

Volatility: És una col·lecció d'eines obertes implementada en Python sota GNU per extreure artefactes digitals de la volatilitat de mostres de memòria RAM.

VolMemLyzer-V2: Extreu característiques de memòria per solucions basades a aprendre a partir de les 26 noves característiques que genera proposades per detectar malware ofuscat. VolMemLyzer extreu les característiques utilitzant plugins de volatilitat.

Per la creació del dataset es va fer a partir de recollir mostres benignes, les mostres benignes són extretes a partir d'executar diverses aplicacions en una màquina simulant el comportament d'un usuari mitjà, i malignes, extretes de 2916 mostres de virus de VirusTotal amb diferents categories, entre les quals s'inclou Ransomware que serà la que ens interessa pel nostre projecte, en el mateix tipus de dispositiu.

Per crear el dataset es van seguir els següents 4 passos:

1. Seleccionar diferents programaris maliciosos actuals, ja que si és per programari antic, per atacs d'avui en dia ja no serviria, i recollir-ne mostres.´
2. El segon pas, fer captures de la memòria, fent captures instantànies de la memòria a partir d'un controlador de Màquines Virtuals de VirtualBox. Això ens permetrà assegurar que la memòria no està contaminada, això evitarà outliers que perjudicarien el model que creem. Les captures de memòria s'han fet d'una màquina Windows 10. S'ha testejat i fet captures de memòria del comportament habitual d'un usuari i de la màquina contaminada amb el malware. S'han fet per cada mostra de malware 10 captures separades per 15 segons. Les mostres benignes s'han fet a partir d'utilitzar diferents aplicacions simulant el comportament habitual d'un usuari amb l'algoritme SMOTE.
3. El tercer pas, és extreure les característiques de les captures de memòria en una màquina Kali Linux utilitzant el VolMemLyzer.
4. Finalment, el quart part és crear el CSV, que ens servirà de dataset amb les característiques extretes.

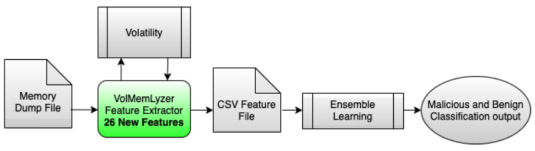


Fig. 14: Procés de creació del dataset.

A.2 MÈTRIQUES

Per avaluar el funcionament d'un model cal definir unes mètriques, uns indicadors que permetin dir que un model funciona millor o pitjor que un altre i ens els permeti comparar els models de forma que ens permeti tria el que més ens interessi, el millor en alguna métrica. A continuació introduirem a continuació:

Matriu de Confusió : la matriu de confusió permet visualitzar per cada classe les dades classificades correctament i les que no, fet que serà molt útil per problemes de classificació. En ella les files representen les instàncies predites de cada classe i les columnes la seva verdadera classificació. D'aquesta forma els elements en la diagonal representaran les instàncies correctament predites. Per entendre millor les següents mètriques és útil particularizar pel cas de sortida binaria, és a dir que les classes de sortida siguin dos (positiu i negatiu), en aquest cas la matriu de confusió tindrà la següent forma:

TAULA 3: MATRIU DE CONFUSIÓ EN CAS BINARI.

		Predicció	
		Positius	**Negatius**
Observació	Positius	Certs Positius (VP)	Falsos Negatius (FN)
	Negatius	Falsos Positius(FP)	Certs Negatius(VN)

Accuracy: es pot calcular com el nombre d'elements correctament classificats entre el nombre total d'elements multiplicat per 100. Si ho mirem respecte la matriu de confussió anterior: Accuracy=(VP + VN)/ (VP+VN+FP+FN)

Precision: en molts casos l'accuracy no representa una bona mètrica per avaluar el model, en especial en aquells casos en que l'atribut objectiu no té una distribució uniforme, la precision té en compte l'encert per cada una de les classes de l'atribut objectiu, per calcular-la es farà: Precision class positive= (VP)/(VP+FP)

Recall: el recall es defineix com el nombre de mostres d'una classe que són predites correctament pel model. Es pot calcular com: Recall class positive= (VP)/(VP+FN)

