ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ
ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

Malware detection with machine learning

Διπλωματική Εργασία

ΕΥΡΙΠΙΔΗΣ ΣΤΕΦΑΝΙΔΗΣ

Θεσσαλονίκη, ΝΟΕΜΒΡΙΟΣ 2022

MALWARE DETECTION WITH MACHINE LEARNING
ΕΥΡΙΠΙΔΗΣ ΣΤΕΦΑΝΙΔΗΣ

ΗΛΕΚΤΡΟΛΟΓΟΣ ΜΗΧΑΝΙΚΟΣ ΤΕΙ

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής/τρια
Κωνσταντίνος Ψαννης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την ηη/μμ/εεεε

Ονοματεπώνυμο 1	Ονοματεπώνυμο 2	Ονοματεπώνυμο 3
	Ξυνόγαλος Στυλιανός	Μαντάς Μιχαήλ
.....	.....	.....

Πληκτρολογήστε εδώ το ονοματεπώνυμο σας

ΕΥΡΙΠΙΔΗΣ ΣΤΕΦΑΝΙΔΗΣ

.....

Περίληψη

Στόχος της παρούσας διπλωματικής εργασίας αποτελεί η ανάπτυξη μεθόδου που βασίζεται σε τεχνικές μηχανικής μάθησης για την αυτόματη ανίχνευση malware. Για το τον σκοπό της αναζήτησης κατάλληλων υπερπαραμέτρων των μοντέλων και της επιλογής χαρακτηριστικών, χρησιμοποιήθηκε τυχαίος διαχωρισμός δεδομένων εκπαίδευσης – ελέγχου για 3 μοντέλα ταξινομητών: τυχαίο δάσος (RF), Decision Tree Classification, Ταξινομητής bagging. Τα αποτελέσματα δείχνουν ότι παρόλο που όλοι οι ταξινομητές μπόρεσαν να αναγνωρίσουν Malware τα καλυτέρα αποτελέσματα είχε ο ταξινομητής bagging και ο random forest.

Λέξεις Κλειδιά:

Random Forest (RF), Decision Tree Classification, Bagging classifier, malware, ταξινομητές, μηχανικήμάθηση.

Abstract

Therefore, the aim of this thesis is to develop a method based on machine learning techniques for automatic malware detection. For the purpose of searching for suitable hyperparameters of the models and feature selection, we used random splitting of training-control data for 3 classifier models:

Random Forest (RF), Decision Tree Classification, Bagging classifier. The results show that although all classifiers were able to identify Malware the best results were obtained by the bagging classifier and random forest.

Keywords:

Random Forest (RF), Decision Tree Classification, Bagging classifier, malware.

Πίνακας περιεχομένων

Κατάλογος Εικόνων

Κατάλογος Πινάκων

1 ΚΕΦΑΛΑΙΟ MALWARE

1.1 Τύποι Ανάλυσης Κακόβουλου Λογισμικού

Η ανάλυση μπορεί να διεξάγεται με τρόπο στατικό, δυναμικό ή υβριδικό των δύο μεθόδων.

1.1.1 Στατική Ανάλυση

Η βασική στατική ανάλυση δεν απαιτεί την εκτέλεση του κώδικα. Αντίθετα, η στατική ανάλυση εξετάζει το αρχείο για ενδείξεις κακόβουλης πρόθεσης. Μπορεί να είναι χρήσιμη για τον εντοπισμό κακόβουλων απειλών.Εντοπίζονται τεχνικοί δείκτες όπως ονόματα αρχείων, κατακερματισμοί, συμβολοσειρές όπως διευθύνσεις IP, τομείς και δεδομένα κεφαλίδας αρχείων που μπορούν να χρησιμοποιηθούν για να προσδιοριστεί αν το αρχείο είναι κακόβουλο.

Επιπλέον, εργαλεία όπως οι disassemblers και αναλυτές δικτύου μπορούν να χρησιμοποιηθούν για την παρατήρηση του κακόβουλου λογισμικού χωρίς να εκτελέσουν τον κώδικα στην πραγματικότητα, προκειμένου να συλλέξουν πληροφορίες σχετικά με τον τρόπο λειτουργίας του κακόβουλου λογισμικού.

Ωστόσο, δεδομένου ότι η στατική ανάλυση δεν εκτελεί στην πραγματικότητα τον κώδικα, το κακόβουλο λογισμικό μπορεί να περιλαμβάνει κακόβουλη συμπεριφορά κατά το χρόνο εκτέλεσης που μπορεί να μην εντοπιστεί. Οι επιχειρήσεις έχουν στραφεί στη δυναμική ανάλυση για μια πληρέστερη κατανόηση της γενικότερης συμπεριφοράς του αρχείου.

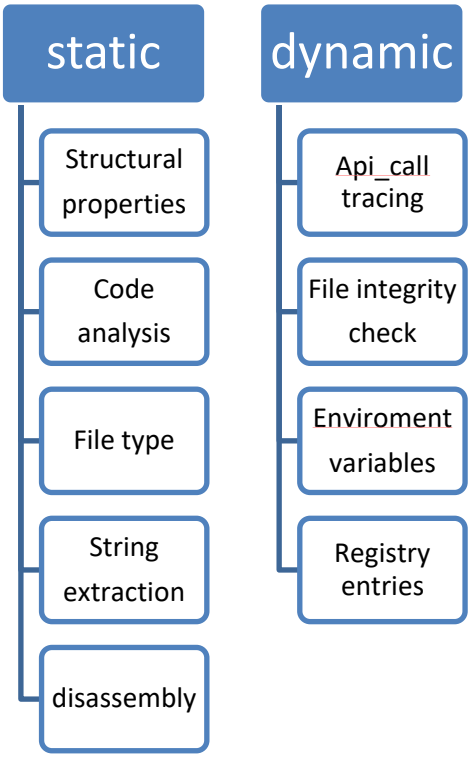
1.1.2 Δυναμική Ανάλυση

Η δυναμική ανάλυση κακόβουλου λογισμικού εκτελεί το ύποπτο κακόβουλο κώδικα σε ένα ασφαλές περιβάλλον που ονομάζεται sandbox [14]. Αυτό το κλειστό σύστημα επιτρέπει στο Security operations center να παρακολουθεί το κακόβουλο λογισμικό σε δράση χωρίς τον κίνδυνο να το αφήσουν να μολύνει το σύστημά τους ή να διαφύγει στο internal network της εκάστοτε επιχείρησης.

Ως δευτερεύον πλεονέκτημα, το sandboxing εξαλείφει το χρόνο που θα χρειαζόταν για την αντίστροφη μηχανική ενός αρχείου για την ανακάλυψη του κακόβουλου κώδικα.

Η πρόκληση με τη δυναμική ανάλυση είναι ότι οι χάκερ γνωρίζουν ότι υπάρχουν sandboxes, οπότε έχουν γίνει πολύ καλοί στο να αποφεύγουν τον εντοπισμό από αυτά.

Για να εξαπατήσουν ένα sandbox, ο κώδικας παραμείνει αδρανής μέχρι να πληρούνται ορισμένες συνθήκες μόνο τότε ο κώδικας εκτελείται.



Εικόνα 1-1: Πως γίνεται η δυναμική και στατική ανάλυση

1.1.3 Υβριδική ανάλυση

Η στατική ανάλυση δεν είναι ένας αξιόπιστος τρόπος για τον εντοπισμό εξελιγμένου κακόβουλου κώδικα και το κακόβουλο λογισμικό μπορεί μερικές φορές να κρυφτεί και από το sandbox. Συνδυάζοντας τεχνικές βασικής και δυναμικής ανάλυσης, η υβριδική ανάλυση παρέχει στην ομάδα ασφαλείας το καλύτερο και από τις δύο προσεγγίσεις -κυρίως επειδή μπορεί να ανιχνεύσει κακόβουλο κώδικα που προσπαθεί να κρυφτεί, και στη συνέχεια μπορεί να εξάγει πολλούς περισσότερους δείκτες συμβιβασμού (IOC) με στατικό και προηγουμένως αθέατο κώδικα. Η υβριδική ανάλυση βοηθά στον εντοπισμό άγνωστων απειλών, ακόμη και εκείνων που προέρχονται από το πιο εξελιγμένο κακόβουλο λογισμικό.

Για παράδειγμα, ένα από τα πράγματα που κάνει η υβριδική ανάλυση είναι η εφαρμογή στατικής ανάλυσης σε δεδομένα που παράγονται από την ανάλυση συμπεριφοράς - όπως όταν εκτελείται ένα κομμάτι κακόβουλου κώδικα και δημιουργεί κάποιες αλλαγές στη μνήμη.

Η δυναμική ανάλυση θα το ανιχνεύσει αυτό και οι αναλυτές θα ειδοποιηθούν να επιστρέψουν και να εκτελέσουν βασική στατική ανάλυση σε αυτό το memorydump.

Ως αποτέλεσμα, θα δημιουργούνταν περισσότερα IOC(Indicator of compromise**)** [13] και θα αποκαλύπτονταν zero days vulnerabilities.

1.2 ΣύγχρονηΚατάστασησχετικά με τα Malwares

1.2.1 Malware YTStealer

Οι χακερς συνήθως διαδίδουν τα malware μέσω διαφημίσεων, ισότοπων phishing, emails με συνημμένα αρχεία που μοιάζουν νόμιμα (υποτιθέμενες προσφορές και μερικές φορές ακόμη και ψεύτικα αιτήματα αναβάθμισης που υποτίθεται ότι αποστέλλονται από νόμιμο λογισμικό). Στην προκειμένη περίπτωση, το YTStealer προωθεί μια ποικιλία συγκεκριμένων εφαρμογών που έχουν σχεδιαστεί για να δελεάσουν τους χρήστες του YouTube.

Αυτές οι εφαρμογές είναι συνήθως εκδόσεις ή ενημερώσεις για εργαλεία επεξεργασίας βίντεο (OBS Studio, Adobe Premierek.λπ.). Μόλις μολυνθεί ένας στόχος, το YTStealer θα εκτελέσει έναν έλεγχο περιβάλλοντος για να διασφαλίσει ότι δεν εκτελείται μέσα σε μια εικονική μηχανή (ή ένα sandbox) και δεν αναλύεται από προγράμματα ασφαλείας.Ο κώδικας που χρησιμοποιείται από το YTStealer για να εκτελέσει τις ενέργειες προέρχεται από το Chacal [15] που υπάρχει στο GitHub.

Εάν το YTStealer ανιχνεύσει ότι αναλύεται, θα αυτοκαταργηθεί. Εάν δεν υπάρχει απειλή,το YTStealer θα αρχίσει να συλλέγει cookies ελέγχου ταυτότητας και διαπιστευτήρια.Το YTStealer θα ανοίξει επίσης το πρόγραμμα περιήγησης στο παρασκήνιο, δηλαδή χωρίςνα εμφανίζεται τίποτα στην οθόνη του υπολογιστή.

Οι χάκερς θα μπορούν στη συνέχεια να κλέβουν cookies και να συνδέονται στη σελίδα στο YouTube. Από εδώ,οι χάκερς μπορούν είτε να δημοσιεύσουν ό,τι θέλουν είτε να συλλέξουν τα δεδομένα.

Αφού το κακόβουλο λογισμικό κλέψει τα κλεμμένα δεδομένα συγκεντρώνονται, κρυπτογραφούνται και αποστέλλονται σε έναν ιδιωτικό διακομιστή που είναι εγγεγραμμένος σε μια νόμιμη εταιρεία

1.2.2 Malware στο Google Play Store

Ερευνητές στον τομέα της κυβερνο-ασφάλειας ανακάλυψαν adware και Malware στο Google Play Store τον περασμένο μήνα, με τουλάχιστον πέντε εφαρμογές να είναι ακόμη διαθέσιμες και να έχουν συγκεντρώσει εκατομμύρια λήψεις από χρηστές.

Τα adware εμφανίζουν ανεπιθύμητες διαφημίσεις που μπορεί να είναι ιδιαίτερα παρεμβατικές, υποβαθμίζουν την εμπειρία του χρήστη, εξαντλούν την μπαταρία, δημιουργούν θερμότητα στην συσκευή και προκαλούν ακόμη και μη εξουσιοδοτημένες χρεώσεις στον χρηστή.

Ωστόσο, τα Trojans που κλέβουν πληροφορίες είναι πολύ πιο κακόβουλα, κλέβουν τα διαπιστευτήρια σύνδεσης για άλλους ισότοπους που επισκέπτεστε, συμπεριλαμβανομένων των λογαριασμών σας στα μέσα κοινωνικής δικτύωσης και των τραπεζικών λογαριασμών σας στο διαδίκτυο.

