ΔΙΕΘΝΕΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΗΣ ΕΛΛΑΔΟΣ

ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΚΑΙ ΗΛΕΚΤΡΟΝΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

«ΑΝΙΧΝΕΥΣΗ ΔΙΑΔΙΚΤΥΑΚΩΝ ΕΠΙΘΕΣΕΩΝ ΜΕ ΧΡΗΣΗ ΑΛΓΟΡΙΘΜΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ»

Του φοιτητή
Κώτση Χαρίλαου
Αρ. Μητρώου: 2019090

Επιβλέπων
Ηλιούδης Χρήστος
Βαθμίδα ……

Ημερομηνία ……

Τίτλος Δ.Ε. Ανίχνευση διαδικτυακών επιθέσεων με τη χρήση αλγορίθμων μηχανικής μάθησης
Κωδικός Δ.Ε. 25279
Ονοματεπώνυμο φοιτητή Κώτσης Χαρίλαος
Ονοματεπώνυμο εισηγητή Ηλιούδης Χρήστος
Ημερομηνία ανάληψης Δ.Ε. 03-07-2025
Ημερομηνία περάτωσης Δ.Ε. …

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

Πρόλογος

Η επιλογή του θέματος της συγκεκριμένης διπλωματικής εργασίας βασίστηκε στο μεγάλο ενδιαφέρον μου για τη κυβερνοασφάλεια και στην επιθυμία μου να εμβαθύνω στις σύγχρονες προσεγγίσεις της ανίχνευσης των διαδικτυακών επιθέσεων (Cybersecurity Monitoring – CM) με τη χρήση της Μηχανικής Μάθησης. Κατά τη διάρκεια των σπουδών μου, η ενασχόληση μου με θέματα της ασφάλειας δεν ήταν εκτεταμένη, το οποίο με οδήγησε στο να αναζητήσω μια εργασία η οποία θα λειτουργήσει ως ένα ουσιαστικό πεδίο μάθησης και πρακτικής εξάσκησης, καθώς με ενδιαφέρει να ασχοληθώ επαγγελματικά με αυτό το χώρο στο μέλλον.

Κατά την εκπόνηση της εργασίας, είχα την ευκαιρία να γνωρίσω σε βάθος έννοιες και προκλήσεις που σχετίζονται με την προεπεξεργασία των δεδομένων, την αξιολόγηση των ταξινομητών στα ανισόρροπα σύνολα και το κίνδυνο της διαρροής πληροφορίας. Επιπλέον, μέσα από τη χρήση της πλατφόρμας OpenSearch και την ανάπτυξη ενός πλήρους αγωγού από την εισαγωγή και την ευρετηρίαση μέχρι τη παραγωγή των προβλέψεων, την οπτικοποίηση και την ειδοποίηση, ανέπτυξα πρακτικές δεξιότητες οι οποίες συνδέουν την ανάλυση των δεδομένων με μια πιο επιχειρησιακή λογική παρακολούθησης.

Η εργασία αυτή ήταν μια σημαντική εμπειρία μάθησης, καθώς ενίσχυσε τόσο το θεωρητικό όσο και το τεχνικό μου υπόβαθρο και ταυτόχρονα επιβεβαίωσε την επιθυμία μου να συνεχίσω να ασχολούμαι ακαδημαϊκά και επαγγελματικά με την κυβερνοασφάλεια.

Περίληψη

Η παρούσα διπλωματική εργασία ασχολείται με το σχεδιασμό, την υλοποίηση και την αξιολόγηση ενός ολοκληρωμένου συστήματος Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring - CM) πάνω στη πλατφόρμα του OpenSearch. Ο βασικός στόχος ήταν να φανεί πρακτικά πως περνάμε από την εκτός σύνδεσης εκπαίδευση των μοντέλων της Μηχανικής Μάθησης σε μία ροή από άκρη σε άκρη η οποία μπορεί να αξιοποιηθεί επιχειρησιακά. Ως μελέτη της περίπτωσης χρησιμοποιήθηκε το dataset CIC-DDoS2019 και συγκεκριμένα τα δεδομένα από την ημέρα 11/03/2019, με χρήση του χρονικού ενδοημερήσιου διαχωρισμού (within-day holdout) για την εκπαίδευση και τον έλεγχο.