F1-Score: és una funció que combina la precisió i la recall, per aquells casos on ambdues són importants. Es pot calcular com: F1-score= 2*Precision*Recall/(Precision+Recall)

ROC: és una corba que mostra el rendiment en funció del seu llindar de tall, que és la línea que separa els elements positius dels elements negatius, és a dir si el resultat d'una mostra sobrepassa el llindar serà positiu i en cas contrari negatiu, essencialment mostra la taxa de VP contra la FP. Quan la ROC tendeixi a la unitat, és a dir la corba blava tendeixi a seguir la forma d'un quadrat millor serà el model per aquesta mètrica. Aquesta métrica ajudar a triar millor el llindar de decisió entre les classes.

A.3 MACHINE LEARNING

L'aprenentatge computacional és una tècnica que té com a objectiu que un ordinador pugui analitzar dades de forma que a partir d'unes dades ens pugui dir el valor d'una altra dada, això ens permetrà utilitzar aquesta predicció per pren-

dre decisions, automatitzar accions o millorar adaptació de sistemes.

En el Machine Learning hi ha bàsicament 3 elements bàsics:

Conjunt de característiques: que descriuen propietats d'algun element que ens serviran per a la predicció, aquestes característiques caldrà per a un procés per escollir, preprocessar i manipular.

Model: que permetrà fer un mapeig entre unes mostres d'aprenentatge i unes sortides, que són l'objectiu de la predicció.

Tasques: que representen el problema que es vol resoldre.

D'aquesta forma utilitzarem unes característiques per construir models que es permetran resoldre tasques.

Sovint les dades contenen característiques que no aporten informació per la predicció o aporten la mateixa informació que altres característiques i la millor opció serà eliminarlos. També entre les dades i les característiques existents sol haver-hi soroll, aquest soroll pot ser degut al fet que entre les mesures pot haver-hi mesures que sobresurtin, coneguts com a outliers, que poden ser deguts errors de mesura de la característica, errors en traspassar les dades...

Una forma de classificar els algoritmes d'aprenentatge computacional és la següent [2]:

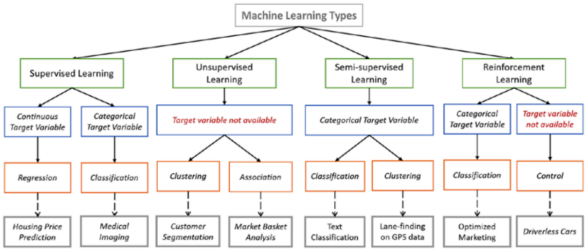


Fig. 15: Tipus d'algoritmes d'aprenentatge computacional.

A.4 APRENENTATGE SUPERVISAT

Si ens centrem en l'aprenentatge supervisat que es basen en la idea de l'existència d'un conjunt d'entrenament.

Quan volem resoldre un problema a través de l'entrenament supervisat, es poden distingir diferents etapes en el seu procés:

1. Primerament, s'observen les característiques, es defineixen la característica/ques objectiu, s'observen els tamanys i formats de les dades, s'observen les correlacions entre les característiques i la característica/ques objectiu...

2. Definir de quin tipus de problema es tracta: de classificació o de regressió.

3. Es fa un preprocessat de les dades on s'eliminen aquelles que no aporten informació, es poden combinar dades per crear noves característiques, es poden canviar de format algunes característiques.

4. Posteriorment, se separen les dades en un conjunt d'entrenament, conjunt de validació i un conjunt de test. Amb aquests conjunts el conjunt d'entrenament servirà per crear un model, el conjunt de validació serviran per optimitzar el model, ajudarà a optimitzar els hiperparàmetres que formen el model, i el conjunt de test servirà per avaluar el model creat.

5. Es crearà el model, s'haurà de decidir de quin tipus serà el model i els seus paràmetres.

6. S'optimitzarà els hiperparàmetres del model.

7. S'avalua el model amb el conjunt de test.

Dins de l'aprenentatge supervisat es poden distingir entre dos tipus de problemes:

- **Classificació**: els problemes de classificació són aquells on es vol predir una etiqueta, és a dir es caracteritzen perquè l'atribut objectiu té un nombre finit de possibles sortides. Un exemple podria ser predir si l'aigua en base a les seves característiques es potable o no potable o donades unes imatges d'escriptura de lletres predir a quina lletra pertany cada imatge.