Συνοψίζοντας ακολουθεί ένας πινάκας που περιγράφει τους κινδύνους που μπορείς να συναντήσει ο χρήστης στο ιντερνέτ.

1.2.3 Επιθέσεις χάκερ σε κρίσιμες δομές

Η ουκρανική εγκατάσταση ηλεκτρικής ενέργειας Prykarpattya Oblenergo το 2016 ήταν μια άλλη περίπτωση επίθεσης από χακερς. Ο μισός πληθυσμός της περιοχής Ivano-Frankivsk στην Ουκρανία έμεινε χωρίς ρεύμα στα μέσα Δεκεμβρίου λόγω επίθεσης malware.

Η ρωσική ομάδα χάκερ Sandworm, χρησιμοποίησε ένα malware με την ονομασία "BlackEnergy 3"[22] .

Πίνακας 1-1: Ταξινόμηση του κακόβουλου λογισμικού

	πολλαπλασιασμός	**μολύνει**	**Αυτό άμυνα**	**ικανότητες**
keylogger	Μολύνει Ιστοσελίδες Usb ή αλλά media	Ευπαθείς browser ή unpatched os ή εφαρμογές	Αντικαθιστά IO device drivers ή API	Συγκεντρώνει τα keystrokes του χρηστή
rootkit	Μολύνει ιστοσελίδες ή εγκαθίσταται από hacker σε server	Ή unpatched os ή εφαρμογές	Αντικατάσταση os kernel-level ή API routines	Συλλογή δεδομένων υποδυόμενη δραστηριότητα χρήστη
Flaws exploits	εκτέλεση εντολών σεελαττωματικό λογισμικό από απομακρυσμένο χρηστή	Ευπαθείς software	Υποδύεται πιστοποιημένους χρηστές	Κατέβασμα και ανέβασμα αρχείων από data repositories
bots	Phishing emails Επικίνδυνα links	Ο χρήστης μπορεί να κάνει μονός του αθελα την εγκατάσταση	Bot updates .παραμένει αδρανής μέχρι να ενεργοποιηθεί	Όταν ενεργοποιείται ο χειρίστης μπορεί να εκτελεί διαφόρους χειρισμούς
Denial of service	Μεγάλα ip packets	Αυτοματοποιεί το packet processing	Ταυτόχρονη επίθεση από πολλές ips	Κατανάλωση υπολογιστικών πόρων και μείωση της απόδοσης του συστήματος

1.3 Τεχνικές ανίχνευσης κακόβουλου λογισμικού

1.3.1 Ανίχνευση με βάση την υπογραφή

Η ανίχνευση με βάση την υπογραφή - χρησιμοποιεί το μοναδικό ψηφιακό αποτύπωμα, γνωστό ως υπογραφή, προγραμμάτων λογισμικού που εκτελούνται σε ένα προστατευμένο σύστημα. Τα προγράμματα προστασίας από ιούς σαρώνουν λογισμικό, αναγνωρίζουν την υπογραφή του και το συγκρίνουν με υπογραφές γνωστού κακόβουλου λογισμικού που διαθέτουν [14].

Τα antivirus χρησιμοποιούν μια μεγάλη βάση δεδομένων, με γνωστές υπογραφές κακόβουλου λογισμικού, που συνήθως συντηρείται από μια ερευνητική ομάδα ασφαλείας, που λειτουργεί από τον προμηθευτή λογισμικών προστασίας από ιούς. Αυτή η βάση δεδομένων ενημερώνεται συχνά και η πιο πρόσφατη έκδοση της συγχρονίζεται με τις προστατευμένες συσκευές που είναι συμβεβλημένες.

Όταν ένα πρόγραμμα προστασίας από ιούς εντοπίζει λογισμικό που πληροί μια γνωστή υπογραφή, σταματά τη διαδικασία και είτε το θέτει σε καραντίνα είτε το διαγράφει. Αυτή είναι μια απλή και αποτελεσματική προσέγγιση για τον εντοπισμό κακόβουλου λογισμικού και είναι σημαντική ως πρώτη γραμμή άμυνας. Ωστόσο, καθώς οι εισβολείς γίνονται πιο εξελιγμένοι, η προσέγγιση που βασίζεται στην υπογραφή δεν μπορεί να εντοπίσει μια μεγάλη ποικιλία νεότερων απειλών που υπάρχουν.

1.3.2 Άθροισμα ελέγχου

Αυτή η μέθοδος είναι ένας τύπος ανάλυσης υπογραφής που περιλαμβάνει τον υπολογισμό των αθροισμάτων ελέγχου κυκλικού πλεονασμού (CRC). Το άθροισμα ελέγχου, βοηθά στην επαλήθευση του ότι τα αρχεία δεν είναι κατεστραμμένα. Το κύριο μειονέκτημα της ανίχνευσης με βάση την υπογραφή είναι η δημιουργία μιας τεράστιας βάσης δεδομένων που δημιουργεί ψευδώς θετικά στοιχεία, τα οποία λειτουργεί το άθροισμα ελέγχου.

Οι χάκερ συχνά χρησιμοποιούν πολυμορφικούς ιούς, για να αποφύγουν τον εντοπισμό με μεθόδους αναγνώρισης που βασίζονται στην υπογραφή. Οι πολυμορφικοί ιοί μπορούν να αλλάξουν τον εαυτό τους– συνήθως ο χάκερ κρυπτογραφεί τυχαία σύνολα εντολών στον κώδικα.

Έτσι, όταν η ομάδα ασφαλείας εντοπίσει μια κακόβουλη υπογραφή, το κακόβουλο λογισμικό δεν περιέχει πλέον το τμήμα κώδικα και δεν μπορεί να βρεθεί.

Η απουσία ανιχνεύσιμης υπογραφής στον κώδικα απαιτεί άλλες τεχνικές ανίχνευσης κακόβουλου κώδικα, όπως:

1. Στατιστική ανάλυση – αναλύει τη συχνότητα των εντολών του επεξεργαστή για να προσδιορίσει εάν ένα αρχείο είναι μολυσμένο.
2. Κρυπτανάλυση – η κρυπτανάλυση γνωστού απλού κειμένου αποκωδικοποιεί κρυπτογραφημένους ιούς χρησιμοποιώντας ένα σύστημα εξισώσεων - Το σύστημα κρυπτανάλυσης αναδομεί τον αλγόριθμο και τα κλειδιά του προγράμματος αποκρυπτογράφησης, εφαρμόζοντας τον αλγόριθμο σε κωδικοποιημένα κομμάτια, για την αποκωδικοποίηση του συνολικού σώματος του κρυπτογραφημένου ιού.
3. Ομάδα ανάλυσης – μια ομάδα ανίχνευσης κακόβουλου λογισμικού, σαρώνει και αναλύει δεδομένα συμπεριφοράς, για να εντοπίσει υπόπτη δραστηριότητα. Η ομάδα πρέπει να αναζητήσει κακόβουλο κώδικα που σχετίζεται με ύποπτη συμπεριφορά.Η ομάδα ασφαλείας μπορεί στη συνέχεια να ιεραρχήσει και να διερευνήσει περαιτέρω ύποπτα περιστατικά.

1.3.3 Λίστα επιτρεπόμενων εφαρμογών

Οι λίστες επιτρεπόμενων εφαρμογών είναι το αντίθετο από την προσέγγιση με βάση την υπογραφή. Αντί να ορίζει ποιο λογισμικό θα πρέπει να αποκλείει το πρόγραμμα προστασίας από ιούς, διατηρεί μια λίστα με εγκεκριμένες εφαρμογές και αποκλείει οτιδήποτε άλλο.

Αυτή η λύση δεν είναι τέλεια, αλλά μπορεί να είναι εξαιρετικά αποτελεσματική, ειδικά σε περιβάλλοντα υψηλής ασφάλειας. Είναι αρκετά σύνηθες οι νόμιμες εφαρμογές να έχουν ευπάθειες ασφαλείας ή να εισάγουν περιττά χαρακτηριστικά, που αυξάνουν το ευρος επίθεσης. Σε ορισμένες περιπτώσεις, η ίδια η εφαρμογή είναι δυνητικά επικίνδυνη αλλά η χρήση της θα μπορούσε να εκθέσει τη συσκευή σε απειλές – για παράδειγμα, σε ορισμένα περιβάλλοντα, μπορεί να χρειαστεί να αποκλειστεί η περιήγηση στον ιστό και το ηλεκτρονικό ταχυδρομείο.

Η λίστα επιτρεπόμενων εφαρμογών, λειτουργεί καλύτερα με συσκευές που είναι αυστηρά εστιασμένες σε εργασίες, όπως διακομιστές ιστού και συσκευές Διαδικτύου των πραγμάτων (IoT).

1.3.4 Ανάλυση Συμπεριφοράς - Μηχανικής Μάθησης

Οι παραπάνω τεχνικές είναι γνωστές ως τεχνικές «στατικής» ανίχνευσης, επειδή βασίζονται σε δυαδικούς κανόνες, που είτε ταιριάζουν είτε όχι με μια διαδικασία που εκτελείται στο περιβάλλον.

Δυναμικές τεχνικές, που βασίζονται στην τεχνητή νοημοσύνη και τη μηχανική μάθηση μπορούν να βοηθήσουν τα εργαλεία ασφαλείας να «μάθουν» ώστε να κάνουν διαφοροποίηση μεταξύ νόμιμων και κακόβουλων αρχείων και διαδικασιών, ακόμα κι αν δεν ταιριάζουν με κάποιο γνωστό μοτίβο ή υπογραφή. Αυτό το κάνουν παρατηρώντας τη συμπεριφορά των αρχείων, την κυκλοφορία του δικτύου, τη συχνότητα των διαδικασιών, τα μοτίβα ανάπτυξης και πολλά άλλα. Με την πάροδο του χρόνου, αυτοί οι αλγόριθμοι μπορούν να μάθουν πώς μοιάζουν τα «κακά» αρχεία, καθιστώντας δυνατό τον εντοπισμό νέου και άγνωστου κακόβουλου λογισμικού.

Η ανίχνευση κακόβουλου λογισμικού με την βοηθεια AI/ML είναι γνωστή ως ανίχνευση συμπεριφοράς, επειδή βασίζεται σε ανάλυση της συμπεριφοράς ύποπτων διεργασιών. Αυτοί οι αλγόριθμοι έχουν ένα όριο για κακόβουλη συμπεριφορά και εάν ένα αρχείο ή διεργασία εμφανίσει ασυνήθιστη συμπεριφορά που ξεπερνά το όριο, προσδιορίζουν ότι είναι κακόβουλο.

Η ανάλυση συμπεριφοράς είναι ισχυρή, αλλά μερικές φορές μπορεί να χάσει κακόβουλες διεργασίες ή να ταξινομήσει εσφαλμένα τις νόμιμες διαδικασίες ως κακόβουλες. Επιπλέον, οι εισβολείς μπορούν να χειριστούν τις διαδικασίες εκπαίδευσης AI/ML.

1.4 τεχνολογίες ανίχνευσης κακόβουλου λογισμικού

Ενώ πολλοί οργανισμοί βασίζονται σε παλαιού τύπου προγραμμάτων έναντι των ιών, ως στρατηγική για την ανίχνευση κακόβουλου λογισμικού, οι σύγχρονοι οργανισμοί ασφαλείας χρησιμοποιούν συνήθως δύο τύπους προηγμένων λύσεων για την άμυνα τους από κακόβουλο λογισμικό – πλατφόρμες προστασίας τελικού σημείου και λύσεις εντοπισμού και απόκρισης τελικού σημείου.

1.4.1 Πλατφόρμες προστασίας τελικού σημείου (EPP)

Τα EPP αναπτύσσονται σε τελικά σημεία όπως σταθμοί εργασίας υπαλλήλων, διακομιστές και πόροι που βασίζονται σε υπολογιστικό νέφος (cloudcomputing). Λειτουργούν ως πρώτη γραμμή άμυνας, που μπορεί να εντοπίσει απειλές και να τις αποκλείσει, προτού προκαλέσουν ζημιά στους σέρβερ η να υποκλέψουν στοιχειά.

