ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων
Πληροφορικής

Σχεδιασμός, Υλοποίηση και Αξιολόγηση Κατάλληλων Συνελικτικών Νευρωνικών Δικτύων και Vision Transformer για την Ανίχνευση Λογισμικού

Διπλωματική Εργασία

του

ΣΑΚΑΛΗ ΑΓΓΕΛΟΥ

Επιβλέπουσα: Ιωάννα Ρουσσάκη
Αναπληρώτρια Καθηγήτρια ΕΜΠ

Αθήνα, Οκτώβριος 2025

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων
Πληροφορικής

Σχεδιασμός, Υλοποίηση και Αξιολόγηση Κατάλληλων Συνελικτικών Νευρωνικών Δικτύων και Vision Transformer για την Ανίχνευση Λογισμικού

Διπλωματική Εργασία

του

ΣΑΚΑΛΗ ΑΓΓΕΛΟΥ

Επιβλέπουσα: Ιωάννα Ρουσσάκη
Αναπληρώτρια Καθηγήτρια ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 27 Οκτωβρίου 2025.

Ιωάννα Ρουσσάκη	Μιλτιάδης Αναγνώστου	Συμεών Παπαβασιλείου
Αναπληρώτρια Καθηγήτρια ΕΜΠ	Ομότιμος Καθηγητής ΕΜΠ	Καθηγητής ΕΜΠ

Αθήνα, Οκτώβριος 2025

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων
Πληροφορικής

ΣΑΚΑΛΗ ΑΓΓΕΛΟΥ

27 Οκτωβρίου 2025

Αθήνα, Οκτώβριος 2025

Περίληψη

Η παρούσα διπλωματική εργασία συγκρίνει και εξετάζει την αποτελεσματικότητα δύο σύγχρονων αρχιτεκτονικών νευρωνικών δικτύων — των Συνελικτικών Νευρωνικών Δικτύων (CNN) και των (Vision Transformers, ViT) — στην ανίχνευση κακόβουλου λογισμικού. Η πειραματική αξιολόγηση πραγματοποιήθηκε σε τρία σύνολα δεδομένων διαφορετικού μεγέθους, με έμφαση στη σύγκριση της ακρίβειας, της ανάκλησης, του F1 score και της απώλειας. Τα αποτελέσματα αναδεικνύουν ότι τα προηγμένα μοντέλα ResNet-18,ViT-B/8 επιτυγχάνουν υψηλή απόδοση ακόμη και με περιορισμένα δεδομένα, ενώ η ενσωμάτωση τεχνικών αυτόματων κωδικοποιητών(autoencoders) για την μείωση των διαστάσεων των εικόνων βελτιώνει τις επιδόσεις των αρχιτεκτονικών. Επιπλέον, αξιολογείται η ανθεκτικότητα των μοντέλων απέναντι σε adversarial επιθέσεις, υποδεικνύοντας αδυναμίες αλλά και την ανάγκη για στρατηγικές ενίσχυσης της ασφάλειας. Τέλος, εφαρμόστηκε ensemble μοντέλο το οποίο συνδυάζει CNN και ViT αρχιτεκτονικές, επιτυγχάνοντας βελτιστοποιημένα αποτελέσματα. Η μελέτη αυτή έχει ως στόχο να συμβάλει στην κατανόηση των πλεονεκτημάτων, περιορισμών και προοπτικών των σύγχρονων μεθόδων μηχανικής μάθησης για εφαρμογές στην κυβερνοασφάλεια.

Λέξεις-κλειδιά: ανίχνευση κακόβουλου λογισμικού, μηχανική μάθηση, βαθιά νευρωνικά δίκτυα, ταξινόμηση αρχείων,Εκτελέσιμα (PE αρχεία) ,Εντοπισμός adversarial επιθέσεων

Abstract

This thesis compares and examines the effectiveness of two modern neural network architectures — Convolutional Neural Networks (CNN) and Vision Transformers (ViT) — in malware detection. The experimental evaluation was conducted on three datasets of varying sizes, focusing on the comparison of accuracy, recall, F1 score, and loss. The results demonstrate that advanced models such as ResNet-18 and ViT-B/8 achieve high performance even with limited data, while the incorporation of autoencoder techniques for image dimensionality reduction enhances the architectures' performance. Furthermore, the robustness of the models against adversarial attacks is evaluated, revealing vulnerabilities as well as the need for strategies to enhance security. Finally, an ensemble model combining CNN and ViT architectures was implemented, achieving optimized results. This study aims to contribute to the understanding of the advantages, limitations, and prospects of modern machine learning methods for applications in cybersecurity.

Keywords: malware detection, machine learning, deep neural networks, file classification, Portable Executable (PE) files, adversarial attack detection

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την καθηγήτρια Ιωάννα Ρουσσάκη για την ευκαιρία που μου έδωσε να εκπονήσω την διπλωματική εργασία στο συγκεκριμένο εργαστήριο. Επιπλέον, ευχαριστώ τον υποψήφιο διδάκτορα Μάριο Παρασκευόπουλο για τη βοήθεια και την καθοδήγηση κατά τη διάρκεια της διπλωματικής μου εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένειά μου και τους φίλους μου, ανθρώπους που με έχουν υποστηρίξει όλα αυτά τα χρόνια και μου έχουν μάθει να προσπαθώ πάντα περισσότερο.

Περιεχόμενα

Κατάλογος Πινάκων

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

1.1 Πρόλογος

Η ραγδαία ανάπτυξη του Διαδικτύου, σε συνδυασμό με την πρόοδο στον τομέα του λογισμικού και των ηλεκτρονικών συσκευών, έχει ενσωματώσει τη χρήση υπολογιστών και ψηφιακών συσκευών σε κάθε πτυχή της καθημερινότητας. Η εκτεταμένη χρήση έχει αναδείξει μια σοβαρή απειλή, την εξάπλωση του κακόβουλου λογισμικού. Σύμφωνα με στατιστικά δεδομένα από τη Microsoft και την Kaspersky , η σοβαρότητα αυτής της απειλής είναι αποδεδειγμένη, με πάνω από 437 εκατομμύρια επιθέσεις κακόβουλου λογισμικού που καταγράφηκαν παγκοσμίως από τον Νοέμβριο 2022 έως τον Οκτώβριο 2023. Αυτές οι επιθέσεις επισημαίνουν την ανάγκη για δυνατούς μηχανισμούς ανίχνευσης και πρόληψης.

Για να κατανοηθεί η έννοια του κακόβουλου λογισμικού, πρέπει να το ορίσουμε. Σύμφωνα με τη Microsoft [3] ως κακόβουλο λογισμικό ορίζεται το λογισμικό που περιγράφει τις κακόβουλες εφαρμογές ή τον κακόβουλο κώδικα που βλάπτουν ή διαταράσσουν την κανονική χρήση των τελικών συσκευών. Όταν μια συσκευή μολυνθεί από κακόβουλο λογισμικό, ενδέχεται να έχει μη εξουσιοδοτημένη πρόσβαση, να παραβιάσει δεδομένα ή να κλειδώσει τη συσκευή και να ζητήσει να πληρωθούν λύτρα. Το malware μπορεί να πάρει πολλές μορφές(όπως Trojans, virus, ransomware spyware κ.λ.π.) και μπορεί να προκαλέσει σημαντικά προβλήματα, όπως κλοπή δεδομένων, οικονομικές απώλειες και ζημιές στα συστήματα.

Σύμφωνα με τα στατιστικά που συλλέχθηκαν από το παγκόσμιο cloud service Kaspersky Security Network (KSN) [4] κατά την περίοδο Νοεμβρίου 2022 έως Οκτωβρίου 2023.

- Αποκλεισμός 437.414.681 επιθέσεων κακόβουλου λογισμικού που ξεκίνησαν από online πόρους παγκοσμίως. Εντοπισμός 106.357.530 μοναδικών κακόβουλων διευθύνσεων URL.
- Απαγόρευση εκτέλεσης κακόβουλου λογισμικού σε 112.922.612 περιπτώσεις με τη βοήθεια των εργαλείων Web Anti-Virus της Kaspersky.
- Αποτροπή επιθέσεων ransomware σε 193.662 μοναδικούς χρήστες
- Αποκλεισμός λογισμικού εξόρυξης κρυπτονομισμάτων (miners) σε 1.140.573 μοναδικούς χρήστες.

- Εμπόδιο σε κακόβουλο λογισμικό σχεδιασμένο να κλέβει χρήματα μέσω online πρόσβασης σε τραπεζικούς λογαριασμούς σε συσκευές 325.225 χρηστών.

Τα στατιστικά αυτά δείχνουν τον τεράστιο όγκο των κυβερνοεπιθέσεων που λαμβάνουν χώρα καθημερινά, σε πεδία του ψηφιακού κόσμου. Παράλληλα, αναδεικνύουν τη σημασία της ύπαρξης ισχυρών μηχανισμών ανίχνευσης και πρόληψης, προκειμένου να διασφαλιστεί η ψηφιακή ασφάλεια των χρηστών και των συστημάτων.

Οι μέθοδοι αντιμετώπισης των κακόβουλων αρχείων είναι πάρα πολλές, από απλές βάσεις δεδομένων που καταγράφουν ήδη γνωστά κακόβουλα αρχεία και τα αποκλείουν εάν εντοπιστούν ξανά στο δίκτυο, μέχρι εξελιγμένα συστήματα που αναλύουν τον κώδικα, τη συμπεριφορά, αλλά και οποιαδήποτε περίεργη σχέση με γνωστά δείγματα κακόβουλου λογισμικού.

Η ανίχνευση κακόβουλου λογισμικού γίνεται κρίσιμη στην ασφάλεια πληροφοριακών συστημάτων, καθώς οι μέθοδοι επίθεσης και αντιμετώπισης απαιτούν συνεχή βελτίωση και προσαρμογή στις νέες απειλές. Έτσι πολλές μελέτες επικεντρώνονται στην ανάπτυξη αποδοτικών και αποτελεσματικών μεθόδων αναγνώρισης και ταξινόμησης, όπως θα εξεταστεί και στη συγκεκριμένη εργασία.

1.2 Περιγραφή του Προβλήματος

Βασικό πρόβλημα στον τομέα είναι ο συνεχής ανταγωνισμός τόσο εξοπλισμών όσο και λογισμικού μεταξύ των ερευνητών ασφαλείας και των δημιουργών κακόβουλου λογισμικού. Όσο δημιουργούνται νέα εργαλεία ανίχνευσης από τη μία τόσο αναπτύσσονται οι τεχνικές απόκρυψης και παραλλαγής από την άλλη. Αυτός ο ανταγωνισμός είναι ασταμάτητος, δεδομένου ότι το βελτιωμένο ή τροποποιημένο κακόβουλο λογισμικό επανακυκλοφορεί συνέχεια και συχνά ξεπερνώντας τις υπάρχουσες μεθόδους ανίχνευσης.

Με την παρούσα εργασία, στοχεύουμε να ερευνήσουμε εργαλεία βασισμένα σε τεχνικές βαθιάς μηχανικής μάθησης, τα οποία θα βοηθούν τους ερευνητές ασφαλείας να διατηρούν το προβάδισμα για μεγαλύτερα χρονικά διαστήματα και να εντοπίσουν τα πλεονεκτήματα ανάμεσα σε διαφορετικές προσεγγίσεις εντοπισμού κακόβουλου λογισμικού. Τα εργαλεία αυτά θα επιχειρούν να αναγνωρίζουν κακόβουλα λογισμικά χωρίς να εξαρτώνται στενά από την προϋπάρχουσα γνώση για παρόμοια δείγματα, προσφέροντας μεγαλύτερη ανθεκτικότητα απέναντι σε νέες παραλλαγές.

Όπως θα δούμε και στο 3ο Κεφάλαιο, η έρευνα στο πεδίο αυτό είναι εκτενής και εξελίσσεται διαρκώς, αποκτώντας νέες μορφές. Καμία μέθοδος δεν μπορεί να θεωρηθεί απόλυτα ασφαλής ή οριστική, καθώς συνεχώς αναπτύσσονται νέοι τρόποι για να την παρακάμψουν.

Σε αυτήν την έρευνα λοιπόν, συγκρίνουμε δύο από τις πιο σύγχρονες τεχνολογίες βαθιάς μηχανικής μάθησης: τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNN) και τα Vision Transformers (ViT). Η ανάλυση επικεντρώνεται στην αποτελεσματικότητα των δύο μοντέλων για την ανίχνευση κακόβουλου λογισμικού, τόσο από άποψη ταχύτητας όσο και ακρίβειας.

1.3 Προκλήσεις και Προβλήματα που Αντιμετωπίζονται

Καθώς οι μέθοδοι ανίχνευσης γίνονται πιο εξελιγμένες, το ίδιο συμβαίνει και με τις τεχνικές που χρησιμοποιούν τα κακόβουλα λογισμικά για να τις παρακάμψουν. Στο πλαίσιο αυτό, προκύπτουν σημαντικές προκλήσεις που δυσχεραίνουν την ανάπτυξη αποδοτικών και πρακτικά εφαρμόσιμων λύσεων. Παρακάτω παρουσιάζονται τα βασικά προβλήματα και περιορισμοί που απασχολούν τον τομέα. [5]

1.3.1 Ανθεκτικότητα στις Πολυμορφικές Τεχνικές

Τα κακόβουλα λογισμικά συχνά χρησιμοποιούν τεχνικές όπως η *πολυμορφική* και *μεταμορφική* απόκρυψη, αλλάζοντας δυναμικά τη μορφή τους [6] για να παρακάμψουν τις υπάρχουσες μεθόδους ανίχνευσης. Οι παραδοσιακές μέθοδοι δυσκολεύονται να εντοπίσουν τέτοιες παραλλαγές, καθώς εξαρτώνται από χαρακτηριστικά που είναι εύκολο να αλλοιωθούν.

1.3.2 Μεγάλη Υπολογιστική Απαίτηση

Η εφαρμογή προηγμένων μοντέλων, όπως τα ViT, απαιτεί σημαντικούς υπολογιστικούς πόρους. Αυτό περιορίζει τη χρήση τους σε συστήματα με περιορισμένες δυνατότητες ή σε περιβάλλοντα πραγματικού χρόνου, όπου η απόδοση και η ταχύτητα είναι κρίσιμες.

1.3.3 Ελλιπής Γενίκευση

Τα υπάρχοντα μοντέλα συχνά αποτυγχάνουν να γενικεύσουν όταν έρχονται αντιμέτωπα με κακόβουλα λογισμικά που δεν έχουν συναντήσει κατά τη διαδικασία εκπαίδευσης. Αυτό μειώνει την αποτελεσματικότητά τους απέναντι σε νέες ή σπάνιες απειλές, καθιστώντας τα λιγότερο αξιόπιστα σε πραγματικές συνθήκες.

1.3.4 Ακρίβεια έναντι Ταχύτητας

Παρόλο που τα ViT προσφέρουν μεγαλύτερη ακρίβεια σε σχέση με άλλες μεθόδους, η αυξημένη υπολογιστική τους πολυπλοκότητα οδηγεί σε μειωμένη ταχύτητα. Αυτό μπορεί να αποτελέσει σημαντικό πρόβλημα για εφαρμογές όπου η άμεση ανίχνευση είναι ζωτικής σημασίας, όπως σε συστήματα κυβερνοασφάλειας πραγματικού χρόνου.

1.3.5 Ανάλυση και Ερμηνευσιμότητα

Τα ViT και τα CNN, λόγω της πολυπλοκότητας τους, λειτουργούν ως *μαύρα κουτιά*, καθιστώντας δύσκολη την κατανόηση του τρόπου λήψης αποφάσεων. Αυτό μπορεί να δημιουργήσει εμπόδια στην υιοθέτησή τους από μη εξειδικευμένους χρήστες, καθώς και να δυσχεράνει τον εντοπισμό ψευδών θετικών ή αρνητικών ανιχνεύσεων.

1.3.6 Ορισμός του Κακόβουλου Λογισμικού

Η κατηγοριοποίηση ενός προγράμματος ως κακόβουλου δεν είναι πάντα σαφής. Ακόμη και νόμιμες εφαρμογές, εάν διαθέτουν λειτουργίες που επιτρέπουν κακόβουλη χρήση,

μπορεί να στοχοποιηθούν λανθασμένα από τις μεθόδους ανίχνευσης, οδηγώντας σε ψευδείς συναγερμούς και δυνητικά λανθασμένες αποφάσεις.
Η κατανόηση και αντιμετώπιση αυτών των προκλήσεων είναι απαραίτητη για την ανάπτυξη πιο αποτελεσματικών και πρακτικών εργαλείων ανίχνευσης κακόβουλου λογισμικού.

1.4 Δομή της Εργασίας

Η δομή της παρούσας εργασίας ακολουθεί μια σταδιακή προσέγγιση για την εξέταση του θέματος, αρχίζοντας με την ανάπτυξη των βασικών εννοιών, την ανάλυση της βιβλιογραφίας, τη μεθοδολογία και τα πειράματα, καταλήγοντας στα συμπεράσματα της συγκριτικής μελέτης. Συγκεκριμένα:

Κεφάλαιο 2 Κακόβουλο Λογισμικό και Ανιχνευτές: Αναπτύσσονται οι θεμελιώδεις έννοιες γύρω από το κακόβουλο λογισμικό, τα είδη του και τη λειτουργία τους. Περιλαμβάνει επίσης ανάλυση των αρχείων PE (Portable Executable) και των ανιχνευτών κακόβουλου λογισμικού.

Κεφάλαιο 3 Συναφή Βιβλιογραφία: Εξετάζονται οι προηγούμενες έρευνες και μελέτες πάνω στη μηχανική μάθηση και τις τεχνικές που χρησιμοποιούνται για την ανίχνευση κακόβουλου λογισμικού. Αναφέρονται παραδοσιακές μέθοδοι, καθώς και σύγχρονες προσεγγίσεις με χρήση δυναμικής και στατικής ανάλυσης.

Κεφάλαιο 4 Προκλήσεις στην Ανίχνευση Κακόβουλου Λογισμικού: Αναλύονται οι προκλήσεις που αντιμετωπίζονται στον τομέα της ανίχνευσης κακόβουλου λογισμικού και παρουσιάζονται προτεινόμενες λύσεις.

Κεφάλαιο 5 Πηγή και Προετοιμασία Δεδομένων: Αναφέρεται η πηγή και η φύση των δεδομένων που χρησιμοποιούνται στα πειράματα, με λεπτομέρειες για τα στατιστικά τους, τις τεχνικές μείωσης διαστάσεων, και τους αλγορίθμους μηχανικής μάθησης που εφαρμόζονται.

Κεφάλαιο 6 Θεωρητικό Υπόβαθρο Πειραμάτων: Περιγράφονται τα μοντέλα και οι προσεγγίσεις βαθιάς μάθησης που χρησιμοποιούνται στα πειράματα.

Κεφάλαιο 7 Διεξαγωγή Πειραμάτων: Παρουσιάζονται τα αποτελέσματα των πειραμάτων με ανάλυση της απόδοσης των μοντέλων και των διαφόρων τεχνικών. Γίνεται σύγκριση των αποτελεσμάτων και αξιολόγηση της ακρίβειας, της ευστοχίας και άλλων δεικτών.

Κεφάλαιο 8 Συγκριτική Μελέτη και Συμπεράσματα: Στο τελικό μέρος, αναλύονται συγκριτικά τα αποτελέσματα των πειραμάτων και παρουσιάζονται τα τελικά συμπεράσματα της μελέτης.

Κεφάλαιο 9 Επίλογος: Στο κεφάλαιο αυτό γίνεται σύνοψη όλων των προηγούμενων, συνοδευόμενη από προτάσεις για μελλοντικές επεκτάσεις.

Η δομή αυτή επιτρέπει μια ολοκληρωμένη και μεθοδική προσέγγιση του θέματος, ξεκινώντας από την εισαγωγή και την ανάπτυξη των βασικών εννοιών, φτάνοντας σταδιακά στα πειράματα και τα συμπεράσματα, ενώ διασφαλίζει ότι καλύπτονται όλες οι κρίσιμες πτυχές του ζητήματος.

Κεφάλαιο 2

Κακόβουλο Λογισμικό και Ανιχνευτές

Το παρόν κεφάλαιο εισάγει τις βασικές έννοιες που σχετίζονται με το κακόβουλο λογισμικό και την ανίχνευσή του. Βλέπουμε τα κύρια είδη κακόβουλου λογισμικού, τη λειτουργία και τους μηχανισμούς εξάπλωσής του. Τέλος, περιγράφονται οι βασικές κατηγορίες ανιχνευτών κακόβουλου λογισμικού, θέτοντας το απαραίτητο υπόβαθρο για την κατανόηση των μεθοδολογιών που ακολουθούν στα επόμενα κεφάλαια.

2.1 Κακόβουλο Λογισμικό

2.1.1 Αναδρομή

Η εξέλιξη του κακόβουλου λογισμικού χωρίζεται σε φάσεις, με βάση το χρονικό διάστημα που εμφανίστηκε [7]. Η πρώτη είναι η φάση όπου το κακόβουλο λογισμικό, γεννήθηκε ως ιδέα με απλά προγράμματα που δεν είχαν κακόβουλη ακριβώς πρόθεση. Στη δεύτερη φάση της εξέλιξης του κακόβουλου λογισμικού εμφανίστηκαν τα πρώτα worms των Windows και των μηνυμάτων email ενώ στην τρίτη φάση ξεκίνησε η εξάπλωση με την ανάπτυξη του Διαδικτύου. Σε αυτή τη φάση, τα worms δικτύου ήταν οι πιο διαδεδομένες απειλές που εξαπλώθηκαν με την πρόοδο του Διαδικτύου. Τα rootkits και τα ransomware εισήχθησαν στην τέταρτη φάση της εξέλιξης και θεωρούνται από τις πιο επικίνδυνες απειλές. Η τελευταία φάση της εξέλιξης είναι ο τύπος κακόβουλου λογισμικού που αντιμετωπίζουμε σήμερα και περιλαμβάνει τη δημιουργία κακόβουλου λογισμικού ακόμη και από εταιρίες ή και χώρες με σκοπό απόκτησης απόρρητων δεδομένων.

Από την τελευταία φάση και έπειτα, παρατηρείται μια σαφής στροφή του κακόβουλου λογισμικού προς πιο εξελιγμένες, στοχευμένες και πολυμορφικές τεχνικές [8]. Οι επιθέσεις πλέον δεν περιορίζονται σε μεμονωμένες προσπάθειες διάδοσης, αλλά αποτελούν συχνά μέρος οργανωμένων επιχειρήσεων με οικονομικά, πολιτικά ή στρατηγικά κίνητρα.

Η χρήση τεχνικών όπως οι zero-day [9] επιθέσεις(κυβερνοεπιθέσεις που εκμεταλλεύονται ευπάθειες λογισμικού οι οποίες είναι άγνωστες στον κατασκευαστή ή στον ευρύτερο τεχνολογικό κόσμο τη στιγμή που πραγματοποιείται η επίθεση), οι τεχνικές αποφυγής εντοπισμού, καθώς και η υιοθέτηση εργαλείων τεχνητής νοημοσύνης για την αυτοματοποίηση επιθέσεων, καθιστούν το σύγχρονο τοπίο της κυβερνοασφάλειας ιδιαίτερα περίπλοκο και επικίνδυνο.

Η συγκεκριμένη πορεία υποδεικνύει την ανάγκη για πιο προσαρμοστικά και ευφυή συστήματα ανίχνευσης που να είναι σε θέση να αναγνωρίζουν τόσο γνωστές όσο και άγνωστες απειλές σε πραγματικό χρόνο, με ταυτόχρονη διατήρηση υψηλής απόδοσης και ερμηνευσιμότητας.

2.1.2 Λειτουργία

Η λειτουργία του κακόβουλου λογισμικού ποικίλλει ανάλογα με τον τύπο του, την τεχνολογία που χρησιμοποιεί, καθώς και τους στόχους των δημιουργών του [10]. Ωστόσο, τα περισσότερα κακόβουλα προγράμματα λειτουργούν μέσω συγκεκριμένων σταδίων που περιλαμβάνουν [11]:

Μόλυνση

Η αρχική φάση της λειτουργίας ενός κακόβουλου λογισμικού είναι η διείσδυση σε έναν υπολογιστή ή σύστημα. Αυτή η φάση μπορεί να επιτευχθεί μέσω πολλών μεθόδων, όπως:

- Ανοίγοντας μολυσμένα συνημμένα σε emails.
- Λήψη μολυσμένων αρχείων από μη ασφαλείς ιστοσελίδες.
- Αξιοποίηση ευπαθειών λογισμικού ή συστημάτων.
- Κοινωνική μηχανική (social engineering), όπως η εξαπάτηση χρηστών για την παροχή πρόσβασης.

Διασπορά

Αφού διεισδύσει, το κακόβουλο λογισμικό μπορεί να προσπαθήσει να εξαπλωθεί και σε άλλα συστήματα. Αυτό επιτυγχάνεται με τη χρήση δικτύων, κοινόχρηστων αρχείων ή ακόμα και αφαιρούμενων μέσων αποθήκευσης (USB drives). Τα worms είναι ιδιαίτερα γνωστά για την ικανότητά τους να εξαπλώνονται γρήγορα μέσω δικτύων.

Εκτέλεση Κακόβουλων Ενεργειών

Ανάλογα με τον τύπο του κακόβουλου λογισμικού, μπορεί να εκτελεστούν διαφορετικές ενέργειες, όπως:

- Κλοπή δεδομένων: Η αποστολή ευαίσθητων πληροφοριών (π.χ. κωδικών, προσωπικών δεδομένων) στον δημιουργό του κακόβουλου λογισμικού.
- Καταστροφή δεδομένων: Η διαγραφή ή αλλοίωση αρχείων και συστημάτων.
- Οικονομική εκμετάλλευση: Τα ransomware κρυπτογραφούν δεδομένα και απαιτούν λύτρα για την επαναφορά τους.
- Απομακρυσμένος έλεγχος: Δημιουργία backdoors για την απομακρυσμένη διαχείριση του συστήματος.
- Χρήση πόρων του συστήματος: Τα botnets χρησιμοποιούν υπολογιστικούς πόρους για κακόβουλους σκοπούς, όπως επιθέσεις DDoS.

Αποφυγή Ανίχνευσης

Το κακόβουλο λογισμικό εφαρμόζει συχνά τεχνικές για να αποφύγει την ανίχνευση από προγράμματα προστασίας, όπως:

- Πολυμορφικός και μεταμορφικός κώδικας: Αλλαγή του κώδικα σε κάθε εκτέλεση για αποφυγή ανίχνευσης.
- Απόκρυψη σε συστήματα: Χρήση rootkits για να κρύψει τη δραστηριότητά του.
- Εντοπισμός εικονικών περιβαλλόντων: Ανίχνευση sandboxes ή εικονικών μηχανών για την αποφυγή ανάλυσης.

Επιμονή

Το κακόβουλο λογισμικό προσπαθεί να εξασφαλίσει τη μακροπρόθεσμη παρουσία του στο σύστημα, ακόμα και αν γίνει επανεκκίνηση ή ενημέρωση. Χρησιμοποιεί τεχνικές όπως:

- Εγκατάσταση στον χώρο εκκίνησης (boot sector).
- Τροποποίηση αρχείων συστήματος.
- Δημιουργία αντιγράφων του εαυτού του σε διαφορετικά σημεία του συστήματος.

Τελικός Στόχος

Ο τελικός στόχος του κακόβουλου λογισμικού εξαρτάται από τον δημιουργό του. Μπορεί να περιλαμβάνει:

- Προσωπικό ή οικονομικό όφελος.
- Κατασκοπεία ή απόκτηση εμπιστευτικών πληροφοριών.
- Σαμποτάζ σε υποδομές ή υπηρεσίες.

Η πολυπλοκότητα της λειτουργίας του κακόβουλου λογισμικού καθιστά δύσκολη την πλήρη κατανόησή του και την αποτελεσματική ανίχνευσή του.

2.2 Είδη Κακόβουλου Λογισμικού

Η κατηγοριοποίηση και η ονοματοδότηση του κακόβουλου λογισμικού μπορεί να διαφέρει ανάλογα με τις προσεγγίσεις που ακολουθούνται από διάφορες πηγές. Ωστόσο, ορισμένες βασικές αρχές παραμένουν κοινές. Τα κακόβουλα προγράμματα ταξινομούνται συχνά με βάση τη λειτουργία τους και τον τρόπο διάδοσής τους. Η ταξινόμηση αυτή διευκολύνει την κατανόηση των διαφόρων απειλών και τη δημιουργία στρατηγικών για την αντιμετώπισή τους, καθώς πολλά κακόβουλα λογισμικά μπορούν να εξυπηρετούν πολλαπλούς σκοπούς και να κατηγοριοποιούνται αναλόγως. [12]

Τα κακόβουλα λογισμικά μπορούν να κατηγοριοποιηθούν με βάση δύο βασικές πτυχές: τον τρόπο διάδοσης και τη λειτουργία τους.

2.2.1 Τρόπος Διάδοσης

Η κατηγοριοποίηση των κακόβουλων λογισμικών με βάση τον τρόπο διάδοσής τους, αφορά τον τρόπο με τον οποίο μεταδίδονται και εξαπλώνονται σε διαφορετικά συστήματα.

Ιός Virus

Οι ιοί είναι κακόβουλα προγράμματα που έχουν πάρει το όνομά τους από τους βιολογικούς ιούς, καθώς μοιράζονται την ίδια λογική επίθεσης και διάδοσης. Συνήθως, ενσωματώνονται σε αρχεία ή προγράμματα και εκτελούνται όταν ο χρήστης ανοίξει το μολυσμένο αρχείο. Ο στόχος τους είναι να εξαπλωθούν σε άλλα αρχεία ή συστήματα.

Σκουλήκι Worm

Τα σκουλήκια είναι αυτοαναπαραγώμενα προγράμματα, παρόμοια με τους ιούς, αλλά διαφέρουν στον τρόπο διάδοσής τους. Αντί να απαιτούν την εκτέλεση από τον χρήστη, τα σκουλήκια διαδίδονται αυτόματα μέσω δικτύων, εκμεταλλευόμενα ευπάθειες των συστημάτων και μεταφέρονται από συσκευή σε συσκευή χωρίς ανθρώπινη παρέμβαση.

Δούρειος Ίππος Trojan

Τα κακόβουλα προγράμματα που χαρακτηρίζονται ως Δούρειοι Ίπποι δεν διαθέτουν μηχανισμούς αυτοαναπαραγωγής και διάδοσης. Αντ΄ αυτού, παρουσιάζονται ως ακίνδυνα προγράμματα ή αρχεία, παραπλανώντας τον χρήστη να τα εκτελέσει. Μόλις εγκατασταθούν, μπορούν να εκτελέσουν κακόβουλες λειτουργίες στο σύστημα του χρήστη.

2.2.2 Λειτουργία

Η κατηγοριοποίηση με βάση τη λειτουργία αφορά το πώς το κακόβουλο λογισμικό εκτελεί τις επιθέσεις του και τι είδους ζημιά προκαλεί στο σύστημα του θύματος.

Ransomware

Το Ransomware [13]είναι ένας τύπος κακόβουλου λογισμικού που κλειδώνει ή κρυπτογραφεί τα αρχεία του θύματος και απαιτεί την καταβολή λύτρων για την επαναφορά της πρόσβασης. Συνήθως διαδίδεται μέσω μολυσμένων συνημμένων σε email ή μέσω κακόβουλων ιστοσελίδων. Οι επιθέσεις Ransomware μπορούν να προκαλέσουν σημαντική οικονομική ζημία και απώλεια δεδομένων.

Spyware

Το Spyware είναι κακόβουλο λογισμικό που σχεδιάζεται για να κατασκοπεύει τις ενέργειες των χρηστών χωρίς τη γνώση τους. Συλλέγει πληροφορίες όπως κωδικούς πρόσβασης, ιστορικό περιήγησης, και προσωπικά δεδομένα, τα οποία στη συνέχεια μεταφέρονται στους επιτιθέμενους για κακόβουλη χρήση.

Adware

Το Adware εμφανίζει ανεπιθύμητες διαφημίσεις στον χρήστη, συχνά με τη μορφή αναδυόμενων παραθύρων ή αλλαγών στις ρυθμίσεις του προγράμματος περιήγησης. Αν και δεν είναι πάντα καταστροφικό, μπορεί να επιβραδύνει τη συσκευή και να συλλέγει δεδομένα για το χρήστη για διαφημιστικούς σκοπούς.

Rootkit

Τα Rootkit είναι κακόβουλα προγράμματα που επιτρέπουν στους εισβολείς να αποκτήσουν απομακρυσμένη πρόσβαση ή πλήρη έλεγχο του συστήματος του θύματος, παρακάμπτοντας τα συστήματα ασφαλείας. Συνήθως είναι πολύ δύσκολο να εντοπιστούν και επιτρέπουν στον επιτιθέμενο να διαχειρίζεται το σύστημα χωρίς την επίγνωση του χρήστη.

Keylogger

Οι Keylogger είναι κακόβουλα προγράμματα που καταγράφουν τις πληκτρολογήσεις του χρήστη. Αυτό επιτρέπει στους επιτιθέμενους να συλλέξουν ευαίσθητες πληροφορίες, όπως κωδικούς πρόσβασης, αριθμούς πιστωτικών καρτών και άλλα προσωπικά δεδομένα, για κακόβουλη χρήση.

Botnets

Τα Botnets αποτελούνται από δίκτυα μολυσμένων συσκευών, οι οποίες ελέγχονται εξ αποστάσεως από έναν επιτιθέμενο. Οι συσκευές αυτές, γνωστές ως 'bots' ή 'zombie", χρησιμοποιούνται συνήθως για την εκτέλεση κακόβουλων δραστηριοτήτων όπως επιθέσεις DDoS , αποστολή ανεπιθύμητων μηνυμάτων , ή ακόμη και εξόρυξη κρυπτονομισμάτων, χωρίς τη γνώση του ιδιοκτήτη τους.

Αυτές οι κατηγορίες καλύπτουν ένα μεγάλο φάσμα από τα πιο κοινά κακόβουλα λογισμικά που μπορεί να συναντήσει κάποιος στον κυβερνοχώρο.

2.3 Portable Executables

Κατά την περίοδο συγγραφής της παρούσας εργασίας, το λειτουργικό σύστημα Windows κατέχει το μεγαλύτερο μερίδιο της αγοράς λειτουργικών συστημάτων για υπολογιστές [14]. Συνεπώς, είναι λογικό να εξετάσουμε τις πιο κοινές μορφές κακόβουλου λογισμικού που στοχεύουν το δημοφιλέστερο λειτουργικό σύστημα, δηλαδή τα αρχεία τύπου *Portable Executable* (PE).

Η μορφή των αρχείων PE χρησιμοποιείται για εκτελέσιμα αρχεία σε πλατφόρμες Windows 32 και 64 bit και είναι ιδιαίτερα διαδεδομένη σε κακόβουλες επιθέσεις. Το πιο γνωστό παράδειγμα είναι τα αρχεία με κατάληξη `.EXE`, τα οποία αντιπροσωπεύουν εκτελέσιμα προγράμματα. Ωστόσο, τα αρχεία PE δεν περιορίζονται μόνο σε αυτήν την επέκταση. Άλλες γνωστές καταλήξεις που χρησιμοποιούν τη δομή PE περιλαμβάνουν τα [15]:

- `.DLL` (Dynamic Link Library): Αρχεία βιβλιοθηκών που περιέχουν κώδικα και δεδομένα, τα οποία μπορούν να χρησιμοποιηθούν από διαφορετικά προγράμματα.

- `.SYS`: Αρχεία συστήματος που χρησιμοποιούνται από τα Windows για τη διαχείριση συσκευών και πόρων του υπολογιστή.

- `.SCR`: Αρχεία προφυλακτήρα οθόνης (screensaver), τα οποία μπορούν να χρησιμοποιηθούν κακόβουλα.

Η εκτενής χρήση της μορφής PE στα Windows τα καθιστά ευάλωτα σε επιθέσεις, καθώς κακόβουλοι παράγοντες εκμεταλλεύονται την ευκολία τροποποίησης και διανομής τέτοιων αρχείων για να εισαγάγουν κακόβουλο κώδικα που εκτελείται χωρίς τη γνώση του χρήστη [16].

2.3.1 Δομή

Η μορφή PE [15] είναι μία από τις πιο σημαντικές και σύνθετες δομές αρχείων που υπάρχουν στα Windows. Πρόκειται για ένα δυαδικό αρχείο που περιέχει τις πληροφορίες που χρειάζεται το λειτουργικό σύστημα για να εκτελέσει έναν πρόγραμμα. Αποτελείται από μια σειρά επικεφαλίδων και τμημάτων τα οποία παρέχουν τις λεπτομέρειες για τον τρόπο με τον οποίο θα φορτωθεί και θα εκτελεστεί το πρόγραμμα στη μνήμη του συστήματος. Οι κύριες επικεφαλίδες περιλαμβάνουν την DOS Header, PE Header , την Section Table και τα διαφορετικά τμήματα που περιέχουν τον πραγματικό κώδικα και δεδομένα του προγράμματος.

Η αρχική επικεφαλίδα είναι η DOS Header , η οποία περιέχει ένα μικρό κομμάτι κώδικα συμβατό με MS-DOS , που συνήθως εμφανίζει ένα μήνυμα αν το αρχείο εκτελεστεί σε παλαιότερο σύστημα. Αμέσως μετά βρίσκεται η PE Header , η οποία περιλαμβάνει σημαντικές πληροφορίες για τη φόρτωση του αρχείου, όπως το είδος του μηχανήματος (32 or 64 bit) , τον αριθμό των sections και το σημείο εκκίνησης του κώδικα entry point .

Τα τμήματα περιλαμβάνουν δεδομένα όπως:

- `.text`: Περιέχει τον εκτελέσιμο κώδικα του προγράμματος.

- `.data`: Περιέχει τα στατικά δεδομένα, όπως μεταβλητές και σταθερές.

- `.rdata`: Περιέχει μόνο δεδομένα ανάγνωσης, όπως πίνακες μεταδεδομένων ή σταθερές συμβολοσειρές.

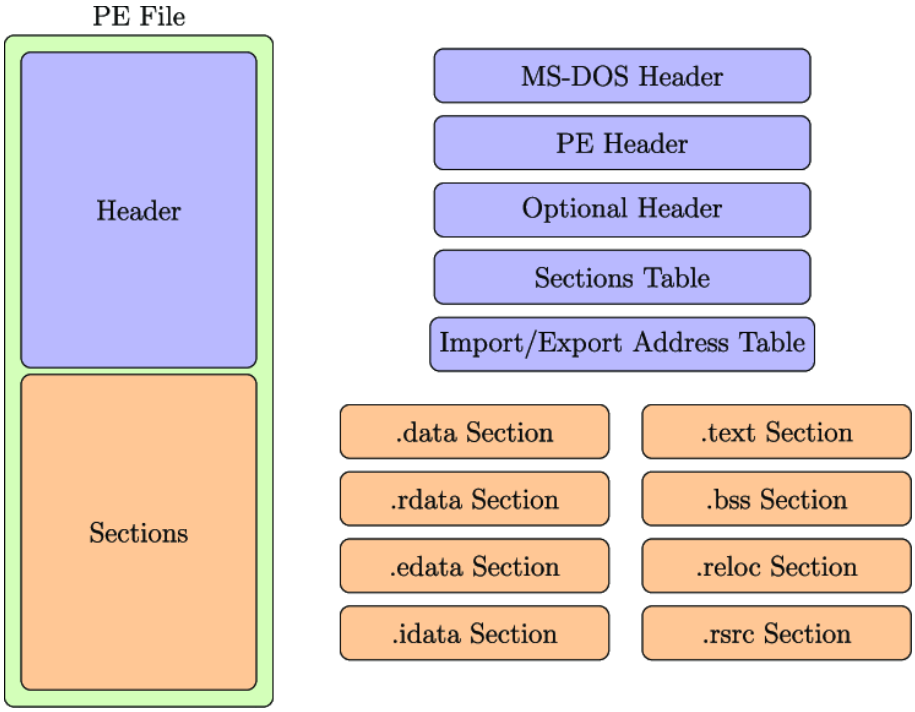
- `.rsrc`: Αποθηκεύει πόρους του προγράμματος, όπως εικονίδια, διαλόγους και άλλα στοιχεία του περιβάλλοντος χρήστη.

2.3.2 Επιμέρους Στοιχεία

Τα PE αρχεία έχουν επιμέρους στοιχεία που εξυπηρετούν διάφορους σκοπούς στη διαδικασία εκτέλεσης και φόρτωσης ενός προγράμματος. Ενδεικτικά, τα κύρια αυτά στοιχεία είναι:

- **Επικεφαλίδες (Headers)**: Περιλαμβάνουν τις πληροφορίες για τον τύπο του αρχείου, τις εντολές που θα εκτελεστούν και το μέγεθος των τμημάτων. Εξασφαλίζουν ότι το πρόγραμμα θα φορτωθεί σωστά στη μνήμη.

- **Sections (Τμήματα)**: Τα τμήματα ενός PE αρχείου καθορίζουν τα διαφορετικά μέρη του κώδικα, τα δεδομένα και τους πόρους που θα χρησιμοποιηθούν κατά την εκτέλεση. Κάθε τμήμα έχει τη δική του λειτουργία και καθορίζεται από τον τύπο του προγράμματος.
- **Import and Export Tables**: Οι πίνακες εισαγωγής και εξαγωγής καθορίζουν τις συναρτήσεις που εισάγονται από άλλες βιβλιοθήκες ή εξάγονται σε άλλα προγράμματα. Είναι κρίσιμο στοιχείο για τη δυναμική σύνδεση βιβλιοθηκών (*dynamic linking*).
- **Address Table**: Ο πίνακας διευθύνσεων βοηθά το λειτουργικό σύστημα να αντιστοιχίσει τον κώδικα στη σωστή θέση της μνήμης όταν το αρχείο PE φορτώνεται.



Σχήμα 2.1: Δομή αρχείου PE (Portable Executable). Πηγή: ResearchGate [1]

2.3.3 Ασφάλεια και Κίνδυνοι

Η ευελιξία και η ευρεία χρήση της μορφής PE, παρόλο που είναι κρίσιμη για το λειτουργικό σύστημα Windows, την καθιστά επίσης στόχο για επιθέσεις. Ο τρόπος με τον οποίο είναι σχεδιασμένη η αρχιτεκτονική των αρχείων PE επιτρέπει τη χρήση εργαλείων όπως *packers* and *obfuscators*, τα οποία μπορούν να κρύψουν ή να μεταμορφώσουν τον κακόβουλο κώδικα, καθιστώντας τον πιο δύσκολα ανιχνεύσιμο από συστήματα antivirus.

Επιπλέον, πολλές επιθέσεις κακόβουλου λογισμικού επικεντρώνονται στην παραβίαση της ακεραιότητας των PE αρχείων μέσω τεχνικών όπως η ενσωμάτωση κακόβουλου κώδικα σε υφιστάμενα εκτελέσιμα αρχεία (file injection). Σημαντική επίσης είναι η τεχνική *DLL injection* , όπου κακόβουλες βιβλιοθήκες DLL φορτώνονται εν αγνοία του χρήστη από μη εξουσιοδοτημένα προγράμματα.

Συνεπώς, η αντιμετώπιση τέτοιων απειλών απαιτεί διαρκή ενημέρωση των συστημάτων ασφαλείας και συνεχή εκπαίδευση των χρηστών, καθώς οι μέθοδοι επίθεσης εξελίσσονται διαρκώς. Προληπτικά μέτρα περιλαμβάνουν τη χρήση σύγχρονων antivirus, την εφαρμογή τεχνικών sandboxing και την αποφυγή εκτέλεσης άγνωστων αρχείων PE . [17]

2.4 Ανιχνευτές

Οι μέθοδοι ανίχνευσης δεν αποτελούν συγκεκριμένες μεθοδολογίες και τεχνικές καθώς κατά περιόδους εμφανίζονται νέες και εξειδικευμένες προσεγγίσεις. Παρ΄όλα αυτά αν κάνουμε μία αναδρομή υπάρχει ένας ορθολογικός πυρήνας που χαρακτηρίζει κάθε περίοδο και μας βοηθά να κάνουμε μία κατηγοριοποίηση [18] [19] [20]

2.4.1 1ης Γενιάς - Signature based

Οι ανιχνευτές πρώτης γενιάς χαρακτηρίζονται από απλές, αλλά βασικές μεθόδους λειτουργίας που στηρίζονται στον εντοπισμό συγκεκριμένων, προκαθορισμένων ακολουθιών bytes μέσα σε κακόβουλα αρχεία. Οι τεχνικές αυτές είναι γνωστές ως 'τεχνικές βασισμένες στις υπογραφές' (signature-based), καθώς απαιτείται η προηγούμενη γνώση των προκαθορισμένων bytes που ταυτοποιούνται στα κακόβουλα λογισμικά.Αυτές οι υπογραφές δημιουργούνται είτε στατικά είτε δυναμικά και αποθηκεύονται για μελλοντική σύγκριση με τα αρχεία που αναλύονται. Οι υπογραφές αυτές μπορεί να περιλαμβάνουν ακολουθίες API calls, opcodes, byte-code series, και μετρήσεις εντροπίας. Η αποτελεσματικότητά τους εξαρτάται άμεσα από την έγκαιρη και τακτική ενημέρωση της βάσης δεδομένων, η οποία περιέχει τις υποψήφιες ακολουθίες bytes προς ταυτοποίηση.

