

ΣΧΟΛΗ ΜΗΧΑΝΙΚΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ
«AI in Cybersecurity»



Του φοιτητή
Εμμανουήλ Καραμανώλη
Αρ. Μητρώου: 2020060

Επιβλέπων
Χρήστος Ηλιούδης
Καθηγητής

Ημερομηνία 30/07/2026

Τίτλος Δ.Ε. AI in Cybersecurity
Κωδικός Δ.Ε. 25164
Ονοματεπώνυμο φοιτητή/τών Εμμανουήλ Καραμανώλης
Ονοματεπώνυμο εισηγητή Χρήστος Ηλιούδης
Ημερομηνία ανάληψης Δ.Ε. 12/03/2025
Ημερομηνία περάτωσης Δ.Ε. 30/07/2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Εμμανουήλ Καραμανώλη που την εκπόνησε/αν. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Πρόλογος

Η κυβερνοασφάλεια ήταν ένα θέμα το οποίο από μικρός μου φαινόταν αρκετά ενδιαφέρον. Όταν παρακολούθησα το μάθημα Ασφάλεια Πληροφοριακών Συστημάτων κατάλαβα ότι με αυτό το αντικείμενο θα ασχοληθώ επαγγελματικά. Η αύξηση των κυβερνοεπιθέσεων και η εξάρτηση του ανθρώπου από το διαδίκτυο με οδήγησαν να ασχοληθώ με το συγκεκριμένο θέμα. Παράλληλα, η ανάπτυξη της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης παρουσιάζει την ευκαιρία για την αξιοποίηση τους στον τομέα της κυβερνοασφάλειας.

Η παρούσα διπλωματική εργασία αποτέλεσε για εμένα μια ευκαιρία να συνδυάσω την Τεχνητή Νοημοσύνη με την κυβερνοασφάλεια. Κατά τη συγγραφή της είχα την ευκαιρία να μελετήσω τεχνικές ανίχνευσης επιθέσεων, να επεξεργαστώ και να καθαρίσω μια μεγάλη βάση δεδομένων, να εξοικειωθώ με διάφορα μοντέλα Μηχανικής Μάθησης και να εφαρμόσω τις γνώσεις αυτές σε ένα πρακτικό πρόβλημα.

Μέσα από αυτή τη διατριβή απέκτησα πολύτιμες γνώσεις και εμπειρίες, τόσο σε ακαδημαϊκό όσο και σε προσωπικό επίπεδο. Η ενασχόληση με το συγκεκριμένο θέμα ενίσχυσε ακόμη περισσότερο το ενδιαφέρον μου για τον χώρο της κυβερνοασφάλειας και επιβεβαίωσε την επιθυμία μου να συνεχίσω να εξελίσσομαι επαγγελματικά και ερευνητικά σε αυτόν τον τομέα.

Περίληψη

Οι όλο και πιο συχνές επιθέσεις στον τομέα της κυβερνοασφάλειας έχουν δημιουργήσει την ανάγκη για πιο εξελιγμένους μηχανισμούς ανίχνευσης εισβολών, ικανούς να εντοπίζουν επιθέσεις zero-day. Τα τυπικά συστήματα ανίχνευσης εισβολών (IDS) βασίζονται σε υπογραφές και, ως εκ τούτου, δεν είναι σε θέση να ανιχνεύσουν επιθέσεις zero-day. Η χρήση της μηχανικής μάθησης (ML) έχει αναδειχθεί ως μια ενδιαφέρουσα μέθοδο για την ενίσχυση της ευελιξίας και της αποτελεσματικότητας των συστημάτων κυβερνοασφάλειας.

Η παρούσα διατριβή επικεντρώνεται σε μεθόδους μηχανικής μάθησης για την ανίχνευση επιθέσεων «zero-day» χρησιμοποιώντας δεδομένα δικτυακής κίνησης. Αναλύθηκαν τόσο οι εποπτευόμενες όσο και οι μη εποπτευόμενες τεχνικές, προκειμένου να αξιολογηθεί η απόδοσή τους στην ανίχνευση κακόβουλων δραστηριοτήτων που δεν περιλαμβάνονταν στα δεδομένα εκπαίδευσης. Για τη μελέτη αυτή επιλέχθηκαν τα μοντέλα Logistic Regression, Support Vector Machines (SVM), Random Forest, K-Means Clustering, Isolation Forest, και One-Class SVM.

Η προεπεξεργασία των δεδομένων, η επιλογή των χαρακτηριστικών, η εκπαίδευση των μοντέλων και η αξιολόγηση της απόδοσης αποτέλεσαν τα στάδια της πειραματικής διαδικασίας. Η αξιολόγηση της απόδοσης των μοντέλων πραγματοποιήθηκε με βάση ορισμένους από τους πιο σχετικούς δείκτες για την ανίχνευση εισβολών, όπως το Recall, το F1-Score, το False Positive Rate (FPR) και το ROC-AUC.

Τα αποτελέσματα έδειξαν ότι υπάρχουν διαφορές μεταξύ των εποπτευόμενων και των μη εποπτευόμενων προσεγγίσεων. Συγκρίθηκε η απόδοση όλων των μοντέλων που αξιολογήθηκαν, και το μοντέλο Random Forest παρουσίασε την καλύτερη συνολική απόδοση, με ικανοποιητική αποτελεσματικότητα ανίχνευσης. Τα μη εποπτευόμενα μοντέλα, ωστόσο, δεν κατάφεραν να εντοπίσουν με την ίδια επιτυχία τις επιθέσεις zero-day, ειδικά όταν η κακόβουλη κίνηση ήταν παρόμοια με την καλοήγητη κίνηση του δικτύου.

Τα αποτελέσματα στο σύνολό τους δείχνουν τις προοπτικές της μηχανικής μάθησης για τη βελτίωση των συστημάτων ανίχνευσης εισβολών, καθώς και τη σημασία της επιλογής του μοντέλου, της ποιότητας των δεδομένων και της μηχανικής χαρακτηριστικών για την παροχή ικανοποιητικών αποτελεσμάτων.

«AI in Cybersecurity»

«Emmanouel Karamanolis»

Abstract

The more sophisticated and frequent attacks on cyber security have led to the need for more sophisticated intrusion detection mechanisms that can detect zero-day attacks. Typical Intrusion Detection Systems (IDS) will use signatures and hence are not able to detect zero day attacks. The use of Machine Learning (ML) has become an intriguing way to enhance the flexibility and effectiveness of cybersecurity systems.

This thesis focuses on machine learning methods for the detection of zero day cyberattacks using network traffic data. The supervised and unsupervised techniques were both analyzed to assess their performance in detecting malicious activities not included in the training period. The Logistic Regression, Support Vector Machines (SVM), Random Forest, K-Means Clustering, Isolation Forest, and One-Class SVM models were chosen for this study.

The data preprocessing, feature selection, model training, and performance evaluation were the steps in the experimental process. The performance evaluation of the models was carried out based on some of the most relevant metrics for intrusion detection such as Recall, F1-Score, False Positive Rate (FPR), and ROC-AUC.

The results showed that there are differences between supervised and unsupervised approaches. The performance of all the evaluated models was compared, and Random Forest showed the best overall performance, with great detection ability and a favorable detection effectiveness. The unsupervised models, however, were not as successful in correctly identifying zero-day attacks, especially when the malicious traffic was similar to normal network traffic.

The results as a whole demonstrate the potential of machine learning to improve intrusion detection systems, and the significance of model selection, data quality, and feature engineering in providing robust zero-day attack detection.

Ευχαριστίες

Αρχικά θα ήθελα να εκφράσω την ειλικρινή μου ευγνωμοσύνη στον κ. Χρήστο Ηλιούδη, επόπτη της διπλωματικής μου εργασίας, για την πολύτιμη καθοδήγησή του, την υποστήριξή του και τις ουσιαστικές συμβουλές του σε κάθε στάδιο της ανάπτυξης του έργου. Η συνεισφορά του υπήρξε καθοριστική για τη διεκπεραίωση της εργασίας αυτής.

Τέλος, θα ήθελα να ευχαριστήσω τους ανθρώπους που βρέθηκαν δίπλα μου όλα αυτά τα χρόνια. Την οικογένειά μου για την αμέριστη υποστήριξη και την εμπιστοσύνη που μου έδειξαν, καθώς και τους φίλους μου για την κατανόηση, την ενθάρρυνση και τις στιγμές που έκαναν αυτή τη διαδρομή πιο όμορφη. Χωρίς τη συμβολή τους, η ολοκλήρωση αυτής της προσπάθειας θα ήταν σαφώς πιο δύσκολη.

Περιεχόμενα

Πρόλογος.....	iii
Περίληψη.....	iv
Abstract.....	v
Ευχαριστίες.....	vi
Περιεχόμενα.....	vii
Κατάλογος Σχημάτων.....	xi
Συντομογραφίες.....	xiii
Κεφάλαιο 1ο:.....	Εισαγωγή
1	
1.1 Στόχοι της Δ.Ε.....	1
1.2 Δομή κεφαλαίων.....	2
1.3 Επίλογος.....	3
Κεφάλαιο 2ο:.....	Τεχνητή νοημοσύνη και συστήματα ανίχνευσης εισβολών στην κυβερνοασφάλεια
4	
2.1 Εισαγωγή στα συστήματα ανίχνευσης εισβολών.....	4
2.2 Τεχνητή νοημοσύνη στα συστήματα ανίχνευσης εισβολών.....	6
2.2.1 Εποπτευόμενες και μη εποπτευόμενες προσεγγίσεις.....	6
2.2.2 Μέτρα αξιολόγησης.....	7
2.2.3 Επίλογος.....	8
2.3 Εποπτευόμενη μηχανική μάθηση για την κυβερνοασφάλεια.....	8
2.3.1 Εποπτευόμενη μηχανική μάθηση.....	8
2.3.2 Μοντέλα εποπτευόμενης μηχανικής μάθησης.....	9
2.3.3 Δυνατά σημεία και περιορισμοί.....	13
2.3.4 Σύγκριση αλγορίθμων εποπτευόμενης μηχανικής μάθησης για την κυβερνοασφάλεια	13
2.4 Μη εποπτευόμενη μηχανική μάθηση για την κυβερνοασφάλεια.....	14
2.4.1 Μη εποπτευόμενη μηχανική μάθηση.....	14
2.4.2 Μοντέλα μη εποπτευόμενης μηχανικής μάθησης.....	14
2.4.3 Δυνατά σημεία και περιορισμοί.....	16
2.4.4 Σύγκριση αλγορίθμων μη εποπτευόμενης μηχανικής μάθησης για την κυβερνοασφάλεια.....	16
2.5 Προκλήσεις στα IDS που βασίζονται σε AI.....	17
2.6 Επίλογος.....	18

Κεφάλαιο 3ο:.....	Επιθέσεις μηδενικής ημέρας
20	
3.1	Εισαγωγή.....20
3.2	Χαρακτηριστικά Επιθέσεων Μηδενικής Ημέρας.....20
3.3	Ταξινόμηση και τύποι επιθέσεων μηδενικής ημέρας.....21
3.3.1	Κατηγοριοποίηση με βάση τον στόχο.....21
3.3.2	Κατηγοριοποίηση με βάση τη μέθοδο επίθεσης.....21
3.3.3	Κατηγοριοποίηση με βάση τον σκοπό της επίθεσης.....22
3.3.4	Παραδείγματα αξιοσημείωτων επιθέσεων Μηδενικής Ημέρας.....22
3.4	Τεχνικές ανίχνευσης για επιθέσεις μηδενικής ημέρας.....22
3.4.1	Ανίχνευση με βάση την υπογραφή.....22
3.4.2	Ανίχνευση με βάση τη συμπεριφορά.....22
3.4.3	Στατιστικές μέθοδοι και μέθοδοι ανίχνευσης ανωμαλιών.....22
3.4.4	Μηχανική μάθηση και τεχνητή νοημοσύνη.....23
3.4.5	Τεχνικές βασισμένες στη βαθιά μάθηση.....23
3.4.6	Πληροφορίες απειλών και συνεργατική ανίχνευση.....23
3.5	Προκλήσεις στην ανίχνευση επιθέσεων μηδενικής ημέρας.....23
3.6	Μελέτες περιπτώσεων επιθέσεων μηδενικής ημέρας.....24
3.6.1	Stuxnet (2010).....24
3.6.2	Log4Shell (2021).....24
3.6.3	Συμπεράσματα από το Stuxnet και το Log4Shell.....24
3.7	Επίλογος.....24
Κεφάλαιο 4ο:.....	Επιλεγμένα μοντέλα μηχανικής μάθησης
26	
4.1	Εισαγωγή.....26
4.2	Logistic Regression.....26
4.3	Random Forest.....27
4.4	Support Vector Machine.....27
4.5	K-Means Ομαδοποίηση.....28
4.6	Isolation Forest.....28
4.7	One-Class Support Vector Machines.....29
4.8	Επίλογος.....29
Κεφάλαιο 5ο:.....	Εργαλεία και Περιβάλλον Λογισμικού
31	
5.1	Εισαγωγή.....31

5.2	Προγραμματιστικό Περιβάλλον.....	31
5.3	Βιβλιοθήκες και Frameworks.....	31
5.4	Προδιαγραφές Συστήματος.....	32
5.5	Επίλογος.....	32
Κεφάλαιο 6ο:.....Επεξεργασία βάσης δεδομένων και μέτρα αξιολόγησης		33
6.1	Εισαγωγή.....	33
6.2	Περιγραφή βάσης δεδομένων.....	33
6.3	Επεξεργασία δεδομένων.....	34
6.3.1	Καθαρισμός δεδομένων.....	35
6.3.2	Επιλογή και επεξεργασία χαρακτηριστικών.....	36
6.3.3	Train/test διαίρεση και σενάρια zero day.....	37
6.4	Μεθοδολογία αξιολόγησης.....	42
6.4.1	Confusion Matrix.....	42
6.4.2	Μετρήσεις απόδοσης.....	43
6.4.3	Τρόπος αξιολόγησης.....	43
6.5	Επίλογος.....	44
Κεφάλαιο 7ο:.....Εκπαίδευση μοντέλων μηχανικής μάθησης		45
7.1	Logistic Regression.....	45
7.1.1	Διαδικασία εκπαίδευσης.....	45
7.1.2	Αποτελέσματα.....	46
7.1.3	Συζήτηση αποτελεσμάτων.....	47
7.1.4	Συμπεράσματα.....	48
7.2	Support Vector Machines (SVM).....	48
7.2.1	Διαδικασία εκπαίδευσης.....	48
7.2.2	Αποτελέσματα.....	49
7.2.3	Συζήτηση αποτελεσμάτων.....	51
7.2.4	Συμπεράσματα.....	52
7.3	Random Forest.....	52
7.3.1	Διαδικασία εκπαίδευσης.....	52
7.3.2	Αποτελέσματα.....	52
7.3.3	Ανάλυση κατωφλίου απόφασης.....	54
7.3.4	Συζήτηση αποτελεσμάτων.....	56
7.3.5	Συμπεράσματα.....	57

7.4	K-Means ομαδοποίηση.....	57
7.4.1	Διαδικασία εκπαίδευσης.....	57
7.4.2	Αποτελέσματα.....	59
7.4.3	Ανάλυση συστάδων.....	59
7.4.4	Συζήτηση αποτελεσμάτων.....	60
7.4.5	Συμπεράσματα.....	61
7.5	Isolation Forest.....	61
7.5.1	Διαδικασία εκπαίδευσης.....	61
7.5.2	Αποτελέσματα.....	62
7.5.3	Ανάλυση Contamination.....	64
7.5.4	Συζήτηση αποτελεσμάτων.....	65
7.5.5	Συμπεράσματα.....	65
7.6	One-Class Support-Vector-Machines.....	66
7.6.1	Διαδικασία Εκπαίδευσης.....	66
7.6.2	Αποτελέσματα.....	67
7.6.3	Ανάλυση nu.....	69
7.6.4	Συζήτηση αποτελεσμάτων.....	70
7.6.5	Συμπεράσματα.....	70
7.7	Συνολικά αποτελέσματα.....	71
Κεφάλαιο 8ο:	Συμπεράσματα και προτάσεις	73
8.1	Συμπεράσματα.....	73
8.2	Περιορισμοί της Δ.Ε.....	74
8.3	Μελλοντικές προτάσεις.....	74
8.4	Τελικές παρατηρήσεις.....	75
	BIBΛΙΟΓΡΑΦΙΑ.....	77

Κατάλογος Σχημάτων

Εικόνα 2.1 Βασική αρχιτεκτονική IDS.....	4
Εικόνα 2.2 Τρόπος λειτουργίας των δύο κατηγοριών.....	7
Εικόνα 2.3 Τρόπος εκπαίδευσης Εποπτευόμενων μοντέλων	9
Εικόνα 2.4 Decision Trees	10
Εικόνα 2.5 Random Forest	10
Εικόνα 2.6 Support Vector Machines	11
Εικόνα 2.7 Logistic Regression	12
Εικόνα 2.8 Artificial Neural Networks	12
Εικόνα 2.9 Αλγόριθμος K-Means	15
Εικόνα 2.10 One-Class Support Vector Machines	15
Εικόνα 2.11 Isolation Forest	16
Εικόνα 6.1 Συγχώνευση των αρχείων της βάσης δεδομένων CICIDS2018.....	34
Εικόνα 6.2 Επιλογή κοινών χαρακτηριστικών και αποθήκευση ενιαίου συνόλου δεδομένων.....	35
Εικόνα 6.3 Αφαίρεση του Timestamp, μετατροπή των χαρακτηριστικών σε αριθμητική μορφή, αντικατάσταση άπειρων τιμών με NaN και αφαίρεση NaN εγγραφών	36
Εικόνα 6.4 Αφαίρεση διπλότυπων εγγραφών, έλεγχος τελικής διάστασης του συνόλου δεδομένων και αποθήκευση του dataset.....	36
Εικόνα 6.5 Κώδικας ανάλυσης συσχέτισης των χαρακτηριστικών.....	37
Εικόνα 6.6 Δημιουργία πρώτου σεναρίου.....	39
Εικόνα 6.7 Δημιουργία δεύτερου σεναρίου.....	40
Εικόνα 6.8 υποδειγματοληψία στο Σενάριο Α.....	41
Εικόνα 6.9 υποδειγματοληψία στο Σενάριο Β.....	42
Εικόνα 7.1 Μέθοδος Min-Max και εκπαίδευση μοντέλου με το σενάριο Α.....	45
Εικόνα 7.2 Αποτελέσματα Α Σεναρίου.....	46
Εικόνα 7.3 Αποτελέσματα Β Σεναρίου.....	47
Εικόνα 7.4 Logistic Regression Σύγκριση Αποτελεσμάτων.....	48
Εικόνα 7.5 Εκπαίδευση SVM μοντέλου.....	49
Εικόνα 7.6 Αποτελέσματα SVM Σενάριο Α.....	50

Εικόνα 7.7 Αποτελέσματα SVM Σενάριο B.....	51
Εικόνα 7.8 SVM Σύγκριση Αποτελεσμάτων.....	51
Εικόνα 7.9 Εκπαίδευση μοντέλου Random Forest.....	52
Εικόνα 7.10 Αποτελέσματα Σεναρίου A.....	53
Εικόνα 7.11 Αποτελέσματα Σεναρίου B.....	53
Εικόνα 7.12 Κώδικας για την Ανάλυση Κατωφλίου Απόφασης.....	54
Εικόνα 7.13 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση το κατώφλι απόφασης στο Σενάριο A.....	55
Εικόνα 7.14 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση το κατώφλι απόφασης στο Σενάριο B.....	56
Εικόνα 7.15 Σύγκριση των δύο Σεναρίων.....	57
Εικόνα 7.16 Εκπαίδευση Μοντέλου K-means.....	58
Εικόνα 7.17 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση των αριθμό συστάδων στο Σενάριο A.....	59
Εικόνα 7.18 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση των αριθμό συστάδων στο Σενάριο B.....	60
Εικόνα 7.19 Εκπαίδευση Μοντέλου Isolation Forest.....	62
Εικόνα 7.20 Αποτελέσματα Isolation Forest Σενάριο A.....	63
Εικόνα 7.21 Ανάλυση Contamination στα δύο Σενάρια Isolation Forest.....	65
Εικόνα 7.22 Εκπαίδευση Μοντέλου One-Class SVM.....	67
Εικόνα 7.23 Αποτελέσματα One-Class SVM Σενάριο A.....	68
Εικόνα 7.24 Αποτελέσματα One-Class SVM Σενάριο B.....	69
Εικόνα 7.25 Ανάλυση nu στα δύο Σενάρια One-Class SVM.....	70

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
IDS	Intrusion Detection Systems
ML	Machine Learning
SVM	Support Vector Machines
FPR	False Positive Rate
AI	Artificial Intelligence
DoS	Denial of Service

Κεφάλαιο 1ο: Εισαγωγή

Η κυβερνοασφάλεια αποτελεί κρίσιμο τομέα για οργανισμούς, επιχειρήσεις και χρήστες του διαδικτύου. Η αυξανόμενη διασύνδεση των πληροφοριακών συστημάτων δημιουργεί περισσότερα σημεία πιθανής επίθεσης και αυξάνει την ανάγκη για αποτελεσματικούς μηχανισμούς προστασίας. Οι παραδοσιακοί μηχανισμοί ασφαλείας, όπως το firewall (τοίχος προστασίας) και τα συστήματα ανίχνευσης εισβολών (IDS), που βασίζονται σε υπογραφές, τείνουν να μην ανιχνεύουν επιθέσεις μηδενικής ημέρας (zero-day attacks), αφού χρησιμοποιούν προκαθορισμένους κανόνες ή υπογραφές από προηγούμενες επιθέσεις [1].

Η τεχνητή νοημοσύνη (AI) και πιο συγκεκριμένα η μηχανική μάθηση (ML) μπορεί να αποτελέσει μια πιθανή λύση στο πρόβλημα της ανίχνευσης επιθέσεων σε ένα δίκτυο. Με τη βοήθεια της μηχανικής μάθησης υπάρχει η δυνατότητα να εκπαιδευτεί ένα μοντέλο μηχανικής μάθησης πάνω σε δεδομένα δικτυακής κίνησης, ώστε να εντοπίζει ανωμαλίες ή να ταξινομεί νέα δείγματα [2].

Το κύριο κομμάτι όταν σχεδιάζεται ένα AI που θα εφαρμοστεί σε ένα IDS είναι το πώς γίνεται η αξιολόγησή του. Η ακρίβεια δεν επαρκεί πάντοτε ως μοναδικός δείκτης απόδοσης. Σε σύνολα δεδομένων με μη ισορροπημένη κατανομή κλάσεων, η καλοήγη δικτυακή δραστηριότητα μπορεί να υπερτερεί σημαντικά της κακόβουλης δραστηριότητας. Σε αυτή την περίπτωση, ένα μοντέλο μπορεί να εμφανίζει υψηλή ακρίβεια, ακόμη και όταν δεν εντοπίζει επαρκώς τις επιθέσεις. Για τον λόγο αυτό, η αξιολόγηση της απόδοσης βασίζεται και σε πρόσθετους δείκτες όπως η ανάκληση (Recall), το ποσοστό ψευδών θετικών (FPR) και το F1-score [3].

Σημαντικό ρόλο έχουν επίσης οι υπολογιστικοί πόροι που απαιτούνται για τη λειτουργία του συστήματος. Ένα μοντέλο με υψηλή θεωρητική απόδοση δεν είναι απαραίτητα κατάλληλο για χρήση σε πραγματικό περιβάλλον. Πρέπει να λειτουργεί έγκαιρα και με αποδεκτό υπολογιστικό κόστος [4]. Επιπλέον, τα συστήματα τεχνητής νοημοσύνης στην κυβερνοασφάλεια πρέπει να είναι ανθεκτικά σε απόπειρες χειραγώγησης των εισόδων τους. Τέτοιες επιθέσεις μπορούν να οδηγήσουν ένα μοντέλο σε λανθασμένες ταξινομήσεις [5].

Η παρούσα διπλωματική εργασία εξετάζει μοντέλα μηχανικής μάθησης με στόχο την ανίχνευση κυβερνοεπιθέσεων μέσω δεδομένων δικτυακής κίνησης. Τα μοντέλα εκπαιδεύτηκαν, ελέγχθηκαν και αξιολογήθηκαν με μετρικές αξιολόγησης, όπως η ανάκληση (Recall), σε ένα σύνολο δεδομένων που περιλαμβάνει τόσο καλοήγη δραστηριότητα (benign traffic) όσο και διάφορους τύπους επιθέσεων.

1.1 Στόχοι της Δ.Ε.

Το κύριο ζήτημα που τίγεται αυτή η διπλωματική είναι η ανίχνευση επιθέσεων σε ένα δίκτυο με τη χρήση μοντέλων μηχανικής μάθησης. Τα παραδοσιακά συστήματα ανίχνευσης εισβολών που βασίζονται σε υπογραφές χρησιμοποιούν γνωστά πρότυπα και κανόνες επιθέσεων. Για τον λόγο αυτό, παρουσιάζουν περιορισμούς στην ανίχνευση επιθέσεων μηδενικής ημέρας [1]. Τα μοντέλα μηχανικής μάθησης μπορούν να εκπαιδευτούν σε δεδομένα δικτυακής δραστηριότητας, ώστε να εντοπίζουν αποκλίσεις από την καλοήγη συμπεριφορά ή να ταξινομούν νέα δείγματα κίνησης. Με τον τρόπο αυτό, μπορούν να υποστηρίξουν την ανίχνευση ύποπτης δραστηριότητας στο δίκτυο [2].

Η αξιολόγηση των μοντέλων αποτελεί σημαντικό στάδιο της διαδικασίας. Η ανάκληση (Recall) δείχνει το ποσοστό των πραγματικών επιθέσεων που εντοπίζονται σωστά. Το ποσοστό ψευδώς θετικών (False

Positive Rate – FPR) δείχνει πόσο συχνά η καλοήθης δραστηριότητα ταξινομείται λανθασμένα ως κακόβουλη. Υψηλό FPR μπορεί να αυξήσει τον αριθμό των ψευδών ειδοποιήσεων και να επιβαρύνει τη λειτουργία του συστήματος. Σε σύνολα δεδομένων με άνιση κατανομή καλοήθης και κακόβουλης δραστηριότητας, το F1-score παρέχει επιπλέον πληροφορίες για την απόδοση του μοντέλου [3].

Οι στόχοι της διπλωματικής είναι:

- I. Καθαρισμός και προετοιμασία του συνόλου δεδομένων για την εκπαίδευση των μοντέλων μηχανικής μάθησης.
- II. Εκπαίδευση των επιλεγμένων μοντέλων μηχανικής μάθησης.
- III. Αξιολόγηση κάθε μοντέλου με ανάκληση, FPR, F1-score, ώστε να προκύπτει ρεαλιστική αποτίμηση της απόδοσής του.
- IV. Σύγκριση των μοντέλων, με στόχο την ανάδειξη της βέλτιστης προσέγγισης για εφαρμογή σε σύστημα ανίχνευσης εισβολών.

Με την ολοκλήρωση των παραπάνω σταδίων, τα επιλεγμένα μοντέλα συγκρίνονται ως προς την ικανότητά τους να εντοπίζουν επιθέσεις και να περιορίζουν τις λανθασμένες ειδοποιήσεις.

1.2 Δομή κεφαλαίων

Το κείμενο της Δ.Ε. χωρίζεται σε 8 κεφάλαια. Η Δ.Ε. ξεκινά με την εισαγωγή του αναγνώστη στο θεωρητικό υπόβαθρο, συνεχίζει με την ανάλυση του πειραματικού μέρους και καταλήγει στη διατύπωση των συμπερασμάτων της έρευνας και μελλοντικών προτάσεων.

Το πρώτο κεφάλαιο της Δ.Ε. είναι η εισαγωγή. Στην εισαγωγή σχολιάζονται τα κίνητρα για την έρευνα στον τομέα της κυβερνοασφάλειας, τίθενται οι στόχοι της Δ.Ε. και τονίζεται η βαρύτητα του θέματος. Με την ανάγνωση του πρώτου κεφαλαίου, ο αναγνώστης μπορεί να κατανοήσει το υπό εξέταση πρόβλημα και τον ρόλο της τεχνητής νοημοσύνης στα συστήματα ανίχνευσης εισβολών.

Το δεύτερο κεφάλαιο παρουσιάζει το θεωρητικό υπόβαθρο και τη σχετική βιβλιογραφία για τις εφαρμογές της τεχνητής νοημοσύνης στα συστήματα ανίχνευσης εισβολών και στην κυβερνοασφάλεια. Εξετάζονται η εποπτευόμενη και η μη εποπτευόμενη μηχανική μάθηση, με έμφαση στα πλεονεκτήματα, στους περιορισμούς και στις εφαρμογές τους σε προηγούμενες έρευνες.

Το τρίτο κεφάλαιο εξετάζει τις επιθέσεις μηδενικής ημέρας, οι οποίες αποτελούν σημαντική πρόκληση για τη σύγχρονη κυβερνοασφάλεια. Παρουσιάζονται τα χαρακτηριστικά τους, οι τεχνικές ανίχνευσής τους, οι βασικές προκλήσεις που σχετίζονται με αυτές και χαρακτηριστικές πραγματικές περιπτώσεις, όπως Stuxnet και Log4Shell.

Το τέταρτο κεφάλαιο παρουσιάζει τα μοντέλα μηχανικής μάθησης που εφαρμόστηκαν στην παρούσα μελέτη. Αναλύεται το θεωρητικό υπόβαθρο των επιλεγμένων αλγορίθμων, όπως Logistic Regression, Random Forest, Support Vector Machines, K-Means Clustering, Isolation Forest και One-Class SVM.

Στο πέμπτο κεφάλαιο περιγράφονται τα εργαλεία λογισμικού και το περιβάλλον ανάπτυξης που χρησιμοποιήθηκαν για τη διεξαγωγή των πειραμάτων. Παρουσιάζονται η γλώσσα προγραμματισμού, οι βιβλιοθήκες και τα πλαίσια που αξιοποιήθηκαν για την επεξεργασία των δεδομένων και την εφαρμογή των μοντέλων.

Το έκτο κεφάλαιο παρουσιάζει το σύνολο δεδομένων που χρησιμοποιήθηκε στην έρευνα και τη μεθοδολογία προεπεξεργασίας του. Επιπλέον περιγράφονται τα μέτρα αξιολόγησης που εφαρμόστηκαν για τη μέτρηση της απόδοσης των μοντέλων.

Το έβδομο κεφάλαιο περιγράφει τη διαδικασία εκπαίδευσης των επιλεγμένων μοντέλων και παρουσιάζει τα πειραματικά αποτελέσματα. Παράλληλα, γίνεται σύγκριση της απόδοσης των μοντέλων με βάση τα επιλεγμένα μέτρα αξιολόγησης.

Το όγδοο κεφάλαιο συνοψίζει τα βασικά ευρήματα της μελέτης και διατυπώνει τα τελικά συμπεράσματα. Επιπλέον, παρουσιάζονται περιορισμοί της εργασίας και προτάσεις για μελλοντική έρευνα.

1.3 Επίλογος

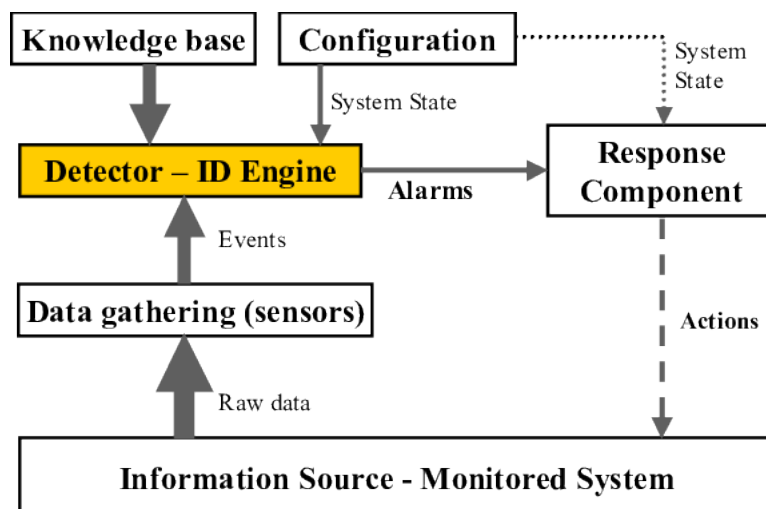
Τα παραδοσιακά συστήματα ανίχνευσης εισβολών συχνά παρουσιάζουν περιορισμούς απέναντι στη συνεχή εξέλιξη και αύξηση των κυβερνοεπιθέσεων. Η αξιοποίηση της τεχνητής νοημοσύνης μπορεί να υποστηρίξει τους ειδικούς κυβερνοασφάλειας στην αποτελεσματικότερη ανίχνευση ύποπτης δραστηριότητας. Με τη σύγκριση των μοντέλων μηχανικής μάθησης, η Δ.Ε. επιδιώκει να εντοπίσει την καταλληλότερη προσέγγιση για την ανίχνευση επιθέσεων σε δίκτυα υπολογιστών.

Κεφάλαιο 2ο: Τεχνητή νοημοσύνη και συστήματα ανίχνευσης εισβολών στην κυβερνοασφάλεια

2.1 Εισαγωγή στα συστήματα ανίχνευσης εισβολών

Τα δίκτυα υπολογιστών αποτελούν σημαντικό στόχο κυβερνοεπιθέσεων, καθώς χρησιμοποιούνται σε επιχειρήσεις, στην επικοινωνία και σε κρίσιμες υποδομές. Οι μηχανισμοί προληπτικής προστασίας, όπως τα τείχη προστασίας, δεν επαρκούν πάντοτε για την πλήρη ασφάλεια ενός δικτύου. Για τον λόγο αυτό, απαιτούνται μηχανισμοί που παρακολουθούν ενεργά τη δραστηριότητα του δικτύου ή των συστημάτων και εντοπίζουν πιθανές κακόβουλες ενέργειες. Τα συστήματα ανίχνευσης εισβολών (IDS) αποτελούν βασικό στοιχείο της αρχιτεκτονικής κυβερνοασφάλειας, καθώς συμβάλλουν στον εντοπισμό ύποπτης ή κακόβουλης δραστηριότητας [4].

Ένα σύστημα ανίχνευσης εισβολών (IDS) είναι σχεδιασμένο να παρακολουθεί και να αναλύει τη δικτυακή κίνηση ή τη δραστηριότητα ενός συστήματος, με στόχο τον εντοπισμό ύποπτων ενεργειών που ενδέχεται να υποδεικνύουν επίθεση. Ο σκοπός του είναι να εντοπίζει πιθανές απειλές και να ειδοποιεί τον διαχειριστή του δικτύου ή έναν αυτοματοποιημένο μηχανισμό άμυνας, ώστε να μπορεί να ληφθεί κατάλληλη ενέργεια αντιμετώπισης. Με την πάροδο του χρόνου, τα συστήματα IDS έχουν εξελιχθεί σημαντικά, καθώς οι κυβερνοεπιθέσεις και τα δίκτυα γίνονται περισσότερο σύνθετα [6].



Εικόνα 2.1 Βασική αρχιτεκτονική IDS [43]

Τα συστήματα ανίχνευσης εισβολών μπορούν να καταταχθούν σε δύο κατηγορίες, με βάση την πηγή των δεδομένων που παρακολουθούν.

- **Host-Based Intrusion Detection Systems (HIDS):** Τα HIDS παρακολουθούν μεμονωμένες συσκευές, όπως διακομιστές, υπολογιστές ή εικονικές μηχανές. Αναλύουν δεδομένα σε επίπεδο συστήματος, όπως κλήσεις συστήματος, αρχεία DLL, αρχεία καταγραφής και κλειδιά μητρώου, με στόχο τον εντοπισμό ύποπτης ή κακόβουλης δραστηριότητας [7].
- **Network-Based Intrusion Detection Systems (NIDS):** Τα NIDS παρακολουθούν τη δικτυακή κίνηση, αναλύοντας πακέτα, ροές δικτύου και μοτίβα επικοινωνίας. Τοποθετούνται συνήθως σε επιλεγμένα σημεία του δικτύου, ώστε να επιβλέπουν την επικοινωνία πολλαπλών συσκευών.