Τα EPP χρησιμοποιούν πολλαπλές τεχνικές, για τον εντοπισμό και τον αποκλεισμό κακόβουλου λογισμικού:

Στατική ανάλυση – Τα EPP αξιοποιούν τις παραδοσιακές μεθόδους στατικής ανάλυσης για να εντοπίσουν γνωστά στελέχη κακόβουλου λογισμικού και να επιτρέψουν/απορρίψουν εφαρμογές που έχουν επισημανθεί από τους διαχειριστές ως επικίνδυνες.

Ανάλυση συμπεριφοράς – Τα EPP προσθέτουν ανάλυση συμπεριφοράς για τον εντοπισμό άγνωστων απειλών ή γνωστού κακόβουλου λογισμικού, που χρησιμοποιεί τακτικές αποφυγής, όπως μετάλλαξη ή κρυπτογράφηση.

Επιθεώρηση Sandbox– Τα EPP μπορούν να εκτελούν ύποπτο περιεχόμενο σε sandbox, απομονωμένο από το κύριο λειτουργικό σύστημα. Αυτό καθιστά δυνατή την εκτέλεση ενός αρχείου, την παρακολούθηση της συμπεριφοράς του και την επιβεβαίωση εάν είναι πραγματικά κακόβουλο ή όχι.

Αφόπλιση και ανασυγκρότηση περιεχομένου (CDR) – Τα EPP καθιστούν δυνατή την αφαίρεση κακόβουλων στοιχείων και επιτρέπουν στον χρήστη να έχει πρόσβαση ο ίδιος στο περιεχόμενο. Για παράδειγμα, εάν ένα έγγραφο του Word έχει μια κακόβουλη μακροεντολή, το CDR μπορεί να καταργήσει τη μακρο εντολή και να επιτρέψει στον χρήστη να έχει πρόσβαση στο αρχείο, αντί να το αποκλείσει εντελώς.

1.4.2 Ανίχνευση και απόκριση τελικού σημείου (EDR)

Οι λύσεις EDR συμπληρώνουν τις λύσεις EPP, επιτρέποντας στις ομάδες ασφαλείας να εντοπίζουν και να ανταποκρίνονται γρήγορα σε επιθέσεις σε συσκευές τελικού σημείου. Εάν το EPP απέτυχε να περιορίσει μια απειλή, το EDR καθιστά δυνατό:

Ειδοποιήσεις διαλογής και διερεύνησης – Το EDR παρέχει δεδομένα που επιτρέπουν στους αναλυτές ασφαλείας να εντοπίζουν σημάδια επίθεσης και να τα διερευνούν για να επιβεβαιώσουν ένα περιστατικό ασφαλείας.

Όταν ένας αναλυτής επιβεβαιώσει μια απειλή σε ένα τελικό σημείο, μπορεί να χρησιμοποιήσει την πλατφόρμα EDR για την απόκριση σε περιστατικό. Για παράδειγμα, οι αναλυτές μπορούν να θέσουν σε καραντίνα όλες τις συσκευές που επηρεάζονται από κακόβουλο λογισμικό, να «καθαρίσουν» και να αποτυπώσουν εκ νέου μολυσμένα τελικά σημεία και να «τρέξουν» πρωτόκολλα ασφαλείας.

1.5 Προηγμένη προστασία κακόβουλου λογισμικού

Προηγμένης ανίχνευσης και απόκρισης απειλών, παρέχει προστασία έναντι απειλών από την πρώτη στιγμή, η οποία περιλαμβάνει προηγμένες επίμονες απειλές (APT), προηγμένο κακόβουλο λογισμικό και Trojan, που μπορούν να αποφύγουν τα παραδοσιακά μέτρα ασφαλείας που βασίζονται σε υπογραφές.

1.5.1 Αποκλεισμός συμπεριφοράς που μοιάζει επικίνδυνη

Αρχικά παρακολουθεί τη μνήμη τελικών σημείων, για να ανακαλύψει μοτίβα συμπεριφοράς που είναι συνήθως επικίνδυνο, όπως ένα ασυνήθιστο αίτημα χειρισμού διεργασίας. Αυτά τα μοτίβα είναι κοινά στη συντριπτική πλειονότητα των επιθέσεων.

1.5.2 Αποκλεισμός κακόβουλου λογισμικού

Χρησιμοποιεί προστασία πολλαπλών επιπέδων κακόβουλου λογισμικού, που περιλαμβάνει στατική ανάλυση μηχανικής μάθησης, sandboxing και παρακολούθηση συμπεριφοράς διεργασιών - Αυτό διασφαλίζει ότι ακόμα κι αν μια απειλή, δημιουργήσει μια σύνδεση με τον εισβολέα και κατεβάσει επιπλέον κακόβουλο λογισμικό θα αποτρέψει την εκτέλεση αυτού του κακόβουλου λογισμικού, ώστε να μην μπορεί να γίνει ζημιά

1.6 Επιθέσεις σε συσκευές IOT

Αυτές τις ημέρες οι επιθέσεις malware έχουν στοχεύσει μαζικά τις συσκευές με περιορισμένους πόρους και δομές όπως το IoT [20] . Οι επιθέσεις malware γίνονται σε συσκευές IoT για πολλαπλούς σκοπούς. Ορισμένοι επιτιθέμενοι προσπαθούν να κλέψουν διαπιστευτήρια, ενώ κάποιοι επιτιθέμενοι προσπαθούν να εγκαταστήσουν το κακόβουλο λογισμικό σε αυτές τις συσκευές και να αποκτήσουν πρόσβαση.

Ωστόσο, το IoT ενέχει κινδύνους, λόγω του γεγονότος ότι οι χάκερς έχουν την ικανότητα να βρίσκουν κενά ασφαλείας στις συσκευές IoT ως εκ τούτου, εισβάλλουν σε αυτές για κακόβουλες δραστηριότητες.

Ως αποτέλεσμα, μπορούν να ελέγχουν πολλές συνδεδεμένες συσκευές σε ένα δίκτυο IoT, μετατρέποντας το IoT σε Botnet of Things (BoT). Στο ένα botnet, οι χάκερ μπορούν να εξαπολύσουν διάφορους τύπους επιθέσεων, όπως οι γνωστές επιθέσεις κατανεμημένης άρνησης παροχής υπηρεσιών (DDoS) και Man in the Middle (MitM), ή και να διαδώσουν διάφορους τύπους κακόβουλου λογισμικού (malware) στις παραβιασμένες συσκευές του δικτύου IoT [20].

Για την προστασία αυτών των συσκευών, χρειαζόμαστε μια προσέγγιση κατά των επιθέσεων malware που μπορεί να ανιχνεύει επιθέσεις χωρίς να χρησιμοποιεί πολλούς πόρους και εύρος ζώνης.

2 ΚΕΦΑΛΑΙΟ ΜΟΝΤΕΛΑ

2.1 Ταξινομητής bagging

Το Bagging είναι μια τεχνική για τη βελτίωση της ακρίβειας των προβλέψεων που γίνονται από έναν εποπτευόμενο αλγόριθμο εκμάθησης. Η βασική ιδέα είναι να εκπαιδεύσουμε έναν αριθμό διαφορετικών μοντέλων σε διαφορετικά τυχαία επιλεγμένα υποσύνολα δεδομένων εκπαίδευσης και στη συνέχεια να συνδυάσουμε τις προβλέψεις αυτών των μοντέλων χρησιμοποιώντας κάποιο είδος σχήματος ψηφοφορίας.

Το κύριο πλεονέκτημα του bagging είναι ότι μπορεί να βελτιώσει την ακρίβεια ενός μοντέλου χωρίς να διακυβεύεται σημαντικά η διακύμανσή του. Αυτό το καθιστά μια καλή επιλογή για καταστάσεις όπου θέλουμε να μειώσουμε τη διακύμανση των προβλέψεών μας χωρίς να θυσιάσουμε υπερβολική ακρίβεια. Αυτό την καθιστά ιδανική τεχνική για προβλήματα όπου το κόστος του λάθους είναι υψηλό (π.χ. στην ιατρική διάγνωση ή τον εντοπισμό απάτης με πιστωτικές κάρτες), καθώς μας επιτρέπει να ανταλλάξουμε ένα ποσοστό ακρίβειας με μαθηματική ευρωστία.

Το κύριο μειονέκτημα του bagging είναι ότι συνήθως απαιτεί περισσότερα δεδομένα εκπαίδευσης από άλλες τεχνικές μηχανικής εκμάθησης, όπως η ενίσχυση και η στοίβαξη. Αυτό μπορεί να είναι πρόβλημα σε ορισμένες περιπτώσεις όπου δεν υπάρχουν αρκετά διαθέσιμα δεδομένα για την εκπαίδευση όλων των μοντέλων του συνόλου.

2.1.1 Πώς να εφαρμόσετε το bagging με Python;

Υπάρχουν πολλοί διαφορετικοί τρόποι υλοποίησης του bagging με Python, αλλά ο πιο συνηθισμένος είναι η χρήση της κλάσης sklearn.ensemble.BaggingClassifier. Αυτή η κλάση παρέχει μια απλή API για εκπαίδευση και χρήση ενός συνόλου.

```
# Create a BaggingClassifier
from sklearn.ensemble import BaggingClassifier
```

Σε αυτήν την ενότητα, θα δείξουμε πώς να χρησιμοποιήσετε την κλάση BaggingClassifier για να δημιουργήσετε ένα σύνολο.

Αρχικά, πρέπει να εισαγάγουμε τη βιβλιοθήκη sklearn και την κλάση BaggingClassifier:

```
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics
from sklearn.metrics import confusion_matrix
from sklearn.ensemble import RandomForestClassifier
```

```
from sklearn.metrics import classification_report
```

Στη συνέχεια, μπορούμε να δημιουργήσουμε ένα αντικείμενο BaggingClassifier και να καθορίσουμε τον αριθμό των μοντέλων στο σύνολο:

Αυτό το αντικείμενο θα φροντίσει για οτιδήποτε άλλο χρειάζεται για την εκπαίδευση και τη χρήση του συνόλου. Μπορούμε τώρα να εκπαιδεύσουμε τα μοντέλα δίνοντας ένα σύνολο δεδομένων και ένα σύνολο παραμέτρων:

```
# Train SFS with our dataset
sfs = sfs.fit(X_train,y_train)
```

Η μέθοδος fit() θα εκπαιδεύσει τα μοντέλα και θα τα αποθηκεύσει στην cache για μελλοντική χρήση. Μπορούμε επίσης να καθορίσουμε μια σειρά από άλλες επιλογές, όπως τον τύπο του σχήματος ψηφοφορίας που θα χρησιμοποιηθεί, τον αριθμό των επαναλήψεων που θα εκτελεστούν και τον αριθμό των δειγμάτων που θα χρησιμοποιηθούν για κάθε μοντέλο.

Τέλος, μπορούμε να κάνουμε προβλέψεις για νέα δεδομένα καλώντας τη μέθοδο predict():

```
from sklearn.ensemble import BaggingClassifier
bagging =BaggingClassifier(random_state=1000)
bagging.fit(X_train_bg, y_train)
y_predict_bagging = bagging.predict(X_test_bg)
```

Αυτό θα επιστρέψει μια λίστα με προβλέψεις, μία για κάθε μοντέλο του συνόλου. Μπορούμε στη συνέχεια να έχουμε τον μέσο όρο των προβλέψεων για να πάρουμε το τελικό αποτέλεσμα.

Σε αυτήν την ανάρτηση, ο ταξινομητής bagging δημιουργείται χρησιμοποιώντας το Sklearn BaggingClassifier, με έναν αριθμό εκτιμητών που έχει οριστεί στο 100, το max_features έχει οριστεί στο 10, το max_samples έχει οριστεί στο 100 και η τεχνική δειγματοληψίας που χρησιμοποιείται είναι η προεπιλεγμένη (bagging). Η μέθοδος που εφαρμόζεται είναι τα τυχαία patches, καθώς τα δείγματα και τα χαρακτηριστικά σχεδιάζονται με τυχαίο τρόπο.

2.1.2 Σύγκριση με άλλες τεχνικές μηχανικής εκμάθησης

Το Bagging είναι μια σχετικά απλή τεχνική, αλλά μπορεί να είναι πολύ αποτελεσματική στη μείωση της διακύμανσης των προβλέψεων που γίνονται από έναν εποπτευόμενο αλγόριθμο εκμάθησης (supervised learning algorithm). Συχνά συγκρίνεται με άλλες τεχνικές μηχανικής εκμάθησης, όπως η ενίσχυση και η στοίβαξη.