Βασικές Τεχνικές

- **Ανίχνευση Αλφαριθμητικών Χαρακτήρων (String Scanning):** Αυτή η απλή μέθοδος βασίζεται στην ανίχνευση και ταυτοποίηση συγκεκριμένων bytes που εμφανίζονται συχνά σε κακόβουλα λογισμικά αλλά όχι σε νόμιμα. Ωστόσο, είναι εύκολο να παρακαμφθεί με μικρές τροποποιήσεις στον πηγαίο κώδικα του κακόβουλου λογισμικού. Έτσι, σπάνια χρησιμοποιείται μόνη της.

- **Wildcards:** Επεκτείνοντας την ανίχνευση αλφαριθμητικών, η χρήση Wildcards καθιστά την ταυτοποίηση πιο ευέλικτη. Δεν απαιτείται να ταυτιστεί ολόκληρη η ακολουθία, αλλά μόνο ορισμένα σημαντικά μέρη της, προσφέροντας μεγαλύτερη ανθεκτικότητα στις τεχνικές παράκαμψης.

- **Αναντιστοιχίες (Mismatches):** Επιτρέποντας συγκεκριμένο αριθμό αναντιστοιχιών (π.χ., N αναντιστοιχίες) σε μια αναζήτηση, διευρύνεται το εύρος των μοτίβων που μπορούν να ταυτοποιηθούν.

- **Γενική Ανίχνευση (General Detection):** Συνδυάζοντας μπαλαντέρ και αναντιστοιχίες, δημιουργούνται γενικά μοτίβα που εντοπίζουν κακόβουλα λογισμικά που έχουν σχεδιαστεί για να παρακάμπτουν τις μεμονωμένες τεχνικές.

Ενισχυτικές Τεχνικές για Βελτίωση Απόδοσης

Δεδομένου ότι η πλήρης σάρωση ενός εκτελέσιμου αρχείου είναι αργή, χρησιμοποιούνται συμπληρωματικές τεχνικές για την επιτάχυνση της διαδικασίας:

- **Κατακερματισμός (Hashing):** Με την κατακερματισμένη σάρωση συγκεκριμένων σημείων του αρχείου και τη σύγκριση με γνωστές hash υπογραφές, επιτυγχάνεται ταχύτερη και πιο αξιόπιστη ανίχνευση.
- **Σελιδοδείκτες (Bookmarks):** Οι ακολουθίες bytes συχνά βρίσκονται σε συγκεκριμένες θέσεις (offsets) μέσα στο αρχείο. Με τη χρήση σελιδοδεικτών, η αναζήτηση περιορίζεται σε αυτά τα σημεία, αυξάνοντας την ταχύτητα και την ακρίβεια.
- **Σάρωση Αρχής και Τέλους (Top-and-Tail Scanning):** Αυτή η τεχνική επικεντρώνεται μόνο στις αρχικές και τελικές περιοχές του αρχείου, που συχνά περιέχουν κακόβουλο φορτίο.
- **Σάρωση Σημείου Εισαγωγής και Σταθερού Σημείου (Entry-Point and Fixed-Point Scanning):** Εστιάζοντας στο σημείο εισαγωγής του εκτελέσιμου κώδικα ή σε ένα σταθερό σημείο, η διαδικασία ανίχνευσης γίνεται πιο αποδοτική. Αν, για παράδειγμα, το σταθερό σημείο είναι η "main" σε διεύθυνση X, η σάρωση ξεκινά από X+M bytes.

Συμπεράσματα

Οι ανιχνευτές πρώτης γενιάς, παρά την απλότητά τους, παραμένουν ένα θεμελιώδες εργαλείο για την ανίχνευση κακόβουλου λογισμικού. Ωστόσο, η αποτελεσματικότητά τους εξαρτάται από τη συχνή ενημέρωση των βάσεων δεδομένων και τη συνδυαστική χρήση ενισχυτικών τεχνικών για να αντιμετωπίζονται οι προκλήσεις της ταχύτητας και της ακρίβειας.

2.4.2 2ης Γενιάς Heuristic-Based

Οι ανιχνευτές βασισμένη στην ευριστική ανίχνευση αποτελούν σημαντική εξέλιξη στην ανίχνευση κακόβουλου λογισμικού, ενσωματώνοντας προηγμένες τεχνικές που ξεπερνούν τους περιορισμούς της πρώτης γενιάς. Αυτές οι τεχνικές στοχεύουν στην πιο αποτελεσματική και ακριβή ανίχνευση, προσαρμοζόμενες στις εξελισσόμενες απειλές.

Τεχνικές Ανίχνευσης

- **Έξυπνη Σάρωση (Smart Scanning):** Αντί για απλή ταυτοποίηση ακολουθιών bytes, η έξυπνη σάρωση αγνοεί μη κρίσιμα στοιχεία όπως οι εντολές NOP. Αυτή η προσέγγιση εντοπίζει παραλλαγές κακόβουλων λογισμικών που διαφορετικά θα διέφευγαν από την ανίχνευση.
- **Ανίχνευση Πυρήνα (Skeleton Detection):** Η τεχνική αυτή αναλύει γραμμή προς γραμμή τον κώδικα ενός αρχείου μακροεντολών (macro) και εξάγει τον «πυρήνα» του. Ο πυρήνας περιλαμβάνει τις κρίσιμες εντολές, επιτρέποντας την ανίχνευση κακόβουλων λογισμικών που βασίζονται σε μακροεντολές.

- **Σχεδόν Ακριβής Ταυτοποίηση (Nearly Exact Identification):** Συνδυάζει πολλαπλές ακολουθίες bytes ή checksums για την ταυτοποίηση. Αν ταυτοποιηθεί περισσότερα από ένα χαρακτηριστικά, είναι σχεδόν βέβαιο ότι το λογισμικό είναι κακόβουλο.

- **Ακριβής Ταυτοποίηση (Exact Identification):** Η τεχνική αυτή χρησιμοποιεί πλήρη ανάλυση και πολλαπλά checksums, προσφέροντας τη μέγιστη ακρίβεια στην ταυτοποίηση απειλών. Αν και απαιτεί περισσότερο χρόνο, διασφαλίζει την πλήρη αναγνώριση της απειλής.

Αλγοριθμικές Μέθοδοι Ανίχνευσης

- **Προσομοίωση Κώδικα (Code Simulation):** Αναλύει τον κώδικα κατά την εκτέλεσή του σε εικονικές μηχανές (virtual machines), εξασφαλίζοντας ότι το σύστημα παραμένει ασφαλές. Ωστόσο, η μέθοδος είναι απαιτητική σε πόρους.

- **Ευρετική Ανάλυση (Heuristic Analysis):** Συνδυάζει στοιχεία από στατική και δυναμική ανάλυση, εντοπίζοντας κακόβουλο λογισμικό βάσει ύποπτων χαρακτηριστικών. Παρότι είναι αποτελεσματική για νέες απειλές, συχνά παράγει ψευδοθετικά αποτελέσματα.

- **Sandboxing:** Με την προσομοίωση ολόκληρου του συστήματος, τα κακόβουλα λογισμικά αναλύονται πλήρως. Παρόλο που είναι πιο ακριβής, αυτή η μέθοδος απαιτεί υψηλούς πόρους και ενδέχεται να επηρεαστεί από εξελιγμένα κακόβουλα λογισμικά που αναγνωρίζουν το περιβάλλον προσομοίωσης.

Συμπεράσματα

Οι ανιχνευτές δεύτερης γενιάς προσφέρουν σημαντικά πλεονεκτήματα, ενσωματώνοντας προηγμένες τεχνικές όπως έξυπνη σάρωση, ανίχνευση πυρήνα και δυναμική ανάλυση. Ωστόσο, η αποτελεσματικότητά τους εξαρτάται από την ενημέρωση της βάσης δεδομένων υπογραφών και τις απαιτήσεις σε πόρους. Παρότι παρουσιάζουν περιορισμούς, αποτελούν αναπόσπαστο μέρος της άμυνας απέναντι στις εξελισσόμενες απειλές.

2.5 Προκλήσεις και Περιορισμοί

Παρά τα σημαντικά πλεονεκτήματα των ανιχνευτών δεύτερης γενιάς, υπάρχουν ορισμένες προκλήσεις και περιορισμοί που επηρεάζουν την αποτελεσματικότητά τους:

- **Υπολογιστικοί Πόροι:** Οι τεχνικές όπως το sandboxing και η προσομοίωση κώδικα απαιτούν σημαντικούς πόρους, γεγονός που μπορεί να επιβραδύνει την απόδοση του συστήματος.

- **Ψευδοθετικά Αποτελέσματα:** Η ευρετική ανάλυση συχνά ταυτοποιεί αβλαβές λογισμικό ως κακόβουλο, απαιτώντας περαιτέρω επιβεβαίωση και αυξάνοντας τον φόρτο εργασίας για τους αναλυτές ασφαλείας.

- **Παράκαμψη από Σύγχρονες Απειλές:** Ορισμένα εξελιγμένα malware χρησιμοποιούν τεχνικές όπως anti-sandboxing και κρυπτογράφηση για να αποφύγουν την ανίχνευση, περιορίζοντας την αποτελεσματικότητα των μεθόδων δεύτερης γενιάς.
- **Ανάγκη Συχνής Ενημέρωσης:** Αν και οι ευρετικές μέθοδοι μειώνουν την εξάρτηση από τις βάσεις δεδομένων υπογραφών, οι ανιχνευτές εξακολουθούν να απαιτούν τακτικές ενημερώσεις για να αντιμετωπίζουν νέες μορφές απειλών.

Παρά τα παραπάνω, οι ανιχνευτές δεύτερης γενιάς παραμένουν ένας απαραίτητος μηχανισμός ασφάλειας, ειδικά όταν χρησιμοποιούνται σε συνδυασμό με πιο σύγχρονες τεχνικές ανίχνευσης βαθιάς μηχανικής μάθησης.

Κεφάλαιο 3

Συναφή Βιβλιογραφία

Η παρούσα ενότητα παρουσιάζει την επιστημονική βιβλιογραφία που σχετίζεται με το αντικείμενο της εργασίας. Στόχος μας είναι η ανάδειξη των βασικών θεωρητικών και εμπειρικών προσεγγίσεων που έχουν ενασχοληθεί με τα ερωτήματα που επιχειρεί να καλύψει η παρούσα εργασία.

Ιδιαίτερη έμφαση δίνεται στην αναγκαιότητα αξιοποίησης μεθόδων μηχανικής μάθησης, στις επιμέρους τεχνικές που έχουν χρησιμοποιηθεί στη βιβλιογραφία, καθώς και στην παρουσίαση χαρακτηριστικών μελετών που έχουν υλοποιηθεί στο συγκεκριμένο πεδίο. Η ανάλυση στοχεύει στην κατανόηση του πλαισίου μέσα στο οποίο έχει κινηθεί η έρευνα, πάνω στο οποίο εντάσσεται η παρούσα εργασία και στην τεκμηρίωση της σημασίας της συμβολής της.

3.1 Μηχανική Μάθηση

Η αυξανόμενη πολυπλοκότητα και συχνότητα εμφάνισης κακόβουλου λογισμικού καθιστά απαραίτητες τις προηγμένες μεθόδους διάγνωσης πέρα από τις παραδοσιακές μεθόδους. Οι παραδοσιακές μέθοδοι, όπως η ανίχνευση βάσει υπογραφών αν και ασφαλώς είναι πολύ γρήγορες για ανίχνευση γνωστών κακόβουλων λογισμικών, έχουν αποδειχθεί ανεπαρκείς για την αναγνώριση νέων και εξελιγμένων που χρησιμοποιούν τεχνικές απόκρυψης και εξαπάτησης. Έμφαση λοιπόν δόθηκε στη διάγνωση βάσει πιο περίπλοκων μηχανισμών, χρησιμοποιώντας τεχνικές μηχανικής μάθησης για να αναλύσουν τις ενέργειες του λογισμικού σε ελεγχόμενα περιβάλλοντα και να τα κατατάξουν ως κακόβουλα ή μη. Κύριες προσεγγίσεις περιλαμβάνουν την εξαγωγή και ανάλυση χαρακτηριστικών από κλήσεις API, τη μελέτη ιδιοτήτων εκτελέσιμων αρχείων (όπως headers, sections, opcodes), καθώς και τη χρήση στατιστικών και μηχανικών μοντέλων μάθησης με στόχο τη βελτίωση της ακρίβειας ανίχνευσης κακόβουλου λογισμικού και τη μείωση των ψευδώς θετικών αποτελεσμάτων.

Η μηχανική μάθηση ξεκίνησε να εφαρμόζεται στον τομέα της κυβερνοασφάλειας και συγκεκριμένα στον εντοπισμό κακόβουλου λογισμικού τη δεκαετία του 2000, ως απάντηση στην αύξηση της πολυπλοκότητας και της συχνότητας των απειλών. Οι παραδοσιακές μέθοδοι διάγνωσης, αποδείχθηκαν ανεπαρκείς στην αναγνώριση νέων και εξελιγμένων malware .

Αρχικά, οι ερευνητές επικεντρώθηκαν στην ανάλυση των χαρακτηριστικών του κακόβουλου λογισμικού, χρησιμοποιώντας στατιστικές μεθόδους και απλά μοντέλα μηχανικής μάθησης. Με την πάροδο του χρόνου, οι μέθοδοι αυτοί εξελίχθηκαν, και πιο περίπλοκες προσεγγίσεις, όπως οι αλγόριθμοι υποστήριξης διανυσμάτων και τα νευ-

ρωνικά δίκτυα, άρχισαν να χρησιμοποιούνται για την αναγνώριση και την ταξινόμηση κακόβουλων και μη κακόβουλων αρχείων. [21] [22]

3.1.1 Κατηγορίες Μηχανικής Μάθησης

Η μηχανική μάθηση διακρίνεται σε τρεις βασικές κατηγορίες [23]: επιβλεπόμενη, μη επιβλεπόμενη και ενισχυτική μάθηση. Κάθε προσέγγιση έχει διαφορετικό τρόπο λειτουργίας και εύρος εφαρμογής στον τομέα της κυβερνοασφάλειας και ειδικότερα στην ανίχνευση κακόβουλου λογισμικού.

Επιβλεπόμενη Μάθηση (Supervised Learning) Η επιβλεπόμενη μάθηση βασίζεται σε σύνολα δεδομένων τα οποία είναι πλήρως ταξινομημένα, ώστε κάθε δείγμα συνοδεύεται από την αντίστοιχη κατηγορία του (π.χ. κακόβουλο ή μη κακόβουλο). Τα μοντέλα εκπαιδεύονται ώστε να αναγνωρίζουν πρότυπα και να προβλέπουν την κατηγορία νέων, αταξινόμητων δεδομένων. Είναι από τις πιο διαδεδομένες μεθόδους στην ανίχνευση malware, καθώς επιτρέπει υψηλή ακρίβεια . Συνήθως εφαρμόζονται αλγόριθμοι όπως Random Forests, SVM, CNN και Transformer τα τελευταία χρόνια.

Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning) Η μη επιβλεπόμενη μάθηση εφαρμόζεται σε δεδομένα χωρίς ετικέτες. Στόχος είναι η εύρεση δομών, ομάδων ή ανωμαλιών μέσα στο σύνολο των δεδομένων, όπως η ομαδοποίηση (clustering) ή η ανίχνευση ακραίων τιμών (outlier detection). Παράδειγμα εφαρμογής είναι ο εντοπισμός νέων, μη καταγεγραμμένων απειλών που διαφέρουν από τα συνηθισμένα πρότυπα. Συχνά χρησιμοποιούνται μέθοδοι όπως K-means, DBSCAN, PCA και autoencoders.

Ενισχυτική Μάθηση (Reinforcement Learning) Η ενισχυτική μάθηση βασίζεται στην έννοια του πράκτορα (agent) που αλληλεπιδρά με ένα περιβάλλον και λαμβάνει ενισχύσεις (rewards) ανάλογα με τις ενέργειές του. Αν και λιγότερο διαδεδομένη στον τομέα του malware detection, έχει αρχίσει να εφαρμόζεται για προσαρμοζόμενα συστήματα ασφάλειας και αντιμετώπισης εξελισσόμενων επιθέσεων.

3.2 Βασικές κατηγορίες ανίχνευσης

Το μεγάλο πρόβλημα είναι η συνεχής μεταβολή και δημιουργία νέων κακόβουλων αρχείων και η δημιουργία νέων τεχνικών εξάπλωσης. Αυτή η δυναμική φύση των απειλών καθιστά την ανίχνευση και την αντιμετώπισή τους ιδιαίτερα δύσκολη, απαιτώντας ενημερωμένα συστήματα ασφαλείας, εκπαίδευση χρηστών και τακτικά αντίγραφα ασφαλείας για την προστασία των δεδομένων και των συστημάτων. Οι μέθοδοι αντιμετώπισης διακρίνονται σε δυναμικές, στατικές και υβριδικές [24] [25] [26]

Η στατική ανίχνευση κακόβουλου λογισμικού αναφέρεται στην ανάλυση του κακόβουλου λογισμικού χωρίς την εκτέλεσή του. Χρησιμοποιεί τεχνικές όπως η ανάλυση του κώδικα, η διερεύνηση των εκτελέσιμων αρχείων (όπως τα PE αρχεία), η σύγκριση με γνωστές υπογραφές κακόβουλου λογισμικού. Είναι γρήγορη και αποδοτική, αλλά μπορεί να είναι αναποτελεσματική ενάντια σε προηγμένες τεχνικές απόκρυψης, όπως η χρήση πολυμορφικού ή μεταμορφικού κώδικα, όπου το κακόβουλο λογισμικό αλλάζει την εμφάνισή του.

Η δυναμική ανίχνευση κακόβουλου λογισμικού, από την άλλη, περιλαμβάνει την παρακολούθηση της συμπεριφοράς του λογισμικού κατά την εκτέλεσή του σε ένα απομονωμένο περιβάλλον, όπως ένα sandbox. Καταγράφει τις ενέργειες που εκτελεί το λογισμικό, όπως οι αλλαγές σε αρχεία, οι προσβάσεις στο δίκτυο ή οι διαδικασίες που δημιουργεί. Αυτή η προσέγγιση είναι πιο αποτελεσματική στην αναγνώριση κακόβουλης συμπεριφοράς, ακόμα και όταν το κακόβουλο λογισμικό προσπαθεί να αποφύγει την ανίχνευση μέσω στατικής ανάλυσης.

Και οι δύο μέθοδοι είναι απαραίτητες για την πλήρη ανάλυση κακόβουλου λογισμικού, καθώς η καθεμία συμπληρώνει τις αδυναμίες της άλλης οπότε σαφώς η υβριδική ανάλυση που έχει χρήση δυναμικών και στατικών προσεγγίσεων σε συνδυασμό είναι ανώτερη αλλά αρκετά πιο απαιτητική σε πόρους.

3.2.1 Στατική Προσέγγιση

Η στατική ανίχνευση κακόβουλου λογισμικού αναφέρεται στην ανάλυση του λογισμικού χωρίς την εκτέλεσή του. Οι βασικές τεχνικές περιλαμβάνουν:

- **Ανάλυση Κώδικα:** Ανάγνωση και διερεύνηση του πηγαίου ή του αποσυμπιεσμένου δυαδικού κώδικα για τον εντοπισμό ύποπτων λειτουργιών ή ακολουθιών bytes.
- **Διερεύνηση Εκτελέσιμων Αρχείων:** Ειδική ανάλυση δομών, όπως των αρχείων Portable Executable (PE), για την ταυτοποίηση χαρακτηριστικών που σχετίζονται με κακόβουλα λογισμικά.
- **Σύγκριση Υπογραφών:** Χρήση βάσεων δεδομένων με γνωστές υπογραφές κακόβουλων λογισμικών για την ταυτοποίηση γνωστών απειλών.
- **Αποσυμπίεση:** Μετατροπή του εκτελέσιμου κώδικα σε επίπεδο υψηλότερης γλώσσας, ώστε να κατανοηθεί καλύτερα η λειτουργία του.

Παρόλο που η στατική ανάλυση είναι ταχύτερη και απαιτεί λιγότερους πόρους, αντιμετωπίζει δυσκολίες όταν το κακόβουλο λογισμικό χρησιμοποιεί τεχνικές απόκρυψης, όπως:

- **Πολυμορφικός Κώδικας:** Το κακόβουλο λογισμικό αλλάζει την υπογραφή του κατά τη μεταφορά ή την εκτέλεση, ώστε να αποφεύγει την ανίχνευση.
- **Μεταμορφικός Κώδικας:** Το λογισμικό μεταβάλλει τη δομή του σε κάθε εκτέλεση, καθιστώντας την ανίχνευση από τα στατικά εργαλεία δυσκολότερη.

3.2.2 Δυναμική Προσέγγιση

Η δυναμική ανίχνευση επικεντρώνεται στη συμπεριφορά του κακόβουλου λογισμικού κατά την εκτέλεσή του. Οι βασικές τεχνικές περιλαμβάνουν:

- **Χρήση Sandbox:** Το κακόβουλο λογισμικό εκτελείται σε απομονωμένο περιβάλλον, όπου καταγράφονται οι ενέργειές του, όπως:
  - Αλλαγές σε αρχεία ή το μητρώο του συστήματος (registry).

  - Προσβάσεις στο δίκτυο για πιθανή επικοινωνία με κακόβουλους διακομιστές.
  - Δημιουργία ή χειρισμός νέων διαδικασιών.

- **Καταγραφή Συμπεριφοράς:** Καταγράφεται η συμπεριφορά του λογισμικού και συγκρίνεται με γνωστά μοτίβα κακόβουλων ενεργειών.
- **Παρακολούθηση Κλήσεων API:** Ανάλυση των κλήσεων σε λειτουργίες του λειτουργικού συστήματος για ύποπτη δραστηριότητα.

Παρόλο που η δυναμική ανάλυση είναι πιο αποτελεσματική απέναντι σε τεχνικές απόκρυψης, παρουσιάζει περιορισμούς, όπως:

- Υψηλές απαιτήσεις σε πόρους, λόγω της ανάγκης προσομοίωσης ολοκληρωμένων συστημάτων.
- Δυνατότητα κακόβουλου λογισμικού να ανιχνεύει το προσομοιωμένο περιβάλλον και να αποκρύπτει τη δραστηριότητά του.

3.2.3 Υβριδική Προσέγγιση/Συνδυασμός Δυναμικής και Στατικής Ανάλυσης

Η χρήση δυναμικών και στατικών προσεγγίσεων σε συνδυασμό παρέχει τα εξής πλεονεκτήματα:

- Συμπληρώνει τις αδυναμίες κάθε μεθόδου: Ο συνδυασμός στατικών και δυναμικών τεχνικών επιτρέπει την πλήρη ανίχνευση, καλύπτοντας τα κενά που αφήνουν οι μεμονωμένες μέθοδοι.
- Αναγνώριση γνωστών και νέων κακόβουλων λογισμικών: Ενισχύει την ικανότητα ανίχνευσης νέων, άγνωστων απειλών και μεταβλητών εκδόσεων κακόβουλου λογισμικού.
- Βελτιστοποίηση της ανάλυσης μέσω διασταύρωσης πληροφοριών από διαφορετικές πηγές.

Η υβριδική προσέγγιση είναι σαφώς η πιο ολοκληρωμένη και ανθεκτική στις εξελιγμένες τεχνικές απόκρυψης, ωστόσο απαιτεί περισσότερους υπολογιστικούς πόρους και είναι πιο περίπλοκη στην εφαρμογή.

3.3 Μελέτες με μεθόδους Μηχανικής Μάθησης

Η έρευνα στον τομέα της ανίχνευσης κακόβουλου λογισμικού είναι εξαιρετικά εκτενής, ακόμη και μόνο για το λειτουργικό σύστημα Windows που ασχολούμαστε στην παρούσα εργασία. Οι διάφορες προσεγγίσεις, τα δεδομένα και τα αποτελέσματα που έχουν, προσφέρουν πλήθος λύσεων και προσεγγίσεων, οι οποίες διαρκώς εξελίσσονται και προσαρμόζονται στις νέες απειλές. Στην παρούσα ενότητα, θα προσπαθήσουμε να αναφέρουμε και να αναλύσουμε κάποιες από τις βασικότερες προσεγγίσεις ανίχνευσης

κακόβουλου λογισμικού στα πρώτα βήματα της όλης διαδικασίας για να δούμε αργότερα πιο πρόσφατες και σύγχρονες προσεγγίσεις. Η αναφορά αυτή λοιπόν θα ξεκινήσει με τις πιο άπλές΄ μεθόδους που βασίζονται στη μηχανική μάθηση και θα προχωρήσει σταδιακά σε σύγχρονα νευρωνικά δίκτυα, τα οποία προσφέρουν εξελιγμένες και πιο ακριβείς λύσεις για την ανίχνευση σύνθετων και εξελιγμένων επιθέσεων.

Όπως φαίνεται και στην μελέτη των ***Firdausi et al.***(2010) [21] η συνεχής αύξηση του κακόβουλου λογισμικού που εκμεταλλεύεται το Διαδίκτυο αποτελούσε σοβαρή απειλή. Ο χειροκίνητος έλεγχος δεν θεωρούνταν πλέον αποτελεσματικός σε σχέση με τον υψηλό ρυθμό εξάπλωσης του κακόβουλου λογισμικού. Έτσι, η αυτοματοποιημένη ανίχνευση κακόβουλου λογισμικού που βασίζεται στη συμπεριφορά, χρησιμοποιώντας τεχνικές μηχανικής μάθησης, θεωρούνταν πλέον μια ουσιαστική λύση. Για τα δεδομένα συλλέχθηκαν 220 unique malware(Indonesian malware) samples. Τα δείγματα του συνόλου δεδομένων για τις μη κακόβουλες περιπτώσεις συλλέχθηκαν από αρχεία συστήματος που βρίσκονται στον κατάλογο "System32" μιας καθαρής εγκατάστασης των Windows XP. Συνολικά αποκτήθηκαν 250 unique benign software samples.

Η εξαγωγή των δεδομένων έγινε μέσω δυναμικής ανάλυσης, δηλαδή παρακολούθησης της συμπεριφοράς των δειγμάτων κατά την εκτέλεσή τους. Συγκεκριμένα, κάθε δείγμα, κακόβουλο ή μη, υποβλήθηκε στην δωρεάν διαδικτυακή υπηρεσία Anubis, η οποία εκτελεί τα προγράμματα και παράγει αναφορές σε μορφή XML. Αυτές οι αναφορές κατεγράφησαν και στη συνέχεια έγινε επεξεργασία, η οποία περιλάμβανε την επιλογή χαρακτηριστικών, τη δημιουργία λεξικού όρων και τη μετατροπή κάθε αναφοράς σε δυαδικό ή σταθμισμένο διανυσματικό μοντέλο, το οποίο αποθηκεύτηκε σε αρχεία ARFF Το ARFF είναι ακρωνύμιο του Attribute-Relation File Format. Πρόκειται για ένα αρχείο κειμένου τύπου ASCII που περιγράφει μια λίστα από εγγραφές οι οποίες μοιράζονται ένα κοινό σύνολο χαρακτηριστικών .Τα αρχεία αυτά χρησιμοποιήθηκαν για εκπαίδευση και ταξινόμηση μέσω:

- k-Nearest Neighbors (kNN),
- Naïve Bayes,
- Δέντρο Απόφασης J48 Decision Tree,
- Support Vector Machine (SVM), και
- Πολυεπίπεδο Νευρωνικό Δίκτυο Multilayer Perceptron Neural Network (MlP).

Με βάση την ανάλυση των δοκιμών και τα πειραματικά αποτελέσματα και των πέντε ταξινομητών, η συνολικά καλύτερη απόδοση επιτεύχθηκε με το δέντρο απόφασης 96.8%.

Στην εργασία στον ***Wang et al.***(2011) [27] ιδιαίτερο ενδιαφέρον έχει ότι πραγματεύεται την ανίχνευση κακόβουλου λογισμικού που χρησιμοποιεί τεχνικές απόκρυψης πετυχαίνοντας ποσοστό μέχρι και 94.54% True Possitives Στη μελέτη αυτή παρουσιάζεται ένα σύστημα ανίχνευσης εκτελέσιμων αρχείων (Packing Detection Framework - PDF) με στόχο την ανίχνευση τόσο γνωστών όσο και άγνωστων packers μέσω στατικής ανάλυσης των PE αρχείων. Για το σύνολο δεδομένων χρησιμοποιήθηκαν 3,784 packed και 1,056 non-packed εκτελέσιμα, από τα οποία τα περισσότερα malicious δείγματα ελήφθησαν από το VX Heaven και τα καλόβουλα δείγματα(benign) από καθαρές εγκαταστάσεις Windows και από το PChome Online. Η επεξεργασία περιλάμβανε εξαγωγή χαρακτηριστικών από τις κεφαλίδες PE αρχείων, DLLs/APIs, μέτρα εντροπίας,

αριθμούς sections και άλλα ευριστικά γνωρίσματα. Για ταξινόμηση χρησιμοποιήθηκαν Support Vector Machines (SVMs) με RBF kernel.

Οι ***Schultz et al.***(2001) [28] χρησιμοποίησαν Naïve Bayes, J48 decision tree, k-nearest neighbors (k-NN), συγκρίνοντάς τα με μεθόδους εύρεσης με βάση υπογραφές σε σύνολο δεδομένων 4,266(3,265 κακόβουλα και 1,001 καλόβουλα) προγράμματα με συλλογή των δεδομένων από διάφορες πηγές και κατηφοριοποίησης τους με χρήση προγράμματος antivirus επιτυγχάνοντας μέχρι και 97.11% ακρίβεια. Ένα υποσύνολο των δεδομένων περιλάμβανε εκτελέσιμα σε μορφή Portable Executable (PE). Η εξαγωγή χαρακτηριστικών έγινε με στατική ανάλυση μέσω της βιβλιοθήκης libBFD, από την οποία συλλέχθηκαν πληροφορίες όπως οι DLLs που χρησιμοποιούνται, οι κλήσεις συναρτήσεων εντός αυτών, και ο αριθμός διαφορετικών συναρτήσεων ανά DLL. Προκειμένου να καλυφθεί ολόκληρο το σύνολο δεδομένων, χρησιμοποιήθηκαν επίσης οι GNU strings για εξαγωγή ερμηνεύσιμων αλφαριθμητικών συμβολοσειρών και το εργαλείο hexdump για εξαγωγή ακολουθιών bytecode. Τα εξαγόμενα χαρακτηριστικά αξιοποιήθηκαν στη συνέχεια από τις μεθόδους μηχανικής μάθησης που προαναφέραμε.

Οι ***Kolter and Maloof*** (2006) [29] εφάρμοσαν n-grams και ενισχυμένα δέντρα αποφάσεων (boosted decision trees). Τα δεδομένα της μελέτης περιλάμβαναν 1.971 καλοήθη και 1.651 κακόβουλα εκτελέσιμα αρχεία. Τα καλοήθη αρχεία συλλέχθηκαν από φακέλους συστημάτων με Windows, καθώς και από την πλατφόρμα SourceForge. Τα κακόβουλα δείγματα προήλθαν από τον ιστότοπο VX Heavens και από ειδικούς. Ορισμένα αρχεία ήταν καλυμμένα με τεχνικές συμπίεσης ή κρυπτογράφησης, χωρίς να είναι γνωστό ποια συγκεκριμένα.Η επεξεργασία των δεδομένων περιλάμβανε τη μετατροπή των εκτελέσιμων αρχείων σε δεκαεξαδική μορφή και τη δημιουργία n-grams [29](συνεχόμενες ακολουθίες από n στοιχεία (συνήθως λέξεις ή χαρακτήρες) που εξάγονται από ένα κείμενο ή πρόταση) μεγέθους 4, προκειμένου να χρησιμοποιηθούν ως χαρακτηριστικά για ταξινόμηση. Η μελέτη χρησιμοποίησε μια υβριδική προσέγγιση με αλγορίθμους όπως TFIDF, Naive Bayes, Support Vector Machine (SVM), και δέντρα απόφασης. Οι ερευνητές πέτυχαν AUC έως και 0.9958.

Οι ***Rieck et al.***(2008) [30]: Μελέτησαν sandbox χαρακτηριστικά με συνολικό δείγμα 10.072 δείγματα από 14 διαφορετικές οικογένειες κακόβουλου λογισμικού πετυχαίνοντας 66 μεση ακρίβεια, χρησιμοποιώντας SVM, φτάνοντας 88 ακρίβεια στην ταξινόμηση στην καλυτερη περίπτωση σε οικογένεια malware..Παρόλο που το ποσοστό δεν είναι όσο υψηλό σε άλλες έρευνες είναι σημαντικό καθώς η μελέτη επεκτάθηκε για να περιλαμβάνει την αναγνώριση άγνωστης συμπεριφοράς, ενσωματώνοντας ένα μηχανισμό απόρριψης μέσω πιθανότητας πρόβλεψης που βασίζεται στην έξοδο των SVM, ταξινομώντας σωστά δείγματα από νέες οικογένειες ή κανονικά λογισμικά ως ¨άγνωστα¨. Επιπλέον, παρουσίασαν έναν μηχανισμό ερμηνείας των μοντέλων ταξινόμησης με βάση τα πιο διακριτικά χαρακτηριστικά κάθε οικογένειας, αποκαλύπτοντας μέχρι και ότι οι οικογένειες κάποιων κακόβουλων δειγμάτων έχουν κοινή καταγωγή.

Η σημασία της έρευνας έγκειται στο ότι συνδυάζει εποπτευόμενη μάθηση με δυναμική ανάλυση για την ταξινόμηση κακόβουλης συμπεριφοράς, προσφέροντας όχι μόνο ακρίβεια στην αναγνώριση γνωστών απειλών, αλλά και έναν αξιόπιστο μηχανισμό απόρριψης για νέα ή μη-κακόβουλα δείγματα. Αυτό αποτελεί κρίσιμο πλεονέκτημα έναντι στατικών μεθόδων, καθώς ενισχύει την ικανότητα γενίκευσης σε άγνωστες επιθέσεις και παρέχει αναλυτική κατανόηση των εντοπισμένων συμπεριφορών,που είναι χρήσιμη σε αναλυτές ασφαλείας.

Οι ***Nataraj et al.*** (2011) [31] κάνανε μετατροπή binary malware σε εικόνες και εξήγαγαν GIST χαρακτηριστικά από την αναπαράσταση του περιεχομένου κακόβουλου

λογισμικού σε γκρι κλίμακα.Στη συνέχεια κάνανε χρήση k-NN and SVM,σε σύνολο δεδομένων 25 διαφορετικών οικογενειών κακόβουλων αρχείων και συνολικά 9,458 δείγματα επιτυγχάνοντας μέχρι και 98% ακρίβεια. Εξήγαγαν GIST χαρακτηριστικά από την αναπαράσταση του περιεχομένου κακόβουλου λογισμικού σε γκρι κλίμακα.

Η τεχνική απεικόνισης δυαδικών αρχείων malware ως εικόνες επιτρέπει την αξιοποίηση οπτικών χαρακτηριστικών για την ανάλυση και ταξινόμησή τους. Η διαδικασία μετατροπής περιλαμβάνει τα εξής βήματα:

- Οι δυαδικοί κώδικες των αρχείων malware διαβάζονται ως ακολουθίες bytes (τιμές 0–255).
- Κάθε byte μετατρέπεται σε pixel με αντίστοιχη γκρι απόχρωση (grayscale), σχηματίζοντας έναν μονοδιάστατο πίνακα pixel.
- Ο πίνακας αυτός αναδιαμορφώνεται σε δισδιάστατη εικόνα με προκαθορισμένο πλάτος και κατάλληλο ύψος.
- Η εικόνα αποθηκεύεται σε μορφή PNG ή παρόμοια, αναπαριστώντας το εκτελέσιμο αρχείο ως οπτική πληροφορία.
- Οι παραγόμενες εικόνες εισάγονται σε σύστημα εξαγωγής χαρακτηριστικών (όπως το GIST στην παραπάνω εργασία) και στη συνέχεια ταξινομούνται .

Οι ***Tian et al.*** (2010) [32]Ανέλυσαν δυναμικά API calls,σε ένα σύνολο δεδομένων 1368 κακόβουλων και 456 καλόβουλων επιτυγχάνοντας 97.4% .Τα κακόβουλα δεδομένα συλλέχθηκαν απο το CA's VET zoo (www.ca.com) και εκτελούνται αρχεία (κακόβουλα και καλόβουλα) μέσα σε ένα εικονικό περιβάλλον. Κατά την εκτέλεση, χρησιμοποιείται ένα εργαλείο για την παρακολούθηση της συμπεριφοράς τους, δημιουργώντας αναφορές (trace reports). Η εξαγωγή χαρακτηριστικών γίνεται με την απομόνωση όλων των κλήσεων API και των παραμέτρων τους από τις αναφορές εκτέλεσης και τη δημιουργία μιας παγκόσμιας λίστας με αυτά τα strings, όπου για κάθε αρχείο καταγράφεται αν κάθε string υπάρχει (1) ή όχι (0), αγνοώντας τον αριθμό εμφανίσεων, καθώς αυτό δεν βελτίωνε την ταξινόμηση. Για την κατηγοριοποίηση, δοκιμάζονται τέσσερις βασικοί αλγόριθμοι (SVM, Random Forest, Decision Table, IB1).

Οι ***Islam et al.*** (2013) [33]με συνδυασμό opcode sequences, API calls Decision Trees και SVM,σε σύνολο δεδομένων 2948 φτάνοντας 99.20 ακρίβεια.Συγκεκριμένα συλλέγονται τόσο στατικά χαρακτηριστικά (opcode sequences) όσο και δυναμικά χαρακτηριστικά (API calls). Η διαδικασία εξαγωγής χαρακτηριστικών περιλαμβάνει την ανάλυση ακολουθιών εντολών και κλήσεων από τα εκτελέσιμα αρχεία. Τα πειράματα δείχνουν ότι ο συνδυασμός αυτών των χαρακτηριστικών και αλγορίθμων βελτιώνει την ανθεκτικότητα του μοντέλου σε μεταλλάξεις κακόβουλου λογισμικού και μειώνει το χρόνο επεξεργασίας σε σύγκριση με προγενέστερες μεθόδους.

Οι ***Anderson et al.*** (2011) [34] ανέλυσαν δυναμικά τις ακολουθίες εκτέλεσης προγραμμάτων χρησιμοποιώντας Markov chains και γραφικούς πυρήνες, σε ένα σύνολο δεδομένων 1.615 κακόβουλων και 615 καλόβουλων προγραμμάτων. Τα δεδομένα συλλέχθηκαν χρησιμοποιώντας το σύστημα ανάλυσης Ether, το οποίο παρέχει προστασία από τεχνικές ανίχνευσης εικονικών μηχανών και αποφυγή τροποποιήσεων του συστήματος-ξενιστή. Επιτεύχθηκε ακρίβεια 96.41% με τη μέθοδο των γραφικών πυρήνων, υπερβαίνοντας την απόδοση των συμβατικών μεθόδων n-gram και των υπογραφών antivirus.

Οι ***Dahl et al.*** [35] (2013) Εφάρμοσαν feed-forward neural networks, επιτυγχάνοντας 99.51% ακρίβεια με 142.000 δείγματα κακόβουλου λογισμικού. Συγκεκριμένα, οι συγγραφείς εφάρμοσαν ένα απλό προωθητικό νευρωνικό δίκτυο σε συνδυασμό με τυχαίες προβολές για τη μείωση της διάστασης των χαρακτηριστικών. Αξιοσημείωτο είναι ότι η προσθήκη επιπλέον κρυφών επιπέδων δεν βελτίωσε την απόδοση, υποδεικνύοντας ότι ένα ρηχό δίκτυο ήταν επαρκές για το συγκεκριμένο πρόβλημα.

Τα δεδομένα για την εκπαίδευση του συστήματος malware classification προέρχονται από πάνω από 2.6 εκατομμύρια αρχεία, από τα οποία περίπου 1.8 εκατομμύρια είναι κακόβουλα και 817 χιλιάδες είναι ακίνδυνα. Τα περισσότερα από αυτά τα αρχεία έχουν επισημανθεί (labeled) χειροκίνητα από αναλυτές, ενώ τα αρχεία benign προέρχονται από αξιόπιστες πηγές, όπως δημοφιλή λογισμικά (π.χ. Adobe Acrobat Reader). Τα κακόβουλα αρχεία ταξινομούνται σε 134 γνωστές οικογένειες malware και μια γενική κατηγορία malware , ενώ τα ακίνδυνα αρχεία αποτελούν ξεχωριστή κλάση.

Για την εξαγωγή χαρακτηριστικών (feature extraction), χρησιμοποιείται το σύστημα ανάλυσης της Microsoft, το οποίο εκτελεί ανάλυση των αρχείων σε ένα ελαφρύ εικονικό περιβάλλον (lightweight virtual machine). Από αυτή την ανάλυση εξάγονται τρεις βασικοί τύποι χαρακτηριστικών:

1. Null-terminated patterns : Πρότυπα συμβολοσειρών που βρίσκονται στη μνήμη κατά την εκτέλεση, τα οποία συνήθως αντιστοιχούν σε αλφαριθμητικές συμβολοσειρές του αρχείου.

2. API tri-grams : Τρίγλωσσα (τριπλέτες) διαδοχικών κλήσεων συστήματος (API calls), που βοηθούν στην κατανόηση της δυναμικής συμπεριφοράς του αρχείου.

3. API call + parameter value : Συνδυασμοί συγκεκριμένων κλήσεων API με τις παραμέτρους τους, που μπορούν να βοηθήσουν στον εντοπισμό ιδιαίτερων οικογενειών malware .

Η συνολική καταμέτρηση αυτών των χαρακτηριστικών φτάνει σε δεκάδες εκατομμύρια, αλλά μετά από μια διαδικασία επιλογής χαρακτηριστικών (feature selection) με βάση το mutual information , περιορίζονται σε περίπου 179 χιλιάδες χαρακτηριστικά που είναι πιο διακριτικά για κάθε κλάση. Αυτή η επιλογή επιτρέπει πιο αποδοτική εκπαίδευση των αλγορίθμων μηχανικής μάθησης.

Συμπεράσματα

Από την ανασκόπηση πιο παλιάς βιβλιογραφίας, προκύπτουν τα ακόλουθα βασικά συμπεράσματα σχετικά με τις τεχνικές ανίχνευσης και ταξινόμησης κακόβουλου λογισμικού:

- **Στατικές μέθοδοι ανάλυσης:** Παρουσιάζουν εξαιρετικά υψηλά ποσοστά ακρίβειας, σε περιπτώσεις αγγίζοντας ή και ξεπερνώντας το 99% . Ωστόσο, είναι ιδιαίτερα ευάλωτες σε τεχνικές απόκρυψης (obfuscation) και συμπίεσης (packing techniques), οι οποίες τροποποιούν την εμφάνιση του κακόβουλου λογισμικού χωρίς να αλλάζουν τη λειτουργικότητά του.

- **Δυναμικές μέθοδοι ανάλυσης:** Είναι πιο ανθεκτικές σε τεχνικές παραλλαγής και απόκρυψης, καθώς βασίζονται στη συμπεριφορά του κακόβουλου λογισμικού κατά την εκτέλεση. Παρ΄ όλα αυτά, έχουν αυξημένες απαιτήσεις σε

Πίνακας 3.1: Πίνακας μελετών ανίχνευσης κακόβουλου λογισμικού

Δημοσίευση	**Πηγή Δεδομένων**	**Συνολικά Δείγματα**	**Αλγόριθμος**	**Στόχος**	**Καλύτερη Επίδοση**
Schultz et al. [28] (2001)	Various sources	4266	Naive Bayes, J48, k-NN	Detection	97.11% acc.
Kolter and Maloof (2006) [29]	VX Heavens, SourceForge, download.com	3622	n-grams with Decision,naive Bayes, Trees	Detection	0.9958 ± 0.0024% AUC.
Rieck et al. (2008) [30]	most from Nepenthes	10.072	SVM	Classification	Up to 88% acc.
Nataraj et al. (2011) [31]	25 malware families	9.458	Binary to Image + k-NN, SVM	Classification	Up to 98% acc.
Tian et al. (2010) [32]	VET Zoo	1824	Bayesian Networks	Classification	97.4% acc.
Islam et al. (2013) [33]	Collected dataset	2948	Opcode + API + DT, SVM	Classification	99.16% acc.
Anderson et al. (2011) [34]	Multiple sources (binary, CFGs)	2.230	Multiple Kernel Learning (MKL)	Detection	96.41% acc.
Firdausi et al. (2010) [21]	VX Heavens, SourceForge,	3622	SVM, Naive Bayes, Decision Trees	Classification	SVM: 96.9% acc.
Dahl et al. (2013) [35]	Windows PE Executables	142.000	Deep Feedforward Neural Network	Detection	99.5% acc.
Wang et al(2011) [27]	VX Heavens, PChome Online	4.830	SVM	Detection	94.54% True Possitives

υπολογιστικούς πόρους και χρόνο εκτέλεσης, γεγονός που μπορεί να περιορίσει τη χρήση τους σε πραγματικούς χρόνους.

- **Υβριδικές προσεγγίσεις:** Ο συνδυασμός στατικών και δυναμικών χαρακτηριστικών προσφέρει βελτιστοποιημένη απόδοση, επιτυγχάνοντας ακρίβεια μεταξύ 98% και 99% σε σενάρια πραγματικής χρήσης. Αυτές οι μέθοδοι επιτυγχάνουν μια ισορροπία μεταξύ ακρίβειας, ταχύτητας και ανθεκτικότητας στις τεχνικές παραπλάνησης του κακόβουλου λογισμικού.

.

Συνοψίζοντας, μπορεί να συναχθεί το συμπέρασμα ότι μια κατηγοριοποίηση που βασίζεται στην αυτόματη ανάλυση κακόβουλου λογισμικού και στη χρήση τεχνικών μηχανικής μάθησης μπορεί να ανιχνεύσει το κακόβουλο λογισμικό αρκετά αποτελεσματικά και αποδοτικά όμως πάντα με σημαντικά περιθώρια λάθους, ειδικά όταν μιλάμε για προγράμματα που η μη ανίχνευση τους μπορεί να έχει καταστροφικές συνέπειες.