Είναι ιδιαίτερα χρήσιμα για τον εντοπισμό επιθέσεων που εκδηλώνονται μέσω της δικτυακής κίνησης, όπως η σάρωση θυρών και οι επιθέσεις άρνησης υπηρεσίας (DoS) [1].

Πέρα από την προηγούμενη διάκριση, τα Συστήματα Ανίχνευσης Εισβολών μπορούν να κατηγοριοποιηθούν και με βάση τον τρόπο ανίχνευσης που χρησιμοποιούν.

- **Signature-Based Detection:** Η ανίχνευση με βάση υπογραφές χρησιμοποιεί προκαθορισμένα μοτίβα γνωστών επιθέσεων, ώστε να εντοπίζει κακόβουλη δραστηριότητα. Η προσέγγιση αυτή είναι αποτελεσματική για επιθέσεις που έχουν ήδη αναγνωριστεί και καταγραφεί. Ωστόσο, παρουσιάζει περιορισμούς στον εντοπισμό επιθέσεων zero-day, για τις οποίες δεν υπάρχουν διαθέσιμες υπογραφές [4].
- **Anomaly-Based Detection:** Η ανίχνευση με βάση ανωμαλίες δημιουργεί ένα μοντέλο της καλοηθούς δικτυακής κίνησης ή της συνηθισμένης συμπεριφοράς ενός συστήματος. Κάθε σημαντική απόκλιση από αυτό το μοντέλο μπορεί να θεωρηθεί πιθανή απειλή. Η προσέγγιση αυτή μπορεί να υποστηρίξει τον εντοπισμό άγνωστων επιθέσεων, αλλά συχνά παρουσιάζει αυξημένο ποσοστό ψευδώς θετικών αποτελεσμάτων [8].
- **Hybrid Approaches:** Οι υβριδικές προσεγγίσεις συνδυάζουν την ανίχνευση με βάση υπογραφές και την ανίχνευση με βάση ανωμαλίες. Στόχος τους είναι να αξιοποιούν τα πλεονεκτήματα και των δύο μεθόδων, βελτιώνοντας την αποτελεσματικότητα και την προσαρμοστικότητα του συστήματος [8].

Ενώ τα IDS αποτελούν βασικό στοιχείο της κυβερνοασφάλειας, τα παραδοσιακά συστήματα αντιμετωπίζουν αρκετούς περιορισμούς.

1. **Προβλήματα επεκτασιμότητας:** Με την αύξηση του όγκου της δικτυακής κίνησης, τα συστήματα ανίχνευσης εισβολών μπορεί να δυσκολεύονται να επεξεργαστούν και να αναλύσουν μεγάλα ποσά δεδομένων σε πραγματικό χρόνο, χωρίς να επηρεάζεται η απόδοσή τους [9].
2. **Υψηλά ποσοστά ψευδώς θετικών:** Συγκεκριμένα στα Anomaly-Based IDS, η καλοήθης κίνηση μπορεί να χαρακτηριστεί λανθασμένα ως απειλή. Όταν συμβαίνει σε μεγάλο βαθμό, υπερφορτώνει τους διαχειριστές του συστήματος με αχρείαστους συναγερμούς [1].
3. **Αδυναμία ανίχνευσης επιθέσεων μηδενικής ημέρας:** Τα Signature-Based IDS παρουσιάζουν περιορισμούς στον εντοπισμό επιθέσεων για τις οποίες δεν υπάρχει διαθέσιμη ή ήδη καταγεγραμμένη υπογραφή [6].
4. **Προκλήσεις κρυπτογραφημένης κυκλοφορίας:** Η αυξανόμενη χρήση κρυπτογράφησης στη δικτυακή κίνηση δυσκολεύει την ανάλυση των δεδομένων από τα NIDS. Ειδικότερα, η επιθεώρηση του ωφέλιμου φορτίου των πακέτων δεν είναι δυνατή χωρίς αποκρυπτογράφηση, γεγονός που μπορεί να περιορίσει την αποτελεσματικότητα της ανίχνευσης [9].

Αυτοί οι περιορισμοί έχουν ενισχύσει την ανάγκη για την ενσωμάτωση της τεχνητής νοημοσύνης (AI) και της μηχανικής μάθησης στα IDS. Τα IDS που χρησιμοποιούν AI στοχεύουν στο να μαθαίνουν από τα δεδομένα, να προσαρμόζονται στις εξελιγμένες απειλές και να βελτιώνουν την ακρίβεια, ενώ παράλληλα μειώνουν τους ψευδείς συναγερμούς. Στη συνέχεια, εξετάζεται η ενσωμάτωση σε λεπτομέρεια.

2.2 Τεχνητή νοημοσύνη στα συστήματα ανίχνευσης εισβολών

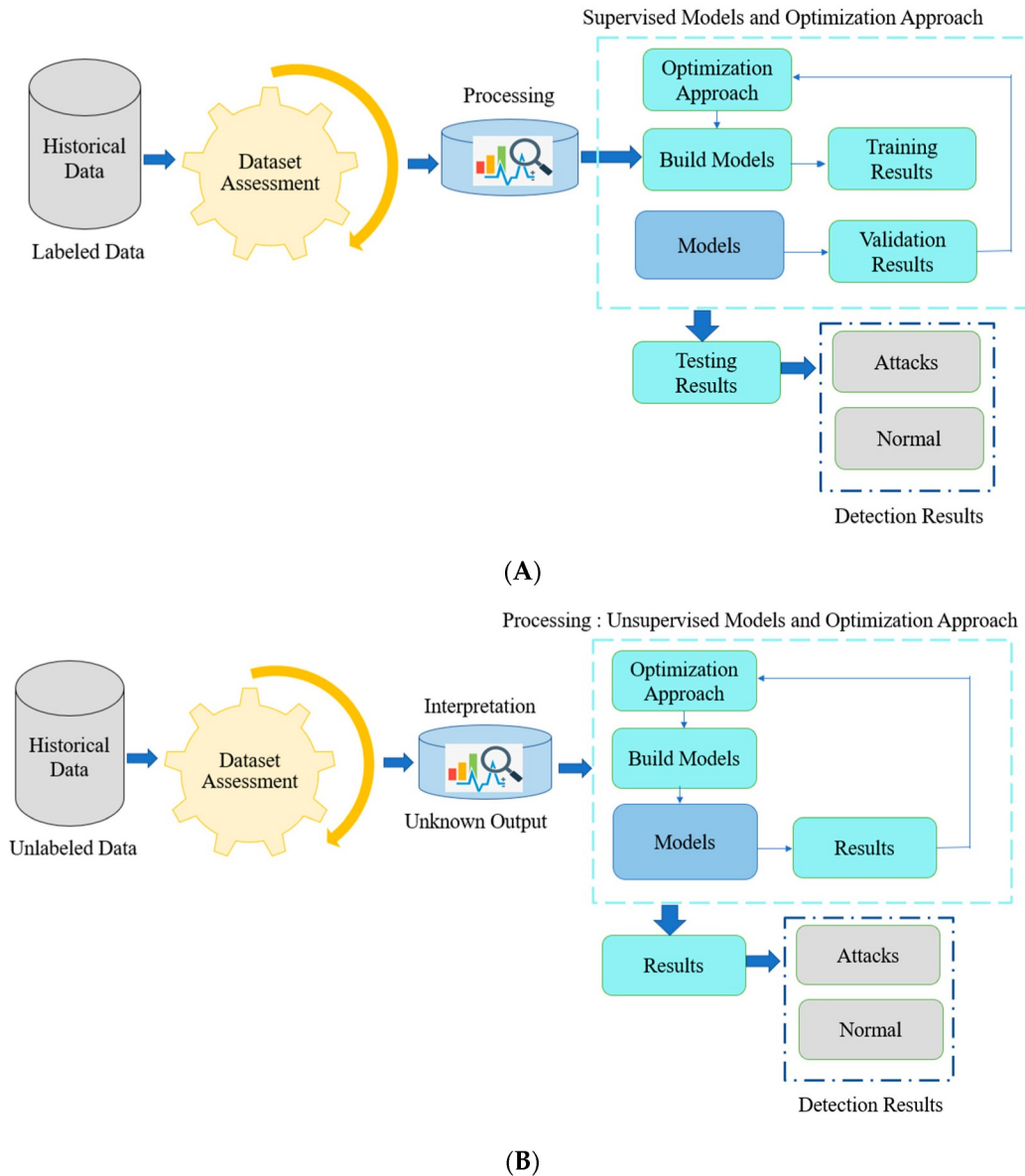
Τα παραδοσιακά IDS χρησιμοποιούνται για την προστασία δικτύων υπολογιστών από κακόβουλες δραστηριότητες. Παρόλα αυτά, όπως προαναφέρθηκε, τα Signature-Based IDS δεν μπορούν να ανιχνεύσουν επιθέσεις που δεν έχουν ήδη αναγνωριστεί, ενώ τα Anomaly-Based IDS ενδέχεται να εμφανίζουν υψηλά ποσοστά ψευδών συναγερμών. Η αύξηση της πολυπλοκότητας των επιθέσεων, σε συνδυασμό με την κλίμακα των σύγχρονων δικτύων, έχει αναδείξει τους περιορισμούς των παραδοσιακών IDS. Για τον λόγο αυτό, εξετάζονται προσεγγίσεις που αξιοποιούν τεχνικές τεχνητής νοημοσύνης και μηχανικής μάθησης [1].

Η τεχνητή νοημοσύνη (AI) και, πιο συγκεκριμένα η μηχανική μάθηση (ML) αποτελούν πιθανή λύση για την αντιμετώπιση των προκλήσεων που υπάρχουν στα IDS. Τα παραδοσιακά συστήματα που βασίζονται σε υπογραφές είναι αποτελεσματικά στον εντοπισμό γνωστών επιθέσεων, αλλά παρουσιάζουν περιορισμούς στην ανίχνευση νέων επιθέσεων. Αντίστοιχα, τα συστήματα που βασίζονται σε ανωμαλίες μπορούν να εντοπίζουν αποκλίσεις από την καλοήγη συμπεριφορά, αλλά ενδέχεται να εμφανίζουν αυξημένα ποσοστά ψευδών συναγερμών. Τα IDS που αξιοποιούν τεχνικές μηχανικής μάθησης μπορούν να αναλύουν δεδομένα από τη δραστηριότητα των συστημάτων και τη δικτυακή κίνηση, με στόχο τον εντοπισμό μοτίβων, ανωμαλιών και συσχετίσεων που ενδέχεται να μην είναι άμεσα εμφανείς μέσω χειροκίνητης ανάλυσης [2].

2.2.1 Εποπτευόμενες και μη εποπτευόμενες προσεγγίσεις

Η AI εφαρμογές στα IDS συχνά χωρίζονται σε δύο κατηγορίες

- **Εποπτευόμενη μάθηση:** Τα μοντέλα εκπαιδεύονται σε επισημασμένα σύνολα δεδομένων που περιέχουν παραδείγματα καλοήθους δικτυακής κίνησης αλλά και γνωστών επιθέσεων. Όταν ολοκληρωθεί η εκπαίδευση τους, μπορούν να ταξινομήσουν νέα δεδομένα δικτυακής κίνησης σύμφωνα με τα μοτίβα που έχουν μάθει. Εποπτευόμενες μέθοδοι, όπως τα Decision Trees, Random Forests, Support Vector Machines (SVM) και τα Νευρωνικά Δίκτυα, μπορούν να έχουν καλή απόδοση όταν τα δεδομένα εκπαίδευσης είναι σωστά επισημασμένα και υπάρχουν αρκετά παραδείγματα τόσο επιθέσεων όσο και καλοήθους δραστηριότητας. Ωστόσο, η ικανότητά τους να εντοπίζουν νέες επιθέσεις μπορεί να είναι περιορισμένη, όταν αυτές διαφέρουν σημαντικά από τα μοτίβα των δεδομένων εκπαίδευσης [8].
- **Μη εποπτευόμενη μάθηση:** Τα μη εποπτευόμενα μοντέλα εκπαιδεύονται σε δεδομένα χωρίς επισημάνσεις. Στόχος τους είναι να εντοπίζουν μοτίβα, συσχετίσεις ή αποκλίσεις στη δικτυακή κίνηση, χωρίς να γνωρίζουν εκ των προτέρων ποιες εγγραφές αντιστοιχούν σε καλοήγη ή κακόβουλη δραστηριότητα. Στα συστήματα ανίχνευσης εισβολών, σημαντικές αποκλίσεις από τη συνήθη συμπεριφορά μπορούν να θεωρηθούν πιθανές απειλές και να ενεργοποιήσουν συναγερμό. Αλγόριθμοι που μπορούν να χρησιμοποιηθούν σε τέτοιες προσεγγίσεις είναι οι K-Means Clustering, Isolation Forest και One-Class SVM. Βασικό πλεονέκτημα της μη εποπτευόμενης μάθησης είναι ότι μπορεί να υποστηρίξει τον εντοπισμό άγνωστων ή μη προηγουμένως παρατηρημένων επιθέσεων, χωρίς να απαιτούνται επισημασμένα παραδείγματα των συγκεκριμένων επιθέσεων. Ωστόσο, οι μέθοδοι αυτές ενδέχεται να εμφανίζουν αυξημένα ποσοστά ψευδώς θετικών αποτελεσμάτων, ιδιαίτερα όταν η καλοήγη δικτυακή δραστηριότητα παρουσιάζει μεγάλη ποικιλία [2].



Εικόνα 2.2 Τρόπος λειτουργίας των δύο κατηγοριών [44]

2.2.2 Μέτρα αξιολόγησης

Για τη σωστή αξιολόγηση των IDS που βασίζονται σε τεχνητή νοημοσύνη δεν αρκεί μόνο η ακρίβεια (Accuracy). Σε μη ισορροπημένα σύνολα δεδομένων όπου η καλοήθης δικτυακή κίνηση υπερσχύει των επιθέσεων, η ακρίβεια μπορεί να δίνει παραπλανητική εικόνα για την πραγματική ικανότητα ανίχνευσης του μοντέλου. Για τον λόγο αυτό, χρησιμοποιούνται συμπληρωματικές μετρικές αξιολόγησης.

- Ανάκληση (Recall): Εκφράζει το ποσοστό των πραγματικών επιθέσεων που εντοπίστηκαν σωστά από το μοντέλο.
- Ποσοστό ψευδώς θετικών αποτελεσμάτων (FPR): Εκφράζει το ποσοστό της καλοήθους δικτυακής κίνησης που χαρακτηρίστηκε λανθασμένα ως επίθεση.

- Precision: Εκφράζει το ποσοστό των περιπτώσεων που χαρακτηρίστηκαν ως επιθέσεις και ήταν πράγματι επιθέσεις.
- F1-Score: Αποτελεί τον αρμονικό μέσο όρο του Precision και του Recall και χρησιμοποιείται για την ισορροπημένη αξιολόγηση της απόδοσης ενός μοντέλου.
- ROC-AUC / PR-AUC: Αποτελούν συγκεντρωτικά μέτρα αξιολόγησης ταξινομητών σε διαφορετικά κατώφλια απόφασης. Οι καμπύλες Precision–Recall είναι ιδιαίτερα χρήσιμες όταν εξετάζεται η απόδοση της θετικής κλάσης σε μη ισορροπημένα δεδομένα [3].

2.2.3 Επίλογος

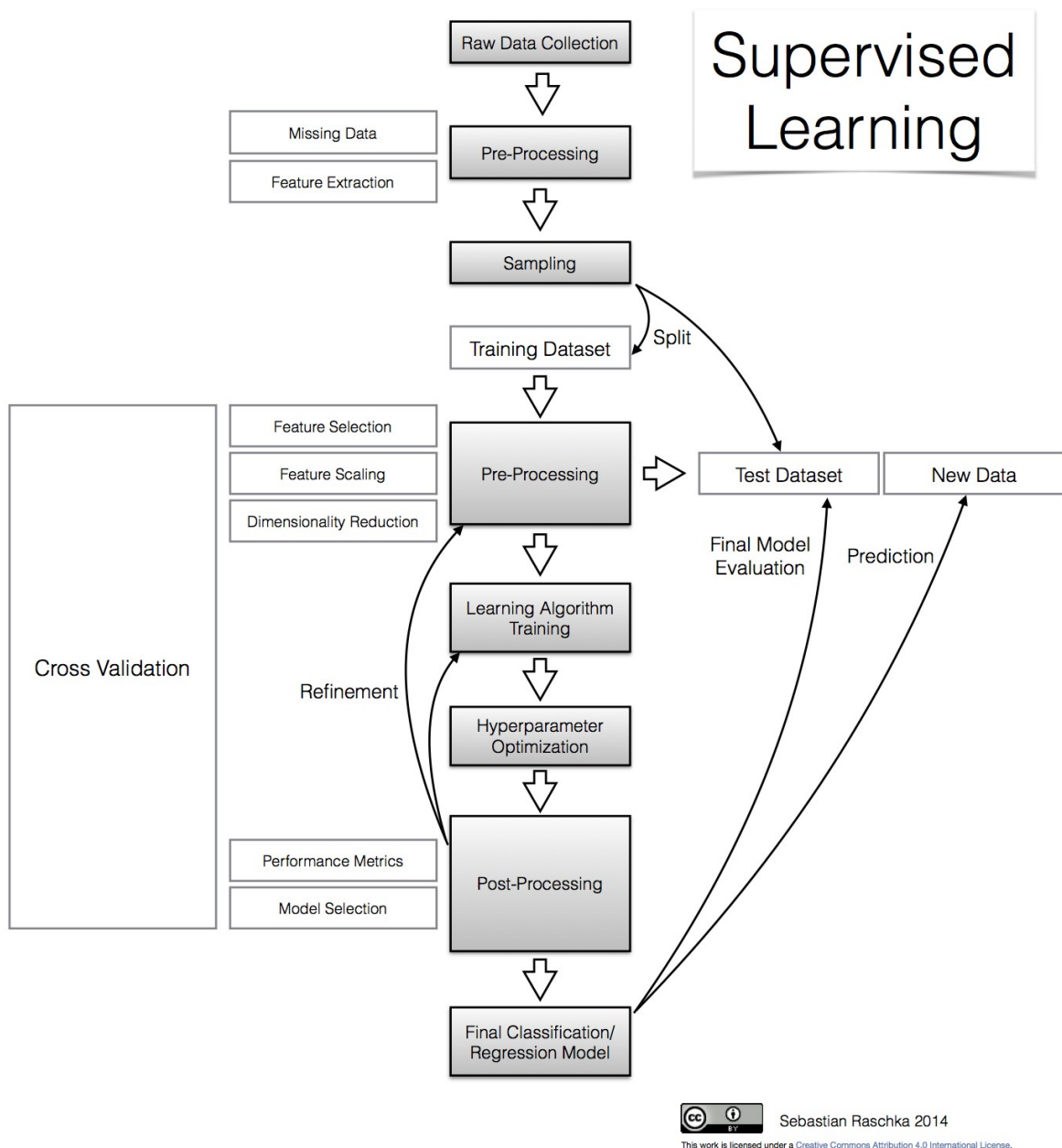
Το AI μπορεί να ενισχύσει τις δυνατότητες των IDS για την ανίχνευση γνωστών και άγνωστων επιθέσεων μέσω μοντέλων μηχανικής μάθησης που αναλύουν δεδομένα δικτυακής κίνησης. Τα εποπτευόμενα μοντέλα χρησιμοποιούνται κυρίως για την αναγνώριση γνωστών μοτίβων επιθέσεων, ενώ τα μη εποπτευόμενα μοντέλα μπορούν να συμβάλουν στον εντοπισμό αποκλίσεων που ενδέχεται να σχετίζονται με επιθέσεις μηδενικής ημέρας. Η συνδυαστική αξιοποίηση των δύο προσεγγίσεων με τη χρήση κατάλληλων μετρικών αξιολόγησης μπορεί να υποστηρίξει την αποτελεσματικότερη αξιολόγηση και λειτουργία των IDS [8], [3].

2.3 Εποπτευόμενη μηχανική μάθηση για την κυβερνοασφάλεια

2.3.1 Εποπτευόμενη μηχανική μάθηση

Η Εποπτευόμενη Μηχανική Μάθηση έχει αξιοποιηθεί σε πολλές προσεγγίσεις για την ανάπτυξη συστημάτων ανίχνευσης εισβολών που χρησιμοποιούν τεχνικές Τεχνητής Νοημοσύνης. Στην εποπτευόμενη μάθηση, τα μοντέλα εκπαιδεύονται σε επισημασμένα δεδομένα, στα οποία κάθε εγγραφή αντιστοιχεί σε μία προκαθορισμένη κλάση, όπως καλοήθης δραστηριότητα ή συγκεκριμένος τύπος επίθεσης. Κατά την εκπαίδευση, το μοντέλο μαθαίνει τη σχέση μεταξύ των χαρακτηριστικών των δεδομένων και της αντίστοιχης κλάσης. Μετά την εκπαίδευση, μπορεί να ταξινομεί νέα δεδομένα με βάση τα μοτίβα που έχει μάθει [6].

Η χρήση εποπτευόμενων μοντέλων διευκολύνεται από τη διαθεσιμότητα επισημασμένων συνόλων δεδομένων, όπως τα NSL-KDD, CICIDS2017, CICIDS2018. Τα σύνολα αυτά περιλαμβάνουν παραδείγματα καλοήθους δικτυακής κίνησης και διαφορετικών τύπων επιθέσεων, επιτρέποντας στους ερευνητές να εκπαιδεύουν και να αξιολογούν μοντέλα σε ελεγχόμενο περιβάλλον [10]. Ωστόσο, η εξάρτηση από επισημασμένα δεδομένα αποτελεί περιορισμό, καθώς τα πρότυπα της δικτυακής κίνησης και των επιθέσεων μπορεί να μεταβάλλονται με την πάροδο του χρόνου. Επομένως, ένα μοντέλο ενδέχεται να χρειάζεται επανεκπαίδευση με νεότερα δεδομένα, ώστε να διατηρεί την απόδοσή του [11].

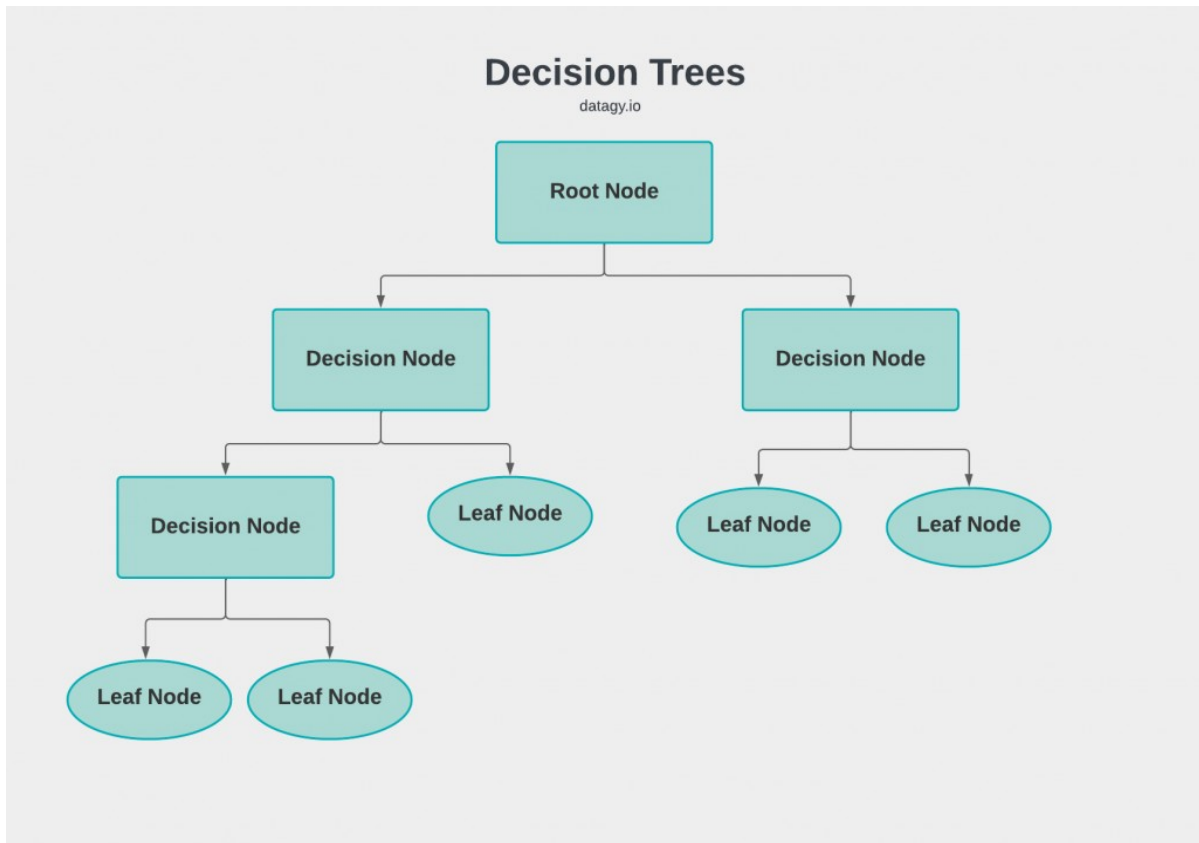


Εικόνα 2.3 Τρόπος εκπαίδευσης Εποπτευόμενων μοντέλων [45]

2.3.2 Μοντέλα εποπτευόμενης μηχανικής μάθησης

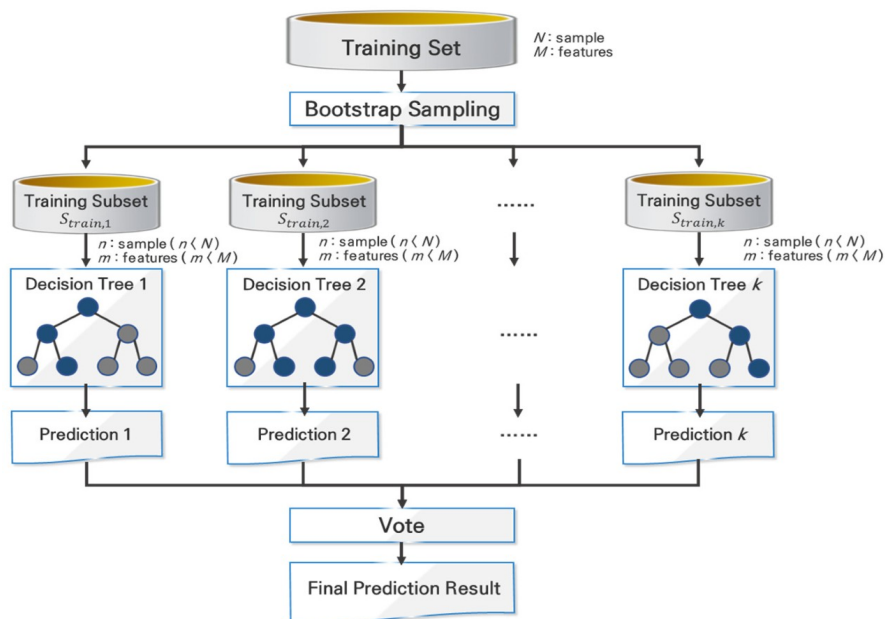
Αρκετοί αλγόριθμοι εποπτευόμενης μηχανικής μάθησης έχουν αποδειχθεί αποτελεσματικοί όπως:

- **Decision Trees (DT):** Τα Decision Trees χωρίζουν αναδρομικά τα δεδομένα σε κλάδους με βάση τις τιμές των χαρακτηριστικών τους και καταλήγουν σε μία τελική κατηγορία. Βασικό πλεονέκτημά τους είναι η ερμηνευσιμότητα, καθώς η διαδικασία ταξινόμησης μπορεί να παρουσιαστεί ως σαφής διαδρομή αποφάσεων. Ωστόσο, χωρίς κατάλληλη κλάδευση, ενδέχεται να υπερπροσαρμοστούν στα δεδομένα εκπαίδευσης και να παρουσιάσουν χαμηλότερη απόδοση σε νέα δεδομένα [12].



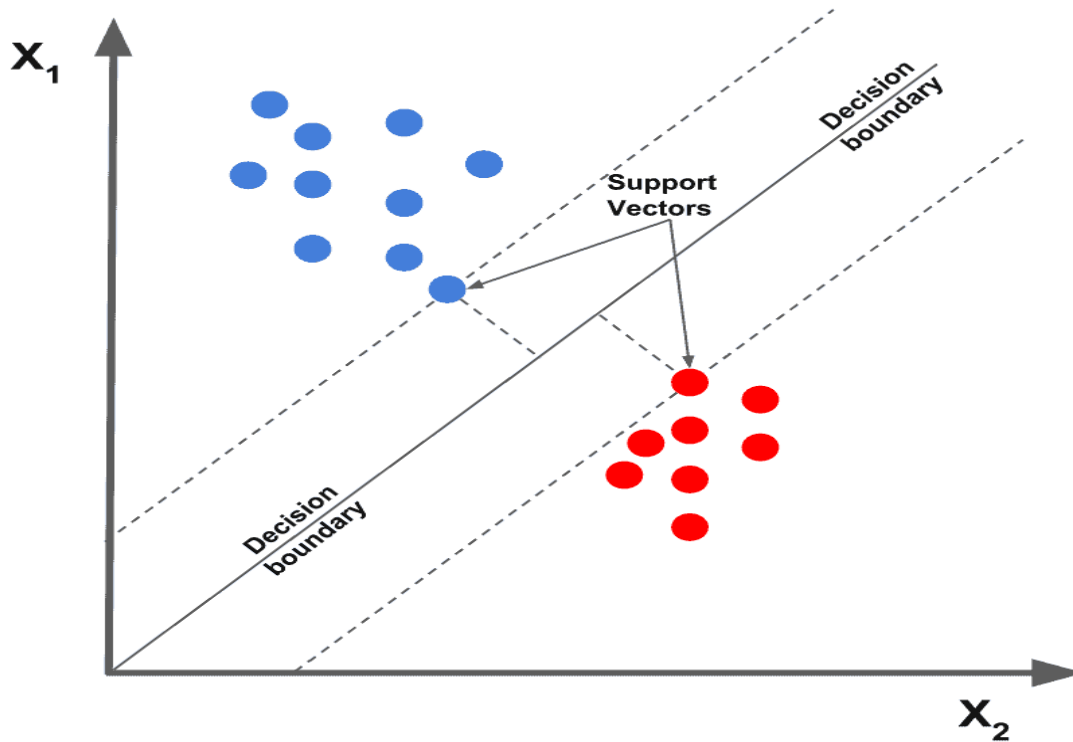
Εικόνα 2.4 Decision Trees [46]

- Random Forest (RF): Το Random Forest είναι ένα σύνολο από πολλά Decision Trees. Κάθε δέντρο εκπαιδεύεται σε τυχαίο υποσύνολο των δεδομένων και των χαρακτηριστικών τους, ενώ η τελική κατηγορία προκύπτει συνήθως μέσω πλειοψηφικής ψηφοφορίας. Η προσέγγιση αυτή μπορεί να βελτιώσει την ανθεκτικότητα και τη δυνατότητα γενίκευσης του μοντέλου [12].



Εικόνα 2.5 Random Forest [47]

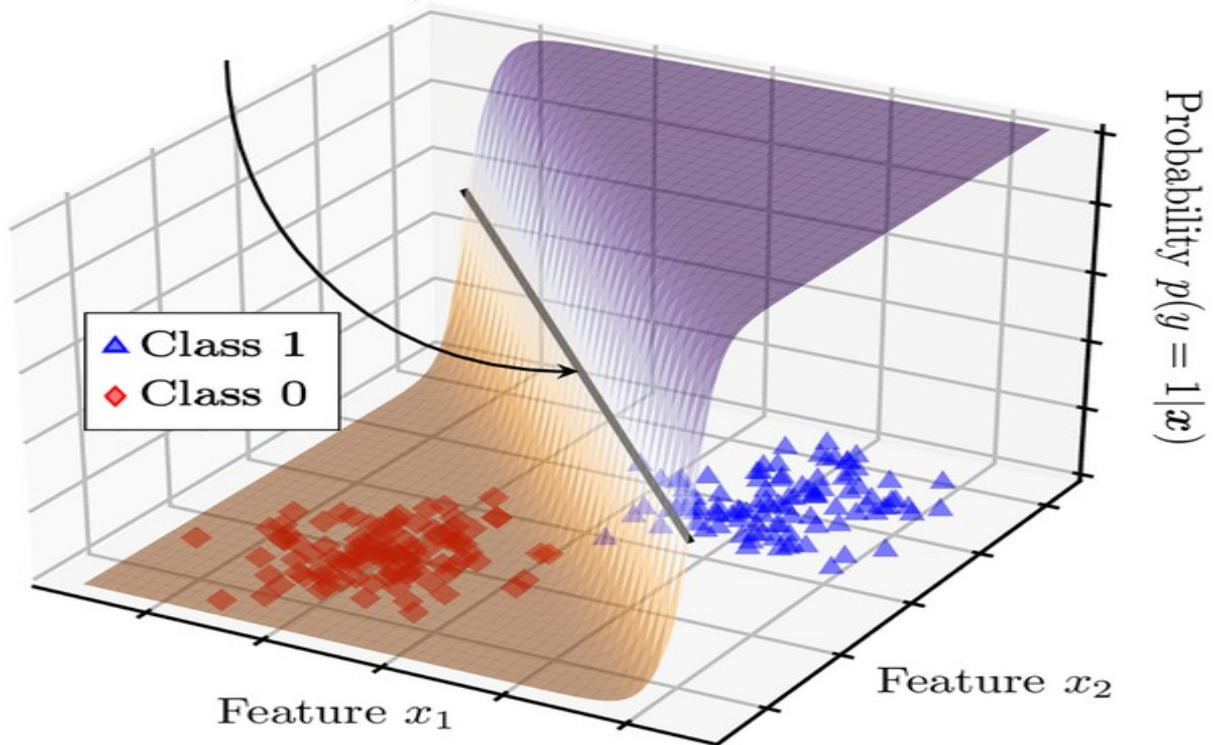
- Support Vector Machines (SVM): Το SVM χρησιμοποιείται για τη διάκριση της καλοήθους από την κακόβουλη δικτυακή κίνηση, αναζητώντας ένα υπερεπίπεδο που διαχωρίζει τις διαφορετικές κλάσεις. Με τη χρήση συναρτήσεων πυρήνα μπορούν να μοντελοποιούν μη γραμμικές σχέσεις μεταξύ των χαρακτηριστικών. Παρόλα αυτά, σε περιβάλλοντα IDS με μεγάλους όγκους δικτυακής κίνησης, η εκπαίδευση και η εφαρμογή του μπορεί να παρουσιάζουν αυξημένο υπολογιστικό κόστος και περιορισμένη κλιμάκωση [13].