Στο πλαίσιο της εργασίας στήθηκε ένα πειραματικό περιβάλλον το οποίο μπορεί να αναπαραχθεί εύκολα με τη χρήση του Docker Compose. Ακόμη, εκπαιδεύτηκαν και αξιολογήθηκαν δύο μοντέλα, το Random Forest και το XGBoost, σε ένα σενάριο χωρίς διαρροή πληροφορίας (non-leaky), ενώ δοκιμάστηκε και ένα ελεγχόμενο σενάριο με διαρροή (leaky) ως αντιπαράδειγμα. Στη συνέχεια, τα μοντέλα εξήχθησαν σε μορφή ONNX και μπήκαν στη πλατφόρμα του OpenSearch μέσω του απομακρυσμένου προγνωστικού εξυπηρετητή και των ML connectors, ώστε οι προβλέψεις να παράγονται κανονικά μέσα στη ροή και να αποθηκεύονται σαν εμπλουτισμένα δεδομένα. Παράλληλα, δημιουργήθηκαν κάποιοι ξεχωριστοί δείκτες για την αποθήκευση των προβλέψεων και για τη ζευγαρωμένη σύγκριση των δύο μοντέλων καθώς και μερικοί πίνακες ελέγχου (dashboards) για την οπτικοποίηση της κίνησης και των αποτελεσμάτων.

Επιπρόσθετα, η αξιολόγηση έδειξε ότι και τα δύο μοντέλα έχουν πολύ υψηλή απόδοση στο σενάριο χωρίς διαρροή πληροφορίας , με μερικές μικρές διαφορές κυρίως στη μειοψηφική κλάση. Επιπλέον, η ανάλυση των διαφωνιών έδειξε ότι οι αποκλίσεις στις προβλέψεις τους είναι σπάνιες και μπορούν να λειτουργήσουν ως ένα χρήσιμο σήμα αβεβαιότητας για μια πιο στοχευμένη διερεύνηση. Αντίθετα, στο σενάριο με διαρροή πληροφορίας προέκυψαν τεχνητά τέλειες μετρικές, κάτι το οποίο επιβεβαιώνει το πόσο παραπλανητική μπορεί να γίνει η διαρροή της πληροφορίας (data leakage). Εν κατακλείδι, η εργασία καταλήγει στο ότι η πραγματική αξία της Μηχανικής Μάθησης στη κυβερνοασφάλεια δεν φαίνεται μόνο στις μετρικές, αλλά κυρίως όταν το μοντέλο εντάσσεται σωστά σε μια ελεγχόμενη και ορατή ροή ενός συστήματος Παρακολούθησης της Ασφάλειας, με τη δυνατότητα την οπτικοποίησης, των ειδοποιήσεων και της πρακτικής αξιοποίησης.

«Cyber attack detection using Machine Learning algorithms»

«Charilaos Kotsis»

Abstract

First of all, this thesis designs, implements and evaluates an edge-to-edge Cybersecurity Monitoring (CM) pipeline on the OpenSearch platform, demonstrating how offline Machine Learning training can be converted into an operational workflow for security monitoring. Furthermore, the experiments use the CIC-DDoS2019 dataset, restricted to network flow records from March 11, 2019, and follow a within-day holdout protocol. Moreover, two supervised classifiers, Random Forest and XGBoost, are trained and assessed with a non-leaky feature set to approximate a realistic deployment. Additionally, a controlled leaky configuration is also tested as a counterexample, showing how the data leakage can inflate the metrics and produce misleading confidence. Also, both models are exported to ONNX to improve the portability and to decouple the inference from the training environment.