- **Regressió**: en els problemes de regressió la sortida és un valor, aquest valor pertany a un conjunt no finit. Un exemple podria ser predir el preu d'una casa en base a algunes característiques o predir.

Una qüestió que sorgeix és com separar els conjunts d'entrenament, la validació i de test en aquest tipus de problemes. És important adonar-se que com més quantitat d'instàncies en el conjunt d'entrenament, millor serà el predictor, ja que podrà captar millor les tendències i patrons de l'atribut objectiu, és per això que normalment el conjunt de test sol representar una quantitat igual o superior al 50% de les dades. Una distribució que es sol utilitzar bastant és 70%-20%-10% pels conjunts d'entrenament, validació i test respectivament.

Una altra tècnica que s'utilitza és la validació creuada, aquesta consisteix a dividir les dades en K subconjunts. A partir d'aquí es repeteix un procés K vegades, per cada una de les vegades un dels subconjunts es fa servir com a conjunt de validació/test i els altres K-1 subconjunts s'utilitzen com a conjunt d'entrenament. Els errors d'estimació es farà com una mitjana de tots ells i ens permet reduir el biaix i la variància. Normalment, s'utilitzen valors de K=3, 5 o 10.

A.5 Xarxes Neuronals

Les xarxes neuronals és un algoritme de predicció utilitzat en intel·ligència artificial i machine learning que intenta simular les connexions neuronals del cervell dels humans, són un conjunt d'unitats de processament interconnectades.

En una xarxa neuronal tenim un conjunt d'entrades que entren a una capa d'entrada, passen per un conjunt de capes ocultes que processen les dades i finalment, a través d'una capa de sortida, ens donen una o més sortides. En les capes ocultes cada una de les neurones passa les seves sortides a altres neurones i així successivament fins la capa de sortida.

A.5.1 PERCEPTRÓ

Les neurones que formen una xarxa neuronal estan basades en la idea del perceptró de Ronseblatt. El perceptró de Rosenblatt és bàsicament una unitat on li entren un conjunt

d'entrades a les quals se'ls associa uns pesos i, a partir d'una funció matemàtica, s'obté una sortida. En la idea inicial del perceptró, les entrades no tenen associades pesos.

La funció matemàtica bàsicament es tracta d'una suma ponderada amb els pesos associats a cada un d'ells i posteriorment es passa per una funció d'activació, aquesta funció es pot entendre com que si el valor de la suma ponderada és més gran que un valor es tornarà un 1 i sinó serà un 0.

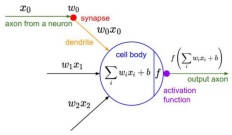


Fig. 16: Perceptró de Rosenblatt.

A.5.2 DESCENS DE GRADIENT

El descens de gradient és un algoritme que permet optimitzar funcions de forma iterativa buscant el mínim global. En Machine Learning s'utilitza per optimitzar una funció de cost entre el valor predit per l'algoritme que s'ha implementat i el valor real, aquest valor real és d'un conjunt d'entrenament, d'aquesta forma es busca el mínim error fent que per futures prediccions el valor predit s'aproxima molt més al valor real que pugui tenir.

El descens de Gradient treballa a partir de la derivada, de forma que si tenim un espai com el de la Figura 17, on es representa el cost i com veiem el descens de gradient va fent passos en la direcció del mínim.

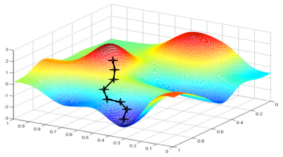


Fig. 17: Funcionament del Descens de Gradient.

També es pot veure d'una forma més visual en la Figura 18, on es representa en dues dimensions, en l'eix de les Y hi ha l'error i en l'eix de les X hi ha els pesos, es pot veure com el descens de gradient va recalculant els pesos per aproximar-se al punt òptim.

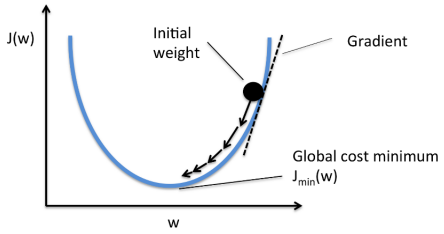


Fig. 18: Descens de Gradient en dues dimensions.