Σύμφωνα και με τους ***Gandotra et al.*** (2014) [36] στο άρθρο "Malware Analysis and Classification: A Survey", στο οποίο πραγματοποιείται εκτενής ανασκόπηση των μεθόδων ανίχνευσης της εποχής, έχει επιτευχθεί σημαντική πρόοδος στην ανίχνευση και κατηγοριοποίηση κακόβουλου λογισμικού. Παρ' όλα αυτά, οι προσεγγίσεις αυτές παρουσιάζουν αδυναμίες στην αντιμετώπιση μη ισορροπημένων συνόλων δεδομένων, καθώς και σε περιπτώσεις όπου χρησιμοποιούνται τεχνικές απόκρυψης από πλευράς του κακόβουλου λογισμικού.

Η αξιοποίηση τεχνικών μηχανικής μάθησης για την ανίχνευση και ταξινόμηση κακόβουλου λογισμικού έχει συμβάλει ουσιαστικά στη βελτίωση της ακρίβειας και της

Πίνακας 3.2: Πίνακας μελετών ανίχνευσης κακόβουλου λογισμικού ανά τύπο ανάλυσης

Δημοσίευση	Πηγή Δεδομένων	Δείγματα	Τύπος Ανάλυσης	Αλγόριθμος	Απόδοση
Schultz et al. [28] (2001)	Διάφορες πηγές	4.266	**Στατική**	Naive Bayes, J48, k-NN	97.11%
Kolter & Maloof [29] (2006)	VX Heavens, SourceForge	3.622	**Στατική**	n-grams, Boosted DT	0.9958 AΥ
Nataraj et al. [31] (2011)	25 οικογένειες	9.458	**Στατική**	Binary-to-Image + k-NN/SVM	98%
Wang et al. [27] (2011)	VX Heavens, PChome	4.830	**Στατική**	SVM (RBF kernel)	94.54% ΤΠ
Rieck et al. [30] (2008)	Nepenthes	10.072	**Δυναμική**	SVM	88%
Tian et al. [32] (2010)	CA's VET Zoo	1.824	**Δυναμική**	Bayesian Networks	97.4%
Anderson et al. [34] (2011)	Πολλαπλές πηγές	2.230	**Δυναμική**	Graph Kernels	96.41%
Firdausi et al. [21] (2010)	VX Heavens	3.622	**Δυναμική**	SVM, Naive Bayes	96.9%
Islam et al. [33] (2013)	Εσωτερικό dataset	2.948	**Υβριδική**	Opcode+API + DT/SVM	99.20%
Dahl et al. [35] (2013)	Windows PE	2.600.000	**Υβριδική**	Νευρωνικό Δίκτυο	99.51%

αποδοτικότητας των συστημάτων ανάλυσης. Στη συνέχεια, θα εξετάσουμε την περαιτέρω εξέλιξη των μεθόδων αυτών, εστιάζοντας σε μελέτες που βασίζονται στο Deep Learning, δηλαδή σε τεχνικές που αξιοποιούν μεγαλύτερο βάθος και περισσότερα επίπεδα επεξεργασίας με σκοπό την επίτευξη βελτιωμένων αποτελεσμάτων.

Όπως θα δούμε πέρα από την χρήση πιο περίπλοκων και απαιτητικών αλγορίθμων ή υπολογιστική και αλγοριθμικοί πρόοδος έχει επιτρέψει και την εκπαίδευση σε πολύ μεγαλύτερα Datasets

Deep Learning Bibliography

Από αυτό το σημείο και έπειτα, εξετάζουμε τεχνικές που είναι πιο σύγχρονες και σχετίζονται κυρίως με βαθύτερα και πιο πολύπλοκα νευρωνικά δίκτυα:

Οι ***Akhtar et Feng*** (2022) [37] παρουσίασαν ένα ολοκληρωμένο σύστημα ανίχνευσης και ταξινόμησης κακόβουλου λογισμικού, χρησιμοποιώντας δεδομένα του Canadian Institute for Cybersecurity. Το σύνολο δεδομένων περιλάμβανε 17.394 δείγματα, 51 οικογένειες κακόβουλου λογισμικού και 279 χαρακτηριστικά, τα οποία προήλθαν από εκτελέσιμα αρχεία σε δυαδική μορφή. Για την προεπεξεργασία των αρχείων χρησιμοποιήθηκε προστατευμένο περιβάλλον (VM) και εργαλεία όπως το PEiD εργαλείο ανάλυσης PE αρχείων για το unpacking.

Η μελέτη περιέλαβε διαδικασίες εξαγωγής και επιλογής χαρακτηριστικών, με σκοπό τη μείωση της υπερπροσαρμογής και την αύξηση της ακρίβειας. Η τεχνική επιλογής βασίστηκε σε ιεράρχηση χαρακτηριστικών μέσω feature rank, επιτρέποντας τον εντοπισμό των πλέον σχετικών με τη συμπεριφορά κακόβουλου λογισμικού.

Οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν περιλάμβαναν τους KNN, CNN, Naive Bayes, Random Forest, SVM Decision Tree (DT). Τα πειραματικά αποτελέσματα έδειξαν ότι το μοντέλο Decision Tree υπερείχε σε απόδοση, επιτυγχάνοντας ακρίβεια 99%,. Το CNN κατέγραψε ακρίβεια 98.76%, ενώ το SVM είχε ακρίβεια 96.41% . Η σύγκριση μεταξύ των μοντέλων υπογράμμισε την υπεροχή του DT, το οποίο επιτυγχάνει υψηλή απόδοση χωρίς ανάγκη εκτέλεσης των αρχείων, βασιζόμενο σε στατική ανάλυση με χαρακτηριστικά PE.

Συνολικά, η μελέτη κατέληξε στο συμπέρασμα ότι οι αλγόριθμοι DT, CNN, SVM αποτελούν τις πιο αξιόπιστες προσεγγίσεις για ανίχνευση κακόβουλου λογισμικού με χαμηλό ποσοστό ψευδώς θετικών, ενώ η επιλογή χαρακτηριστικών και η κατάλληλη προεπεξεργασία διαδραματίζουν κρίσιμο ρόλο στην ακρίβεια των αποτελεσμάτων.

Οι ***Akhtar et Feng*** (2022) [38] σε επόμενη τους έρευνα προτείνουν μια υβριδική μέθοδο για την ανίχνευση κακόβουλου λογισμικού, χρησιμοποιώντας την τεχνική

CNN-LSTM. Η μέθοδος αυτή συνδυάζει τις δυνατότητες των *Convolutional Neural Networks* (CNN) για την εξαγωγή τοπικών χωρικών χαρακτηριστικών με τις δυνατότητες των *Long Short-Term Memory networks* (LSTM) για την ανάλυση εξαρτήσεων. Τα δεδομένα που χρησιμοποιήθηκαν συλλέχθηκαν από το Kaggle και περιλάμβαναν δεδομένα από πέντε διαφορετικές οικογένειες κακόβουλου λογισμικού. Συνολικά, το σύνολο δεδομένων περιλάμβανε 43.867 δείγματα, τα οποία περιείχαν 100 στήλες και 5 γραμμές, με κάθε δείγμα να αντιπροσωπεύει ένα σύνολο χαρακτηριστικών που εξήχθησαν από καταγεγραμμένα αρχεία καταγραφής (log files).

Η διαδικασία εξαγωγής χαρακτηριστικών περιλάμβανε τη χρήση του CNN για την απομόνωση τοπικών χαρακτηριστικών που είναι σημαντικά για την ανίχνευση κακόβουλου λογισμικού. Τα χαρακτηριστικά αυτά περιλάμβαναν API calls, παραμέτρους εκτέλεσης και άλλες στατικές και δυναμικές πληροφορίες, οι οποίες χρησιμοποιήθηκαν για να κατανοήσουν τα μοτίβα και τις αλληλεπιδράσεις του κακόβουλου λογισμικού. Η εξαγωγή αυτών των χαρακτηριστικών επιτρέπει στο μοντέλο να αναγνωρίσει σημαντικά μοτίβα κακόβουλης δραστηριότητας χωρίς να απαιτεί από πριν τη σημείωση των δεδομένων.

Τα εξαγόμενα χαρακτηριστικά από το εισάγονται στην αρχιτεκτονική LSTM, η οποία έχει σχεδιαστεί για να επεξεργάζεται αλληλουχίες δεδομένων με εξάρτηση. Η αρχιτεκτονική LSTM, η οποία περιλαμβάνει πολλές επίπεδες μονάδες, εξάγει τις χρονικές σχέσεις μεταξύ των χαρακτηριστικών του κακόβουλου λογισμικού, επιτρέποντας την ανίχνευση μεγάλων χρονικών εξαρτήσεων και τη σωστή κατηγοριοποίηση του κακόβουλου λογισμικού. Η LSTM χρησιμοποιεί το μηχανισμό της μακροχρόνιας μνήμης για να αποθηκεύει και να ανακαλεί πληροφορίες από προηγούμενες χρονικές στιγμές, εξασφαλίζοντας έτσι ότι δεν παραβλέπεται καμία σημαντική πληροφορία κατά τη διάρκεια της διαδικασίας ανάλυσης.

Η συνδυασμένη αυτή αρχιτεκτονική CNN-LSTM πέτυχε εντυπωσιακή ακρίβεια 99%, επιτυγχάνοντας τα υψηλότερα ποσοστά ακριβείας σε σύγκριση με άλλες μεθόδους όπως τα *Decision Trees (DT)* και *Support Vector Machines (SVM)*. Η προσέγγιση αυτή εξαλείφει την ανάγκη για εκτενή μηχανική χαρακτηριστικών, ενώ παράλληλα βελτιώνει την ανίχνευση κακόβουλου λογισμικού, χωρίς να απαιτεί εξωτερική παρέμβαση στην εξαγωγή χαρακτηριστικών. Όλα τα παραπάνω οδηγούν σε μια πιο αποδοτική λύση για την ασφάλεια των δικτύων.

Οι ***Rizvi et al.*** (2021) [39] πρότειναν μια μη επιβλεπόμενη μέθοδο για την ανίχνευση κακόβουλου λογισμικού, αξιοποιώντας χαρακτηριστικά που εξήχθησαν από τις κεφαλίδες των εκτελέσιμων αρχείων τύπου PE. Συγκεκριμένα, από κάθε αρχείο δημιουργήθηκε ένα διάνυσμα 38 χαρακτηριστικών. Καθώς περίπου το 80% των διαθέσιμων δειγμάτων ήταν μη επισημασμένα, χρησιμοποιήθηκε ο αλγόριθμος k-means για τη δημιουργία ψευδό-ετικετών.

Ακολούθως, τα διανύσματα χαρακτηριστικών μαζί με τις παραγόμενες ετικέτες αποτέλεσαν είσοδο σε ένα πλήρως συνδεδεμένο νευρωνικό δίκτυο , στο οποίο ενσωματώθηκε μηχανισμός προσοχής. Ο μηχανισμός αυτός εφαρμόστηκε μετά το δεύτερο πλήρως συνδεδεμένο επίπεδο, με στόχο να ενισχύσει την ικανότητα του δικτύου να εστιάζει στα πιο σημαντικά ενδιάμεσα χαρακτηριστικά, βελτιώνοντας έτσι την απόδοσή του.

Σε ένα σύνολο 15.457 δειγμάτων, εκ των οποίων τα 6.681 ήταν καλοήθη, η προτεινόμενη μέθοδος πέτυχε εντυπωσιακή ακρίβεια 98.09%, αναδεικνύοντας την αποτελεσματικότητα του συνδυασμού μη επιβλεπόμενης μάθησης και μηχανισμού προσοχής για την ανίχνευση κακόβουλου λογισμικού.

Οι ερευνητές συνέλεξαν ένα σύνολο δεδομένων κακόβουλου λογισμικού σε πραγματικό χρόνο, μέσω της εγκατάστασης interaction honeypots σε δίκτυο επιχειρησιακού οργανισμού.

Οι ***Jha et al.*** (2020) [22] προτείνουν μια αποτελεσματική μέθοδο εντοπισμού χρησιμοποιώντας Recurrent Neural Networks (RNN) με βάση χαρακτηριστικά που εξάγονται από αρχεία assembly. Σκοπός είναι να Να εντοπιστεί κακόβουλο λογισμικό χρησιμοποιώντας τεχνικές Natural Language Processing (NLP) και RNN μοντέλα. Η βασική ιδέα είναι ότι οι εντολές αντιμετωπίζονται σαν ¨λέξεις' σε ένα ¨κείμενο¨, όπως γίνεται με φυσική γλώσσα. Οι συγγραφείς της μελέτης χρησιμοποίησαν ένα σύστημα δύο φάσεων για την ανίχνευση κακόβουλου λογισμικού μέσω συναρμολόγησης (.asm) αρχείων. Στην πρώτη φάση (Feature Extraction), εξήγαγαν τις εντολές μηχανής και τις μετέτρεψαν σε διανύσματα με τρεις μεθόδους: one-hot encoding, τυχαία διανύσματα, και Word2Vec embeddings, με εκπαίδευση του μοντέλου Skip-gram χρησιμοποιώντας τεχνική negative sampling. Στη δεύτερη φάση, χρησιμοποίησαν επαναληπτικό νευρωνικό δίκτυο (RNN) με LSTM cells, μελετώντας την απόδοσή του μεταβάλλοντας τη διάσταση εισόδου, το step size και τον αριθμό των νευρώνων. Σε πειράματα με 5.000 κακόβουλα και 5.000 μη-κακόβουλα δείγματα από το Microsoft Malware Classification Challenge (Kaggle 2015), διαπιστώθηκε ότι το μέγεθος του step επηρεάζει σημαντικά την απόδοση, ενώ τα διανύσματα Word2Vec υπερείχαν των άλλων τεχνικών. Χρησιμοποιήθηκε dropout (0.5) και ο βελτιστοποιητής RMSProp (με ρυθμό μάθησης 0.001) για τον περιορισμό της υπερπροσαρμογής. Συμπερασματικά, ο συνδυασμός Word2Vec embeddings και μικρού μεγέθους εισόδου αποτελεί την πιο αποδοτική προσέγγιση, ενώ η χρήση μεγάλου step size θα πρέπει να αποφεύγεται.

Οι ***Jayasudha et al.***(2023) [40] μελέτησαν την αποτελεσματικότητα έξι δημοφιλών αρχιτεκτονικών συνελικτικών νευρωνικών δικτύων (CNN) σε τρία σύνολα δεδομένων με διαφορετικό βαθμό ανισορροπίας μεταξύ των κλάσεων: το Malimg (έντονα ανισόρροπο), το MaleVis (ισορροπημένο) και ένα μεικτό σύνολο (μέτρια ανισόρροπο). Οι συγγραφείς επισημαίνουν τις αδυναμίες των παραδοσιακών μεθόδων εντοπισμού κακόβουλου λογισμικού που βασίζονται σε υπογραφές και προτείνουν τη transfer learning ως λύση, λόγω της ικανότητάς της να αυτοματοποιεί την εξαγωγή χαρακτηριστικών και να αναγνωρίζει πρότυπα που υποδηλώνουν κακόβουλη συμπεριφορά. Τα αποτελέσματα δείχνουν ότι η απόδοση των μοντέλων επηρεάζεται σημαντικά από την ανισορροπία των κλάσεων, με τα πιο ανισόρροπα σύνολα να απαιτούν λιγότερες εποχές εκπαίδευσης για σύγκλιση. Τα μοντέλα ResNet50, EfficientNetB0 και DenseNet169 παρουσίασαν καλή απόδοση σε όλα τα σύνολα.Στην εργασία όλες οι εικόνες τυποποιούνται σε μέγεθος 75×75 pixels και μετατρέπονται σε RGB, ενώ εφαρμόζεται augmentation με περιστροφές και μετατοπίσεις για να ενισχυθεί η ανθεκτικότητα των μοντέλων σε παραλλαγές των δεδομένων. Τα αποτελέσματα δείχνουν ότι η καλύτερη απόδοση επιτυγχάνεται στο blended dataset με το ResNet50, το οποίο πετυχαίνει 95% precision, υποδεικνύοντας ότι η μερική ισορροπία στα δεδομένα βελτιώνει τη συνολική σταθερότητα και ακρίβεια του ανιχνευτή. Επιπλέον, το EfficientNetB0 ξεχωρίζει στο πολύ μη ισορροπημένο Malimg, φτάνοντας μέχρι και 97% precision.

Οι ***Yung et Wang***(2023) [41] πρότειναν μία μέθοδο για την malware classification μέσω Convolutional Neural Networks (CNN), εστιάζοντας στην πρόκληση των imbalanced datasets. Συγκεκριμένα, τα εκτελέσιμα αρχεία μετατράπηκαν σε εικόνες grayscale, οι οποίες στη συνέχεια χρησιμοποιήθηκαν ως είσοδος σε ένα μοντέλο CNN.

Το σύνολο δεδομένων που χρησιμοποιήθηκε αποτελείται από 25 διαφορετικές οικογένειες κακόβουλου λογισμικού, περιλαμβάνοντας συνολικά περίπου 10.000 δείγματα.

Υπήρχε έντονη ανισορροπία, καθώς κάποιες οικογένειες περιλάμβαναν περισσότερα από 1.000 δείγματα, ενώ άλλες μόλις 50-150 δείγματα.

Η προτεινόμενη μέθοδος πέτυχε μέσο όρο ακρίβειας (accuracy) 94.5 , ενώ στις σπάνιες κατηγορίες παρατηρήθηκε σημαντική βελτίωση του F1-score, φτάνοντας έως και 91.

Οι **Seneviratne et al.**(2023) [42] πρότειναν μια καινοτόμο προσέγγιση για την ανίχνευση κακόβουλου λογισμικού με χρήση Vision Transformers (ViTs) και αυτοεπιβλεπόμενης μάθησης. Βασικό χαρακτηριστικό της μεθόδου τους ήταν η μετατροπή των εκτελέσιμων αρχείων σε εικόνες grayscale και η εφαρμογή τεχνικών masked autoencoding για την αυτόματη εξαγωγή χαρακτηριστικών.

Το πειραματικό πλαίσιο περιελάμβανε εκτεταμένη αξιολόγηση σε δημοφιλή σύνολα δεδομένων όπως το EMBER και δείγματα από το VirusTotal, με συνολικά πάνω από 100.000 δείγματα που ανήκαν σε 50+ οικογένειες κακόβουλου λογισμικού. Η μέθοδος τους ξεπέρασε σημαντικά τις υπάρχουσες προσεγγίσεις, πετυχαίνοντας:

- Ακρίβεια (accuracy) 98.2 σε γνωστά δείγματα
- Επιτυχία ανίχνευσης 93.5 για zero-day malware
- Βελτίωση του F1-score κατά 15 σε σπάνιες οικογένειες malware

Οι ***Venkatraman et al.*** (2019) [43] προτείνουν ένα ισχυρό υβριδικό μοντέλο βαθιάς μάθησης για την ανίχνευση και ταξινόμηση κακόβουλου λογισμικού μέσω ανάλυσης εικόνας και εξαγωγής χαρακτηριστικών. Η μελέτη χρησιμοποιεί έναν συνδυασμό εποπτευόμενων και μη εποπτευόμενων μοντέλων μάθησης με στόχο την κατηγοριοποίηση αρχείων κακόβουλου λογισμικού σε οικογένειες. Η πηγή δεδομένων περιλαμβάνει τόσο δημόσια σύνολα δεδομένων, όπως το Microsoft Malware Classification Challenge (BIG 2015) και το Malimg, όσο και ιδιωτικά σύνολα δεδομένων που συλλέχθηκαν από τους ίδιους τους ερευνητές, φτάνοντας συνολικά περίπου 75,000 δείγματα. Το 77% αυτών ήταν packed malware, που εντοπίστηκαν μέσω τεχνικών ανάλυσης δυαδικών αρχείων, ενώ το υπόλοιπο 23% ήταν unpacked executables.

Στο στάδιο εξαγωγής χαρακτηριστικών (feature extraction), χρησιμοποιούνται διάφορες τεχνικές: μετατροπή των εκτελέσιμων αρχείων σε grayscale images για ανάλυση μέσω CNN, εξαγωγή n-gram features από δυαδικά δεδομένα (n=1 έως 5), και στατιστική ανάλυση των Windows API call sequences, που αποτελούν δείκτες της συμπεριφοράς ενός αρχείου. Επιπλέον, εφαρμόζονται τεχνικές similarity mining χρησιμοποιώντας μετρικές απόστασης όπως η cosine similarity, για τη δημιουργία similarity matrices μεταξύ παραλλαγών κακόβουλου λογισμικού. Οι υψηλής διάστασης απεικονιστικές δυνατότητες του CNN μειώνονται μέσω PCA και t-SNE, ώστε να καταστεί δυνατή η ομαδοποίηση των δεδομένων με k-means clustering. Το τελικό μοντέλο πέτυχε accuracy έως και 96.3%, αποδεικνύοντας την αποτελεσματικότητα της προσέγγισης σε συνθήκες πραγματικού κόσμου.

Οι ***Nayr and Syam***(2024). [44] διεξήγαγαν μια συγκριτική ανάλυση μεταξύ Vision Transformers Convolutional Neural Networks (CNN) και Convolution Vision Transformer(CvT) για την ανίχνευση κακόβουλου λογισμικού. Η μελέτη τους, εξέτασε συστηματικά τις επιδόσεις και των τριών αρχιτεκτονικών σε διάφορα σύνολα δεδομένων. Η μελέτη αξιοποιεί το σύνολο δεδομένων MaleVis, το οποίο περιλαμβάνει 14.226 RGB εικόνες κακόβουλου και νόμιμου λογισμικού, μετασχηματισμένες από δυαδικά αρχεία μέσω του εργαλείου bin2png. Τα δεδομένα χωρίζονται σε σύνολα εκπαίδευσης, επικύρωσης και δοκιμής. Ακολουθεί προεπεξεργασία, η οποία περιλαμβάνει

αλλαγή μεγέθους, κανονικοποίηση καναλιών RGB και ενίσχυση δεδομένων (data augmentation) για την ενίσχυση της γενίκευσης. Στη συνέχεια, γίνεται λεπτομερής προσαρμογή (fine-tuning) τριών προεκπαιδευμένων μοντέλων: Vision Transformer (ViT), Convolutional Vision Transformer (CvT) και EfficientNet-B0. Για τα δύο πρώτα χρησιμοποιείται η βιβλιοθήκη Hugging Face Transformers, ενώ το EfficientNet-B0 βασίζεται στο TensorFlow Keras. Η προσαρμογή των μοντέλων περιλαμβάνει τροποποίηση της εξόδου για ταξινόμηση 26 κλάσεων και χρήση κατάλληλων παραμέτρων (π.χ. batch size 16, learning rate 0.0002). Η αξιολόγηση γίνεται με διάφορες μετρικές όπως accuracy, precision, recall και F1-score, με έμφαση στο τελευταίο λόγω της ανισορροπίας μεταξύ των κλάσεων. Τα αποτελέσματα δείχνουν ότι το CvT υπερέχει τόσο σε απόδοση (F1-score 0.96054) όσο και σε αποδοτικότητα εκπαίδευσης, καθιστώντας το την πιο κατάλληλη επιλογή για ανίχνευση κακόβουλου λογισμικού σε εικόνες τύπου byteplot.

Το πειραματικό πλαίσιο περιελάμβανε:

- Εκτεταμένη αξιολόγηση σε 3 δημοφιλή datasets (συνολικά 50.000+ δείγματα)
- Σύγκριση 6 διαφορετικών μοντέλων (3 Transformers,3 CNN)
- Ανάλυση υπολογιστικής απόδοσης και απαιτήσεων πόρων

Τα κύρια ευρήματα της έρευνας έδειξαν ότι:

- Οι Transformers πετύχαν υψηλότερη ακρίβεια (96.8 vs 94.2) σε μεγάλα σύνολα δεδομένων
- Οι CNN είχαν καλύτερη απόδοση σε μικρά σύνολα δεδομένων (92.1 vs 89.5)
- Οι Transformers απαιτούσαν 2-3 φορές περισσότερους υπολογιστικούς πόρους

Οι ***Rustam et al.***(2023) [45] προτείνουν μια προσέγγιση ταξινόμησης (classification) για την ανίχνευση κακόβουλου λογισμικού μέσω μεταφοράς μάθησης (transfer learning) και συνδυαστικών μοντέλων. Για το σκοπό αυτό χρησιμοποιείται το σύνολο δεδομένων Malimg, το οποίο περιέχει 9339 δείγματα από 25 διαφορετικές κατηγορίες malware. Κάθε δείγμα αντιπροσωπεύει μια γκρι εικόνα που προκύπτει από τη μετατροπή του αρχικού εκτελέσιμου αρχείου σε δισδιάστατη μορφή εικόνας. Αυτές οι εικόνες εισάγονται σε δύο προεκπαιδευμένα μοντέλα συνελικτικών νευρωνικών δικτύων (CNN), τα VGG16 και ResNet-50, τα οποία χρησιμοποιούνται μόνο για την εξαγωγή χαρακτηριστικών (feature extraction) χωρίς να επανεκπαιδεύονται. Το κάθε μοντέλο εξάγει 1024 χαρακτηριστικά για κάθε εικόνα, τα οποία στη συνέχεια συνενώνονται για να σχηματίσουν ένα υβριδικό διάνυσμα χαρακτηριστικών 2048 διαστάσεων. Το ενιαίο αυτό διάνυσμα τροφοδοτεί ένα διπλό σύστημα ταξινόμησης (Bi-model architecture), όπου το πρώτο μοντέλο (SVC, RF, LR ή KNN) παράγει πιθανότητες πρόβλεψης για κάθε κατηγορία, και το δεύτερο μοντέλο εκπαιδεύεται πάνω σε αυτές τις πιθανότητες ώστε να βελτιώσει την τελική πρόβλεψη. Ο τελικός στόχος της μεθόδου είναι η ακριβής ταξινόμηση της εικόνας σε μία από τις 25 κατηγορίες. Το μοντέλο στις περιπτώσεις Bi-SVC, RF, LR πέτυχε ακόμη και 100% επιτυχία. [46]

Στην εργασία του ***Λαμπράκη Δημήτρη*** (2022) [47] πραγματοποιείται συλλογή δεδομένων από το VirusShare για κακόβουλα δείγματα, καθώς και από το λειτουργικό σύστημα Windows για καλόβουλα. Τα εκτελέσιμα αρχεία μετατρέπονται σε γκρι εικόνες (grayscale), ακολουθώντας τη μεθοδολογία των Nataraj et al.(2011) [31] δηλαδή

την μετατροπή των binary σε εικόνες. Στη συνέχεια, με τεχνικές padding ή cropping, οι εικόνες κανονικοποιούνται στη διάσταση 1024×1024 και χρησιμοποιούνται για την εκπαίδευση ενός συνελικτικού νευρωνικού δικτύου (CNN).

Παρόλο που τα αποτελέσματα είναι ικανοποιητικά, η διαδικασία είναι ιδιαίτερα χρονοβόρα και απαιτητική σε υπολογιστικούς πόρους. Για τον λόγο αυτό, διερευνάται η μείωση της διάστασης των εικόνων με χρήση autoencoder σε μορφή $512 \times 512 \times 4$, και η εκπαίδευση του δικτύου ResNet-18. Τα τελικά αποτελέσματα δείχνουν σημαντική βελτίωση τόσο στην ακρίβεια όσο και στην αποδοτικότητα των πόρων με ποσοστό accuracy εώς και 96%.

Οι ***Ben Abdel Ouahab et al.*** (2023) [48] στη μελέτη τους συγκρίνουν την απόδοση των μοντέλων Vision Transformer (ViT) με τα παραδοσιακά Συνελικτικά Νευρωνικά Δίκτυα

Το μοντέλο ViT πέτυχε ακρίβεια 98.38% και απώλεια 5.17, ενώ το CNN εμφάνισε ακρίβεια 97.9% και απώλεια 12.29. Η αξιολόγηση βασίστηκε στη βάση δεδομένων Malimg, η οποία περιλαμβάνει 9369 εικόνες κακόβουλου λογισμικού, κατανεμημένες σε 25 διαφορετικές κατηγορίες.

Παρόλο που τα μοντέλα ViT απαιτούν αυξημένους υπολογιστικούς πόρους και μεγαλύτερο χρόνο εκπαίδευσης,στην έρευνα υπερτερούν σε απόδοση, ιδίως σε μεγάλα και πολύπλοκα σύνολα δεδομένων. Αντίθετα, τα CNN παραμένουν αποτελεσματικά για μικρότερα σύνολα, λόγω της αποδοτικότητας και της ταχύτερης εκπαίδευσής τους.

3.4 Σύνοψη και Συμπεράσματα

Οι μέθοδοι αντιμετώπισης του κακόβουλου λογισμικού παραθέσαμε παραπάνω αποτελούν μόνο ένα μικρό μέρος της συνολικής έρευνας που έχει πραγματοποιηθεί στο πεδίο και αποτελούν ένα κομμάτι από τις εργασίες που έχουν ξεχωρίσει. Η διαθεσιμότητα μεγάλων και αρκετά ετερογενών συνόλων δεδομένων, σε συνδυασμό με τη ραγδαία πρόοδο της μηχανικής μάθησης, προσφέρει πολλά νέα περιθώρια πειραματισμού και ουσιαστικής βελτίωσης των υφιστάμενων συστημάτων ανίχνευσης.

Προφανώς όμως, οι απειλές αυξάνονται καθημερινά όπως και οι μέθοδοι ανίχνευσεις. Χαρακτηριστικά, συγκρίνοντας τη βιβλιογραφία των τελευταίων χρόνων παραπάνω, παρατηρείται σημαντική αύξηση τόσο στον αριθμό όσο και στην πολυπλοκότητα των κακόβουλων δειγμάτων, αλλά και στις τεχνικές που χρησιμοποιούνται για την παραγωγή τους. Ταυτόχρονα όμως, οι επαγγελματίες στον τομέα της ασφάλειας απαντούν με την ανάπτυξη όλο και πιο σύνθετων συστημάτων, τα οποία εκπαιδεύονται πάνω σε τεράστιους όγκους δεδομένων και με πολλές διαφορετικές προσεγγίσεις.

Λαμβάνοντας υπόψη τα παραπάνω και με βάση τη σχετική βιβλιογραφία, στόχος της παρούσας εργασίας είναι η ανάπτυξη ενός εργαλείου το οποίο να διατηρεί χαμηλές απαιτήσεις σε υπολογιστικούς πόρους, ενώ ταυτόχρονα να είναι ικανό να ανταποκρίνεται σε σύγχρονες και μελλοντικές τεχνικές παρακάμψης συστημάτων ανίχνευσης.

Δύο από τις τεχνολογίες που αξιολογούνται στην παρούσα μελέτη είναι τα Convolutional Neural Networks (CNN) και οι Vision Transformers (ViT), των οποίων τα πειραματικά αποτελέσματα κρίνονται ιδιαίτερα ενθαρρυντικά. Εξετάζουμε την απόδοσή τους σε ένα πολυδιάστατο και ανομοιογενές σύνολο δεδομένων, το οποίο περιλαμβάνει δείγματα διαφορετικών μεγεθών, λειτουργιών και σκοπών — στοιχεία που αντικατοπτρίζουν τις συνθήκες του πραγματικού κόσμου.

Από την παραπάνω ανασκόπηση παρατηρούμε ότι η Δυναμική ανάλυση ξεχωρίζει στα αποτελέσματα, παρόλα αυτά είναι πιο απαιτητική σε χρόνο και πόρους. Στο πλα-

Πίνακας 3.3: Πίνακας Deep Learning

Δημοσίευση	**Πηγή Δεδομένων**	**Συνολικά Δείγματα**	**Αλγόριθμος**	**Στόχος**	**Καλύτερη Επίδοση**
Akhtar et Feng. (2022) [37]	Canadian Institute for Cybersecurity	17.394	Detection Tree(DT)	Classification	99% acc.
Akhtar et Feng. (2022) [38]	Kaggle	43.867	CNN-LSTM	Detection	99% acc.
Rivzi et al. (2021) [39]	Self created with honeypots	19.611	ANN+Attention mechanism	Detection	98.0% acc.
Jha et al. (2020) [22]	Microsoft malware challenge (Kaggle)	10.000	Recurrent Neural Networks (RNN)	Detection	97.8% AUC.
Nayr and Syam. (2024) [44]	MaleVis	14.226	CvT	Classification	, 95.93% acc.
Seneviratne et al. (2022) [42]	AndroZoo	1.2 million Android applications	SHERLOCK self supervised ViT	Detection	97%acc .
Athiwatarkul et al. (2017)	Self created	75000	50%	Classification	LSTM, 80% TPR
Δημητρης Λαμπράκης(2022)	VirusShare	5000	CNN	Detection	LSTM, 80% TPR
Ben Abdel Ouahab et al. (2023) [48]	Malimg	9.339	ViT,CNN	Classification	ViT 98,38% acc

ίσιο των vision transformers είναι πάρα πολύ καλά στο παιδίο του classification απαιτώντας βέβαια αρκετά περισσότερα δεδομένα από άλλα δίκτυα και αρκετά ισορροπημένα δεδομένων. Από τις παρπάνω εργασίες η εργασία του Δημήτρη Λαμπράκη (2022) [47] αποτελεί ένα σημαντικό σημείο αναφοράς, παρουσιάζοντας την αποτελεσματικότητα του ResNet-18 στην ανίχνευση κακόβουλου λογισμικού οπώς και των Nayr and Syam [44] και Ben Abdel Ouahab et al(2023) [48] που συγκρίνανε Συνελικτικά Νευρωνικά Δίκτυα με Vision Transformer. Στο πλαίσιο της εργασίας μας, επιδιώκουμε να συγκρίνουμε δίκτυα CNN με την τεχνολογία των Vision Transformers, με σκοπό την αποτίμηση της σχετικής απόδοσής τους.

Δεδομένων των αυξημένων υπολογιστικών απαιτήσεων του δικτύου ViT,θέλουμε να επιχειρήσουμε την αξιοποίηση τεχνικών συμπίεσης μέσω autoencoders,σε ακόμη μεγαλύτερο βαθμό με στόχο τη βελτιστοποίηση των πόρων και τη διατήρηση της ακρίβειας σε υψηλά επίπεδα και να κάνουμε αυτήν την διαδικασία συγκριτικά με ΄NN.

Οπότε όσον αφορά τη δική μας μελέτη, πιο αναλυτικά επικεντρωνόμαστε στη συγκριτική αξιολόγηση των Συνελικτικών Νευρωνικών Δικτύων (CNN) και των Vision

Transformers (ViT). Από τα διαθέσιμα δεδομένα, η υπεροχή των ViT είναι εμφανής, ιδιαίτερα σε καθήκοντα ταξινόμησης. Παρ' όλα αυτά, παρατηρείται έλλειψη μελετών που να εστιάζουν σε σενάρια όπου το σύνολο δεδομένων είναι εξαιρετικά ανομοιογενές ή περιλαμβάνει adversarial επιθέσεις. Σε αυτό ακριβώς το πλαίσιο εντάσσεται η παρούσα εργασία, η οποία στοχεύει να διερευνήσει την αποδοτικότητα των ViT υπό συνθήκες αυξημένης ποικιλομορφίας και πολυπλοκότητας των εισόδων, καθώς και την ανοχή τους σε τεχνικές παραπλάνησης.

Κεφάλαιο 4

Προκλήσεις στην Ανίχνευση Κακόβουλου Λογισμικού

Η ραγδαία εξέλιξη της τεχνολογίας συνοδεύεται αναπόφευκτα από την παράλληλη ανάπτυξη απειλών που στοχεύουν στην υπονόμευση της ασφάλειας των πληροφοριακών συστημάτων. Μία από τις πιο πολύπλοκες και επίμονες προκλήσεις σε αυτό το πλαίσιο είναι η συνεχώς εξελισσόμενη απειλή του κακόβουλου λογισμικού. Οι επιτιθέμενοι προσαρμόζουν διαρκώς τις τεχνικές τους, αξιοποιώντας τις τεχνολογικές εξελίξεις τόσο για τη δημιουργία νέων μορφών κακόβουλου κώδικα όσο και για την αποφυγή ανίχνευσης από τα συστήματα ασφαλείας.

Σύγχρονες τεχνικές απόκρυψης, όπως η κρυπτογράφηση και το packing, καθώς και πιο εξελιγμένες στρατηγικές αποφυγής, χρησιμοποιούνται για την παράκαμψη των αμυντικών μηχανισμών. Οι τεχνικές αυτές καθιστούν την αξιόπιστη και έγκαιρη ανίχνευση ιδιαίτερα δύσκολη, εντείνοντας την ανάγκη για καινοτόμες και πιο αποτελεσματικές προσεγγίσεις.

Μέσα σε αυτό το απαιτητικό περιβάλλον, η αξιοποίηση τεχνικών μηχανικής μάθησης – και ιδίως των νευρωνικών δικτύων – προσφέρει σημαντικές δυνατότητες. Τα συστήματα αυτά έχουν την ικανότητα να μαθαίνουν από δεδομένα και να γενικεύουν, εντοπίζοντας ακόμη και μη προφανή πρότυπα κακόβουλης συμπεριφοράς. Ωστόσο, η χρήση τους συνοδεύεται από σοβαρές προκλήσεις. Η ευπάθεια τους σε adversarial attacks συγκαταλέγεται στους σημαντικότερους περιορισμούς: μικρές, σχεδόν ανεπαίσθητες τροποποιήσεις στο κακόβουλο λογισμικό ενδέχεται να παραπλανήσουν τα συστήματα ανίχνευσης, οδηγώντας σε ψευδώς θετικά ή αρνητικά αποτελέσματα.

Περαιτέρω δυσκολίες απορρέουν από την ανομοιομορφία και την ασυμμετρία των δεδομένων εκπαίδευσης, γεγονός που μπορεί να προκαλέσει φαινόμενα υπερεκπαίδευσης ή κακής γενίκευσης των μοντέλων. Επιπλέον, η συνεχής δημιουργία νέων παραλλαγών κακόβουλου λογισμικού με μεταβαλλόμενα χαρακτηριστικά και συμπεριφορές καθιστά την ταξινόμησή τους εξαιρετικά περίπλοκη και απαιτητική.

Σκοπός του παρόντος κεφαλαίου είναι πριν ξεκινήσουμε την πειραματική διαδικασία να εξετάσουμε τα βασικά προβλήματα που αντιμετωπίζουν τα συστήματα ανίχνευσης κακόβουλου λογισμικού, καθώς και να αναδείξει τις δυνατότητες που προσφέρουν οι σύγχρονες τεχνολογίες, με έμφαση στις μεθόδους τεχνητής νοημοσύνης και τη συμβολή τους στη δημιουργία πιο προσαρμόσιμων και ανθεκτικών μηχανισμών ανίχνευσης. Η προσέγγιση που ακολουθείται είναι διττή: αφενός επικεντρώνεται στην υπολογιστική πολυπλοκότητα του ίδιου του προβλήματος, και αφετέρου στις τεχνικές που έχουν αναπτυχθεί για να παρακάμπτουν ή να παραπλανούν τους ανιχνευτές.

4.1 Η Φύση του Προβλήματος

Η ανίχνευση κακόβουλου λογισμικού αποτελεί ένα εξαιρετικά περίπλοκο και απαιτητικό ζήτημα, το οποίο έχει χαρακτηριστεί ως και πρόβλημα της κατηγορίας NP-complete [49]. Αυτή η κατηγοριοποίηση βασίζεται στη θεωρία υπολογιστικής πολυπλοκότητας και αναδεικνύει τη θεμελιώδη δυσκολία εξεύρεσης αποδοτικών, καθολικών λύσεων για την ανίχνευση όλων των πιθανών μορφών κακόβουλου κώδικα [18, 50].Συγκεκριμένα η ανίχνευση κακόβουλου λογισμικού αποτελεί ένα εξαιρετικά περίπλοκο και απαιτητικό ζήτημα. Η αξιόπιστη ταυτοποίηση ιών περιορισμένου μήκους έχει αποδειχθεί ότι είναι πρόβλημα της κατηγορίας NP-complete, γεγονός που αναδεικνύει τη θεμελιώδη δυσκολία εξεύρεσης καθολικών και αποδοτικών λύσεων για την ανίχνευση όλων των πιθανών μορφών κακόβουλου λογισμικού.

4.1.1 NP και NP-complete Προβλήματα

Τα προβλήματα της κατηγορίας NP (Nondeterministic Polynomial time) χαρακτηρίζονται από την ιδιότητα ότι, αν δοθεί μια υποψήφια λύση, η ορθότητά της μπορεί να επαληθευτεί σε πολυωνυμικό χρόνο από έναν ντετερμινιστικό αλγόριθμο. Ένα πρόβλημα χαρακτηρίζεται ως NP-complete όταν:

- Ανήκει στην κατηγορία NP.
- Κάθε άλλο πρόβλημα της κατηγορίας NP μπορεί να αναχθεί σε αυτό μέσω πολυωνυμικής μείωσης.

4.1.2 Η ανίχνευση κακόβουλου λογισμικού ως Αλγοριθμικό πρόβλημα

Η ανίχνευση κακόβουλου λογισμικού συχνά προϋποθέτει την αναγνώριση του αν ένα πρόγραμμα εκτελεί κακόβουλες λειτουργίες, κάτι που, σε αρκετές περιπτώσεις, συνδέεται με το *Πρόβλημα Σταματήματος* (Halting Problem) [51].

Αναγωγή από το Πρόβλημα Σταματήματος

Το Πρόβλημα Σταματήματος αποτελεί ένα κλασικό παράδειγμα μη υπολογίσιμου προβλήματος και διατυπώνεται ως εξής: *«Δεδομένου ενός προγράμματος και μιας εισόδου, υπάρχει τρόπος να αποφασίσουμε αν το πρόγραμμα θα τερματίσει ή θα συνεχίσει την εκτέλεσή του επ' αόριστον·»*

Δεδομένου ότι το Πρόβλημα Σταματήματος έχει αποδειχθεί ότι είναι άλυτο, δεν υπάρχει καθολικός αλγόριθμος που να μπορεί να το επιλύσει για κάθε δυνατή είσοδο. Αυτή η θεμελιώδης αδυναμία έχει σημαντικές συνέπειες για την ασφάλεια λογισμικού. Συγκεκριμένα, η ασφαλής αναγνώριση κακόβουλης συμπεριφοράς προϋποθέτει, σε πολλές περιπτώσεις, πλήρη κατανόηση της συμπεριφοράς ενός προγράμματος υπό οποιεσδήποτε συνθήκες εκτέλεσης. Εφόσον αυτό είναι θεωρητικά ανέφικτο, δεν μπορεί να υπάρξει απόλυτα αξιόπιστη και γενική μέθοδος εντοπισμού όλων των μορφών κακόβουλου λογισμικού.

Συνεπώς, οι υφιστάμενες μέθοδοι ανίχνευσης βασίζονται σε προσεγγιστικές τεχνικές, οι οποίες αξιοποιούν υποθέσεις, μοτίβα και στατιστικά χαρακτηριστικά. Αν και

αυτές οι τεχνικές μπορούν να είναι ιδιαίτερα αποτελεσματικές στην πράξη, συνοδεύονται πάντοτε από πιθανότητα σφάλματος, είτε υπό τη μορφή ψευδώς θετικών, είτε ψευδώς αρνητικών εντοπισμών.

Μη υπολογισιμότητα και ανάλυση συμπεριφοράς

Άρα η ανίχνευση κακόβουλου λογισμικού, στη γενική της μορφή, συνδέεται στενά με το Πρόβλημα Σταματήματος, το οποίο έχει αποδειχθεί ότι είναι μη υπολογίσιμο. Συνεπώς, δεν υπάρχει καθολικός αλγόριθμος που να μπορεί να αποφασίσει με απόλυτη ακρίβεια αν ένα οποιοδήποτε πρόγραμμα είναι κακόβουλο, σε κάθε πιθανό σενάριο εκτέλεσης.

Ωστόσο, σε περιορισμένες εκδοχές ή όταν επιβάλλονται συγκεκριμένοι περιορισμοί στα δεδομένα ή στα χαρακτηριστικά του προς ανάλυση κώδικα, το πρόβλημα μπορεί να γίνει υπολογίσιμο και να προσεγγιστεί με αλγορίθμους από την κατηγορία NP ή ακόμα και P. Σε κάποιες πρακτικές υλοποιήσεις, προβλήματα που σχετίζονται με την ανίχνευση κακόβουλου λογισμικού μπορεί να είναι NP-hard (δηλαδή ιδιαίτερα δύσκολα), αλλά όχι NP-complete στη γενική τους μορφή.

Πιο συγκεκριμένα, λοιπόν κατά περιπτώσεις:

- Μπορεί να ανήκει στην κατηγορία P, αν υπάρχει ένας αποδοτικός αλγόριθμος που μπορεί να το λύσει σε λογικό χρόνο. Στην πράξη, αυτό συμβαίνει όταν το πρόβλημα είναι καλά περιορισμένο - για παράδειγμα, όταν ψάχνουμε για συγκεκριμένες, γνωστές υπογραφές κακόβουλου λογισμικού.

- Μπορεί να ανήκει στην κατηγορία NP, αν μπορούμε τουλάχιστον να ελέγξουμε γρήγορα αν μια προτεινόμενη λύση (π.χ., η ταξινόμηση ενός αρχείου ως κακόβουλου) είναι σωστή. Αυτό είναι συχνά εφικτό όταν χρησιμοποιούμε μοντέλα μηχανικής μάθησης.