Εικόνα 2.6 Support Vector Machines [48]

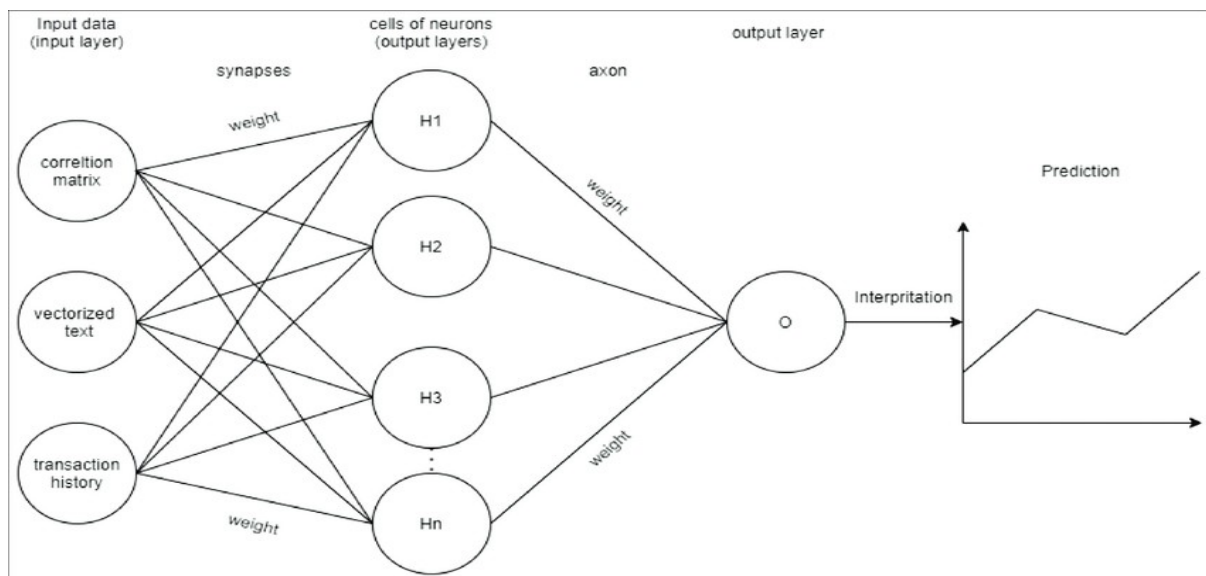
- Logistic Regression (LR): Η Logistic Regression εκτιμά την πιθανότητα μία εγγραφή να ανήκει σε συγκεκριμένη κλάση. Βασικό πλεονέκτημά της είναι η ερμηνευσιμότητα, καθώς οι συντελεστές του μοντέλου δείχνουν τη σχέση των χαρακτηριστικών με την τελική πρόβλεψη. Ωστόσο, οι γραμμικές παραδοχές της μπορεί να περιορίσουν την αποτελεσματικότητά της στην ανίχνευση σύνθετων ή μη γραμμικών μοτίβων επιθέσεων [14].

Decision boundary



Εικόνα 2.7 Logistic Regression [49]

- Artificial Neural Networks (ANN): Τα ANN αποτελούνται από επίπεδα νευρώνων που επεξεργάζονται τα δεδομένα εισόδου και παράγουν μία τελική πρόβλεψη. Στην εποπτευόμενη μάθηση, μπορούν να εκπαιδευτούν με τη μέθοδο της οπισθοδιάδοσης σφάλματος, ώστε να μαθαίνουν τη σχέση μεταξύ των χαρακτηριστικών εισόδου και της αντίστοιχης κλάσης. Τα ANN έχουν αξιοποιηθεί σε συστήματα ανίχνευσης εισβολών για την ανάλυση και ταξινόμηση της δικτυακής δραστηριότητας [15].



Εικόνα 2.8 Artificial Neural Networks [50]

2.3.3 Δυνατά σημεία και περιορισμοί

Η εποπτευόμενη μηχανική μάθηση βασίζεται σε επισημασμένα δεδομένα εκπαίδευσης και μπορεί να αξιοποιηθεί για τον εντοπισμό γνωστών μοτίβων επιθέσεων. Ωστόσο, η αποτελεσματικότητά της εξαρτάται από την ποιότητα και το μέγεθος των δεδομένων εκπαίδευσης, ενώ μπορεί να παρουσιάζει περιορισμούς απέναντι σε νέες απειλές [16].

2.3.4 Σύγκριση αλγορίθμων εποπτευόμενης μηχανικής μάθησης για την κυβερνοασφάλεια

Μοντέλο	Ερμηνευσιμότητα	Ικανότητα μοντελοποίησης μη γραμμικών σχέσεων	Υπολογιστικές απαιτήσεις και κλιμάκωση	Βασικός περιορισμός
Decision Trees [12],38	Υψηλή, καθώς η ταξινόμηση μπορεί να παρουσιαστεί μέσω διαδοχικών κανόνων και μιας διαδρομής απόφασης.	Μπορούν να αναπαραστήσουν μη γραμμικές σχέσεις μέσω διαδοχικών διαχωρισμών των δεδομένων.	Παρουσιάζουν σχετικά χαμηλές απαιτήσεις κατά την πρόβλεψη και μπορούν να χρησιμοποιηθούν για γρήγορη ταξινόμηση.	Είναι επιρρεπή σε υπερπροσαρμογή, ιδιαίτερα όταν το βάθος του δέντρου δεν περιορίζεται.
Logistic Regression [14], [38]	Υψηλή, καθώς οι συντελεστές του μοντέλου μπορούν να χρησιμοποιηθούν για την εξέταση της σχέσης των χαρακτηριστικών με την πρόβλεψη.	Περιορισμένη, επειδή στη βασική του μορφή δημιουργεί γραμμικό όριο απόφασης.	Παρουσιάζει σχετικά χαμηλές υπολογιστικές απαιτήσεις και μπορεί να εφαρμοστεί αποτελεσματικά σε μεγάλα σύνολα δεδομένων.	Μπορεί να εμφανίζει μειωμένη αποτελεσματικότητα όταν οι κλάσεις διαχωρίζονται μέσω σύνθετων μη γραμμικών σχέσεων.
Support Vector Machine [13], [40]	Μέτρια έως χαμηλή, ιδιαίτερα όταν χρησιμοποιούνται μη γραμμικές συναρτήσεις πυρήνα.	Μπορεί να αναπαραστήσει μη γραμμικές σχέσεις μέσω κατάλληλων συναρτήσεων πυρήνα.	Η εκπαίδευσή του μπορεί να γίνει υπολογιστικά απαιτητική σε σύνολα δεδομένων με μεγάλο αριθμό εγγραφών και χαρακτηριστικών.	Η απόδοσή του επηρεάζεται από την επιλογή του πυρήνα, των υπερπαραμέτρων και την κατάλληλη κανονικοποίηση των χαρακτηριστικών.
Artificial Neural Networks [12],	Χαμηλή, καθώς η εσωτερική	Μπορούν να μάθουν σύνθετα	Η εκπαίδευσή τους μπορεί να απαιτεί	Απαιτούν κατάλληλη

[15], [22]	διαδικασία λήψης αποφάσεων είναι συνήθως δύσκολο να ερμηνευτεί άμεσα.	και μη γραμμικά μοτίβα από τα δεδομένα.	σημαντικούς υπολογιστικούς πόρους, ανάλογα με το μέγεθος και την πολυπλοκότητα της αρχιτεκτονικής.	επιλογή αρχιτεκτονικής, υπερπαραμέτρων και επαρκή δεδομένα εκπαίδευσης.
------------	---	---	--	---

Ο παραπάνω πίνακας παρουσιάζει μια συνοπτική ποιοτική σύγκριση επιλεγμένων μοντέλων εποπτευόμενης μηχανικής μάθησης. Η σύγκριση πραγματοποιείται ως προς την ερμηνευσιμότητα, την ικανότητα αναπαράστασης μη γραμμικών σχέσεων, τις υπολογιστικές απαιτήσεις και τους βασικούς περιορισμούς κάθε μοντέλου.

2.4 Μη εποπτευόμενη μηχανική μάθηση για την κυβερνοασφάλεια

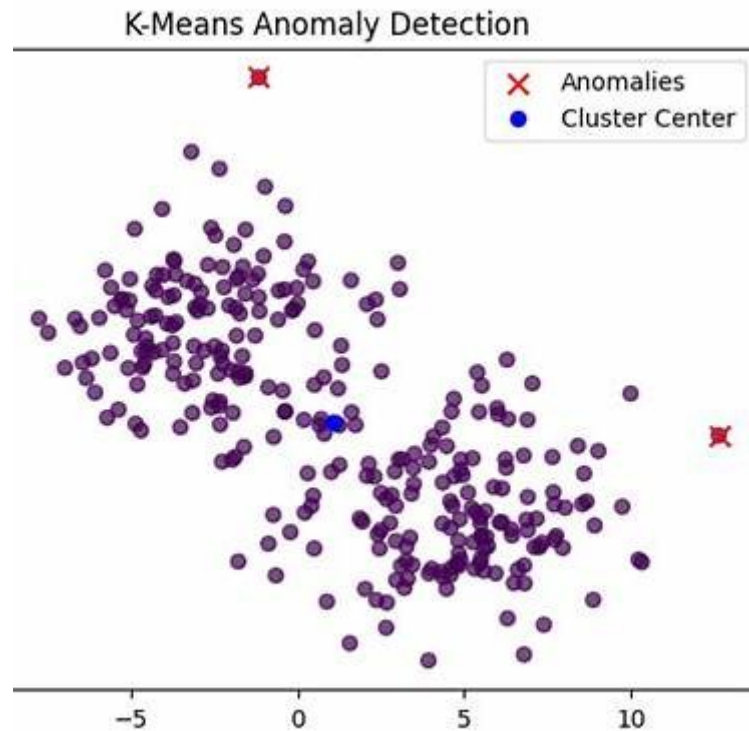
2.4.1 Μη εποπτευόμενη μηχανική μάθηση

Σε αντίθεση με την εποπτευόμενη μάθηση, η μη εποπτευόμενη μηχανική μάθηση δεν απαιτεί επισημασμένα δεδομένα εκπαίδευσης. Τα μοντέλα αυτά αναζητούν μοτίβα, ομάδες ή αποκλίσεις στα δεδομένα δικτυακής κίνησης χωρίς να γνωρίζουν εκ των προτέρων ποιες εγγραφές αντιστοιχούν σε καλοήγη ή κακόβουλη δραστηριότητα. Για τον λόγο αυτό, μπορούν να αξιοποιηθούν για την ανίχνευση άγνωστων επιθέσεων [17].

2.4.2 Μοντέλα μη εποπτευόμενης μηχανικής μάθησης

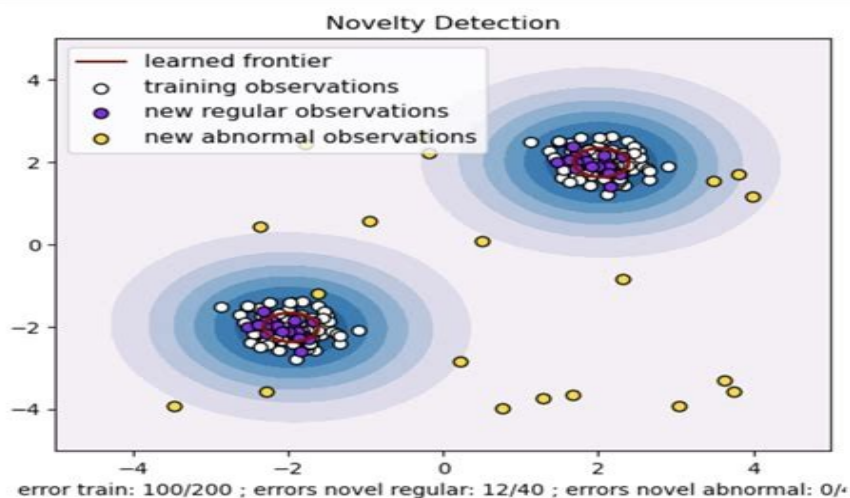
Αρκετοί αλγόριθμοι μη εποπτευόμενης μηχανικής μάθησης έχουν αποδειχθεί αποτελεσματικοί όπως:

- K-Means Clustering: Ο K-Means είναι ένας μη εποπτευόμενος αλγόριθμος ομαδοποίησης που χωρίζει τα δεδομένα σε έναν προκαθορισμένο αριθμό ομάδων (K). Στα συστήματα ανίχνευσης εισβολών, ο K-Means μπορεί να εφαρμοστεί σε μη επισημασμένα δεδομένα δικτυακής κίνησης, με στόχο τον διαχωρισμό διαφορετικών προτύπων δραστηριότητας και τον εντοπισμό πιθανών ανωμαλιών ή επιθέσεων. Βασικός περιορισμός του είναι ότι ο κατάλληλος αριθμός ομάδων πρέπει να επιλεγεί πριν από την εφαρμογή του αλγορίθμου, κάτι που μπορεί να είναι δύσκολο σε μεγάλα σύνολα δεδομένων [18].



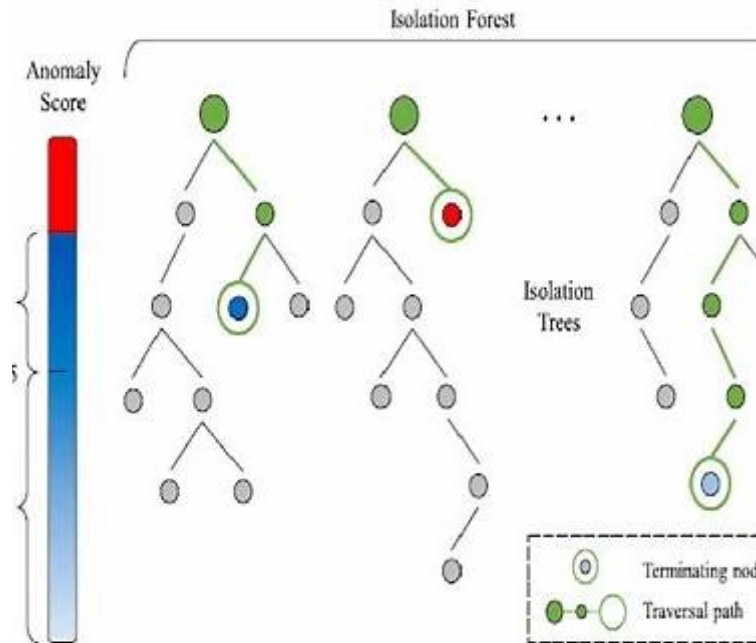
Εικόνα 2.9 Αλγόριθμος K-Means [51]

- One-Class Support Vector Machines (One-Class SVM): Το One-Class SVM είναι ένας αλγόριθμος ανίχνευσης ανωμαλιών που εκπαιδεύεται με δεδομένα καλοήθους δραστηριότητας. Αντί να διακρίνει μεταξύ πολλαπλών κλάσεων, μαθαίνει ένα όριο αποφάσεων γύρω από την κατανομή των κανονικών δεδομένων. Κατά την εφαρμογή του μοντέλου, εγγραφές που βρίσκονται εκτός του ορίου αυτού μπορούν να χαρακτηριστούν ως πιθανές ανωμαλίες. Η προσέγγιση αυτή είναι χρήσιμη όταν τα δεδομένα κακόβουλης δραστηριότητας είναι περιορισμένα ή δεν είναι διαθέσιμα κατά την εκπαίδευση [19].



Εικόνα 2.10 One-Class Support Vector Machines [52]

- **Isolation Forest:** Ο Isolation Forest είναι ένας αλγόριθμος ανίχνευσης ανωμαλιών που βασίζεται στην τυχαία κατάτμηση των δεδομένων. Δημιουργεί ένα σύνολο από δέντρα απομόνωσης, στα οποία επιλέγονται τυχαία χαρακτηριστικά και τιμές διαχωρισμού. Οι ανωμαλίες τείνουν να είναι σπάνιες και να διαφέρουν από τα κανονικά δεδομένα, επομένως απομονώνονται συνήθως με λιγότερους διαχωρισμούς και έχουν μικρότερο μέσο μήκος διαδρομής στα δέντρα. Στα IDS, ο Isolation Forest μπορεί να εφαρμοστεί σε δεδομένα δικτυακής κίνησης για τον εντοπισμό πιθανών ανωμαλιών [19].



Εικόνα 2.11 Isolation Forest [53]

2.4.3 Δυνατά σημεία και περιορισμοί

Τα μοντέλα μη εποπτευόμενης μηχανικής μάθησης μπορούν να υποστηρίξουν την ανίχνευση άγνωστων επιθέσεων, συμπεριλαμβανομένων επιθέσεων μηδενικής ημέρας, μέσω του εντοπισμού αποκλίσεων από τη συνήθη δικτυακή δραστηριότητα. Παρόλα αυτά μπορεί να παρουσιάσουν υψηλά ποσοστά ψευδώς θετικών συναγερμών, καθώς η καλοήγης δικτυακή κίνηση που διαφέρει από τη συμπεριφορά που έχει μάθει το μοντέλο ενδέχεται να χαρακτηριστεί ως ανωμαλία [8]. Παρά τους περιορισμούς αυτούς, η μη εποπτευόμενη μηχανική μάθηση αποτελεί χρήσιμη προσέγγιση στα IDS για την αναγνώριση άγνωστων προτύπων δραστηριότητας.

2.4.4 Σύγκριση αλγορίθμων μη εποπτευόμενης μηχανικής μάθησης για την κυβερνοασφάλεια

Μοντέλο	Ερμηνευσιμότητα	Ικανότητα μοντελοποίησης σύνθετων προτύπων	Υπολογιστικές απαιτήσεις και κλιμάκωση	Βασικός περιορισμός
K-mens Clustering	Μέτρια, καθώς οι	Περιορισμένη	Σχετικά χαμηλό	Απαιτεί εκ των

[2], [17], [18]	συστάδες και τα κέντρα τους μπορούν να εξεταστούν, αλλά η αντιστοίχιση τους σε καλοήγη ή κακόβουλη δραστηριότητα δεν είναι πάντα άμεση.	όταν οι συστάδες έχουν ακανόνιστο σχήμα ή επικαλύπτονται.	υπολογιστικό κόστος και καλή εφαρμογή σε μεγάλα σύνολα δεδομένων.	προτέρων καθορισμό του αριθμού των συστάδων και επηρεάζεται από την κλίμακα των χαρακτηριστικών και την αρχικοποίηση.
One-Class SVM [19], [24], [40]	Χαμηλή, ιδιαίτερα όταν χρησιμοποιούνται μη γραμμικοί πυρήνες, επειδή το όριο απόφασης δεν είναι εύκολο να ερμηνευτεί.	Υψηλή με τη χρήση κατάλληλων συναρτήσεων πυρήνα.	Η εκπαίδευση μπορεί να είναι υπολογιστικά απαιτητική σε μεγάλα σύνολα δεδομένων.	Η απόδοση του εξαρτάται από την επιλογή πυρήνα, την κανονικοποίηση και τη ρύθμιση των υπερπαραμέτρων.
Isolation Forest [2], [19], [41]	Μέτρια, καθώς μπορεί να εξεταστεί ο τρόπος απομόνωσης των παρατηρήσεων και η συμβολή των χαρακτηριστικών, αλλά η συνολική απόφαση δεν είναι πάντοτε άμεσα διαφανής.	Μπορεί να εντοπίσει ανωμαλίες σε πολυδιάστατα δεδομένα χωρίς συγκεκριμένη υπόθεση για την κατανομή τους.	Παρουσιάζει καλή κλιμάκωση και είναι κατάλληλο για μεγάλα σύνολα δεδομένων.	Η απόδοσή του μπορεί να μειωθεί όταν οι ανωμαλίες μοιάζουν με την καλοήγη δραστηριότητα ή όταν τα χαρακτηριστικά περιέχουν θόρυβο.

Ο παραπάνω πίνακας παρουσιάζει μια συνοπτική ποιοτική σύγκριση των μη εποπτευόμενων μοντέλων που χρησιμοποιούνται στην ανίχνευση ανωμαλιών. Η σύγκριση πραγματοποιείται ως προς την ερμηνευσιμότητα, την ικανότητα μοντελοποίησης σύνθετων προτύπων, τις υπολογιστικές απαιτήσεις και τους κύριους περιορισμούς κάθε αλγορίθμου.

2.5 Προκλήσεις στα IDS που βασίζονται σε ΑΙ

Το ΑΙ μπορεί να βελτιώσει σημαντικά την απόδοση των IDS. Ωστόσο, προκύπτουν σημαντικές προκλήσεις όταν μοντέλα που αναπτύσσονται και αξιολογούνται σε ερευνητικά περιβάλλοντα εφαρμόζονται σε πραγματικά δίκτυα.

1. Ποιότητα και Διαθεσιμότητα Δεδομένων: Τα μοντέλα εποπτευόμενης μηχανικής μάθησης απαιτούν σύνολα δεδομένων με επισημασμένες εγγραφές καλοηθής και κακόβουλης δραστηριότητας. Ωστόσο, η διαθεσιμότητα πραγματικών δεδομένων δικτυακής κίνησης είναι περιορισμένη, κυρίως λόγω ζητημάτων ιδιωτικότητας και ασφάλειας. Για τον λόγο αυτό, χρησιμοποιούνται συχνά δημόσια σύνολα δεδομένων, όπως τα NSL-KDD και CICIDS2017, για την εκπαίδευση και αξιολόγηση των μοντέλων. Παρ' όλα αυτά, κάθε σύνολο δεδομένων

έχει συγκεκριμένο τρόπο συλλογής, χαρακτηριστικά κίνησης και σενάρια επιθέσεων. Επομένως, η αξιολόγηση ενός μοντέλου πρέπει να λαμβάνει υπόψη κατά πόσο το σύνολο δεδομένων είναι κατάλληλο για το περιβάλλον όπου πρόκειται να εφαρμοστεί [10], [6].

2. Μη Ισορροπημένα Δεδομένα: Σε σύνολα δεδομένων για συστήματα ανίχνευσης εισβολών, οι εγγραφές καλοήθους δραστηριότητας συχνά υπερτερούν αριθμητικά των εγγραφών επιθέσεων. Η ανισορροπία αυτή μπορεί να οδηγήσει ένα μοντέλο να ευνοεί την πλειοψηφική κλάση και να μειώσει την ικανότητά του να εντοπίζει σπάνιες επιθέσεις. Για τον λόγο αυτό, η αξιολόγηση τέτοιων μοντέλων δεν πρέπει να βασίζεται μόνο στην ακρίβεια, αλλά και σε μετρικές όπως η ανάκληση, το FPR και το F1-score [3].
3. Μεταβολή της Δικτυακής Κίνησης: Η δικτυακή κίνηση και οι τεχνικές επιθέσεων μεταβάλλονται με την πάροδο του χρόνου, με αποτέλεσμα να αλλάζει και η κατανομή των δεδομένων που παρατηρεί ένα IDS. Επομένως, ένα μοντέλο που έχει εκπαιδευτεί σε παλαιότερα δεδομένα ενδέχεται να μην αποδίδει το ίδιο αποτελεσματικά σε νεότερη δικτυακή κίνηση και να χρειάζεται επανεκπαίδευση ή προσαρμογή σε νέα δεδομένα [11].
4. Επιθέσεις σε μοντέλα τεχνητής νοημοσύνης: Τα συστήματα πρόληψης εισβολών που βασίζονται σε τεχνικές μηχανικής μάθησης μπορεί να αποτελέσουν στόχο επιθέσεων αποφυγής ανίχνευσης. Σε τέτοιες επιθέσεις, ο επιτιθέμενος μπορεί να προσαρμόσει τα χαρακτηριστικά της κακόβουλης δικτυακής δραστηριότητας ώστε να μοιάζουν περισσότερο με καλοήθη κίνηση και να παραπλανήσει το μοντέλο [20]. Η προστασία μοντέλων μηχανικής μάθησης από τέτοιες τεχνικές αποτελεί σημαντική πρόκληση.
5. Περιορισμοί υπολογιστικών πόρων και κλιμάκωσης: Ορισμένα μοντέλα μηχανικής μάθησης απαιτούν σημαντικούς υπολογιστικούς πόρους τόσο κατά την εκπαίδευσή τους όσο και κατά τη λειτουργία τους σε πραγματικό χρόνο. Σε δίκτυα υψηλής ταχύτητας, οι απαιτήσεις αυτές δημιουργούν προκλήσεις ως προς την έγκαιρη ανάλυση της δικτυακής κίνησης και την κλιμάκωση του συστήματος [8]. Επιπλέον, η εφαρμογή IDS που βασίζονται σε τεχνητή νοημοσύνη σε συσκευές περιορισμένων πόρων, όπως οι συσκευές IoT, είναι δύσκολη λόγω περιορισμών σε επεξεργαστική ισχύ, μνήμη και κατανάλωση ενέργειας [21].
6. Ερμηνευσιμότητα και εμπιστοσύνη: Για την αποτελεσματική αξιοποίηση συστημάτων ανίχνευσης εισβολών, οι διαχειριστές πρέπει να μπορούν να αξιολογούν τους συναγερμούς που παράγουν. Σύνθετα μοντέλα βαθιάς μάθησης συχνά λειτουργούν ως «μαύρα κουτιά», καθώς η διαδικασία λήψης αποφάσεων δεν είναι εύκολο να εξηγηθεί. Η περιορισμένη ερμηνευσιμότητα μπορεί να δυσκολεύει την ανθρώπινη αξιολόγηση και την εμπιστοσύνη στις προβλέψεις του μοντέλου [22].

Οι παραπάνω προκλήσεις δείχνουν ότι η αποτελεσματική εφαρμογή IDS που βασίζονται σε τεχνητή νοημοσύνη δεν εξαρτάται μόνο από την επιλογή του κατάλληλου αλγορίθμου. Απαιτεί επίσης αξιόπιστα δεδομένα, συνεχή προσαρμογή των μοντέλων στις νέες συνθήκες και προσεκτική αξιολόγηση πριν από την εφαρμογή τους σε πραγματικά δίκτυα.

2.6 Επίλογος

Τα IDS μπορούν να επωφεληθούν από τη χρήση της τεχνητής νοημοσύνης. Τα παραδοσιακά IDS αντιμετωπίζουν περιορισμούς απέναντι στην αυξανόμενη κίνηση των σύγχρονων δικτύων. Η τεχνητή νοημοσύνη, μέσω της μηχανικής μάθησης, μπορεί να βοηθήσει τα IDS στην ανίχνευση γνωστών επιθέσεων με τη χρήση της εποπτευόμενης μηχανικής μάθησης και να υποστηρίξει τον εντοπισμό

επιθέσεων μηδενικής ημέρας με τη χρήση της μη εποπτευόμενης μηχανικής μάθησης. Η ενσωμάτωση της τεχνητής νοημοσύνης στα IDS παρουσιάζει αρκετές προκλήσεις, ιδιαίτερα όταν εφαρμόζεται σε περιβάλλοντα πραγματικού χρόνου και όχι μόνο σε ερευνητικό πλαίσιο. Τα μη ισορροπημένα δεδομένα, η ποιότητα και η διαθεσιμότητα των δεδομένων, καθώς και αρκετές άλλες προκλήσεις, κάνουν αυτή την ενσωμάτωση δύσκολη. Για τη χρήση AI-based IDS πρέπει να διασφαλίζεται η αξιοπιστία των επιλεγμένων μοντέλων μέσω κατάλληλων μέτρων αξιολόγησης, ώστε να ωφεληθεί σημαντικά ο τομέας της κυβερνοασφάλειας.

Κεφάλαιο 3ο: Επιθέσεις μηδενικής ημέρας

3.1 Εισαγωγή

Οι επιθέσεις μηδενικής ημέρας παρουσιάζουν ένα από τα μεγαλύτερα προβλήματα στην σύγχρονη κυβερνοασφάλεια. Ο όρος zero-day(μηδενική ημέρα) προέρχεται από το γεγονός ότι οι διαχειριστές των συστημάτων έχουν μηδέν μέρες να προετοιμαστούν για την εκμετάλλευση της αδυναμίας στο σύστημα, από την στιγμή που δεν είναι γνωστή στους ειδικούς ασφαλείας [23]. Οι επιθέσεις αυτές εκμεταλλεύονται άγνωστες αδυναμίες στο λογισμικό, υλικό ή στα πρωτόκολλα δικτύου επιτρέποντας ορισμένες φορές την παράκαμψη συμβατικών αμυνών όπως το τείχος προστασίας, IDS που βασίζονται σε υπογραφές και antivirus.

Οι επιθέσεις μηδενικής ημέρας είναι ιδιαίτερα επικίνδυνες, επειδή εκμεταλλεύονται ευπάθειες που δεν έχουν αναγνωριστεί ακόμα. Εξαιτίας της απουσίας διαθέσιμων υπογραφών ανίχνευσης, τα συστήματα ασφαλείας που βασίζονται σε γνωστά πρότυπα επιθέσεων ενδέχεται να μην μπορούν να τις εντοπίσουν έγκαιρα [24]. Οι επιτιθέμενοι αξιοποιούν αυτές τις ευπάθειες για να αποκτήσουν πρόσβαση στο σύστημα, να κλέψουν δεδομένα ή να διακόψουν κρίσιμες υπηρεσίες. Ένα παράδειγμα όπως το Stuxnet τονίζει το πόσο καταστροφικές είναι αυτού του είδους οι επιθέσεις [25].

Οι παραδοσιακοί μηχανισμοί ανίχνευσης βασίζονται κυρίως σε γνωστές υπογραφές επιθέσεων πράγμα το οποίο οδηγεί σε περιορισμούς στην ανίχνευση άγνωστων επιθέσεων. Η τεχνητή νοημοσύνη και η μηχανική μάθηση έχουν την δυνατότητα να αναγνωρίσουν ανώμαλα μοτίβα που μπορεί να υποδεικνύουν επιθέσεις μηδενικής ημέρας. Συγκεκριμένα, μοντέλα ανίχνευσης ανωμαλιών μπορούν να εκπαιδευτούν με δεδομένα δικτυακής κίνησης ώστε να εντοπίζουν αποκλίσεις από τη συνήθη συμπεριφορά [8].

Λόγω της απροβλεψιμότητας και του κινδύνου που παρουσιάζουν οι επιθέσεις μηδενικής ημέρας, αποτελούν κρίσιμο πεδίο έρευνας τόσο για τον ακαδημαϊκό χώρο όσο και για τη βιομηχανία. Αυτή η ενότητα απαριθμεί τα χαρακτηριστικά τους και τους τρόπους ανίχνευσης, και στη συνέχεια σχολιάζει τις προκλήσεις και κάποια παραδείγματα επιθέσεων.

3.2 Χαρακτηριστικά Επιθέσεων Μηδενικής Ημέρας

Η διαφορά των επιθέσεων Μηδενικής Ημέρας από τις υπόλοιπες είναι ότι εκμεταλλεύονται ευπάθειες του συστήματος που δεν είναι γνωστές στους ειδικούς ασφαλείας τη στιγμή της επίθεσης[23]. Αυτή η ιδιαιτερότητα κάνει τις επιθέσεις σχεδόν αόρατες στα συστήματα ανίχνευσης. Τα χαρακτηριστικά που ορίζουν μια επίθεση μηδενικής ημέρας είναι:

1. Απουσία προηγούμενης γνώσης και απρόβλεπτη φύση: Οι επιθέσεις μηδενικής ημέρας στοχεύουν σε ευπάθειες του συστήματος που δεν έχουν καταγραφεί ακόμα. Με αποτέλεσμα οι ειδικοί της κυβερνοασφάλειας να μην διαθέτουν προηγούμενη γνώση, κατάλληλες υπογραφές ή προκαθορισμένα μοτίβα για την έγκαιρη αναγνώριση τους [24]. Η απρόβλεπτη φύση αυτών των επιθέσεων τις καθιστά ως μία από τις πιο επικίνδυνες.
2. Διακριτικότητα: Οι επιτιθέμενοι μπορούν να χρησιμοποιούν πολυμορφικές τεχνικές, μεταβάλλοντας τη μορφή ή τη συμπεριφορά της επίθεσης ώστε να δυσχεραίνουν την ανίχνευση της από συστήματα ανίχνευσης εισβολών. Σε ορισμένες περιπτώσεις, οι παραλλαγές αυτές αποκρύπτουν κρυπτογραφημένο κακόβουλο κώδικα και μπορούν να παρακάμπτουν μηχανισμούς ανίχνευσης που βασίζονται σε υπογραφές [26].

3. Ελάχιστος χρόνος απόκρισης: Όταν μια ευπάθεια γίνει γνωστή, τα επηρεαζόμενα συστήματα παραμένουν εκτεθειμένα μέχρι να εφαρμοστεί η κατάλληλη ενημέρωση ασφαλείας (patch). Η καθυστέρηση στην εγκατάσταση των ενημερώσεων ασφαλείας αυξάνει το χρονικό διάστημα κατά το οποίο οι επιτιθέμενοι μπορούν να εκμεταλλευτούν την ευπάθεια [27].
4. Στοχευμένη φύση και υψηλό αντίκτυπο: Ορισμένες επιθέσεις μηδενικής ημέρας χρησιμοποιούνται στοχευμένα κατά οργανισμών ή κρίσιμων υποδομών, με σκοπό τη συλλογή πληροφοριών, την κατασκοπεία ή τη διατάραξη λειτουργιών [28]. Το Stuxnet αποτελεί χαρακτηριστικό παράδειγμα στοχευμένης επίθεσης κατά βιομηχανικών συστημάτων ελέγχου [25].
5. Κύκλος ζωής ενός Zero-Day Exploit: Ο κύκλος ζωής μιας επίθεσης μηδενικής ημέρας ξεκινά με τον εντοπισμό μιας ευπάθειας, με την ανάπτυξη της επίθεσης, την χρήση της και τελικά όταν ανιχνευτεί ακολουθεί η διόρθωση της ευπάθειας. Στο χρονικό περιθώριο μεταξύ του εντοπισμού της ευπάθειας και την διόρθωση της οι επιτιθέμενοι έχουν μεγάλο πλεονέκτημα έναντι των ειδικών ασφαλείας[23].

3.3 Ταξινόμηση και τύποι επιθέσεων μηδενικής ημέρας

Οι επιθέσεις μηδενικής ημέρας μπορούν να ταξινομηθούν σε διάφορες κατηγορίες ανάλογα με την ευπάθεια που εκμεταλλεύονται, τον στόχο τους και τον φορέα επίθεσης. Παρακάτω παρουσιάζονται οι βασικές κατηγορίες επιθέσεων σύμφωνα με αυτά τα κριτήρια.

3.3.1 Κατηγοριοποίηση με βάση τον στόχο

- Λειτουργικά συστήματα: Επιθέσεις που στοχεύουν ευπάθειες σε επίπεδο πυρήνα που οδηγούν στην απόκτηση αυξημένων δικαιωμάτων στο σύστημα.
- Εφαρμογές: Επιθέσεις που στοχεύουν προγράμματα περιήγησης, λογισμικό ανάγνωσης εγγράφων ή ευρέως χρησιμοποιούμενο εταιρικό λογισμικό.
- Υπηρεσίες δικτύου: Επιθέσεις ενάντια σε διακομιστές ιστού, υπηρεσίες DNS ή πρωτόκολλα όπως τα SSL/TLS.
- IoT και Ενσωματωμένα Συστήματα: Επιθέσεις που στοχεύουν συσκευές IoT οι οποίες συχνά δεν διαθέτουν ισχυρούς μηχανισμούς ενημέρωσης λογισμικού.
- Συστήματα Βιομηχανικού Ελέγχου (ICS): Επιθέσεις που στοχεύουν συστήματα SCADA και άλλα συστήματα βιομηχανικού ελέγχου.

3.3.2 Κατηγοριοποίηση με βάση τη μέθοδο επίθεσης

- Υπερχείλιση προσωρινής μνήμης και παραβίαση μνήμης: Κλασικές τεχνικές εκμετάλλευσης που μπορούν να οδηγήσουν σε εκτέλεση κώδικα.
- Επιθέσεις εισαγωγής δεδομένων: SQL injection, εισαγωγή εντολών ή σεναρίων εναντίον διαδικτυακών εφαρμογών.
- Λογικά ελαττώματα: Εκμετάλλευση σφαλμάτων σχεδιασμού σε μηχανισμούς αυθεντικοποίησης, διαχείρισης συνεδριών ή ελέγχου πρόσβασης.