For operational inference, a standalone REST prediction service is implemented and integrated into OpenSearch via ML connectors and remote models, enabling the remote inference through the OpenSearch ML plugin. In addition, ingestion is extended with ingest pipelines and dedicated indices for raw flow records, model predictions and pairwise comparisons. Also, OpenSearch Dashboards provide three views, event monitoring, prediction monitoring and pairwise comparison, while OpenSearch Alerting uses monitors and webhook channels to forward notifications to an external Flask-based alert collector. Finally, results show strong overall performance for both models in the non-leaky setting but also confirm that class imbalance requires interpretation beyond accuracy alone. Furthermore, disagreement analysis reveals very high agreement, with rare divergences serving as useful uncertainty signals for targeted investigation. Overall, the work shows that the value of Machine Learning in cybersecurity depends not only on offline scores but also on robust, reproducible integration into a controlled monitoring pipeline.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω θερμά τον καθηγητή μου, κ. Χρήστο Ηλιούδη, για την πολύτιμη καθοδήγηση και υποστήριξή του κατά τη διάρκεια της διπλωματικής μου εργασίας. Η πρότασή του για το θέμα της εργασίας μου, μου έδωσε τη δυνατότητα να ασχοληθώ με έναν τομέα που αναμφίβολα αποτελεί το μέλλον της τεχνολογίας και με ενδιαφέρει πάρα πολύ. Η συμβολή του ήταν καθοριστική για την ολοκλήρωση της εργασίας μου και την ακαδημαϊκή και επαγγελματική μου εξέλιξη.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κατάλογος Εικόνων

Συντομογραφίες

Δ.Ε.	Διπλωματική Εργασία
ΔΙΠΑΕ	Διεθνές Πανεπιστήμιο Ελλάδος
Π.Ε.	Πτυχιακή Εργασία
ACK	Acknowledgment
AI	Artificial Intelligence
API	Application Programming Interface
APT	Advanced Persistent Threat
AUC	Area Under the Curve
CIA	Confidentiality, Integrity, Availability
CIC	Canadian Institute for Cybersecurity
CLI	Command Line Interface
CM	Cybersecurity Monitoring
CPU	Central Processing Unit
DHCP	Dynamic Host Configuration Protocol
DL	Deep Learning
DNS	Domain Name System
DoS	Denial of Service
DDoS	Distributed Denial of Service
DT	Decision Tree
DVB	Digital Video Broadcasting
EA	Enterprise Architecture
F1	F1-score
FedAvg	Federated Averaging
FN	False Negatives
FP	False Positives
FPR	False Positive Rate
GRU	Gated Recurrent Unit
GUI	Graphical User Interface
HIDS	Host-based Intrusion Detection System
HTTP	Hypertext Transfer Protocol

IDS	Intrusion Detection System
IEEE	Institute of Electronics Engineers
IP	Internet Protocol
ICS	Industrial Control Systems
IoT	Internet of Things
JSON	JavaScript Object Notation
KDD	Knowledge Discovery and Data Mining
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LR	Logistic Regression
LSTM	Long Short-Term Memory
MAC	Media Access Control
MAD	Median Absolute Deviation
ML	Machine Learning
MSSQL	Microsoft Structured Query Language Server
NAT	Network Address Translation
NBA	Network Behavior Analysis
NetBIOS	Network Basic Input Output System
NIDS	Network-based Intrusion Detection System
NSL-KDD	Network Security Laboratory-Knowledge Discovery in Databases
NTP	Network Time Protocol
ONNX	Open Neural Network Exchange
PCAP	Packet Capture
PCA	Principal Component Analysis
PPV	Positive Predictive Value
RAM	Random Access Memory
REST	Representational State Transfer
RF	Random Forest
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SCADA	Supervisory Control and Data Acquisition
SIEM	Security Information and Event Management
SNMP	Simple Network Management Protocol

SQL	Structured Query Language
SSDP	Simple Service Discovery Protocol
SSH	Secure Shell
SVM	Support Vector Machines
SYN	Synchronize
TCP	Transmission Control Protocol
TFTP	Trivial File Transfer Protocol
TLS	Transport Layer Security
TN	True Negatives
TP	True Positives
TPR	True Positive Rate
TTL	Time To Live
UDP	User Datagram Protocol
VANETs	Vehicular Ad hoc Networks
VM	Virtual Machine
VPN	Virtual Private Network
WAN	Wide Area Network
WebDDoS	Web Distributed Denial of Service
WIDS	Wireless Intrusion Detection System
HCF	Hop Count Filtering
XGB	XGBoost
XGBoost	Extreme Gradient Boosting
Z-score	Standard Score