Η ενίσχυση Boosting είναι μια τεχνική που συνδυάζει πολλά αδύναμα μοντέλα σε ένα μόνο ισχυρό μοντέλο. Αυτό μπορεί να γίνει με διάφορους τρόπους, αλλά ο πιο συνηθισμένος είναι η χρήση ενός σταθμισμένου μέσου όρου των μοντέλων. Το κύριο πλεονέκτημα της ενίσχυσης είναι ότι μπορεί να βελτιώσει την ακρίβεια ενός μοντέλου χωρίς να διακυβεύεται σημαντικά η διακύμανση του.

Η στοίβαξη Stacking είναι μια τεχνική που συνδυάζει πολλά μοντέλα σε ένα ενιαίο, πιο περίπλοκο μοντέλο. Αυτό μπορεί να γίνει με διάφορους τρόπους, αλλά ο πιο συνηθισμένος είναι η χρήση ενός σταθμισμένου μέσου όρου των μοντέλων. Το κύριο πλεονέκτημα της στοίβαξης (Stacking) είναι ότι μπορεί να βελτιώσει την ακρίβεια και να μειώσει την πολυπλοκότητα ενός μοντέλου χωρίς να διακυβεύεται σημαντικά η διακύμανση του.

2.1.3 Συμπεράσματα

Ο ταξινομητής classifier bagging είναι ένας ταξινομητής συνόλου που δημιουργείται χρησιμοποιώντας πολλαπλούς εκτιμητές που μπορούν να εκπαιδευτούν χρησιμοποιώντας διαφορετικές τεχνικές δειγματοληψίας.Ο ταξινομητής bagging βοηθά στη μείωση της διακύμανσης των μεμονωμένων εκτιμητών μέσω της τεχνικής δειγματοληψίας και του συνδυασμού των προβλέψεων.

Εξετάστε το ενδεχόμενο να χρησιμοποιήσετε τον ταξινομητή bagging για έναν αλγόριθμο που οδηγεί σε ασταθείς ταξινομητές (ο ταξινομητής έχει υψηλή διακύμανση). Για παράδειγμα, το δέντρο απόφασης έχει ως αποτέλεσμα την κατασκευή ασταθούς ταξινομητή με υψηλή διακύμανση και χαμηλή προκατάληψη.

2.2 Ο αλγόριθμος Random Forests

Random forests [10] είναι ένας αλγόριθμος μάθησης με επίβλεψη. Μπορεί να χρησιμοποιηθεί τόσο για ταξινόμηση όσο και για παλινδρόμηση. Είναι επίσης ο πιο ευέλικτος και εύκολος στη χρήση αλγόριθμος.Τα τυχαία δάση δημιουργούν δέντρα απόφασης σε τυχαία επιλεγμένα δείγματα δεδομένων, παίρνουν πρόβλεψη από κάθε δέντρο και επιλέγουν την καλύτερη λύση μέσω ψηφοφορίας. Παρέχει επίσης έναν αρκετά καλό δείκτη της σημασίας του χαρακτηριστικού.

Τα τυχαία δάση έχουν ποικίλες εφαρμογές, όπως μηχανές συστάσης επιλογών, ταξινόμηση εικόνων και επιλογή χαρακτηριστικών. Μπορεί να χρησιμοποιηθεί για την ταξινόμηση πιστώτων αιτούντων σε δάνεια, τον εντοπισμό δόλιων δραστηριοτήτων και την πρόβλεψη ασθενειών.

Τεχνικά είναι μια μέθοδος συνόλου δέντρων απόφασης που δημιουργούνται σε ένα τυχαία διαχωρισμένο σύνολο δεδομένων. Αυτή η συλλογή ταξινομητών δέντρων απόφασης είναι επίσης γνωστή ως δάσος. Τα επιμέρους δέντρα αποφάσεων δημιουργούνται χρησιμοποιώντας έναν δείκτη επιλογής χαρακτηριστικών, όπως το κέρδος πληροφορίας, ο λόγος κέρδους και ο δείκτης Gini για κάθε χαρακτηριστικό.

Κάθε δέντρο εξαρτάται από ένα ανεξάρτητο τυχαίο δείγμα. Σε ένα πρόβλημα ταξινόμησης, κάθε δέντρο ψηφίζει και ως τελικό αποτέλεσμα επιλέγεται η πιο δημοφιλής κλάση. Στην περίπτωση της παλινδρόμησης, ως τελικό αποτέλεσμα θεωρείται ο μέσος όρος όλων των αποτελεσμάτων των δέντρων. Είναι απλούστερος και ισχυρότερος σε σύγκριση με τους άλλους μη γραμμικούς αλγορίθμους ταξινόμησης

2.2.1 Πώς λειτουργεί ο αλγόριθμος;

Λειτουργεί σε τέσσερα βήματα

- Επιλογή τυχαίων δειγμάτων από ένα σύνολο δεδομένων.
- Δημιουργούμε ένα δέντρο απόφασης για κάθε σύνολο και λαμβάνουμε ένα αποτέλεσμα για κάθε πρόβλεψη.
- Εκτελέστε μια ψηφοφορία για κάθε προβλεπόμενο αποτέλεσμα.
- Επιλέξτε το πλειοψηφικό αποτέλεσμα έως τελική πρόβλεψη

Οι αλγόριθμοι τυχαίου δάσους έχουν τρεις κύριες υπερπαραμέτρους, οι οποίες πρέπει να οριστούν πριν από την εκπαίδευση. Αυτές περιλαμβάνουν το μέγεθος των κόμβων, τον αριθμό των δέντρων και τον αριθμό των δειγματοληπτικών χαρακτηριστικών. Από εκεί και πέρα, ο ταξινομητής τυχαίου δάσους μπορεί να χρησιμοποιηθεί για την επίλυση προβλημάτων παλινδρόμησης ή ταξινόμησης.

Ο αλγόριθμος random forest αποτελείται από μια συλλογή δέντρων απόφασης και κάθε δέντρο στο σύνολο αποτελείται από ένα δείγμα δεδομένων που αντλείται από ένα σύνολο εκπαίδευσης με αντικατάσταση, το οποίο ονομάζεται δείγμα bootstrap. Από αυτό το δείγμα εκπαίδευσης, το ένα τρίτο αυτού του δείγματος τίθεται στην άκρη ως δεδομένα δοκιμής, γνωστό ως δείγμα out-of-bag (oob).

Ανάλογα με τον τύπο του προβλήματος, ο προσδιορισμός της πρόβλεψης θα διαφέρει. Για ένα έργο παλινδρόμησης, τα μεμονωμένα δέντρα απόφασης θα υπολογίσουν το μέσο όρο, ενώ για ένα έργο ταξινόμησης, η ψήφος πλειοψηφίας -δηλαδή η πιο συχνή κατηγορική μεταβλητή- θα δώσει την προβλεπόμενη κλάση. Τέλος, το δείγμα oob χρησιμοποιείται στη συνέχεια για διασταυρούμενη επικύρωση, οριστικοποιώντας την εν λόγω πρόβλεψη

2.2.2 Πλεονεκτήματα:

Τα τυχαία δάση θεωρούνται ως μια εξαιρετικά ακριβής και ισχυρή μέθοδος λόγω του αριθμού των δέντρων απόφασης που συμμετέχουν στη διαδικασία.

Μειώνει το overfitting - Ο κύριος λόγος είναι ότι λαμβάνει το μέσο όρο όλων των προβλέψεων , μειώνει επίσης tobiasστο εκπαιδευόμενο μοντέλο.Ο αλγόριθμος μπορεί να χρησιμοποιηθεί τόσο σε προβλήματα ταξινόμησης όσο και σε προβλήματα παλινδρόμησης.

Παρέχει ευελιξία: Δεδομένου ότι το random forest μπορεί να χειριστεί τόσο τις εργασίες παλινδρόμησης όσο και τις εργασίες ταξινόμησης με υψηλό βαθμό ακρίβειας, είναι μια δημοφιλής μέθοδος.

Εύκολος προσδιορισμός της σημασίας των χαρακτηριστικών: Το τυχαίο δάσος καθιστά εύκολη την αξιολόγηση της σημασίας των μεταβλητών ή της συμβολής τους στο μοντέλο. Υπάρχουν μερικοί τρόποι για την αξιολόγηση της σημασίας των χαρακτηριστικών – το Gini και η (MDI) χρησιμοποιούνται συνήθως για να μετρήσουν πόσο μειώνεται η ακρίβεια του μοντέλου όταν αποκλείεται μια συγκεκριμένη μεταβλητή.

Model 2:RandomForestClassifier

In [33]:
```
rf =RandomForestClassifier(criterion='gini', max_depth=8, max_features='sqrt',
                           n_estimators=200)
rf.fit(X_train, y_train)
y_predict_rf = rf.predict(X_test)
# confusion_matrix
cm = confusion_matrix(y_test, y_predict_rf)
sns.heatmap(cm, annot=True, fmt="d")
```

Εικόνα 2-1: κύριες υπερπαραμέτροι

2.2.3 Μειονεκτήματα:

Χρονοβόρα διαδικασία: Δεδομένου ότι οι αλγόριθμοι τυχαίου δάσους μπορούν να χειριστούν μεγάλα σύνολα δεδομένων, μπορούν να παρέχουν ακριβέστερες προβλέψεις, αλλά μπορεί να είναι αργοί στην επεξεργασία δεδομένων, καθώς υπολογίζουν δεδομένα για κάθε μεμονωμένο δέντρο απόφασης.

Απαιτεί περισσότερους πόρους: Δεδομένου ότι τα τυχαία δάση επεξεργάζονται μεγαλύτερα σύνολα δεδομένων, απαιτούν περισσότερους πόρους για την αποθήκευση των δεδομένων αυτών.

Πιο πολύπλοκη: Η πρόβλεψη ενός μεμονωμένου δέντρου απόφασης είναι ευκολότερο να ερμηνευτεί σε σύγκριση με ένα δάσος μεσα από αυτά.

2.3 Decision Tree Classification

Η εκμάθηση δένδρων απόφασης χρησιμοποιεί μια στρατηγική διαίρει και βασίλευε, πραγματοποιώντας μια άπληστη αναζήτηση για τον εντοπισμό των βέλτιστων σημείων διαχωρισμού εντός ενός δένδρου. Αυτή η διαδικασία διάσπασης επαναλαμβάνεται στη συνέχεια με αναδρομικό τρόπο από πάνω προς τα κάτω, έως ότου όλες ή η πλειονότητα των εγγραφών ταξινομηθούν σε συγκεκριμένες ετικέτες κλάσης.

Το κατά πόσον όλα τα σημεία δεδομένων ταξινομούνται ως ομοιογενή σύνολα εξαρτάται σε μεγάλο βαθμό από την πολυπλοκότητα του δέντρου απόφασης. Τα μικρότερα δέντρα είναι πιο εύκολα σε θέση να επιτύχουν αμιγείς κόμβους - δηλαδή σημεία δεδομένων σε μία μόνο κλάση. Ωστόσο, καθώς ένα δέντρο μεγαλώνει σε μέγεθος, γίνεται όλο και πιο δύσκολο να διατηρηθεί αυτή η καθαρότητα και αυτό συνήθως έχει ως αποτέλεσμα να εμπίπτουν πολύ λίγα δεδομένα σε ένα δεδομένο υποδέντρο. Όταν συμβαίνει αυτό, είναι γνωστό ως κατακερματισμός δεδομένων και μπορεί συχνά να οδηγήσει σε υπερπροσαρμογή.

Για να μειωθεί η πολυπλοκότητα και να αποφευχθεί η υπερπροσαρμογή, χρησιμοποιείται συνήθως το κλάδεμα- πρόκειται για μια διαδικασία, η οποία αφαιρεί κλάδους που χωρίζονται σε χαρακτηριστικά με χαμηλή σημασία. Η προσαρμογή του μοντέλου μπορεί στη συνέχεια να αξιολογηθεί μέσω της διαδικασίας της διασταυρούμενης επικύρωσης. Ένας άλλος τρόπος με τον οποίο τα δέντρα απόφασης μπορούν να διατηρήσουν την ακρίβειά τους είναι ο σχηματισμός ενός συνόλου μέσω ενός αλγορίθμου τυχαίου δάσους- αυτός ο ταξινομητής προβλέπει ακριβέστερα αποτελέσματα, ιδίως όταν τα μεμονωμένα δέντρα δεν συσχετίζονται μεταξύ τους.Τα δέντρα αποφάσεων είναι ιδιαίτερα χρήσιμα για εργασίες εξόρυξης (mining).