- Μπορεί να είναι NP-hard, δηλαδή τόσο δύσκολο που αν βρεθεί τρόπος να λυθεί αποδοτικά, τότε θα μπορούσαμε να λύσουμε και πολλά άλλα εξίσου δύσκολα προβλήματα. Αυτή είναι η κατάσταση για πολλές μορφές γενικής ανίχνευσης κακόβουλου λογισμικού.

- Μπορεί ακόμη να ξεφεύγει από αυτές τις κλασικές κατηγορίες, όπως συμβαίνει όταν το πρόβλημα περιλαμβάνει στοχαστικά στοιχεία, συνεχείς παραμέτρους, ή απειροελάχιστες αλλαγές που το κάνουν ακόμη πιο πολύπλοκο από τα τυπικά προβλήματα πολυπλοκότητας.

Αυτή η ταξινόμηση βοηθά να κατανοήσουμε γιατί δεν υπάρχει μια λύση για το πρόβλημα του κακόβουλου λογισμικού, και συνέχεια προκύπτουν νέες προσεγγίσεις που συχνά βασίζονται σε πολύπλοκους αλγορίθμους, στατιστικά μοντέλα και προσεγγιστικές λύσεις που ισορροπούν ανάμεσα στην ακρίβεια και την υπολογιστική αποδοτικότητα.

Η ανάλυση συμπεριφοράς ενός προγράμματος προϋποθέτει την προσομοίωση της εκτέλεσής του σε ποικίλα σενάρια, κάτι που απαιτεί σημαντικούς υπολογιστικούς πόρους. Τεχνικές που βασίζονται σε στατιστικά μοντέλα και τεχνητή νοημοσύνη προσπαθούν να μετριάσουν αυτή τη δυσκολία, ωστόσο καμία μέθοδος δεν μπορεί να προσφέρει απόλυτη ακρίβεια ή καθολική κάλυψη.

Παρακάτω θα δούμε τις βασικές μεθοδολογίες που χρησιμοποιούνται για να παραπλανήσουν έναν ανιχνευτή.

4.2 Βασικές Τεχνικές Απόκρυψης και εξαπάτησης ανιχνευτών

Η μεγαλύτερη πρόκληση στην ανίχνευση κακόβουλου λογισμικού είναι η συνεχής εξέλιξή του. Οι προγραμματιστές κακόβουλου λογισμικού χρησιμοποιούν προχωρημένες τεχνικές για να αποφεύγουν τον εντοπισμό, όπως:

4.2.1 Τεχνικές Απόκρυψης

Οι τεχνικές απόκρυψης επιδιώκουν να αλλάξουν την εμφάνιση του κακόβουλου κώδικα ή να δυσκολέψουν την ανάλυσή του, είτε στατικά είτε δυναμικά. Η αποτελεσματικότητα αυτών των μεθόδων αυξάνεται διαρκώς, ιδιαίτερα όταν συνδυάζονται μεταξύ τους. Παρακάτω παρουσιάζονται οι πιο διαδεδομένες μορφές απόκρυψης που χρησιμοποιούνται σήμερα.

- Polymorphism and Metamorphism: Το κακόβουλο λογισμικό μπορεί να αλλάζει τη μορφή του κάθε φορά που εκτελείται, ώστε να αποφεύγει την ανίχνευση από παραδοσιακά εργαλεία ασφαλείας.
- Obfuscation: Οι τεχνικές σύγχυσης κώδικα καθιστούν δυσκολότερη την ανάλυση ενός προγράμματος, προσθέτοντας επιπλέον μη αναγνωρίσιμες εντολές ή αναδιατάσσοντας τις υπάρχουσες.
- Packing (πακετάρισμα,τεχνική συμπίεσης ή απόκρυψης εκτελέσιμων αρχείων): Το κακόβουλο λογισμικό συχνά κρύβει τον πραγματικό του κώδικα, αποκαλύπτοντας τον μόνο κατά την εκτέλεση.
- Zero-Day Attacks: Εκμετάλλευση άγνωστων ευπαθειών που δεν έχουν ακόμη επιδιορθωθεί.

4.2.2 Adversarial Attacks

Οι επιθέσεις adversarial αποτελούν μια εξίσου σημαντική απειλή για τα συστήματα μηχανικής μάθησης, ειδικά στον τομέα της ανίχνευσης κακόβουλου λογισμικού. Στόχος τους είναι να παραπλανήσουν τους ταξινομητές και τα φίλτρα ασφαλείας μέσω εσκεμμένων, αλλά συχνά ανεπαίσθητων, τροποποιήσεων των εισόδων. Οι επιθέσεις αυτές λοιπόν αξιοποιούν αδυναμίες και αλλάζουν χαρακτηριστικά του κακόβουλου λογισμικού, ώστε να παραμένει λειτουργικό αλλά να μην αναγνωρίζεται από τα προγράμματα που έχουν φτιαχτεί για την ανίχνευση του.

Τέτοιες τεχνικές μπορούν να περιλαμβάνουν την προσθήκη αχρείαστου κώδικα (code padding), την αναδιάταξη εντολών, ή την έγχυση δεδομένων που παρακάμπτουν τους ελέγχους, χωρίς να επηρεάζουν την κακόβουλη συμπεριφορά του αρχείου. Αυτό καθιστά τα συστήματα που βασίζονται αποκλειστικά στην παραδοσιακή εποπτευόμενη μάθηση ευάλωτα.

Η αντιμετώπιση αυτών των επιθέσεων απαιτεί την ανάπτυξη ανθεκτικών μοντέλων και στρατηγικών άμυνας. Ένας βασικός μηχανισμός είναι η χρήση adversarial training, όπου το μοντέλο εκπαιδεύεται όχι μόνο σε κανονικά, αλλά και σε εσκεμμένα παραμορφωμένα δείγματα, ώστε να μάθει να εντοπίζει και τέτοιες παραλλαγές. Επιπλέον, η

ενσωμάτωση αμυντικών αλγορίθμων, όπως η ανίχνευση ανωμαλιών, η κανονικοποίηση εισόδων, ή ακόμη και τεχνικές βαθύτερης μάθησης (deep learning), ενισχύει την ανθεκτικότητα των συστημάτων και μειώνει την αποτελεσματικότητα των adversarial attacks

4.2.3 Ανισορροπία Δεδομένων

Ένα από τα μεγαλύτερα προβλήματα στη δημιουργία ακριβών μοντέλων ανίχνευσης είναι η ανισορροπία των δεδομένων. Τα σύνολα δεδομένων συχνά περιέχουν σημαντικά περισσότερα δείγματα από την ασφαλή κατηγορία σε σχέση με τα κακόβουλα, γεγονός που μπορεί να οδηγήσει σε μεροληψία του μοντέλου προς την ασφαλή κατηγορία. Τεχνικές όπως η αύξηση δεδομένων (data augmentation), η επαναδειγματοληψία (resampling) και οι προσαρμοσμένες απώλειες (weighted loss functions) χρησιμοποιούνται για τη μείωση αυτής της ανισορροπίας χωρίς όμως πάντα να είναι βέλτιστα απαραίτητα τα αποτελέσματά τους.

Πιο συγκεκριμένα, αυτές οι τεχνικές δεν εγγυώνται πάντα ιδανικά αποτελέσματα. Σε ορισμένες περιπτώσεις, μπορεί να οδηγήσουν σε υπερεκπαίδευση ή σε ψευδώς θετικά αποτελέσματα, ιδιαίτερα όταν τα τεχνητά ή υπερδειγματοληπτημένα δεδομένα δεν αντιπροσωπεύουν επαρκώς τη συμπεριφορά πραγματικών επιθέσεων. Επομένως, απαιτείται προσεκτικός σχεδιασμός και αξιολόγηση των μεθόδων εξισορρόπησης ανάλογα με τα χαρακτηριστικά του κάθε συνόλου δεδομένων και το πλαίσιο εφαρμογής.

4.2.4 Συνεχώς Ανανεωμένες Κατηγορίες Κακόβουλου Λογισμικού

Το κακόβουλο λογισμικό αποτελεί έναν από τους πιο δυναμικούς και συνεχώς εξελισσόμενους τομείς στην ασφάλεια πληροφοριών. Νέες μορφές, παραλλαγές και τεχνικές επίθεσης αναπτύσσονται και εμφανίζονται καθημερινά, γεγονός που καθιστά την ταξινόμησή τους σε σταθερές και προκαθορισμένες κατηγορίες μια ιδιαίτερα σύνθετη και προβληματική διαδικασία. Πολλές φορές, τα νέα δείγματα malware δεν ταιριάζουν απόλυτα στις ήδη υπάρχουσες οικογένειες ή κατηγορίες, είτε επειδή εισάγουν πρωτοποριακές τεχνικές απόκρυψης, είτε επειδή συνδυάζουν χαρακτηριστικά από πολλαπλές κατηγορίες, δημιουργώντας υβρίδια.

Η ανάγκη για ευέλικτες και αυτοματοποιημένες μεθόδους ανίχνευσης και ταξινόμησης οδήγησε στη χρήση τεχνικών μη επιβλεπομένης μάθησης, οι οποίες δεν βασίζονται σε προϋπάρχουσες ετικέτες ή κατηγορίες. Μέσα σε αυτές τις τεχνικές, η ομαδοποίηση (clustering) παίζει καθοριστικό ρόλο, καθώς επιτρέπει την ομαδοποίηση δειγμάτων με βάση τα κοινά τους χαρακτηριστικά, αποκαλύπτοντας κρυφές δομές και πιθανές νέες κατηγορίες malware χωρίς την ανάγκη χειροκίνητης σήμανσης από ειδικούς.

Παράλληλα, οι μέθοδοι που βασίζονται σε βαθιά ενσωμάτωση χαρακτηριστικών (deep embeddings) χρησιμοποιούν νευρωνικά δίκτυα για να μετασχηματίσουν τα αρχικά δεδομένα (όπως κώδικας ή δικτυακή κίνηση) σε συμπαγείς, πλούσιες αναπαραστάσεις σε έναν πολυδιάστατο χώρο χαρακτηριστικών. Αυτές οι αναπαραστάσεις επιτρέπουν την καλύτερη ομαδοποίηση και ανίχνευση ανωμαλιών, καθώς και την ταχύτερη προσαρμογή σε νέες ή εξελισσόμενες οικογένειες malware.

Η συνδυασμένη χρήση αυτών των τεχνικών ενισχύει την ικανότητα των συστημάτων ασφάλειας να ανιχνεύουν και να κατηγοριοποιούν αποτελεσματικά το κακόβουλο λογισμικό, ακόμα και όταν αυτό μεταλλάσσεται ή εμφανίζονται νέες απειλές που δεν

έχουν προηγούμενο ιστορικό. Με αυτόν τον τρόπο, η ανίχνευση γίνεται πιο ευέλικτη και ανθεκτική απέναντι στην ταχύτατη εξέλιξη των απειλών.

4.3 Κίνδυνοι και Προκλήσεις

Η χρήση ενός μόνο τύπου ανιχνευτή δεν επαρκεί για την αποτελεσματική προστασία απέναντι σε κακόβουλο λογισμικό. Οι εξελιγμένες τεχνικές αποφυγής ανίχνευσης καθιστούν αναγκαία την ανάπτυξη πολυεπίπεδων μεθόδων, οι οποίες συνδυάζουν στατική και δυναμική ανάλυση, καθώς και τεχνητή νοημοσύνη. Σίγουρα η εκτέλεση ενός αρχείου σε απομακρυσμένο περιβάλλον μπορεί να είναι η πιο αποδοτική λύση για την λειτουργία του, κάτι τέτοιο όταν μιλάμε για εκατομμύρια καινούργιες απειλές κάθε χρόνο καταλαβαίνουμε εύκολα ότι είναι υπολογιστικά πολύ απαιτητικό και ασύμφορο.

Η ανάπτυξη στατικών μεθόδων ανάλυσης ικανών να εντοπίζουν νέες και προηγμένες απειλές χωρίς να απαιτούν την εκτέλεση του αρχείου αποτελεί στρατηγικής σημασίας στόχο. Ένα αποτελεσματικό σύστημα στατικής ανάλυσης μπορεί να λειτουργήσει ως φίλτρο πρώτου επιπέδου, περιορίζοντας το φορτίο των δυναμικών μηχανισμών και βελτιώνοντας τη συνολική αποδοτικότητα του συστήματος ασφαλείας.

Κεφάλαιο 5

Πηγή και Προετοιμασία Δεδομένων

Στόχος αυτής της εργασίας είναι να γίνει δοκιμή πειραμάτων με δεδομένα τα οποία δεν παρουσιάζουν ομοιογένεια. Σίγουρα όπως βλέπουμε και στην βιβλιογραφία που αναλύεται πιο πάνω ένα μοντέλο μηχανικής μάθησης μπορεί να επεξεργαστεί πολύ καλύτερα δεδομένα που παρουσιάζουν σημαντικά χαρακτηριστικά ομοιογένειας και να ομαδοποιήσει πολύ καλύτερα έτσι τα όποια "patterns" συναντάει. Παρόλα αυτά τα κακόβουλα αρχεία που θα χτυπήσουν ένα σύστημα ποικίλουν σε τεχνικές μέγεθος και σκοπό ενώ παράλληλα το ίδιο φαίνεται και στα αρχεία καλόβουλου σκοπού. Παρακάτω αναλύουμε την πηγή, τα στατιστικά και τις κατηγορίες δεδομένων που μαζέψαμε.

5.1 Πηγή

Η επιτυχία ενός μοντέλου ανίχνευσης και ταξινόμησης κακόβουλου λογισμικού δεν εξαρτάται αποκλειστικά από τον αλγόριθμο που χρησιμοποιείται, αλλά επηρεάζεται άμεσα από την ποιότητα και, συχνά, από την ποσότητα των δεδομένων εκπαίδευσής του. Στην παρούσα εργασία, εστιάζουμε στην ανάλυση δεδομένων που προέρχονται από δύο διαφορετικές πηγές: κακόβουλο και καλοήθη λογισμικό.

Δεδομένου ότι ένας από τους βασικούς στόχους μας είναι η ταχύτητα ανίχνευσης κακόβουλου λογισμικού, επιλέγουμε να επικεντρωθούμε αποκλειστικά σε στατικά χαρακτηριστικά. Τα χαρακτηριστικά αυτά μπορούν να εξαχθούν γρήγορα και με τη μέγιστη δυνατή ασφάλεια, χωρίς να απαιτείται η εκτέλεση του κώδικα.

Η αποτελεσματικότητα των μοντέλων εξαρτάται σε μεγάλο βαθμό από την ποικιλία και την αξιοπιστία των δεδομένων εκπαίδευσης. Για τη δημιουργία ισχυρών μοντέλων, είναι ζωτικής σημασίας η συλλογή και η ανάλυση δεδομένων τόσο από κακόβουλες όσο και από καλοήθεις πηγές λογισμικού. Τα κακόβουλα δεδομένα παρέχουν παραδείγματα κακόβουλης συμπεριφοράς, βοηθώντας το μοντέλο να αναγνωρίζει και να ταξινομεί νέες απειλές. Αντίστοιχα, τα καλοήθη δεδομένα επιτρέπουν στο μοντέλο να διακρίνει το νόμιμο λογισμικό από το κακόβουλο, μειώνοντας τα ψευδώς θετικά αποτελέσματα και βελτιώνοντας τη συνολική του ακρίβεια.

5.2 Δεδομένα

5.2.1 Συλλογή Δεδομένων Κακόβουλου Λογισμικού

Για την εκπαίδευση και αξιολόγηση των μοντέλων μας, αντλήσαμε δεδομένα και πληροφορίες για αυτά από δύο βασικές πηγές: το VirusShare και το VirusTotal. Αυτές οι πλατφόρμες αποτελούν πολύτιμες πηγές δειγμάτων κακόβουλου λογισμικού, επιτρέποντας στους ερευνητές να μελετήσουν και να ταξινομήσουν απειλές με βάση τα χαρακτηριστικά τους.

Το VirusShare [52] είναι ένα αποθετήριο δειγμάτων κακόβουλου λογισμικού, σχεδιασμένο να διευκολύνει την έρευνα στον τομέα της κυβερνοασφάλειας. Η πρόσβαση στη βάση δεδομένων του γίνεται αποκλειστικά μέσω προσκλήσεων από τους διαχειριστές, καθώς περιέχει ενεργά δείγματα κακόβουλου λογισμικού. Ο κύριος στόχος του VirusShare είναι να παρέχει σε ερευνητές και επαγγελματίες ασφαλείας μια κεντρική πλατφόρμα συλλογής και ανταλλαγής δεδομένων, συμβάλλοντας έτσι στην ανίχνευση και καταπολέμηση νέων απειλών.

Αντί να απαιτείται η λήψη κάθε δείγματος ξεχωριστά, οι διαχειριστές της πλατφόρμας συγκεντρώνουν και δημοσιεύουν μεγάλα πακέτα δεδομένων που περιλαμβάνουν χιλιάδες δείγματα κακόβουλου λογισμικού. Αυτή η διαδικασία επιτρέπει στους ερευνητές να αποκτήσουν μαζικά δεδομένα, μειώνοντας τον χρόνο συλλογής και επεξεργασίας.

Στην παρούσα εργασία, αξιοποιήσαμε τέσσερα πακέτα δεδομένων που έχουν συλλεχθεί με κάποια χρονολογική διαφορά για μεγαλύτερη ποικιλομορφία και που περιείχαν συνολικά 30.000 δείγματα κακόβουλου λογισμικού αφού εφαρμόσαμε φιλτράρισμα για την απομάκρυνση όλων των δειγμάτων που δεν αφορούσαν το λειτουργικό σύστημα Windows.

Το VirusTotal [53] από την άλλη είναι μια online υπηρεσία ανάλυσης κακόβουλου λογισμικού που επιτρέπει στους χρήστες να ανεβάζουν ύποπτα αρχεία ή διευθύνσεις URL προκειμένου να σαρωθούν από πολλαπλές μηχανές ανίχνευσης ιών και απειλών. Η πλατφόρμα συγκεντρώνει αποτελέσματα από διαφορετικά antivirus και μηχανές ανίχνευσης, παρέχοντας μια ολοκληρωμένη εικόνα σχετικά με την επικινδυνότητα ενός αρχείου ή μιας διαδικτυακής διεύθυνσης.

Στη μελέτη μας, χρησιμοποιήσαμε το VirusTotal για να αξιολογήσουμε την εγκυρότητα των δειγμάτων που αντλήσαμε από το VirusShare. Συγκεκριμένα, υποβάλαμε τα κακόβουλα δείγματα στην πλατφόρμα και αναλύσαμε τις αναφορές των διαφόρων μηχανών ανίχνευσης ιών. Αυτή η διαδικασία μας επέτρεψε να επιβεβαιώσουμε την ταυτοποίηση των κακόβουλων αρχείων και να αποκλείσουμε τυχόν ψευδώς θετικά δείγματα που θα μπορούσαν να επηρεάσουν την ακρίβεια των μοντέλων μας.

5.2.2 Συλλογή Δεδομένων Καλόβουλου Λογισμικού

Για τη δημιουργία ενός αξιόπιστου συνόλου δεδομένων καλοήθους λογισμικού, πραγματοποιήσαμε συλλογή εκτελέσιμων αρχείων από εκδόσεις του λειτουργικού συστήματος *Windows*. Η επιλογή αυτών των εκδόσεων βασίστηκε στην εκτεταμένη χρήση τους και στη διαφοροποίηση των εκτελέσιμων αρχείων τους, γεγονός που μας επέτρεψε να καλύψουμε ένα ευρύ φάσμα εφαρμογών και συστημικών διεργασιών.

Η συλλογή των δεδομένων έγινε σε δύο στάδια:

1. **Βασικά εκτελέσιμα αρχεία του λειτουργικού συστήματος Windows:** Καταγράψαμε τα εκτελέσιμα αρχεία που περιλαμβάνονται στις προ εγκα-

τεστημένες εφαρμογές και υπηρεσίες. Αυτά τα αρχεία αποτελούν τον πυρήνα του συστήματος και περιλαμβάνουν εφαρμογές, βοηθητικά προγράμματα και κρίσιμες διαδικασίες που είναι απαραίτητες για τη λειτουργία του λειτουργικού συστήματος.

2. **Εκτελέσιμα αρχεία από αξιόπιστες πηγές:** Συλλέξαμε επιπλέον εκτελέσιμα αρχεία που προέρχονται από επίσημες και αξιόπιστες πηγές λογισμικού. Αυτά τα αρχεία λήφθηκαν από δημοφιλείς πλατφόρμες διανομής λογισμικού, επίσημες ιστοσελίδες κατασκευαστών και προγράμματα που έχουν επαληθευτεί ως ασφαλή από τα ενσωματωμένα συστήματα προστασίας των Windows.

3. **Εκτελέσιμα αρχεία από διαφορετικό λειτουργικό Windows:**Αν και η μορφή των PE files είναι αρκετά κοινή γίνεται αρκετά διαφορετική χρήση δυναμικών βιβλιοθηκών DLLs, API calls, καθώς και η υλοποίηση διαφορετικών μηχανισμών φόρτωσης ή προστασίας μνήμης. Ως αποτέλεσμα παρουσιάζονται διαφορετικά byte patterns ή δομές, που πιθανά να δυσχεραίνουν την γενίκευση και την ακρίβεια των μοντέλων ανίχνευσης.Για τον λόγο αυτό η χρήση τους έγινε μόνο σε ένα από τα τρία πειράματά μας ώστε να δούμε και το αποτέλεσμα αυτής της επιλογής.

Όλα τα εκτελέσιμα αρχεία που συμπεριλήφθηκαν στο σύνολο δεδομένων υποβλήθηκαν σε έλεγχο ασφαλείας για να διασφαλιστεί η απουσία κακόβουλου κώδικα. Η επαλήθευση πραγματοποιήθηκε μέσω των ενσωματωμένων μηχανισμών ασφαλείας του *Windows Defender*

Η συλλογή και ανάλυση αυτών των δεδομένων μάς επιτρέπει να εκπαιδεύσουμε το μοντέλο ανίχνευσης κακόβουλου λογισμικού με επαρκή παραδείγματα νόμιμου λογισμικού, βελτιώνοντας έτσι την ικανότητά του να διακρίνει ανάμεσα σε κακόβουλες και καλοήθεις εφαρμογές, μειώνοντας τα ψευδώς θετικά αποτελέσματα.

5.3 Στατιστικά των Δεδομένων

Βασικό χαρακτηριστικό των εκτελέσιμων αρχείων είναι η έντονη ετερογένεια που παρουσιάζουν τόσο ως προς το μέγεθός τους όσο και ως προς τη συμπεριφορά τους κατά την εκτέλεση στο σύστημα. Αυτή η ποικιλομορφία από τη μία μπορεί να λειτουργήσει θετικά, καθώς προσφέρει πολύτιμες ενδείξεις για τη διάκριση μεταξύ καλοήθους και κακόβουλου λογισμικού καθώς συχνά, τα επιμέρους χαρακτηριστικά ενός εκτελέσιμου αρχείου αποκαλύπτουν τις προθέσεις του. Από την άλλη, η ίδια αυτή ετερογένεια θέτει σημαντικές προκλήσεις για την ανάπτυξη λογισμικού ανάλυσης, καθώς καθιστά δύσκολη την ομοιόμορφη επεξεργασία τους και τον εντοπισμό κοινών συνδετικών χαρακτηριστικών που θα επέτρεπαν μια αξιόπιστη και αυτοματοποιημένη ταξινόμηση ή ανίχνευση.

5.3.1 Μέγεθος Εκτελέσιμων

Στους παρακάτω πίνακες βλέπουμε αναλυτικά το εύρος από τα μεγέθη στα εκτελέσιμα, ενώ στην συνέχεια θα δούμε και χαρακτηριστικά ως προς τη λειτουργία τους:

Size Statistics

Table 5.1: Size Statistics Summary of Malware

Size Range (KB)	Dataset 346	Dataset 434	Dataset 375	Dataset 401
Less than 10	55	77	4660	77
10-30	49	590	410	590
30-60	46	296	923	296
60-100	96	251	1029	251
100-200	129	643	4124	643
200-500	227	5150	2940	5150
500-1000	771	2032	1853	2032
Above 1000	1230	4709	3010	4709

Table 5.2: Size Statistics Summary of Benign

Size Range (KB)	Benign 1	Benign 2	Benign 3
Less than 10	55	321	257
10-30	49	648	955
30-60	46	645	728
60-100	96	488	608
100-200	129	608	827
200-500	227	633	956
500-1000	771	290	517
Above 1000	1230	373	640

Size Statistics

Εύκολα παρατηρούμε ότι υπάρχει αναντιστοιχία στα μεγέθη των εκτελέσιμων αρχείων, τόσο μεταξύ διαφορετικών datasets όσο και μεταξύ κακόβουλων και καθαρών δειγμάτων. Η ποικιλία αυτή επηρεάζει την απόδοση των μοντέλων μηχανικής μάθησης, καθώς αν και δεν βασίζονται σε χαρακτηριστικά που σχετίζονται με το μέγεθος οι εικόνες που θα παραχθούν καθώς θα είναι όλες ίδιου μεγέθους, επηρεάζεται ο τρόπος που φτιάχνονται. Η περαιτέρω ανάλυση της κατανομής των μεγεθών μπορεί να αποκαλύψει μοτίβα που συμβάλλουν στην πιο αποτελεσματική διάκριση μεταξύ malware και benign εκτελέσιμων.

Παρ όλες τις δυσκολίες που δημιουργεί αυτή η ανομοιογένεια, στην παρούσα εργασία επιδιώκουμε να αξιολογήσουμε την απόδοση των μοντέλων μας και υπό αυτές τις συνθήκες. Συγκεκριμένα, θέλουμε να διαπιστώσουμε κατά πόσο μπορούν να ανταποκριθούν όταν το dataset δεν είναι ειδικά διαμορφωμένο για να εξυπηρετεί τις ανάγκες τους, αλλά αντίθετα αντικατοπτρίζει μια πιο ρεαλιστική και ποικιλόμορφη κατάσταση.

5.3.2 Κατηγοριοποίηση Κακόβουλου Λογισμικού

Για την κατηγοριοποίηση των δεδομένων, χρησιμοποιήθηκε η πλατφόρμα VirusTotal. Τα στοιχεία της πλατφόρμας προσφέρουν πολύτιμες πληροφορίες σχετικά με το είδος του αρχείου και την ανάλυσή του για την ανίχνευση κακόβουλου λογισμικού. Η

πλατφόρμα VirusTotal παρέχει σημαντικές πληροφορίες που αφορούν τα αρχεία που υποβάλλονται σε ανάλυση και αποδεικνύεται χρήσιμη για την κατηγοριοποίηση και την ανίχνευση κακόβουλου λογισμικού. Ακολουθεί μια αναλυτική περιγραφή των πεδίων που παρατίθενται στην ανάλυση της πλατφόρμας:

Name: Το όνομα του αρχείου που αναλύεται. Αυτό το πεδίο είναι χρήσιμο για την ταυτοποίηση και την ανίχνευση του αρχείου στην πλατφόρμα ή στο σύστημά σας.

Type: Ο τύπος του αρχείου, όπως π.χ. Win32 EXE, Win32 DLL, κ.λπ. Αυτό προσδιορίζει αν το αρχείο είναι εκτελέσιμο ή βιβλιοθήκη, παρέχοντας αρχικές ενδείξεις για το είδος του.

File type: Μια λίστα με υποκατηγορίες του τύπου του αρχείου, η οποία μπορεί να περιλαμβάνει ειδικά χαρακτηριστικά του αρχείου ή το εργαλείο που χρησιμοποιήθηκε για την δημιουργία του.

Times Submitted: Ο αριθμός των φορών που το συγκεκριμένο αρχείο έχει υποβληθεί στην πλατφόρμα VirusTotal. Ένας υψηλότερος αριθμός υποβολών μπορεί να υποδεικνύει ότι το αρχείο είναι γνωστό και έχει αναλυθεί πολλές φορές, καθιστώντας το πιο εύκολα εντοπίσιμο ως κακόβουλο.

Library Name: Η λίστα των βιβλιοθηκών που χρησιμοποιούνται από το αρχείο. Αυτή η πληροφορία μπορεί να παράσχει στοιχεία για τη λειτουργία του αρχείου και τις εξαρτήσεις του.

Malicious: Ο αριθμός των antivirus λογισμικών που αναγνώρισαν το αρχείο ως κακόβουλο. Αυτό το πεδίο είναι ζωτικής σημασίας για την αξιολόγηση της επικινδυνότητας του αρχείου.

Undetected: Ο αριθμός των antivirus λογισμικών που δεν έχουν αναγνωρίσει το αρχείο ως κακόβουλο. Ένα υψηλό ποσοστό στην κατηγορία αυτή μπορεί να υποδηλώνει ότι το αρχείο είναι καινούργιο ή οι μέθοδοι ανίχνευσης δεν έχουν εντοπίσει ακόμη το κακόβουλο περιεχόμενό του.

Kaspersky result, BitDefender result, Avast result, ESET-NOD32 result: Αυτά τα πεδία παρέχουν τα αποτελέσματα ανίχνευσης από συγκεκριμένα antivirus λογισμικά, όπως οι πλατφόρμες Kaspersky, BitDefender, Avast, και ESET-NOD32. Αν ένα από αυτά τα αποτελέσματα δείχνει κακόβουλο ή ύποπτο χαρακτήρα για το αρχείο, ενισχύει την πιθανότητα του να είναι επιβλαβές.

Creation Date: Η ημερομηνία δημιουργίας του αρχείου. Αυτό το πεδίο είναι χρήσιμο για να κατανοήσουμε πότε δημιουργήθηκε το αρχείο, καθιστώντας δυνατό να προσδιορίσουμε αν πρόκειται για νέο ή παλαιότερο αρχείο.

Η κατηγοριοποίηση των κακόβουλων αρχείων είναι μια διαδικασία που πολλές φορές αποδεικνύεται δύσκολη, καθώς τα κακόβουλα αρχεία μπορεί να εντάσσονται σε περισσότερες από μία κατηγορίες, ανάλογα με τις λειτουργίες και τις συμπεριφορές τους. Ένα αρχείο, για παράδειγμα, μπορεί να ανήκει ταυτόχρονα σε κατηγορίες όπως "Trojan", "Adware" "Ransomware", γεγονός που καθιστά τη διαδικασία κατηγοριοποίησης πιο περίπλοκη.

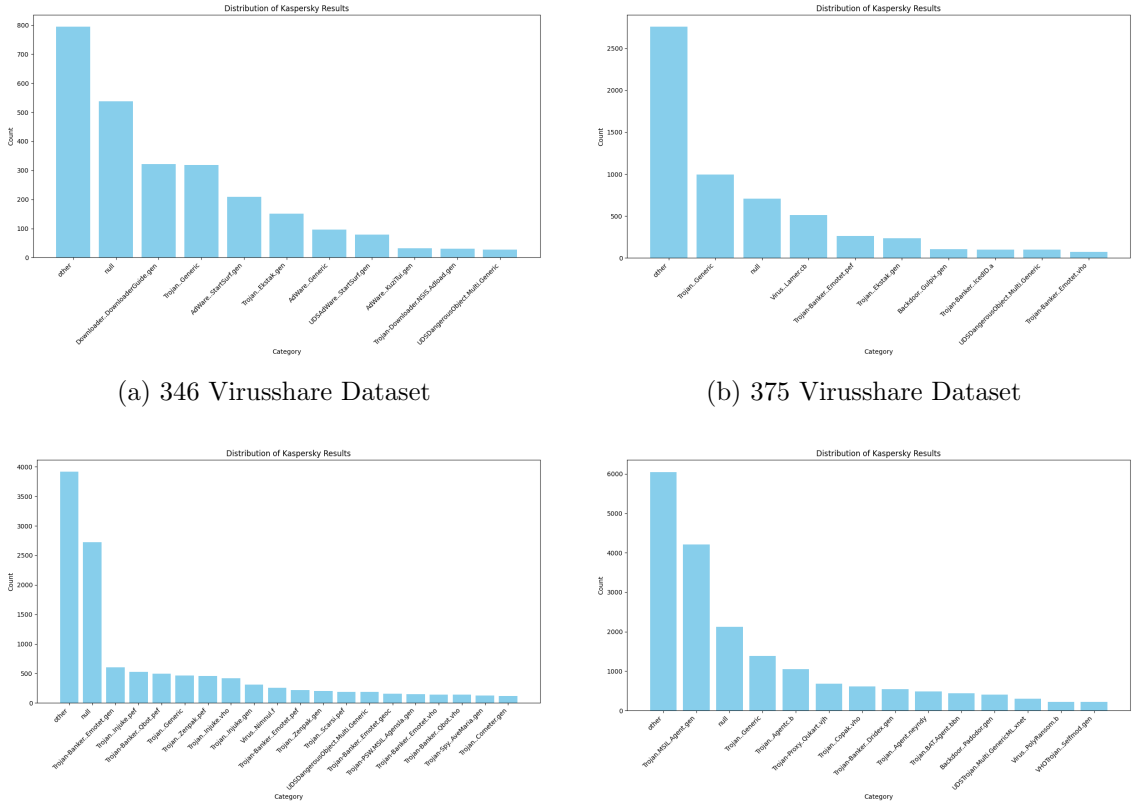
Επιπλέον, παρατηρήσαμε και στα δεδομένα της πλατφόρμας VirusTotal όπου εμφανίζονταν κακόβουλα αρχεία τα οποία είχαν διαφορετικές κατηγοριοποιήσεις ανάλογα με τις πλατφόρμες antivirus που τα ανέλυσαν ενώ πάρα πολλά δείγματα δεν εντοπίζονταν ως κακόβουλα από όλες τις πλατφόρμες. Αυτή η ασυνέπεια οφείλεται στις διαφορετικές μεθόδους ανίχνευσης και αξιολόγησης που χρησιμοποιούν οι διάφορες πλατφόρμες, γεγονός που οδηγεί σε διαφορετικά αποτελέσματα για το ίδιο αρχείο.

Για αυτόν τον λόγο, αποφασίσαμε να επιλέξουμε την κατηγοριοποίηση μιας μόνο αξιόπιστης πλατφόρμας για όλα τα πειράματά μας. Αυτή η προσέγγιση εξασφαλίζει ότι

υπάρχει μια κοινή βάση αναφοράς για τα δεδομένα μας, διευκολύνοντας τη διαδικασία ανάλυσης και εξαγωγής πιο αξιόπιστων και συνεπών αποτελεσμάτων σχετικά με την κατηγοριοποίηση των κακόβουλων αρχείων.

Στα παρακάτω διαγράμματα παρατηρούμε την κατανομή των malware κατηγοριών και φαίνεται εύκολα ότι πολλά αρχεία φέρουν την ένδειξη null ή other. Στην περίπτωση του null, πρόκειται για αρχεία των οποίων η κακόβουλη λειτουργία είχε επιβεβαιωθεί από διάφορες πηγές της πλατφόρμας VirusTotal, αλλά δεν είχαν εντοπιστεί από το antivirus που χρησιμοποιήσαμε. Αντίστοιχα, η κατηγορία others περιλαμβάνει δείγματα που εμφανίζονται σπάνια στο σύνολο των δεδομένων, και δεν ανήκουν σε κάποια από τις κύριες κατηγορίες.

Results



(a) 346 Virusshare Dataset

(b) 375 Virusshare Dataset

(c) 401 Virusshare Dataset

(d) 434 Virusshare Dataset

Figure 5.1: Κατανομή των δειγμάτων ανά κατηγορία για τέσσερα σύνολα δεδομένων του Virusshare.

5.3.3 Κατηγοριοποίηση Καλόβουλου Λογισμικού

Σε αντίθεση με το κακόβουλο λογισμικό, του οποίου η κατηγοριοποίηση βασίζεται κυρίως στις δράσεις του και τη ζημιογόνο συμπεριφορά του (κλοπή δεδομένων, εγκατάσταση εφαρμογών, κρυπτογράφηση αρχείων κ.ο.κ.), το καλόβουλο λογισμικό δεν υπόκειται σε μια τέτοια ταξινόμηση. Αντιθέτως, η κατηγοριοποίησή του βασίζεται κυρίως στη λειτουργικότητα (π.χ. προγράμματα επεξεργασίας κειμένου, λογιστικά πακέτα, προγράμματα επικοινωνίας κ.ο.κ), στην πλατφόρμα για την οποία έχει αναπτυχθεί (π.χ. Windows, Linux, Android), καθώς και στα χαρακτηριστικά αναπτυξής

του, όπως η γλώσσα προγραμματισμού, η χρήση συγκεκριμένων βιβλιοθηκών, ή το framework ανάπτυξης. Επιπλέον, στο πλαίσιο πειραμάτων μηχανικής μάθησης και ταξινόμησης εκτελέσιμων αρχείων, τα καλόβουλα δείγματα χρησιμοποιούνται κυρίως ως σημείο αναφοράς σε σχέση με τα κακόβουλα δείγματα, χωρίς να απαιτείται περαιτέρω υποκατηγοριοποίηση. Συνεπώς, το καλόβουλο λογισμικό δεν αποτελεί αντικείμενο αναλυτικής κατηγοριοποίησης σε αυτήν την εργασία, καθώς η ουσία της ανάλυσης επικεντρώνεται στον εντοπισμό, διαφοροποίηση και κατανόηση της συμπεριφοράς του κακόβουλου λογισμικού.

Η ανάλυση λοιπόν αυτών των στοιχείων μπορεί να βοηθήσει στην αποτελεσματική κατηγοριοποίηση των αρχείων και στην αξιολόγηση του βαθμού επικινδυνότητάς τους. Οι πληροφορίες αυτές είναι χρήσιμες για να αποφασιστεί εάν απαιτείται η λήψη μέτρων, όπως η διαγραφή ή η καραντίνα του αρχείου, προκειμένου να προστατευτεί το σύστημα από ενδεχόμενο κίνδυνο.

Σχολιασμός Κακόβουλων Δειγμάτων

346

Το VirusShare 346 dataset, αφού φιλτραρίστηκε ώστε να περιλαμβάνει αποκλειστικά εκτελέσιμα αρχεία που σχετίζονται με το λειτουργικό σύστημα Windows, περιλαμβάνει συνολικά 2.649 αρχεία. Η ανάλυση των αποτελεσμάτων από την Kaspersky δείχνει ότι σημαντικό ποσοστό κατέχουν οι κατηγορίες κακόβουλου λογισμικού Downloader-Guide.gen (12,36%) και Trojan.Generic (12,28%).Η κατηγορία "other" καταλαμβάνει το 30,60% και αποτελεί δείγματα που εμφανίζονταν σε λιγότερο απο 1% του συνολικού δείγματος, ακολουθούμενη από την κατηγορία "null" με 20,71% και πρόκειται για δείγματα τα οποία καταφέρνανε να διαφύγουν από μεγάλο κομμάτι των ανιχνευτών όπως και της Kaspersky που χρησιμοποιούμαι αλλά είναι γνωστή η κακόβουλη λειτουργία τους.Η κατανομή έχει μεγάλη ποικιλία καθιστώντας το κατάλληλο για την ανάπτυξη και αξιολόγηση αποτελεσματικών μεθόδων ανίχνευσης

Πίνακας 5.3: Κατανομή κατηγοριών στο `VirusShare 346` dataset μετά το φιλτράρισμα για εκτελέσιμα `Windows` αρχεία

Κατηγορία	**Πλήθος Αρχείων**	**Ποσοστό (%)**
other	795	30.60
null	538	20.71
Downloader..DownloaderGuide.gen	321	12.36
Trojan..Generic	319	12.28
AdWare..StartSurf.gen	209	8.04
Trojan..Ekstak.gen	151	5.81
AdWare..Generic	96	3.70
UDSAdWare..StartSurf.gen	79	3.04
AdWare..KuziTui.gen	32	1.23
Trojan-Downloader.NSIS.Adload.gen	31	1.19
UDSDangerousObject.Multi.Generic	27	1.04

375

Στο σύνολο των αποτελεσμάτων από την Kaspersky, το dataset 375 περιλαμβάνει συνολικά 5.841 αρχεία, εκ των οποίων το 47.26% κατηγοριοποιούνται ως "other". Η πιο συχνή συγκεκριμένη κατηγορία είναι η "Trojan..Generic" με 17.07%, ακολουθούμενη από την κατηγορία "null" με 12.11%. Αρκετά σημαντική παρουσία έχουν επίσης οι κατηγορίες "Virus..Lamer.cb" και "Trojan-Banker..Emotet.pef" με 8.72% και 4.49% αντίστοιχα.

Πίνακας 5.4: Κατανομή κατηγοριών στο dataset από τα αποτελέσματα του Kaspersky

Κατηγορία	**Πλήθος Αρχείων**	**Ποσοστό (%)**
other	2760	47.26
Trojan..Generic	997	17.07
null	707	12.11
Virus..Lamer.cb	509	8.72
Trojan-Banker..Emotet.pef	262	4.49
Trojan..Ekstak.gen	233	3.99
Backdoor..Gulpix.gen	103	1.76
Trojan-Banker..IcedID.a	100	1.71
UDSDangerousObject.Multi.Generic	98	1.68
Trojan-Banker..Emotet.vho	71	1.22

401

Στο συγκεκριμένο σύνολο δεδομένων μετά το φιλτράρισμα έχει 11.810 αρχεία, με το 33.21% να ανήκουν στην κατηγορία "other" και το 23.09% να παραμένει μη ταξινομημένο ("null"). Οι πιο διαδεδομένες απειλές περιλαμβάνουν αρκετές παραλλαγές του Trojan-Banker, με την πιο συχνή να είναι η "Trojan-Banker..Emotet.gen" στο 5.13%. Πρόκειται για ένα σύνολο με μεγάλη ποικιλία κακόβουλου λογισμικού και πολλαπλές παραλλαγές των Trojans.

Πίνακας 5.5: Κατανομή κατηγοριών στο dataset από τα αποτελέσματα του Kaspersky

Κατηγορία	**Πλήθος Αρχείων**	**Ποσοστό (%)**
other	3919	33.21
null	2725	23.09
Trojan-Banker..Emotet.gen	605	5.13
Trojan..Injuke.pef	529	4.48
Trojan-Banker..Qbot.pef	492	4.17
Trojan..Generic	464	3.93
Trojan..Zenpak.pef	459	3.89
Trojan..Injuke.vho	418	3.54
Trojan..Injuke.gen	311	2.64
Virus..Nimnul.f	260	2.20
Trojan-Banker..Emotet.pef	214	1.81
Trojan..Zenpak.gen	203	1.72
Trojan..Scarsi.pef	187	1.58
UDSDangerousObject.Multi.Generic	185	1.57
Trojan-Banker..Emotet.geoc	155	1.31
Trojan-PSW.MSIL.Agensla.gen	151	1.28
Trojan-Banker..Emotet.vho	143	1.21
Trojan-Banker..Qbot.vho	139	1.18
Trojan-Spy..AveMaria.gen	124	1.05
Trojan..Cometer.gen	119	1.01

434

Στο φιλτραρισμένο σύνολο δεδομένων του 434 παρατηρούμε ότι η κατηγορία "Trojan.MSIL.Agent.gen" αποτελεί το 27.14% των αρχείων, καθιστώντας την την πιο συχνή απειλή στο δείγμα. Το 30.35% των αρχείων ανήκει στην κατηγορία "other", ενώ το 10.01% παραμένει μη ταξινομημένο ("null"). Επιπλέον, σημαντικές κατηγορίες όπως "Trojan..Generic", "Trojan-Proxy..Qukart.vjh" και "Trojan..Copak.vho" συγκεντρώνουν επίσης σημαντικά ποσοστά.

Πίνακας 5.6: Φιλτραρισμένη κατανομή κατηγοριών στο dataset από τα αποτελέσματα του Kaspersky

Κατηγορία	**Πλήθος Αρχείων**	**Ποσοστό (%)**
Trojan.MSIL.Agent.gen	4208	27.14
null	1553	10.01
Trojan..Generic	1263	8.14
Trojan-Proxy..Qukart.vjh	685	4.42
Trojan..Copak.vho	609	3.93
Trojan..Agent.neyndy	484	3.12
Trojan.BAT.Agent.bbn	439	2.83
Backdoor..Padodor.gen	401	2.59
Trojan-Banker..Dridex.gen	218	1.41
Virus..PolyRansom.b	217	1.40
VHOTrojan..Selfmod.gen	214	1.38
Trojan-PSW.MSIL.Coins.gen	176	1.13
Trojan..SelfDel.pef	175	1.13
Trojan..Copak.pef	158	1.02
other	4707	30.35

Τα τέσσερα datasets παρουσιάζουν σημαντικές διαφορές στις κατηγορίες malware, γεγονός που ανταναικλά την ποικιλία και την πολυπλοκότητα των απειλών που υπάρχουν. Στο πρώτο dataset(VirusShare346), η παρουσία κατηγοριών όπως το "other" και το "null" είναι αρκετά σημαντική, αλλά ταυτόχρονα διακρίνονται κυρίαρχες κατηγορίες.

Στο δεύτερο και τρίτο dataset(VirusShare 375,401) παρατηρούμε μεγαλύτερη εστίαση σε συγκεκριμένα οικογένειες Trojan, όπως οι Trojan-Banker τύποι (π.χ. Emotet, Qbot) και επιπλέον μεγάλη παρουσία κατηγοριών "null" και "other", που μπορεί να οφείλεται σε μη ταξινομημένα ή σπάνια κακόβουλα δείγματα. Αυτό δείχνει ότι αυτά τα datasets περιέχουν πιο εξειδικευμένες και συχνά πιο επικίνδυνες απειλές.

Το τέταρτο dataset(VirusShare 434) είναι το πιο ισορροπημένο και εκτεταμένο, με έντονη παρουσία της κατηγορίας "Trojan.MSIL.Agent.gen" που καλύπτει πάνω από το ένα τέταρτο του δείγματος, ενώ ταυτόχρονα διαθέτει και μια σημαντική ποσοστιαία αναλογία στην κατηγορία "other". Η πολυπλοκότητα ενισχύεται από την παρουσία πολλών υποκατηγοριών Trojan και backdoor, γεγονός που το καθιστά κατάλληλο για πιο λεπτομερείς αναλύσεις και αξιολόγηση άλλων δικτύων.