Κεφάλαιο 1ο: Εισαγωγή

1.1 Αντικείμενο και σημασία της διπλωματικής

Πρώτα από όλα, η ραγδαία ψηφιοποίηση των υπηρεσιών και η εξάρτηση των οργανισμών από τα πληροφοριακά τους συστήματα έχουν καταστήσει τη δικτυακή ασφάλεια ένα κρίσιμο ζήτημα για τον ιδιωτικό αλλά και το δημόσιο τομέα. Ακόμη, η συνεχόμενη αύξηση του όγκου της διακινούμενης πληροφορίας, η πολυπλοκότητα των δικτυακών υποδομών και η εμφάνιση νέων εξελιγμένων μορφών επιθέσεων (όπως DDoS, botnets, brute force, scanning, exploitation ευπαθειών, κακόβουλο λογισμικό κ.ά.) δημιουργούν ένα περιβάλλον στο οποίο η έγκαιρη ανίχνευση της κακόβουλης δραστηριότητας είναι απαραίτητη για τη διασφάλιση της διαθεσιμότητας, της ακεραιότητας και της εμπιστευτικότητας των δεδομένων.

Επιπρόσθετα, τα πιο παλιά χρονιά η ανίχνευση των επιθέσεων βασιζόταν στα συστήματα ανίχνευσης των εισβολών (Intrusion Detection Systems – IDS) που χρησιμοποιούν κανόνες και υπογραφές (signatures) ήδη γνωστών επιθέσεων. Όμως, παρότι τα συστήματα που βασίζονται σε υπογραφές (signature-based IDS) εξακολουθούν να είναι χρήσιμα παρουσιάζουν σοβαρούς περιορισμούς όταν χρειάζεται να αντιμετωπίσουν άγνωστες ή συνεχώς εξελισσόμενες απειλές[1]. Επιπλέον, σε περιβάλλοντα με πολύ μεγάλο όγκο δεδομένων η χειροκίνητη ανάλυση που χρησιμοποιείται στα IDS είναι πρακτικά αδύνατη[1]. Γι' αυτό το λόγο, ήταν σημαντικό να βρεθούν νέες πιο έξυπνες λύσεις.

Σε αυτό το σημείο, έρχεται να συμβάλει η Μηχανική Μάθηση (Machine Learning) η οποία βελτιώνει τις δυνατότητες στην κατηγορία της ανίχνευσης επιθέσεων του διαδικτύου. Αυτό συμβαίνει, μέσα από την ανάλυση δεδομένων ροών δικτύου καθώς, με αυτό το τρόπο ανακαλύπτει σύνθετα μοτίβα και αναγνωρίζει ανωμαλίες που δεν μπορούν εύκολα να περιγραφούν με απλούς στατικούς κανόνες[2]. Ωστόσο, η σωστή αξιοποίηση της Μηχανικής Μάθησης δεν είναι μια εύκολη λύση προϋποθέτει προσεκτική προεπεξεργασία των δεδομένων, σωστή επιλογή χαρακτηριστικών αλλά και ιδιαίτερη προσοχή για τη αποφυγή φαινομένων όπως η διαρροή δεδομένων (data leakage)[2].

Στο πλαίσιο αυτό εντάσσεται η παρούσα διπλωματική εργασία, η οποία αναλύει την ανίχνευση των διαδικτυακών επιθέσεων, και στο πρακτικό κομμάτι εστιάζει στην ανίχνευση των επιθέσεων της άρνησης εξυπηρέτησης (DDoS), με τη χρήση των αλγορίθμων της Μηχανικής Μάθησης και πιο συγκεκριμένα των αλγορίθμων Random Forest και XGBoost, καθώς και τη πλήρη ενσωμάτωσή τους στη πλατφόρμα του OpenSearch. Η πλατφόρμα OpenSearch είναι ένα σύγχρονο σύστημα αναζήτησης ανοιχτού κώδικα και ανάλυσης δεδομένων που προσφέρει εργαλεία για αποθήκευση, οπτικοποίηση και εφαρμογή μοντέλων Μηχανικής Μάθησης, μέσω του πρόσθετου μηχανικής μάθησης (ML plugin)[3]. Αλλά προσφέρεται και σαν εργαλείο για την υλοποίηση μηχανισμών που σαν σκοπό έχουν την προειδοποίηση του χρήστη (alerting)[3]. Για το πρακτικό σκέλος, τα πειράματα και η ενσωμάτωση υλοποιήθηκαν σε δεδομένα του CIC-DDoS2019 τα οποία αντιστοιχούν αποκλειστικά στην ημέρα 11/03/2019 ώστε η εκπαίδευση και η αξιολόγηση να βρίσκονται σε ενιαίο πλαίσιο.