Ας εξερευνήσουμε τα βασικά οφέλη της χρήσης των δέντρων αποφάσεων πιο κάτω:

2.3.1 Πλεονεκτήματα

Εύκολη ερμηνεία: Η λογική Boole και οι οπτικές αναπαραστάσεις των δέντρων αποφάσεων τα καθιστούν ευκολότερα κατανοητά. Η ιεραρχική φύση ενός δέντρου αποφάσεων καθιστά επίσης εύκολο να δούμε ποια χαρακτηριστικά είναι πιο σημαντικά, κάτι που δεν είναι πάντα σαφές με άλλους αλγορίθμους, όπως τα νευρωνικά δίκτυα [16].

Απαιτείται ελάχιστη προετοιμασία δεδομένων: Τα δέντρα αποφάσεων έχουν ορισμένα χαρακτηριστικά, τα οποία τα καθιστούν πιο ευέλικτα από άλλους ταξινομητές. Μπορούν να χειριστούν διάφορους τύπους δεδομένων - δηλαδή διακριτές ή συνεχείς τιμές.

Πιο ευέλικτα: Τα δέντρα αποφάσεων μπορούν να αξιοποιηθούν τόσο για εργασίες ταξινόμησης όσο και για εργασίες παλινδρόμησης, καθιστώντας τον, πιο ευέλικτο από ορισμένους άλλους αλγορίθμους.

2.3.2 Μειονεκτήματα

Είναι επιρρεπής στην υπερβολική προσαρμογή(overfitting): Τα πολύπλοκα δέντρα αποφάσεων τείνουν να υπερπροσαρμόζονται και δεν γενικεύονται καλά σε νέα δεδομένα παρουσιάζοντας χαμηλά ποσοστά σφάλματος και η υψηλή διακύμανση.

Εκτιμητές υψηλής διακύμανσης: Μικρές παραλλαγές εντός των δεδομένων μπορούν να δημιουργήσουν ένα πολύ διαφορετικό δέντρο απόφασης. Το bagging, ή ο μέσος όρος των εκτιμήσεων, μπορεί να είναι μια μέθοδος μείωσης της διακύμανσης των δέντρων απόφασης. Ωστόσο, η προσέγγιση αυτή είναι περιορισμένη, καθώς μπορεί να οδηγήσει σε υψηλά συσχετιζόμενους προγνωστικούς παράγοντες.

Πιο δαπανηρό σε πόρους: Δεδομένου ότι τα δέντρα αποφάσεων ακολουθούν μια προσέγγιση άπληστης αναζήτησης κατά την κατασκευή τους, μπορεί να είναι πιο δαπανηρή η εκπαίδευσή τους σε σύγκριση με άλλους αλγορίθμους.

3 Κεφάλαιο 3ο: Ανάλυση - Επεξεργασία Δεδομένων – Μεθοδολογία

3.1 Ανάλυση δεδομένων

Το σύνολο των δεδομένων μας αποτελείται από 57 στήλες και 10539 γραμμές.

```
df.shape
```

```
(10539, 57)
```

Εικόνα 3-1: αποτέλεσμα εντολής df.shape

```
]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10539 entries, 0 to 10538
Data columns (total 57 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   Name                         10539 non-null  object
 1   md5                          10539 non-null  object
 2   Machine                      10539 non-null  int64
 3   SizeOfOptionalHeader         10539 non-null  int64
 4   Characteristics              10539 non-null  int64
 5   MajorLinkerVersion           10539 non-null  int64
 6   MinorLinkerVersion           10539 non-null  int64
 7   SizeOfCode                   10539 non-null  int64
 8   SizeOfInitializedData        10539 non-null  int64
 9   SizeOfUninitializedData      10539 non-null  int64
 10  AddressOfEntryPoint          10539 non-null  int64
 11  BaseOfCode                   10539 non-null  int64
 12  BaseOfData                   10539 non-null  int64
 13  ImageBase                    10539 non-null  int64
 14  SectionAlignment             10539 non-null  int64
 15  FileAlignment                10539 non-null  int64
 16  MajorOperatingSystemVersion  10539 non-null  int64
```

Εικόνα 3-2: αποτέλεσμα εντολής df.info()

RangeIndex: 10539 entries, 0 to 10538

Data columns (total 57 columns):

#	Column	Non-Null Count	Dtype
0	Name	10539 non-null	object
1	md5	10539 non-null	object
2	Machine	10539 non-null	int64
3	SizeOfOptionalHeader	10539 non-null	int64
4	Characteristics	10539 non-null	int64
5	MajorLinkerVersion	10539 non-null	int64
6	MinorLinkerVersion	10539 non-null	int64

7	SizeOfCode	10539 non-null int64
8	SizeOfInitializedData	10539 non-null int64
9	SizeOfUninitializedData	10539 non-null int64
10	AddressOfEntryPoint	10539 non-nullint64
11	BaseOfCode	10539 non-nullint64
12	BaseOfData	10539 non-nullint64
13	ImageBase	10539 non-nullint64
14	SectionAlignment	10539 non-nullint64
15	FileAlignment	10539 non-nullint64
16	MajorOperatingSystemVersion	10539 non-nullint64
17	MinorOperatingSystemVersion	10539 non-nullint64
18	MajorImageVersion	10539 non-nullint64
19	MinorImageVersion	10539 non-nullint64
20	MajorSubsystemVersion	10539 non-nullint64
21	MinorSubsystemVersion	10539 non-nullint64
22	SizeOfImage	10539 non-nullint64
23	SizeOfHeaders	10539 non-nullint64
24	CheckSum	10539 non-nullint64
25	Subsystem	10539 non-nullint64
26	DllCharacteristics	10539 non-nullint64
27	SizeOfStackReserve	10539 non-nullint64
28	SizeOfStackCommit	10539 non-nullint64
29	SizeOfHeapReserve	10539 non-nullint64
30	SizeOfHeapCommit	10539 non-nullint64
31	LoaderFlags	10539 non-nullint64
32	NumberOfRvaAndSizes	10539 non-nullint64
33	SectionsNb	10539 non-nullint64
34	SectionsMeanEntropy	10539 non-nullfloat64
35	SectionsMinEntropy	10539 non-nullfloat64
36	SectionsMaxEntropy	10539 non-nullfloat64
37	SectionsMeanRawsize	10539 non-nullfloat64
38	SectionsMinRawsize	10539 non-nullint64
39	SectionMaxRawsize	10539 non-nullint64
40	SectionsMeanVirtualsize	10539 non-nullfloat64

41SectionsMinVirtualsize	10539 non-nullint64
42SectionMaxVirtualsize	10539 non-nullint64
43ImportsNbDLL	10539 non-nullint64
44 ImportsNb	10539 non-null int64
45 ImportsNbOrdinal	10539 non-null int64
46 ExportNb	10539 non-null int64
47 ResourcesNb	10539 non-null int64
48 ResourcesMeanEntropy	10539 non-null float64
49 ResourcesMinEntropy	10539 non-null float64
50 ResourcesMaxEntropy	10539 non-null float64
51 ResourcesMeanSize	10539 non-null float64
52 ResourcesMinSize	10539 non-null int64
53 ResourcesMaxSize	10539 non-null int64
54 LoadConfigurationSize	10539 non-null int64
55 VersionInformationSize	10539 non-null int64
56 legitimate	10539 non-null int64

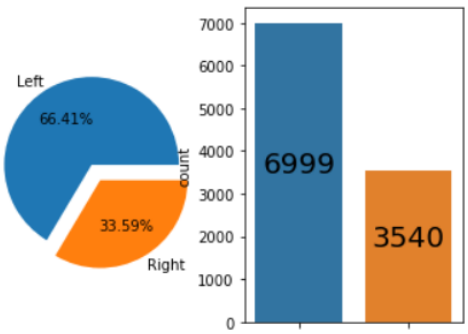
dtypes: float64(9), int64(46), object(2)

memoryusage: 4.6+ MB

3.1.1 Μη ισορροπημένα δεδομένα

```
]: plt.subplot(121)
   plt.pie(x = df.groupby(['legitimate']).legitimate.count().to_list(),
           labels = ["Left", "Right"], autopct='%1.2f%%', explode = (0, 0.2))

   plt.subplot(122)
   sns.countplot(data = df, x = 'legitimate')
   zero, one = df.legitimate.value_counts()
   plt.text(1, one//2, one, fontsize = 20,  horizontalalignment='center')
   plt.text(0, zero//2, zero, fontsize = 20, horizontalalignment='center')
```

```
]: Text(0, 3499, '6999')
```



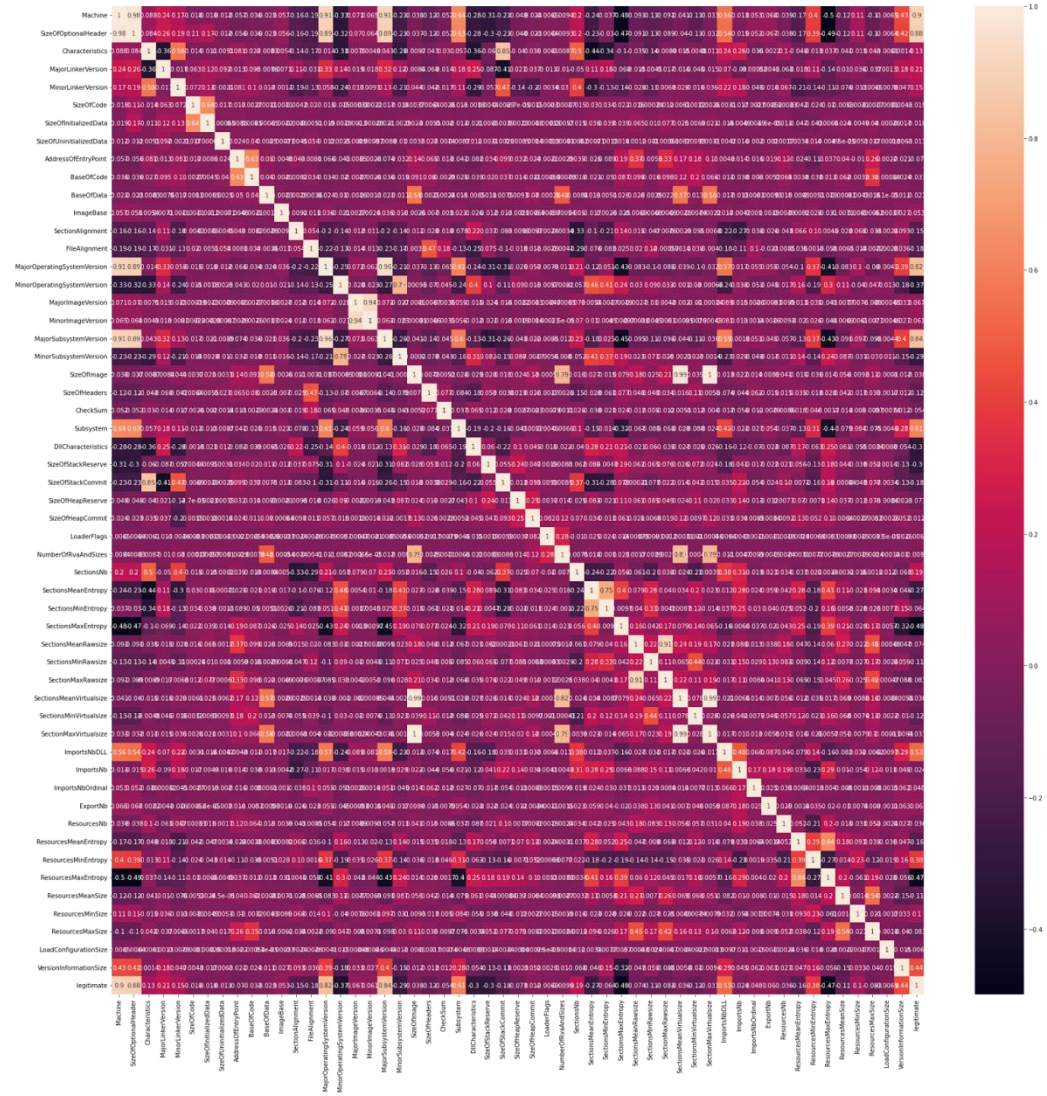
Εικόνα 3-3: μη ισορροπημένα δεδομένα

Το Datasheet είναι inbalanced .σε γενικότερες περίπτωσης η ακρίβεια μπορεί να είναι καλή για μια κλάση που έχει περισσότερα σημεία δεδομένων, αλλά για μια άλλη κλάση που έχει λίγα σημεία δεδομένων θα έχει πολύ κακή απόδοση αλλάστην δική μας περίπτωση δεν έγινε προσπάθεια εξισορροπήσεις γιατί τα αποτελέσματα στην ανίχνευση malware ήταν πολύ καλά.