Καλόβουλο Λογισμικό

Όσο αφορά το καλόβουλο λογισμικό έχει για ευκολία ονομάζουμε Benign 1, Benign 2, Benign 3 και συνδυάστηκαν με το VirusShare(346,375,401) αντίστοιχα. Το Benign 1 έχει ένα 30% αρχεία από το λειτουργικό των windows 7 και τα υπόλοιπα είναι από το σύστημα των Windows 10 και ένα μεγάλο(20%) από αρχεία κατεβασμένα από το διαδίκτυο.Τα άλλα δυο (Benign 2,Benign 3) έχουν κατά κύριο λόγο αρχεία από το σύστημα των Windows 10 και σε ένα ποσοστό περίπου 10% κατεβασμένα από το διαδίκτυο.

5.3.4 Χαρακτηριστικά Εκτελέσιμων

Ένα βασικό χαρακτηριστικό αλλά και πρόβλημα στην ανίχνευση κακόβουλου λογισμικού ειδικά στην συγκεκριμένη εργασία είναι η τεράστια ποικιλομορφία των κακόβουλων αρχείων. Τα κακόβουλα λογισμικά διαφέρουν σημαντικά μεταξύ τους ως προς το μέγεθος, τον σκοπό και τον τρόπο λειτουργίας τους, γεγονός που δυσκολεύει τον εντοπισμό τους με παραδοσιακές μεθόδους. Δεν ακολουθούν ενιαία μορφή ή πρότυπα, ενώ συχνά χρησιμοποιούν τεχνικές όπως η απόκρυψη, η κωδικοποίηση ή η δυναμική τροποποίηση του κώδικα (obfuscation, polymorphism) για να αποφύγουν τον εντοπισμό. Αυτή η ετερογένεια καθιστά την ανίχνευση νέων ή εξελιγμένων απειλών ιδιαίτερα απαιτητική και αναδεικνύει την ανάγκη για πιο εξελιγμένες μεθόδους όπως η ανάλυση συμπεριφοράς και η χρήση τεχνητής νοημοσύνης.

Αυτά τα χαρακτηριστικά εντοπίζονται και στο δικό μας dataset, το οποίο περιλαμβάνει δείγματα με μεγάλη ποικιλομορφία ως προς το μέγεθος, τη λειτουργικότητα και τη συμπεριφορά. Στο πλαίσιο αυτό, στόχος μας είναι να διερευνήσουμε την απόδοση των μοντέλων ανίχνευσης κακόβουλου λογισμικού υπό αυτές τις απαιτητικές συνθήκες. Η ύπαρξη τόσο ανομοιογενών δειγμάτων μάς επιτρέπει να αξιολογήσουμε την ικανότητα των μοντέλων να γενικεύουν και να εντοπίζουν άγνωστες ή εξελιγμένες απειλές, όπως αυτές που εμφανίζονται σε πραγματικά περιβάλλοντα.

5.4 Δημιουργία Εικόνων από Εκτελέσιμα Αρχεία

Η δημιουργία εικόνων από εκτελέσιμα αρχεία είναι μια σημαντική τεχνική στην ανάλυση κακόβουλου λογισμικού και την ταξινόμηση λογισμικού. Αυτή η διαδικασία μετατρέπει τα δυαδικά δεδομένα των εκτελέσιμων αρχείων σε οπτικές αναπαραστάσεις, οι οποίες μπορούν να αναλυθούν με τεχνικές μηχανικής μάθησης. Παρακάτω παρουσιάζονται οι κύριοι τρόποι δημιουργίας εικόνων από εκτελέσιμα αρχεία:

5.4.1 Βασικές Μέθοδοι

Opcode

Οπως βλέπουμε και στην εργασία των Zhang et al [54] οι opcodes (κωδικοί λειτουργίας) είναι οι εντολές που εκτελεί η CPU. Σε αυτή τη μέθοδο, εξάγονται οι opcodes από το εκτελέσιμο αρχείο και μετατρέπονται σε μια ακολουθία αριθμών, όπου κάθε opcode αντιστοιχεί σε έναν μοναδικό αριθμό. Η ακολουθία των αριθμών μετατρέπεται σε μια εικόνα, όπου κάθε pixel αντιπροσωπεύει έναν opcode. Αυτή η μέθοδος εστιάζει στη λειτουργική συμπεριφορά του κώδικα και είναι χρήσιμη για την ανίχνευση κακόβουλου λογισμικού που χρησιμοποιεί πολυμορφικές ή μεταμορφικές τεχνικές.

Raw-bytes

Σε αυτή τη μέθοδο όπως περιγράφετε και από Nataraj et al [31], τα δυαδικά δεδομένα του εκτελέσιμου αρχείου αντιμετωπίζονται ως μια ακολουθία bytes. Κάθε byte μετατρέπεται σε ένα pixel με ένταση από 0 έως 255 (κλίμακα του γκρι) και η εικόνα δημιουργείται διατηρώντας τη σειρά των bytes. Αυτή η μέθοδος διατηρεί τη δομή του αρχείου όπως είναι στον δίσκο και είναι απλή και γρήγορη στη δημιουργία εικόνων.

Hex Dump

Το εκτελέσιμο αρχείο μετατρέπεται σε δεκαεξαδική μορφή (hex dump) και κάθε δεκαεξαδικό ψηφίο αντιστοιχεί σε ένα pixel σε μια εικόνα. Αυτή η μέθοδος διατηρεί όλες τις πληροφορίες του αρχείου και είναι χρήσιμη για την οπτική ανάλυση του δυαδικού περιεχομένου.

Section-based Visualization

Σε αυτή τη μέθοδο, το εκτελέσιμο αρχείο χωρίζεται σε τμήματα (π.χ., .text, .data, .rdata) κλπ όπως αναλύονται στην δομή του PE παραπάνω και κάθε τμήμα μετατρέπεται σε εικόνα ξεχωριστά. Αυτή η μέθοδος εστιάζει σε συγκεκριμένα τμήματα του αρχείου που μπορεί να περιέχουν κακόβουλο κώδικα.

Frequency-based Visualization

Εδώ υπολογίζεται η συχνότητα εμφάνισης κάθε byte ή opcode και οι συχνότητες χαρτογραφούνται σε μια εικόνα, όπου κάθε pixel αντιπροσωπεύει τη συχνότητα ενός byte ή opcode. Αυτή η μέθοδος είναι χρήσιμη για την ανίχνευση μοτίβων και ανωμαλιών.

5.4.2 Μεθοδολογία Δημιουργίας Εικόνων που Ακολουθήθηκε

Στην παρούσα εργασία όπως περιγράφετε και από πάνω Nataraj et al [31],μετατρέψαμε τα δυαδικά δεδομένα του εκτελέσιμου αρχείου σε γκρι εικόνες.Η φαινομενικά απλή διαδικασία μετατροπής ενός εκτελέσιμου αρχείου σε εικόνα γκρι κλίμακας κρύβει σημαντικές λεπτομέρειες. Τα εκτελέσιμα είναι μονοδιάστατα δεδομένα (μια γραμμική ακολουθία bytes), ενώ οι εικόνες είναι δισδιάστατες δομές με εγγενείς χωρικές συσχετίσεις. Αυτό σημαίνει πως η επιλογή του πλάτους κατά τη διαδικασία δημιουργίας της εικόνας είναι κρίσιμη, καθώς μπορεί να εισάγει τεχνητές συσχετίσεις που δεν υπάρχουν στο πρωτογενές δυαδικό περιεχόμενο αλλά και να κρύψει πληροφορίες αν δεν είναι όλα φτιαγμένα με ένα εννοιολογικό κρητήριο.

Ακολουθώντας την προσέγγιση των Nataraj et al., καθορίσαμε το πλάτος της εικόνας ανάλογα με το συνολικό μέγεθος του εκτελέσιμου αρχείου. Με αυτόν τον τρόπο, επιτυγχάνεται ένας λογικός διαχωρισμός των εκτελέσιμων σε ομάδες παρόμοιου μεγέθους, με αποτέλεσμα οι εικόνες που προκύπτουν να παρουσιάζουν παρόμοια χαρακτηριστικά και συσχετίσεις εντός της κάθε ομάδας. Με αυτόν τον τρόπο διατηρείται η συνοχή μεταξύ των δειγμάτων και μειώνεται η πιθανότητα παραμορφώσεων που θα μπορούσαν να επηρεάσουν αρνητικά την ακρίβεια του ανιχνευτή μας.

Παρακάτω παρουσιάζετε μία από τις εικόνες που δημιουργήθηκαν κατά την αρχική μετατροπή των δειγμάτων:

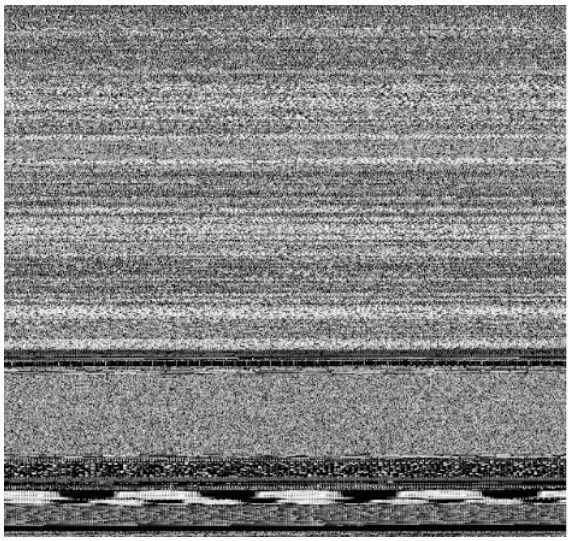


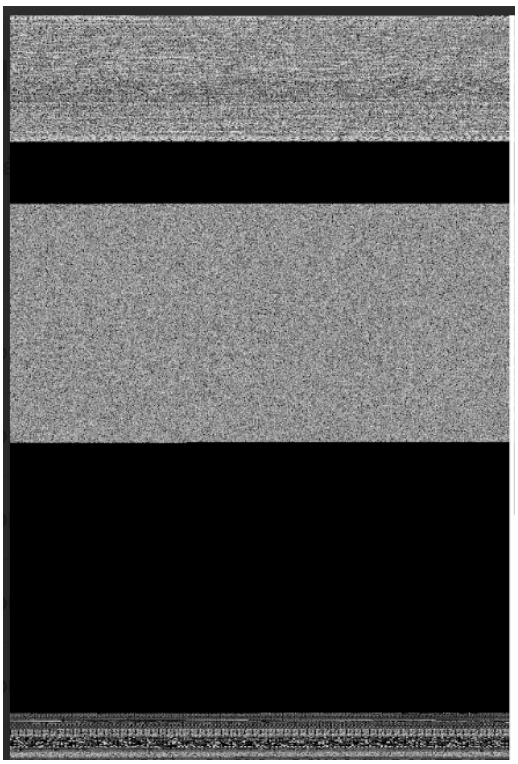
Figure 5.2: Original image sample from VirusShare dataset.

Ωστόσο, η δημιουργία εικόνων δεν είναι από μόνη της επαρκής για την εκπαίδευση των μοντέλων. Για να μπορέσουν τα μοντέλα να μάθουν αποτελεσματικά, είναι απαραίτητο οι εικόνες να έχουν τις ίδιες διαστάσεις, ώστε να διασφαλίζεται η ομοιομορφία των εισόδων και η σωστή λειτουργία των αλγορίθμων κατά την επεξεργασία.

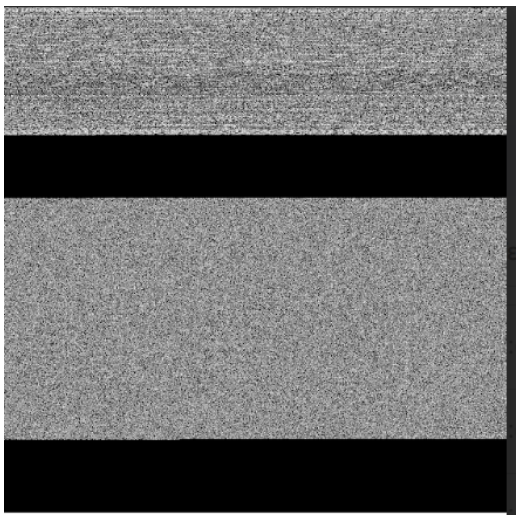
Για τον σκοπό αυτό χρησιμοποιήθηκαν δύο τεχνικές, ανάλογα με την περίπτωση:

- **Cropping** (κόψιμο) για εικόνες μεγαλύτερες από 1024×1024 pixels.
- **Padding** (συμπλήρωση) για εικόνες μικρότερες από 1024×1024 pixels.

Παράδειγμα Cropping:

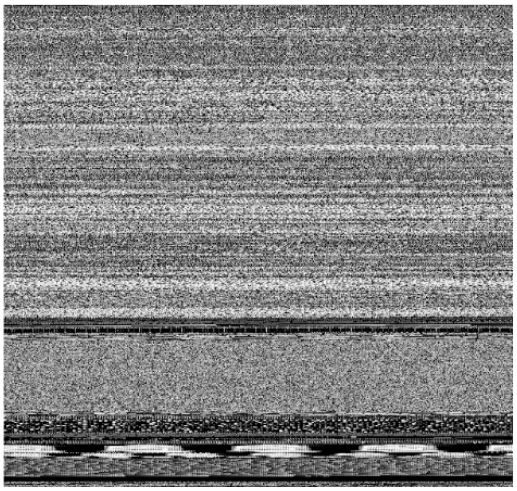


Σχήμα 5.3: Αρχική εικόνα (μεγαλύτερη από 1024×1024)

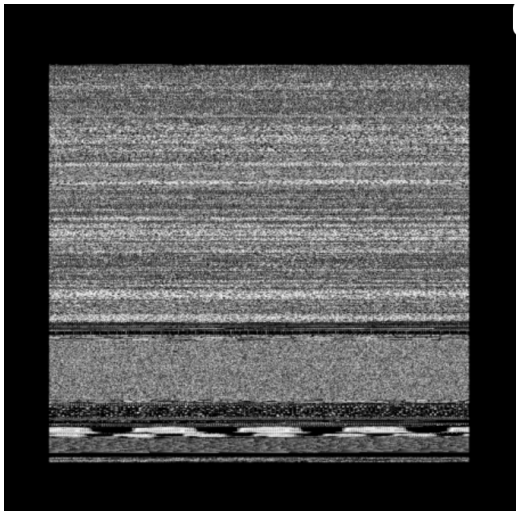


Σχήμα 5.4: Μετά το cropping (1024×1024)

Παράδειγμα Padding:



Σχήμα 5.5: Αρχική εικόνα (μικρότερη από 1024×1024)



Σχήμα 5.6: Μετά το padding (1024×1024)

Αυτή η προσέγγιση ακολουθήθηκε για τα μοντέλα που εκπαιδεύσαμε από την αρχή μόνοι μας. Στη συνέχεια θέλαμε να δούμε κάποια προεκπαιδευμένα μοντέλα και κάποια επίσης δικά μας τα οποία θα είχαν μικρότερη είσοδο για καλύτερη απόδοση. Έτσι χρησιμοποιήσαμε μία λίγο διαφορετική προσέγγιση. Όπως είδαμε παραπάνω τα δεδομένα αποτελούνται από μεγάλο ποσοστό εικόνων μικρότερων ακόμη και autoencoders CNN για να δημιουργήσουμε αναπαραστάσεις τρισδιάστατες από τις εικόνες μας.

Autoencoders

Οι autoencoders είναι νευρωνικά δίκτυα που χρησιμοποιούνται για τη συμπίεση και την ανακατασκευή δεδομένων. Σε αυτή τη μέθοδο, το εκτελέσιμο αρχείο μετατρέπεται σε μια ακολουθία δεδομένων και ο autoencoder συμπιέζει τα δεδομένα και τα μετατρέπει σε εικόνα. Αυτή η μέθοδος είναι χρήσιμη για την εξαγωγή χαρακτηριστικών και την ταξινόμηση.

Μέχρι στιγμής, πραγματοποιείται προεπεξεργασία όλων των εικόνων μέσω resize ή crop, όπως περιγράφηκε προηγουμένως. Σε αυτό το στάδιο, η χρήση padding παρατηρήθηκε ότι προκαλεί απώλεια πληροφορίας λόγω της προσθήκης μαύρων περιοχών, γεγονός που επηρεάζει αρνητικά την ποιότητα των δεδομένων.

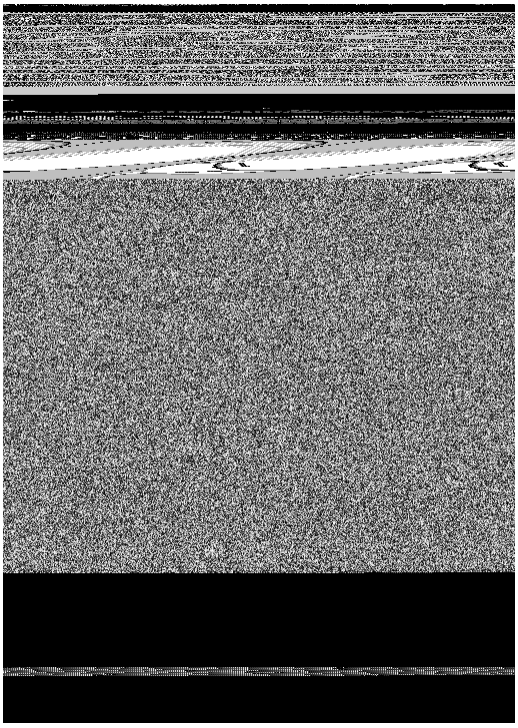
Για την αλλαγή μεγέθους χρησιμοποιείται η βιβλιοθήκη torchvision.transforms και ο αλγόριθμος Lanczos resampling [55]. Ο συγκεκριμένος αλγόριθμος αποτελεί τεχνική παρεμβολής υψηλής ποιότητας, βασισμένη σε παραθύρωση του sinc φίλτρου, επιτυγχάνοντας λεπτομερή και ομαλή αναπαραγωγή εικόνας κατά την αλλαγή διαστάσεων, περιορίζοντας φαινόμενα aliasing και θολότητας.

Πιο συγκεκριμένα, όταν η εικόνα είναι μεγαλύτερη από τις επιθυμητές διαστάσεις, εφαρμόζεται περικοπή με στόχο τη διατήρηση των πιο σημαντικών περιοχών, ενώ σε περίπτωση μικρότερης εικόνας πραγματοποιείται μεγέθυνση αντί για χρήση padding.

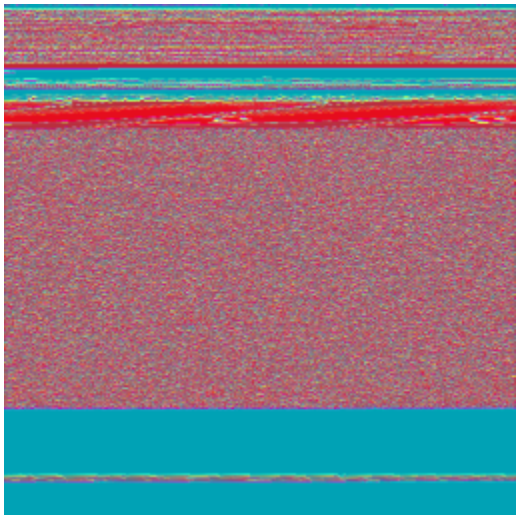
Αρχικά, επιχειρήθηκε η τροφοδότηση των autoencoders με τις εικόνες όπως είχαν περιγραφεί παραπάνω, όμως τα αποτελέσματα δεν ήταν ικανοποιητικά. Η υιοθέτηση της παραπάνω διαδικασίας βελτίωσε σημαντικά την απόδοση του συστήματος, καθώς

μειώνει την υπολογιστική πολυπλοκότητα, αυξάνει την ταχύτητα εκπαίδευσης και διασφαλίζει τη συνέπεια και την επεξεργασία των εισόδων, μειώνοντας παράλληλα την παραμόρφωση των εικόνων.

Παράδειγμα Εικόνας από Autoencoder:



Σχήμα 5.7: Before Autoencoder



Σχήμα 5.8: Autoencoder Image

5.5 Σύνοψη Κεφαλαίου και συνέχεια

Στο πλαίσιο λοιπόν της παρούσας εργασίας, χρησιμοποιήθηκαν τρία σύνολα δεδομένων διαφορετικού μεγέθους και χαρακτηριστικών, προκειμένου να αξιολογηθεί η απόδοση των μοντέλων CNN και ViT σε πραγματικές συνθήκες. Τα δεδομένα περιλάμβαναν:

- Κακόβουλα αρχεία (malware) από τη δημόσια πηγή VirusShare
- Καλόβουλα εκτελέσιμα (benign) από καθαρές εγκαταστάσεις Windows και εμπιστευμένες εφαρμογές,
- Επεκτάσεις adversarial (τροποποιημένα δείγματα) για έλεγχο ανθεκτικότητας.

Η μεθοδολογία περιελάμβανε:

- Μετατροπή των δυαδικών αρχείων σε εικόνες
- Προ επεξεργασία (κανονικοποίησης, padding, για βελτιστοποίηση της εκπαίδευσης,
- Διαχωρισμό σε train/validation/test sets (70-20-10) και μείωση με autoencoders.

Ο στόχος ήταν να εξεταστεί η ικανότητα των μοντέλων να γενικεύουν σε μη ισορροπημένα και πολύπλοκα δεδομένα, πλησιάζοντας τις προκλήσεις του πραγματικού κόσμου. Στο επόμενο κεφάλαιο θα κάνουμε θεωρητική ανάλυση των μοντέλων μηχανικής μάθησης που θα χρησιμοποιήσουμε σε αυτήν την εργασία. Στη συνέχεια θα δούμε την μεθοδολογία των πειραμάτων και θα αναλύσουμε τα αποτελέσματα των πειραμάτων μας.

Κεφάλαιο 6

Θεωρητικό Υπόβαθρο Πειραμάτων

Σε αυτό το κεφάλαιο παρουσιάζονται αναλυτικά τα τεχνικά χαρακτηριστικά των αλγορίθμων που χρησιμοποιήθηκαν, καθώς και οι παράμετροι και οι επιλογές που έγιναν κατά την ανάπτυξη τους και κατά περιπτώσεις στην εκπαίδευσή τους ώστε στα επόμενα κεφάλαια να δούμε αναλυτικά τα αποτελέσματα και στη συνέχεια να τα αξιολογήσουμε.

6.1 Models

6.1.1 CNN

Γενική Θεωρία των Συνελικτικών Νευρωνικών Δικτύων

Τα Convolutional Neural Networks (CNN) είναι ένας τύπος τεχνητών νευρωνικών δικτύων που είναι ειδικά σχεδιασμένα για την επεξεργασία δεδομένων που έχουν τη μορφή πινάκων, όπως οι εικόνες. Ο βασικός μηχανισμός των CNN είναι η χρήση συνελικτικών φίλτρων (convolutional filters), τα οποία εφαρμόζονται στην είσοδο για να εντοπίσουν συγκεκριμένα χαρακτηριστικά, όπως άκρα, σχήματα ή υφές.

Μία από τις πρώτες πετυχιμένες εφαρμογές πρωτοπαρουσιάστηκε στο έργο των LeCun et al. (1998) [56], όπου εφαρμόστηκε με επιτυχία στην αναγνώριση χειρόγραφων ψηφίων ενώ πλέον βρίσκει εφαρμογές σε ένα ευρύ φάσμα πεδίων, όπως η ιατρική απεικόνιση, η αυτόνομη οδήγηση, η ανίχνευση κακόβουλου λογισμικού, καθώς και η αναγνώριση προσώπων και αντικειμένων.

Η ιδιαιτερότητα των CNN σε σχέση με τα παραδοσιακά νευρωνικά δίκτυα είναι ότι χρησιμοποιούν στρώματα συνελικτικής επεξεργασίας (convolutional layers) που επιτρέπουν στο δίκτυο να μάθει τοπικές χωρικές ιεραρχίες από τα δεδομένα εισόδου. Αυτό είναι ιδιαίτερα χρήσιμο στην αναγνώριση και κατηγοριοποίηση εικόνων, όπου οι σχέσεις μεταξύ γειτονικών εικονοστοιχείων είναι κρίσιμες.

Στρώματα Συνελικτικής Επεξεργασίας

Τα βασικά συστατικά των CNN είναι τα συνελικτικά φίλτρα (convolutional filters), τα οποία εφαρμόζονται μέσω συνελίξεων(applied via convolution) πάνω στην είσοδο. Το φίλτρο (kernel) είναι ένας μικρός πίνακας βαρών που μαθαίνει να εντοπίζει συγκεκριμένα χαρακτηριστικά που αντιπροσωπεύουν ένα συγκεκριμένο μοτίβο ή χαρακτηριστικό. Όταν εφαρμόζεται στην εικόνα, παράγεται ένας πίνακας χαρακτηριστικών

(feature map), ο οποίος αντιπροσωπεύει τα σημεία της εικόνας που αντιστοιχούν σε αυτό το χαρακτηριστικό.

Αυτό επιτρέπει στο δίκτυο να ¨αντιλαμβάνεται' βασικά χαρακτηριστικά της εικόνας, όπως ακμές ή γωνίες, τα οποία στη συνέχεια συνδυάζονται για να δημιουργηθούν πιο σύνθετα χαρακτηριστικά στα επόμενα επίπεδα.

Στρώματα Συγκέντρωσης (Pooling Layers)

Τα pooling layers χρησιμοποιούνται για να μειώσουν τη διάσταση των feature maps, διατηρώντας ωστόσο τις πιο σημαντικές πληροφορίες. Το πιο κοινό στρώμα συγκέντρωσης είναι το max pooling, όπου από μια περιοχή του πίνακα χαρακτηριστικών επιλέγεται το μεγαλύτερο αριθμητικό αποτέλεσμα. Αυτό επιτρέπει τη μείωση της ανάλυσης της εικόνας, κάτι που συμβάλλει στην αύξηση της αποδοτικότητας του μοντέλου και στη μείωση της υπερπροσαρμογής (overfitting).

Βασικές Παράμετροι Εκπαίδευσης

Κατά την εκπαίδευση των Convolutional Neural Networks (CNN), σημαντικές παράμετροι που επηρεάζουν την απόδοση και τη σύγκλιση του μοντέλου είναι οι εξής:

- Learning rate (η): Η παράμετρος που καθορίζει το μέγεθος του βήματος κατά την ενημέρωση των βαρών, δηλαδή, για κάθε βάρος w, η ενημέρωση γίνεται ως:

$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

  όπου L είναι η loss function.

- Batch size: Ο αριθμός των δειγμάτων που χρησιμοποιούνται για τον υπολογισμό της κλίσης σε κάθε βήμα ενημέρωσης.

- Epochs: Ο συνολικός αριθμός διελεύσεων από ολόκληρο το σύνολο εκπαίδευσης.

- Optimizer: Ο αλγόριθμος που καθορίζει τον τρόπο με τον οποίο ενημερώνονται τα βάρη, όπως ο Stochastic Gradient Descent (SGD) ο Adam,Adagrad κλπ .

Διαδικασία Backpropagation

Η εκπαίδευση των CNN βασίζεται στον αλγόριθμο του backpropagation, που υπολογίζει τις παραγώγους της loss function L ως προς τα βάρη w μέσω του κανόνα της αλυσιδας (chain rule).

Κατά το forward pass, τα δεδομένα εισόδου x περνούν μέσα από τα στρώματα του δικτύου με συναρτήσεις ενεργοποίησης f, δίνοντας την έξοδο $y = f(x; w)$.

Υπολογίζεται το σφάλμα μέσω της loss function, π.χ.:

$$L = \frac{1}{N} \sum_{i=1}^{N} \ell(y_i, \hat{y}_i)$$

όπου ℓ είναι η loss per sample (π.χ. cross-entropy) και $\hat{y}_i$ η πραγματική ετικέτα.

Κατά το backward pass, υπολογίζονται οι μερικές παράγωγοι:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w}$$

και στη συνέχεια τα βάρη ενημερώνονται όπως:

$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

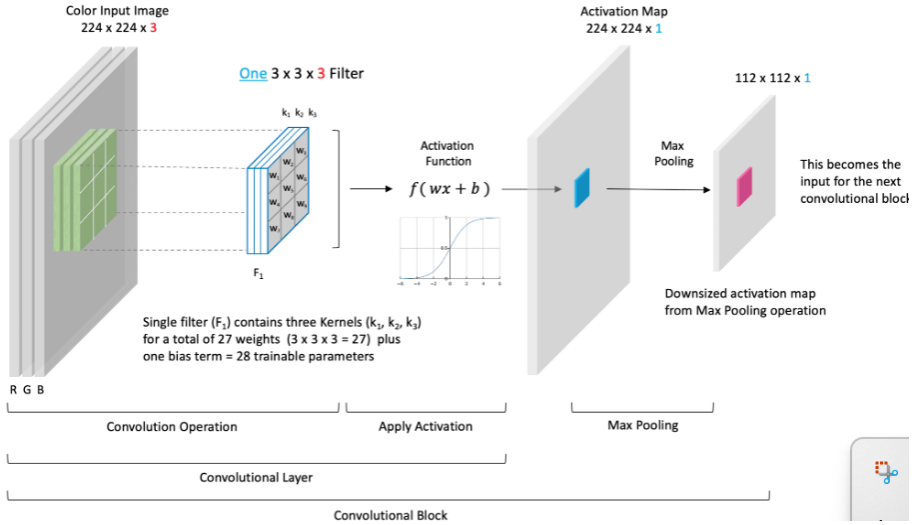
Η επανάληψη της διαδικασίας αυτής καθ' όλη τη διάρκεια της εκπαίδευσης επιτρέπει στο δίκτυο να βελτιώνει σταδιακά την απόδοσή του.

Πλεονεκτήματα των CNN στην επεξεργασία εικόνας

- Η χρήση συνελικτικών φίλτρων επιτρέπει την αναγνώριση τοπικών χαρακτηριστικών χωρίς την ανάγκη χειροκίνητης εξαγωγής χαρακτηριστικών.
- Είναι αποτελεσματικά από άποψη υπολογιστικής ισχύος, καθώς η συνελικτική διαδικασία εκμεταλλεύεται την τοπική χωρική δομή των εικόνων.
- Είναι ανθεκτικά σε μικρές μετατοπίσεις, αλλαγές κλίμακας ή περιστροφές της εισόδου.
- Ιδανικά για εφαρμογές όπως:
  - Ανίχνευση αντικειμένων
  - Αναγνώριση προσώπων

Περιορισμοί των CNN

- Τα συνελικτικά φίλτρα επικεντρώνονται σε τοπικές περιοχές της εικόνας και μπορεί να αγνοούν παγκόσμιες σχέσεις μεταξύ απομακρυσμένων σημείων.
- Αυτό είναι πρόβλημα όταν οι σχέσεις μεταξύ μακρινών περιοχών είναι κρίσιμες για την ανάλυση της εικόνας.
- Οι Vision Transformer προσπαθούν να αντιμετωπίσουν αυτόν τον περιορισμό μέσω του μηχανισμού προσοχής [57].



Σχήμα 6.1: Βήμα-βήμα λειτουργία ενός Convolutional Neural Network (CNN). Πηγή: ResearchGate

Pretrained CNN Models

Τα Convolutional Neural Networks (CNN) αποτελούν τη βασική κατηγορία μοντέλων που έχουν χρησιμοποιηθεί για την επεξεργασία και κατανόηση εικόνων, λόγω της ικανότητάς τους να εκμεταλλεύονται τη χωρική δομή των δεδομένων. Με την πάροδο του χρόνου, έχουν αναπτυχθεί πολλές διαφορετικές αρχιτεκτονικές CNN με σκοπό τη βελτίωση της ακρίβειας, της αποδοτικότητας και της ικανότητας γενίκευσης.

Ένα σημαντικό πλεονέκτημα αυτών των μοντέλων είναι ότι μπορούν να **προεκπαιδευτούν** σε μεγάλες βάσεις δεδομένων, όπως το ImageNet [58], το οποίο περιέχει πάνω από ένα εκατομμύριο εικόνες ταξινομημένες σε χιλιάδες κατηγορίες. Η προεκπαίδευση επιτρέπει στο μοντέλο να μάθει γενικά χαρακτηριστικά εικόνων, όπως ακμές, υφές, σχήματα και σύνθετες οπτικές δομές, τα οποία είναι χρήσιμα και σε άλλες εργασίες επεξεργασίας εικόνας.

Με τη μέθοδο της **μεταφοράς μάθησης** (transfer learning), τα προεκπαιδευμένα μοντέλα μπορούν να προσαρμοστούν σε νέες εργασίες με λιγότερα δεδομένα, κάνοντας είτε:

- **Feature extraction** [59] – χρήση των εξαγόμενων χαρακτηριστικών από τα τελευταία συνελικτικά στρώματα, ή
- **Fine-tuning** [60] – επανεκπαίδευση ορισμένων στρωμάτων (ή όλου του μοντέλου) πάνω στο νέο σύνολο δεδομένων.

Σε αυτή την εργασία χρησιμοποιούμε το **ResNet-18**

ResNet-18

Το ResNet-18 είναι μια έκδοση του ResNet, ενός πολύ σημαντικού αρχιτεκτονικού μοντέλου για τη βαθιά μάθηση που επιλύει το πρόβλημα της υποβάθμισης της απόδοσης σε πολύ βαθιά νευρωνικά δίκτυα. Το πρόβλημα αυτό συμβαίνει όταν το μοντέλο γίνεται τόσο βαθύ που η απόδοσή του αρχίζει να χειροτερεύει αντί να βελτιώνεται.

Το ResNet εισήγαγε τα υπολειμματικά μπλοκ (residual blocks), τα οποία επιτρέπουν στα δεδομένα να παρακάμπτουν ορισμένα στρώματα του δικτύου και μεταφέρουν την πληροφορία απευθείας στα επόμενα. Αυτό βελτιώνει τη ροή της πληροφορίας και βοηθάει στην αποφυγή του προβλήματος του εξαφανιζόμενου σφάλματος (vanishing gradient), που εμφανίζεται όταν οι βαθμώσεις (gradients) μηδενίζονται ή αποδυναμώνονται δραστικά κατά τη διάρκεια της εκπαίδευσης. Το ResNet-18 είναι μια ελαφρύτερη έκδοση της αρχιτεκτονικής, με 18 στρώματα, και είναι ιδιαίτερα χρήσιμο για εφαρμογές με περιορισμένους υπολογιστικούς πόρους.

6.1.2 ViT

Τα Transformer είναι μια αρχιτεκτονική νευρωνικών δικτύων που παρουσιάστηκε το 2017 από την ερευνητική ομάδα της Google στο άρθρο με τίτλο "Attention is All You Need" [61]. Εισήγαγαν μια νέα προσέγγιση στη μηχανική μάθηση και την επεξεργασία δεδομένων, επιτρέποντας πιο αποδοτικές λύσεις σε προβλήματα όπως η φυσική γλώσσα (Natural Language Processing - NLP) και η επεξεργασία εικόνας (Vision Processing). Σε αντίθεση με προηγούμενες αρχιτεκτονικές όπως τα επαναληπτικά νευρωνικά δίκτυα (RNN) και τα συνελικτικά (CNN), οι Transformer χρησιμοποιούν έναν μηχανισμό προσοχής (attention mechanism), ο οποίος προσφέρει σημαντικά πλεονεκτήματα σε πολλές εφαρμογές. Αν και σχεδιάστηκαν για γλωσσικά μοντέλα, η χρήση τους έχει επεκταθεί και στην ανίχνευση κακόβουλου λογισμικού [62].

Γενική Θεωρία Vision Transformer

Η βασική ιδέα πίσω από τα Vision Transformers είναι ο ίδιος μηχανισμός προσοχής [63] που χρησιμοποιείται στους κλασικούς Transformer, με τη διαφορά ότι αντί για λέξεις, ως είσοδος δίνονται εικόνες. Ο μηχανισμός αυτός επιτρέπει στο δίκτυο να εστιάζει σε σημαντικές περιοχές της εισόδου, αγνοώντας περιττές πληροφορίες. Έτσι, αποκτά την ικανότητα να «βλέπει» ολόκληρη την είσοδο και να εντοπίζει τα πιο κρίσιμα τμήματά της ανάλογα με το πλαίσιο. Ο πυρήνας του attention mechanism είναι ο υπολογισμός ενός πίνακα προσοχής (attention map), ο οποίος καθορίζει ποιες περιοχές της εισόδου λαμβάνουν μεγαλύτερη βαρύτητα.

Οι Vision Transformers χρησιμοποιούν κυρίως αυτο-προσοχή (self-attention), δηλαδή κάθε τμήμα της εισόδου υπολογίζει την αλληλεπίδρασή του με όλα τα άλλα. Αυτό είναι ιδιαίτερα χρήσιμο για ακολουθιακά δεδομένα (όπως κείμενο ή βίντεο), όπου οι εξαρτήσεις μεταξύ των στοιχείων είναι σύνθετες και μακροπρόθεσμες. Οι πολλαπλές κεφαλές προσοχής (multi-head attention) επιτρέπουν στο δίκτυο να εντοπίσει διαφορετικούς τύπους σχέσεων ταυτόχρονα, από πολλαπλές ¨οπτικές γωνίες¨.

Οι Transformer αποτελούνται από δύο βασικά μέρη: τον κωδικοποιητή (encoder) και τον αποκωδικοποιητή (decoder). Ο κωδικοποιητής μετατρέπει την είσοδο σε διανύσματα χαρακτηριστικών, ενώ ο αποκωδικοποιητής τα χρησιμοποιεί για να δημιουργήσει την έξοδο. Και τα δύο μέρη χρησιμοποιούν μηχανισμούς προσοχής, με τον κωδικοποιητή να εστιάζει αποκλειστικά στην είσοδο και τον αποκωδικοποιητή τόσο στην είσοδο όσο και στην έξοδο που έχει παραχθεί έως εκείνο το σημείο.

Vision Transformer (ViT-B/8)

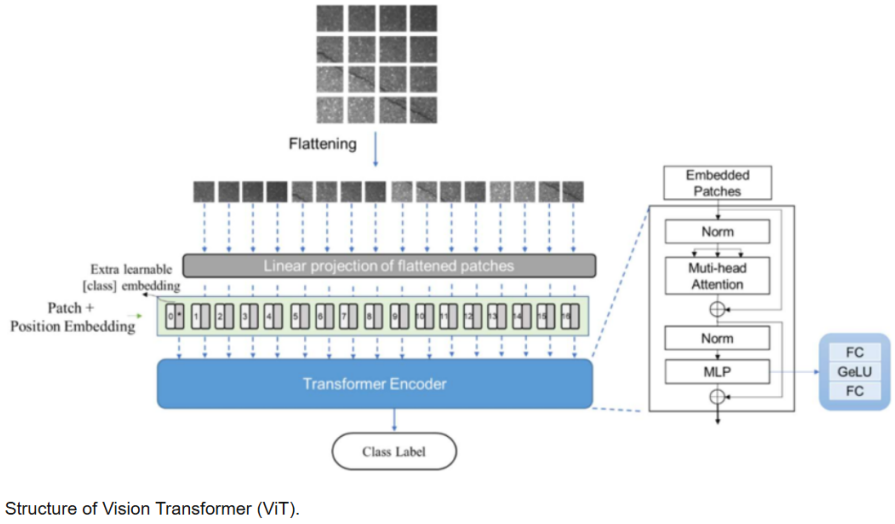
Το Vision Transformer (ViT-B/8) είναι μια έκδοση του ViT, που παρουσιάστηκε από τη Google το 2020. Το "224" στο όνομα αναφέρεται στο μέγεθος της εικόνας εισόδου

(224x224 pixels). Αντί να επεξεργάζεται ολόκληρη την εικόνα ως ενιαίο σύνολο, το μοντέλο χωρίζει την εικόνα σε μη επικαλυπτόμενα μικρά κομμάτια (patches) και εφαρμόζει προσοχή πάνω σε αυτά.

Η αρχιτεκτονική είναι Google, αλλά με μικρή παραλλαγή vit_small_patch8_224 στο PyTorch είναι community/third-party υλοποίηση με pretrained weights.

Το Vision Transformer (ViT-B/8) είναι μια έκδοση του ViT, που παρουσιάστηκε από τη Google το 2020. Το ¨224' στο όνομα αναφέρεται στο μέγεθος της εικόνας εισόδου (224x224 pixels). Αντί να επεξεργάζεται ολόκληρη την εικόνα ως ενιαίο σύνολο, το μοντέλο χωρίζει την εικόνα σε μη επικαλυπτόμενα μικρά κομμάτια (patches) και εφαρμόζει προσοχή πάνω σε αυτά.

Αυτή η προσέγγιση επιτρέπει στο δίκτυο να ενσωματώνει τόσο τοπικές όσο και παγκόσμιες πληροφορίες από την εικόνα. Έχει αποδειχθεί ιδιαίτερα αποτελεσματική σε μεγάλα σύνολα δεδομένων, όπως το ImageNet, ξεπερνώντας σε αρκετές περιπτώσεις την απόδοση των κλασικών CNN.



Σχήμα 6.2: Βήμα-βήμα λειτουργία ενός Vision Transformer (ViT) [2]. Πηγή: ResearchGate

Ο Μηχανισμός Προσοχής

Η προσοχή είναι ο τρόπος με τον οποίο το μοντέλο εντοπίζει τις πιο κρίσιμες πληροφορίες της εισόδου. Αυτό επιτυγχάνεται μέσω τριών διανυσμάτων: query (ερώτημα), key (κλειδί) και value (τιμή). Το ερώτημα αφορά τη θέση της εισόδου που εξετάζεται, τα κλειδιά περιέχουν πληροφορίες για όλες τις άλλες θέσεις και οι τιμές είναι τα αντίστοιχα διανύσματα χαρακτηριστικών. Ο υπολογισμός των εσωτερικών γινομένων μεταξύ των query και key διανυσμάτων καθορίζει τη ¨σχετικότητα' μεταξύ των θέσεων. Στη συνέχεια, οι τιμές (value vectors) σταθμίζονται αναλόγως, ώστε να δοθεί έμφαση στα πιο σχετικά τμήματα.

Vision Transformer (ViT)

Οι ViT αποτελούν ουσιαστικά μια προσαρμογή των Transformer για εικόνες και παρουσιάστηκαν στο άρθρο "An Image is Worth 16x16 Words" [64]. Αντί για λέξεις, η είσοδος χωρίζεται σε μικρά τετραγωνικά κομμάτια (patches), τα οποία μετατρέπονται σε διανύσματα εισόδου όπως γίνεται και με τις λέξεις στα γλωσσικά μοντέλα.

Κατακερματισμός Εικόνας σε Patches

Η εικόνα χωρίζεται σε τετραγωνικά patches συγκεκριμένου μεγέθους. Κάθε patch αντιμετωπίζεται ως ανεξάρτητη μονάδα, όπως οι λέξεις σε ένα κείμενο. Μετατρέπεται σε διάνυσμα μέσω γραμμικού μετασχηματισμού και στη συνέχεια εισάγεται στον κωδικοποιητή. Αυτή η προσέγγιση επιτρέπει στο μοντέλο να αναγνωρίζει σχέσεις τόσο τοπικές όσο και παγκόσμιες.

Απαιτήσεις και Αποτελέσματα

Οι ViT απαιτούν μεγάλη υπολογιστική ισχύ και εκπαίδευση σε μεγάλα σύνολα δεδομένων. Ωστόσο, όταν πληρούνται αυτές οι προϋποθέσεις, η απόδοσή τους είναι εξαιρετική. Σε πολλές περιπτώσεις υπερτερούν των CNN, ειδικά όταν έχουν εκπαιδευτεί σε μεγάλα datasets όπως το ImageNet.

Γενική Θεωρία Transformer

Οι Transformer, όπως περιγράφονται στο [63], βασίζονται στον μηχανισμό προσοχής. Η βασική εξίσωση είναι:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6.1}$$

όπου Q, K, V προέρχονται από την είσοδο μέσω γραμμικών μετασχηματισμών, και d_k είναι η διάσταση των κλειδιών. Η συνάρτηση softmax εξασφαλίζει ότι τα βάρη έχουν άθροισμα 1.

Αυτο-Προσοχή Όταν Q, K, V προέρχονται από την ίδια είσοδο X, τότε έχουμε αυτο-προσοχή:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V \tag{6.2}$$

Αυτό επιτρέπει σε κάθε στοιχείο να εξετάζει όλα τα υπόλοιπα και να ενσωματώνει πληροφορίες από ολόκληρη την ακολουθία.

Κωδικοποιητής-Αποκωδικοποιητής Ο κωδικοποιητής περιλαμβάνει:

- Πολλαπλή προσοχή (multi-head attention)
- Πλήρως συνδεδεμένο δίκτυο με ενεργοποίηση GELU

$$\text{FFN}(x) = \text{GELU}(xW_1 + b_1)W_2 + b_2 \tag{6.3}$$

- Υπολειμματικές συνδέσεις και κανονικοποίηση στρώματος

Ο αποκωδικοποιητής προσθέτει ένα υπόστρωμα διασταυρούμενης προσοχής (cross-attention), με Q από την έξοδο του αποκωδικοποιητή και K, V από την έξοδο του κωδικοποιητή.