Τέλος, η αξία της εργασίας φαίνεται στο ότι δεν περιορίζεται απλά σε μια καθαρά θεωρητική ή offline αξιολόγηση των μοντέλων Μηχανικής Μάθησης, αλλά υλοποιεί ένα ολοκληρωμένο αγωγό επεξεργασίας (pipeline) και ανίχνευσης των επιθέσεων από άκρη σε άκρη (edge-to-edge). Ο αγωγός αυτός ξεκινάει από το σύνολο δεδομένων (dataset) CIC-DDoS2019 το οποίο περνάει από μερικά στάδια προεπεξεργασία και εκπαίδευσης των μοντέλων της Μηχανικής Μάθησης και καταλήγει στη λειτουργική ενσωμάτωσή τους στο OpenSearch, μέσα από τους πίνακες οπτικοποίησης (dashboards) και τους μηχανισμούς των ειδοποιήσεων σε πραγματικό χρόνο. Με αυτό τον τρόπο, η εργασία

συνδυάζει τη θεωρητική αλλά και την επιστημονική διάσταση με την πρακτική χρηστικότητα, φέρνοντάς την πιο κοντά στις πραγματικές ανάγκες ενός σύγχρονου περιβάλλοντος διαδικτυακής κυβερνοασφάλειας.

1.2 Στόχοι της διπλωματικής εργασίας

Αρχικά, ο στόχος της συγκεκριμένης διπλωματικής εργασίας είναι η ανάπτυξη και η αξιολόγηση ενός συστήματος ανίχνευσης επιθέσεων στο διαδίκτυο (NIDS) για επιθέσεις τύπου DDoS, το οποίο βασίζεται σε αλγορίθμους Μηχανικής Μάθησης και στη συγκεκριμένη περίπτωση χρησιμοποιήθηκαν οι Random Forest και XGBoost. Ακόμη, όλα τα στοιχεία που χρησιμοποιήθηκαν ενσωματώθηκαν πλήρως λειτουργικά στην πλατφόρμα OpenSearch, αξιοποιώντας δεδομένα μέσα από το dataset CIC-DDoS2019 (DDoS evaluations dataset(CIC-DDoS2019)). Για να επιτευχθεί ο στόχος αυτός, χρειάζεται το θέμα να διερευνηθεί θεωρητικά, να αναλυθεί πειραματικά και να υπάρξει μια πρακτική υλοποίηση σε ένα ενιαίο πλαίσιο. Επιπλέον, για να τεκμηριωθεί εμπειρικά η διαφορά των επιβλεπόμενων μοντέλων της Μηχανικής Μάθησης έναντι των κλασικών προσεγγίσεων, πραγματοποιείται συγκριτική αξιολόγηση και με δύο παραδοσιακούς, μη επιβλεπόμενους στατιστικούς ανιχνευτές αναφοράς, τον Z-score (standard score) και τον MAD (Median Absolute Deviation), κάτω από τις ίδιες πειραματικές συνθήκες.

Πρώτα από όλα, το θεωρητικό επίπεδο της εργασίας έχει ως στόχο την παρουσίαση των βασικών εννοιών της κυβερνοασφάλειας και της ασφάλειας των δικτύων, εστιάζοντας στις κατηγορίες των επιθέσεων που παρατηρούνται σε αυτά τα περιβάλλοντα. Παράλληλα, στο κομμάτι αυτό γίνεται ανάλυση στα συστήματα ανίχνευσης των εισβολών (Intrusion Detection Systems – IDS), στους περιορισμούς που υπάρχουν στις παραδοσιακές προσεγγίσεις ανίχνευσης των επιθέσεων και στις ταξινομήσεις των βασικών τύπων των εισβολών αυτών. Δηλαδή, στις εισβολές σε συστήματα που βασίζονται στο δίκτυο (network-based), σε συστήματα που βασίζονται στον κεντρικό υπολογιστή ή στο τερματικό (host-based), σε συστήματα που βασίζονται σε υπογραφές (signature-based) καθώς και σε συστήματα ανίχνευσης ανωμαλιών (anomaly-based IDS). Στο πλαίσιο αυτό, γίνεται μια μελέτη στις βασικές αρχές της Μηχανικής Μάθησης για την ανίχνευση των επιθέσεων, εστιάζοντας στους αλγορίθμους Random Forest και XGBoost και στις μετρικές αξιολόγησης που χρησιμοποιούνται σε προβλήματα που έχουν να κάνουν με τη ταξινόμηση των ροών του δικτύου. Βέβαια, παρότι το θεωρητικό υπόβαθρο παρουσιάζεται στο γενικό πλαίσιο των διαδικτυακών επιθέσεων, η πειραματική αξιολόγηση και η υλοποίηση της εργασίας εξειδικεύονται σε σενάρια Dos / DDoS επιθέσεων.