- Κλιμάκωση δικαιωμάτων: Χρήση ευπαθειών για την απόκτηση υψηλότερων επιπέδων πρόσβασης στο σύστημα.
- Απομακρυσμένη Εκτέλεση Κώδικα (RCE): Εκμετάλλευση ευπαθειών που επιτρέπουν στους εισβολείς να εκτελούν κώδικα σε απομακρυσμένο σύστημα.

3.3.3 Κατηγοριοποίηση με βάση τον σκοπό της επίθεσης

- Κατασκοπεία: Εξαγωγή δεδομένων και παρακολούθηση.
- Σαμποτάζ: Διαταραχή συστημάτων, συχνά σε κρίσιμους τομείς (π.χ. Stuxnet).
- Οικονομικό κέρδος: Κλοπή διαπιστευτηρίων, χρήση ransomware (λογισμικό εκβιασμού) και διαδικτυακή απάτη.
- Χακτιβισμός: Επιθέσεις που υποκινούνται από ιδεολογία ή πολιτικά κίνητρα.

3.3.4 Παραδείγματα αξιοσημείωτων επιθέσεων Μηδενικής Ημέρας

- Stuxnet (2010): Στοχοποίησε ιρανικές πυρηνικές εγκαταστάσεις χρησιμοποιώντας πολλαπλές zero-day ευπάθειες στα Windows.
- Log4Shell (2021): Μια ευπάθεια Μηδενικής Ημέρας στη βιβλιοθήκη καταγραφής Log4j για Java, η οποία επέτρεπε απομακρυσμένη εκτέλεση κώδικα.

3.4 Τεχνικές ανίχνευσης για επιθέσεις μηδενικής ημέρας

Η ανίχνευση επιθέσεων μηδενικής ημέρας είναι ένα δύσκολο εγχείρημα από την στιγμή που δεν υπάρχει προηγούμενη γνώση της συγκεκριμένης ευπάθειας [23]. Με αποτέλεσμα οι παραδοσιακές τεχνικές ανίχνευσης να αδυνατούν να τις εντοπίσουν. Οι ερευνητές και οι ειδικοί ασφαλείας έχουν αναπτύξει τεχνικές για την αντιμετώπιση αυτού του προβλήματος, από μεθόδους εύρεσης έως και προηγμένη τεχνητή νοημοσύνη.

3.4.1 Ανίχνευση με βάση την υπογραφή

Παραδοσιακά συστήματα ανίχνευσης εισβολής, όπως τα Snort και Suricata, βασίζονται σε γνωστές υπογραφές κακόβουλης δραστηριότητας. Παρότι είναι αποτελεσματικά στην ανίχνευση γνωστών επιθέσεων, ενδέχεται να αδυνατούν να εντοπίσουν επιθέσεις μηδενικής ημέρας, καθώς δεν υπάρχουν διαθέσιμες υπογραφές για τις συγκεκριμένες απειλές [23].

3.4.2 Ανίχνευση με βάση τη συμπεριφορά

Αυτή η μέθοδος στοχεύει στην αναγνώριση ύποπτων μοτίβων ή ανωμαλιών σε ένα σύστημα ή δίκτυο [8]. Παρότι είναι χρήσιμη, μπορεί να απαιτεί σημαντικούς υπολογιστικούς πόρους, ιδιαίτερα σε περιβάλλοντα με περιορισμένες δυνατότητες [21].

3.4.3 Στατιστικές μέθοδοι και μέθοδοι ανίχνευσης ανωμαλιών

Τα στατιστικά μοντέλα καθορίζουν μια γραμμή βάσης για την καλοήγη δραστηριότητα ενός δικτύου και επισημαίνουν τις αποκλίσεις ως πιθανές επιθέσεις Μηδενικής Ημέρας. Τεχνικές όπως η ομαδοποίηση και η εκτίμηση πυκνότητας ανήκουν σε αυτήν την κατηγορία [17]. Το κύριο πλεονέκτημα τους είναι η ικανότητα να εντοπίζουν συμπεριφορές που δεν είχαν παρατηρηθεί προηγουμένως, παρόλα αυτά συχνά υποφέρουν από υψηλά ποσοστά ψευδώς θετικών αποτελεσμάτων σε δυναμικά περιβάλλοντα [29].

3.4.4 Μηχανική μάθηση και τεχνητή νοημοσύνης

Η Μηχανική Μάθηση είναι ένα ισχυρό εργαλείο για την ανίχνευση επιθέσεων Μηδενικής Ημέρας. Με την εκμάθηση γενικών μοτίβων από δεδομένα εκπαίδευσης, τα μοντέλα μπορούν να εντοπίζουν ανωμαλίες που ενδέχεται να υποδηλώνουν νέες επιθέσεις:

- Τα εποπτευόμενα μοντέλα (π.χ. Logistic Regression, Decision Trees, Random Forests, SVM και νευρωνικά δίκτυα) ταξινομούν τη δραστηριότητα με βάση μοτίβα που έχουν μάθει από επισημασμένα δεδομένα.
- Τα μη εποπτευόμενα μοντέλα (π.χ. One-Class SVM, Isolation Forest και K-means) μπορούν να εντοπίζουν άγνωστες ή ασυνήθιστες συμπεριφορές μέσω της ανίχνευσης ανωμαλιών.
- Οι υβριδικές προσεγγίσεις συνδυάζουν εποπτευόμενες και μη εποπτευόμενες τεχνικές για την ανίχνευση εισβολών [6].

3.4.5 Τεχνικές βασισμένες στη βαθιά μάθηση

Τα μοντέλα βαθιάς Μάθησης όπως τα συνελκτικά νευρωνικά δίκτυα (CNN) και τα επαναλαμβανόμενα νευρωνικά δίκτυα (RNN), χρησιμοποιούνται για την ανίχνευση εισβολών και την ανάλυση πολύπλοκων μοτίβων δικτυακής κίνησης [15]. Οι Autoencoders, ειδικότερα εφαρμόζονται για τη μοντελοποίηση της δικτυακής κυκλοφορίας και τον εντοπισμό ανωμαλιών μέσω σφάλματος ανακατασκευής [30].

3.4.6 Πληροφορίες απειλών και συνεργατική ανίχνευση

Πέρα από τις αλγοριθμικές προσεγγίσεις, η αξιοποίηση πληροφοριών για κυβερνοαπειλές και η ανταλλαγή σχετικών δεδομένων μεταξύ οργανισμών μπορούν να συμβάλουν στον ταχύτερο εντοπισμό και την αντιμετώπιση επιθέσεων μηδενικής ημέρας [31]. Η συνεργατική ανίχνευση μπορεί να βελτιώσει την ορατότητα των απειλών, αλλά δημιουργεί και ζητήματα απορρήτου και εμπιστοσύνης [32].

3.5 Προκλήσεις στην ανίχνευση επιθέσεων μηδενικής ημέρας

Έχει γίνει σημαντική πρόοδος στην ανάπτυξη μεθόδων ανίχνευσης επιθέσεων μηδενικής ημέρας, παρόλα αυτά εξακολουθούν να υπάρχουν προκλήσεις για την αποτελεσματική εφαρμογή τους.

1. Περιορισμένη διαθεσιμότητα επισημασμένων δεδομένων : Τα μοντέλα εποπτευόμενης μηχανικής μάθησης απαιτούν επισημασμένα παραδείγματα καλοήθους και κακόβουλης δικτυακής κίνησης. Ωστόσο, η συλλογή και η χειροκίνητη επισήμανση μεγάλων συνόλων δεδομένων είναι δαπανηρή και δύσκολη, ιδιαίτερα όταν πρόκειται για άγνωστες επιθέσεις μηδενικής ημέρας [33].
2. Ανισορροπία Δεδομένων: Τα σύνολα δεδομένων για την ανίχνευση εισβολών είναι συχνά ανισόρροπα, καθώς η καλοήθης δικτυακή κίνηση περιλαμβάνει πολύ περισσότερα παραδείγματα από την κακόβουλη δραστηριότητα. Οι επιθέσεις μηδενικής ημέρας είναι ιδιαίτερα δύσκολο να εκπροσωπηθούν σε τέτοια σύνολα δεδομένων, γεγονός που επηρεάζει αρνητικά την εκπαίδευση και την αξιολόγηση των μοντέλων [34].
3. Μετατόπιση εννοιών και εξελισσόμενες απειλές : Η δικτυακή κίνηση εξελίσσεται συνεχώς λόγω νέων εφαρμογών, αλλαγών στη συμπεριφορά των χρηστών και νέων τεχνικών επίθεσης. Επομένως τα μοντέλα που έχουν εκπαιδευτεί σε παλαιότερα δεδομένα ενδέχεται να παρουσιάζουν μειωμένη απόδοση σε νεότερα δεδομένα ή άγνωστες επιθέσεις. Το φαινόμενο αυτό είναι γνωστό ως μετατόπιση εννοιών [11].

4. Εχθρική Παράκαμψη Εντοπισμού : Οι επιτιθέμενοι μπορούν να προσαρμόζουν σκόπιμα τα μοτίβα δικτυακής κίνησης, ώστε να αποφεύγουν τον εντοπισμό από συστήματα ανίχνευσης εισβολών που αξιοποιούν μηχανική μάθηση [20]. Τέτοιες προσαρμογές μπορούν να επηρεάσουν την απόδοση των μοντέλων ανίχνευσης.
5. Περιορισμοί σε Πραγματικό Χρόνο: Τα δίκτυα υψηλής ταχύτητας παράγουν μεγάλο όγκο δικτυακής κίνησης, πράγμα που καθιστά απαιτητική την ανίχνευση σε πραγματικό χρόνο ως προς τους υπολογιστικούς πόρους [13].

3.6 Μελέτες περιπτώσεων επιθέσεων μηδενικής ημέρας

Για την καλύτερη κατανόηση του αντίκτυπου των επιθέσεων μηδενικής ημέρας είναι χρήσιμο να εξεταστούν αξιοσημείωτα περιστατικά από τον πραγματικό κόσμο.

3.6.1 Stuxnet (2010)

Το Stuxnet παραμένει μία από τις πιο διάσημες επιθέσεις μηδενικής ημέρας. Στοχοποίησε βιομηχανικά συστήματα ελέγχου (ICS) σε ιρανικές πυρηνικές εγκαταστάσεις αξιοποιώντας πολλαπλές ευπάθειες μηδενικής ημέρας στα Microsoft Windows [25]. Το Stuxnet ήταν ικανό να επαναπρογραμματίσει προγραμματιζόμενους λογικούς ελεγκτές (PLC) ενώ ταυτόχρονα παρήγαγε ψευδή δεδομένα στα συστήματα παρακολούθησης αποκρύπτοντας τη δραστηριότητα του. Αυτή η επίθεση έδειξε πως ορισμένες ευπάθειες μηδενικής ημέρας μπορούν να εκμεταλλευτούν για κυβερνοπόλεμο και βιομηχανική δολιοφθορά.

3.6.2 Log4Shell (2021)

Η Log4Shell (CVE-2021-44228) ήταν ευπάθεια μηδενικής ημέρας στη βιβλιοθήκη καταγραφής Apache Log4j για Java. Επέτρεψε απομακρυσμένη εκτέλεση κώδικα (RCE) μέσω κακόβουλα διαμορφωμένων μηνυμάτων καταγραφής. Η ευπάθεια ήταν αντικείμενο ενεργής εκμετάλλευσης από επιτιθέμενους, αναδεικνύοντας την ανάγκη για γρήγορη εφαρμογή ενημερώσεων ασφαλείας [35], [36].

3.6.3 Συμπεράσματα από το Stuxnet και το Log4Shell

- Οι ευπάθειες μηδενικής ημέρας μπορούν να χρησιμοποιηθούν σε στοχευμένες επιθέσεις και σε επιθέσεις ευρύτερης κλίμακας.
- Οι επιπτώσεις τέτοιων επιθέσεων δεν είναι μόνο τεχνικές ζημιές αλλά μπορεί να έχουν και γεωπολιτικές ή οικονομικές συνέπειες. Επίσης πολλές φορές οδηγούν σε δυσφήμιση του οργανισμού ή της εταιρείας.
- Η γρήγορη ανίχνευση και αντίδραση έχουν μεγάλη σημασία.

3.7 Επίλογος

Σε αυτό το κεφάλαιο εξετάστηκαν οι επιθέσεις μηδενικής ημέρας, ένα από τα πιο κρίσιμα προβλήματα της κυβερνοασφάλειας. Έγινε μια εισαγωγή στην έννοια των επιθέσεων και πιο συγκεκριμένα στα κύρια χαρακτηριστικά τους όπως η καινοτομία τους η δυσκολία ανίχνευσής τους και οι πιθανές επιπτώσεις τους όταν δεν εντοπίζονται έγκαιρα. Οι επιθέσεις κατηγοριοποιήθηκαν με βάση τον στόχο επίθεσης, τη μέθοδο επίθεσης και τον σκοπό του επιτιθέμενου.

Μετά αναφέρθηκαν οι τεχνικές ανίχνευσης από τα παραδοσιακά IDS που βασίζονται σε υπογραφές μέχρι IDS που ανιχνεύουν ανωμαλίες στην δικτυακή κίνηση και IDS που έχουν ως βάση το AI. Παρόλο

που η τεχνητή νοημοσύνη προσφέρει υποσχόμενα αποτελέσματα στην ανίχνευση επιθέσεων μηδενικής ημέρας, υπάρχουν προκλήσεις όπως ανισορροπία δεδομένων και περιορισμένη διαθεσιμότητα επισημασμένων δεδομένων.

Παραδείγματα πραγματικών επιθέσεων όπως το Stuxnet και η εκμετάλλευση της ευπάθειας Log4Shell παρουσιάζουν τις επιπτώσεις των επιθέσεων μηδενικής ημέρας σε διάφορα περιβάλλοντα, από συστήματα βιομηχανικού ελέγχου έως εταιρικά δίκτυα. Αυτά τα παραδείγματα ενισχύουν την ανάγκη για πιο αποτελεσματικούς και ανθεκτικούς μηχανισμούς ανίχνευσης εισβολών.

Τέλος οι επιθέσεις μηδενικής ημέρας τονίζουν τους περιορισμούς των παραδοσιακών μηχανισμών ασφαλείας και την ανάγκη για την ενσωμάτωση της μηχανικής μάθησης και της ανίχνευσης ανωμαλιών στα IDS. Στο τέταρτο κεφάλαιο αναλύονται τα μοντέλα μηχανικής μάθησης που επιλέχθηκαν για το πειραματικό κομμάτι της Δ.Ε..

Κεφάλαιο 4ο: Επιλεγμένα μοντέλα μηχανικής μάθησης

4.1 Εισαγωγή

Τα Μοντέλα Μηχανικής Μάθησης βρίσκονται στο επίκεντρο της προσοχής καθώς παρέχουν μια εναλλακτική και ευέλικτη λύση για τον εντοπισμό κακόβουλης δικτυακής κίνησης. Σε αντίθεση με τα συστήματα που βασίζονται σε κανόνες, τα μοντέλα μηχανικής μάθησης μπορούν να μάθουν μοτίβα απευθείας από τα δεδομένα και να κάνουν προβλέψεις για περιπτώσεις που δεν έχουν παρατηρηθεί στο παρελθόν. Αυτό είναι ιδιαίτερα σημαντικό στην κυβερνοασφάλεια, όπου οι μέθοδοι επιθέσεων αλλάζουν συνεχώς.

Ο σχεδιασμός ενός IDS είναι ένας τομέας που απαιτεί την επιλογή κατάλληλων μοντέλων μηχανικής μάθησης. Στην παρούσα Δ.Ε. επιλέχθηκαν έξι μοντέλα μηχανικής μάθησης.

Τα μοντέλα εποπτευόμενης μηχανικής μάθησης που επιλέχθηκαν είναι η Logistic Regression, το Random Forest και τα Support Vector Machines (SVM). Εκπαιδεύτηκαν σε επισημασμένα δεδομένα, ώστε να μάθουν ένα όριο απόφασης που διαχωρίζει την καλοήγη από την κακόβουλη δικτυακή κίνηση.

Τα μοντέλα μη εποπτευόμενης μηχανικής μάθησης που επιλέχθηκαν είναι ο K-Means Clustering, ο Isolation Forest και το One-Class SVM. Δεν απαιτούν επισημασμένα δεδομένα επιθέσεων, αλλά αναζητούν ανωμαλίες στη δικτυακή κίνηση.

Στο υπόλοιπο του κεφαλαίου παρουσιάζονται η θεωρητική βάση, ο τρόπος λειτουργίας, καθώς και τα πλεονεκτήματα και τα μειονεκτήματα των επιλεγμένων μοντέλων στο πλαίσιο της ανίχνευσης επιθέσεων.

4.2 Logistic Regression

Η Logistic Regression είναι ένα απλό αλλά αποτελεσματικό μοντέλο που χρησιμοποιείται συχνά ως αφετηρία στην έρευνα για την ανίχνευση εισβολών.

Σε αντίθεση με την Linear Regression, η οποία χρησιμοποιείται για την πρόβλεψη συνεχών τιμών, η Logistic Regression εκτιμά την πιθανότητα μια συγκεκριμένη είσοδος να ανήκει σε μια κατηγορία. Όταν εφαρμόζεται στην ανίχνευση εισβολών, το μοντέλο εκτιμά την πιθανότητα μια ροή δικτύου να αντιστοιχεί σε καλοήγη ή κακόβουλη δραστηριότητα.

Τα πλεονεκτήματα της Logistic Regression είναι τα παρακάτω :

- Παρέχει πιθανοτικές προβλέψεις για τις κλάσεις.
- Ερμηνευσιμότητα των βαρών των χαρακτηριστικών
- Κατάλληλο ως βασικό μοντέλο σύγκρισης

Τα μειονεκτήματα της Logistic Regression είναι τα παρακάτω :

- Σε μη γραμμικά μοτίβα ενδέχεται να έχει χαμηλή απόδοση
- Η απόδοση μπορεί να μειωθεί όταν υπάρχουν πολύπλοκες συμπεριφορές επίθεσης [37].

Για αυτόν τον λόγο, το Logistic regression συγκρίνεται συχνά με πιο προηγμένα μη γραμμικά μοντέλα, όπως το Random Forest.

4.3 Random Forest

Το Random Forest είναι ένα μοντέλο μηχανικής μάθησης που στοχεύει στη βελτίωση της ακρίβειας και της δυνατότητας γενίκευσης σε σχέση με τα Decision Trees. Λειτουργεί δημιουργώντας πολλαπλά δέντρα αποφάσεων κατά τη φάση της εκπαίδευσης και παράγει την τελική ταξινόμηση μέσω πλειοψηφικής ψηφοφορίας.

Ο αλγόριθμος βασίζεται στα ακόλουθα βήματα:

1. Δημιουργεί πολλαπλά δείγματα bootstrap από το αρχικό σύνολο δεδομένων.
2. Εκπαιδεύει ένα Decision Tree σε κάθε δείγμα bootstrap.
3. Σε κάθε διαχωρισμό ενός δέντρου, επιλέγει τυχαία ένα υποσύνολο χαρακτηριστικών αντί να λαμβάνει υπόψη όλα τα χαρακτηριστικά.
4. Ψηφοφορία πλειοψηφίας των συγκεντρωτικών προβλέψεων από όλα τα δέντρα.

Η τυχαιότητα τόσο της δειγματοληψίας δεδομένων όσο και της επιλογής χαρακτηριστικών μειώνει τη συσχέτιση μεταξύ των δέντρων και βελτιώνει την απόδοση της γενίκευσης.

Τα πλεονεκτήματα του Random Forest είναι ότι συμβάλει στη μείωση της διακύμανσης και της υπερπροσαρμογής σε σχέση με ένα μεμονωμένο Decision Tree. Επιπλέον μπορεί να μοντελοποιεί σύνθετες και μη γραμμικές σχέσεις μεταξύ χαρακτηριστικών. Ενώ το μειονέκτημα του είναι ότι είναι λιγότερο ερμηνεύσιμο από ένα μεμονωμένο Decision Tree [38], [39].

4.4 Support Vector Machine

Το SVM είναι αλγόριθμοι εποπτευόμενης μηχανικής μάθησης που χρησιμοποιούνται σε προβλήματα ταξινόμησης και παλινδρόμησης. Στην πιο απλή μορφή του, προσπαθεί να εντοπίσει ένα υπερεπίπεδο διαχωρισμού μεταξύ δύο κλάσεων, με τέτοιο τρόπο ώστε να μεγιστοποιείται η απόσταση μεταξύ των κοντινότερων σημείων και του υπερεπιπέδου. Τα σημεία αυτά ονομάζονται διανύσματα υποστήριξης και καθορίζουν τη θέση του υπερεπιπέδου απόφασης.

Παρόλα αυτά τα δεδομένα δεν μπορούν πάντα να διαχωριστούν γραμμικά. Για αυτό το SVM χρησιμοποιεί συναρτήσεις πυρήνα οι οποίες επιτρέπουν την δημιουργία μη γραμμικών ορίων απόφασης. Οι πιο συνηθισμένες συναρτήσεις πυρήνα είναι ο Γραμμικός, ο πολυωνυμικός, ο RBF και ο σιγμοειδής.

Η απόδοσή του μοντέλου εξαρτάται και από την επιλογή πυρήνα η οποία καθορίζεται από τα χαρακτηριστικά των δεδομένων και από την κατάλληλη ρύθμιση των παραμέτρων του μοντέλου. Ο RBF χρησιμοποιείται συχνά σε εφαρμογές SVM και έχει χρησιμοποιηθεί επίσης και στην ανίχνευση εισβολών.

Το πλεονέκτημα του SVM είναι ότι μπορεί να χειριστεί δεδομένα τα οποία δεν διαχωρίζονται γραμμικά μέσω των συναρτήσεων πυρήνα που δημιουργούν μη γραμμικά όρια απόφασης. Ταυτόχρονα υπάρχει η επιλογή διαφορετικών πυρήνων που επιτρέπουν στο μοντέλο να προσαρμοστεί καλύτερα στα χαρακτηριστικά του συνόλου δεδομένων.

Το μειονέκτημα του SVM είναι ότι σε μεγάλα σύνολα δεδομένων η εκπαίδευση του είναι υπολογιστικά απαιτητική ιδιαίτερα όταν χρησιμοποιούνται μη γραμμικοί πυρήνες. Επίσης σε ανισόρροπα σύνολα δεδομένων η απόδοση του μπορεί να επηρεαστεί αρνητικά, ιδιαίτερα όταν η μειοψηφική κλάση εκπροσωπείται από πολύ λιγότερα δείγματα [40].

4.5 K-Means Ομαδοποίηση

Ένας από τους πιο δημοφιλείς αλγόριθμους μη εποπτευόμενης μάθησης για την εκτέλεση ανάλυσης ομαδοποίησης, είναι ο K-Means. Δεν απαιτεί επισημασμένα δεδομένα κατά την εκπαίδευση αλλά χωρίζει τις εγγραφές με βάση την ομοιότητα τους σε συστάδες. Κάθε εγγραφή αντιστοιχίζεται στη συστάδα με το πλησιέστερο κέντρο, δηλαδή το κεντροειδές της συστάδας,

Ο αλγόριθμος ακολουθεί τα παρακάτω βήματα:

1. Επιλέγεται ο αριθμός των συστάδων πριν την εκπαίδευση.
2. Ο αλγόριθμος θέτει αρχικά κεντροειδή σημεία.
3. Κάθε σημείο δεδομένων αντιστοιχίζεται στο πλησιέστερο κεντροειδές με βάση την ευκλείδεια απόσταση.
4. Τα κεντροειδή υπολογίζονται ξανά ως ο μέσος όρος των σημείων κάθε συστάδας.
5. Η διαδικασία επαναλαμβάνεται μέχρι τα κεντροειδή να σταθεροποιηθούν.

Στην ανίχνευση εισβολών ο K-Means μπορεί να χρησιμοποιηθεί για την ομαδοποίηση μη επισημασμένων δεδομένων δικτυακής κίνησης και τον εντοπισμό διαφορετικών προτύπων δραστηριότητας. Μετά το τέλος της εκπαίδευσης οι συστάδες αξιολογούνται με βάση τις διαθέσιμες ετικέτες ως συστάδες κυρίως καλοήθους ή κακόβουλης δραστηριότητας.

Βασικό πλεονέκτημα του K-means είναι ότι δεν απαιτεί επισημασμένα δεδομένα και παρουσιάζει σχετικά χαμηλή υπολογιστική πολυπλοκότητα. Επομένως μπορεί να αξιοποιηθεί για την ομαδοποίηση μεγάλων συνόλων δεδομένων δικτυακής κίνησης στο κατάλληλο υπολογιστικό περιβάλλον.

Παρόλα αυτά η επιλογή του κατάλληλου αριθμού των συστάδων είναι δύσκολη ιδικά σε μεγάλα και πολύπλοκα σύνολα δεδομένων. Επιπλέον τα αρχικά κεντροειδή μπορούν να επηρεάσουν το τελικό αποτέλεσμα της ομαδοποίησης.

Ακόμη, ο αλγόριθμος υποθέτει ότι οι συστάδες έχουν περίπου σφαιρική μορφή και μπορούν να διαχωριστούν με βάση την ευκλείδεια απόσταση. Αυτό ενδέχεται να μην ισχύει σε δεδομένα δικτυακής κίνησης όπου οι επιθέσεις μπορεί να σχηματίζουν ακανόνιστα ή επικαλυπτόμενα μοτίβα. Συνεπώς τα αποτελέσματα του K-Means πρέπει να αξιολογούνται προσεκτικά στο πλαίσιο της ανίχνευσης εισβολών [18].

4.6 Isolation Forest

Ο Isolation Forest είναι ένας αλγόριθμος μη εποπτευόμενης ανίχνευσης ανωμαλιών. Αντί να βασίζεται στην απόσταση ή στην πυκνότητα των σημείων των δεδομένων όπως ο K-Means, εντοπίζει ανωμαλίες μέσω της απομόνωσης τους. Η βασική ιδέα είναι ότι οι ανωμαλίες δεν εμφανίζονται συχνά και

διαφέρουν από τις κανονικές παρατηρήσεις. Για αυτό μπορούν να απομονωθούν με λιγότερους τυχαίους διαχωρισμούς σε σύγκριση με τα κανονικά σημεία δεδομένων.

Ο αλγόριθμος δημιουργεί ένα σύνολο τυχαίων δυαδικών δέντρων που ονομάζονται Isolation Trees. Σε κάθε κόμβο ενός δέντρου επιλέγεται τυχαία ένα χαρακτηριστικό και στη συνέχεια μία τιμή διαχωρισμού μεταξύ της ελάχιστης και της μέγιστης τιμής του συγκεκριμένου χαρακτηριστικού. Η διαδικασία επαναλαμβάνεται μέχρι να απομονωθεί μια παρατήρηση ή να φτάσει το καθορισμένο όριο βάθους του δέντρου.

Μετάπειτα με βάση το μήκος της διαδρομής, δηλαδή τον αριθμό των διαχωρισμών που απαιτούνται για να απομονωθεί μία παρατήρηση, χαρακτηρίζεται η εγγραφή ως ανωμαλία ή όχι. Μικρότερο μήκος διαδρομής σημαίνει ότι η εγγραφή είναι πιθανότερο να είναι ανωμαλία καθώς απομονώνεται πιο γρήγορα από τις υπόλοιπες.

Ο Isolation Forest μπορεί να συμβάλει στην ανίχνευση επιθέσεων μηδενικής ημέρας από την στιγμή που εμφανίζουν διαφορετική συμπεριφορά από την καλοήγη δικτυακή δραστηριότητα.

Τα πλεονεκτήματα του είναι ότι δεν απαιτεί επισημασμένα δεδομένα κατά την εκπαίδευση, μπορεί να εφαρμοστεί σε σύνολα δεδομένων υψηλής διάστασης και να διαχειριστεί μεγάλα σύνολα δεδομένων καθώς βασίζεται σε τυχαίους διαχωρισμούς αντί για τον υπολογισμό αποστάσεων μεταξύ όλων των εγγραφών.

Από την άλλη, η αποτελεσματικότητα του εξαρτάται από το πόσο οι επιθέσεις έχουν χαρακτηριστικά που διαφέρουν επαρκώς από την καλοήγη δικτυακή δραστηριότητα [41].

4.7 One-Class Support Vector Machines

Σε αντίθεση με το κλασικό SVM που χρησιμοποιείται κυρίως για δυαδική ταξινόμηση, το One-Class SVM μαθαίνει ένα όριο απόφασης που περιβάλλει την πλειονότητα των δεδομένων της κανονικής κλάσης. Οι παρατηρήσεις που αποκλίνουν από αυτό το όριο χαρακτηρίζονται ως ανωμαλίες. Στην ανίχνευση εισβολών μπορεί να εκπαιδευτεί με δεδομένα καλοήθους δικτυακής δραστηριότητας και στην συνέχεια να εντοπίσει αποκλίσεις που ενδέχεται να σχετίζονται με επιθέσεις.

Το One-Class SVM χρησιμοποιεί συναρτήσεις πυρήνα για την χαρτογράφηση των δεδομένων σε χώρο χαρακτηριστικών υψηλής διάστασης ώστε να μπορεί να μοντελοποιεί σύνθετες και μη γραμμικές σχέσεις.

Τα πλεονεκτήματα του είναι ότι είναι κατάλληλο για ανισόρροπα δεδομένα όπου οι ανωμαλίες εμφανίζονται σπάνια και με την χρήση κατάλληλων πυρήνων μπορεί να χειριστεί μη γραμμικές σχέσεις.

Ωστόσο όταν οι ανωμαλίες δεν αποκλείουν αρκετά από την κανονική κλάση η απόδοση του μοντέλου μειώνεται. Ακόμα η επιλογή των κατάλληλων υπερπαραμέτρων μπορεί να είναι δύσκολη διαδικασία [42].

4.8 Επίλογος

Αυτό το κεφάλαιο παρουσίασε τα μοντέλα μηχανικής μάθησης που έχουν επιλεγεί στην πειραματική ανάλυση αυτής της Δ.Ε. Συζητήθηκαν έξι μοντέλα, τόσο εποπτευόμενα όσο και μη εποπτευόμενα

μοντέλα μηχανικής μάθησης. Μοντέλα όπως η Logistic Regression, το Random Forest και το Support Vector Machines εκπαιδεύτηκαν σε επισημασμένα δεδομένα και κατηγοριοποιούν την δραστηριότητα του δικτύου ως καλοήγη ή κακόβουλη.

Εκτός από τα μοντέλα ταξινόμησης, παρουσιάστηκαν επίσης τρεις μη εποπτευόμενοι αλγόριθμοι, ο K-Means, το Isolation Forest και το One-Class SVM. Ο K-Means μπορεί να αξιοποιηθεί για την ομαδοποίηση και τον εντοπισμό πιθανών ανωμαλιών ενώ ο Isolation Forest και το One-Class SVM χρησιμοποιούνται άμεσα για την ανίχνευση ανωμαλιών.

Το επόμενο κεφάλαιο περιγράφει τα εργαλεία λογισμικού και το περιβάλλον ανάπτυξης που χρησιμοποιήθηκαν για την επεξεργασία του συνόλου δεδομένων, την εκπαίδευση των μοντέλων και τη αξιολόγηση τους.

Κεφάλαιο 5ο: Εργαλεία και Περιβάλλον Λογισμικού

5.1 Εισαγωγή

Το κεφάλαιο αυτό παρουσιάζει τα εργαλεία, το περιβάλλον, τις βιβλιοθήκες και το σύνολο δεδομένων που χρησιμοποιήθηκαν για τον προγραμματισμό και την αξιολόγηση των μοντέλων. Αυτή η διαδικασία απαιτήσε τη χρήση γενικών προγραμματιστικών εργαλείων, βιβλιοθηκών που εξειδικεύονται στη μηχανική μάθηση αλλά και εργαλείων για την επεξεργασία των δεδομένων της βάσεις και για την οπτικοποίηση των αποτελεσμάτων.

Ο σκοπός του κεφαλαίου δεν είναι να αναλύσει τη μεθοδολογία εκπαίδευσης των μοντέλων το οποίο θα γίνει στο κεφάλαιο 7, αλλά να αναφέρει την τεχνική υποδομή που αξιοποιήθηκε. Με την αναφορά της τεχνικής υποδομής διασφαλίζεται η αναπαραγωγικότητα του πειράματος και παρέχεται μια σαφής κατανόηση του υπολογιστικού περιβάλλοντος στο οποίο έγινε η έρευνα.

5.2 Προγραμματιστικό Περιβάλλον

Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε στην επεξεργασία των δεδομένων και στην εκπαίδευση των μοντέλων είναι η Python, η οποία επιλέχθηκε λόγω της ύπαρξης πολλών βιβλιοθηκών για Μηχανική Μάθηση και της ευελιξίας της ως γλώσσα.

Η διαδικασία ανάπτυξης έγινε στο ολοκληρωμένο περιβάλλον του Visual Studio Code (VS Code), το οποίο βοήθησε στην οργάνωση των φακέλων και των αρχείων παρέχοντας ταυτόχρονα μία μεγάλη γκάμα από επεκτάσεις. Το εικονικό περιβάλλον Python .venv διασφαλίζει την αναπαραγωγικότητα του πειράματος. Με τη χρήση των δύο παραπάνω εργαλείων οι βιβλιοθήκες παραμένουν στην ίδια έκδοση παρέχοντας σταθερότητα στην προγραμματιστική διαδικασία.

Επιπλέον το IPython χρησιμοποιήθηκε ως διαδραστικό κέλυφος για να διευκολύνει τις δοκιμές, τη διόρθωση σφαλμάτων και την ταχεία δημιουργία πρωτοτύπων σε αποσπάσματα κώδικα. Το συγκεκριμένο εργαλείο ήταν πολύ χρήσιμο για την επεξεργασία των δεδομένων στην βάση αλλά και για την εκπαίδευση των μοντέλων.

Σε συνδυασμό η Python, το VS Code, το .venv και το IPython δημιούργησαν ένα σταθερό και ευέλικτο περιβάλλον προγραμματισμού για την εκπαίδευση και την αξιολόγηση των μοντέλων.

5.3 Βιβλιοθήκες και Frameworks

Το πειραματικό κομμάτι της Δ.Ε. βασίζεται σε αρκετές βιβλιοθήκες Python ανοιχτού κώδικα οι οποίες είναι απαραίτητες για τη διαχείριση των δεδομένων, την εκπαίδευση των μοντέλων και την αξιολόγηση τους. Παρακάτω αναγράφονται οι βιβλιοθήκες και τα frameworks:

- NumPy: Παρέχει υποστήριξη για αριθμητικούς υπολογισμούς και αποτελεί τη βάση για τις περισσότερες επιστημονικές υπολογιστικές εργασίες σε Python.
- Pandas: Χρησιμοποιείται για τον χειρισμό και την ανάλυση δεδομένων, ιδιαίτερα για δεδομένα σε μορφή πίνακα, και την προετοιμασία χαρακτηριστικών για τα μοντέλα μηχανικής μάθησης.
- Scikit-learn: Είναι η κύρια βιβλιοθήκη για την εκπαίδευση αλγορίθμων μηχανικής μάθησης όπως Decision Trees, Random Forest, Support Vector Machines και K-Means. Παρέχει επίσης

εργαλεία για την αξιολόγηση μοντέλων, την προεπεξεργασία δεδομένων και τις μετρήσεις απόδοσης.

- **Matplotlib:** Χρησιμοποιείται για την οπτικοποίηση των αποτελεσμάτων όπως η απεικόνιση των μετρικών αξιολόγησης.
- **Joblib:** Χρησιμοποιείται για την αποθήκευση και φόρτωση εκπαιδευμένων μοντέλων, διασφαλίζοντας ότι τα πειράματα μπορούν να αναπαραχθούν χωρίς να απαιτείται εκπαίδευση από την αρχή.