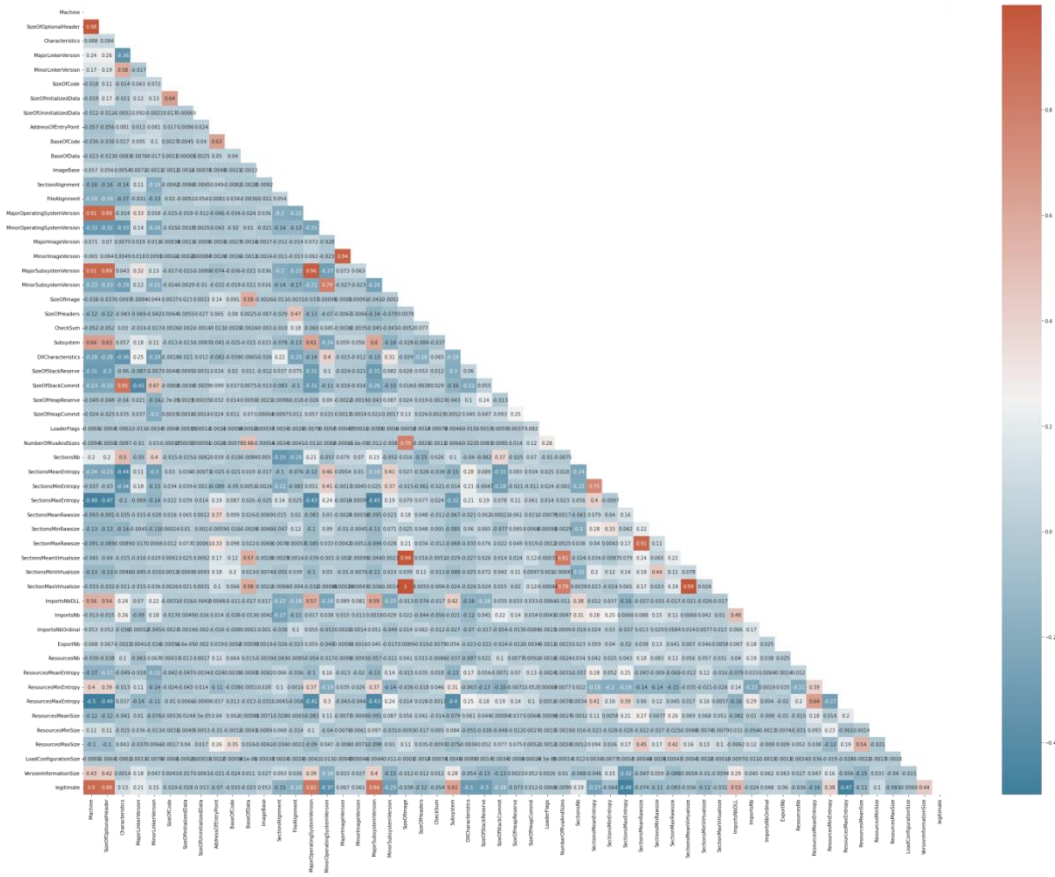
3.1.2 συσχέτιση (correlation)

Η συσχέτιση δείχνει τη σχέση μεταξύ των χαρακτηριστικών. Για παράδειγμα, εάν ένα χαρακτηριστικό αυξάνεται, ένα άλλο χαρακτηριστικό μπορεί να μειωθεί ή να αυξηθεί. Η συσχέτιση δείχνει δείχνει την αναλογικότητα μεταξύ των σημείων των χαρακτηριστικών.

Στο παρακάτω σχήμα χρησιμοποιείται ο χάρτης θερμότητας () για να δείξει μια αναπαράσταση των δεδομένων σε δύο διαστάσεις. Τα πιο σκούρα χρώματα δείχνουν την υψηλότερη αρνητική συσχέτιση και το πιο ανοιχτό χρώμα δείχνει την υψηλή θετική συσχέτιση μεταξύ δύο χαρακτηριστικών.



Εικόνα 3-4: Θερμικός Πίνακας συσχέτισης 1



Εικόνα 3-5: Θερμικός Πίνακας συσχέτισης 2

Δείχνει την αναλογικότητα μεταξύ των σημείων των χαρακτηριστικών. Στο παρακάτω σχήμα χρησιμοποιείται ο χάρτης θερμότητας () για να δείξει μια αναπαράσταση των δεδομένων σε δύο διαστάσεις. Τα πιο σκούρα χρώματα δείχνουν την υψηλότερη αρνητική συσχέτιση και το πιο ανοιχτό χρώμα δείχνει την υψηλή θετική συσχέτιση μεταξύ δύο χαρακτηριστικών.

3.1.3 Εύρεση υψηλά συσχετιζόμενων χαρακτηριστικών

```
ResourcesMaxEntropy              0.470880
SectionsMaxEntropy               0.476792
ImportsNbDLL                     0.526379
Subsystem                        0.606242
MajorOperatingSystemVersion      0.823720
MajorSubsystemVersion            0.840383
SizeOfOptionalHeader             0.883789
Machine                          0.900539
legitimate                       1.000000
Name: legitimate, dtype: float64
```

```
cor = df.corr()
corr_target = abs(cor['legitimate'])
```

```
corr_target.sort_values()
```

Εικόνα 3-6: συσχέτιση χαρακτηριστικών

Η corr() χρησιμοποιείται για την εύρεση της συσχέτισης κατά ζεύγη όλων των στηλών στο πλαίσιο δεδομένων. Τυχόν τιμές na αποκλείονται αυτόματα. Για οποιεσδήποτε στήλες μη αριθμητικού τύπου δεδομένων στο πλαίσιο δεδομένων αγνοείται.

Machine,SizeOfOptionalHeader,MajorSubsystemVersionκαι MajorOperatingSystemVersion συσχετίζονται έντονα θετικά με το χαρακτηριστικό γνώρισμα-στόχο legitimate.

Defining inputs and output as x and y respectively

In [17]:
```
X = final_df.drop(['legitimate'],axis=1)
y = final_df['legitimate'].values
```

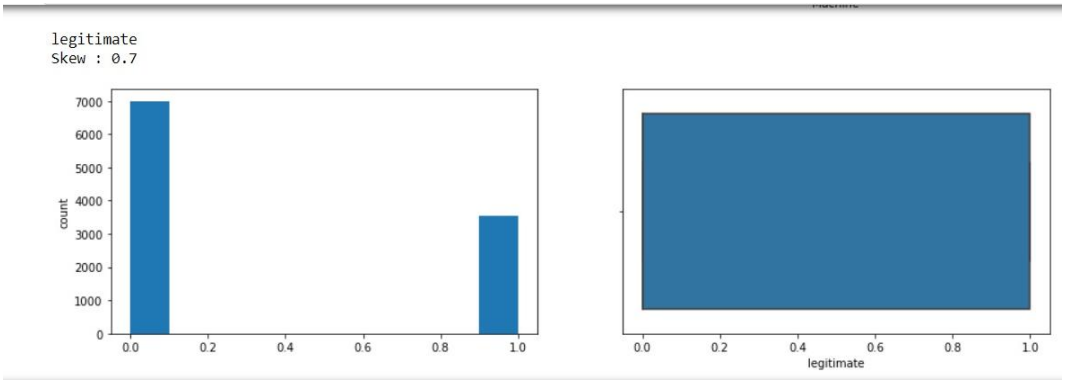
test train split of data

In [18]:
```
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.33, random_state=42)
```

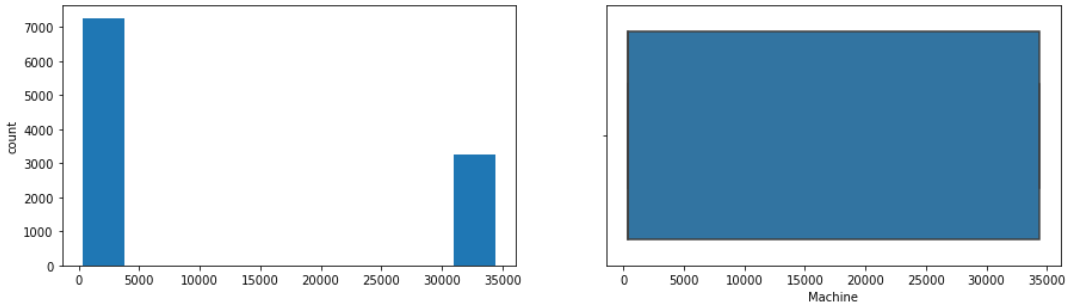
Εικόνα 3-7: test train split of data

3.1.4 διαγράμματα κατανομής και διαγράμματα πλαισίου αριθμητικών χαρακτηριστικών

```
In [16]: for col in final_df.columns:
             print(col)
             print('Skew :',round(final_df[col].skew(),2))
             plt.figure(figsize=(15,4))
             plt.subplot(1,2,1)
             final_df[col].hist(bins=10, grid=False)
             plt.ylabel('count')
             plt.subplot(1,2,2)
             sns.boxplot(x=final_df[col])
             plt.show()
```



Εικόνα 3-8: διαγράμματα κατανομής και διαγράμματα πλαισίου αριθμητικών χαρακτηριστικών legitimate skew 0.7



Εικόνα 3-9: διαγράμματα κατανομής και διαγράμματα πλαισίου αριθμητικών χαρακτηριστικών machine skew 0.82

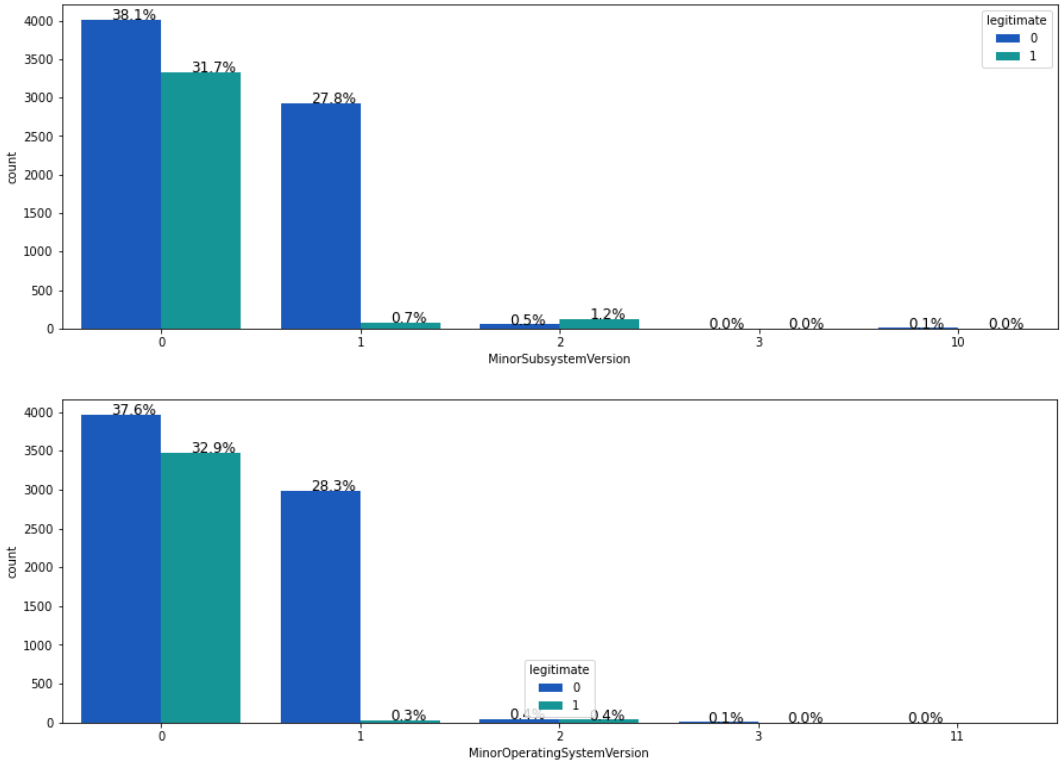
3.1.5 Δημιουργία barplots για το ποσοστό κάθε κατηγορίας.

```
In [8]: # Function to create barplots that indicate percentage for each category.
def perc_on_bar(plot, feature):
    '''
    plot
    feature: categorical feature
    the function won't work if a column is passed in hue parameter
    '''

    total = len(feature) # length of the column
    for p in ax.patches:
        percentage = '{:.1f}%'.format(100 * p.get_height()/total) # percentage of each class of the category
        x = p.get_x() + p.get_width() / 2 - 0.05 # width of the plot
        y = p.get_y() + p.get_height()           # height of the plot
        ax.annotate(percentage, (x, y), size = 12) # annotate the percentage

    plt.show()
```

```
In [9]: plt.figure(figsize=(15,5))
ax = sns.countplot(df["MinorSubsystemVersion"],palette='winter',hue=df['legitimate'])
perc_on_bar(ax,df["MinorSubsystemVersion"])
```

```
plt.figure(figsize=(15,5))
ax = sns.countplot(df["MinorOperatingSystemVersion"],palette='winter',hue=df['legitimate'])
perc_on_bar(ax,df["MinorOperatingSystemVersion"])
```