Ενοποίηση Transformer και Vision Transformer

Οι Transformer και οι ViT μοιράζονται τον ίδιο πυρήνα: τον μηχανισμό προσοχής. Η βασική εξίσωση είναι:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{6.4}$$

ViT, η εικόνα $x \in \mathbb{R}^{H\times W\times C}$ χωρίζεται σε $N = \frac{HW}{P^2}$ patches, τα οποία μετατρέπονται σε διανύσματα μέσω γραμμικού μετασχηματισμού E:

$$z_0 = [x_{\text{class}};, x_{p_1}E;, \ldots;, x_{p_N}E] + E_{\text{pos}} \tag{6.5}$$

Στην συγκεκριμένη εργασία η ενασχόλησή μας πειραματικά έχει να κάνει μόνο με Vision Transformer παρόλα αυτά η ανάλυση και ηκατανόηση σε βασικό βαθμό των Transformer κρίθηκε σκόπιμη ώστε να έχουμε καλύτερη κατανόηση των συγκεκριμένων αρχιτεκτονικών που οι ViT αποτελούν βασικό τους κομμάτι

6.1.3 Autoencoders

Οι απλοί αποκωδικοποιητές Autoencoders είναι ένας τύπος νευρωνικού δικτύου που χρησιμοποιείται για μη επιβλεπόμενη μάθηση και χρησιμοποιείται για συμπίεση δεδομένων. Αποτελούνται από δύο κύρια μέρη, τον κωδικοποιητή και τον αποκωδικοποιητή. Ο κωδικοποιητής παίρνει τα δεδομένα εισόδου και τα συμπιέζει σε έναν χώρο χαμηλότερων διαστάσεων ,γνωστό ως bottleneck , ενώ ο αποκωδικοποιητής προσπαθεί να ανακατασκευάσει τα αρχικά δεδομένα από αυτή τη συμπιεσμένη αναπαράσταση.

Ο στόχος του Autoencoder είναι να ελαχιστοποιήσει τη διαφορά μεταξύ της αρχικής εισόδου και της ανακατασκευασμένης εξόδου.Αυτό επιτρέπει στο δίκτυο να μάθει αποτελεσματικές και συνοπτικές αναπαραστάσεις των δεδομένων . κάτι που βοηθάει στη συμπίεση δεδομένων και στη μείωση διαστάσεων.

Autoencoders

Οι **Autoencoders** [65] είναι νευρωνικά δίκτυα που χρησιμοποιούνται για τη μη επιβλεπόμενη μάθηση αναπαραστάσεων δεδομένων. Στόχος τους είναι να συμπιέσουν τα δεδομένα σε έναν μικρότερο χώρο και στη συνέχεια να ανακατασκευάσουν την αρχική είσοδο. Από μαθηματική άποψη, ο στόχος τους είναι να μάθουν μια συμπιεσμένη αναπαράσταση των δεδομένων, ελαχιστοποιώντας το σφάλμα ανακατασκευής.

Ένας Autoencoder αποτελείται από δύο βασικά μέρη:

- **Κωδικοποιητής (Encoder)**: Συμπιέζει τα δεδομένα εισόδου σε έναν μικρότερο χώρο (χαμηλής διάστασης αναπαράσταση).
- **Αποκωδικοποιητής (Decoder)**: Ανακατασκευάζει την αρχική είσοδο από τη συμπιεσμένη αναπαράσταση.

Μαθηματική Περιγραφή

Αν υποθέσουμε ότι τα δεδομένα εισόδου είναι $\mathbf{x} \in \mathbb{R}^n$, ο κωδικοποιητής προσπαθεί να μάθει μια αναπαράσταση $\mathbf{z} \in \mathbb{R}^m$, όπου $m < n$. Η λειτουργία του κωδικοποιητή μπορεί να περιγραφεί ως εξής:

$$\mathbf{z} = f(\mathbf{x}) = \sigma(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e)$$

όπου:

- $\mathbf{W}_e$ είναι τα βάρη του κωδικοποιητή με διαστάσεις $m \times n$,
- $\mathbf{b}_e$ είναι το διάνυσμα μετατόπισης του κωδικοποιητή με διαστάσεις m,
- σ είναι η μη γραμμική συνάρτηση ενεργοποίησης (π.χ. ReLU or Sigmoid).

Ο αποκωδικοποιητής ανακατασκευάσει την είσοδο από την αναπαράσταση $\mathbf{z}$:

$$\hat{\mathbf{x}} = g(\mathbf{z}) = \sigma(\mathbf{W}_d\mathbf{z} + \mathbf{b}_d)$$

όπου:

- $\mathbf{W}_d$ είναι τα βάρη του αποκωδικοποιητή με διαστάσεις $n \times m$,
- $\mathbf{b}_d$ είναι το διάνυσμα μετατόπισης του αποκωδικοποιητή με διαστάσεις n,
- $\hat{\mathbf{x}}$ είναι η ανακατασκευή των δεδομένων εισόδου $\mathbf{x}$.

Ο στόχος του Autoencoder είναι να ελαχιστοποιήσει τη διαφορά μεταξύ της εισόδου $\mathbf{x}$ και της ανακατασκευασμένης εξόδου $\hat{\mathbf{x}}$. Ένα συνηθισμένο μέτρο σφάλματος είναι η **μέση τετραγωνική απόκλιση (Mean Squared Error, MSE)**:

$$\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2$$

Η συνάρτηση απώλειας αυτή δείχνει πόσο καλά ο αποκωδικοποιητής ανακατασκευάζει την αρχική είσοδο.

Τύποι Autoencoders

Υπάρχουν διάφορες κατηγορίες Autoencoders [66], καθεμία με συγκεκριμένες εφαρμογές:

- **Undercomplete Autoencoders**: Ο χώρος της αναπαράστασης $\mathbf{z}$ είναι μικρότερος από τον χώρο εισόδου, αναγκάζοντας το δίκτυο να μάθει σημαντικά χαρακτηριστικά.
- **Denoising Autoencoders**: Εκπαιδεύονται για να αφαιρούν θόρυβο από τα δεδομένα.
- **Sparse Autoencoders**: Αναγκάζουν την αναπαράσταση $\mathbf{z}$ να έχει πολλές μηδενικές τιμές, επικεντρώνοντας την προσοχή σε σημαντικά χαρακτηριστικά.
- **Variational Autoencoders (VAEs)**: Χρησιμοποιούνται για τη μοντελοποίηση της κατανομής των δεδομένων εισόδου και την παραγωγή νέων δεδομένων.
- **Convolutional Autoencoders (CAEs)**: Εφαρμόζουν συνελικτικά επίπεδα, κατάλληλα για επεξεργασία εικόνων.

Χρήση του Autoencoder στο παρόν Πείραμα

Στο συγκεκριμένο πείραμα, χρησιμοποιείται ένας **(Convolutional Autoencoder)**. Αυτός ο τύπος Autoencoder είναι ιδιαίτερα κατάλληλος για δεδομένα εικόνων, καθώς τα συνελικτικά επίπεδα βοηθούν στην εξαγωγή σημαντικών χαρακτηριστικών και στη μείωση των διαστάσεων χωρίς να χάνεται ουσιαστική πληροφορία.

Η κωδικοποιημένη αναπαράσταση που προκύπτει έχει 3 χαρακτηριστικά κανάλια (feature maps), κάτι που επιτρέπει την καλύτερη αποθήκευση και ανάλυση των δεδομένων. Χρησιμοποιείται στη διαδικασία εντοπισμού κακόβουλου λογισμικού, όπου η συμπίεση των δεδομένων μέσω του Autoencoder βοηθά στη μείωση του χρόνου επεξεργασίας και στην εξαγωγή ουσιαστικών πληροφοριών για τον εντοπισμό ανωμαλιών.

6.2 Adversarial

Εισαγωγή

Η εργασία Adversarial attacks against Windows PE malware detection: A survey of the state-of-the-art [67] παρέχει μια εκτενή επισκόπηση των **adversarial επιθέσεων** σε συστήματα ανίχνευσης κακόβουλου λογισμικού τύπου Windows PE (Portable Executable). Το PE format είναι το κύριο εκτελέσιμο αρχείο στα Windows, και τα συστήματα ασφαλείας το αναλύουν χρησιμοποιώντας στατικά και δυναμικά χαρακτηριστικά.

Οι ερευνητές εξετάζουν **τεχνικές παραπλάνησης** που εκμεταλλεύονται τις ευπάθειες των συστημάτων ανίχνευσης, με στόχο τη διαφυγή ανίχνευσης. Η εργασία κατηγοριοποιεί τις επιθέσεις, αναλύει τις μεθόδους επίθεσης και προτείνει αμυντικές στρατηγικές.

Κατηγορίες Επιθέσεων

Οι επιθέσεις χωρίζονται σε δύο κύριες κατηγορίες:

1. **Επιθέσεις στο χώρο χαρακτηριστικών** (Feature-space attacks)
2. **Επιθέσεις βασισμένες σε παραμορφώσεις** (Perturbation-based attacks)

Επιθέσεις στο Χώρο των Χαρακτηριστικών

Οι Feature-space attacks δεν αλλάζουν το ίδιο το εκτελέσιμο αρχείο αλλά στοχεύουν το μαθηματικό μοντέλο του συστήματος ανίχνευσης. Οι επιθέσεις αυτές βασίζονται στο γεγονός ότι τα μοντέλα μηχανικής μάθησης (ML-based detectors) χρησιμοποιούν διανύσματα χαρακτηριστικών. Οι επιτιθέμενοι εκμεταλλεύονται αυτήν την αναπαράσταση και δημιουργούν adversarial examples, τα οποία ξεγελούν το μοντέλο χωρίς να επηρεάζουν τη λειτουργία του προγράμματος.

Παραδείγματα τέτοιων επιθέσεων:

- **Gradient-based attacks**: Χρησιμοποιούν πληροφορίες από το gradient descent του μοντέλου για να βρουν τις μικρότερες δυνατές αλλαγές στα χαρακτηριστικά που οδηγούν σε εσφαλμένη ταξινόμηση.

- **Feature-space optimization**: Αλλάζει μόνο συγκεκριμένα χαρακτηριστικά που επηρεάζουν το αποτέλεσμα, διατηρώντας τα υπόλοιπα ανέπαφα.

Επιθέσεις Βασισμένες σε Παραμορφώσεις

Οι perturbation-based attacks στοχεύουν το ίδιο το PE binary, αλλάζοντας μικρά τμήματα του αρχείου με τρόπο που να μην επηρεάζει τη λειτουργικότητά του.

Μορφές αυτών των επιθέσεων:

- **FGSM (Fast Gradient Sign Method)**: Τροποποιεί τα raw bytes του αρχείου βασιζόμενο στο gradient του μοντέλου.
- **Section injection**: Προσθέτει άχρηστες (dummy) sections στο PE header ώστε να αλλάξει το αποτύπωμα του αρχείου.
- **Padding attack**: Εισάγει μη εκτελέσιμο κώδικα (NOPs ή junk bytes) ώστε να επηρεάσει την ανάλυση από το antivirus.

Μοντέλα Απειλής

Η εργασία χωρίζει τις επιθέσεις σε white-box και black-box επιθέσεις.

White-box επιθέσεις

Ο επιτιθέμενος έχει πλήρη γνώση του συστήματος ανίχνευσης και μπορεί να χρησιμοποιήσει:

- **Gradient-based attacks** για να προσαρμόσει το input.
- **Reverse engineering** των χαρακτηριστικών που χρησιμοποιούνται από τον ανιχνευτή.

Black-box επιθέσεις

Ο επιτιθέμενος δεν γνωρίζει το ακριβές μοντέλο ανίχνευσης και χρησιμοποιεί:

- Δημιουργία πολλών δοκιμαστικών δειγμάτων.
- Substitute models για να προσομοιώσει τον ανιχνευτή.

Αμυντικές Στρατηγικές

Οι βασικότερες στρατηγικές άμυνας περιλαμβάνουν:

- **Adversarial training**: Εκπαίδευση με adversarial examples.
- **Feature squeezing**: Μείωση των δυνατοτήτων παραμόρφωσης των χαρακτηριστικών.

- **Ensemble learning**: Χρήση πολλαπλών μοντέλων για αυξημένη ανθεκτικότητα.
- **Detection of adversarial examples**: Έλεγχος εισόδων για πιθανές επιθέσεις.

Συμπεράσματα

Οι adversarial επιθέσεις αποτελούν σημαντική απειλή για τα συστήματα ανίχνευσης κακόβουλου λογισμικού Windows PE. Απαιτείται συνεχής έρευνα για την ανάπτυξη ανθεκτικότερων ανιχνευτών και καλύτερη κατανόηση των επιθέσεων.

6.3 Επανεκπαίδευση

Η επανεκπαίδευση ενός μοντέλου αποτελεί ένα σημαντικό στάδιο για την προσαρμογή του σε νέα δεδομένα, ενώ παράλληλα διατηρεί τις γνώσεις που έχει αποκτήσει από τα παλαιότερα δεδομένα. Στην παρούσα ενότητα εξετάζουμε τις διαφορετικές τεχνικές επανεκπαίδευσης που μπορούν να χρησιμοποιηθούν για την αποτελεσματική προσαρμογή των μοντέλων σε μεταβαλλόμενα σύνολα δεδομένων.

6.3.1 Τύποι Επανεκπαίδευσης

Μεταφορά μάθησης μέσω Fine-Tuning (Transfer Learning)

Το fine-tuning είναι μια τεχνική μεταφοράς μάθησης όπου προσαρμόζουμε ένα προεκπαιδευμένο μοντέλο σε ένα νέο σύνολο δεδομένων. Αυτό επιτρέπει στο μοντέλο να διατηρήσει τις γνώσεις που απέκτησε από τα προηγούμενα δεδομένα και να τις προσαρμόσει για την επίλυση μιας νέας εργασίας.

- Βήματα Fine-Tuning [68]:
  1. Φόρτωση των προ-εκπαιδευμένων βαρών.
  2. Ρύθμιση διαφορετικών ρυθμών εκμάθησης για τα παλιότερα και τα νέα επίπεδα.
  3. Πάγωμα (ή ξεπάγωμα) επιπέδων κατά τη διάρκεια της εκπαίδευσης για καλύτερη προσαρμογή.

Τεχνικές Συνεχούς Μάθησης (Continual Learning)

Στη συνεχή μάθηση, προσπαθούμε να εκπαιδεύσουμε ένα μοντέλο σε νέα δεδομένα χωρίς να ξεχάσει αυτά που έχει ήδη μάθει. Δύο σημαντικές τεχνικές είναι:

Elastic Weight Consolidation (EWC) Βοηθά στην αποφυγή της ¨καταστροφικής λήθης' [69] (catastrophic forgetting) βάζοντας περιορισμό στις παραμέτρους του μοντέλου που είναι σημαντικές για την παλιά εργασία. Οι σημαντικές παράμετροι προστατεύονται κατά την εκπαίδευση στη νέα εργασία.

- Υπολογίζεις τον πίνακα Fisher (Fisher Information Matrix) από τα παλιά δεδομένα.

- Προσθέτεις έναν όρο τακτικοποίησης (regularization) στη συνάρτηση απώλειας που περιορίζει τις αλλαγές στις σημαντικές παραμέτρους.

Κανονικοποίηση (Regularization) για Διατήρηση Γνώσης

Η χρήση τεχνικών κανονικοποίησης όπως η L2 Regularization και η Gradual Unfreezing βοηθά στην αποφυγή της καταστροφικής λήθης και στη διατήρηση των γνώσεων από τα παλιά δεδομένα.

- L2 Regularization: Ελαχιστοποιεί το μέγεθος των βαρών, αποτρέποντας υπερπροσαρμογή αλλά όχι απαραίτητα την καταστροφική λήθη catastrophic forgetting
- Gradual Unfreezing: Σταδιακό ξεπάγωμα των επιπέδων του μοντέλου για καλύτερη προσαρμογή στα νέα δεδομένα.

6.3.2 Συνδυασμός Μεθόδων

Η συνδυασμένη χρήση αυτών των τεχνικών μπορεί να προσφέρει πλεονεκτήματα στη διατήρηση των παλαιών γνώσεων και στην εισαγωγή νέων πληροφοριών, χωρίς να καταστραφεί η προηγούμενη μάθηση. Ωστόσο, υπάρχουν προκλήσεις όπως η διαφορετικότητα των κατανομών δεδομένων και η καταστροφική λήθη.

Πλεονεκτήματα: Διατήρηση παλαιάς γνώσης: Εάν συνδυάσεις το παλιό dataset με το νέο, το μοντέλο θα εξακολουθεί να ¨βλέπει' τις παλιές πληροφορίες κατά την εκπαίδευση, βοηθώντας στη διατήρηση των παλιών γνώσεων. Αυτό μπορεί να αποτρέψει την ¨καταστροφική λήθη' (catastrophic forgetting).

Εμπλουτισμός με νέες πληροφορίες: Το νέο dataset προσθέτει επιπλέον πληροφορίες που μπορούν να βελτιώσουν την απόδοση του μοντέλου σου, ειδικά αν τα νέα δεδομένα σχετίζονται με τις ίδιες ή παρόμοιες κατηγορίες.

Απλή διαδικασία: Δεν χρειάζεται να εφαρμόσεις πιο σύνθετες τεχνικές όπως Elastic Weight Consolidation (EWC) or knowledge distillation. Απλώς εκπαιδεύεις το μοντέλο από την αρχή με τα δύο datasets.

Προκλήσεις: Διαφορετικές Κατανομές Δεδομένων: Αν τα δύο datasets έχουν πολύ διαφορετικές κατανομές (π.χ. το ένα έχει πολύ περισσότερα δείγματα από το άλλο ή περιέχουν διαφορετικά χαρακτηριστικά), αυτό μπορεί να προκαλέσει προβλήματα κατά την εκπαίδευση. Το μοντέλο μπορεί να ¨ξεχάσει' τις γνώσεις του από το πρώτο dataset αν το δεύτερο είναι υπερβολικά μεγάλο ή πολύ διαφορετικό.

Μεγάλος χρόνος εκπαίδευσης: Αν ενώσεις τα δύο datasets και το νέο σύνολο είναι πολύ μεγάλο, η εκπαίδευση θα χρειαστεί περισσότερο χρόνο και υπολογιστική ισχύ.

Καταστροφική Λήθη: Αν το νέο dataset έχει περισσότερα δεδομένα ή το μοντέλο είναι υπερβολικά προσαρμοσμένο στα νέα δεδομένα, μπορεί να ξεχάσει τις γνώσεις από το παλιό dataset, ειδικά αν οι κατηγορίες είναι πολύ διαφορετικές.

6.4 Explainable Ai(XAI)

Η Εξηγήσιμη Τεχνητή Νοημοσύνη αποτελεί έναν ταχέως αναπτυσσόμενο τομέα της τεχνητής νοημοσύνης, με στόχο την ερμηνεία και κατανόηση των αποφάσεων που λαμβάνονται από πολύπλοκα μοντέλα, όπως τα βαθιά νευρωνικά δίκτυα. Καθώς τα

μοντέλα αυτά βρίσκουν εφαρμογή σε κρίσιμους τομείς, όπως η ιατρική διάγνωση και η κυβερνοασφάλεια, η ανάγκη για διαφάνεια και εμπιστοσύνη στις προβλέψεις τους καθίσταται επιτακτική.

Στο πλαίσιο αυτής της εργασίας η μέθοδος Grad-CAM (Gradient-weighted Class Activation Mapping) [70] έχει αναδειχθεί ως ένα ισχυρό εργαλείο για την οπτικοποίηση των περιοχών εισόδου που συμβάλλουν περισσότερο στην τελική απόφαση ενός μοντέλου. Η Grad-CAM χρησιμοποιεί τις κλίσεις των εξόδων του μοντέλου σε σχέση με τα χαρακτηριστικά των τελευταίων επιπέδων, δημιουργώντας χάρτες θερμότητας που υποδεικνύουν τις σημαντικότερες περιοχές της εικόνας για την πρόβλεψη μιας συγκεκριμένης κλάσης. Αυτή η προσέγγιση είναι ιδιαίτερα χρήσιμη για την ερμηνεία μοντέλων όπως τα Convolutional Neural Networks (CNN) και τα Vision Transformer (ViT),που χρησιμοποιούμε στην παρούσα εργασία τα οποία συχνά θεωρούνται ως ¨μαύρα κουτιά' λόγω της πολυπλοκότητας τους.

6.5 Αξιολόγηση Αποτελεσμάτων

Η αξιολόγηση των αποτελεσμάτων αποτελεί κρίσιμο στάδιο σε κάθε πειραματική διαδικασία μηχανικής μάθησης, καθώς επιτρέπει την αντικειμενική εκτίμηση της απόδοσης ενός μοντέλου. Στην παρούσα εργασία, η αξιολόγηση βασίζεται σε ένα σύνολο στατιστικών μετρικών, οι οποίες προσφέρουν διαφορετικές οπτικές γωνίες ως προς την ακρίβεια, την ευαισθησία, τη γενίκευση και τη σταθερότητα του μοντέλου.

Κάθε μετρική έχει τον δικό της ρόλο και χρησιμότητα, ανάλογα με τα χαρακτηριστικά του προβλήματος. Ειδικότερα, χρησιμοποιούνται οι εξής βασικές μετρικές:

Απώλεια Δοκιμής

(Test Loss) Η απώλεια δοκιμής είναι ένα μέτρο που χρησιμοποιείται για να αξιολογηθεί πόσο καλά ένα μοντέλο μαθαίνει από τα δεδομένα του. Αναφέρεται στην απόκλιση μεταξύ των πραγματικών τιμών και των προβλέψεων που κάνει το μοντέλο κατά τη φάση της δοκιμής. Μια χαμηλότερη τιμή της απώλειας υποδεικνύει καλύτερη απόδοση του μοντέλου.

Η γενική μορφή της απώλειας δοκιμής είναι:

$$\text{Απώλεια Δοκιμής} = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(y_i, \hat{y}_i)$$

όπου:

- $\mathcal{L}(y_i, \hat{y}_i)$ είναι η συνάρτηση απώλειας για το i-οστό δείγμα.
- n είναι ο συνολικός αριθμός των δειγμάτων στο σύνολο δοκιμής.

Ακρίβεια

(Accuracy) Η ακρίβεια είναι το ποσοστό των σωστών προβλέψεων επί του συνολικού αριθμού των προβλέψεων. Υπολογίζεται ως το πηλίκο των σωστών προβλέψεων δια του συνολικού αριθμού των δειγμάτων. Ο τύπος της ακρίβειας είναι:

$$\text{Ακρίβεια} = \frac{\text{Αληθείς Θετικές Προβλέψεις} + \text{Αληθείς Αρνητικές Προβλέψεις}}{\text{Συνολικός Αριθμός Δειγμάτων}}$$

Η ακρίβεια παρέχει μια γενική εκτίμηση του μοντέλου, αλλά μπορεί να είναι παραπλανητική αν οι κλάσεις είναι ακανόνιστες ή αν υπάρχουν πολλά ψευδώς αρνητικά/θετικά.

Ευστοχία

(Precision) Η ευστοχία είναι η αναλογία των σωστών θετικών προβλέψεων προς το συνολικό αριθμό των θετικών προβλέψεων (αληθών και ψευδών). Υπολογίζεται ως:

$$\text{Ευστοχία} = \frac{\text{Αληθείς Θετικές Προβλέψεις}}{\text{Αληθείς Θετικές Προβλέψεις} + \Psi\text{ευδώς Θετικές Προβλέψεις}}$$

Η ευστοχία είναι σημαντική σε περιπτώσεις όπου το κόστος των ψευδώς θετικών είναι υψηλό, δηλαδή όταν το να κατατάξει το μοντέλο μια αρνητική περίπτωση ως θετική είναι επιβλαβές.

Ανάκληση

(Recall) Η ανάκληση ή ευαισθησία αναφέρεται στην ικανότητα του μοντέλου να εντοπίζει σωστά όλα τα θετικά δείγματα. Υπολογίζεται ως η αναλογία των σωστών θετικών προβλέψεων προς τον συνολικό αριθμό των πραγματικών θετικών δειγμάτων. Ο τύπος της ανάκλησης είναι:

$$\text{Ανάκληση} = \frac{\text{Αληθείς Θετικές Προβλέψεις}}{\text{Αληθείς Θετικές Προβλέψεις} + \Psi\text{ευδώς Αρνητικές Προβλέψεις}}$$

Η ανάκληση είναι σημαντική όταν το κόστος των ψευδώς αρνητικών είναι υψηλό, δηλαδή όταν το να παραλείψει το μοντέλο μια θετική περίπτωση είναι επιβλαβές.

F1 Δείκτης

(F1 Score) Ο δείκτης Φ1 είναι το αρμονικό μέσο της ευστοχίας και της ανάκλησης. Είναι ένας μόνος αριθμός που αξιολογεί την ισορροπία μεταξύ αυτών των δύο μετρικών. Υπολογίζεται ως:

$$F1\ \Delta\text{είκτης} = 2 \cdot \frac{\text{Ευστοχία} \cdot \text{Ανάκληση}}{\text{Ευστοχία} + \text{Ανάκληση}}$$

Ο F1 δείκτης είναι ιδιαίτερα χρήσιμος όταν υπάρχει ανισορροπία μεταξύ των κλάσεων, καθώς δίνει περισσότερη βαρύτητα στην ανάκληση και την ευστοχία παρά στην ακρίβεια.

Υπάρχουν πολλές ακόμα στατιστικές μετρικές για την αξιολόγηση μοντέλων μηχανικής μάθησης. Ωστόσο, επιλέξαμε τις παραπάνω μετρικές καθώς είναι οι πιο κατάλληλες για το συγκεκριμένο πρόβλημα και μας παρέχουν μια ισχυρή και ολοκληρωμένη εκτίμηση της απόδοσης του μοντέλου.

Κεφάλαιο 7

Διεξαγωγή Πειραμάτων

7.1 Πειραματική Διάταξη

Έχοντας πλέον συλλέξει όλα τα απαραίτητα δεδομένα στη μορφή που επιθυμούμε, είμαστε έτοιμοι να προχωρήσουμε στην εκτέλεση των πειραμάτων. Κύριοι στόχοι μας είναι η επίτευξη υψηλών ποσοστών επιτυχίας, παρά τους περιορισμούς του διαθέσιμου συστήματος. Κάθε πείραμα θα ακολουθεί την τυπική κατανομή 70-20-10 για τα δεδομένα (εκπαίδευση, επικύρωση, δοκιμή), ενώ θα εφαρμόζεται μηχανισμός επιλογής και αποθήκευσης του καλύτερου αποτελέσματος ανά εκτέλεση.

Επιπλέον, κάθε νευρωνικό δίκτυο θα ελεγχθεί και με το επιπρόσθετο σύνολο δεδομένων Virushare 434.

Επιπλέον, θα γίνει και μία αξιολόγηση των πιο επιτυχημένων μοντέλων σε αντοχή σε adversarial attacks, καθώς και προσπάθειες αξιοποίησης μοντέλων εκπαιδευμένων σε μικρότερα σύνολα δεδομένων για επανεκπαίδευση.

Η υλοποίηση θα πραγματοποιηθεί με Python και PyTorch. Όλες οι εικόνες θα μετατραπούν σε διαστάσεις 1024×1024, ενώ στα προεκπαιδευμένα μοντέλα θα εφαρμοστεί μείωση διαστάσεων μέσω autoencoder, ώστε να καταλήγουμε σε αναπαράσταση $256 \times 256 \times 3$.

7.1.1 Πληροφορίες Συστήματος

- **Λειτουργικό Σύστημα:** Windows, Linux (Ubuntu)
- **Πλατφόρμα:** Google Colaboratory Pro
- **GPU:** NVIDIA L4
- **RAM:** 50 GB
- **GPU RAM:** 15 GB
- **Λογισμικό:** Python

7.1.2 Λογισμικό που Χρησιμοποιήθηκε

Pytorch

Το PyTorch [71] είναι ένα δημοφιλές framework για βαθιά μάθηση (deep learning), το οποίο αναπτύχθηκε από την Facebook . Παρέχει υποστήριξη για ανάπτυξη και ευέλικτη

δημιουργία νευρωνικών δικτύων. Χρησιμοποιείται ευρέως για έρευνα και ανάπτυξη εφαρμογών τεχνητής νοημοσύνης.

Pandas

Η Pandas [72] είναι μια ισχυρή βιβλιοθήκη επεξεργασίας και ανάλυσης δεδομένων στην Python, η οποία παρέχει δομές δεδομένων . Με αυτές τις δομές, η Pandas επιτρέπει την εύκολη διαχείριση, φιλτράρισμα, ομαδοποίηση και μετατροπή μεγάλων συνόλων δεδομένων με αποτελεσματικό και ευανάγνωστο τρόπο.

Tqdm

Η tqdm [73] είναι μια ελαφριά βιβλιοθήκη αρκετά χαμηλών υπολογιστικών απαιτήσεων, για την προσθήκη γραμμών προόδου (progress bars) σε βρόχους και διαδικασίες. Βοηθά στην παρακολούθηση της προόδου και του χρόνου εκτέλεσης μεγάλων επεξεργασιών, βελτιώνοντας την εμπειρία χρήστη και επιτρέποντας την εκτίμηση του υπολοίπου χρόνου.

ViT-Pytorch

Η ViT-PyTorch είναι μια υλοποίηση των Vision Transformers (ViT) που βασίζεται στη βιβλιοθήκη PyTorch. Τα Vision Transformers αποτελούν μια σύγχρονη αρχιτεκτονική βαθιάς μάθησης που εφαρμόζει τις τεχνικές μετασχηματιστών (transformers) σε εικόνες, προσφέροντας σημαντικά πλεονεκτήματα στην επεξεργασία οπτικών δεδομένων.

Numpy

Η NumPy [74] είναι η βασική βιβλιοθήκη για επιστημονικούς υπολογισμούς στην Python. Παρέχει υποστήριξη για πολυδιάστατους πίνακες δεδομένων (arrays) και μια μεγάλη γκάμα μαθηματικών συναρτήσεων για αποδοτικούς υπολογισμούς, κάνοντας την απαραίτητη για επεξεργασία δεδομένων, στατιστική και αριθμητική ανάλυση.

PIL

Η PIL (Python Imaging Library) [75] και η ανανεωμένη της έκδοση Pillow είναι βιβλιοθήκες επεξεργασίας εικόνας στην Python. Υποστηρίζουν ανάγνωση, εγγραφή και μετατροπή εικόνων σε διάφορες μορφές, καθώς και βασικές λειτουργίες όπως αλλαγή μεγέθους, περιστροφή, φιλτράρισμα και επεξεργασία εικονοστοιχείων.

Sklearn.Metrics

Η υποενότητα sklearn.metrics της βιβλιοθήκης Scikit-learn [76] περιλαμβάνει πλήθος συναρτήσεων αξιολόγησης μοντέλων μηχανικής μάθησης. Παρέχει μετρικές όπως η accuracy, precision, recall, F1-score και άλλες, που χρησιμοποιούνται για την εκτίμηση της απόδοσης αλγορίθμων σε ταξινόμηση, παλινδρόμηση και ομαδοποίηση.

Pefile Library

Η βιβλιοθήκη pefile είναι ένα ισχυρό εργαλείο γραμμένο σε Python που επιτρέπει την ανάλυση και επεξεργασία των αρχείων Portable Executable (PE) στα Windows. Τα αρχεία PE είναι η κύρια μορφή για εκτελέσιμα προγράμματα, βιβλιοθήκες δυναμικής σύνδεσης (DLL), και άλλα αρχεία συστήματος στα Windows. Η pefile παρέχει τη δυνατότητα να διαβάζουμε τα headers και τις δομές ενός PE αρχείου, όπως τις ενότητες (sections), τις εξαρτήσεις, και άλλες σημαντικές πληροφορίες. Επίσης, επιτρέπει την τροποποίηση αυτών των δομών, γεγονός που την καθιστά χρήσιμη για διάφορες εφαρμογές, όπως το reverse engineering, τον εντοπισμό κακόβουλου λογισμικού, ή τη δημιουργία adversarial παραδείγματων [77]. Με προσοχή όμως καθώς οι μη εξειδικευμένες τροποποιήσεις PE αρχείων μπορεί να καταστρέψουν την επεκτασιμότητά τους [67] .

7.2 Περιγραφή Πειραματικής Διαδικασίας

Σε αυτή την ενότητα παρουσιάζονται τα βασικά βήματα που έγιναν σε όλα τα πειράματα για την ανίχνευση κακόβουλου λογισμικού, με σκοπό τη συγκριτική αξιολόγηση μεταξύ των Συνελικτικών Νευρωνικών Δικτύων και των Vision Transformers. Η σύγκριση πραγματοποιείται σε πολλαπλά επίπεδα, με βασικό στόχο την εκτίμηση της αποδοτικότητας διαφορετικών αρχιτεκτονικών και μεθοδολογικών προσεγγίσεων.

Αρχικά, αξιολογούνται απλές δομές νευρωνικών δικτύων σε εικόνες μεγέθους 1024× 1024, μονόχρωμες, οι οποίες έχουν παραχθεί μέσω μετατροπής εκτελέσιμων αρχείων σε μορφή εικόνας όπως περιγράφεται στο paper των Nataraj et al.(2011) [31]. Έπειτα εφαρμόζεται η τεχνική Cropping and Padding ανάλογα με το μέγεθος της εικόνας, προκειμένου οι είσοδοι στα δίκτυα να αποκτήσουν ομοιόμορφες διαστάσεις.

Στη συνέχεια, σε άλλα πειράματα οι εικόνες υφίστανται μείωση διαστάσεων σε $256 \times 256 \times 3$ με χρήση autoencoder, καθιστώντας δυνατή τη μετατροπή τους σε έγχρωμες. Ακολουθούν πειράματα τόσο με απλά όσο και με προεκπαιδευμένα μοντέλα, με στόχο τη συνολική αξιολόγηση της απόδοσής τους.

Για τα μοντέλα που επιδεικνύουν τις καλύτερες επιδόσεις, πραγματοποιούνται επιπλέον πειράματα μετεκπαίδευσης, προκειμένου να διερευνηθεί η δυνατότητα προσαρμογής τους σε νέα δεδομένα. Παράλληλα, εξετάζεται η ανθεκτικότητά τους μέσω επιθέσεων με Adversarial δείγματα.

Τέλος, αναλύονται οι επιδόσεις των μοντέλων σε διαφορετικά σύνολα δεδομένων και παραμέτρους εκπαίδευσης, με ιδιαίτερη έμφαση στην ακρίβεια και στον χρόνο εκπαίδευσης. Τα αποτελέσματα παρουσιάζονται συγκριτικά, προκειμένου να αναδειχθούν τα πλεονεκτήματα και οι περιορισμοί κάθε προσέγγισης. Επιπλέον, αξιολογείται η επίδραση παραγόντων όπως το μέγεθος των δεδομένων και οι ρυθμίσεις εκπαίδευσης στην τελική απόδοση των μοντέλων, προσφέροντας μια ολοκληρωμένη εικόνα των δυνατοτήτων τους.

Κύρια Χαρακτηριστικά Υλοποίησης για αποφυγή Overfitting

- **Πρόωρη Διακοπή (Early Stopping):** Διακόπτει την εκπαίδευση όταν δεν παρατηρείται βελτίωση στην απώλεια επικύρωσης.

- **Δυναμική Ρύθμιση Ρυθμού Μάθησης:** Προσαρμογή του learning rate μέσω scheduler όταν η απόδοση σταθεροποιείται.
- **Ολοκληρωμένη Αξιολόγηση:** Χρήση πολλαπλών μετρικών για πλήρη αποτίμηση της γενίκευσης του μοντέλου.

7.3 Αναλυτική Διαδικασία

Η διαδικασία διεξαγωγής των πειραμάτων περιλαμβάνει τα ακόλουθα στάδια:

- **Μετατροπή εκτελέσιμων αρχείων σε εικόνες**: Τα κακόβουλα και καλόβουλα δείγματα μετατράπηκαν σε εικόνες, ώστε να καταστούν κατάλληλα για επεξεργασία από νευρωνικά δίκτυα.
- **Δημιουργία συνόλων δεδομένων (datasets)**:
  - **Κακόβουλα δείγματα:**
    * Virusshare-346 (2.598 samples)
    * Virusshare-375(5.840 samples)
    * Virusshare-401(11.802 samples)
    * Virusshare-434 (18,976 samples)
  - **Καλόβουλα δείγματα:**
    * Benign Sample 1 (2.071 samples)
    * Benign Sample 2 (3.991 samples)
    * Benign Sample 3 (6.419 samples)
- **Εφαρμογή διαφορετικών μοντέλων μηχανικής μάθησης**: Τα δεδομένα τροφοδοτήθηκαν σε μοντέλα, συνελικτικών νευρωνικών δικτύων (CNN) και Vision Transformers, για τη συγκριτική αξιολόγηση της απόδοσής τους.
- **Αξιολόγηση μοντέλων με χρήση στατιστικών μετρικών**: Η απόδοση κάθε μοντέλου μετρήθηκε μέσω δεικτών όπως:
  - **Ακρίβεια** (Accuracy)
  - **Ανάκληση** (Recall)
  - **Ακρίβεια θετικών προβλέψεων** (Precision)
  - **F1 score**

Για τη διαχείριση και αποδοτική αποθήκευση των δεδομένων εκπαίδευσης, επικύρωσης και δοκιμής, αξιοποιήθηκαν αρχεία τύπου NPZ, τα οποία επιτρέπουν τη συμπιεσμένη αποθήκευση πολλαπλών πινάκων NumPy. Κάθε αρχείο περιέχει τις διαδρομές των εικόνων και τις αντίστοιχες ετικέτες, οργανωμένες ανά υποσύνολο δεδομένων. Η συγκεκριμένη προσέγγιση διασφάλισε ότι όλα τα μοντέλα εκπαιδεύτηκαν, επικυρώθηκαν και αξιολογήθηκαν με τα ίδια ακριβώς δείγματα, διατηρώντας συνέπεια και αξιοπιστία στα αποτελέσματα.

7.4 Μοντέλα

7.4.1 CNN και ViT

Όπως προαναφέρθηκε σε πρώτο στάδιο, πραγματοποιήσαμε μια βασική σειρά πειραμάτων, όπου όλα τα εκτελέσιμα αρχεία μετατράπηκαν σε εικόνες γκρι κλίμακας. Στη συνέχεια, όλες οι εικόνες προσαρμόστηκαν σε κοινή ανάλυση 1024×1024 και εκπαιδεύτηκαν χρησιμοποιώντας απλές αρχιτεκτονικές. Αρχικά, εφαρμόσαμε ένα Συνελικτικό Νευρωνικό Δίκτυο (CNN) για την ταξινόμηση, ενώ στη συνέχεια αξιοποιήσαμε ένα Vision Transformer (ViT) για την ανάλυση των δεδομένων, διερευνώντας τη συγκριτική τους απόδοση στην ανίχνευση κακόβουλου λογισμικού.

CNN

1. Το πρώτο συνελικτικό επίπεδο με πυρήνα 5 × 5 και έξοδο 15 feature maps.
2. MaxPooling επίπεδο με πυρήνα 4 × 4 και βήμα 4.
3. Δεύτερο συνελικτικό επίπεδο με πυρήνα 5 × 5 και έξοδο 25 feature maps.
4. MaxPooling επίπεδο με πυρήνα 4 × 4 και βήμα 4.
5. Το πρώτο Fully connected επίπεδο με είσοδο 25 × 62 × 62 και έξοδο 1000.
6. Το τελικό Fully Connected επίπεδο με είσοδο 1000 και έξοδο 2, τις 2 πιθανές κατηγορίες.

- Batch size: Χρησιμοποιήθηκε μέγεθος δέσμης ίσο με 32, επιλογή που προσφέρει ισορροπία μεταξύ ταχύτητας εκπαίδευσης και σταθερότητας σύγκλισης.
- Loss function: Ως συνάρτηση απώλειας επιλέχθηκε η CrossEntropyLoss, κατάλληλη για προβλήματα ταξινόμησης πολλών κατηγοριών — εδώ, δυαδική.
- Optimizer: Χρησιμοποιήθηκε ο αλγόριθμος Adagrad με learning rate ίσο με 0.001. Ο συγκεκριμένος αλγόριθμος προσφέρει δυναμική προσαρμογή του ρυθμού μάθησης ανά βάρος, ενισχύοντας την απόδοση σε αραιά χαρακτηριστικά.
- Epochs: Το μέγιστο πλήθος εποχών ορίστηκε στις 50, με δυνατότητα πρόωρης διακοπής (early stopping) αν δεν παρατηρείται βελτίωση για 10 διαδοχικές εποχές.
- Learning rate scheduler: Εφαρμόστηκε μηχανισμός προσαρμογής του ρυθμού μάθησης ανάλογα με την απώλεια επικύρωσης (ReduceLROnPlateau).
- Early stopping: Χρησιμοποιήθηκε για να αποφευχθεί υπερπροσαρμογή του μοντέλο, με patience ίσο με 10 και αυτόματη αποθήκευση του βέλτιστου μοντέλου.

ViT

1. **Patch Embedding**
   (a) Είσοδος: Εικόνα διαστάσεων $1024 \times 1024 \times 1$ (grayscale).
   (b) Διαχωρισμός της εικόνας σε patches διαστάσεων 64×64.
   (c) Κάθε patch μετατρέπεται σε διάνυσμα διαστάσεων 256 μέσω γραμμικού embedding.

2. **Position Embeddings**
   (a) Προσθήκη θέσεων embeddings στα patches για διατήρηση της χωρικής πληροφορίας.

3. **Transformer Encoder**
   (a) Το μοντέλο αποτελείται από 12 διαδοχικά Transformer Blocks, καθένα εκ των οποίων περιλαμβάνει:
      i. **Multi-Head Self Attention** με 16 κεφαλές.
      ii. **MLP Block** με διαστάσεις 512.
      iii. **Dropout** 0.1.
      iv. **Layer Normalization**.

4. **Classification Token**
   (a) Προσθήκη ενός επιπλέον learnable token ("CLS") στην είσοδο.
   (β′) Το τελικό embedding του token χρησιμοποιείται για την τελική ταξινόμηση.

5. **MLP Head**
   (a) Πλήρως συνδεδεμένο επίπεδο (Fully Connected Layer) με είσοδο 256 και έξοδο 2 (ταξινόμηση σε δύο κατηγορίες).
   (b) **Softmax** για πρόβλεψη πιθανοτήτων.

Τα παραπάνω Νευρωνικά δίκτυα χρησιμοποιήθηκαν με 3 διαφορετικά Datasets όπως έχουμε πει ότι χωρίστηκαν στην ενότητα των δεδομένων.Τα αποτελέσματα φαίνονται στο παρακάτω πινακάκι

Πίνακας 7.1: Σύγκριση Απόδοσης Μοντέλων 1ου Πειράματος CNN/ViT

Μοντέλο	Ακρίβεια	Ανάκληση	Precision	F1 Score	Απώλεια (Loss)
CNN-346	0.8726	0.8522	0.9136	0.8818	0.3112
ViT-346	0.9229	0.9194	0.9411	0.9301	0.2280
CNN-375	0.9054	0.9170	0.9233	0.9202	0.2702
ViT-375	0.9039	0.9247	0.9146	0.9196	0.3891
CNN-401	0.9259	0.9452	0.9489	0.9471	0.1995
ViT-401	0.9055	0.9531	0.9206	0.9366	0.3105

7.5 Autoencoder

Τα αποτελέσματα των παραπάνω πειραμάτων, αν και σε έναν βαθμό είναι ικανοποιητικά δεδομένης της απλότητας τους, παρουσιάζουν σημαντικά περιθώρια βελτίωσης, όπως προκύπτει και από τη σχετική βιβλιογραφία.Τόσο η ακρίβεια όσο και η ταχύτητα των μοντέλων είναι κρίσιμο να ενισχυθούν.

Για τον σκοπό αυτό αξιοποιήσαμε την τεχνική των CNN autoencoders, εκπαιδεύοντας τρία διαφορετικά μοντέλα, καθένα βασισμένο σε ένα από τα τρία διαθέσιμα datasets. Στη συνέχεια, παρουσιάζουμε αρχικά την αρχιτεκτονική του δικτύου που χρησιμοποιήθηκε και κατόπιν τα αποτελέσματα της εκπαίδευσης για κάθε περίπτωση.

7.5.1 Αρχιτεκτονική Αυτόματου Κωδικοποιητή

Η αρχιτεκτονική του αυτόματου κωδικοποιητή αποτελείται από δύο βασικά μέρη:

- **Κωδικοποιητής (Encoder):**
  - Περιλαμβάνει συνελικτικά επίπεδα (*Conv2d*) που μειώνουν τις διαστάσεις της εικόνας και εξάγουν σημαντικά χαρακτηριστικά.
  - Χρησιμοποιεί (*pooling layers*) για τη μείωση της χωρικής διάστασης.
  - Το τελικό αποτέλεσμα του encoder είναι ένας τρισδιάστατος πίνακας διαστάσεων $3 \times 256 \times 256$.

- **Αποκωδικοποιητής (Decoder):**
  - Περιλαμβάνει αντίστροφα συνελικτικά επίπεδα (*ConvTranspose2d*) για επαναφορά των διαστάσεων στην αρχική μορφή.
  - Το τελικό επίπεδο χρησιμοποιεί την ενεργοποίηση *Sigmoid*, ώστε η έξοδος να βρίσκεται στο διάστημα $[0, 1]$, κατάλληλο για εικόνες σε αποχρώσεις του γκρι.

7.5.2 Συνάρτηση Απώλειας και Βελτιστοποίηση

Για την εκπαίδευση του αυτόματου κωδικοποιητή χρησιμοποιήθηκαν:

- **Συνάρτηση Απώλειας:** Η *Mean Squared Error Loss (MSELoss)*, κατάλληλη για την αξιολόγηση της διαφοράς μεταξύ της αρχικής και της ανακατασκευασμένης εικόνας.
- **Βελτιστοποιητής:** Ο αλγόριθμος *Adam*, ο οποίος προσφέρει αποτελεσματική και σταθερή εκπαίδευση με ρυθμό εκμάθησης 1e−3 και ρύθμιση βάρους (*΄weight decay*) 1e−5.