5.4 Προδιαγραφές Συστήματος

Το πειραματικό κομμάτι της διπλωματικής εργασίας πραγματοποιήθηκε σε προσωπικό υπολογιστή με τις παρακάτω προδιαγραφές:

- Λειτουργικό Σύστημα: Windows 11 Pro
- Επεξεργαστής (CPU): Intel Core i7-14700F @ 2.1 GHz – 5.4 GHz
- Μνήμη (RAM): 32 GB DDR5
- Κάρτα Γραφικών (GPU): NVIDIA RTX 3060
- Χώρος Αποθήκευσης: 2 TB Lexar M.2 SSD

5.5 Επίλογος

Το κεφάλαιο αυτό παρουσίασε το περιβάλλον προγραμματισμού και τα χαρακτηριστικά του συστήματος που χρησιμοποιήθηκαν για τη διεξαγωγή των πειραμάτων στην παρούσα Δ.Ε. Η γλώσσα προγραμματισμού που επιλέχθηκε ήταν η Python στο περιβάλλον Visual Studio Code, με την υποστήριξη ενός εικονικού περιβάλλοντος (.venv) και του IPython για διαδραστικές δοκιμές. Μια γκάμα από βιβλιοθήκες Python όπως NumPy, Pandas, Scikit-learn, Joblib και Matplotlib παρείχαν τα απαραίτητα εργαλεία για την επεξεργασία δεδομένων, την εκπαίδευση μοντέλων και την αξιολόγηση τους. Τέλος περιγράφηκαν τα χαρακτηριστικά του προσωπικού υπολογιστή στον οποίο διεξήχθησαν τα πειράματα.

Τα εργαλεία και οι πόροι που αναφέρθηκαν έθεσαν μια σταθερή βάση για την έρευνα, επιτρέποντας την εκπαίδευση και την αξιολόγηση πολλαπλών μοντέλων μηχανικής μάθησης για την ανίχνευση εισβολών. Το επόμενο κεφάλαιο θα βασιστεί σε αυτό το περιβάλλον περιγράφοντας τα βήματα επεξεργασίας των δεδομένων της βάσης.

Κεφάλαιο 6ο: Επεξεργασία βάσης δεδομένων και μέτρα αξιολόγησης

6.1 Εισαγωγή

Ο σκοπός του κεφαλαίου είναι να παρουσιάσει τη μεθοδολογία για την επεξεργασία και προετοιμασία της βάσης δεδομένων για την εκπαίδευση των μοντέλων Μηχανικής Μάθησης. Ενώ το κεφάλαιο 5 έθεσε τα εργαλεία και το προγραμματιστικό περιβάλλον που χρησιμοποιήθηκε στη Δ.Ε., αυτό το κεφάλαιο επικεντρώνεται στα πρακτικά βήματα που έγιναν για την πραγματοποίηση του πειράματος.

Η βάση δεδομένων σε αυτή την έρευνα είναι η CICIDS2018, η οποία αποτελεί ένα σύνολο δεδομένων που χρησιμοποιείται ευρέως για μελέτες ανίχνευσης εισβολών. Η βάση δεδομένων περιέχει καλοήγη δραστηριότητα μαζί με πολλαπλές κατηγορίες επιθέσεων, καθιστώντας την ιδανική για την αξιολόγηση της αποτελεσματικότητας των διαφόρων μοντέλων μηχανικής μάθησης στην ανίχνευση εισβολών.

Πριν από την εκπαίδευση των μοντέλων, τα δεδομένα της βάσης πρέπει να επεξεργαστούν για να διασφαλιστεί η ποιότητα των δεδομένων και να βελτιστοποιηθεί η απόδοση των μοντέλων. Αυτό περιλαμβάνει καθαρισμό, μετασχηματισμό χαρακτηριστικών, ομαλοποίηση και τον χειρισμό της ανισορροπίας μεταξύ των κλάσεων. Μετά την επεξεργασία η βάση χωρίστηκε σε υποσύνολα εκπαίδευσης και δοκιμών για τη σωστή εκπαίδευση και αξιολόγηση των μοντέλων.

6.2 Περιγραφή βάσης δεδομένων

Τα πειράματα σε αυτή τη Δ.Ε. πραγματοποιήθηκαν με τη βάση δεδομένων CICIDS2018, η οποία αναπτύχθηκε από το Καναδικό Ινστιτούτο Κυβερνοασφάλειας. Η CICIDS2018 είναι μία από τις πιο ευρέως χρησιμοποιημένες βάσεις δεδομένων για την έρευνα ανίχνευσης εισβολών καθώς παρέχει μία ρεαλιστική και ολοκληρωμένη αναπαράσταση της σύγχρονης κίνησης δικτύων.

Τα δεδομένα συλλέχθηκαν σε διάστημα δέκα ημερών και περιλαμβάνουν καλοήγη δραστηριότητα και μία μεγάλη ποικιλία κακόβουλων δραστηριοτήτων που προσομοιώνουν σενάρια επιθέσεων στον πραγματικό κόσμο. Αυτές περιλαμβάνουν κατανεμημένες επιθέσεις άρνησης υπηρεσίας (DDoS), brute force προσπάθειες εισχώρησης, δραστηριότητα botnet, επιθέσεις ιστού και διάφορα exploits. Η κίνηση καταγράφηκε σε μορφή ακατέργαστων πακέτων (αρχεία PCAP) και υποβλήθηκε σε επεξεργασία σε αμφίδρομες ροές χρησιμοποιώντας το εργαλείο CICFlowMeter, με αποτέλεσμα τα δεδομένα να είναι σε μορφή πίνακα που είναι κατάλληλο για εφαρμογές μηχανικής μάθησης.

Το CICIDS2018 περιέχει συνολικά εκατομμύρια εγγραφές δικτυακής δραστηριότητας. Κάθε ροή αποτελείται από 79 χαρακτηριστικά και τη στήλη Label που χαρακτηρίζει αν είναι καλοήγη ή κάποια μορφή επίθεσης. Στα 79 χαρακτηριστικά περιγράφονται ιδιότητες όπως διάρκεια ροής μήκος πακέτου, χρόνοι μεταξύ αφίξεων, χαρακτηριστικά ειδικά για το πρωτόκολλο. Ο μεγάλος αριθμός εγγραφών και χαρακτηριστικών καθιστά τη συγκεκριμένη βάση ιδιαίτερα πολύτιμη για την αξιολόγηση μοντέλων μηχανικής μάθησης σε συνθήκες που προσεγγίζουν λειτουργικά περιβάλλοντα.

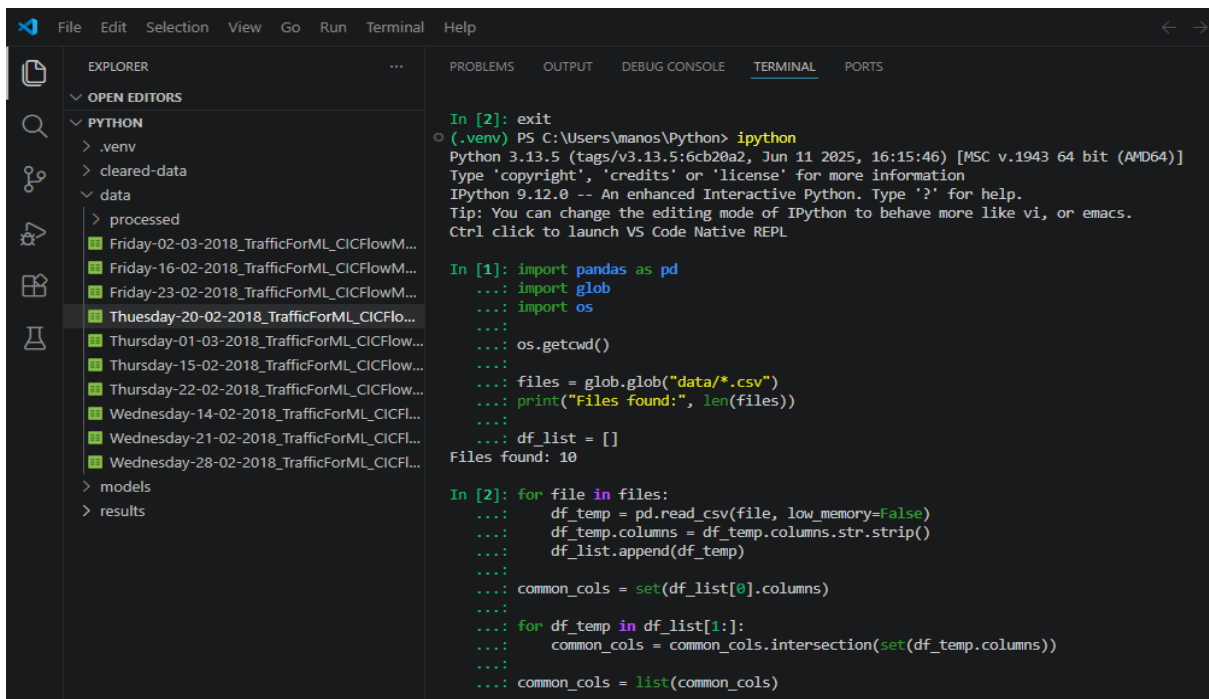
Οι παράγοντες που οδήγησαν στην επιλογή της CICIDS2018 για την παρούσα Δ.Ε. είναι οι εξής:

- **Ρεαλισμός:** Περιλαμβάνει την κίνηση που παράγεται τόσο από κανονικούς χρήστες όσο και από προσομοιωμένες επιθέσεις, γεγονός που κάνει τα δεδομένα να μοιάζουν με δραστηριότητα πραγματικού δικτύου.
- **Ποικιλομορφία επιθέσεων:** Οι πολλαπλές κατηγορίες επιθέσεων επιτρέπουν τη δοκιμή μοντέλων έναντι διαφορετικών τύπων απειλών.
- **Πλούτος χαρακτηριστικών:** Ο μεγάλος αριθμός χαρακτηριστικών παρέχει ευκαιρίες τόσο για κλασικούς όσο και για προηγμένους αλγορίθμους μηχανικής μάθησης.
- **Αποδοχή από την ερευνητική κοινότητα:** Το CICIDS2018 έχει υιοθετηθεί ευρέως στην έρευνα IDS, επιτρέποντας ουσιαστική σύγκριση με άλλες μελέτες.

6.3 Επεξεργασία δεδομένων

Τα δεδομένα στις βάσεις σπάνια είναι κατάλληλα για άμεση χρήση στη μηχανική μάθηση. Για τη σωστή εκπαίδευση των μοντέλων η βάση δεδομένων CICIDS2018 υποβλήθηκε σε μία σειρά βημάτων προεπεξεργασίας. Τα βήματα αυτά στόχευαν στον καθαρισμό των δεδομένων, στη μετατροπή τους και στην αντιμετώπιση προκλήσεων όπως η ανισορροπία των κλάσεων.

Αρχικά τα αρχεία της βάσης τα οποία αντιστοιχούν σε διαφορετικά σενάρια επιθέσεων ενοποιήθηκαν σε ένα ενιαίο σύνολο δεδομένων. Κατά την διαδικασία της συγχώνευσης παρατηρήθηκαν διαφοροποιήσεις στη δομή των αρχείων, καθώς ένα από τα δέκα αρχεία είχε διαφορετικό αριθμό χαρακτηριστικών. Για την αντιμετώπιση του προβλήματος διατηρήθηκαν μόνο τα κοινά χαρακτηριστικά που υπήρχαν σε όλα τα αρχεία.



```

File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
PYTHON
.venv
cleared-data
data
  processed
    Friday-02-03-2018_TrafficForML_CICFlowM...
    Friday-16-02-2018_TrafficForML_CICFlowM...
    Friday-23-02-2018_TrafficForML_CICFlowM...
    Tuesday-20-02-2018_TrafficForML_CICFlo...
    Thursday-01-03-2018_TrafficForML_CICFlow...
    Thursday-15-02-2018_TrafficForML_CICFlow...
    Thursday-22-02-2018_TrafficForML_CICFlow...
    Wednesday-14-02-2018_TrafficForML_CICFI...
    Wednesday-21-02-2018_TrafficForML_CICFI...
    Wednesday-28-02-2018_TrafficForML_CICFI...
  models
  results

In [2]: exit
(.venv) PS C:\Users\manos\Python> ipython
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.12.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: You can change the editing mode of IPython to behave more like vi, or emacs.
Ctrl click to launch VS Code Native REPL

In [1]: import pandas as pd
...: import glob
...: import os
...:
...: os.getcwd()
...:
...: files = glob.glob("data/*.csv")
...: print("Files found:", len(files))
...:
...: df_list = []
Files found: 10

In [2]: for file in files:
...:     df_temp = pd.read_csv(file, low_memory=False)
...:     df_temp.columns = df_temp.columns.str.strip()
...:     df_list.append(df_temp)
...:
...: common_cols = set(df_list[0].columns)
...:
...: for df_temp in df_list[1:]:
...:     common_cols = common_cols.intersection(set(df_temp.columns))
...:
...: common_cols = list(common_cols)
  
```

Εικόνα 6.1 Συγχώνευση των αρχείων της βάσης δεδομένων CICIDS2018.

Κ

```

In [4]: df_list_clean = [df_temp[common_cols] for df_temp in df_list]
...:
...: df = pd.concat(df_list_clean, ignore_index=True)
...:
...: if "Label" in df.columns:
...:     print(df["Label"].value_counts())
...:
...: os.makedirs("data/processed", exist_ok=True)
...: df.to_csv("data/processed/merged_dataset.csv", index=False)
...:
...: print("Merged dataset saved.")

```

Label	
Benign	13484708
DDoS attack-HOIC	686012
DDoS attacks-LOIC-HTTP	576191
DoS attacks-Hulk	461912
Bot	286191
FTP-BruteForce	193360
SSH-Bruteforce	187589
Infiltration	161934
DoS attacks-SlowHTTPTest	139890
DoS attacks-GoldenEye	41508
DoS attacks-Slowloris	10990
DDoS attack-LOIC-UDP	1730
Brute Force -Web	611
Brute Force -XSS	230
SQL Injection	87
Label	59

```

Name: count, dtype: int64
Merged dataset saved.

```

Εικόνα 6.2 Επιλογή κοινών χαρακτηριστικών και αποθήκευση ενιαίου συνόλου δεδομένων

6.3.1 Καθαρισμός δεδομένων

Το ενιαίο σύνολο δεδομένων περιείχε ελλείπουσες τιμές, άπειρες τιμές και διπλότυπες καταχωρήσεις. Αρχικά οι άπειρες τιμές μετατράπηκαν σε NaN. Στη συνέχεια οι εγγραφές με ελλείπουσες τιμές αφαιρέθηκαν καθώς το μικρό τους ποσοστό δεν επηρέασε σημαντικά την αντιπροσωπευτικότητα του συνόλου δεδομένων. Επιπλέον αφαιρέθηκε το χαρακτηριστικό Timestamp το οποίο δεν προσέφερε ουσιώδη πληροφορία και ήταν σε μορφή string.

Πραγματοποιήθηκε μετατροπή όλων των χαρακτηριστικών σε αριθμητική μορφή εκτός από το Label καθώς ορισμένες στήλες περιείχαν μη αριθμητικές τιμές. Οι μη έγκυρες τιμές μετατράπηκαν σε NaN και αφαιρέθηκαν στο στάδιο καθαρισμού.

Τέλος εντοπίστηκε μεγάλος αριθμός διπλότυπων εγγραφών οι οποίες αφαιρέθηκαν για τη σωστή εκπαίδευση των μοντέλων.

```

In [1]: import pandas as pd

In [2]: import numpy as np

In [3]: df = pd.read_csv("data/processed/merged_dataset.csv", low_memory= False)

In [4]: df.drop(columns=["Timestamp"], inplace=True)

In [5]: for col in df.columns:
...:     if col != "Label":
...:         df[col] = pd.to_numeric(df[col], errors="coerce")
...:

In [6]:

In [6]: In [8]: df.replace([np.inf, -np.inf], np.nan, inplace=True)
...:     df.dropna(inplace=True)

```

Εικόνα 6.3 Αφαίρεση του Timestamp, μετατροπή των χαρακτηριστικών σε αριθμητική μορφή, αντικατάσταση άπειρων τιμών με NaN και αφαίρεση NaN εγγραφών

```

In [9]: df.shape
Out[9]: (16137183, 79)

In [10]: df.duplicated().sum()
Out[10]: np.int64(4221045)

In [11]: df.drop_duplicates(inplace=True)

In [12]: df.shape
Out[12]: (11916138, 79)

In [13]: df.to_csv("data/processed/cleaned_dataset.csv", index=False)

```

Εικόνα 6.4 Αφαίρεση διπλότυπων εγγραφών, έλεγχος τελικής διάστασης του συνόλου δεδομένων και αποθήκευση του dataset

6.3.2 Επιλογή και επεξεργασία χαρακτηριστικών

Η CICIDS2018 έχει πάνω από 80 χαρακτηριστικά που σχετίζονται με την δικτυακή δραστηριότητα, αρκετά από τα οποία είναι πολύ συσχετισμένα ή περιττά, γεγονός που περιορίζει την αποτελεσματικότητα της ταξινόμησης. Τα μέτρα που υιοθετήθηκαν για τη βελτίωση της αποτελεσματικότητας των υπολογισμών και τη μείωση της υπερπροσαρμογής περιελάμβαναν:

- Πραγματοποιήθηκε ανάλυση συσχέτισης με σκοπό τον εντοπισμό χαρακτηριστικών με υψηλό βαθμό συσχέτισης (<0.95), τα οποία διαγράφηκαν ώστε να διατηρηθούν τα πιο κατατοπιστικά.

```

Help
Welcome X

Start

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

(.venv) PS C:\Users\manos\Python> ipython
Python 3.13.5 (tags/v3.13.5:6cb20a2, Jun 11 2025, 16:15:46) [MSC v.1943 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 9.12.0 -- An enhanced Interactive Python. Type '?' for help.
Tip: You can change the editing mode of IPython to behave more like vi, or emacs.
Ctrl click to launch VS Code Native REPL

In [1]: import numpy as np

In [2]: import pandas as pd

In [3]: df = pd.read_csv("data/processed/cleaned_dataset.csv", low_memory= False)

In [4]: corr_matrix = df.drop(columns=["Label"]).corr().abs()

In [5]: upper = corr_matrix.where(
...:     np.triu(np.ones(corr_matrix.shape), k=1).astype(bool) )

In [6]: high_corr_cols = [col for col in upper.columns if any(upper[col] > 0.95)]

In [7]: df_reduced = df.drop(columns=high_corr_cols)

In [8]: df.shape
Out[8]: (11916138, 79)

In [9]: df_reduced.shape
Out[9]: (11916138, 52)

In [10]: 

```

Εικόνα 6.5 Κώδικας ανάλυσης συσχέτισης των χαρακτηριστικών

- Κανονικοποίηση συνεχών χαρακτηριστικών με κλιμάκωση Min-Max έτσι ώστε όλες οι μεταβλητές να βρίσκονται στο διάστημα $[0,1]$. Η οποία έχει ιδιαίτερη σημασία για αλγορίθμους που βασίζονται σε αποστάσεις όπως SVM, καθώς εξασφαλίζει ότι όλα τα χαρακτηριστικά συνεισφέρουν ισότιμα στη διαδικασία εκπαίδευσης και αποτρέπει την κυριαρχία χαρακτηριστικών με μεγαλύτερες αριθμητικές τιμές.

Η κανονικοποίηση εφαρμόστηκε μόνο στα σύνολα δεδομένων εκπαίδευσης και δοκιμής των μοντέλων που απαιτούν κλιμακώμενα δεδομένα και θα παρουσιαστεί στο επόμενο κεφάλαιο.

6.3.3 Train/test διαίρεση και σενάρια zero day

Για να ελεγχθεί η ικανότητα των μοντέλων να εντοπίζουν μέχρι τώρα άγνωστες επιθέσεις, υιοθετήθηκε ένα πειραματικό σχέδιο τύπου «zero-day». Αντί να χρησιμοποιηθεί τυχαία διαίρεση δεδομένων εκπαίδευσης και δοκιμής, το σύνολο εκπαίδευσης απέκλειε ορισμένες κατηγορίες επιθέσεων που

εφαρμόστηκαν μόνο στη δοκιμή. Πρόκειται για μια προσομοιωμένη πραγματική κατάσταση, στην οποία τα μοντέλα αναμένεται να αναγνωρίσουν νέα μοτίβα επιθέσεων που δεν είχαν παρατηρηθεί κατά τη διαδικασία της εκπαίδευσης.

Δημιουργήθηκαν δύο διαφορετικά σενάρια zero-day για την αξιολόγηση των μοντέλων σε διαφορετικές περιπτώσεις. Στο πρώτο σενάριο, οι τύποι επιθέσεων που είχαν σχετικά μεγάλο αριθμό δειγμάτων (Bot και Infiltration) θεωρήθηκαν ως άγνωστες επιθέσεις και υπήρχαν μόνο στο σύνολο δοκιμής. Το δεύτερο σενάριο περιελάμβανε πιο απαιτητικές συνθήκες αξιολόγησης, με πιο σπάνιες κατηγορίες επιθέσεων (SQL Injection, Brute Force - XSS και Brute Force - Web) να επιλέγονται ως επιθέσεις zero-day.

Σε κάθε σενάριο, το σύνολο εκπαίδευσης αποτελούνταν από καλοήγη δραστηριότητα και όλους τους άλλους γνωστούς τύπους επιθέσεων, ενώ το σύνολο δοκιμής αποτελούνταν από καλοήγη δραστηριότητα και τους αντίστοιχους τύπους επιθέσεων «zero-day». Με αυτόν τον τρόπο διασφαλίστηκε ότι τα μοντέλα δοκιμάστηκαν ως προς την ικανότητά τους να διακρίνουν μεταξύ καλοήθους και κακόβουλης κίνησης σε περίπτωση εμφάνισης προηγουμένως άγνωστων τύπων επιθέσεων.

Δεδομένου ότι η καλοήγη δραστηριότητα στο σύνολο δεδομένων είναι πολύ περισσότερη των επιθέσεων, πραγματοποιήθηκε ένα επιπλέον βήμα προεπεξεργασίας, η υποδειγματοληψία της κατηγορίας της benign κίνησης. Ωστόσο, η διαδικασία εφαρμόστηκε μόνο στο σύνολο εκπαίδευσης και δεν εφαρμόστηκε για την εκπαίδευση όλων των μοντέλων. Το σύνολο δοκιμής αφέθηκε σκόπιμα αμετάβλητο, προκειμένου να διατηρηθεί η ρεαλιστική κατανομή των δεδομένων και να διασφαλιστεί η ορθή αξιολόγηση.

```

In [1]: import pandas as pd

In [2]: df = pd.read_csv("data/processed/reduced_dataset.csv", low_memory= Fa
      : lse)

In [3]: scenario_a_zero_day = ["Bot", "Infiltration"]

In [4]: train_df_a = df[~df["Label"].isin(scenario_a_zero_day)].copy()

In [5]: test_df_a = df[
...:     (df["Label"].isin(scenario_a_zero_day)) |
...:     (df["Label"] == "Benign")
...: ].copy()

In [6]: train_df_a["BinaryLabel"] = train_df_a["Label"].apply(lambda x: 0 if
      : x == "Benign" else 1)

In [7]: test_df_a["BinaryLabel"] = test_df_a["Label"].apply(lambda x: 0 if x
      : == "Benign" else 1)

In [8]: print("Scenario A - Train labels")
...: print(train_df_a["Label"].value_counts())
Scenario A - Train labels
Label
Benign                10564771
DDoS attacks-LOIC-HTTP    575364
DDoS attack-H0IC         198861
DoS attacks-Hulk         145199
SSH-Bruteforce           94048
DoS attacks-GoldenEye    41406
DoS attacks-Slowloris    9908
DDoS attack-LOIC-UDP     1730
Brute Force -Web         555
Brute Force -XSS         228
SQL Injection            84
DoS attacks-SlowHTTPTest  55
FTP-BruteForce           53
Name: count, dtype: int64

In [9]: print("Scenario A - Test labels")
...: print(test_df_a["Label"].value_counts())
Scenario A - Test labels
Label
Benign                10564771
Bot                   144535
Infiltration          139341
Name: count, dtype: int64

```

Εικόνα 6.6 Δημιουργία πρώτου σεναρίου

```

In [14]: scenario_b_zero_day = [
...:     "SQL Injection",
...:     "Brute Force -XSS",
...:     "Brute Force -Web"
...: ]

In [15]: train_df_b = df[~df["Label"].isin(scenario_b_zero_day)].copy()
...:

In [16]: test_df_b = df[
...:     (df["Label"].isin(scenario_b_zero_day)) |
...:     (df["Label"] == "Benign")
...: ].copy()

In [17]: train_df_b["BinaryLabel"] = train_df_b["Label"].apply(lambda x: 0 if x == "Benign" else 1)

In [18]: test_df_b["BinaryLabel"] = test_df_b["Label"].apply(lambda x: 0 if x == "Benign" else 1)

In [19]: print("Scenario B - Train labels")
...: print(train_df_b["Label"].value_counts())
Scenario B - Train labels
Label
Benign                10564771
DDoS attacks-LOIC-HTTP    575364
DDoS attack-HOIC        198861
DoS attacks-Hulk        145199
Bot                    144535
Infiltration            139341
SSH-Bruteforce           94048
DoS attacks-GoldenEye    41406
DoS attacks-Slowloris     9908
DDoS attack-LOIC-UDP      1730
DoS attacks-SlowHTTPTest    55
FTP-BruteForce           53
Name: count, dtype: int64

In [20]: print("Scenario B - Test labels")
...: print(test_df_b["Label"].value_counts())
Scenario B - Test labels
Label
Benign                10564771
Brute Force -Web         555
Brute Force -XSS         228
SQL Injection             84
Name: count, dtype: int64

```

Εικόνα 6.7 Δημιουργία δεύτερου σεναρίου

```

In [7]: benign_df_a = df_train[df_train["BinaryLabel"] == 0]
...: attack_df_a = df_train[df_train["BinaryLabel"] == 1]

In [8]:
...: print("Scenario A - before undersampling")
...: print("Benign:", len(benign_df_a))
...: print("Attack:", len(attack_df_a))
Scenario A - before undersampling
Benign: 10564771
Attack: 1067491

In [9]: benign_sample_a = benign_df_a.sample(n=len(attack_df_a) * 2, random_state=42)

In [10]: train_df_a_balanced = pd.concat([benign_sample_a, attack_df_a], ignore_index=True)

In [11]: print("Scenario A - after undersampling")
...: print(train_df_a_balanced["BinaryLabel"].value_counts())
...: print(train_df_a_balanced["Label"].value_counts())
Scenario A - after undersampling
BinaryLabel
0    2134982
1    1067491
Name: count, dtype: int64
Label
Benign                2134982
DDoS attacks-LOIC-HTTP  575364
DDoS attack-HOIC       198861
DoS attacks-Hulk       145199
SSH-Bruteforce         94048
DoS attacks-GoldenEye   41406
DoS attacks-Slowloris   9908
DDoS attack-LOIC-UDP    1730
Brute Force -Web        555
Brute Force -XSS        228
SQL Injection           84
DoS attacks-SlowHTTPTest 55
FTP-BruteForce          53
Name: count, dtype: int64

In [12]: train_df_a_balanced.to_csv("data/processed/scenario_a/train_df_a_balanced.csv", index=False)

```

Εικόνα 6.8 υποδειγματοληψία στο Σενάριο Α

```

In [1]: import pandas as pd

In [2]: df_train= pd.read_csv("data/processed/scenario_b/train_scenario_b.csv", low_memory= False)

In [3]: benign_df_b = df_train[df_train["BinaryLabel"] == 0]
...: attack_df_b = df_train[df_train["BinaryLabel"] == 1]

In [4]: print("Scenario B - before undersampling")
...: print("Benign:", len(benign_df_b))
...: print("Attack:", len(attack_df_b))
Scenario B - before undersampling
Benign: 10564771
Attack: 1350500

In [5]: benign_sample_b = benign_df_b.sample(n=len(attack_df_b) * 2, random_state=42)

In [6]: train_df_b_balanced = pd.concat([benign_sample_b, attack_df_b], ignore_index=True)

In [7]: print("Scenario B - after undersampling")
...: print(train_df_b_balanced["BinaryLabel"].value_counts())
...: print(train_df_b_balanced["Label"].value_counts())
Scenario B - after undersampling
BinaryLabel
0    2701000
1    1350500
Name: count, dtype: int64
Label
Benign                2701000
DDoS attacks-LOIC-HTTP  575364
DDoS attack-HOIC       198861
DoS attacks-Hulk       145199
Bot                    144535
Infiltration           139341
SSH-Bruteforce         94048
DoS attacks-GoldenEye   41406
DoS attacks-Slowloris   9908
DDoS attack-LOIC-UDP    1730
DoS attacks-SlowHTTPTest  55
FTP-BruteForce          53
Name: count, dtype: int64

In [8]: train_df_b_balanced.to_csv("data/processed/scenario_b/train_df_b_balanced.csv", index=False)

```

Εικόνα 6.9 υποδειγματοληψία στο Σενάριο B

6.4 Μεθοδολογία αξιολόγησης

Χρησιμοποιήθηκε ένα συστηματικό πλαίσιο αξιολόγησης για την εκτίμηση της απόδοσης των μοντέλων μηχανικής μάθησης. Δεδομένου ότι η ανίχνευση εισβολών είναι μια δραστηριότητα που σχετίζεται με την ασφάλεια, η συνολική ακρίβεια δεν είναι αρκετή.

6.4.1 Confusion Matrix

Ο πίνακας σύγχυσης είναι η βάση για την αξιολόγηση της απόδοσης. Αποτελείται από τέσσερα στοιχεία :

- True Positives (TP) : Κακόβουλη δραστηριότητα που ταξινομήθηκε σωστά ως επίθεση.
- True Negatives (TN) : Καλοήγη δραστηριότητα που ταξινομήθηκε σωστά ως καλοήγη.

- False Positives (FP) : καλοήγη δραστηριότητα που ταξινομήθηκε εσφαλμένα ως επίθεση.
- False Negatives (FN) : Κακόβουλη δραστηριότητα που ταξινομήθηκε εσφαλμένα ως καλοήγη.

Στην ανίχνευση εισβολών τα False Negatives είναι ιδιαίτερα κρίσιμα, καθώς αντιπροσωπεύουν επιθέσεις που δεν ανιχνεύθηκαν.

6.4.2 Μετρήσεις απόδοσης

Με βάση του Confusion Matrix υπολογίστηκαν οι ακόλουθες μετρήσεις :

- $$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Αν και το Accuracy παρέχει ένα γενικό μέτρο ορθότητας, μπορεί να είναι παραπλανητικό σε μη ισορροπημένες βάσεις δεδομένων.

- $$\text{Precision} = \frac{TP}{TP + FP}$$

Το Precision μετρά πόσες από τις επιθέσεις που ανιχνεύθηκαν ήταν πραγματικές επιθέσεις. Το υψηλό Precision μειώνει τους ψευδείς συναγερμούς.

- $$\text{Recall} = \frac{TP}{TP + FN}$$

Το Recall είναι ζωτικής σημασίας στην κυβερνοασφάλεια, καθώς μετρά την ικανότητα ανίχνευσης πραγματικών επιθέσεων.

- $$\text{F1-Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Το F1-score εξισορροπεί το Precision και το Recall καθιστώντας τον κατάλληλο για μη ισορροπημένα σύνολα δεδομένων.

- $$\text{False Positive Rate (FPR)} = \frac{FP}{FP + TN}$$

Ένα χαμηλό FPR είναι σημαντικό στα πραγματικά IDS ώστε να μην υπάρχει μεγάλος όγκος ψευδών ειδοποιήσεων.

- ROC και AUC. Η καμπύλη ROC απεικονίζει το ποσοστό του Recall σε σχέση με το False Positive Rate σε διάφορα όρια ταξινόμησης. Η περιοχή κάτω από την καμπύλη (AUC) δείχνει τη συνολική ικανότητα διάκρισης του μοντέλου.

6.4.3 Τρόπος αξιολόγησης

Κάθε ένα από τα μοντέλα εκπαιδεύτηκε και δοκιμάστηκε στις ίδιες συνθήκες με τη χρήση παρόμοιας διαδικασίας προεπεξεργασίας και ίσης κατανομής εκπαίδευσης/δοκιμής. Αυτή η μεθοδολογία διευκόλυνε τη σύγκριση των εποπτευόμενων και μη εποπτευόμενων μοντέλων με δίκαιο τρόπο.

Η τελική σύγκριση των μοντέλων βασίστηκε στα παρακάτω :

- Recall
- F1-score
- ROC-AUC
- False Positive Rate

6.5 Επίλογος

Σε αυτό το κεφάλαιο, περιγράφεται η μεθοδολογία που χρησιμοποιήθηκε για την προετοιμασία του συνόλου δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση των μοντέλων μηχανικής μάθησης, και την αξιολόγηση της απόδοσης τους.

Στα δεδομένα εφαρμόστηκε μια διαδικασία προεπεξεργασίας, η οποία περιελάμβανε τον καθαρισμό των δεδομένων, τη μετατροπή των χαρακτηριστικών, την κανονικοποίηση των δεδομένων, τη διαχείριση της ισορροπίας των κλάσεων και τον διαχωρισμό του συνόλου δεδομένων. Αυτά τα βήματα διασφάλισαν ότι τα δεδομένα έχουν τη σωστή μορφή για την εκπαίδευση μοντέλων μηχανικής μάθησης.

Για την αξιολόγηση των μοντέλων χρησιμοποιήθηκαν μέτρα όπως το Accuracy, το Precision, το Recall, το F1-score, το ROC-AUC και το FPR. Ιδιαίτερη προσοχή δόθηκε στο Recall και στο FN καθώς είναι κρίσιμα για τα συστήματα ανίχνευσης εισβολών.

Με το σύνολο δεδομένων έτοιμο και τα κριτήρια αξιολόγησης καθορισμένα, το επόμενο κεφάλαιο παρουσιάζει τη διαδικασία εκπαίδευσης των μοντέλων.

Κεφάλαιο 7ο: Εκπαίδευση μοντέλων μηχανικής μάθησης

7.1 Logistic Regression

7.1.1 Διαδικασία εκπαίδευσης

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν τα δύο σενάρια zero-day που περιγράφηκαν στο προηγούμενο κεφάλαιο. Τα δεδομένα κανονικοποιήθηκαν με χρήση της μεθόδου Min-Max, ώστε όλα τα χαρακτηριστικά να βρίσκονται στο διάστημα[0,1].

Επιπλέον, επιλέχθηκαν τα αρχεία εκπαίδευσης των δύο σεναρίων τα οποία έχουν υποστεί υποδειγματοληψία της benign κίνησης για τη βελτίωση της απόδοσης τους. Το μοντέλο εκπαιδεύτηκε με τις προεπιλεγμένες παραμέτρους χωρίς περαιτέρω ρύθμιση του ορίου απόφασης (threshold), ώστε να διασφαλιστεί μια αντικειμενική και συγκρίσιμη αξιολόγηση.

```
In [1]: import pandas as pd

In [2]: train_df_a_balanced= pd.read_csv("data/processed/scenario_a/train_df_
      : a_balanced.csv", low_memory= False)

In [3]: test_df_a= pd.read_csv("data/processed/scenario_a/test_scenario_a.csv
      : ", low_memory= False)

In [4]: X_train_a = train_df_a_balanced.drop(columns=["Label", "BinaryLabel"]
      : )
      ...: y_train_a = train_df_a_balanced["BinaryLabel"]

In [5]: X_test_a = test_df_a.drop(columns=["Label", "BinaryLabel"])
      ...: y_test_a = test_df_a["BinaryLabel"]

In [6]: from sklearn.preprocessing import MinMaxScaler

In [7]: scaler = MinMaxScaler()

In [8]: X_train_a_scaled = scaler.fit_transform(X_train_a)
      ...: X_test_a_scaled = scaler.transform(X_test_a)

In [9]: from sklearn.linear_model import LogisticRegression
In [10]: model_lra = LogisticRegression(
      ...:     class_weight="balanced",
      ...:     max_iter=1000,
      ...:     random_state=42
      ...: )

In [11]: model_lra.fit(X_train_a_scaled, y_train_a)
Out[11]: LogisticRegression(class_weight='balanced', max_iter=1000, random_state=42)
```

Εικόνα 7.1 Μέθοδος Min-Max και εκπαίδευση μοντέλου με το σενάριο A
Με τον ίδιο τρόπο έγινε η εκπαίδευση και στο δεύτερο σενάριο

7.1.2 Αποτελέσματα

Η απόδοση του μοντέλου αξιολογήθηκε με τα μέτρα Recall, F1-score, ROC-AUC και FPR.