Εικόνα 3-10: barplots για το ποσοστό κάθε κατηγορίας



Εικόνα 3-11: barplots για το ποσοστό κάθε κατηγορίας

4 Κεφάλαιο 4ο: Αποτελέσματα

4.1 Random Forest Classifier

Για τα προβλήματα ταξινόμησης, εάν πρέπει να επιλέξει κανείς έναταξινομητή μεταξύ του συνόλου ταξινομητών που βασίζονται σε δέντρα χρησιμοποιήουμε το Random Forest για προβλημάτα ταξινόμησης [12]

Model 2:RandomForestClassifier

In [33]:

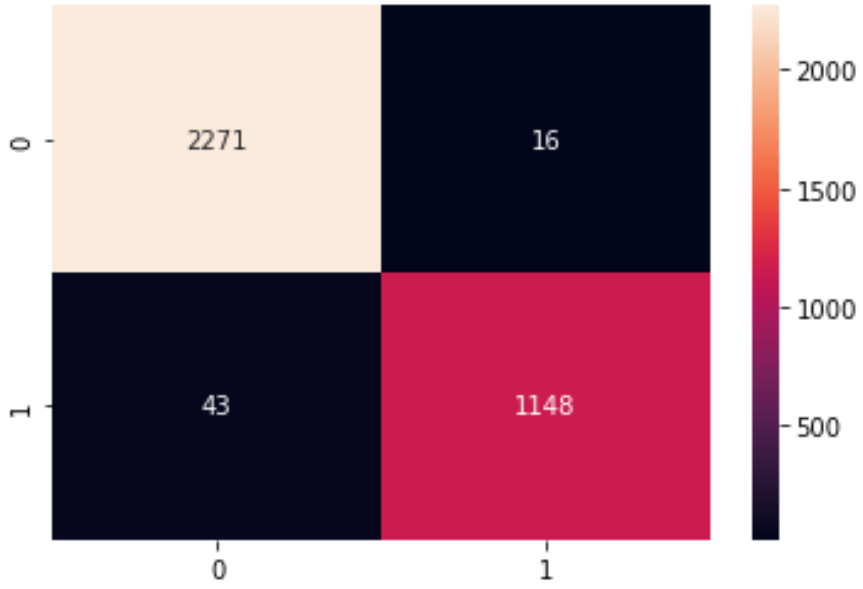
```
rf =RandomForestClassifier(criterion='gini', max_depth=8, max_features='sqrt',
                           n_estimators=200)
rf.fit(X_train, y_train)
y_predict_rf = rf.predict(X_test)
# confusion_matrix
cm = confusion_matrix(y_test, y_predict_rf)
sns.heatmap(cm, annot=True, fmt="d")
```

Εικόνα 4-1: Οι υπερ-παράμετροι που χρησιμοποιούνται

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

1. n_estimators: 200. Ο αριθμός των δέντρων του μοντέλου για επεξεργασία.
2. Max_depth: 8 το μέγιστο βάθος των δένδρων με ανοικτά φύλλα που περιέχουν λιγότερα από min_samples_split δείγματα.
3. criterion='gini': η μέθοδος εκτίμησης της ποιότητας του διαχωρισμού στους κόμβους
4. max_features='sqrt': ο αριθμός των χαρακτηριστικών για τον διαχωρισμό δεδομένων

Ο τρόπος κατανόησης του Max features είναι "Αριθμός χαρακτηριστικών που επιτρέπεται να γίνει ο καλύτερος διαχωρισμός κατά τη δημιουργία του δέντρου". Ο λόγος για τη χρήση αυτής της υπερ-παραμέτρου είναι ότι, αν επιτρέψετε όλα τα χαρακτηριστικά για κάθε διάσπαση, θα καταλήξετε σε ακριβώς τα ίδια δέντρα σε ολόκληρο το τυχαίο δάσος, κάτι που μπορεί να μην είναι χρήσιμο. Για να το ξεπεράσουμε αυτό, αφήνουμε το μοντέλο να επιλέξει τυχαία έναν σταθερό αριθμό χαρακτηριστικών, σε αυτή την περίπτωση, το no of features allowed = Square root of total no of features in your dataset.[2].



Εικόνα 4-2: Heatmap του ταξινομητή Random forest

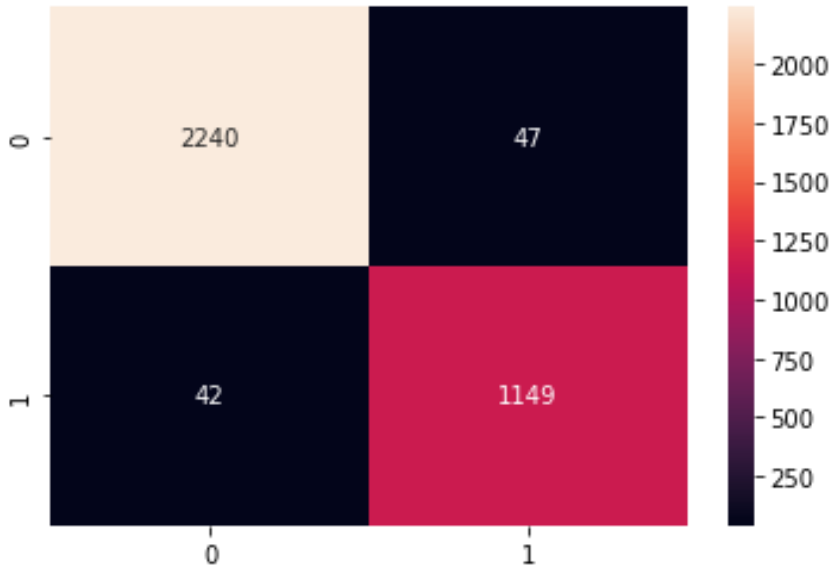
	precision	recall	f1-score	support
0	0.98	0.99	0.99	2287
1	0.99	0.96	0.97	1191
accuracy			0.98	3478
macro avg	0.98	0.98	0.98	3478
weighted avg	0.98	0.98	0.98	3478

True negative	False positive
2271	16
43	1148
False negative	True positive

Εικόνα 4-3: Σωστές προβλέψεις 3419 λάθος προβλέψεις 59

Υπολογίζουμε την Accuracy, precision, and recall του ταξινομητή randomforest που είναι 97%, 97% και 97% αντίστοιχα

4.1.1 DecisionTreeClassifier



Εικόνα 4-4: Heatmap του DecisionTreeClassifier

	precision	recall	f1-score	support
0	0.98	0.98	0.98	2287
1	0.96	0.96	0.96	1191
accuracy			0.97	3478
macro avg	0.97	0.97	0.97	3478
weighted avg	0.97	0.97	0.97	3478

True negative	False positive
2240	47
42	1149
False negative	True positive

Εικόνα 4-5: Σωστές προβλέψεις 3389 λάθος προβλέψεις 89

```
dt =DecisionTreeClassifier(ccp_alpha=0.001, criterion='entropy', max_depth=9,
                           max_features='auto', random_state=1024)
dt.fit(X_train, y_train)
y_predict_dt = dt.predict(X_test)
# confusion_matrix
cm = confusion_matrix(y_test, y_predict_dt)
sns.heatmap(cm, annot=True, fmt="d")
```

Οι υπερ-παράμετροι που χρησιμοποιούνται με την χρήση αυτής της μεθόδου:

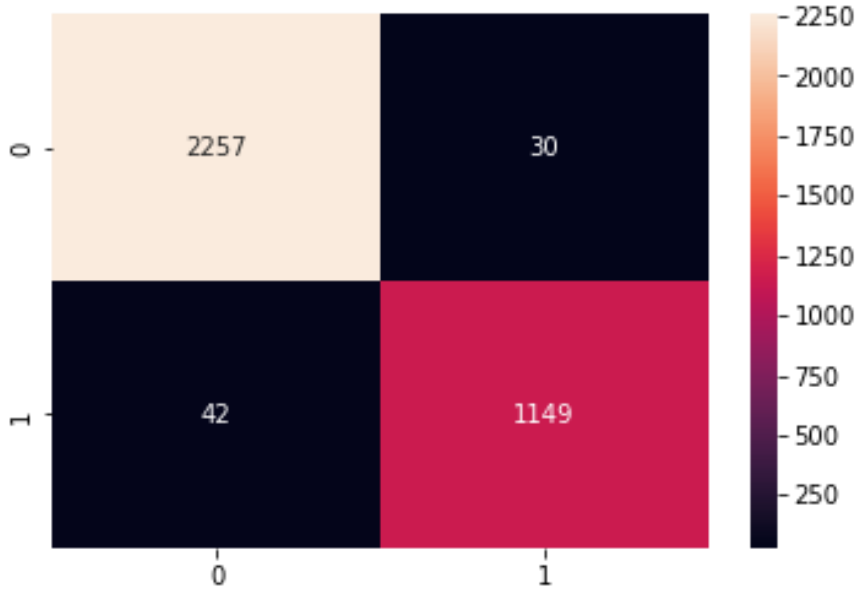
Criterion: η μέθοδος εκτίμησης της ποιότητας του διαχωρισμού στους κόμβους

ccp_alpha: αριθμός που δηλώνει την πολυπλοκότητα, τιμές με μεγαλύτερο κόστος θα έχουν ως αποτέλεσμα το κούρεμα των δένδρων. Μεγαλύτερες τιμές της ccp_alpha αυξάνουν τον αριθμό των κόμβων που κλαδεύονται.

max_features: ο αριθμός των χαρακτηριστικών για τον διαχωρισμό δεδομένων

Υπολογίζουμε την Accuracy, precision, and recall του ταξινομητή decisiontree που είναι 96%, 96% και 96% αντίστοιχα.

4.1.2 BaggingClassifier



Εικόνα 4-6: Heatmap του ταξινομητή bagging

	precision	recall	f1-score	support
0	0.98	0.99	0.98	2287
1	0.97	0.96	0.97	1191
accuracy			0.98	3478
macro avg	0.98	0.98	0.98	3478
weighted avg	0.98	0.98	0.98	3478

Εικόνα 4-7: BaggingClassifier

True negative	False positive
2257	30
42	1149
False negative	True positive

Εικόνα 4-8: Σωστές προβλέψεις 3406 λάθος προβλέψεις 72

Υπολογίζουμε την Accuracy, precision, and recall του ταξινομητή Bagging που είναι 98%, 98% και 98% αντίστοιχα

4.2 Σύνοψη και συμπεράσματα

Ολοκληρώνοντας την παρουσίαση των αποτελεσμάτων, έπεται η συζήτηση αναφορικά με αυτά. Συνοπτικά, η διαδικασία που ακολουθήθηκε αποτελούταν αρχικά από την επιλογή μοντέλων μηχανικής μάθησης. Τα ντετερμινιστικά μοντέλα BaggingClassifier, DecisionTreeClassifier και Random forest συντέλεσαν στην ανάλυση χαρακτηριστικών και στην εύρεση των καλύτερων τεχνικών εξισορρόπησης δεδομένων

Για την αξιολόγηση των επιδόσεων, πρέπει να αναλύσουμε το αληθώς θετικό ποσοστό (TPR), το αληθές αρνητικό ποσοστό (TNR), το ψευδώς θετικό ποσοστό (FPR), το ψευδώς αρνητικό ποσοστό (FNR) και την ακρίβεια. Αυτές οι μετρήσεις και η διαδικασία υπολογισμού τους περιγράφονται με σαφήνεια στον πίνακα 3. Ο πίνακας 2 παρουσιάζει τα αποτελέσματα με διαφορετικούς ταξινομητές.