Bàsicament el descens de gradient el que fa és donada una funció de cost, com per exemple la MSE (error quadràtic mig) calcula la seva derivada respecte els pesos i actualitza els pesos per minimitzar l'error. L'actualització de pesos es farà de la següent forma per la funció de cost MSE:

$$\theta_j \leftarrow \theta_j - \alpha \frac{\delta}{\delta \theta_j} J(\theta)$$

L'α és un paràmetre que s'anomena coeficient d'aprenentatge que indiquen com de ràpid es faran els passos cap al punt òptim, si l'α és molt petita tardarà molt a convergir l'algoritme i és probable que ens quedem en un subòptim, per altra banda quan més gran és el coeficient d'aprenentatge tardarà menys però és possible que mai convergeixi el descens de gradient.

A.5.3 BACKPROPAGATION

El Backpropagation es tracta d'un algoritme utilitzat en xarxes neuronals que serveix per actualitzar les neurones i obtenir una millor predicció. Bàsicament es calcula l'error que s'ha obtingut i desde la darrera capa fins l'inici es van actualitzant les neurones i els seus pesos d'acord a la influència que han tingut en l'error. Per minimitzar l'error comès per una neurona en general en les xarxes neuronals s'utilitza la funció del Descens de Gradient.

La funció del Descens de Gradient ajuda a actualitzar les neurones ja que cal calcular els vectors de gradient de cada una de les neurones, és per això que serà important calcular les derivades intermitges de cada una de les funcions. S'han de calcular d'aquesta forma 3 derivades: la derivada del cost respecte funció d'activació, la derivada de la funció d'activació respecte la suma ponderada i la derivada de la suma ponderada respecte els pesos de la neurona.

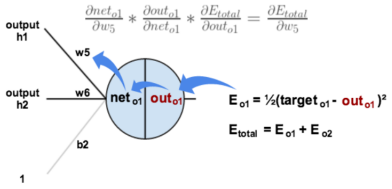


Fig. 19: Backpropagation dins el perceptró.

D'aquesta forma l'algoritme de Backpropagation va propagant els errors a les capes anteriors.

A.5.4 PARÀMETRES D'UNA XARXA NEURONAL

Quan es crea una xarxa neuronal hi ha alguns paràmetres que s'han de considerar i decidir quin valor donals-hi per tenir un millor predictor, molts d'ells es tracta d'hiperparàmetres, és a dir són paràmetres que només es pot saber el seu efecte mitjançant la prova, no són fàcils de definir quins són millors, a continuació veurem aquests paràmetres:

- **Nombre de capes ocultes i el nombre de perceptrons que tindrà cada una de les capes**: normalment, el més recomanable per trobar el punt òptim ràpidament és començar amb capes amb poques neurones i afegirne en cas de ser necessari, de la mateixa forma amb el nombre de capes. Quan més complex sigui, en general més capes i més neurones per capa seran necessàries per obtenir bons resultats.
- **Dropout**: és una tècnica de regularització per evitar l'overfitting. Consisteix en que en cada iteració s'anulen un nombre de neurones que s'escolleixen aleatòriament.

- **Inicialització de pesos**: els pesos de les entrades a les neurones es poden inicialitzar de diferents formes segons la funció d'activació, la tècnica més utilitzada és utilitzar una distribució uniforme.

- **Funció d'activació**: és gràcies a ella que les xarxes neuronals poden simular funcions no lineals. Alguns exemples són: ReLu, sigmoidea, tanh.

- **Coeficient Aprenentatge**: el coeficient d'aprenentatge del descens de gradient ens defineix la forma en la que l'algoritme convergeix. Es poden utilitzar tècniques com la Decaying Learning Rate on el que es fa és començar per un coeficient més gran i es va disminuint poc a poc.

- **Momentum**: ajudar a trobar el mínim global de la funció de cost, evitant que l'algoritme s'estanqui en mínims locals.

- **Nombre Epochs**: és el nombre de vegades que les dades d'entrenament es mostren mentres s'entrena. Per trobar l'òptim es va augmentant el nombre d'epochs fins que la precisió del conjunt de validació comença a decaure, aquest serà el punt òptim. Es pot entendre com el número d'iteracions.