7.5.3 Διαδικασία Εκπαίδευσης

Η διαδικασία εκπαίδευσης περιλαμβάνει τα εξής βήματα:

1. Φόρτωση των εικόνων μέσω του `DataLoader`, σε παρτίδες (*batches*).
2. Υπολογισμός της εξόδου του μοντέλου (ανακατασκευασμένη εικόνα) για κάθε παρτίδα.
3. Υπολογισμός της απώλειας μεταξύ της αρχικής και της παραγόμενης εικόνας μέσω της MSELoss.
4. Πραγματοποίηση διάδοσης προς τα πίσω (*backpropagation*) και ενημέρωση των βαρών.

7.5.4 Χρήση του Αυτόματου Κωδικοποιητή

Στο πλαίσιο της παρούσας μελέτης, ο αυτόματος κωδικοποιητής (*autoencoder*) χρησιμοποιείται για τη συμπίεση και την αναπαράσταση εικόνων από τον φάκελο. Η διαδικασία περιλαμβάνει τη φόρτωση, τη μετατροπή και τη συμπίεση των εικόνων, καθώς και την αποθήκευση των συμπιεσμένων εικόνων. Ακολουθεί η περιγραφή της μεθόδου:

1. **Φόρτωση και Προετοιμασία Εικόνων:**
   - Εφαρμόζεται η προσαρμοσμένη μετατροπή ώστε όλες οι εικόνες να έχουν διαστάσεις 1024×1024.
   - Οι εικόνες μετατρέπονται σε μορφή `Tensor` για να μπορούν να υποστούν επεξεργασία από το μοντέλο.
2. **Συμπίεση μέσω του Encoder:**
   - Το εκπαιδευμένο μοντέλο φορτώνεται από το αρχείο
   - Ο **encoder** του autoencoder εφαρμόζεται στην είσοδο, παράγοντας μια συμπιεσμένη αναπαράσταση μεγέθους $8 \times 256 \times 256$.
3. **Δημιουργία Τριών Καναλιών:**
   - Οι 8 αρχικοί δίαυλοι της συμπιεσμένης εικόνας ομαδοποιούνται ως εξής:
     (α΄) **Πρώτο κανάλι:** Υπολογίζεται ο μέσος όρος των πρώτων 3 διαύλων.
     (β΄) **Δεύτερο κανάλι:** Υπολογίζεται ο μέσος όρος των επόμενων 3 διαύλων.

(γ′) **Τρίτο κανάλι:** Υπολογίζεται ο μέσος όρος των τελευταίων 2 διαύλων.

- Τα τρία νέα κανάλια στοιβάζονται για τη δημιουργία μιας νέας εικόνας $3 \times 256 \times 256$.

4. **Κανονικοποίηση και Αποθήκευση Εικόνων:**
   - Η συμπιεσμένη εικόνα κανονικοποιείται σε τιμές μεταξύ $[0, 1]$ για οπτικοποίηση και αποθήκευση.
   - Οι εικόνες μετατρέπονται σε μορφή `png` και αποθηκεύονται στον φάκελο διατηρώντας τα αρχικά ονόματα αρχείων.

7.5.5 Πλεονεκτήματα της Μεθόδου

Η χρήση του αυτοενσοδερ παρέχει:

- **Συμπίεση Δεδομένων:** Η αρχική εικόνα μετατρέπεται σε μια πιο συμπιεσμένη μορφή, διατηρώντας σημαντικές πληροφορίες.
- **Απλοποίηση Αναπαράστασης:** Ομαδοποιώντας τα κανάλια, επιτυγχάνεται μείωση της πολυπλοκότητας της εικόνας.
- **Ευκολία Αποθήκευσης:** Οι συμπιεσμένες εικόνες είναι μικρότερες σε μέγεθος και αποθηκεύονται σε καθορισμένο φάκελο για περαιτέρω χρήση.

Πίνακας 7.2: Σύγκριση μεταξύ των τριών Autoencoders

Μοντέλο	**Epochs**	**Loss**	**Bottleneck**
Autoencoder 1	3	0.0152	$3 \times 256 \times 256$
Autoencoder 2	3	0.0138	$3 \times 256 \times 256$
Autoencoder 3	3	0.0121	$3 \times 256 \times 256$

7.6 Autoencoder with CNN/ViT

Σε αυτό το στάδιο, πραγματοποιήσαμε μια διαφορετική προσέγγιση για να υπάρχει και μία βελτίωση τόσο της απόδοσης όσο και της πολυπλοκότητας του δικτύου.Όλα τα εκτελέσιμα αφού μετατράπηκαν σε εικόνες γκρι μέσω δικτύου Autoencoder μετατράπηκαν σε έγχρωμες εικόνες $3 \times 256 \times 256$. Το δίκτυο έλαβε ως είσοδο ασπρόμαυρες εικόνες διαστάσεων $1 \times 1024 \times 1024$ και στο τέλος του **encoder** παράγετε μια συμπιεσμένη αναπαράσταση $3 \times 256 \times 256$, η οποία αξιοποιήθηκε ως έξοδος για επόμενα βήματα.

Αν και οι αρχικές εικόνες είναι μονόχρωμες, το **bottleneck** του Autoencoder σχεδιάστηκε να έχει τρία κανάλια, ώστε να λειτουργεί ως ψευδο-έγχρωμη αναπαράσταση. Αυτή η τρισδιάστατη μορφή διατηρεί χωρική πληροφορία, ενώ επιτρέπει τη χρήση μοντέλων που περιμένουν έγχρωμες εικόνες ως είσοδο, όπως τα CNN και οι Vision Transformers (ViT).

Η προσέγγιση αυτή αποδείχθηκε αποτελεσματική, καθώς διατηρεί σημαντικά χαρακτηριστικά των αρχικών εικόνων ενώ μειώνει δραστικά το μέγεθος και τη διάσταση των δεδομένων.

7.6.1 Autoencoder with CNN

1. Convolutional layer με πυρήνα 5×5, 15 feature maps, και είσοδο 3 καναλιών (RGB).
2. MaxPooling επίπεδο με πυρήνα 4×4 και βήμα 4.
3. Δεύτερο convolutional layer με πυρήνα 5×5 και 25 feature maps.
4. Δεύτερο MaxPooling επίπεδο με πυρήνα 4×4 και βήμα 4.
5. Adaptive Average Pooling για σταθερή έξοδο 14×14 ανεξάρτητα από το μέγεθος εισόδου.
6. Πλήρως συνδεδεμένο επίπεδο (Fully Connected Layer) με είσοδο $25 \times 14 \times 14$ και έξοδο 1000.
7. Τελικό Fully Connected Layer με είσοδο 1000 και έξοδο 2 malware or benign

7.6.2 Autoencoder with ViT

1. **Patch Embedding**
   (a) Είσοδος: Εικόνα διαστάσεων $256 \times 256 \times 3$ (grayscale).
   (β′) Διαχωρισμός της εικόνας σε patches διαστάσεων 32×32.
   (c) Κάθε patch μετατρέπεται σε διάνυσμα διαστάσεων 256 μέσω γραμμικού embedding.
2. **Position Embeddings**
   (a) Προσθήκη θέσεων embeddings στα patches για διατήρηση της χωρικής πληροφορίας.
3. **Transformer Encoder**
   (a) Το μοντέλο αποτελείται από 12 διαδοχικά Transformer Blocks, καθένα εκ των οποίων περιλαμβάνει:
      i. **Multi-Head Self Attention** με 16 κεφαλές.
      ii. **MLP Block** με διαστάσεις 512.
      iii. **Dropout** 0.2.
      iv. **Layer Normalization**.
4. **Classification Token**
   (a) Προσθήκη ενός επιπλέον learnable token (”CLS”) στην είσοδο.
   (β′) Το τελικό embedding του token χρησιμοποιείται για την τελική ταξινόμηση.
5. **MLP Head**

(a) Πλήρως συνδεδεμένο επίπεδο (Fully Connected Layer) με είσοδο 256 και έξοδο 2 (ταξινόμηση σε δύο κατηγορίες).

(b) **Softmax** για πρόβλεψη πιθανοτήτων.

Τα παραπάνω Νευρωνικά δίκτυα χρησιμοποιήθηκαν με 3 διαφορετικά Datasets όπως έχουμε πει οτι χωρίστηκαν στην ενότητα των δεδομένων.Τα αποτελέσματά φαίνονται στο παρακάτω πίνακα.

Πίνακας 7.3: Σύγκριση Απόδοσης Μοντέλων 2ου Πειράματος με εικόνες μειωμένες με χρήση Autoencoder

Μοντέλο	Ακρίβεια	Ανάκληση	Precision	F1 Score	Απώλεια (Loss)
Auto/CNN-346	0.8972	0.8887	0.9242	0.9061	0.2448
Auto/ViT-346	0.9101	0.9040	0.9327	0.9181	0.3079
Auto/CNN-375	0.9309	0.9547	0.9308	0.9426	0.1910
Auto/ViT-375	0.9273	0.9478	0.9311	0.9394	0.2516
Auto/CNN-401	0.9259	0.9562	0.9393	0.9407	0.1907
Auto/ViT-401	0.9265	0.9581	0.9383	0.9481	0.4139

7.7 Pretrained CNN/ViT ResNet18-Vit Base 8

Χρήση Προεκπαιδευμένων Μοντέλων

Έχοντας διαπιστώσει ότι η μείωση των διαστάσεων βελτιώνει την υπολογιστική ισχύ, τον χρόνο εκτέλεσης και την ακρίβεια επιχειρούμε να δουλέψουμε και με πιο περίπλοκα μοντέλα. Σε αυτό το στάδιο λοιπόν, αντί να δημιουργήσουμε από την αρχή τα δικά μας νευρωνικά δίκτυα, επιλέξαμε να αξιοποιήσουμε προεκπαιδευμένα μοντέλα, τα οποία έχουν ήδη εκπαιδευτεί σε μεγάλα σύνολα δεδομένων. Η χρήση τέτοιων μοντέλων επιτρέπει τη μεταφορά μάθησης (transfer learning), βελτιώνοντας τόσο την ακρίβεια όσο και την αποδοτικότητα της ανίχνευσης κακόβουλου λογισμικού.

7.7.1 ResNet-18

1. Initial Convolution: Συνελικτικό επίπεδο με πυρήνα 7×7, βήμα 2, και έξοδο 64 feature maps, ακολουθούμενο από BatchNorm και ReLU.

2. MaxPooling: Επίπεδο με πυρήνα 3×3, βήμα 2, για μείωση διαστάσεων.

3. Residual Blocks:

   - Block 1: 2 υποεπίπεδα με πυρήνες 3×3 και έξοδο 64 feature maps.
   - Block 2: 2 υποεπίπεδα με πυρήνες 3×3 και έξοδο 128 feature maps.
   - Block 3: 2 υποεπίπεδα με πυρήνες 3×3 και έξοδο 256 feature maps.
   - Block 4: 2 υποεπίπεδα με πυρήνες 3×3 και έξοδο 512 feature maps.

4. Global Average Pooling: Μετατροπή των τελικών διαστάσεων σε 1×1×512.

5. Fully Connected Layer: Επεκταμένο δίκτυο:

   - Γραμμικό επίπεδο (512 → 256).
   - Ενεργοποίηση ReLU.
   - Dropout 0.2
   - Τελικό γραμμικό επίπεδο (256 → 2).

Αν και η βασική αρχιτεκτονική (στρώματα συνέλιξης) παραμένει αμετάβλητη, διατηρώντας τις προεκπαιδευμένες ικανότητες εξαγωγής χαρακτηριστικών έχουμε αλλάξει το input/output στα δικά μας ζητούμενα [78]

7.7.2 ViT base patch 8

1. **Patch Embedding**

   (a) Είσοδος: Εικόνα διαστάσεων 224 × 224 × 3 224×224×3 (RGB).

   (b) Διαχωρισμός της εικόνας σε (224/8)2=784(224/8) 2=784 patches διαστάσεων 8 × 8 8×8.

   (c) Κάθε patch (8×8×3=192 8×8×3=192 pixels) προβάλλεται σε διάνυσμα 768 διαστάσεων μέσω γραμμικού μετασχηματισμού.

2. **Position Embeddings**

   (a) Προσθήκη positional embeddings στα διανύσματα των patches για διατήρηση της χωρικής τους πληροφορίας.

3. **Transformer Encoder**

   (a) 12 επαναληπτικά στρώματα Transformer, καθένα περιλαμβάνει:

      i. **Multi-Head Self Attention (MSA)**: 12 κεφαλές (εμβέλεια 768/12 = 64 διαστάσεων ανά κεφαλή).

      ii. **MLP Block**: Δίκτυο 2 γραμμικών επιπέδων (768 → 3072 → 768) με ενεργοποίηση GELU.

      iii. Layer Normalization πριν από κάθε υποστρώμα (Pre-LN).

4. **Classification Token ("[CLS]")**

   (a) Προσθήκη ενός διανύσματος (token) στην ακολουθία των patches.

   (b) Το τελικό embedding του CLS token (768 διαστάσεις) χρησιμοποιείται για την ταξινόμηση.

5. **Classification Head**

   (a) Γραμμικό επίπεδο που αντιστοιχεί το CLS token embedding (768 → num_classes).

   (b) Συνάρτηση Softmax για την πρόβλεψη πιθανοτήτων κατηγοριοποίησης.

Αντί για εικόνες 224 × 224 pixels, χρησιμοποιήσαμε εικόνες 256 × 256 pixels.

Τροποποίηση Classifier Head: Αντικαταστήσαμε το απλό γραμμικό επίπεδο ταξινόμησης του ViT με ένα πιο πολύπλοκο δίκτυο και τελικό γραμμικό επίπεδο ($512 \rightarrow 2$ κλάσεις) [64].

Όπως και με τις προηγούμενες δύο αρχιτεκτονικές νευρωνικών δικτύων και εδώ έγινε εκπαίδευση με τα ίδια Datasets

Πίνακας 7.4: Σύγκριση Απόδοσης Μοντέλων

Μοντέλο	Ακρίβεια	Ανάκληση	Precision	F1 Score	Απώλεια (Loss)
ResNet-18(346)	0.9743	0.9770	0.9770	0.9770	0.1078
ViT B/8-346	0.9572	0.9655	0.9581	0.9618	0.1793
ResNet-18(375)	0.9649	0.9670	0.9653	0.9707	0.1440
ViT B/8-375	0.9649	0.9812	0.9606	0.9708	0.1390
ResNet-18(401)	0.9660	0.9789	0.9728	0.9758	0.1385
ViT B/8-401	0.9665	0.9840	0.9688	0.9763	0.1489

7.8 Retraining

Η διαδικασία επανεκπαίδευση νευρωνικών δικτύων αποτελεί μια ιδιαίτερα σημαντική τεχνική στον τομέα της μηχανικής μάθησης, δεδομένου ότι μέσο αυτής δίνετε η δυνατότητα βελτιστοποίησης της απόδοσης ενός ήδη εκπαιδευμένου μοντέλου χωρίς να απαιτείται πλήρης εκπαίδευση από την αρχή όπως κάναμε και με τα ResNet-18, ViT(B/8). Μέσω της συγκεκριμένης διαδικασίας, μπορούν να επιτευχθούν καλύτερα αποτελέσματα σε νέα δεδομένα ή σε συνθήκες που αλλάζουν με την πάροδο του χρόνου, ενώ ταυτόχρονα μειώνεται ο υπολογιστικός φόρτος χρόνος επεξεργασίας σε σύγκριση με την εκπαίδευση από την αρχής.

Ωστόσο, η επανεκπαίδευση δεν είναι πάντα η βέλτιστη επιλογή. Ιδίως σε περιπτώσεις όπου το μοντέλο έχει ήδη φτάσει σε υψηλό επίπεδο απόδοσης, η παρέμβαση ενδέχεται να προκαλέσει υποβάθμιση της απόδοσης (performance degradation), υπερπροσαρμογή (overfitting) ή ακόμη και μείωση της γενικευτικής του ικανότητας. Επιπλέον, η διαδικασία αυτή απαιτεί λεπτούς χειρισμούς, προσεκτική επιλογή παραμέτρων και κατανόηση των μηχανισμών μεταφοράς μάθησης. Σε περιβάλλοντα όπου η διαθεσιμότητα νέων δεδομένων είναι περιορισμένη ή όταν οι διαφορές μεταξύ των νέων και παλαιών δεδομένων είναι μικρές, η επανεκπαίδευση μπορεί να μην προσφέρει ουσιαστικό όφελος και, σε ορισμένες περιπτώσεις, να είναι αντιπαραγωγική.

Σύμφωνα με τους Yosinski et [79] al. (2014), σε πειράματα που πραγματοποιήθηκαν με convolutional neural networks (CNN) σε datasets εικόνων, παρατηρήθηκε ότι η επανεκπαίδευση των πρώτων στρωμάτων του δικτύου (τα οποία μαθαίνουν γενικά χαρακτηριστικά, όπως ακμές και χρώματα) είχε μικρό ή και αρνητικό αντίκτυπο στην τελική απόδοση, σε αντίθεση με την επανεκπαίδευση των τελευταίων στρωμάτων που είχε σαφώς θετικό αποτέλεσμα σε συγκεκριμένες περιπτώσεις. Αυτό το εύρημα υπογραμμίζει τη σημασία του να γνωρίζουμε ποια μέρη του δικτύου πρέπει να επανεκπαιδεύονται και πότε.

Συνεπώς, η απόφαση για επανεκπαίδευση ενός νευρωνικού δικτύου πρέπει να λαμβάνεται με βάση τεχνικά κριτήρια, όπως η φύση των νέων δεδομένων, και το κόστος

υπολογιστικών πόρων. Δεν είναι μια βέβαιη έτοιμη λύση που εφαρμόζεται πάντα, αλλά μια στρατηγική που απαιτεί κατανόηση, πειραματισμό και σωστή ερμηνεία των αποτελεσμάτων.

Στην παρούσα εργασία πραγματοποιήθηκαν διαφορετικές στρατηγικές επανεκπαίδευσης (fine-tuning) σε προεκπαιδευμένο μοντέλο Vision Transformer (ViT):

1. **Επανεκπαίδευση μόνο του ταξινομητή (classifier head):**
   Αρχικά, παγώθηκαν όλα τα υπόλοιπα layers του μοντέλου (`param.requires_grad = False`) και εκπαιδεύτηκε μόνο ο ταξινομητής (`param.requires_grad = True` μόνο για τον head).

2. **Μερική επανεκπαίδευση (partial fine-tuning):**

   - Παγώθηκαν όλα τα layers του backbone του μοντέλου (Vision Transformer).
   - Ξεπαγώθηκαν μόνο τα δύο τελευταία transformer blocks (block 10 και 11).
   - Ο ταξινομητής (classifier head) αντικαταστάθηκε και εκπαιδεύτηκε από την αρχή.
   - Χρησιμοποιήθηκαν δύο διαφορετικά learning rates: χαμηλό για τα ξεπαγωμένα blocks και υψηλότερο για τον head.

3. **Πλήρης επανεκπαίδευση:**
   Όλο το μοντέλο εκπαιδεύτηκε κανονικά από άκρη σε άκρη, με μικρό learning rate για σταθερή και προσεκτική προσαρμογή.

- 401: Εκπαίδευση μόνο στο dataset 401.
- (375,401): Πρώτα εκπαίδευση στο 375 και στη συνέχεια fine-tuning στο 401.
- H: Fine-tuning μόνο στο ¨κεφάλι' του μοντέλου (head).
- H,B : Fine-tuning στο ¨κεφάλι' και μέρος του ¨σώματος' (body).
- Full : Fine-tuning σε όλο το δίκτυο.

Ανάλυση Αποτελεσμάτων

Τα αποτελέσματα δείχνουν:

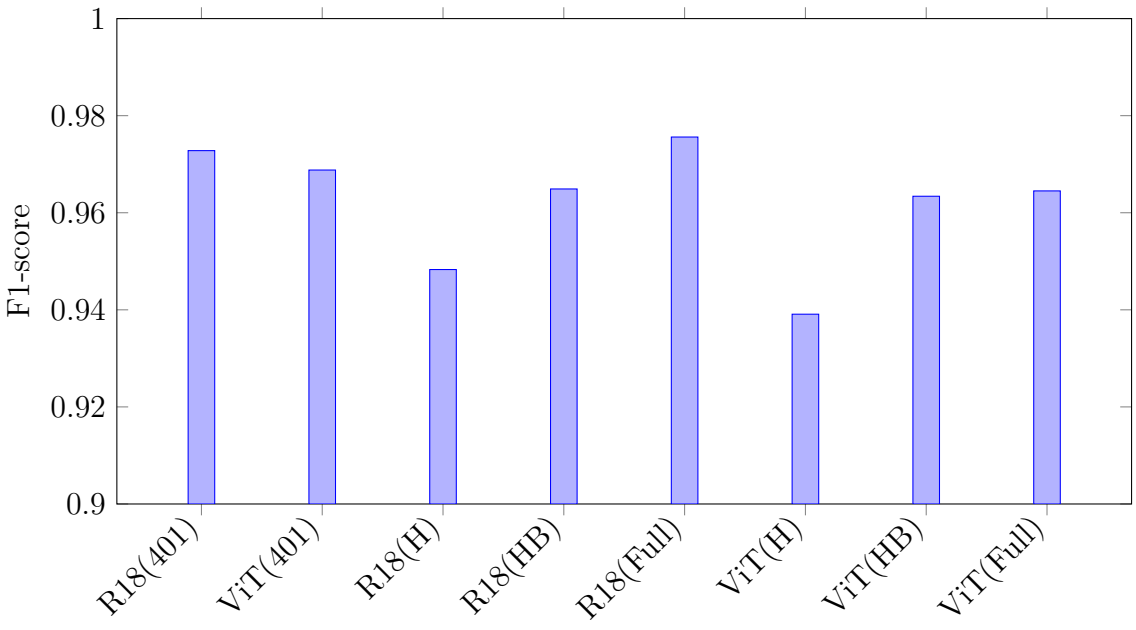
- **ViT base 8 (401)** πέτυχε το υψηλότερο **Recall (0.9840)** και συνολικά πολύ καλή απόδοση.
- Τα μοντέλα που εκπαιδεύτηκαν μόνο στο 401 είχαν ελαφρώς καλύτερη απόδοση σε Precision και F1-score από τα μοντέλα με προηγούμενη εκπαίδευση στο 375.
- Το πλήρες fine-tuning(**Full**) σε ResNet και ViT παρουσίασε τις υψηλότερες συνολικές επιδόσεις, αποδεικνύοντας τη σημασία της πλήρους επανεκπαίδευσης.

Πίνακας 7.5: Σύγκριση Απόδοσης Μοντέλων Επανεκπαίδευσης

Μοντέλο	Ακρίβεια	Ανάκληση	Precision	F1 Score	(Loss)
ResNet18(401)	0.9660	0.9789	0.9728	0.9758	0.1385
ViT base 8(401)	0.9665	0.9840	0.9688	0.9763	0.1489
ResNet18(375,401)H	0.9405	0.9679	0.9483	0.9580	0.1687
ResNet18(375,401)H,B	0.9591	0.9773	0.9649	0.9394	0.9710
ResNet18(375,401)Full	0.9624	0.9797	0.9756	0.9731	0.1386
ViT base 8(375,401)H	0.9240	0.9534	0.9391	0.9462	0.1934
ViT base 8(375,401)H,B	0.9512	0.9671	0.9634	0.9672	0.1282
ViT base 8(375,401)Full	0.9599	0.9789	0.9645	0.9717	0.1246

- Το fine-tuning μόνο του **head** είχε τα χειρότερα αποτελέσματα, κάτι που δείχνει ότι δεν είναι επαρκές όταν αλλάζει το dataset.
- Η μέθοδος **H,B** (κεφάλι και σώμα) έδωσε καλή ισορροπία μεταξύ ταχύτητας εκπαίδευσης και απόδοσης.

Διάγραμμα F1-score ανά Μοντέλο



7.9 Ensemble Model

Επεξήγηση Διαδικασίας

Εκτελούμε αξιολόγηση με χρήση των μοντέλων (ResNet-18, Vision Transformer), εφαρμόζοντας μέθοδο ensemble (συνδυασμού) των δύο μοντέλων ως εξής:

- **Ensemble Voting:** Υλοποιείται κλάση για συνδυασμό προβλέψεων πολλαπλών μοντέλων με χρήση *soft voting*.

  Σχόλιο: Στο soft voting το ensemble επιλέγει την κατηγορία που συγκεντρώνει τις περισσότερες πιθανότητες από τα επιμέρους μοντέλα. Το soft voting βασίζεται στον μέσο όρο των πιθανοτήτων πρόβλεψης των μοντέλων, επιτρέποντας μια πιο «μαλακή» και ενδεχομένως ακριβέστερη συνένωση, όταν οι μοντέλα παρέχουν καλά εκτιμώμενες πιθανότητες.

- **Αξιολόγηση:**
  - Οι προβλέψεις γίνονται στο σύνολο δοκιμής χρησιμοποιώντας το ensemble.
  - Υπολογίζονται οι βασικές μετρικές απόδοσης: *precision*, *recall*, *F1-score*.
  - Εκτυπώνεται πίνακας σύγχυσης (*confusion matrix*).

Table 7.6: Συνολική Ακρίβεια Μοντέλων στο Dataset 434

Μοντέλο	**Accuracy (%)**
ResNet-18	97.27
ViT (B/8)	97.18
Ensemble	**97.87**

Table 7.7: Απόδοση Μοντέλων στο Dataset 401

Μοντέλο	**Accuracy**	**Precision**	**Recall**	**F1 Score**
ResNet-18	0.9660	0.9728	0.9789	0.9758
ViT	0.9665	0.9688	0.9840	0.9763
Ensemble	0.9680	0.9732	0.9812	0.9772

Πίνακας Σύγχυσης (Ensemble):

$$\begin{bmatrix} 1020 & 69 \\ 48 & 2508 \end{bmatrix}$$

Τα αποτελέσματα δείχνουν ότι και τα δύο μοντέλα, ResNet-18 και Vision Transformer, επιτυγχάνουν πολύ υψηλή απόδοση ξεχωριστά, με ποσοστά ακρίβειας πάνω από 96.5%. Η ελαφρώς υψηλότερη απόδοση του ViT όσον αφορά το recall (0.9840 έναντι 0.9789 του ResNet-18) υποδηλώνει καλύτερη ικανότητα ανίχνευσης θετικών δειγμάτων (κακόβουλου λογισμικού), ενώ το ResNet-18 παρουσιάζει ελαφρώς καλύτερο precision, δηλαδή λιγότερα ψευδώς θετικά.

Η εφαρμογή της μεθόδου ensemble με soft voting οδηγεί σε μια μικρή αλλά σημαντική βελτίωση στην τελική ακρίβεια (96.8% στο dataset 401+Benign3(+0.2%) και 97.87%) στο 434(+0.6%), καθώς και στις βασικές μετρικές απόδοσης, επιβεβαιώνοντας τη συμπληρωματικότητα των δύο μοντέλων. Ο πίνακας σύγχυσης δείχνει μειωμένα λάθη κατηγοριοποίησης, γεγονός που επιβεβαιώνει την ενίσχυση της αξιοπιστίας του συστήματος μέσω του ensemble.

Η βελτίωση μέσω του ensemble υπογραμμίζει τη δύναμη του συνδυασμού διαφορετικών αρχιτεκτονικών, καθώς κάθε μοντέλο μπορεί να καλύψει τις αδυναμίες του άλλου, οδηγώντας σε πιο σταθερή και γενικεύσιμη απόδοση σε δεδομένα δοκιμής.

Η διαφορά στην ακρίβεια μεταξύ των μοντέλων και του ensemble παραμένει μικρή, ωστόσο οι βελτιώσεις αυτές όταν αναφερόμαστε σε κακόβουλες ενέργιες μπορεί να είναι ζωτικής σημασίας.

7.10 Adversarials

Στο πλαίσιο της παρούσας εργασίας, εξετάζονται και διαφορετικές τεχνικές δημιουργίας adversarial examples με στόχο να μελετηθεί η ανθεκτικότητα των εκπαιδευμένων μοντέλων απέναντι σε στοχευμένες παραποιήσεις εισόδου. Οι επιθέσεις αυτού του τύπου επιχειρούν να παραπλανήσουν τα μοντέλα μηχανικής μάθησης μέσω ελάχιστων αλλά καίριων μεταβολών στα δεδομένα εισόδου, οδηγώντας σε λανθασμένη ταξινόμηση ή δραματική μείωση της ακρίβειας.

Οι adversarial attacks αποτελούν σήμερα ένα ενεργό πεδίο έρευνας καθώς έχουν αποδειχθεί ικανές να επηρεάσουν ακόμη και μοντέλα υψηλής ακρίβειας. Στη βιβλιογραφία υπάρχουν σημαντικά έργα που περιγράφουν τέτοιες τεχνικές κυρίως για εικόνες. Ωστόσο, η μεταφορά τέτοιων τεχνικών σε binary executable files παραμένει ένα δύσκολο πρόβλημα λόγω της αυστηρής δομής και της ανάγκης διατήρησης της λειτουργικότητας.

Για τον σκοπό αυτό, αναπτύχθηκαν και εφαρμόστηκαν απλοποιημένες αλλά ρεαλιστικές μέθοδοι τροποποίησης αρχείων τύπου PE (Portable Executable), με στόχο τη δημιουργία adversarial samples που δεν διαταράσσουν την εκτελεσιμότητα των αρχείων, αλλά επηρεάζουν τη συμπεριφορά του ταξινομητή.

7.10.1 Απλή προσθήκη μηδενικών

Στη συγκεκριμένη μέθοδο, τροποποιούμε αρχεία τύπου PE (Portable Executable), προσθέτοντας μια νέα ενότητα (section) και ενσωματώνοντας επιπλέον δεδομένα στο τέλος του αρχείου. Συγκεκριμένα, το πρόγραμμα διαβάζει το περιεχόμενο κάθε αρχείου PE file και το επεκτείνει, προσθέτοντας μια προκαθορισμένη ακολουθία από μηδενικά bytes στο τέλος του αρχείου ή στην προστιθέμενη ενότητα.

Αν και η συγκεκριμένη αλλαγή δεν επηρεάζει τη λειτουργικότητα του εκτελέσιμου, αποτελεί μια μορφή τεχνητής παραποίησης του περιεχομένου με σκοπό την παραπλάνηση του μοντέλου ταξινόμησης. Τέτοιες τεχνικές χρησιμοποιούνται συχνά στο πλαίσιο malware injection, είτε για να αποκρύψουν την κακόβουλη φύση του αρχείου είτε για να παρακάμψουν στατικούς ανιχνευτές υπογραφών.

Στο πλαίσιο αυτής της εργασίας, γίνεται χρήση απλής προσθήκης μηδενικών (zero-padding) με στόχο να ελεγχθεί η ανθεκτικότητα των εκπαιδευμένων μοντέλων απέναντι σε στοιχειώδεις μορφές παραποίησης που διατηρούν ανέπαφη τη συμπεριφορά του αρχείου κατά την εκτέλεση.

7.10.2 Προσθήκη Βιβλιοθηκών

Η δεύτερη τεχνική που εφαρμόζεται αφορά την προσθήκη νέων Dynamic-Link Libraries (DLLs) στο πεδίο εισαγόμενων βιβλιοθηκών ενός εκτελέσιμου. Οι βιβλιοθήκες αυτές μπορεί να είναι είτε πραγματικές, χρησιμοποιούμενες από άλλα προγράμματα, είτε ψευδείς και άνευ ουσιαστικής λειτουργικότητας, με σκοπό να αλλοιώσουν τη δομή του αρχείου χωρίς να επηρεάζουν την εκτέλεσή του.

Η συγκεκριμένη τεχνική αποτελεί περισσότερο εξελιγμένη μορφή παραποίησης, καθώς απαιτεί γνώση της εσωτερικής δομής των PE headers και μπορεί να επηρεάσει χαρακτηριστικά που λαμβάνονται υπόψη από τους ταξινομητές, όπως οι πίνακες εισαγωγών και το entropy profile του αρχείου. Η μέθοδος στοχεύει στη δημιουργία benign-looking adversarial variants, με έμφαση στην απόκρυψη μέσω ψευδούς συνάφειας.

7.10.3 Συνδυασμός Μεθόδων

Στην τρίτη υποενότητα επιχειρείται ο συνδυασμός των δύο παραπάνω τεχνικών σε ένα ενιαίο αρχείο και η προσθήκη κώδικα από καλόβουλα αρχεία στο τέλος του κάθε αρχείου. Πιο συγκεκριμένα, σε κάθε PE αρχείο προστίθενται ταυτόχρονα μηδενικά bytes καθώς και επιπλέον DLL entries, με στόχο να αυξηθεί η πολυπλοκότητα της παραποίησης.

Αυτός ο συνδυασμός αποσκοπεί στο να δοκιμάσει την ανθεκτικότητα των μοντέλων απέναντι σε πολύπλευρες επιθέσεις που αλλοιώνουν ταυτόχρονα διαφορετικά επίπεδα της δομής του εκτελέσιμου. Όπως δείχνουν και τα αποτελέσματα, η απόδοση των μοντέλων μειώνεται σημαντικά – ιδίως στην περίπτωση των Vision Transformers (ViT) – φανερώνοντας την ανάγκη για μελλοντική εστίαση σε στρατηγικές άμυνας έναντι σύνθετων adversarial manipulations.

Method	CNN	ViT
434	0.9726	0.9717
Add zeros_00	0.9880	0.9877
Add Libraries	0.9751	0.9754
2 Methods	0.9519	0.7465

Table 7.8: Adversarial

Αφού έχουν πλέον ολοκληρωθεί όλα τα πειράματα και έχουμε στη διάθεσή μας τα αντίστοιχα αποτελέσματα, στο επόμενο κεφάλαιο προχωρούμε στην αναλυτική τους αξιολόγηση. Εξετάζουμε τη συμπεριφορά των μοντέλων μας υπό διαφορετικές συνθήκες, συγκρίνουμε επιδόσεις, και επιχειρούμε να ερμηνεύσουμε τα αποτελέσματα βάσει των θεωρητικών προσδοκιών αλλά και της πρακτικής συμπεριφοράς των μεθόδων που εφαρμόσαμε.

Κεφάλαιο 8

Συγκριτική Μελέτη και Συμπεράσματα

Συνοπτικά, όπως αναφέρθηκε παραπάνω, οι Vision Transformers (ViT) και τα Convolutional Neural Networks (CNN) αποτελούν δύο βασικούς τύπους τεχνολογιών βαθιάς μάθησης που εφαρμόζονται στην οπτική υπολογιστική (computer vision) για την επίλυση παρόμοιων προβλημάτων, όπως η ταξινόμηση εικόνων και η ανίχνευση αντικειμένων. Ωστόσο, η προσέγγισή τους σε αυτά τα προβλήματα διαφέρει σημαντικά.

Τα Convolutional Neural Networks (CNN) θεωρούνται η κλασική μέθοδος στην οπτική υπολογιστική. Βασίζονται σε συνελικτικά επίπεδα που εξάγουν χαρακτηριστικά της εικόνας σε ιεραρχική δομή. Οι συνελικτικές στρώσεις εφαρμόζουν φίλτρα σε όλη την εικόνα, αναγνωρίζοντας τοπικά χαρακτηριστικά όπως άκρες, γωνίες και σχήματα. Λόγω της απλότητας της αρχιτεκτονικής τους, είναι υπολογιστικά πιο αποδοτικά, ειδικά όταν η ποσότητα των δεδομένων εκπαίδευσης είναι περιορισμένη.

Αντίθετα, οι Vision Transformers (ViT) αποτελούν μια νεότερη προσέγγιση που έχει αναδειχθεί τα τελευταία χρόνια ως ιδιαίτερα αποτελεσματική. Σπάσουν την εικόνα σε μικρότερα τμήματα (patches) και χρησιμοποιούν μηχανισμούς προσοχής (attention mechanisms) για να κατανοήσουν τις αλληλεπιδράσεις μεταξύ των τμημάτων αυτών καθώς και ολόκληρης της εικόνας. Αυτό επιτρέπει στο ViT να έχει μια πιο ολιστική και ολοκληρωμένη αντίληψη των σχέσεων μέσα στην εικόνα, γεγονός που το καθιστά εξαιρετικά χρήσιμο για πολύπλοκες εργασίες. Ωστόσο, η εκπαίδευση των ViT συχνά απαιτεί μεγαλύτερους υπολογιστικούς πόρους και περισσότερα δεδομένα σε σύγκριση με τα CNN.

Σε αυτό το κεφάλαιο θα παρουσιάσουμε αναλυτικά και με απλό τρόπο τα αποτελέσματα των πειραμάτων, συγκρίνοντας τα δίπλα δίπλα ώστε να κατανοήσουμε καλύτερα τις διαφορές και τις ομοιότητες μεταξύ των μοντέλων.

8.1 Αξιολόγηση Πειραμάτων

8.1.1 Αποτελέσματα

Μία από τις ποιο βασικές μετρήσεις της επιτυχίας ενός μοντέλου είναι η ακρίβεια πρόβλεψης των αποτελεσμάτων. Στην ανίχνευση κακόβουλου λογισμικού ιδιαίτερη σημασία έχει η ανίχνευση κακόβουλου έναντι του καλόβουλου λογισμικού καθώς η λαθεμένη πρόβλεψη κακόβουλου ως καλόβουλου μπορεί να έχει ολέθριες συνέπειες για το σύστημά μας ενώ το αντίθετο αν και όχι επιθυμητό μπορεί να επιλυθεί. Τα

συνολικά αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν παρουσιάζονται στον παρακάτω πίνακα για ευκολία.

Πίνακας 8.1: Σύγκριση Απόδοσης Μοντέλων

Μοντέλο	**Ακρίβεια**	**Ανάκληση**	**Precision**	**F1 Score**	**Απώλεια (Loss)**
CNN-346	0.8726	0.8522	0.9136	0.8818	0.3112
Auto/CNN-346	0.8972	0.8887	0.9242	0.9061	0.2448
ResNet-18(346)	0.9743	0.9770	0.9770	0.9670	0.1078
ViT-346	0.9229	0.9194	0.9411	0.9301	0.2280
Auto/ViT-346	0.9101	0.9040	0.9327	0.9181	0.3079
ViT B/8-346	0.9572	0.9655	0.9581	0.9618	0.1793
CNN-375	0.9054	0.9170	0.9233	0.9202	0.2702
Auto/CNN-375	0.9309	0.9547	0.9308	0.9426	0.1910
ResNet-18(375)	0.9649	0.9670	0.9653	0.9707	0.1440
ViT-375	0.9039	0.9247	0.9146	0.9196	0.3891
Auto/ViT-375	0.9456	0.9681	0.9680	0.9634	0.1517
ViT B/8-375	0.9449	0.9812	0.9606	0.9708	0.1390
CNN-401	0.9259	0.9452	0.9489	0.9471	0.1995
Auto/CNN-401	0.9259	0.9562	0.9383	0.9407	0.1907
ResNet-18(401)	0.9660	0.9789	0.9728	0.9758	0.1385
ViT-401	0.9055	0.9531	0.9206	0.9366	0.3105
Auto/ViT-401	0.9265	0.9581	0.9383	0.9481	0.4139
ViT B/8-401	0.9665	0.9840	0.9688	0.9763	0.1489

Αναλυτική Ανάλυση Αποτελεσμάτων Πειραμάτων

Στην παρούσα μελέτη λοιπόν συνοπτικά διερευνήθηκε η απόδοση διαφόρων αρχιτεκτονικών μοντέλων μηχανικής μάθησης σε τρία σύνολα δεδομένων διαφορετικού μεγέθους:

- **Dataset-346**: Μικρότερο σύνολο
- **Dataset-375**: Μεσαίου μεγέθους
- **Dataset-401**: Μεγαλύτερο σύνολο

Βασικά Ευρήματα

1. Επίδραση Μεγέθους Δεδομένων

- **Κλασικά Μοντέλα (CNN/ViT)**: Βελτιώνουν την απόδοση με αύξηση του μεγέθους δεδομένων

$$\text{CNN-346 (F1: 88.18\%)} \rightarrow \text{CNN-401 (F1: 94.71\%)}$$
$$\text{ViT-346 (F1: 93.01\%)} \rightarrow \text{ViT-401 (F1: 93.66\%)}$$

- **Προηγμένα Μοντέλα (ResNet/ViT B/8)**: Διατηρούν υψηλή απόδοση ακόμα και στο μικρό dataset-346

$$\text{ResNet-18 (346): F1} = 96.70\%$$
$$\text{ViT B/8 (346): F1} = 96.18\%$$

2. Σύγκριση Αρχιτεκτονικών

Πίνακας 8.2: Μέσος όρος F1 Score ανά τύπο μοντέλου

Τύπος Μοντέλου	Μέσο F1 (%)
ResNet-18	97.45
ViT B/8	**97.63**
Auto/ViT	94.59
Auto/CNN	92.98
Βασικό ViT	92.89
Βασικό CNN	91.64

3. Βέλτιστες Επιδόσεις ανά Σύνολο Δεδομένων

Πίνακας 8.3: Κορυφαία μοντέλα ανά dataset

Dataset	Βέλτιστο Μοντέλο	F1 (%)
346	ResNet-18	96.70
375	ViT B/8	97.08
401	ViT B/8	**97.63**

Σημαντικές Παρατηρήσεις

Αποτελεσματικότητα Τεχνικών με Κωδικοποιητές (Autoencoders)

- Οι **Autoencoder** βελτιώνουν σημαντικά τα βασικά μοντέλα (+4.38% F1 για ViT-375)
- Ωστόσο, δεν φτάνει την απόδοση ειδικευμένων αρχιτεκτονικών (Κατώτερη κατά 3.04% από ViT B/8)

Ευστάθεια Εκπαίδευσης

- **ResNet-18**: Εξαιρετικά σταθερό (loss ˜ 0.15 σε όλα τα datasets)
- **Auto/ViT-401**: Εμφανίζει αστάθεια (Μέγιστη απώλεια: 0.4139)

Συμπεράσματα και Προτάσεις

1. Για μικρά σύνολα δεδομένων (dataset-346):
    - Προτείνεται **ResNet-18** (F1: 96.70%, loss: 0.1078)
2. Για μεσαία/μεγάλα σύνολα (datasets 375, 401):

- Προτείνεται **ViT B/8** (Μέσος F1: 97.36%)

3. Τα βασικά CNN/ViT μοντέλα απαιτούν βελτιστοποίηση για ανταγωνιστική απόδοση

8.2 Έλεγχος σε διαφορετικά Dataset(434)

Πίνακας 8.4: Ακρίβεια ταξινόμησης (%) σε κάθε σύνολο δεδομένων ανά μοντέλο

Μοντέλο	Ακρίβεια (%)
ResNet-18(346)	97.19
ViT B/8-346	94.63
ResNet-18(375)	72.61
ViT B/8-375	96.84
ResNet-18(401)	97.27
ViT B/8-401	97.18
Ensemble	**97.88**

Αξιοσημείωτο είναι το γεγονός ότι το μοντέλο ResNet-18(346), παρότι εκπαιδεύτηκε στο μικρότερο υποσύνολο δεδομένων, πέτυχε εξαιρετικά υψηλή συνολική ακρίβεια. Ωστόσο, η απόδοσή του σε καλόβουλα δείγματα παρουσίασε σημαντική υστέρηση. Συγκεκριμένα, στο σύνολο Benign2 η ακρίβεια του υποχώρησε στο 72.16%, τη στιγμή που όλα τα υπόλοιπα μοντέλα κατέγραψαν επιδόσεις άνω του 94%. Το εύρημα αυτό υποδηλώνει ότι, παρά την υψηλή συνολική επίδοση, το συγκεκριμένο μοντέλο ενδέχεται να παρουσιάζει μειωμένη γενίκευση σε συγκεκριμένες υποκατηγορίες των δεδομένων.

Πίνακας 8.5: Μέσος όρος F1 Score ανά τύπο μοντέλου

Τύπος Μοντέλου	Μέσο F1 (%)
ResNet-18	97.45
ViT B/8	**97.63**
Auto/ViT	94.59
Auto/CNN	92.98
Βασικό ViT	92.89
Βασικό CNN	91.64

Λανθασμένες Ανιχνεύσεις Καλόβουλων ως Χαμηλής Εμπιστοσύνης:

Ιδιαίτερο ενδιαφέρον παρουσιάζει το γεγονός ότι όλα τα μοντέλα εμφανίζουν κοινή συμπεριφορά ως προς την εσφαλμένη ταξινόμηση ορισμένων καλόβουλων δειγμάτων. Συγκεκριμένα, παρατηρείται ότι επαναλαμβανόμενα ταξινομούν συγκεκριμένα εκτελέσιμα αρχεία ως κακόβουλα, με ιδιαίτερα υψηλό ποσοστό εμπιστοσύνης.

Παρακάτω παρατίθενται χαρακτηριστικά παραδείγματα τέτοιων περιπτώσεων:

Χαρακτηριστικά Παραδείγματα:

- xampp-start-processed.png — Ταξινομήθηκε ως κακόβουλο με εμπιστοσύνη **99.99%**

Πιθανές Λανθασμένες Θετικές Ενδείξεις σε Νόμιμο Λογισμικό: Το μοντέλο ταξινόμησε πολλά γνωστά και έγκυρα λογισμικά ως κακόβουλα, με εξαιρετικά υψηλά ποσοστά εμπιστοσύνης:

- Chrome installers: **99.89%**
- Git installers: **98.92%** – **99.68%**
- AutoCAD installers: **99.73%** – **99.91%**
- Microsoft SQL Server components: **76.51%** – **99.60%**

Παρατηρούμενα Μοτίβα Συμπεριφοράς:

- Τα downloaders και γενικότερα πακεταρισμένα εκτελέσιμα αρχεία φαίνεται να χαρακτηρίζονται συστηματικά ως κακόβουλα.
- Εργαλεία ανάπτυξης της Microsoft παρουσιάζουν προβλέψεις με εμπιστοσύνη στην περιοχή **75%** – **90%**.
- Αρχεία που σχετίζονται με λογισμικό προστασίας (όπως το Avast Antivirus) χαρακτηρίστηκαν επίσης ως κακόβουλα.