7.1.2.1 Σενάριο Α

Στο πρώτο σενάριο όπου οι επιθέσεις Bot και Infiltration θεωρήθηκαν ως άγνωστες, το μοντέλο παρουσίασε πολύ χαμηλή ικανότητα ανίχνευσης:

- Recall: 0.039
- F1-score: 0.022
- ROC-AUC: 0.483
- FPR: 0.066

Τα αποτελέσματα δείχνουν ότι στο σενάριο Α το μοντέλο απέτυχε να αναγνωρίσει την πλειονότητα των επιθέσεων, ενώ ταυτόχρονα διατήρησε χαμηλό ποσοστό ψευδών συναγερμών. Η συμπεριφορά του μοντέλου δείχνει είναι συντηρητικό με αποτέλεσμα να αναγνωρίζει τα περισσότερα δείγματα ως καλοήγη.

```
In [13]: y_pred_a = model_lra.predict(X_test_a_scaled)
...: y_prob_a = model_lra.predict_proba(X_test_a_scaled)[: , 1]

In [14]: from sklearn.metrics import confusion_matrix, roc_auc_score

In [15]: cm = confusion_matrix(y_test_a, y_pred_a)
...: tn, fp, fn, tp = cm.ravel()
...:
...: recall = tp / (tp + fn)
...: precision = tp / (tp + fp) if (tp + fp) > 0 else 0
...: f1 = 2 * (precision * recall) / (precision + recall) if (precision +
...:     : recall) > 0 else 0
...: fpr = fp / (fp + tn)
...: roc_auc = roc_auc_score(y_test_a, y_prob_a)
...:
...: print("Confusion Matrix:")
...: print(cm)
...:
...: print("Recall:", recall)
...: print("F1:", f1)
...: print("ROC-AUC:", roc_auc)
...: print("FPR:", fpr)

Confusion Matrix:
[[9864711  700060]
 [ 272842  11034]]
Recall: 0.03886908368442559
F1: 0.022179563202910643
ROC-AUC: 0.483152204620083
FPR: 0.06626362275150119
```

Εικόνα 7.2 Αποτελέσματα Α Σεναρίου

7.1.2.2 Σενάριο B

Στο δεύτερο σενάριο θεωρήθηκαν πιο σπάνιες κατηγορίες επιθέσεων ως zero-day όπως SQL Injection, Brute Force -XSS, Brute Force -Web, το μοντέλο παρουσίασε διαφορετική συμπεριφορά:

- Recall: 0.532
- F1-score: 0.0008
- ROC-AUC: 0.696
- FPR: 0.106

Παρατηρείται σημαντική αύξηση του Recall και του ROC-AUC, γεγονός που υποδηλώνει ότι το μοντέλο κατάφερε να εντοπίσει ένα σημαντικό ποσοστό των άγνωστων επιθέσεων. Ωστόσο η αύξηση αυτή συνοδεύτηκε από πολύ χαμηλή τιμή F1-score και αύξηση του FPR, πράγμα που σημαίνει ότι το μοντέλο μπερδεύει την καλοήγη δραστηριότητα με επίθεση.

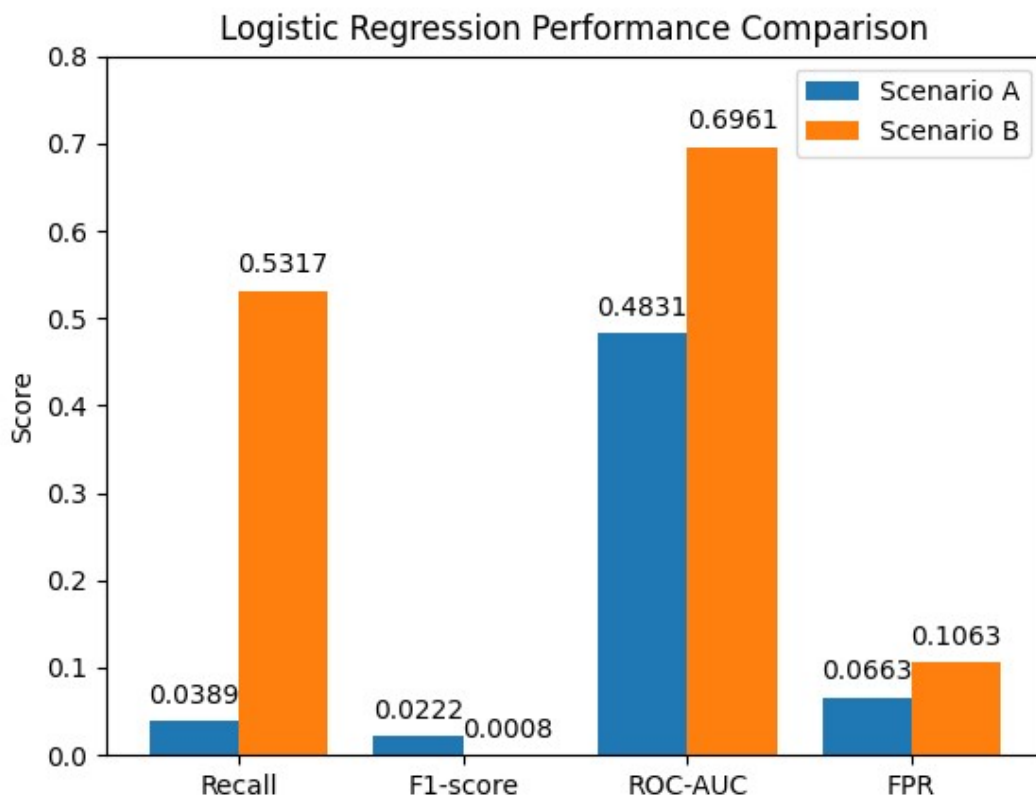
```
Confusion Matrix:  
[[9441214 1123557]  
 [   406    461]]  
Recall: 0.5317185697808535  
F1: 0.0008196393409104042  
ROC-AUC: 0.6961098153552726  
FPR: 0.10634939460590295
```

Εικόνα 7.3 Αποτελέσματα B Σεναρίου

7.1.3 Συζήτηση αποτελεσμάτων

Τα αποτελέσματα των δύο σεναρίων αποδεικνύουν ότι το μοντέλο Logistic Regression αποτυγχάνει να ανιχνεύσει τις επιθέσεις (Σενάριο A), είτε παρουσιάζει υψηλό ποσοστό FP (Σενάριο B).

Η συμπεριφορά αυτή υποδεικνύει ότι τα γραμμικά μοντέλα δεν μπορούν να διακρίνουν τα μοτίβα που χαρακτηρίζουν τις επιθέσεις zero-day γεγονός που τονίζει η ανισορροπία μεταξύ Recall και FPR.



Εικόνα 7.4 Logistic Regression Σύγκριση Αποτελεσμάτων

7.1.4 Συμπεράσματα

Το Logistic Regression λειτούργησε ως ένα χρήσιμο βασικό μοντέλο, ωστόσο τα αποτελέσματα δείχνουν ότι δεν είναι κατάλληλο για IDS εφαρμογές. Παρότι το μοντέλο μπορεί να εντοπίσει ένα ποσοστό επιθέσεων, η συνολική του απόδοση περιορίζεται είτε από χαμηλή ανίχνευση επιθέσεων είτε από υψηλό ποσοστό ψευδών συναγερμών.

Τα αποτελέσματα αυτά καθιστούν αναγκαία τη χρήση πιο σύνθετων μοντέλων, τα οποία θα εξεταστούν στα επόμενα τμήματα.

7.2 Support Vector Machines (SVM)

7.2.1 Διαδικασία εκπαίδευσης

Λόγω της υψηλής υπολογιστικής πολυπλοκότητας του SVM, δεν ήταν εφικτή η χρήση ολόκληρου του συνόλου εκπαίδευσης. Για αυτό τον λόγο για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε ένα υποσύνολο των δεδομένων, αποτελούμενο από ίσο αριθμό δειγμάτων benign και κακόβουλης κίνησης (100.000 ανά κατηγορία). Πρέπει να σημειωθεί ότι ακόμα και με το υποσύνολο δεδομένων η εκπαίδευση και η πρόβλεψη των αποτελεσμάτων διήρκεσε αρκετές ώρες.

Τα δεδομένα κανονικοποιήθηκαν με χρήση της μεθόδου Min-Max scaling, καθώς το SVM είναι ιδιαίτερα ευαίσθητο στην κλίμακα των χαρακτηριστικών. Το μοντέλο εκπαιδεύτηκε χωρίς περαιτέρω

ρύθμιση υπερπαραμέτρων, χρησιμοποιώντας τις προεπιλεγμένες τιμές, με στόχο την αντικειμενική σύγκριση με τα υπόλοιπα μοντέλα.

```
In [4]: benign_a = train_df_a_balanced[train_df_a_balanced["BinaryLabel"] ==
...: 0]
...: attack_a = train_df_a_balanced[train_df_a_balanced["BinaryLabel"] ==
...: 1]

In [5]: benign_sample_a = benign_a.sample(n=100000, random_state=42)
...: attack_sample_a = attack_a.sample(n=100000, random_state=42)

In [6]: train_sample_a = pd.concat([benign_sample_a, attack_sample_a], ignore
...: _index=True)

In [7]: train_sample_a = train_sample_a.sample(frac=1, random_state=42).reset
...: _index(drop=True)

In [8]: train_sample_a["BinaryLabel"].value_counts()
Out[8]:
BinaryLabel
1    100000
0    100000
Name: count, dtype: int64

In [9]: X_train_a_svm = train_sample_a.drop(columns=["Label", "BinaryLabel"])
...: y_train_a_svm = train_sample_a["BinaryLabel"]

In [10]: X_test_a_svm = test_df_a.drop(columns=["Label", "BinaryLabel"])
...: y_test_a_svm = test_df_a["BinaryLabel"]

In [11]: from sklearn.preprocessing import MinMaxScaler
...:
...: scaler = MinMaxScaler()
...:
...: X_train_a_svm_scaled = scaler.fit_transform(X_train_a_svm)
...: X_test_a_svm_scaled = scaler.transform(X_test_a_svm)

In [12]: from sklearn.svm import SVC
...:
...: svm_model_a = SVC(
...:     kernel="rbf",
...:     class_weight="balanced",
...:     probability=True,
...:     random_state=42
...: )

In [13]: svm_model_a.fit(X_train_a_svm_scaled, y_train_a_svm)
Out[13]: SVC(class_weight='balanced', probability=True, random_state=42)
```

Εικόνα 7.5 Εκπαίδευση SVM μοντέλου

7.2.2 Αποτελέσματα

Η απόδοση του μοντέλου αξιολογήθηκε με τα μέτρα Recall, F1-score, ROC-AUC και FPR.

7.2.2.1 Σενάριο A

Στο πρώτο σενάριο zero-day το SVM παρουσίασαν πολύ χαμηλή απόδοση:

- Recall: 0.020
- F1-score: 0.017
- ROC-AUC: 0.493
- FPR: 0.035

Το μοντέλο απέτυχε να εντοπίσει τις περισσότερες άγνωστες επιθέσεις, ενώ ταυτόχρονα είχε χαμηλό ποσοστό ψευδών συναγερμών. Τα αποτελέσματα δείχνουν ότι το μοντέλο είναι αρκετά συντηρητικό.

```
In [14]: y_pred_a_svm = svm_model_a.predict(X_test_a_svm_scaled)
...: y_prob_a_svm = svm_model_a.predict_proba(X_test_a_svm_scaled)[: , 1]

In [15]: from sklearn.metrics import confusion_matrix, roc_auc_score
...:
...: cm = confusion_matrix(y_test_a_svm, y_pred_a_svm)
...: tn, fp, fn, tp = cm.ravel()
...:
...: recall = tp / (tp + fn)
...: precision = tp / (tp + fp) if (tp + fp) > 0 else 0
...: f1 = 2 * (precision * recall) / (precision + recall) if (precision +
...:     : recall) > 0 else 0
...: fpr = fp / (fp + tn)
...: roc_auc = roc_auc_score(y_test_a_svm, y_prob_a_svm)
...:
...: print("Confusion Matrix:")
...: print(cm)
...:
...: print("Recall:", recall)
...: print("F1:", f1)
...: print("ROC-AUC:", roc_auc)
...: print("FPR:", fpr)
Confusion Matrix:
[[10195658  369113]
 [ 278298   5578]]
Recall: 0.019649424396567517
F1: 0.01693981022432038
ROC-AUC: 0.49332307865302705
FPR: 0.03493809756974382
```

Εικόνα 7.6 Αποτελέσματα SVM Σενάριο Α

7.2.2.2 Σενάριο Β

Στο δεύτερο σενάριο, το SVM παρουσίασε βελτιωμένη απόδοση:

- Recall: 0.319
- F1-score: 0.0012
- ROC-AUC: 0.521
- FPR: 0.044

Το Recall αυξήθηκε πράγμα που δείχνει ότι το μοντέλο κατάφερε να ανιχνεύσει ένα μέρος των zero-day επιθέσεων. Η απόδοση παραμένει χαμηλή, καθώς το ROC-AUC είναι ελαφρώς υψηλότερο από τυχαία ταξινόμηση και το F1-score παραμένει μικρό.

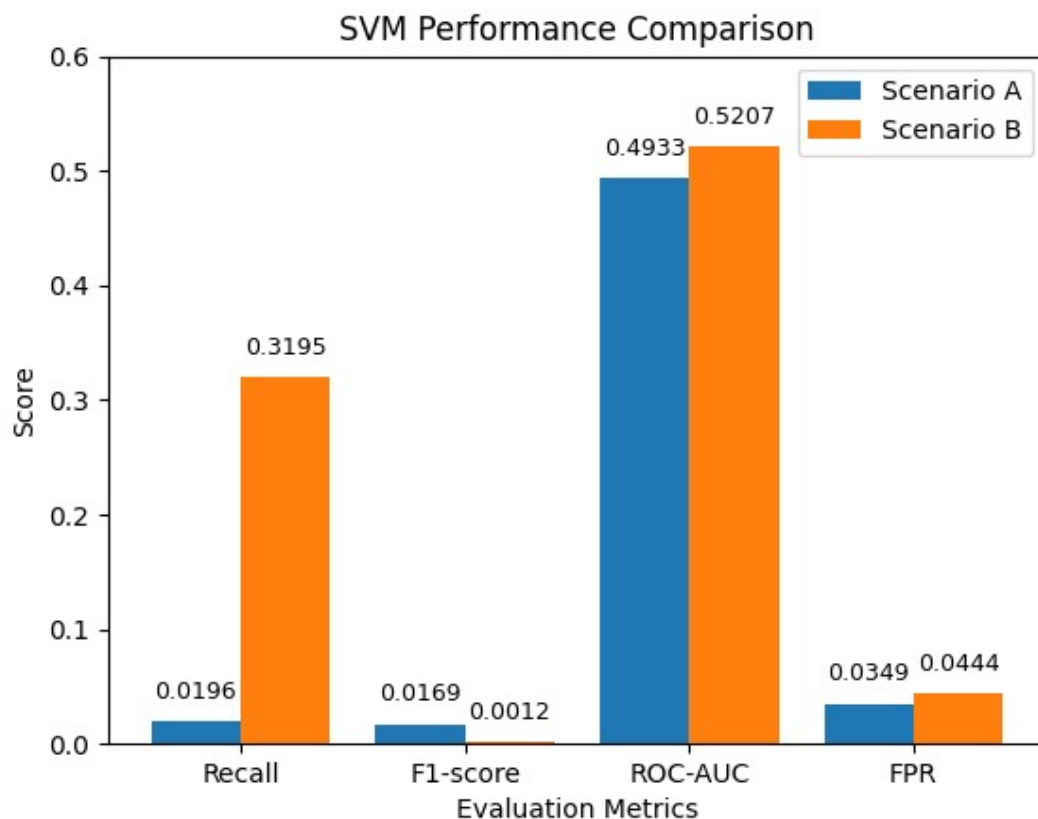
```
Confusion Matrix:
[[10095581  469190]
 [      590      277]]
Recall: 0.31949250288350634
F1: 0.0011778863531022635
ROC-AUC: 0.5206892609880772
FPR: 0.04441080644341463
```

Εικόνα 7.7 Αποτελέσματα SVM
Σενάριο B

7.2.3 Συζήτηση αποτελεσμάτων

Με την σύγκριση των αποτελεσμάτων φαίνεται ότι το μοντέλο SVM παρουσιάζει περιορισμένη ικανότητα ανίχνευσης σε συνθήκες zero-day. Εμφανίζει καλύτερη συμπεριφορά στο Σενάριο B σε σχέση με το Σενάριο A, παρόλα αυτά η συνολική του απόδοση παραμένει χαμηλή.

Σε σύγκριση με το Logistic Regression, το SVM είναι πιο συντηρητικό, έχοντας χαμηλότερο False Positive Rate, αλλά και χαμηλότερη ικανότητα ανίχνευσης επιθέσεων. Το γεγονός αυτό αναδεικνύει το πρόβλημα μεταξύ ανίχνευσης επιθέσεων και ψευδών συναγερμών στα IDS.



Εικόνα 7.8 SVM Σύγκριση Αποτελεσμάτων

7.2.4 Συμπεράσματα

Το μοντέλο SVM δεν κατάφερε να αποδώσει ικανοποιητικά στην ανίχνευση zero-day επιθέσεων. Η απόδοσή του χαρακτηρίζεται είτε από πολύ χαμηλή ικανότητα ανίχνευσης είτε από περιορισμένη διάκριση μεταξύ καλοήθους και κακόβουλης δραστηριότητας.

Ωστόσο, το SVM μπορεί να χρησιμοποιεί διαφορετικούς πυρήνες και τιμές παραμέτρων. Πριν την απόρριψη του ως μοντέλο για χρήση σε συστήματα ανίχνευσης εισβολών απαιτείται περαιτέρω έρευνα.

7.3 Random Forest

7.3.1 Διαδικασία εκπαίδευσης

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν τα σενάρια A και B. Σε αντίθεση με άλλα μοντέλα δεν χρειάστηκε κανονικοποίηση των δεδομένων καθώς το Random Forest δεν επηρεάζεται από την κλίμακα των χαρακτηριστικών. Επιπλέον οι υπερπαραμέτροι δεν άλλαξαν για την αντικειμενική σύγκριση με τα υπόλοιπα μοντέλα.

```
In [1]: import pandas as pd

In [2]: train_df_a_balanced= pd.read_csv("data/processed/scenario_a/train_df_
      : a_balanced.csv", low_memory= False)

In [3]: test_df_a= pd.read_csv("data/processed/scenario_a/test_scenario_a.csv
      : ", low_memory= False)

In [4]: X_train_a = train_df_a_balanced.drop(columns=["Label", "BinaryLabel"]
      : )
      ...: y_train_a = train_df_a_balanced["BinaryLabel"]

In [5]: X_test_a = test_df_a.drop(columns=["Label", "BinaryLabel"])
      ...: y_test_a = test_df_a["BinaryLabel"]

In [6]: from sklearn.ensemble import RandomForestClassifier
      ...:
      ...: rf_model_a = RandomForestClassifier(
      ...:     n_estimators=100,
      ...:     max_depth=None,
      ...:     n_jobs=-1,
      ...:     random_state=42
      ...: )

In [7]: rf_model_a.fit(X_train_a, y_train_a)
Out[7]: RandomForestClassifier(n_jobs=-1, random_state=42)
```

Εικόνα 7.9 Εκπαίδευση μοντέλου Random Forest

7.3.2 Αποτελέσματα

Η απόδοση του μοντέλου αξιολογήθηκε με τα μέτρα Recall, F1-score, ROC-AUC και FPR.

7.3.2.1 Σενάριο A

- Recall ≈ 0.00001
- F1-score ≈ 0.00002
- FPR ≈ 0.00003
- ROC-AUC= 0.746

```
In [7]: rf_model_a.fit(X_train_a, y_train_a)
Out[7]: RandomForestClassifier(n_jobs=-1, random_state=42)

In [8]: y_pred_a = rf_model_a.predict(X_test_a)
...: y_prob_a = rf_model_a.predict_proba(X_test_a)[:, 1]

In [9]: from sklearn.metrics import confusion_matrix, roc_auc_score
...:
...: cm = confusion_matrix(y_test_a, y_pred_a)
...: tn, fp, fn, tp = cm.ravel()
...:
...: recall = tp / (tp + fn)
...: precision = tp / (tp + fp) if (tp + fp) > 0 else 0
...: f1 = 2 * (precision * recall) / (precision + recall) if (precision +
...: : recall) > 0 else 0
...: fpr = fp / (fp + tn)
...: roc_auc = roc_auc_score(y_test_a, y_prob_a)
...:
...: print("Recall:", recall)
...: print("F1:", f1)
...: print("ROC-AUC:", roc_auc)
...: print("FPR:", fpr)
Recall: 1.0567994476461553e-05
F1: 2.110869924677125e-05
ROC-AUC: 0.7468056836995456
FPR: 3.445413061958465e-05
```

Εικόνα 7.10 Αποτελέσματα Σεναρίου A

7.3.2.2 Σενάριο B

- Recall ≈ 0.01
- F1-score ≈ 0.00001
- FPR ≈ 0.015
- ROC-AUC= 0.803

```
Confusion Matrix:
[[10403910  160861]
 [      857        10]]
Recall: 0.011534025374855825
F1: 0.00012365677824629955
ROC-AUC: 0.8030818808579254
FPR: 0.015226170070321449
```

Εικόνα 7.11 Αποτελέσματα Σεναρίου B

Παρατηρείται ότι παρά το πολύ χαμηλό Recall και F1-score, το ROC-AUC παραμένει υψηλό και στα δύο σενάρια. Το γεγονός αυτό υποδηλώνει ότι το μοντέλο έχει την ικανότητα να διαχωρίζει τις δύο κλάσεις αλλά δεν πραγματοποιεί θετικές προβλέψεις λόγω του αυστηρού κατωφλίου απόφασης (0.5).

7.3.3 Ανάλυση κατωφλίου απόφασης

Για την διερεύνηση της συμπεριφοράς του μοντέλου και την τελική αξιολόγηση του πραγματοποιήθηκε ανάλυση για διαφορετικές τιμές του κατωφλίου απόφασης.

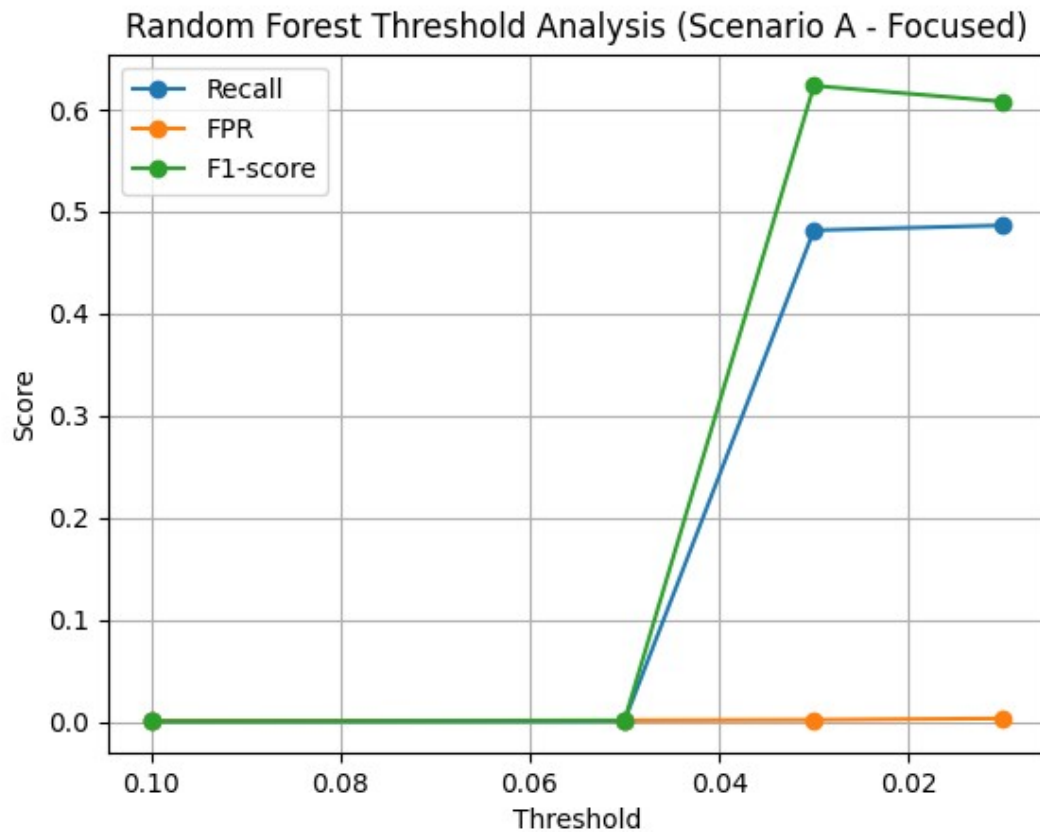
```
In [18]: thresholds = [0.5, 0.3, 0.2, 0.1, 0.05, 0.03, 0.01]
...:
...: for t in thresholds:
...:     y_pred_t = (y_prob_a > t).astype(int)
...:
...:     cm = confusion_matrix(y_test_a, y_pred_t)
...:     tn, fp, fn, tp = cm.ravel()
...:
...:     recall = tp / (tp + fn)
...:     precision = tp / (tp + fp) if (tp + fp) > 0 else 0
...:     f1 = 2 * (precision * recall) / (precision + recall) if (precision + recall) > 0 else 0
...:     fpr = fp / (fp + tn)
...:
...:     print("Threshold:", t)
...:     print("Recall:", recall)
...:     print("F1:", f1)
...:     print("FPR:", fpr)
...:     print("Confusion Matrix:")
...:     print(cm)
...:     print("-" * 40)
```

Εικόνα 7.12 Κώδικας για την Ανάλυση Κατωφλίου Απόφασης

7.3.3.1 Σενάριο A

Στο πρώτο σενάριο παρατηρήθηκε απότομη μεταβολή της απόδοσης του μοντέλου στην τιμή κατωφλίου 0.03:

- Recall = 0.48
- F1-score= 0.62
- FPR= 0.0017

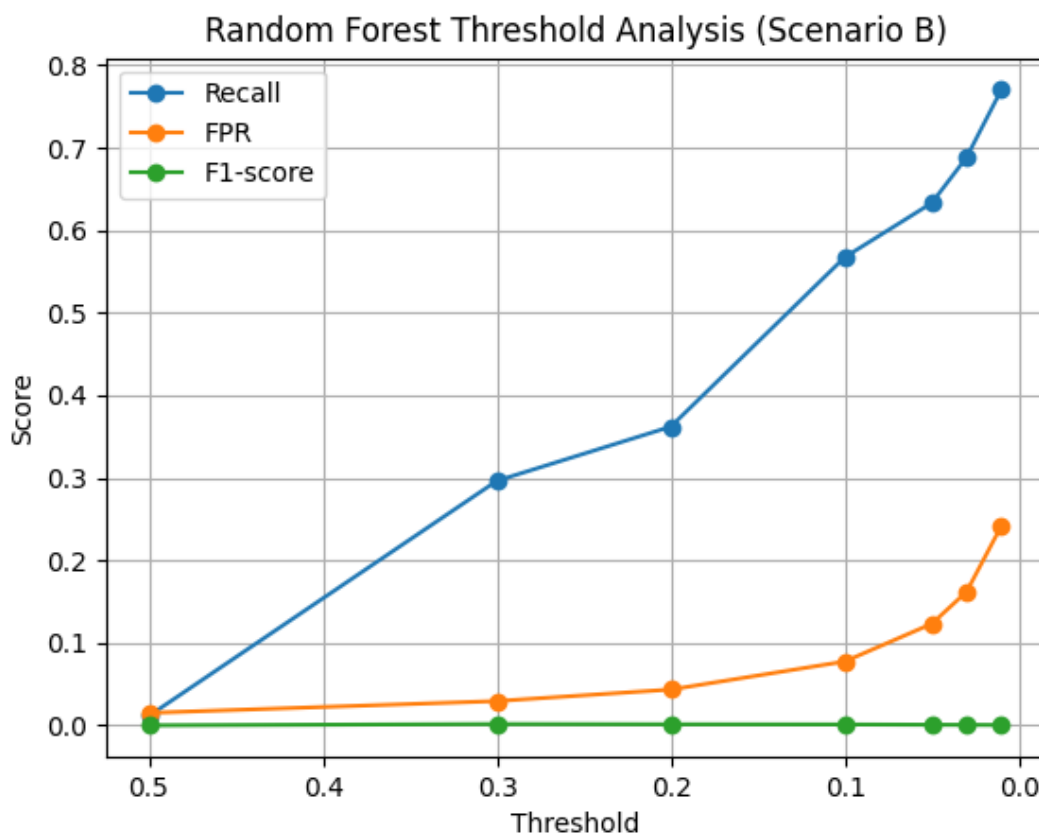


Εικόνα 7.13 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση το κατώφλι απόφασης στο Σενάριο Α
Το αποτέλεσμα αυτό δείχνει ότι το μοντέλο είναι σε θέση να ανιχνεύσει ένα μέρος των επιθέσεων zero-day διατηρώντας ταυτόχρονα χαμηλό ποσοστό ψευδών συναγερμών. Η συμπεριφορά αυτή υποδηλώνει διαχωρισμό μεταξύ καλοήθους και κακόβουλης δραστηριότητας.

7.3.3.2 Σενάριο Β

Στο δεύτερο σενάριο η βελτίωση της απόδοσης ήταν πιο σταδιακή. Για threshold =0.1 :

- Recall =0.57
- F1-score= 0.001
- FPR =0.07

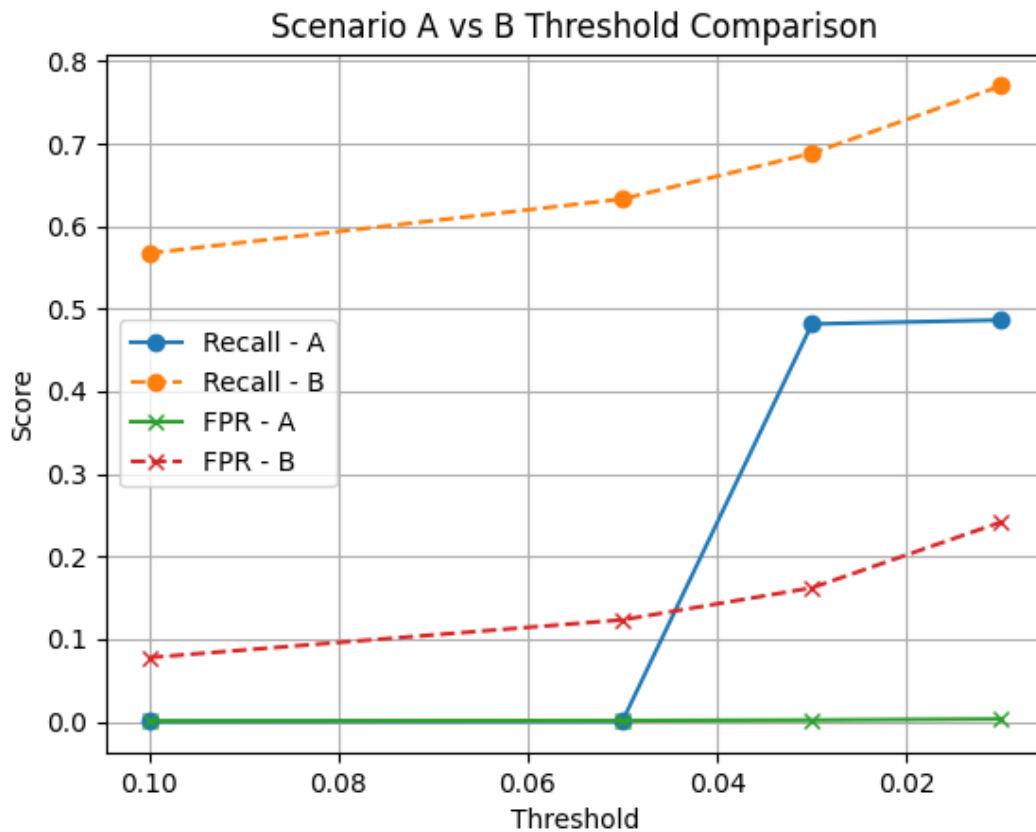


Εικόνα 7.14 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση το κατώφλι απόφασης στο Σενάριο B. Παρατηρείται ότι η αύξηση του Recall συνοδεύεται από σημαντική αύξηση του False Positive Rate, γεγονός που δείχνει την ύπαρξη μεγαλύτερης επικάλυψης μεταξύ των κλάσεων.

7.3.4 Συζήτηση αποτελεσμάτων

Η ανάλυση των αποτελεσμάτων αναδεικνύει τη μεγάλη ευαισθησία του Random Forest στην επιλογή του κατωφλίου απόφασης. Η αξιολόγηση με σταθερό threshold (0.5) οδηγεί σε παραπλανητικά συμπεράσματα, καθώς το μοντέλο εμφανίζεται να αποτυγχάνει ενώ στην πραγματικότητα διαθέτει υψηλή διαχωριστική ικανότητα.

Επιπλέον, παρατηρείται σημαντική διάφορα μεταξύ των δύο σεναρίων. Στο σενάριο A, το μοντέλο επιτυγχάνει ισχυρή ισορροπία μεταξύ Recall και FPR, γεγονός που δείχνει αποτελεσματική γενίκευση σε άγνωστες επιθέσεις. Αντίθετα, στο σενάριο B, η βελτίωση του Recall συνοδεύεται από αύξηση των ψευδών θετικών, γεγονός που καταδεικνύει μεγαλύτερη δυσκολία διάκρισης.



Εικόνα 7.15 Σύγκριση των δύο Σεναρίων

Τα αποτελέσματα αυτά υπογραμμίζουν ότι η απόδοση του μοντέλου δεν εξαρτάται μόνο από τον αλγόριθμο, αλλά και από τη φύση των δεδομένων και των επιθέσεων.

7.3.5 Συμπεράσματα

Το Random Forest παρουσίασε υψηλή ικανότητα διάκρισης μεταξύ καλοήθους και κακόβουλης δραστηριότητας το οποίο φαίνεται από την τιμή ROC-AUC και στα δύο σενάρια. Σε αντίθεση με τα υπόλοιπα μοντέλα, το Random Forest κατάφερε να συλλάβει πιο πολύπλοκα και μη γραμμικά μοτίβα στα δεδομένα, γεγονός που το καθιστά ιδιαίτερα κατάλληλο για προβλήματα ανίχνευσης εισβολών.

Συνολικά, τα αποτελέσματα δείχνουν ότι η απόδοση του Random Forest δεν εξαρτάται μόνο από τον ίδιο τον αλγόριθμο, αλλά σε μεγάλο βαθμό από τη φύση των δεδομένων και των επιθέσεων που καλείται να ανιχνεύσει. Επιπλέον, αναδεικνύεται η κρίσιμη σημασία της επιλογής του κατωφλίου απόφασης, το οποίο μπορεί να μεταβάλλει δραστικά τη συμπεριφορά του μοντέλου.