Εδώ, μπορούμε να δούμε ότι η υψηλότερη ακρίβεια προσφέρθηκε από τον BaggingClassifier και Random Forest Classifier η οποία είναι 98%. Το BaggingClassifier παράγει ψευδώς θετικό με ποσοστό 0,47% που το καθιστά την προτεινόμενη προσέγγισή μας αξιόπιστη και κατανοητή. Ωστόσο, ένα σκορ ανάκλησης 98% υποδηλώνει ότι είμαστε σε θέση να ταξινομήσουμε το 98% του συνόλου των περιπτώσεων σωστά.Επιπλέον, υπολογίσαμε επίσης την ακρίβεια, την ανάκληση και σκορ f1 που παρουσιάζονται στο κεφάλαιο 4.

Πίνακας 4-1: Αποτέλεσμα της αληθούς και ψευδώς θετικής ταξινόμησης.

Αποτέλεσμα της αληθούς και ψευδώς θετικής ταξινόμησης.				
Algorithms	True positiverate	False positiverate	True negativerate	False negativerate
BaggingClassifier	**1149**	**47**	**2257**	**42**
DecisionTreeClassifier	**1149**	**47**	**2240**	**42**
Random Forest Classifier	**1148**	**16**	**2271**	**43**

Πίνακας 4-2: Μέτρο απόδοσης που χρησιμοποιείται στην προσέγγισή μας.

Τύπος	
TPR	TP/(TP+FN)
TNR	TN/(TN+FP)
FPR	FP/(FP+TN)
FNR	FN/(FN+TP)
Precision	TP/(TP+FP)
Recall	TP/(TP+FN)
F1-score	2 *(precision * recall)/ (precision +recall)
Accuracy	(TP+TN)/ (TP+TN+FN+FP)

Στην προσέγγιση μας χρησιμοποιήσαμε τον αλγόριθμο συσχέτισης Spearman. Το εύρος της συσχέτισης Spearman κυμαίνεται μεταξύ 0 και 1. Η τιμή συσχέτισης κοντά στο 1 σημαίνει ότι τα χαρακτηριστικά συμβάλλουν σε μεγάλο βαθμό, ενώ μια τιμή κοντά στο 0 σημαίνει ότι τα χαρακτηριστικά συμβάλλουν ελάχιστα.

Είναι συνηθισμένο στη μηχανική μάθηση ότι κάθε φορά που αυξάνεται η τιμή ανάκλησης,η ακρίβεια μειώνεται. Αυτό καλείται συμβιβασμός μεταξύ ακρίβειας και ανάκλησης και έχει αποφευχθεί σε μεγάλο βαθμό στην περίπτωση μας.

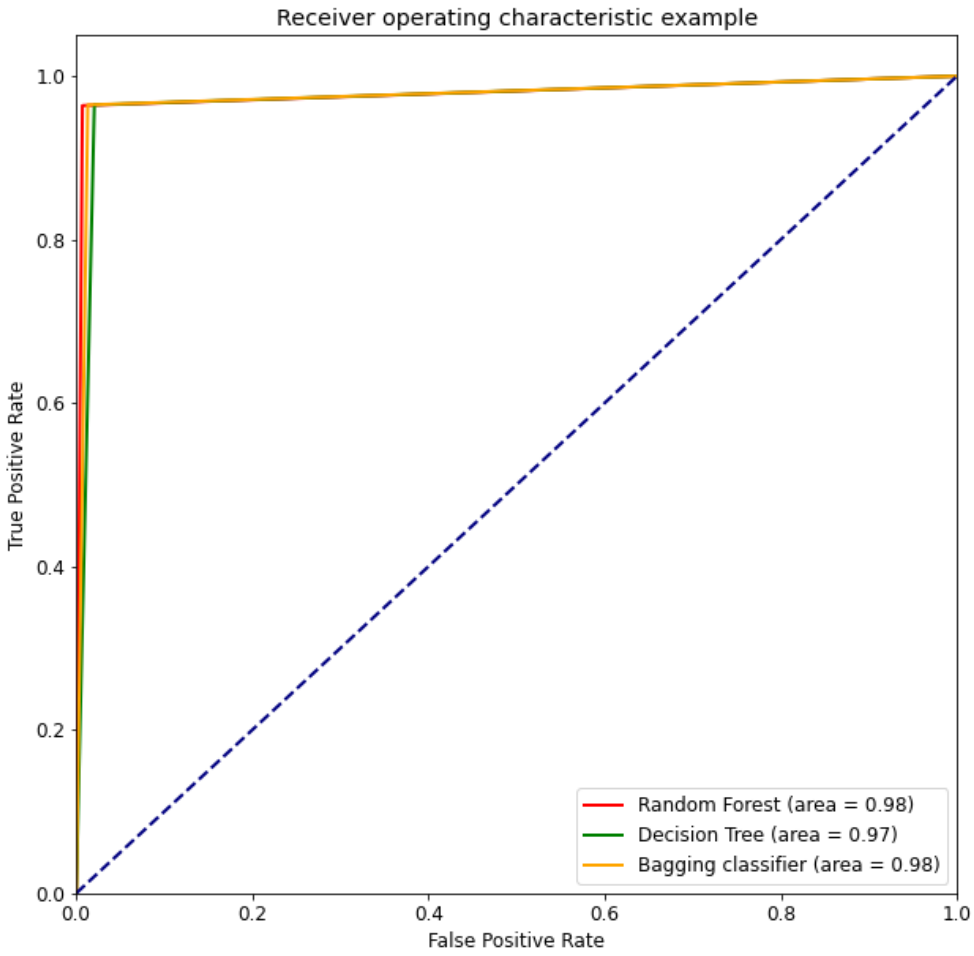
Πίνακας 4-3: Αποτελέσματα από διαφορετικούς ταξινομητές.

	Precision (%)	Confusion matrix	Recall (%)	F1 score (%)	Accuracy (%)
BaggingClassifier	98	[2257 30] [42 1149]	98	98	98
DecisionTreeClassifier	96	[2240 47] [42 1149]	96	96	96
Random Forest Classifier	97	[2271 16] [43 1148]	97	97	97

Η καμπύλη ROC (**Πίνακας 5)** απεικονίζει TPR έναντι του ποσοστού ψευδώς θετικών (FPR), το οποίο βοηθά στον υπολογισμό της ακρίβεια ενός αλγορίθμου [18].

Ο συντονισμός των υπερπαραμέτρων είναι μια διαδικασία επιλογής των σωστών παραμέτρων για έναν ταξινομητή - Η επιλογή των σωστών παραμέτρου θα ενισχύσει την ακρίβεια ενός ταξινομητή πράγμα που επιτευχθηκε στο έπακρο με την προσέγγιση μας. [19] .

Πίνακας 4-4: Καμπύλη ROC για όλους τους ταξινομητές



- Ο ταξινομητής bagging και ο ταξινομητής random forest λειτουργούν ως τα καλύτερα μοντέλα
- το Datasheet είναι inbalanced . δεν έγινε προσπάθεια εξισορροπήσεις γιατί τα αποτελέσματα ήταν πολύ καλά.
- Δεν υπάρχει overfitting σε κανένα από τα μοντέλα - overfitting σημαίνει μικρότερη ακρίβεια στην δοκιμή, αλλά όλα τα μοντέλα δίνουν καλή απόδοση στα δεδομένα της δοκιμής.
- Χρησιμοποιήσαμε μια αντιπαλο-κεντρική μεθοδολογία, για τον ακριβή εντοπισμό απειλών, σε όλη την αλυσίδα επιθέσεων - σκεφτήκαμε σαν αντίπαλος, εντοπίζοντας συμπεριφορές και δείκτες σε τελικά σημεία, αρχεία, χρήστες και

δίκτυα.Μια ολιστική περιγραφή της λειτουργίας μιας επίθεσης, ανεξάρτητα από το πού μπορεί να πραγματοποιηθεί η επίθεση.

- Το μοντέλο randomforest παρέχει πολύ καλά αποτελέσματα χωρίς καμία προ-επεξεργασία των δεδομένων.
- Επιτεύχθηκε με το μοντέλο μας ηβέλτιστη αξιοποίηση των υπολογιστικών πόρων, δηλαδή, αύξηση της ταχύτητας και μείωσητης μνήμης.
- Η κλιμάκωση δεν είναι απαραίτητη, το μοντέλο Random Forest είναι αναδρομικό μοντέλο διαμοιρασμού που εξαρτάται από των διαμοιρασμό των δεδομένων, επειδή λειτουργεί με διαχωρισμό των τιμών των χαρακτηριστικών και δεν κάνει υπολογισμούς σε αυτούς.

4.3 Καινοτομία

Εκτός από τις συσκευές με περιορισμένους πόρους, η δική μας η προσέγγισή μας μπορεί επίσης να ενσωματωθεί σε οποιεσδήποτε συσκευές που είναι πιο επιρρεπείς σε επιθέσεις malware [21] οπως συσκευές IoT μπορεί εύκολα να αναπτυχθεί σε περιβάλλον cloud.

4.4 Μελλοντικές Επεκτάσεις

Η ανάλυση malware και ο εντοπισμός τους με την βοήθεια της μηχανικής μάθησης μπορεί ναχρησιμοποιηθεί για τον εντοπισμό και την αποτροπή επιθέσεων σε δίκτυα ηλεκτρικής ενεργείας και δίκτυα μεταφοράς καύσιμων. [17] που μπορεί να προκαλέσουν σημαντικές αυξομειώσεις των τιμών και μεταβλητότητα στην παγκόσμια αγορά .

Βιβλιογραφία

10 .Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001

12. J. Ali, R. Khan, N. Ahmad, and I. Maqsood, “Random Forests and Decision Trees,”2012, Accessed: Oct. 27, 2021. [Online]. Available: www.IJCSI.org

14. V. Memos and K. Psannis, “A New Security Model based on Cloud Computing for Efficient Threats Detection”

16. Comparative Study of K-NN, Naive Bayes and Decision Tree Classification Techniques Sayali D. Jadhav , H. P. Channe

17. john W.Goodell ShaenCorbetbc Finance Research Letters : “Commodity Market Interactions With Energy-Firm Distress: Evidence From the Colonial Pipeline Ransomware Attack”

18. A novel approach for phishing URLs detection using lexical based machine- learning in a real-time environment. Brij B. Gupta, Krishna Yadav, Imran Razzak , Konstantinos Psannis, Arcangelo Castiglione, Xiaojun Chang f

19. V. A. Memos and K. E. Psannis, "AI-Powered Honeypots for Enhanced IoT Botnet Detection," 2020 3rd World Symposium on Communication Engineering (WSCE), 2020, pp. 64-68.

20 . Y.M.P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, “IoTPOT: A Novel Honeypot for Revealing Current IoT Threats”, Journal of Information Processing, Vol. 24, Issue 3, pp. 522-533, May 2016

[21] B.B Gupta, Aakanksha Tewari ¨Security, privacy and trust of different layers in Internet-of-Things (IoTs) framework¨ Future Generation ComputerSystems Volume 108, July 2020, Pages 909-920

21. Jochen Bauer; Michael Masuch; Jörg Franke : “An Analysis of Black Energy 3, Crashoverride, and Trisis, Three Malware Approaches Targeting Operational Technology Systems”

A.1 Ιστοσελίδες - webpages

15 Golang anti-vm framework for Red Team and Pentesters

Available https://github.com/p3tr0v/chacal

9. Decision Tree Classification in Python Tutorial

Available https://www.datacamp.com/tutorial/decision-tree-classification-python

13. Indicator of Compromise (IoC)

Available https://encyclopedia.kaspersky.com/glossary/indicator-of-compromise-ioc/

1. sklearn.ensemble.RandomForestClassifier Available https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

2. grid-search-result-max-features-sqrt-in-random-forest-how-to-understand Available https://datascience.stackexchange.com/questions/82560/grid-search-result-max-features-sqrt-in-random-forest-how-to-understand

3. RandomForests Available https://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm

4. How does 'max_samples' keyword for a Bagging classifier Available https://stackoverflow.com/questions/38772035/

5. sklearn.model Available https://scikitlearn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

6. split train test data using sklearn Available https://www.projectpro.io/recipes/split-train-test-data-using-sklearn-and-python

7. Available https://stackoverflow.com/questions/28064634/random-state-pseudo-random-number-in-scikit-learn

8. gini or entropy Available https://stats.stackexchange.com/questions/19639/which-is-a-better-cost-function-for-a-random-forest-tree-gini-index-or-entropy%20[%E2%86%A9]