8.3 Αναλυτική Παρουσίαση Πιο Επιτυχημένων Μοντέλων

Τα μοντέλα ResNet-18 και ViT(B/8) που εκπαιδεύτηκαν στο μεγαλύτερο πλήθος δεδομένων είναι και αυτά που πέτυχαν τα μεγαλύτερα ποσοστά επιτυχίας. Έχοντας στη διάθεσή μας τις κατηγορίες των κακόβουλων αρχείων που περιλαμβάνονταν στα εν λόγω δεδομένα, παρουσιάζουμε στους παρακάτω πίνακες τα ποσοστά ανίχνευσης κακόβουλων αρχείων ανά κατηγορία, καθώς και τις περιπτώσεις όπου και τα δύο μοντέλα είχαν την καλύτερη απόδοση.

Category	Overall %	Correctly Classified %	Wrongly Classified %
null	23.09	32.89	61.11
other	33.21	31.45	33.3
Trojan-Banker..Emotet.gen	5.13	5.08	
Trojan..Injuke.pef	4.48	3.52	
Trojan-Banker..Qbot.pef	4.17	3.80	
Trojan..Generic	3.93	3.52	3.70
Trojan..Zenpak.pef	3.89	4.20	
Trojan..Injuke.vho	3.54	4.12	
Trojan..Injuke.gen	2.64	1.88	
Virus..Nimnul.f	2.20	1.52	
Trojan-Banker..Emotet.pef	1.81	1.64	
Trojan..Zenpak.gen	1.72	1.52	
UDSDangerousObject.Multi.Generic	1.57	1.24	1.85
Trojan-PSW.MSIL.Agensla.gen	1.28	1.40	
Trojan-Banker..Emotet.vho	1.21	1.16	
Trojan..Cometer.gen	1.01	1.08	

Table 8.6: Percentage Distribution of Kaspersky Results for Overall, Correctly Classified, and Wrongly Classified Malware for model ViT(B/8) and dataset 401

Category	Overall %	Correctly Classified %	Wrongly Classified %
null	23.09	33.04	60.98
other	33.21	31.49	34.16
Trojan-Banker..Emotet.gen	5.13	5.05	
Trojan..Injuke.pef	4.48	4.10	
Trojan-Banker..Qbot.pef	4.17	3.78	
Trojan..Generic	3.93	3.54	2.44
Trojan..Zenpak.pef	3.89	4.17	
Trojan..Injuke.vho	3.54	4.10	
Trojan..Injuke.gen	2.64	1.87	
Virus..Nimnul.f	2.20	1.51	
Trojan-Banker..Emotet.pef	1.81	1.63	
Trojan..Zenpak.gen	1.72	1.51	
UDSDangerousObject.Multi.Generic	1.57	1.24	2.44
Trojan-PSW.MSIL.Agensla.gen	1.28	1.35	
Trojan-Banker..Emotet.vho	1.21	1.15	
Trojan..Cometer.gen	1.01	1.07	

Table 8.7: Percentage Distribution of Kaspersky Results for Overall, Correctly Classified, and Wrongly Classified Malware for model ResNet-18 and Dataset 401

Metric	**ViT-B/8**	**ResNet-18**
Test Loss	0.0634	0.0752
Test Accuracy	0.9840	0.9789
Recall	0.9840	0.9789
F1 Score	0.9919	0.9893
Correctly Classified Malware (count)	2515	2502
Correctly Classified Malware (%)	98.4%	97.89%
Wrongly Classified Malware (count)	41	54
Wrongly Classified Malware (%)	1.6%	2.11%

Table 8.8: Performance metrics and classification stats for ViT-B/8 and ResNet-18 on Malware Dataset 401

Τα δεδομένα στους Πίνακες 1 και 2 παρουσιάζουν την ποσοστιαία κατανομή των κατηγοριών κακόβουλου λογισμικού, καθώς και τα ποσοστά σωστής και λανθασμένης ταξινόμησης για δύο μοντέλα, το ViT-B/8 και το ResNet-18, στο dataset 401. Και τα δύο μοντέλα ακολουθούν παρόμοια κατανομή κατηγοριών, με τις κατηγορίες «null» και «other» να κυριαρχούν στο σύνολο των δεδομένων. Το ViT-B/8 εμφανίζει ελαφρώς υψηλότερα ποσοστά σωστής ταξινόμησης και χαμηλότερα ποσοστά λάθους σε σχεδόν όλες τις κατηγορίες σε σύγκριση με το ResNet-18. Αυτό επιβεβαιώνεται και από τα συνοπτικά μετρικά απόδοσης στον Πίνακα 3, όπου το ViT-B/8 παρουσιάζει μικρότερη απώλεια δοκιμής (0.0634 έναντι 0.0752) και ελαφρώς καλύτερη ακρίβεια (98.40% έναντι 97.89%), ανάκληση και F1 score. Συνολικά, τα αποτελέσματα αυτά υποδεικνύουν ότι το ViT-B/8 παρέχει πιο αξιόπιστη και ακριβή ανίχνευση malware στο συγκεκριμένο dataset.

Τα μοντέλα ViT όπως βλέπουμε ως εδώ υπερέχουν σε σύγκριση με πιο απλές δομές, αποτελέσματα που συμφωνούν και με την βιβλιογραφία που είδαμε παραπάνω. Παρόλα αυτά παρακάτω θα δούμε αναλυτικά την αντίδραση σε κακόβουλο λογισμικό επόμενης εποχής και σε adversarial δείγματα.

8.4 Adversarial

Method	CNN	ViT
434	0.9727	0.9718
Add zeros_00	0.9880	0.9877
Add Libraries	0.9751	0.9754
2 Methods	0.9519	0.7465

Table 8.9: Adversarial

Επεξήγηση του Πίνακα

Ο Πίνακας 8.9 παρουσιάζει τις επιδόσεις των δύο αρχιτεκτονικών CNN (Convolutional Neural Networks) και ViT (Vision Transformers) σε διαφορετικές δοκιμές.

- **434:** Η βασική ακρίβεια των μοντέλων χωρίς καμία επίθεση ή τροποποίηση.

- **Add zeros:** Δοκιμή όπου προστίθενται μηδενικά στα εισαγόμενα δεδομένα, με σκοπό να επηρεάσουν την ταξινόμηση. Η ακρίβεια παραμένει υψηλή, υποδηλώνοντας ανθεκτικότητα.

- **Add Libraries:** Τροποποίηση βιβλιοθηκών, που αλλάζει τη συμπεριφορά των μοντέλων. Η ακρίβεια μειώνεται ελαφρώς σε σχέση με τη βασική τιμή.

- **2 Methods:** Συνδυασμός δύο τεχνικών επίθεσης, ο οποίος επηρεάζει σημαντικά την απόδοση, ιδιαίτερα στο ViT, που εμφανίζει αρκετά μεγάλη πτώση της ακρίβειας.

Το ResNet-18 φαίνεται πιο ανθεκτικό στις adversarial επιθέσεις σε σχέση με το ViT-Base-8 , κάτι που υποστηρίζεται από τα heatmap στατιστικά που θα δούμε και παρακάτω. Το ResNet-18 εμφανίζει υψηλότερη μέση ενεργοποίηση και μεγαλύτερη τυπική απόκλιση, γεγονός που δείχνει ότι εστιάζει σε πιο ποικιλόμορφα και έντονα χαρακτηριστικά, καθιστώντας το λιγότερο ευάλωτο σε μικρές τοπικές αλλοιώσεις. Αντίθετα, το ViT , που βασίζεται σε global attention και χωρίζει την εικόνα σε patches, μπορεί να ξεγελαστεί πιο εύκολα από στοχευμένες επιθέσεις.

Από τα αποτελέσματα των heatmaps ουσιαστικά, παρατηρούμε ότι το ResNet-18 παρουσιάζει καλύτερη επίδοση σε σύγκριση με το ViT για το συγκεκριμένο σύνολο δεδομένων κακόβουλου λογισμικού.Μία πιθανή αιτία είναι η μη ισορροπημένη φύση του συνόλου δεδομένων.

8.5 Στατιστικά Heatmaps

Για το μοντέλο vit υλοποιήσαμε το μοντέλο μας ενισχυμένο με δυνατότητες Explainable AI (XAI) μέσω της τεχνικής Grad-CAM. Το μοντέλο διαχειρίζεται τις εικόνες όχι ως συνεχές πλέγμα εικονοστοιχείων, αλλά ως ακολουθίες και τις επεξεργάζεται μέσω μηχανισμών προσοχής. Για την εξήγηση των προβλέψεων, αξιοποιείται το Grad-CAM (Gradient-weighted Class Activation Mapping), μία τεχνική που παράγει heatmaps σημασίας, δηλαδή χρωματικές απεικονίσεις που αναδεικνύουν ποιες περιοχές της εισόδου συνέβαλαν περισσότερο στην τελική απόφαση του μοντέλου.

Η διαδικασία του Grad-CAM περιλαμβάνει την καταγραφή των ενεργοποιήσεων (forward pass) και των παραγώγων (backward pass) ενός εσωτερικού επιπέδου του μοντέλου. Συγκεκριμένα, στην περίπτωσή μας εφαρμόζει στο τελευταίο fully connected επίπεδο. Κατά τη forward propagation, αποθηκεύονται οι ενεργοποιήσεις του επιπέδου για την είσοδο. Στη συνέχεια, γίνεται backward propagation μόνο ως προς την προβλεπόμενη (ή καθορισμένη) κλάση, και συλλέγονται τα gradients αυτής της εξόδου σε σχέση με τις ενεργοποιήσεις. Οι παράγωγοι αυτοί σταθμίζονται με μέσο όρο ως προς τον χώρο δημιουργώντας ένα σύνολο από ¨βάρη σημασίας¨. Τα βάρη αυτά εφαρμόζονται πάνω στους αντίστοιχους activation maps, και από αυτόν τον συνδυασμό προκύπτει ένας πρώτος χάρτης σημασίας, ο οποίος κατόπιν μετασχηματίζεται σε έναν κανονικοποιημένο θερμοχάρτη. Ο τελικός θερμοχάρτης υπερτίθεται πάνω στην αρχική εικόνα, προσφέροντας μια οπτικά κατανοητή εξήγηση του ποια σημεία της εικόνας ¨προκάλεσαν' την πρόβλεψη του μοντέλου.

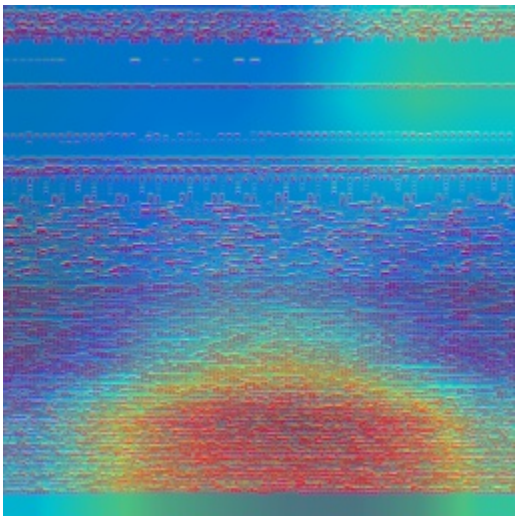
Για το μοντέλο ResNet-18 χρησιμοποιείται η τεχνική ερμήνευσης Grad-CAM (Gradient-weighted Class Activation Mapping), προκειμένου να εντοπιστούν οι περιοχές της εικόνας που επηρεάζουν περισσότερο την απόφαση του νευρωνικού δικτύου. Πιο συγκεκριμένα, εφαρμόζεται στο τελευταίο συνελικτικό επίπεδο, το οποίο διατηρεί την

χωρική πληροφορία που είναι απαραίτητη για την παραγωγή ερμηνεύσιμων θερμικών χαρτών. Κατά την προώθηση της εικόνας , καταγράφονται οι ενεργοποιήσεις αυτού του επιπέδου, ενώ στη συνέχεια ακολουθεί στοχευμένη διάδοση προς τα πίσω μόνο ως προς τη νευρωνική έξοδο της προβλεπόμενης κλάσης. Δηλαδή, υπολογίζονται οι παράγωγοι της συγκεκριμένης κατηγορίας , επιτρέποντας στο Grad-CAM να απομονώσει τα χαρακτηριστικά που συνέβαλαν σε αυτήν την απόφαση. Οι παράγωγοι αυτοί κατά μέσο όρο παράγουν ένα σύνολο βαρών, τα οποία συνδυάζονται γραμμικά με τις αντίστοιχες ενεργοποιήσεις και οδηγούν στη δημιουργία ενός ¨χάρτη προσοχής' όπως προαναφέραμε. Ο τελικός θερμικός χάρτης πάλι προβάλλεται πάνω στην αρχική εικόνα, προσφέροντας οπτική εξήγηση για το πού ¨κοιτάει' το δίκτυο όταν ταξινομεί την είσοδο.

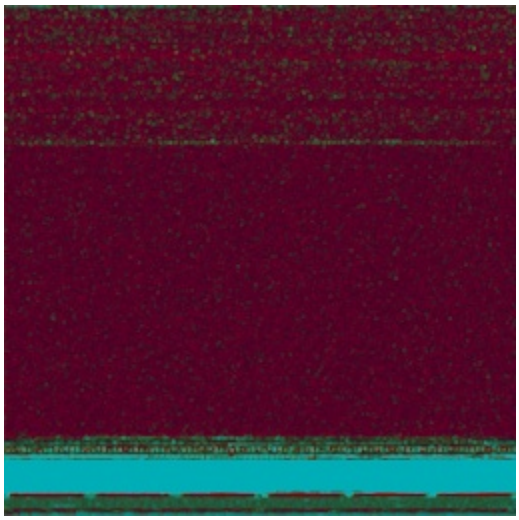
Ουσιαστικά το Grad-CAM εφαρμόζεται διαφορετικά σε CNN και Vision Transformers λόγω της διαφορετικής αρχιτεκτονικής τους. Στα πρώτα, ο θερμικός χάρτης παράγεται από τα τελευταία συνελικτικά επίπεδα, με βάση τις παραγώγους της προβλεπόμενης κλάσης ενώ στα ViTs, δεν υπάρχουν χωρικά feature maps, οπότε γίνετε προσαρμογή ώστε να χρησιμοποιεί τις ενεργοποιήσεις του class token και των attention heads. Παρά τις διαφορές, και στις δύο περιπτώσεις γίνεται backward propagation μόνο ως προς την έξοδο της στοχευμένης κλάσης, ώστε να εντοπιστούν τα πιο σημαντικά σημεία της εικόνας για την τελική απόφαση.

- ResNet-18:
  - Ελάχιστη τιμή: 0.00
  - Μέγιστη τιμή: 255.00
  - Μέση τιμή: 125.97
  - Τυπική απόκλιση: 44.78
- ViT:
  - Ελάχιστη τιμή: 0.00
  - Μέγιστη τιμή: 209.00
  - Μέση τιμή: 107.22
  - Τυπική απόκλιση: 30.52

Παράδειγμα HeatMap:



Σχήμα 8.1: Vit GradCam



Σχήμα 8.2: ResNet-18 GradCam

8.6 Ερμηνεία Αποτελεσμάτων

Το ResNet-18 εμφανίζει:

- Υψηλότερη μέγιστη ενεργοποίηση (255 έναντι 209)
- Μεγαλύτερη μέση τιμή (125.97 έναντι 107.22)
- Σημαντικά υψηλότερη τυπική απόκλιση (44.78 ς 30.52)

Αυτό υποδηλώνει ότι το ResNet-18:

- Εστιάζει πιο έντονα σε συγκεκριμένες περιοχές
- Εμφανίζει μεγαλύτερη διακριτική ικανότητα
- Αγνοεί περισσότερο το θόρυβο και τα μη σχετικά χαρακτηριστικά

8.7 Πιθανές Αιτίες

- Το ResNet-18 ως CNN:
  - Εξειδικεύεται στην ανίχνευση τοπικών χαρακτηριστικών
  - Είναι πιο ανθεκτικό σε μη ισορροπημένα δεδομένα
  - Διαθέτει μεγαλύτερη μετάφραση
- Το ViT:
  - Απαιτεί περισσότερα δεδομένα για την εκμάθηση καθολικών εξαρτήσεων
  - Μπορεί να υπερεκπαιδεύεται στην πλειοψηφική κλάση
  - Δυσκολεύεται σε μικρά/τοπικά χαρακτηριστικά

8.8 Επανεκπαίδευση

Όσον αφορά τη διαδικασία της επανεκπαίδευσης, είναι εμφανές ότι η βέλτιστη στρατηγική είναι η εξής:

- Εκπαίδευση μόνο στο target dataset (401), εφόσον υπάρχουν επαρκή δεδομένα.
- Αν απαιτείται προ-εκπαίδευση, το full fine-tuning αποτελεί τη μόνη αποδοτική επιλογή.

Σημαντικά:

- Πρέπει να αποφεύγουμε το fine-tuning μόνο του (head), καθώς οδηγεί σε σημαντικά υποβέλτιστη απόδοση.
- Η επανεκπαίδευση head και backbone (H,B) χρειάζετε αρκετά μεγαλύτερα χρονικά διαστήματα εκπαίδευσης και δεν επιτυγχάνει επιθυμητά αποτελέσματα

Εξαγωγή Γνώσης:
Η προ-εκπαίδευση σε ετερογενές dataset (375) δεν φαίνεται να μεταφέρει αποτελεσματικά γνώση στο target dataset (401), πιθανώς λόγω:

- Μεγάλων διαφορών στα δεδομένα.
- Ανεπαρκούς σχετικότητας μεταξύ των δύο datasets.
- Υπερβολικής ευαισθησίας των μοντέλων σε αλλαγές στα δεδομένα.

Σίγουρα υπάρχει περιθώριο βελτίωσης της επανεκπαίδευσης, αν πειραματιζόμασταν περισσότερο με παραμέτρους όπως dropout, learning rate, και παγώματα/ξεπαγώματα επιπέδων. Ωστόσο, αυτή η διαδικασία δεν είναι αποδοτική ούτε γρήγορη, που ήταν και ο στόχος της παρούσας εργασίας. Επιπλέον, ενδεχομένως αν η εξαγωγή χαρακτηριστικών δεν γινόταν με ξεχωριστό autoencoder για κάθε dataset,αλλά και τα είδη του κακόβουλου λογισμικού ήταν πιο ομοιογενή θα μπορούσαμε να έχουμε καλύτερα αποτελέσματα.

Συγκριτική Μελέτη Ανίχνευσης Malware

Η ανάλυση των αποτελεσμάτων δείχνει ότι το μοντέλο ResNet παρουσιάζει ελαφρώς μεγαλύτερο ποσοστό λανθασμένων ταξινομήσεων σε σχέση με το ViT Base 8. Συγκεκριμένα, το ResNet δυσκολεύεται περισσότερο να ανιχνεύσει malware κατηγοριών όπως Trojan..Generic, Trojan-Banker..Gozi.lop και Packed.Multi.MultiPacked.gen, οι οποίες εμφανίζονται πιο συχνά στα λάθη του. Αντίθετα, το ViT Base 8 δείχνει μικρότερα ποσοστά λανθασμένης ταξινόμησης και αντιμετωπίζει καλύτερα την ανίχνευση σε κατηγορίες όπως Trojan.MSIL.Dnoper.gen, Trojan-Spy.MSIL.Quasar.gen και Backdoor..Agent.myuaoc.

Και τα δύο μοντέλα δυσκολεύονται σημαντικά στην ακριβή ταξινόμηση των δειγμάτων με κατηγορία null (μη κατηγοριοποιημένα), με το ποσοστό αυτών να είναι πάνω από το 30% στους σωστά ταξινομημένους και πάνω από 60% στα λανθασμένα. Αυτό υποδεικνύει ότι τα null δείγματα μπορεί είναι δύσκολα στην ανίχνευση όπως ήταν και αναμενόμενο μιας και είναι τα δείγματα που δυσκολεύονταν ήδη αρκετοί ανιχνευτές

να εντοπίσουν. Επιπλέον, η κατηγορία other εμφανίζει επίσης υψηλή παρουσία στα σωστά ταξινομημένα δείγματα, γεγονός που υποδηλώνει ότι τα μοντέλα αναγνωρίζουν καλά ευρύ φάσμα διαφορετικών τύπων malware, αλλά η ασαφής φύση της κατηγορίας αυτής απαιτεί περαιτέρω διερεύνηση.

Συνολικά, το ViT(B/8) υπερέχει ελαφρώς στην ακρίβεια ταξινόμησης, ενώ το ResNet φαίνεται να παρουσιάζει μεγαλύτερη ευαισθησία σε συγκεκριμένες κατηγορίες malware , ανάλογα με την πολυπλοκότητα των χαρακτηριστικών που αναγνωρίζει.

Κεφάλαιο 9

Επίλογος

Η ραγδαία εξάπλωση του κακόβουλου λογισμικού, σε συνδυασμό με τη δυναμική και διαρκώς μεταβαλλόμενη φύση των απειλών στον κυβερνοχώρο, καθιστά την ανίχνευση και την πρόληψή του ένα από τα πιο κρίσιμα και απαιτητικά προβλήματα της σύγχρονης ψηφιακής ασφάλειας. Η παρούσα εργασία επιχείρησε να συμβάλει ουσιαστικά στο πεδίο αυτό, αξιοποιώντας προηγμένες μεθόδους deep learning για την αυτοματοποιημένη αναγνώριση κακόβουλων εκτελέσιμων αρχείων.

Κεντρικός στόχος υπήρξε η συγκριτική αξιολόγηση δύο σύγχρονων και διακεκριμένων προσεγγίσεων: των Convolutional Neural Networks (CNN) και των Vision Transformers (ViT). Μέσα από τη συστηματική ανάλυση, τον σχεδιασμό και την υλοποίηση ενός πλήρους πειραματικού πλαισίου, αποτυπώθηκαν με ακρίβεια οι επιδόσεις των παραπάνω μοντέλων ως προς την ακρίβεια ταξινόμησης, τη γενίκευση σε νέα δεδομένα, την ταχύτητα, και την ανθεκτικότητά τους απέναντι σε τεχνικές παραπλάνησης.

Τα αποτελέσματα ήταν ενθαρρυντικά. Τα μοντέλα ResNet-18 και ViT-B/8 επέδειξαν εξαιρετικές επιδόσεις στις βασικές μετρικές (accuracy, recall, F1-score), επιβεβαιώνοντας τη δυναμική τους για πρακτική αξιοποίηση. Επιπλέον, η ενσωμάτωση τεχνικών autoencoder για μείωση της διαστατικότητας βελτίωσε αισθητά την ταξινόμηση, ιδιαίτερα σε περιπτώσεις σύνθετων ή θορυβωδών δεδομένων.

Παράλληλα, αναδείχθηκε η ανάγκη για την ανάπτυξη ανθεκτικών μοντέλων, ικανών να ανταποκρίνονται σε adversarial attacks. Η εργασία ανέδειξε πως ακόμη και ισχυρά δίκτυα παραμένουν ευάλωτα σε τεχνητές παραποιήσεις εισόδου, ένα ζήτημα που απαιτεί μελλοντική αντιμετώπιση με κατάλληλες στρατηγικές άμυνας και ενίσχυσης της ανθεκτικότητας.

9.1 Στάδια Αξιολόγησης

Πειράματα

Σύμφωνα και με τις ενδείξεις της σχετικής βιβλιογραφίας, τα πειράματα επιβεβαίωσαν ότι η αύξηση του όγκου των δεδομένων οδήγησε σε βελτιωμένα αποτελέσματα. Ιδιαίτερο ενδιαφέρον παρουσιάζει το γεγονός ότι η ανομοιογένεια του δείγματος επέτρεψε στα Συνελικτικά Νευρωνικά Δίκτυα (CNN) να ανταγωνίζονται —και σε ορισμένες περιπτώσεις να υπερέχουν— των Vision Transformers.

9.1.1 Έλεγχος σε δεδομένα άλλης γενιάς

Τα δύο μοντέλα πέτυχαν υψηλά ποσοστά ανίχνευσης σε κακόβουλο λογισμικό που δεν είχαν ξαναδεί, το οποίο βασιζόταν σε τεχνολογίες αισθητά διαφορετικές από εκείνες του αρχικού συνόλου δεδομένων. Αξιοσημείωτο είναι ότι τα Συνελικτικά Δίκτυα εμφάνισαν ελαφρώς καλύτερες επιδόσεις από τα Vision Transformers. Όπως έδειξαν και τα heatmaps, η ανάγκη των ViT για καθολική πληροφόρηση από το σύνολο των εισόδων λειτούργησε περιοριστικά σε αυτό το σενάριο, οδηγώντας σε ελαφρώς χαμηλότερη απόδοση.

9.1.2 Ensemble Model

Η χρήση ενός ensemble μοντέλου βελτίωσε την ακρίβεια των προβλέψεων κατά περίπου +1%, αποδεικνύοντας ότι οι δύο διαφορετικές αρχιτεκτονικές μπορούν να λειτουργήσουν συμπληρωματικά. Το αποτέλεσμα αυτό υπογραμμίζει πως δεν πρόκειται για ανταγωνιστικές προσεγγίσεις, αλλά για συνεργατικές τεχνολογίες με ενισχυτική δυναμική.

9.1.3 Adversarial Attacks

Η αξιολόγηση με παραλλαγμένα δείγματα (adversarial examples) παρήγαγε ορισμένα από τα πιο ενδιαφέροντα ευρήματα. Μικρές τροποποιήσεις στα δεδομένα δεν επηρέασαν ουσιαστικά την απόδοση των μοντέλων. Ωστόσο, όταν εισήχθησαν μεγάλες ποσότητες καλοήθους λογισμικού με αλλοιωμένα χαρακτηριστικά, το ViT-B/8 παρουσίασε αισθητή πτώση στην ακρίβεια (περίπου -20%).

Το εύρημα αυτό ενδέχεται να σχετίζεται με το γεγονός ότι, αν και το εν λόγω μοντέλο διαθέτει έναν πιο καθολικό και σύνθετο μηχανισμό επεξεργασίας, όταν η διαφοροποίηση των χαρακτηριστικών είναι έντονη, χάνει την ικανότητά του να γενικεύει, οδηγούμενο ευκολότερα σε λανθασμένες προβλέψεις. Αντίθετα, τα CNN, βασιζόμενα κυρίως στην αναγνώριση τοπικών και πιο απλών μοτίβων, έδειξαν μεγαλύτερη ανθεκτικότητα και σταθερότητα στην απόδοσή τους.

9.1.4 Επανεκπαίδευση

Η επανεκπαίδευση προϋπάρχοντων νευρωνικών δικτύων δεν αποδείχθηκε απλή διαδικασία. Κατά την εκπαίδευση μόνο ορισμένων μερών του δικτύου (*παρτιαλ φινε-τυνινγ*), τα αποτελέσματα ήταν περιορισμένα και η διαδικασία χρονοβόρα. Αντιθέτως, η καθολική επανεκπαίδευση (*full fine-tuning*) απαιτούσε ιδιαίτερα προσεκτική παραμετροποίηση ώστε να επιτευχθεί ικανοποιητική απόδοση.

Το βασικό όφελος που προέκυψε ήταν μια μικρή μείωση στον χρόνο εκτέλεσης, χωρίς ωστόσο σημαντική αύξηση στην ακρίβεια ταξινόμησης.

9.2 Συμπεράσματα

Βάσει των αποτελεσμάτων αλλά και της βιβλιογραφικής ανάλυσης, προκύπτει ότι η επιλογή κατάλληλου μοντέλου για την ανίχνευση κακόβουλου λογισμικού δεν είναι ενιαία ή απόλυτη. Εξαρτάται από τον σκοπό της εφαρμογής, τους διαθέσιμους πόρους και τη φύση των δεδομένων. Οι πιο σύνθετες αρχιτεκτονικές παρουσιάζουν αυξημένες

απαιτήσεις και ενδέχεται να δυσκολεύονται σε ανομοιογενή ή αδόμητα δεδομένα. Όταν όμως τα δεδομένα είναι ποιοτικά και καλά οργανωμένα, τα αποτελέσματα είναι ιδιαίτερα ικανοποιητικά.

Το προτεινόμενο σύστημα, λόγω των χαμηλών υπολογιστικών απαιτήσεών του, μπορεί να λειτουργήσει ως μηχανισμός προ-φιλτραρίσματος σε πρώιμο στάδιο της διαδικασίας ανίχνευσης.

Συνοψίζοντας, η παρούσα εργασία χτίζει πάνω σε εξαιρετικές προσεγγίσεις που έχουν αναφερθεί στην διεθνή βιβλιογραφία με στόχο την επίτευξη μιας ισορροπίας μεταξύ του χρόνου εκπαίδευσης αλγορίθμων μηχανικής μάθησης, δηλαδή του μη ωφέλιμου χρόνου, και της ακρίβειας στην ανίχνευση κακόβουλου λογισμικού. Η μελλοντική έρευνα οφείλει να εστιάσει στην ενίσχυση της γενίκευσης, της ερμηνευσιμότητας και της ανθεκτικότητας των συστημάτων, ώστε να καταστούν πιο αποτελεσματικά και αξιόπιστα σε συνθήκες πραγματικού κόσμου, απέναντι σε εξελισσόμενες απειλές.

9.3 Προεκτάσεις

Οι μεθοδολογίες για την ανίχνευση απειλών ποικίλλουν σημαντικά τόσο σε επίπεδο αλγορίθμων όσο και παραμετροποίησης. Κάθε στάδιο της διαδικασίας —από την επιλογή των δεδομένων και την εξαγωγή χαρακτηριστικών (feature extraction), μέχρι τον διαχωρισμό τους σε σύνολα train, test, validation και την επιλογή του κατάλληλου μοντέλου— εξαρτάται από τις απαιτήσεις του συστήματος και τους διαθέσιμους πόρους.

Προτεινόμενες κατευθύνσεις για μελλοντική έρευνα:

- Χρήση τεχνικών μείωσης διαστάσεων, όπως οι Variational Autoencoders (VAE), για πιο συμπαγή και εκφραστική αναπαράσταση των δεδομένων.
- Ενσωμάτωση adversarial examples στο σύνολο εκπαίδευσης, με στόχο τη βελτίωση της ανθεκτικότητας του μοντέλου.
- Αξιοποίηση προεκπαιδευμένων μοντέλων (pretrained models) και εφαρμογή τεχνικών fine-tuning, ώστε να μειωθεί ο χρόνος εκπαίδευσης και να βελτιωθεί η απόδοση.
- Επέκταση του μοντέλου πέρα από τεχνικές υπολογιστικής όρασης, με σκοπό τη δημιουργία ενός πιο πολυδιάστατου ensemble συστήματος ανίχνευσης.
- Συλλογή και επαίδευση σε ποικιλόμορφο δείγμα καλόβουλων δειγμάτων

Οι παραπάνω κατευθύνσεις αποτελούν σημαντικά βήματα προς την ανάπτυξη πιο ανθεκτικών και γενικεύσιμων συστημάτων ανίχνευσης. Η ενίσχυση της ακρίβειας, η καλύτερη προσαρμογή σε μη ορατά δείγματα και η βελτίωση της επεξηγησιμότητας των αποτελεσμάτων είναι εφικτές μέσα από στοχευμένες τεχνικές, τόσο σε επίπεδο προεπεξεργασίας δεδομένων όσο και σε επίπεδο αρχιτεκτονικής μοντέλου. Μέσω αυτών των βελτιώσεων, μπορεί να επιτευχθεί ένα πιο ευέλικτο και αξιόπιστο πλαίσιο, ικανό να ανταποκρίνεται αποτελεσματικά σε πραγματικές και μεταβαλλόμενες συνθήκες απειλών.

Bibliography

[1] D. Gibert, C. Mateu, and J. Planes, "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," *Journal of Network and Computer Applications*, 03 2020.

[2] Y. Chen, X. Gu, Z. Liu, and J. Liang, "A fast inference vision transformer for automatic pavement image classification and its visual interpretation method," *Remote Sensing*, vol. 14, p. 1877, 04 2022.

[3] "Kaspersky security bulletin 2023. statistics."

[4] Microsoft Corporation, "What is malware? definition and types." `https://www.microsoft.com/el-gr/security/business/security-101/what-is-malware`, 2025.

[5] M. Naseer, J. F. Rusdi, N. M. Shanono, S. Salam, Z. B. Muslim, N. A. Abu, and I. Abadi, "Malware detection: Issues and challenges," vol. 1807, no. 1, p. 012011. Publisher: IOP Publishing.

[6] J. R. S. Alrzini and D. Pennington, "A review of polymorphic malware detection techniques," vol. 11, no. 12, pp. 1238–1247. Num Pages: 305567 Number: 12.

[7] M. N. Alenezi, H. K. Alabdulrazzaq, A. A. Alshaher, and M. M. Alkharang, "Evolution of malware threats and techniques: a review," vol. 12, no. 3.

[8] J. Ferdous, R. Islam, A. Mahboubi, and M. Z. Islam, "A review of state-of-the-art malware attack trends and defense mechanisms," vol. 11, pp. 121118–121141.

[9] K. Radhakrishnan, R. R. Menon, and H. V. Nath, "A survey of zero-day malware attacks and its detection methodology," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, pp. 533–539, 2019.

[10] S. K. Pandey and B. Mehtre, "A lifecycle based approach for malware analysis," in *2014 Fourth International Conference on Communication Systems and Network Technologies*, pp. 767–771, 2014.

[11] N. Z. Gorment, A. Selamat, K. Lim, and O. Krejcar, "Machine learning algorithm for malware detection: Taxonomy, current challenges and future directions," *IEEE Access*, vol. PP, pp. 1–1, 01 2023.

[12] Department of Computer Science, Virtual University of Pakistan and R. Tahir, "A study on malware and malware detection techniques," vol. 8, no. 2, pp. 20–30.

[13] H. Oz, A. Aris, A. Levi, and A. S. Uluagac, "A survey on ransomware: Evolution, taxonomy, and defense solutions," *ACM Comput. Surv.*, vol. 54, Sept. 2022.

[14] StatCounter GlobalStats, "Desktop operating system market share worldwide." `https://gs.statcounter.com/os-market-share/desktop/worldwide`, 2025. Accessed: Aug., 2025.

[15] Microsoft Developer Documentation, "Portable executable (pe) format." `https://learn.microsoft.com/en-us/windows/win32/debug/pe-format`.

[16] J.-E. J. Tevis and J. A. Hamilton, "Static analysis of anomalies and security vulnerabilities in executable files," in *Proceedings of the 44th annual ACM Southeast Conference*, ACMSE '06, pp. 560–565, Association for Computing Machinery.

[17] A. Bensaoud, J. Kalita, and M. Bensaoud, "A survey of malware detection using deep learning," *Machine Learning with Applications*, vol. 16, p. 100546, 2024.

[18] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A. S. Al-rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware detection issues, challenges, and future directions: A survey," vol. 12, no. 17, p. 8482. Number: 17 Publisher: Multidisciplinary Digital Publishing Institute.

[19] A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," vol. 8, pp. 6249–6271. Conference Name: IEEE Access.

[20] P. Szor, *The Art of Computer Virus Research and Defense*. Pearson Education, 2005.

[21] I. Firdausi, C. lim, A. Erwin, and A. S. Nugroho, "Analysis of machine learning techniques used in behavior-based malware detection," in *2010 Second International Conference on Advances in Computing, Control, and Telecommunication Technologies*, pp. 201–203.

[22] S. Jha, D. Prashar, H. V. Long, and D. Taniar, "Recurrent neural network for detecting malware," vol. 99, p. 102037.

[23] K. Mduma and Khamis, "A survey of machine learning approaches and techniques for student dropout prediction," *Data Science Journal*, vol. 18, p. 14, 2019.

[24] A. Damodaran, F. D. Troia, C. A. Visaggio, T. H. Austin, and M. Stamp, "A comparison of static, dynamic, and hybrid analysis for malware detection," vol. 13, no. 1, pp. 1–12.

[25] A.-R. Belea, "Methods for detecting malware using static, dynamic and hybrid analysis," in *Proceedings of the International Conference on Cybersecurity and Cybercrime - 2023*, pp. 258–265, Asociatia Romana pentru Asigurarea Securitatii Informatiei.

[26] O. Or-Meir, N. Nissim, Y. Elovici, and L. Rokach, "Dynamic malware analysis in the modern era—a state of the art survey," vol. 52, no. 5, pp. 88:1–88:48.

[27] T.-Y. Wang and C.-H. Wu, "Detection of packed executables using support vector machines," in *2011 International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 717–722. ISSN: 2160-1348.

[28] M. Schultz, E. Eskin, F. Zadok, and S. Stolfo, "Data mining methods for detection of new malicious executables," in *Proceedings 2001 IEEE Symposium on Security and Privacy. S&P 2001*, pp. 38–49. ISSN: 1081-6011.

[29] J. Z. Kolter and M. A. Maloof, "Learning to detect malicious executables in the wild," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, (New York, NY, USA), p. 470–478, Association for Computing Machinery, 2004.

[30] K. Rieck, T. Holz, C. Willems, P. Düssel, and P. Laskov, "Learning and classification of malware behavior," in *Detection of Intrusions and Malware, and Vulnerability Assessment* (D. Zamboni, ed.), pp. 108–125, Springer.

[31] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," in *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, VizSec '11, pp. 1–7, Association for Computing Machinery.

[32] R. Tian, R. Islam, L. Batten, and S. Versteeg, "Differentiating malware from cleanware using behavioural analysis," in *2010 5th International Conference on Malicious and Unwanted Software*, pp. 23–30.

[33] R. Islam, R. Tian, L. M. Batten, and S. Versteeg, "Classification of malware based on integrated static and dynamic features," vol. 36, no. 2, pp. 646–656.

[34] B. Anderson, D. Quist, J. Neil, C. Storlie, and T. Lane, "Graph-based malware detection using dynamic analysis," vol. 7, no. 4, pp. 247–258.

[35] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, "Large-scale malware classification using random projections and neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3422–3426. ISSN: 2379-190X.

[36] E. Gandotra, D. Bansal, and S. Sofat, "Malware analysis and classification: A survey," vol. 2014. Publisher: Scientific Research Publishing.

[37] M. S. Akhtar and T. Feng, "Malware analysis and detection using machine learning algorithms," vol. 14, no. 11, p. 2304. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[38] M. S. Akhtar and T. Feng, "Detection of malware by deep learning as CNN-LSTM machine learning techniques in real time," vol. 14, no. 11, p. 2308. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

[39] S. K. J. Rizvi, W. Aslam, M. Shahzad, S. Saleem, and M. M. Fraz, "PROUD-MAL: static analysis-based progressive framework for deep unsupervised malware classification of windows portable executable," vol. 8, no. 1, pp. 673–685.

[40] J. M, A. Shaik, G. Pendharkar, S. Kumar, M. K. B, and S. Balaji, "Comparative analysis of imbalanced malware byteplot image classification using transfer learning," vol. 1097, pp. 313–324.

[41] S. Yue and T. Wang, "Imbalanced malware images classification: a CNN based approach."

[42] S. Seneviratne, R. Shariffdeen, S. Rasnayaka, and N. Kasthuriarachchi, "Self-supervised vision transformers for malware detection," vol. 10, pp. 103121–103135. Conference Name: IEEE Access.

[43] S. Venkatraman, M. Alazab, and R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection," vol. 47, pp. 377–389.

[44] S. J. Nair and S. R. Syam, "Comparing transformers and CNN approaches for malware detection: A comprehensive analysis," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–6. ISSN: 2473-7674.

[45] F. Rustam, I. Ashraf, A. D. Jurcut, A. K. Bashir, and Y. B. Zikria, "Malware detection using image representation of malware data and transfer learning," vol. 172, pp. 32–50.

[46] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE malware detection using ensemble learning," vol. 8, no. 1, p. 10. Number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[47] D. Lambrakis, "Application of machine learning methods for malware detection," 2022. Available at: dspace.lib.ntua.gr.

[48] I. Ben abdel ouahab, L. Elaachak, and M. Bouhorma, "Enhancing malware classification with vision transformers: A comparative study with traditional CNN models," in *Proceedings of the 6th International Conference on Networking, Intelligent Systems & Security*, NISS '23, pp. 1–5, Association for Computing Machinery.

[49] D. Spinellis, "Reliable identification of bounded-length viruses is np-complete," *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 280–284, 2003.

[50] A. Moser, C. Kruegel, and E. Kirda, "Limits of static analysis for malware detection," in *Twenty-Third Annual Computer Security Applications Conference (ACSAC 2007)*, pp. 421–430. ISSN: 1063-9527.

[51] S. Lucas, "The origins of the halting problem," *Journal of Logical and Algebraic Methods in Programming*, vol. 121, p. 100687, 06 2021.

[52] VirusShare Project, "Virusshare malware repository." `https://virusshare.com/`.

[53] VirusTotal, "Virustotal: Online malware analysis service." `https://www.virustotal.com/`.

[54] J. Zhang, Z. Qin, H. Yin, L. Ou, S. Xiao, and Y. Hu, "Malware variant detection using opcode image recognition with small training sets," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–9.

[55] C. Duchon, "Lanczos filtering in one and two dimensions," 08 1979.

[56] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," vol. 86, no. 11, pp. 2278–2324.

[57] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions," vol. 11, no. 3, p. 52. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[58] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: a large-scale hierarchical image database," 06 2009.

[59] A. O. Salau and S. Jain, "Feature extraction: A survey of the types, techniques, applications," 2019.

[60] "What is fine-tuning? | IBM."

[61] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc.

[62] M. Alshomrani, A. Albeshri, B. Alturki, F. S. Alallah, and A. A. Alsulami, "Survey of transformer-based malicious software detection systems," vol. 13, no. 23, p. 4677. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute.

[63] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[64] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale."

[65] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2023.

[66] "Different types of autoencoders in machine learning."

[67] X. Ling, L. Wu, J. Zhang, Z. Qu, W. Deng, X. Chen, Y. Qian, C. Wu, S. Ji, T. Luo, J. Wu, and Y. Wu, "Adversarial attacks against windows PE malware detection: A survey of the state-of-the-art," vol. 128, p. 103134.

[68] M. Siddiqui, "Data mining methods for malware detection," 01 2008.

[69] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks."

[70] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra, "Grad-CAM: Why did you say that?."

[71] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: an imperative style, high-performance deep learning library," 2019.

[72] "pandas - python data analysis library -https://pandas.pydata.org/."

[73] C. da Costa-Luis, S. K. Larroque, K. Altendorf, H. Mary, richardsheridan, M. Korobov, N. Yorav-Raphael, I. Ivanov, M. Bargull, N. Rodrigues, Shawn, M. Dektyarev, M. Górny, mjstevens777, M. D. Pagel, M. Zugnoni, JC, CrazyPython, C. Newey, A. Lee, pgajdos, Todd, S. Malmgren, redbug312, O. Desh, N. Nechaev, M. Boyle, M. Nordlund, MapleCCC, and J. McCracken, "tqdm: A fast, extensible progress bar for python and CLI."

[74] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," Sept. 2020.

[75] Pillow Developers, "Pillow: Python imaging library." `https://pypi.org/project/pillow/`.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," Nov. 2011.

[77] M. Kozák, M. Jureček, and R. Lórencz, "Generation of adversarial malware and benign examples using reinforcement learning," in *Artificial Intelligence for Cybersecurity* (M. Stamp, C. Aaron Visaggio, F. Mercaldo, and F. Di Troia, eds.), Advances in Information Security, pp. 3–25, Springer International Publishing.

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," pp. 770–778.

[79] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in Neural Information Processing Systems*, vol. 27, Curran Associates, Inc.

Ακρωνύμια:

Res-Net Residual Network
ANN Artificial Neural Network.
API Application Programming Interface.
ARFF Attribute-Relation File Format
Auto Autoencoder
AutoCNN Autoencoder created images with CNN
AutoViT Autoencoder created images with Vision Transformer
BDT Boosted Decision Trees.
BNB Bernoulli Naive Bayes.
CFG Call Function Graphs.
CNN Convolutional Neural Network.
CPU Central Processing Unit.
DBN Deep Belief Network.
DBSCN Density-Based Spatial Clustering of Applications with Noise.
DBSCAN Density-Based Spatial Clustering of Applications with Noise
DLL Dynamic-link Library.
DOS Disk Operating System.
DT Decision Tree.
FFN Feed Forward Network.
GBDT Gradient Boosting Decision Trees.
GPU Graphical Proccesing Unit
HMM Hidden Markov Model.
IL Instance-based Learner.
K-NN K-Nearest Neighbors.
LR Logistic Regression.
LSTM Long Short-Term Memory.
NB Naive Bayes.
Opcode Operation Code.
PCA Principal Component Analysis.
PEiD Packed Executable Identifier
PE Portable Executable.
RBM Restricted Boltzmann Machine.
RF Random Forest.
ViT Vision Transformer
SGD Stochastic Gradient Descent.
SVDD Support Vector Domain Description.
SVM Support Vector Machine.
TF Term Frequency.
TF-IDF Term Frequency - Inverse Document Frequency.
TPR True Positive Rate.
VP Voted Perceptron.
XGB Extreme Gradient Boosting.
XAI Explainable Ai
URL Uniform Resource Locator
Σ.Η.Μ.Μ.Υ Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

κ.λπ. και λοιπά.
κ.ο.κ. και ούτω καθεξής.