7.4 K-Means ομαδοποίηση

7.4.1 Διαδικασία εκπαίδευσης

Το σενάριο A και το σενάριο B χρησιμοποιήθηκαν για την εκπαίδευση του μοντέλου K-Means. Εφαρμόστηκε κανονικοποίηση Min-Max, καθώς το K-Means είναι ένας αλγόριθμος που βασίζεται στην απόσταση.

Ο αλγόριθμος ρυθμίστηκε αρχικά σε δύο συστάδες ($K=2$), προκειμένου να επιχειρηθεί η ταξινόμηση της καλοήθους και της κακόβουλης δραστηριότητας. Στη συνέχεια πραγματοποιήθηκε πείραμα με μεγαλύτερο αριθμό συστάδων για την σωστή αξιολόγηση του μοντέλου. Σε αντίθεση με τους αλγόριθμους εποπτευόμενης μάθησης, ο K-Means δεν χρησιμοποιεί ετικέτες κατά τη διάρκεια της εκπαίδευσης. Ως εκ τούτου, μετά την ομαδοποίηση, κάθε ομάδα αντιστοιχίστηκε στην κατηγορία που εμφανιζόταν πιο συχνά εντός της συγκεκριμένης ομάδας στα δεδομένα εκπαίδευσης.

```
In [2]: train_df_a_balanced= pd.read_csv("data/processed/scenario_a/train_df_
      : a_balanced.csv", low_memory= False)

In [3]: test_df_a= pd.read_csv("data/processed/scenario_a/test_scenario_a.csv
      : ", low_memory= False)

In [4]: X_train_a = train_df_a_balanced.drop(columns=["Label", "BinaryLabel"]
      : )
      ...: y_train_a = train_df_a_balanced["BinaryLabel"]

In [5]: X_test_a = test_df_a.drop(columns=["Label", "BinaryLabel"])
      ...: y_test_a = test_df_a["BinaryLabel"]

In [6]: from sklearn.preprocessing import MinMaxScaler
      ...:
      ...: scaler_km_a = MinMaxScaler()
      ...:
      ...: X_train_a_scaled = scaler_km_a.fit_transform(X_train_a)
      ...: X_test_a_scaled = scaler_km_a.transform(X_test_a)

In [7]: from sklearn.cluster import KMeans
      ...:
      ...: kmeans_a = KMeans(n_clusters=2, random_state=42, n_init=10)
      ...:
      ...: kmeans_a.fit(X_train_a_scaled)
Out[7]: KMeans(n_clusters=2, n_init=10, random_state=42)

In [8]: train_clusters = kmeans_a.predict(X_train_a_scaled)

In [9]: cluster_df = pd.DataFrame({
      ...:     "cluster": train_clusters,
      ...:     "label": y_train_a
      ...: })
      ...:
      ...: mapping = {}

In [10]: for c in cluster_df["cluster"].unique():
      ...:     majority_label = cluster_df[cluster_df["cluster"] == c]["label"]
      ...:     .mode()[0]
      ...:     mapping[c] = majority_label
      ...:
      ...: print(mapping)
{np.int32(1): np.int64(0), np.int32(0): np.int64(0)}

In [11]: test_clusters = kmeans_a.predict(X_test_a_scaled)
      ...:
      ...: y_pred_a = [mapping[c] for c in test_clusters]
```

Εικόνα 7.16 Εκπαίδευση Μοντέλου K-means

7.4.2 Αποτελέσματα

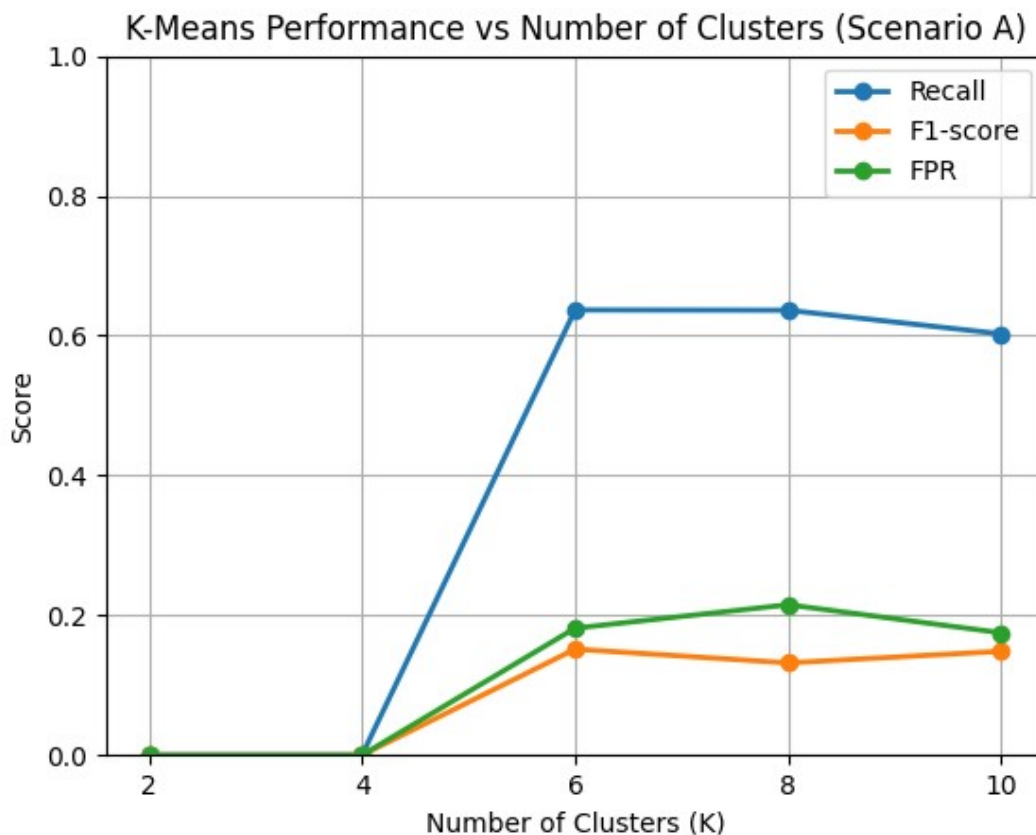
Η απόδοση του μοντέλου αξιολογήθηκε με τα μέτρα Recall, F1-score, και FPR.

Σε αντίθεση με τα μοντέλα εποπτευόμενης μηχανικής μάθησης, το K-Means δεν παράγει βαθμολογίες πιθανότητας ή εκτίμησης εμπιστοσύνης για κάθε πρόβλεψη. Ως αποτέλεσμα, δεν ήταν δυνατό να υπολογιστεί ο δείκτης ROC-AUC.

Και στα δύο σενάρια το μοντέλο δεν κατάφερε να ανιχνεύσει καμία επίθεση και χαρακτηρίστηκαν όλες ως καλοήγη δραστηριότητα πράγμα που οδήγησε σε μηδενικό recall και F1-score. Για την σωστή αξιολόγηση του μοντέλου ακολουθεί επιπλέον πείραμα στο οποίο το μοντέλο θα έχει αριθμό συστάδων 4,6,8,10.

7.4.3 Ανάλυση συστάδων

Στο σενάριο A, παρατηρήθηκε ξαφνική βελτίωση στην απόδοση όταν ο αριθμός των συστάδων αυξήθηκε από $K=4$ σε $K=6$. Αυτή η συμπεριφορά υποδηλώνει ότι οι επιθέσεις zero-day δεν σχηματίζουν μία ενιαία συστάδα, αλλά αποτελούνται από πολλαπλές υποομάδες με διαφορετικά χαρακτηριστικά.



Εικόνα 7.17 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση τον αριθμό συστάδων στο Σενάριο A

Στο σενάριο B, η αύξηση του αριθμού των συστάδων βελτίωσε επίσης σημαντικά το Recall. Ωστόσο, το FPR παρέμεινε υψηλό σε όλες τις διαμορφώσεις, υποδηλώνοντας σημαντική επικάλυψη μεταξύ της καλοήθους και της κακόβουλης δραστηριότητας.



Εικόνα 7.18 Αλλαγή τιμών των μέτρων αξιολόγησης με βάση τον αριθμό συστάδων στο Σενάριο B

Επιπλέον, η απόδοση του μοντέλου σταθεροποιήθηκε μετά το $K \geq 6$, υποδηλώνοντας ότι οι περιορισμοί της μεθόδου K-Means σχετίζονται κυρίως με τη δομή των δεδομένων και όχι με την επιλογή παραμέτρων.

7.4.4 Συζήτηση αποτελεσμάτων

Τα αποτελέσματα παρουσιάζουν σημαντικούς περιορισμούς. Η μέθοδος K-Means βασίζεται αποκλειστικά στη γεωμετρική ομοιότητα μεταξύ των σημείων δεδομένων και προϋποθέτει την ύπαρξη σαφώς διαχωρίσιμων ομάδων.

Ωστόσο, τα δεδομένα δικτυακής κίνησης είναι εξαιρετικά πολύπλοκα και παρουσιάζουν σημαντική επικάλυψη μεταξύ της καλοήθους και της κακόβουλης δραστηριότητας, καθώς και σημαντική ποικιλομορφία μεταξύ των τύπων επιθέσεων. Ως αποτέλεσμα, η κακόβουλη δραστηριότητα δεν σχηματίζει διακριτές ομάδες στον χώρο των χαρακτηριστικών.

Η αύξηση του αριθμού των συστάδων βελτίωσε το Recall και στα δύο σενάρια, πράγμα που σημαίνει ότι οι επιθέσεις αποτελούνται από πολλαπλές υποομάδες. Ωστόσο, αυτή η βελτίωση συνοδεύτηκε από υψηλό FPR, δείχνοντας ότι τα καλοήθη και τα κακόβουλα δείγματα εξακολουθούν να επικαλύπτονται σημαντικά.

Τα αποτελέσματα δείχνουν επίσης ότι η απόδοση του μοντέλου εξαρτάται όχι μόνο από τον αριθμό των συστάδων, αλλά και από τη φύση των επιθέσεων και την κατανομή των δεδομένων.

7.4.5 Συμπεράσματα

Χρησιμοποιήθηκε το K-Means ως βασικό μοντέλο μη εποπτευόμενης μηχανικής μάθησης που λειτουργεί με ομαδοποίηση, με σκοπό να διερευνηθεί η δυνατότητα ανίχνευσης επιθέσεων zero-day.

Η αξιολόγηση έδειξε ότι η απόδοση του μοντέλου εξαρτάται σε μεγάλο βαθμό από τον αριθμό των συστάδων. Μικρές τιμές του K οδήγησαν σε πλήρη αποτυχία ανίχνευσης επιθέσεων, ενώ ο μεγαλύτερος αριθμός συστάδων βελτίωσε σημαντικά την ευαισθησία του μοντέλου.

Ωστόσο, η αύξηση του Recall συνοδεύτηκε από υψηλά ποσοστά ψευδών θετικών ταξινομήσεων και χαμηλές βαθμολογίες F1, περιορίζοντας την πρακτική εφαρμογή του μοντέλου σε πραγματικά IDS.

Συνολικά, τα αποτελέσματα δείχνουν ότι οι απλές προσεγγίσεις ομαδοποίησης είναι ανεπαρκείς για την αξιόπιστη ανίχνευση εισβολών zero-day. Αν και το K-Means κατάφερε να αποκαλύψει ενδιαφέροντα δομικά μοτίβα μέσα στα δεδομένα και να προσδιορίσει υποομάδες σχετικές με επιθέσεις, απέτυχε να παρέχει σταθερό και αξιόπιστο διαχωρισμό μεταξύ καλοήθους και κακόβουλης δραστηριότητας.

7.5 Isolation Forest

7.5.1 Διαδικασία εκπαίδευσης

Το μοντέλο Isolation Forest εκπαιδεύτηκε χρησιμοποιώντας τα Σεναρίου Α και Β. Σε αντίθεση με αλγόριθμους ομαδοποίησης όπως το K-Means, το Isolation Forest έχει σχεδιαστεί ειδικά για την ανίχνευση ανωμαλιών. Ο αλγόριθμος απομονώνει τις παρατηρήσεις διαχωρίζοντας αναδρομικά τον χώρο χαρακτηριστικών χρησιμοποιώντας τυχαία επιλεγμένα χαρακτηριστικά και τιμές διαχωρισμού. Τα δείγματα που απομονώνονται πιο γρήγορα θεωρούνται ανωμαλίες.

Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκε μόνο καλοήθης δραστηριότητα. Το Isolation Forest προσπαθεί να ανιχνεύσει ανωμαλίες στην κίνηση του δικτύου μαθαίνοντας την καλοήγη συμπεριφορά από τα δεδομένα και αναγνωρίζοντας τις αποκλίσεις ως πιθανές επιθέσεις.

Αν και το Isolation Forest είναι λιγότερο ευαίσθητο στην κλιμάκωση χαρακτηριστικών σε σύγκριση με αλγόριθμους που βασίζονται στην απόσταση, εφαρμόστηκε κανονικοποίηση Min-Max για να εξασφαλιστεί η συνέπεια σε όλα τα πειράματα και τα μοντέλα.

Το μοντέλο αρχικά εκπαιδεύτηκε με τιμή contamination 0.01 και μετέπειτα με άλλες τιμές ώστε να διερευνηθεί η απόδοση του.

```

In [2]: df_train_a= pd.read_csv("data/processed/scenario_a/train_scenario_a.c
      : sv", low_memory= False)

In [3]: test_df_a= pd.read_csv("data/processed/scenario_a/test_scenario_a.csv
      : ", low_memory= False)

In [4]: train_benign_a = df_train_a[
      : df_train_a["BinaryLabel"] == 0
      : ]

In [5]: X_train_a = train_benign_a.drop(
      : columns=["Label", "BinaryLabel"])

In [6]: X_test_a = test_df_a.drop(
      : columns=["Label", "BinaryLabel"])

In [7]: y_test_a = test_df_a["BinaryLabel"]

In [8]: from sklearn.preprocessing import MinMaxScaler
      : scaler_if_a = MinMaxScaler()
      : X_train_a_scaled = scaler_if_a.fit_transform(X_train_a)
      : X_test_a_scaled = scaler_if_a.transform(X_test_a)

In [9]: from sklearn.ensemble import IsolationForest
      : iso_a = IsolationForest(
      :     n_estimators=100,
      :     contamination=0.01,
      :     random_state=42,
      :     n_jobs=-1
      : )
      : iso_a.fit(X_train_a_scaled)
Out[9]: IsolationForest(contamination=0.01, n_jobs=-1, random_state=42)

In [10]: y_pred_raw = iso_a.predict(X_test_a_scaled)

```

Εικόνα 7.19 Εκπαίδευση Μοντέλου Isolation Forest

7.5.2 Αποτελέσματα

Η απόδοση του μοντέλου αξιολογήθηκε με τα μέτρα Recall, F1-score, ROC-AUC και FPR.

7.5.2.1 Σενάριο Α

Για την αρχική αξιολόγηση του μοντέλου Isolation Forest, η παράμετρος contamination ορίστηκε στο 0,01. Με αυτή τη ρύθμιση, το μοντέλο έδειξε πολύ περιορισμένη ικανότητα ανίχνευσης επιθέσεων «zero-day».

Τα αποτελέσματα ήταν τα εξής:

- Recall: 0,004
- F1-score: 0.006
- FPR: 0.01

- ROC-AUC: 0.51

Η πολύ χαμηλή τιμή Recall δείχνει ότι το μοντέλο εντόπισε μόνο ένα ελάχιστο ποσοστό των κακόβουλων εγγραφών. Αυτό μπορεί να εξηγηθεί από το ότι οι επιθέσεις στο σενάριο A δεν διαφοροποιήθηκαν επαρκώς από την καλοήγη δικτυακή δραστηριότητα στον χώρο των χαρακτηριστικών. Επειδή το Isolation Forest βασίζεται στην υπόθεση ότι οι ανωμαλίες είναι σπάνιες και απομονώνονται ευκολότερα από τις καλοήγους παρατηρήσεις, όταν η κακόβουλη δραστηριότητα επικαλύπτεται με την καλοήγη οι κακόβουλες εγγραφές δεν απομονώνονται αποτελεσματικά και ταξινομούνται ως καλοήγη.

Παράλληλα, το χαμηλό FPR δείχνει ότι το μοντέλο ήταν αρκετά συντηρητικό ταξινομώντας μόνο ένα μικρό ποσοστό εγγραφών ως ανωμαλίες. Ωστόσο αυτό δεν οδήγησε σε αποτελεσματική ανίχνευση επιθέσεων.

Επιπλέον, η τιμή ROC-AUC είναι πολύ κοντά στην τυχαία συμπεριφορά, υποδηλώνοντας ότι το μοντέλο δεν ήταν σε θέση να διακρίνει αποτελεσματικά μεταξύ καλοήθους και κακόβουλης δραστηριότητας στο σενάριο A.

```
In [11]: import numpy as np
...: y_pred_a = np.where(y_pred_raw == -1, 1, 0)

In [12]: from sklearn.metrics import confusion_matrix
...:
...: cm = confusion_matrix(y_test_a, y_pred_a)
...:
...: tn, fp, fn, tp = cm.ravel()
...:
...: recall = tp / (tp + fn)
...:
...: precision = tp / (tp + fp) if (tp + fp) > 0 else 0
...:
...: f1 = (
...:     2 * (precision * recall) / (precision + recall)
...:     if (precision + recall) > 0 else 0
...: )
...: fpr = fp / (fp + tn)
...: print("Confusion Matrix:")
...: print(cm)
...: print("Recall:", recall)
...: print("F1:", f1)
...: print("FPR:", fpr)
Confusion Matrix:
[[10459124  105647]
 [ 282679   1197]]
Recall: 0.00421662979610816
F1: 0.006127149877149877
FPR: 0.00999993279551445

In [13]: from sklearn.metrics import roc_auc_score
...:
...: scores_a = -iso_a.decision_function(X_test_a_scaled)
...:
...: roc_auc = roc_auc_score(y_test_a, scores_a)
...:
...: print("ROC-AUC:", roc_auc)
ROC-AUC: 0.505988365173824
```

7.5.2.2 Σενάριο B

- Recall: 0.01
- F1-score: 0.0002
- FPR: 0.01
- ROC-AUC: 0.58

Σε σύγκριση με το σενάριο A, το μοντέλο πέτυχε υψηλότερη τιμή ROC-AUC, πράγμα που δείχνει ελαφρώς βελτιωμένη ικανότητα διαχωρισμού της καλοήθους από την κακόβουλη δραστηριότητα. Αυτό υποδηλώνει ότι οι επιθέσεις που εμφανίζονται στο σενάριο B παρουσίαζαν ισχυρότερα ανώμαλα χαρακτηριστικά σε σχέση με την καλοήγη δραστηριότητα.

Ωστόσο, το Recall και το F1-score παρέμειναν πολύ χαμηλά, δείχνοντας ότι το μοντέλο εξακολουθούσε να μην ανιχνεύει της περισσότερες επιθέσεις. Παρόμοια με το σενάριο A, το χαμηλό FPR σημαίνει ότι το μοντέλο συμπεριφέρθηκε συντηρητικά, χαρακτηρίζοντας μόνο ένα μικρό ποσοστό δειγμάτων ως ανωμαλίες.

7.5.3 Ανάλυση Contamination

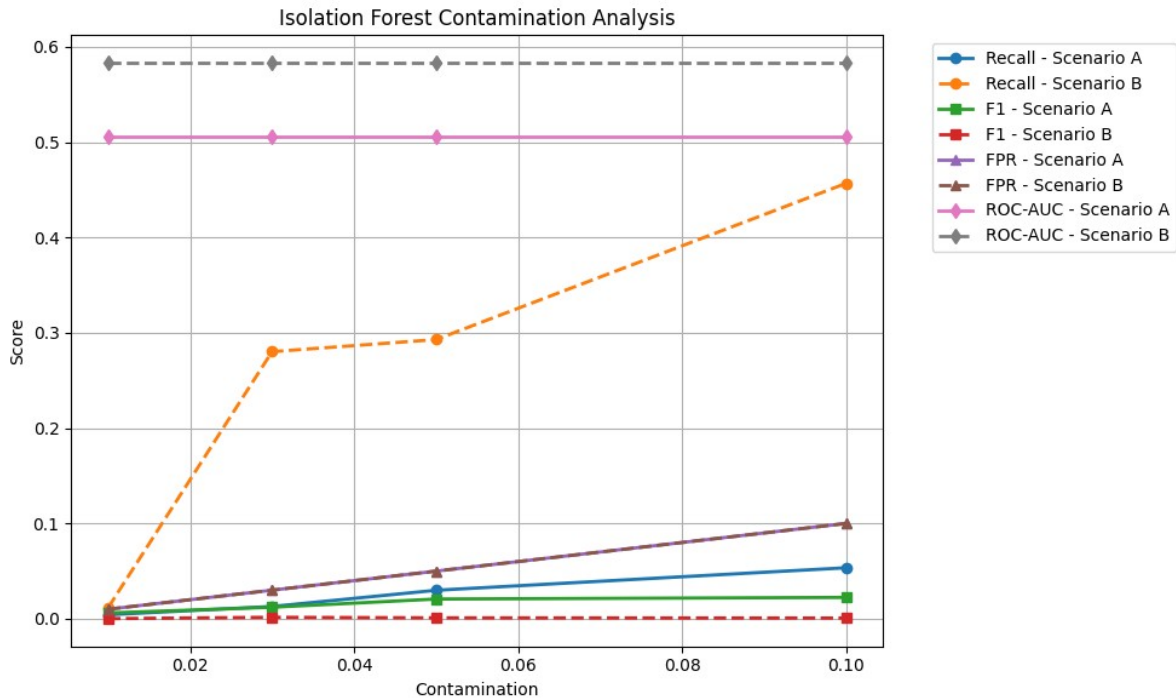
Προκειμένου να διερευνηθεί περαιτέρω η συμπεριφορά του μοντέλου, διεξήχθησαν πειράματα χρησιμοποιώντας διαφορετικές τιμές contamination.

Και στα δύο σενάρια, η αύξηση της παραμέτρου είχε ως αποτέλεσμα υψηλότερες τιμές Recall. Αυτή η συμπεριφορά παρατηρείται επειδή οι μεγαλύτερες τιμές μόλυνσης αναγκάζουν το μοντέλο να ταξινομήσει ένα μεγαλύτερο ποσοστό δειγμάτων ως ανωμαλίες.

Ωστόσο, η αύξηση του Recall συνοδεύονταν σταθερά από μια ανάλογη αύξηση του FPR. Ταυτόχρονα, η ROC-AUC παρέμεινε σταθερή σε όλα τα πειράματα και στα δύο σενάρια.

Αυτή η παρατήρηση είναι ιδιαίτερα σημαντική, καθώς υποδηλώνει ότι η αλλαγή της παραμέτρου δεν βελτιώνει την ικανότητα διάκρισης του μοντέλου. Αντίθετα, αλλάζει μόνο το σημείο λειτουργίας του ταξινομητή τροποποιώντας το όριο ανωμαλίας.

Ως αποτέλεσμα, η ευαισθησία του μοντέλου μπορεί να αυξηθεί, χωρίς όμως να βελτιωθεί η απόδοση του.



Εικόνα 7.21 Ανάλυση Contamination στα δύο Σενάρια Isolation Forest

7.5.4 Συζήτηση αποτελεσμάτων

Τα αποτελέσματα αναδεικνύουν διάφορους σημαντικούς περιορισμούς των προσεγγίσεων ανίχνευσης εισβολών που βασίζονται σε ανωμαλίες.

Το μοντέλο Isolation Forest υποθέτει ότι οι επιθέσεις συμπεριφέρονται ως στατιστικές ανωμαλίες σε σχέση με την καλοήγη δραστηριότητα. Στην πραγματικότητα, τα αποτελέσματα δείχνουν ότι αυτή η υπόθεση δεν ισχύει πάντα σε σενάρια ανίχνευσης εισβολών zero-day.

Στο σενάριο A, η τιμή ROC-AUC παρέμεινε κοντά στην τυχαία συμπεριφορά (≈ 0.5), δείχνοντας ότι οι επιθέσεις δεν αποκλίνουν σημαντικά από τη καλοήγη δραστηριότητα στον χώρο των χαρακτηριστικών. Ως αποτέλεσμα, το μοντέλο δεν μπόρεσε να απομονώσει αποτελεσματικά τα κακόβουλα δείγματα.

Στο σενάριο B, το μοντέλο πέτυχε καλύτερες τιμές ROC-AUC, υποδηλώνοντας ότι οι επιθέσεις παρουσίαζαν ισχυρότερα χαρακτηριστικά ανωμαλίας. Παρόλα αυτά, το FPR παρέμεινε υψηλό, περιορίζοντας την πρακτική εφαρμογή του μοντέλου.

Συνολικά, τα αποτελέσματα δείχνουν ότι οι επιθέσεις zero-day δεν αποτελούν απαραίτητα στατιστικές ανωμαλίες και, ως εκ τούτου, δεν μπορούν πάντα να ανιχνευθούν αποτελεσματικά χρησιμοποιώντας τεχνικές που βασίζονται αποκλειστικά σε ανωμαλίες.

7.5.5 Συμπεράσματα

Το Isolation Forest χρησιμοποιήθηκε ως μοντέλο ανίχνευσης ανωμαλιών χωρίς επίβλεψη για τον εντοπισμό επιθέσεων zero-day.

Η αξιολόγηση έδειξε ότι το μοντέλο παρουσίασε περιορισμένη απόδοση και στα δύο σενάρια. Η αύξηση της παραμέτρου contamination βελτίωσε το Recall, αλλά αύξησε αναλογικά και FPR. Επιπλέον, η

καμπύλη ROC-AUC παρέμεινε σε μεγάλο βαθμό αμετάβλητη σε όλα τα πειράματα, υποδηλώνοντας ότι η ρύθμιση της παραμέτρου contamination από μόνη της δεν μπορεί να βελτιώσει την ικανότητα διάκρισης του μοντέλου.

Το σενάριο B παρήγαγε καλύτερα αποτελέσματα από το σενάριο A, δείχνοντας ότι οι επιθέσεις στο σενάριο B παρουσίαζαν πιο διακριτή ανώμαλη συμπεριφορά. Παρ' όλα αυτά, η συνολική απόδοση παρέμεινε ανεπαρκής για αξιόπιστη ανίχνευση εισβολών στον πραγματικό κόσμο.

Πέρα από τη ρύθμιση της παραμέτρου contamination πιθανές βελτιώσεις περιλαμβάνουν την δοκιμή διαφορετικού $n_estimators$ (αριθμός δέντρων), $max_samples$ (αριθμός εγγράφων σε κάθε δέντρο) και την επιλογή χαρακτηριστικών που διαχωρίζουν καλύτερα την καλοήγη από την κακόβουλη δισκτυακή δραστηριότητα.

Συνολικά, το Isolation Forest παρείχε χρήσιμες πληροφορίες, αλλά δεν κατάφερε να επιτύχει σταθερή και αξιόπιστη απόδοση ανίχνευσης στο συγκεκριμένο πείραμα.

7.6 One-Class Support-Vector-Machines

7.6.1 Διαδικασία Εκπαίδευσης

Για την εκπαίδευση του μοντέλου One-Class SVM (OCSVM) χρησιμοποιήθηκε τα σενάρια A και B. Το One-Class SVM είναι ένας αλγόριθμος ανίχνευσης ανωμαλιών μη εποπτευόμενης μηχανικής μάθησης, ο οποίος μαθαίνει τα όρια των καλοηθών δεδομένων και αναγνωρίζει ως ανωμαλίες τα δεδομένα που βρίσκονται εκτός αυτών των ορίων.

Για τη διαδικασία εκπαίδευσης χρησιμοποιήθηκαν μόνο καλοήγη δεδομένα. Αυτή η προσέγγιση είναι παρόμοια με την παραδοσιακή προσέγγιση ανίχνευσης ανωμαλιών, η οποία εκπαιδεύει το μοντέλο με τη φυσιολογική λειτουργία του συστήματος και ανιχνεύει τις ανωμαλίες ως πιθανές επιθέσεις.

Δεδομένου ότι το One-Class SVM είναι ευαίσθητο σε προβλήματα κλιμάκωσης χαρακτηριστικών, πριν από την εκπαίδευση πραγματοποιήθηκε κανονικοποίηση όλων των χαρακτηριστικών χρησιμοποιώντας τη μέθοδο Min-Max. Κατά τη διάρκεια της εκπαίδευσης, χρησιμοποιήθηκε ένα υποσύνολο 50.000 καλοηθών εγγράφων λόγω της υψηλής υπολογιστικής πολυπλοκότητας του αλγορίθμου.

Επιπλέον, δοκιμάστηκαν διάφορες τιμές της παραμέτρου ν (0,01, 0,03, 0,05, 0,10) για να εξεταστεί η επίδραση του ορίου της ανωμαλίας στη συμπεριφορά του μοντέλου. Η παράμετρος ν χρησιμοποιείται για τη ρύθμιση του αριθμού των δειγμάτων που βρίσκονται εκτός του ορίου κανονικότητας που έχει μάθει το μοντέλο, ουσιαστικά αποτελεί ένα κατώφλι ανωμαλίας.

```

In [2]: train_df_a= pd.read_csv("data/processed/scenario_a/train_scenario_a.c
: sv", low_memory= False)

In [3]: test_df_a= pd.read_csv("data/processed/scenario_a/test_scenario_a.csv
: ", low_memory= False)

In [4]: train_benign_a = train_df_a[
....:     train_df_a["BinaryLabel"] == 0
....: ]

In [5]: train_benign_a_sample = train_benign_a.sample(
....:     n=50000,
....:     random_state=42
....: )

In [6]: X_train_a = train_benign_a_sample.drop(
....:     columns=["Label", "BinaryLabel"]
....: )
....:
....: X_test_a = test_df_a.drop(
....:     columns=["Label", "BinaryLabel"]
....: )
....:
....: y_test_a = test_df_a["BinaryLabel"]

In [7]: from sklearn.preprocessing import MinMaxScaler
....:
....: scaler_ocsvm_a = MinMaxScaler()
....:
....: X_train_a_scaled = scaler_ocsvm_a.fit_transform(X_train_a)
....:
....: X_test_a_scaled = scaler_ocsvm_a.transform(X_test_a)

In [8]: from sklearn.svm import OneClassSVM
....: ocsvm_a = OneClassSVM(
....:     kernel="rbf",
....:     gamma="scale",
....:     nu=0.01)
....: ocsvm_a.fit(X_train_a_scaled)
Out[8]: OneClassSVM(nu=0.01)

In [9]: y_pred_raw = ocsvm_a.predict(X_test_a_scaled)

In [10]: import numpy as np
....: y_pred_a = np.where(y_pred_raw == -1, 1, 0)

```

Εικόνα 7.22 Εκπαίδευση Μοντέλου One-Class SVM

7.6.2 Αποτελέσματα

Η απόδοση του μοντέλου αξιολογήθηκε με τα μέτρα Recall, F1-score, ROC-AUC και FPR.

7.6.2.1 Σενάριο Α

Αρχικά το μοντέλο εκπαιδεύτηκε με $\text{nu} = 0.01$ και τα αποτελέσματα είναι τα εξής:

- Recall=0.013
- F1-score= 0.018
- FPR= 0.01
- ROC-AUC= 0.498

Τα αποτελέσματα μας δείχνουν ότι το μοντέλο δεν κατάφερε να διακρίνει αποτελεσματικά την καλοήγη από τη κακόβουλη δραστηριότητα. Όσον αφορά το FPR, παρέμεινε σχετικά χαμηλό, αλλά το Recall είναι αρκετά μικρό, καθώς εντοπίστηκε μόνο ένας μικρός αριθμός επιθέσεων.

Επιπλέον, η τιμή ROC-AUC δεν απέκλινε σημαντικά από την τυχαία απόδοση, γεγονός που υποδηλώνει χαμηλή διαχωριστικότητα των ανωμαλιών στο σενάριο A.

```
In [11]: from sklearn.metrics import confusion_matrix, roc_auc_score
...:
...: cm = confusion_matrix(y_test_a, y_pred_a)
...:
...: tn, fp, fn, tp = cm.ravel()
...:
...: recall = tp / (tp + fn)
...:
...: precision = tp / (tp + fp) if (tp + fp) > 0 else 0
...:
...: f1 = (
...:     2 * (precision * recall) / (precision + recall)
...:     if (precision + recall) > 0 else 0
...: )
...: scores_a = -ocsvm_a.decision_function(X_test_a_scaled)
...: roc_auc = roc_auc_score(y_test_a, scores_a)
...:
...: fpr = fp / (fp + tn)
...:
...: print("Confusion Matrix:")
...: print(cm)
...:
...: print("Recall:", recall)
...: print("F1:", f1)
...: print("ROC-AUC:", roc_auc)
...: print("FPR:", fpr)
Confusion Matrix:
[[10453431  111340]
 [ 280111   3765]]
Recall: 0.01326283306795925
F1: 0.018873079169183495
ROC-AUC: 0.4984599459826535
FPR: 0.010538799184572956
```

Εικόνα 7.23 Αποτελέσματα One-Class SVM Σενάριο A

7.6.2.2 Σενάριο B

Η τιμή του nu είναι ίδια με το σενάριο A. Τα αποτελέσματα είναι τα παρακάτω:

- Recall= 0.444

- F1-score= 0.006
- FPR= 0.01
- ROC-AUC= 0.596

Το μοντέλο παρουσίασε καλύτερη απόδοση όσον αφορά τις τιμές Recall και ROC-AUC σε σύγκριση με το σενάριο A. Τα αποτελέσματα υποδεικνύουν ότι οι επιθέσεις που περιλαμβάνονται στο σενάριο B παρουσίαζαν περισσότερες ανώμαλες συμπεριφορές σε σύγκριση με την καλοήγη δραστηριότητα.

Επιπλέον, το μοντέλο παρουσίασε σχετικά καλή ισορροπία μεταξύ Recall και FPR με την προεπιλεγμένη τιμή ν . Τα αποτελέσματα υπογραμμίζουν ότι ένα σημαντικό ποσοστό των επιθέσεων εντοπίστηκε σωστά από το μοντέλο με ένα σχετικά συντηρητικό όριο ανωμαλίας, παρόλο που η βαθμολογία F1 ήταν χαμηλή.

```
Confusion Matrix:
[[10453431  111340]
 [      482     385]]
Recall: 0.44405997693194926
F1: 0.0068388517834304385
FPR: 0.010538799184572956
ROC-AUC: 0.5960482441814356
```

Εικόνα 7.24 Αποτελέσματα One-Class SVM Σενάριο B

7.6.3 Ανάλυση ν

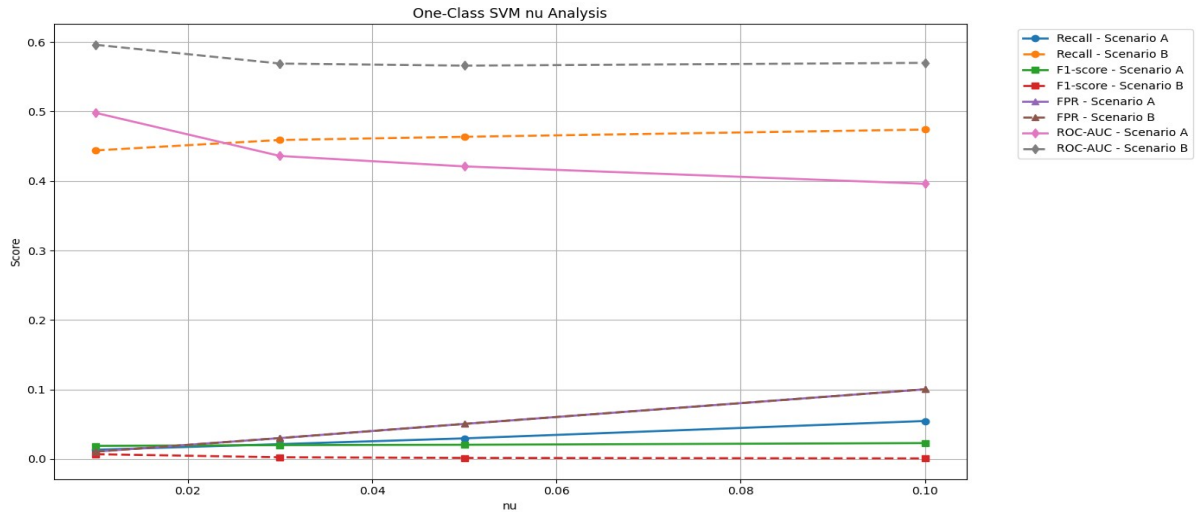
Για να κατανοήσουμε καλύτερα τη συμπεριφορά του μοντέλου, πραγματοποιήθηκαν πειράματα με διαφορετικές τιμές της παραμέτρου ν .