- **Mida del Batch**: Aquest paràmetre defineix el nombre de mostres que es donen a la xarxa després de l'actualització de cada paràmetre.

A.6 Resultats Algoritmes de predicció

En la següent taula es pot observar els resultats dels diferents algoritmes que s'han provat inicialment, a partir dels quals s'han sel·leccionat els models SVM i HistGradientBoosting:

TAULA 4: RESULTATS ALGORITMES DE PREDICCIÓ.

Model	accuracy	f1 score	recall	temps test
SVM rbf	0.999420	0.999437	0.999437	0.129571
SVM sigmoide	0.500000	0.642215	0.749520	11.033297
SVM polinomi	0.999949	0.999923	0.999923	0.021345
SVM linear	0.999086	0.999182	0.999181	0.071629
Logistic Regression	0.998994	0.999105	0.999105	0.023502
Gaussian Naive Bayes	0.994591	0.991933	0.991890	0.019007
Linear Discriminant Analysis	0.996694	0.997799	0.997800	0.024935
Decision Tree	0.999813	0.999872	0.999872	0.011418
K Nearest Neighbors	0.999743	0.999821	0.999821	8.562271
Extra Trees	0.999933	0.999949	0.999949	0.072164
Random Forest	0.999915	0.999923	0.999923	0.442206
HistGradientBoosting	0.999966	0.999949	0.999949	0.048980
ADABoosting	0.999914	0.999923	0.999923	0.232996
Bagging Classifier	0.994505	0.991807	0.991762	0.077892
GradientBoostingClassifier	0.999914	0.999923	0.999923	0.016349

També veurem en més detall en aquesta secció els resultats dels algoritmes que s'han sel·leccionat. Primerament veurem els resultats pel model HistGradientBoosting.

Podem observar en la Figura 20 la corba precision-recall pel model HistGradientBoosting i en la Figura 21 la corba ROC pel model HistGradientBoosting:

També veurem els resultats pel model SVM polinomial. Podem observar en la Figura 22 la corba precision-recall pel model SVM polinomi i en la Figura 23 la corba ROC pel model SVM polinomi:

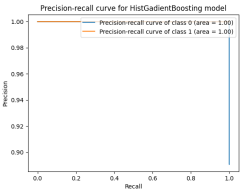


Fig. 20: Corba precision-recall per model HistGradientBoosing.

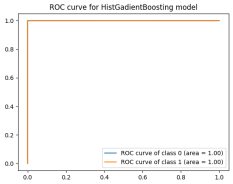


Fig. 21: Corba ROC per model HistGradientBoosing.

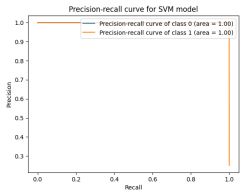


Fig. 22: Corba precision-recall per model SVM polinomi.

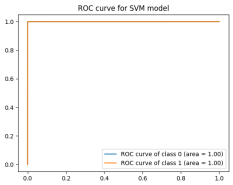


Fig. 23: Corba ROC per model SVM polinomi.

Es pot observar clarament que s'apropen molt a l'òptim en totes les corbes, indicant que els models ofereixen un molt bon rendiment per aquestes mètriques, quasi perfecte.

També podem veure l'evolució de l'accuracy en 2000 iteracions pel model de xarxa neuronal que hem creat, en la Figura 24. A partir d'unes 750 iteracions, l'accuracy no millora, és per això que s'han definit 1000 iteracions perquè ja s'aconsegueix la màxima precisió, tot i que a vegades depèn de l'execució i calen més iteracions per aconseguir aquesta precisió en la majoria d'iteracions ja s'aconsegueix la màxima precisió, ja que depèn de les inicialitzacions aleatòries d'alguns paràmetres de la xarxa.

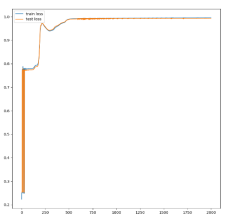


Fig. 24: Evolució de l'accuracy en relació el número d'EPOCHS (2000) dataset de Ransomware.