Το Recall αυξήθηκε καθώς αυξανόταν η τιμή του ν και στις δύο περιπτώσεις. Αυτή η συμπεριφορά οφείλεται στο γεγονός ότι όσο μεγαλύτερη είναι η τιμή του ν , τόσο λιγότερο περιορίζεται το όριο των ανωμαλιών και τόσο περισσότερα δείγματα απαιτούνται να αναγνωριστούν ως ανωμαλίες. Ωστόσο, η αύξηση του Recall συνοδεύονταν πάντα από αύξηση του FPR.

Μία από τις βασικές διαφορές μεταξύ των δύο σεναρίων ήταν ότι η συμπεριφορά του ROC-AUC ήταν διαφορετική. Το ROC-AUC μειωνόταν προοδευτικά με την αύξηση του ν έως ότου έπεσε κάτω από την τυχαία συμπεριφορά στο σενάριο A. Αυτό υποδηλώνει ότι η ευαισθησία του μοντέλου δεν αύξησε την ικανότητα του μοντέλου να διακρίνει τις ανωμαλίες, αλλά οδήγησε σε όλο και πιο ασταθείς ταξινομήσεις ανωμαλιών.

Οι τιμές ROC-AUC για το Σενάριο B ήταν πιο υψηλές για όλες τις διαμορφώσεις ν από το Σενάριο A, υποδηλώνοντας ότι οι επιθέσεις είχαν μεγαλύτερη διαχωριστικότητα ανωμαλιών από την καλοήγη δραστηριότητα. Παρόλα αυτά παρατηρήθηκε πτώση στις τιμές του ROC-AUC με την αύξηση του ν .

Οι παρατηρήσεις υποδηλώνουν ότι η απόδοση του One-Class SVM επηρεάζεται σε μεγάλο βαθμό από τη δομή των δεδομένων και το βαθμό αλληλεπικάλυψης μεταξύ της καλοήθους και της κακόβουλης δραστηριότητας.

Εικόνα 7.25 Ανάλυση ν στα δύο Σενάρια One-Class SVM

7.6.4 Συζήτηση αποτελεσμάτων

Τα πειραματικά αποτελέσματα αναδεικνύουν τα πλεονεκτήματα και τα μειονεκτήματα της τεχνικής ανίχνευσης ανωμαλιών με βάση τα όρια. Σε αντίθεση με το Isolation Forest, το One-Class SVM επιχειρεί να εντοπίσει ένα μαθηματικό όριο που διαχωρίζει την καλοήγη από την κακόβουλη δραστηριότητα. Ως εκ τούτου, το μοντέλο είναι ιδιαίτερα ευαίσθητο στον βαθμό επικάλυψης και στην κατανομή των χαρακτηριστικών μεταξύ των δεδομένων.

Στο σενάριο A, το μοντέλο δεν έμαθε ένα ουσιαστικό όριο διαχωρισμού και οι τιμές ROC-AUC ήταν κοντά στις τυχαίες, ενώ η ικανότητα ανίχνευσης επιθέσεων ήταν μικρή. Η διακριτική ικανότητα του μοντέλου μειώθηκε, ενώ το Recall αυξήθηκε με την αύξηση της τιμής του ν .

Τα αποτελέσματα του μοντέλου στο Σενάριο B ήταν σημαντικά καλύτερα, με Recall άνω του 44% και σχετικά χαμηλά ποσοστά ψευδών θετικών για μικρές τιμές ν . Τα αποτελέσματα έδειξαν ότι οι επιθέσεις στο σενάριο B ήταν πιο εμφανώς ανώμαλες σε σχέση με τη καλοήγη δραστηριότητα, γεγονός που οδήγησε το μοντέλο στη δημιουργία ενός καλύτερου ορίου καλοήθους δραστηριότητας.

7.6.5 Συμπεράσματα

Τα αποτελέσματα έδειξαν ότι η απόδοση του μοντέλου διέφερε σημαντικά μεταξύ των δύο σεναρίων. Το μοντέλο δεν κατάφερε να επιτύχει σημαντική διάκριση μεταξύ της καλοήθους και της κακόβουλης δραστηριότητας στο σενάριο A, αλλά παρουσίασε σημαντικά καλύτερη απόδοση στην ανίχνευση ανωμαλιών στο σενάριο B.

Με την αύξηση του ν αυξήθηκε και το Recall αλλά και το FPR και μείωσε την απόδοση ROC-AUC, υποδηλώνοντας ότι οι αδυναμίες του μοντέλου προέρχονται περισσότερο από τα δεδομένα και λιγότερο από την έλλειψη πληροφοριών εκπαίδευσης.

Τα συνολικά αποτελέσματα δείχνουν ότι το One-Class SVM μπορεί να αποδώσει σημαντική ανίχνευση ανωμαλιών σε καταστάσεις όπου οι επιθέσεις είναι εξαιρετικά ανώμαλες σε σχέση με την καλοήγη δραστηριότητα. Ωστόσο, είναι ευαίσθητο στην επικάλυψη δεδομένων και στην επιλογή παραμέτρων, γεγονός που το καθιστά λιγότερο αξιόπιστο για τα πραγματικά IDS.

7.7 Συνολικά αποτελέσματα

Αποτελέσματα Α σεναρίου

Μοντέλο	Recall	F1-score	FPR	ROC-AUC
Logistic Regression	0.039	0.022	0.066	0.483
SVM	0.020	0.017	0.035	0.493
Random Forest (threshold = 0.03)	0.48	0.62	0.0017	0.746
K-means (k=6)	0.63	0.15	0.18	N/A
Isolation Forest (contamination = 0.1)	0.05	0.022	0.1	0.506
One-class SVM (nu= 0.01)	0.013	0.018	0.01	0.498

Αποτελέσματα Β σεναρίου

Μοντέλο	Recall	F1-score	FPR	ROC-AUC
Logistic Regression	0.532	0.0008	0.106	0.696
SVM	0.319	0.0012	0.044	0.521
Random Forest (threshold =0.1)	0.57	0.001	0.077	0.803
K-means (k=6)	0.82	0.0007	0.18	N/A
Isolation Forest (contamination =0.1)	0.46	0.0007	0.1	0.583
One-class SVM (nu= 0.01)	0.44	0.0068	0.01	0.596

Συνολικά, τα αποτελέσματα δείχνουν ότι τα μοντέλα εποπτευόμενης μηχανικής μάθησης είναι πιο αποτελεσματικά στην ανίχνευση επιθέσεων μηδενικής ημέρας στο συγκεκριμένο σύνολο δεδομένων. Παρότι τα μοντέλα μη εποπτευόμενης μηχανικής μάθησης έχουν το πλεονέκτημα ότι μπορούν να εκπαιδεύονται αποκλειστικά σε καλοήγη δικτυακή κίνηση, η απόδοσή τους εξαρτάται σε μεγάλο βαθμό από το κατά πόσο η συμπεριφορά των επιθέσεων διαφοροποιείται από την κανονική δικτυακή δραστηριότητα. Σύμφωνα με τα αποτελέσματα της παρούσας μελέτης, το μοντέλο Random Forest παρουσίασε τη μεγαλύτερη αξιοπιστία και τη σταθερότερη συνολική απόδοση, επιτυγχάνοντας την καλύτερη ισορροπία μεταξύ ικανότητας ανίχνευσης επιθέσεων και FPR.

Κεφάλαιο 8ο: Συμπεράσματα και προτάσεις

8.1 Συμπεράσματα

Σκοπός της παρούσας διατριβής ήταν η διερεύνηση και η αξιολόγηση μεθόδων μηχανικής μάθησης για την ανίχνευση εισβολών τύπου zero day, με δεδομένα δικτυακής δραστηριότητας. Η μελέτη εξέτασε την ικανότητα μεθόδων εποπτευόμενης και μη εποπτευόμενης μάθησης να ανιχνεύουν επιθέσεις που δεν είχαν παρατηρηθεί κατά τη διάρκεια της εκπαίδευσης.

Για τον σκοπό αυτό, εφαρμόστηκαν και δοκιμάστηκαν διάφορα μοντέλα μηχανικής μάθησης και στα δύο σενάρια zero day. Τα μοντέλα που χρησιμοποιήθηκαν ήταν η Logistic Regression, το Support Vector Machines (SVM), το Random Forest, ο K-Means Clustering, το Isolation Forest και το One-Class SVM. Αξιολογήθηκαν με βάση τους ακόλουθους δείκτες: Recall, F1-score, False Positive Rate (FPR) και ROC-AUC.

Τα πειραματικά αποτελέσματα κατέδειξαν σημαντικές διαφορές μεταξύ των εποπτευόμενων και των μη εποπτευόμενων μοντέλων.

Το μοντέλο Random Forest παρουσίασε την υψηλότερη συνολική απόδοση μεταξύ όλων των μοντέλων που αξιολογήθηκαν. Αρχικά, το μοντέλο κατέγραψε πολύ χαμηλές τιμές ανάκλησης (Recall), ωστόσο, η ρύθμιση του Threshold απόδειξε ότι το μοντέλο διέθετε καλή ικανότητα διάκρισης. Το Random Forest παρουσίασε μια επιθυμητή σχέση μεταξύ Recall και FPR καθώς και υψηλό ROC-AUC. Αυτό σήμαινε ότι το μοντέλο μπορούσε να μάθει περίπλοκες και μη γραμμικές συσχετίσεις με τα δεδομένα και να κάνει καλές προβλέψεις για επιθέσεις που δεν είχε συναντήσει ποτέ.

Μέτρια απόδοση είχε το μοντέλο Support Vector Machine. Το γραμμικό SVM είχε χαμηλές τιμές Recall. Ταυτόχρονα το FPR δεν ήταν υψηλό. Αυτό οδηγεί στο συμπέρασμα ότι το SVM είναι ένα συντηρητικό μοντέλο που δεν παράγει πολλούς λάθος συναγεμμούς αλλά δεν καταφέρνει να αναγνωρίσει τις επιθέσεις zero day.

Το Logistic Regression απέδωσε σταθερά αλλά περιορισμένα αποτελέσματα σε σύγκριση με πιο σύνθετα μοντέλα. Το μοντέλο κατάφερε να εντοπίσει μέρος της κακόβουλης κυκλοφορίας, αλλά αντιμετώπισε δυσκολίες στην ανίχνευση πιο σύνθετων σχέσεων εντός των δεδομένων.

Η ομαδοποίηση K-Means έδειξε ότι η αύξηση του αριθμού των ομάδων βελτίωσε την απόδοση ανίχνευσης επιθέσεων και στα δύο σενάρια. Ωστόσο, οι βελτιώσεις συνοδεύονταν από υψηλά FPR, περιορίζοντας την πρακτική εφαρμογή του μοντέλου. Τα αποτελέσματα έδειξαν ότι η καλοήθης και η κακόβουλη δραστηριότητα επικαλύπτονται σημαντικά στον χώρο των χαρακτηριστικών και δεν σχηματίζουν φυσικά ομάδες που να διαχωρίζονται σαφώς.

Το Isolation Forest είχε χαμηλή συνολική απόδοση. Για το σενάριο A, το ROC-AUC δεν διέφερε σημαντικά από τυχαία κατανομή, πράγμα που σημαίνει ότι οι επιθέσεις δεν είχαν αρκετά ισχυρά ανώμαλα χαρακτηριστικά. Στο σενάριο B, το μοντέλο είχε ελαφρώς καλύτερη απόδοση, υποδηλώνοντας ότι οι επιθέσεις απέκλιναν λίγο περισσότερο από τη συνήθη κυκλοφορία. Ωστόσο, η αύξηση της παραμέτρου μόλυνσης βελτίωσε το Recall μόνο αυξάνοντας τον αριθμό των δειγμάτων που ταξινομήθηκαν ως ανωμαλίες, χωρίς να βελτιώσει την εγγενή ικανότητα διάκρισης του μοντέλου.

Μεταξύ των μη εποπτευόμενων μεθόδων, το One-Class SVM απέδωσε τα πιο ενδιαφέροντα αποτελέσματα. Η δυνατότητα διαχωρισμού μεταξύ καλοήθους και κακόβουλης κυκλοφορίας δεν ήταν

σημαντική στο σενάριο A. Οι τιμές Recall και ROC-AUC για το μοντέλο ήταν υψηλότερες στο σενάριο B, ενώ το FPR βρισκόταν σε σχετικά χαμηλά επίπεδα για μικρές τιμές της παραμέτρου ν . Τα αποτελέσματα δείχνουν ότι το One-Class SVM είναι σε θέση να ανιχνεύσει εν μέρει την παρουσία μιας επίθεσης zero day όταν η κακόβουλη κίνηση παρουσιάζει επαρκή ανώμαλη συμπεριφορά.

Ένα από τα βασικά συμπεράσματα αυτής της εργασίας είναι ότι οι επιθέσεις zero day δεν αποτελούν απαραίτητα στατιστικές ακραίες τιμές. Ωστόσο, πολλές επιθέσεις είναι παρόμοιες με τη καλοήγη δραστηριότητα και είναι δύσκολο να ανιχνευθούν μόνο με μεθόδους που βασίζονται σε ανωμαλίες. Αυτή η παρατήρηση αποτελεί τον λόγο για την επιτυχία της εποπτευόμενης μάθησης έναντι των μη εποπτευόμενων μοντέλων ανίχνευσης ανωμαλιών σε όλα τα πειράματα.

Ένα άλλο σημαντικό αποτέλεσμα αφορά τη ρύθμιση των παραμέτρων. Η ρύθμιση του threshold στα εποπτευόμενα μοντέλα και η ρύθμιση των παραμέτρων contamination/ ν στα μοντέλα ανίχνευσης ανωμαλιών είχαν σημαντική επίδραση στους δείκτες Recall και FPR. Ακόμη και αν είναι δυνατή η ρύθμιση των παραμέτρων, αυτό δεν αρκεί για την αντιμετώπιση της χαμηλής διαχωρισιμότητας μεταξύ της καλοήθους και της κακόβουλης κυκλοφορίας.

Συνολικά, τα πειραματικά αποτελέσματα δείχνουν ότι τα εποπτευόμενα μοντέλα μηχανικής μάθησης, ιδίως το μοντέλο Random Forest, προσφέρουν σημαντικά πιο αξιόπιστες επιδόσεις στην ανίχνευση εισβολών zero-day, ξεπερνώντας την καθαρά μη εποπτευόμενη προσέγγιση ανίχνευσης ανωμαλιών.

8.2 Περιορισμοί της Δ.Ε

Παρά τα θετικά αποτελέσματα, πρέπει να επισημανθούν ορισμένοι περιορισμοί.

Πρώτον, τα πειράματα διεξήχθησαν χρησιμοποιώντας ένα συγκεκριμένο σύνολο δεδομένων. Διαφορετικά σύνολα δεδομένων ή περιβάλλοντα δικτύου ενδέχεται να οδηγήσουν σε διαφορετικά αποτελέσματα.

Δεύτερον, η αξιολόγηση επικεντρώθηκε κυρίως στη δυαδική ταξινόμηση μεταξύ καλοήθους και κακόβουλης δραστηριότητας. Η κατηγοριοποίηση επιθέσεων σε πολλαπλές κατηγορίες δεν εμπίπτει στο πεδίο της παρούσας εργασίας.

Επιπλέον, ορισμένα μοντέλα, όπως το One-Class SVM, απαιτούσαν υποσύνολα εκπαίδευσης μειωμένου μεγέθους λόγω περιορισμών στην υπολογιστική πολυπλοκότητα. Πειράματα μεγαλύτερης κλίμακας θα μπορούσαν ενδεχομένως να οδηγήσουν σε διαφορετικά αποτελέσματα.

Ένας άλλος περιορισμός αφορά τη μηχανική χαρακτηριστικών. Τα πειράματα βασίστηκαν στα χαρακτηριστικά του παρεχόμενου συνόλου δεδομένων, χωρίς εκτενή χειροκίνητη επιλογή χαρακτηριστικών ή τεχνικές εξόρυξης χαρακτηριστικών βαθιάς μάθησης.

Τέλος, τα πειράματα διεξήχθησαν σε περιβάλλον εκτός σύνδεσης. Η ανίχνευση εισβολών σε πραγματικό χρόνο ενέχει πρόσθετες προκλήσεις, όπως η καθυστέρηση, η επεξεργασία δεδομένων ροής και η μετατόπιση εννοιών.

8.3 Μελλοντικές προτάσεις

Από τα ευρήματα της παρούσας διατριβής προκύπτουν διάφορες κατευθύνσεις για μελλοντική έρευνα.

Μια πιθανή επέκταση είναι η διερεύνηση αρχιτεκτονικών βαθιάς μάθησης για την ανίχνευση εισβολών. Μοντέλα όπως οι Autoencoders, τα δίκτυα LSTM και οι αρχιτεκτονικές που βασίζονται σε Transformer ενδέχεται να καταγράφουν πιο σύνθετες σχέσεις εντός της δικτυακής κίνησης.

Μια άλλη σημαντική κατεύθυνση είναι η ανάπτυξη υβριδικών IDS που συνδυάζουν εποπτευόμενες και βασισμένες σε ανωμαλίες προσεγγίσεις. Τέτοια συστήματα θα μπορούσαν ενδεχομένως να αξιοποιήσουν τα πλεονεκτήματα που προσφέρουν και οι δύο μέθοδοι: θα μπορούσαν να έχουν υψηλή ικανότητα διάκρισης και την ικανότητα να ανιχνεύουν επιθέσεις που δεν έχουν ξαναδεί.

Ιδιαίτερη έμφαση θα μπορούσε να δοθεί στη βελτίωση της επιλογής χαρακτηριστικών για τα μοντέλα ανίχνευσης ανωμαλιών, όπως το Isolation Forest και το One-Class SVM. Η απόδοση των συγκεκριμένων μοντέλων εξαρτάται σημαντικά από τον χώρο χαρακτηριστικών στον οποίο εκπαιδεύονται. Χαρακτηριστικά με περιορισμένη διακριτική ικανότητα ή αυξημένο θόρυβο μπορεί να δυσκολεύουν τον διαχωρισμό της καλοήθους από την κακόβουλη δικτυακή δραστηριότητα. Για τον λόγο αυτό, μια μελλοντική μελέτη θα μπορούσε να εξετάσει τους συντελεστές του μοντέλου Logistic Regression, ώστε να εντοπιστούν τα χαρακτηριστικά που συνεισφέρουν περισσότερο στην ταξινόμηση της δικτυακής κίνησης. Η κατεύθυνση και το απόλυτο μέγεθος των συντελεστών μπορούν να παρέχουν ενδείξεις για τη σχέση κάθε χαρακτηριστικού με την καλοήθη ή την κακόβουλη κλάση. Στη συνέχεια, τα χαρακτηριστικά με τη μεγαλύτερη διακριτική συνεισφορά θα μπορούσαν να χρησιμοποιηθούν στην εκπαίδευση των μοντέλων ανίχνευσης ανωμαλιών. Με αυτόν τον τρόπο, θα μπορούσε να περιοριστεί ο θόρυβος στα δεδομένα και να βελτιωθεί η ικανότητα των μοντέλων να εντοπίζουν επιθέσεις που διαφοροποιούνται από την καλοήθη δικτυακή συμπεριφορά.

Ωστόσο, η ανάλυση των συντελεστών της Logistic Regression θα πρέπει να πραγματοποιείται μετά την κατάλληλη κανονικοποίηση των χαρακτηριστικών, ώστε τα μεγέθη τους να είναι συγκρίσιμα. Επίσης, θα πρέπει να λαμβάνονται υπόψη οι πιθανές συσχετίσεις μεταξύ των χαρακτηριστικών, καθώς μπορεί να επηρεάζουν τις τιμές των συντελεστών. Η πρόταση αυτή συνδέεται άμεσα με τα αποτελέσματα της εργασίας, στα οποία τα μοντέλα ανίχνευσης ανωμαλιών παρουσίασαν δυσκολίες όταν η κακόβουλη κίνηση εμφάνιζε ομοιότητες με την καλοήθη δικτυακή δραστηριότητα.

Επιπλέον, σε μελλοντική έρευνα θα μπορούσε να πραγματοποιηθεί συστηματικότερη διερεύνηση των υπερπαραμέτρων των μοντέλων, με στόχο τη βελτιστοποίηση της απόδοσής τους. Στα μοντέλα SVM και One-Class SVM θα μπορούσαν να εξεταστούν διαφορετικές συναρτήσεις πυρήνα καθώς και οι αντίστοιχες παράμετροι που καθορίζουν τη μορφή του ορίου απόφασης. Στο Isolation Forest θα μπορούσαν να μελετηθούν παράμετροι όπως ο αριθμός των δέντρων, ο αριθμός των δειγμάτων που χρησιμοποιούνται για την εκπαίδευση κάθε δέντρου και το ποσοστό των χαρακτηριστικών που επιλέγεται κατά την κατασκευή τους. Αντίστοιχα, στο Random Forest θα μπορούσαν να εξεταστούν το μέγιστο βάθος των δέντρων, ο ελάχιστος αριθμός δειγμάτων για τη δημιουργία νέου διαχωρισμού και ο αριθμός των χαρακτηριστικών που εξετάζονται σε κάθε κόμβο.

Τέλος, τα μοντέλα θα μπορούσαν να αξιολογηθούν σε πραγματικό χρόνο, μέσω συνεχούς επεξεργασίας δικτυακών ροών. Μια τέτοια αξιολόγηση θα επέτρεπε τη μελέτη παραμέτρων όπως ο χρόνος απόκρισης, η χρήση υπολογιστικών πόρων και η μεταβολή της δικτυακής συμπεριφοράς με την πάροδο του χρόνου. Με αυτόν τον τρόπο, θα μπορούσε να εξεταστεί πληρέστερα η πρακτική εφαρμογή των μοντέλων σε ένα πραγματικό σύστημα ανίχνευσης εισβολών.

8.4 Τελικές παρατηρήσεις

Η εξέλιξη των απειλών στον κυβερνοχώρο και η αυξανόμενη πολυπλοκότητα των επιθέσεων zero-day συνεχίζουν να αποτελούν σημαντικές προκλήσεις για τα σύγχρονα IDS.

Η παρούσα διατριβή απέδειξε ότι οι μέθοδοι μηχανικής μάθησης μπορούν να συμβάλουν θετικά στην ανάπτυξη IDS, ιδίως μέσω μεθόδων εποπτευόμενης μάθησης. Παράλληλα τα πειράματα έδειξαν τις

αδυναμίες των μοντέλων ανίχνευσης ανωμαλιών σε συνθήκες όπου η κακόβουλη δραστηριότητα επικαλύπτεται σε μεγάλο βαθμό τη καλοήθη.

Τα αποτελέσματα συνολικά υπογραμμίζουν την ανάγκη για ισχυρούς αλγόριθμους μηχανικής μάθησης, καθώς και για δεδομένα υψηλής ποιότητας, κατάλληλη αναπαράσταση χαρακτηριστικών και λεπτή ρύθμιση του μοντέλου, προκειμένου να είναι αποτελεσματική η ανίχνευση εισβολών zero-day.

Τα αποτελέσματα της ερευνητικής εργασίας θα είναι χρήσιμα για την κατανόηση της συμπεριφοράς διαφόρων τεχνικών μηχανικής μάθησης στον τομέα της ανίχνευσης εισβολών zero-day και θα αποτελέσουν αφετηρία για περαιτέρω έρευνα στον τομέα των συστημάτων κυβερνοασφάλειας.

ΒΙΒΛΙΟΓΡΑΦΙΑ

- [1] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Network anomaly detection: Methods, systems and tools,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 303–336, First Quarter 2014, doi: 10.1109/SURV.2013.052213.00046.
- [2] A. B. Nassif, M. A. Talib, Q. Nasir, and F. M. Dakalbab, “Machine learning for anomaly detection: A systematic review,” *IEEE Access*, vol. 9, pp. 78658–78700, 2021, doi: 10.1109/ACCESS.2021.3083060.
- [3] A. Tharwat, “Classification assessment methods,” *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, 2021, doi: 10.1016/j.aci.2018.08.003.
- [4] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, “Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure,” *Sensors*, vol. 23, no. 5, Art. no. 2415, 2023, doi: 10.3390/s23052415.
- [5] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proc. 2017 ACM Asia Conf. Comput. Commun. Security (ASIA CCS)*, 2017, pp. 506–519, doi: 10.1145/3052973.3053009.
- [6] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, “Survey of intrusion detection systems: Techniques, datasets and challenges,” *Cybersecurity*, vol. 2, Art. no. 20, 2019, doi: 10.1186/s42400-019-0038-7.
- [7] H. Satılmış, S. Akleylek, and Z. Y. Tok, “A Systematic Literature Review on Host-Based Intrusion Detection Systems,” *IEEE Access*, vol. 12, pp. 27237–27266, 2024, doi: 10.1109/ACCESS.2024.3367004.
- [8] A. L. Buczak and E. Guven, “A survey of data mining and machine learning methods for cyber security intrusion detection,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016, doi: 10.1109/COMST.2015.2494502.
- [9] L. Diana, P. Dini, and D. Paolini, “Overview on Intrusion Detection Systems for Computers Networking Security,” *Computers*, vol. 14, no. 3, Art. no. 87, 2025, doi: 10.3390/computers14030087.
- [10] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “A survey of network-based intrusion detection data sets,” *Computers & Security*, vol. 86, pp. 147–167, 2019, doi: 10.1016/j.cose.2019.06.005.
- [11] S. S. Bagui, M. P. Khan, C. Valmyr, S. C. Bagui, and D. Mink, “Model retraining upon concept drift detection in network traffic big data,” *Future Internet*, vol. 17, no. 8, Art. no. 328, 2025, doi: 10.3390/fi17080328.
- [12] J. R. Trillo, F. González-López, J. A. Morente-Molinera, R. Magán-Carrión, and P. García-Sánchez, “Evaluation of Explainable, Interpretable and Non-Interpretable Algorithms for Cyber Threat Detection,” *Electronics*, vol. 14, no. 15, Art. no. 3073, 2025, doi: 10.3390/electronics14153073.
- [13] H. M. R. U. Rehman, S. Liaquat, M. J. Gul, M. Z. Jhandir, D. Gavilanes, M. M. Vergara, and I. Ashraf, “A systematic literature study of machine learning techniques based intrusion detection: Datasets, models, challenges, and future directions,” *Journal of Big Data*, vol. 12, Art. no. 264, 2025, doi: 10.1186/s40537-025-01323-2.

- [14] N. Khan, K. Ahmad, A. Al Tamimi, M. M. Alani, A. Bermak, and I. Khalil, "Explainable AI-Based Intrusion Detection Systems for Industry 5.0 and Adversarial XAI: A Systematic Review," *Information*, vol. 16, no. 12, Art. no. 1036, 2025, doi: 10.3390/info16121036.
- [15] A. Drewek-Ossowicka, M. Pietrołaj, and J. Rumiński, "A Survey of Neural Networks Usage for Intrusion Detection Systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, pp. 497–514, 2021, doi: 10.1007/s12652-020-02014-x.
- [16] N. Mohamed, "Artificial Intelligence and Machine Learning in Cybersecurity: A Deep Dive into State-of-the-Art Techniques and Future Paradigms," *Knowledge and Information Systems*, vol. 67, pp. 6969–7055, 2025, doi: 10.1007/s10115-025-02429-y.
- [17] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine Learning in Network Anomaly Detection: A Survey," *IEEE Access*, vol. 9, pp. 152379–152396, 2021, doi: 10.1109/ACCESS.2021.3126834.
- [18] S. S. Bagui, G. C. S. D. Carvalho, A. Mishra, D. Mink, S. C. Bagui, and S. Eager, "Detecting Cyber Threats in UWF-ZeekDataFall22 Using K-Means Clustering in the Big Data Environment," *Future Internet*, vol. 17, no. 6, art. no. 267, 2025, doi: 10.3390/fi17060267.
- [19] D. Paolini, P. Dini, E. Soldaini, and S. Saponara, "One-Class Anomaly Detection for Industrial Applications: A Comparative Survey and Experimental Study," *Computers*, vol. 14, no. 7, art. no. 281, 2025, doi: 10.3390/computers14070281.
- [20] M. Rigaki and S. Garcia, "Bringing a GAN to a knife-fight: Adapting malware communication to avoid detection," in *Proc. 2018 IEEE Security and Privacy Workshops (SPW)*, 2018, pp. 70–75, doi: 10.1109/SPW.2018.00019.
- [21] M. Fatima, O. Rehman, I. M. H. Rahman, A. Ajmal, and S. J. Park, "Towards Ensemble Feature Selection for Lightweight Intrusion Detection in Resource-Constrained IoT Devices," *Future Internet*, vol. 16, no. 10, art. no. 368, 2024, doi: 10.3390/fi16100368.
- [22] W. Samek, T. Wiegand, and K.-R. Müller, "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models," *ITU Journal: ICT Discoveries*, vol. 1, no. 1, no. 5, pp. 39–48, 2017.
- [23] L. Bilge and T. Dumitras, "Before we knew it: An empirical study of zero-day attacks in the real world," in *Proc. 19th ACM Conf. Comput. Commun. Security (CCS '12)*, 2012, pp. 833–844, doi: 10.1145/2382196.2382284.
- [24] T. Zoppi, A. Ceccarelli, and A. Bondavalli, "Unsupervised Algorithms to Detect Zero-Day Attacks: Strategy and Application," *IEEE Access*, vol. 9, pp. 90603–90615, 2021, doi: 10.1109/ACCESS.2021.3090957.
- [25] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, May–Jun. 2011, doi: 10.1109/MSP.2011.67.
- [26] S. M. Sohi, J.-P. Seifert, and F. Ganji, "RNNIDS: Enhancing Network Intrusion Detection Systems through Deep Learning," *Computers & Security*, vol. 102, Art. no. 102151, 2021, doi: 10.1016/j.cose.2020.102151.
- [27] T. Diamond, A. Kerman, M. Souppaya, K. Stine, B. Johnson, C. Peloquin, V. Ruffin, M. Simos, S. Sweeney, and K. Scarfone, *Improving Enterprise Patching for General IT Systems: Utilizing Existing*

Tools and Performing Processes in Better Ways, NIST Special Publication 1800-31, Apr. 2022, doi: 10.6028/NIST.SP.1800-31.

[28] Symantec, Internet Security Threat Report, vol. 24, Feb. 2019. [Online]. Available: <https://docs.broadcom.com/doc/istr-24-2019-en>

[29] R. Sommer and V. Paxson, “Outside the closed world: On using machine learning for network intrusion detection,” in Proc. 2010 IEEE Symp. Security and Privacy, 2010, pp. 305–316, doi: 10.1109/SP.2010.25.

[30] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, “A deep learning approach to network intrusion detection,” IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 41–50, Feb. 2018, doi: 10.1109/TETCI.2017.2772792.

[31] M. Guarascio, N. Cassavia, F. S. Pisani, and G. Manco, “Boosting cyber-threat intelligence via collaborative intrusion detection,” Future Generation Computer Systems, vol. 135, pp. 30–43, 2022, doi: 10.1016/j.future.2022.04.028

[32] T. D. Wagner, K. Mahbub, E. Palomar, and A. E. Abdallah, “Cyber threat intelligence sharing: Survey and research directions,” Computers & Security, vol. 87, Art. no. 101589, 2019, doi: 10.1016/j.cose.2019.101589.

[33] W. Wang, S. Jian, Y. Tan, Q. Wu, and C. Huang, “Robust Unsupervised Network Intrusion Detection with Self-Supervised Masked Context Reconstruction,” Computers & Security, vol. 128, Art. no. 103131, 2023, doi: 10.1016/j.cose.2023.103131.

[34] A. Abdelkhalek and M. Mashaly, “Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning,” The Journal of Supercomputing, vol. 79, pp. 10611–10644, 2023, doi: 10.1007/s11227-023-05073-x.

[35] National Institute of Standards and Technology, “CVE-2021-44228 Detail,” National Vulnerability Database, Dec. 10, 2021.

[36] Cybersecurity and Infrastructure Security Agency, “Mitigating Log4Shell and Other Log4j-Related Vulnerabilities,” Cybersecurity Advisory AA21-356A, Dec. 2021. [Online]. Available: <https://www.cisa.gov/news-events/cybersecurity-advisories/aa21-356a>.

[37] C. Molnar, Interpretable Machine Learning: A Guide for Making Black Box Models Explainable, 2nd ed. Self-published, 2022.

[38] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning: With Applications in R. New York, NY, USA: Springer, 2013, doi: 10.1007/978-1-4614-7138-7.

[39] L. Langsetmo, J. T. Schousboe, B. C. Taylor, J. A. Cauley, H. A. Fink, P. M. Cawthon, D. M. Kado, and K. E. Ensrud, “Advantages and Disadvantages of Random Forest Models for Prediction of Hip Fracture Risk Versus Mortality Risk in the Oldest Old,” JBMR Plus, vol. 7, no. 8, Art. no. e10757, 2023, doi: 10.1002/jbm4.10757.

[40] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, “A comprehensive survey on support vector machine classification: Applications, challenges and trends,” Neurocomputing, vol. 408, pp. 189–215, 2020, doi: 10.1016/j.neucom.2019.10.118.

- [41] Y. Cao, H. Xiang, H. Zhang, Y. Zhu, and K. M. Ting, “Anomaly Detection Based on Isolation Mechanisms: A Survey,” *Machine Intelligence Research*, vol. 22, no. 5, pp. 849–865, 2025, doi: 10.1007/s11633-025-1554-4.
- [42] A. Kumar and J. A. Gutierrez, “Impact of Machine Learning on Intrusion Detection Systems for the Protection of Critical Infrastructure,” *Information*, vol. 16, no. 7, Art. no. 515, 2025, doi: 10.3390/info16070515.
- [43] Sabahi, F., & Movaghar, A. (2008). *Intrusion Detection: A Survey*
- [44] Talaei Khoei & Kaabouch (2023). “A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on Intrusion Detection Systems”
- [45] “Supervised Learning – a workflow chart,” *Reddit /r/MachineLearning* (Sebastian Raschka, 2014).
- [46] Datagy, *Decision Tree Classifier with Sklearn in Python*.
- [47] Wang, Y. et al., “Heatwave Damage Prediction Using Random Forest Model in China,” *Applied Sciences* (2020).
- [48] LearnOpenCV. *Support Vector Machines (SVM)*. LearnOpenCV, July 11, 2018
- [49] K. Shyrokykh, M. Girnyk και L. Dellmuth, “Short text classification with machine learning in the social sciences: The case of climate change on Twitter,” *PLOS ONE*, 2023, $\Sigma\chi$. 1, doi: 10.1371/journal.pone.0290762.
- [50] Baboshkin & Uandykova (2021), “Multi-source Model of Heterogeneous Data Analysis for Oil Price Forecasting”
- [51] “K-Means Anomaly Detection” from DataTechNotes
- [52] “SVM One-Class Classifier For Anomaly Detection,” *Analytics Vidhya*.
- [53] *Isolation forest tutorial*, Anitra Bourne