Στη συνέχεια, το πειραματικό επίπεδο έχει ως στόχο την προεπεξεργασία του dataset CIC-DDoS2019 (DDoS evaluations dataset(CIC-DDoS2019)) και συγκεκριμένα της ημέρας καταγραφής 11/03/2019 του dataset, με τρόπο συστηματικό και τεκμηριωμένο. Σημειώνεται, ότι στο πρακτικό σκέλος αξιοποιείται μόνο η συγκεκριμένη ημέρα και η αξιολόγηση πραγματοποιείται ως ενδοημερήσια διάσπαση εκπαίδευσης και ελέγχου (within-day holdout) χωρίς ξεχωριστή αξιολόγηση σε διαφορετική ημέρα (cross-day testing). Αρχικά, η διαδικασία αυτή περιλαμβάνει τη συγχώνευση των αρχείων του dataset, τον καθαρισμό των δεδομένων (data cleaning), την κανονικοποίηση των χαρακτηριστικών (normalization) και την επιλογή κατάλληλων γνωρισμάτων (feature selection). Στη συνέχεια, δημιουργούνται δύο διακριτές εκδόσεις του dataset. Μια έκδοση χωρίς διαρροή δεδομένων (non-leaky), η οποία βασίζεται σε ένα περιορισμένο και πιο ασφαλές σύνολο χαρακτηριστικών τα οποία δεν οδηγούν σε διαρροή πληροφορίας και μια έκδοση με διαρροή δεδομένων (leaky), η οποία έχει χαρακτηριστικά με ισχυρή συσχέτιση προς την ετικέτα (label) με σκοπό την σύγκριση και την ανάδειξη των επιπτώσεων της διαρροής δεδομένων (data leakage). Τέλος, στο επίπεδο αυτό επιδιώκεται η εκπαίδευση και η αξιολόγηση των μοντέλων της Μηχανικής Μάθησης Random Forest και XGBoost. Και στις δύο αυτές

εκδοχές αυτό επιτυγχάνεται με τη χρήση κατάλληλων μετρικών όπως η ακρίβεια (accuracy), η ακρίβεια θετικών προβλέψεων (precision), η ανάκληση (recall), ο δείκτης F1 (F1-score) και τέλος ο πίνακας σύγχυσης (confusion matrix).

Επιπρόσθετα, σε ότι έχει να κάνει με το επίπεδο της υλοποίησης στόχος είναι η εκπαίδευση των μοντέλων Random Forest και XGBoost σε μορφή ONNX (Open Neural Network Exchange), ώστε να μπορούν να χρησιμοποιηθούν κατάλληλα από το χρήστη ανεξάρτητα από τη γλώσσα προγραμματισμού και το περιβάλλον εκτέλεσης που θα χρησιμοποιήσει. Πάνω σε αυτή τη βάση αναπτύσσεται ένας προγνωστικός εξυπηρετητής (REST predictor) σε γλώσσα προγραμματισμού Python, με τη χρήση του πλαισίου Flask και της βιβλιοθήκης ONNX Runtime, ο οποίος δέχεται ως είσοδο τα διανύσματα των χαρακτηριστικών και επιστρέφει τις προβλέψεις για το αν μια ροή αξιολογείτε σαν κακόβουλη ή όχι. Στη συνέχεια, γίνονται οι απαραίτητες ρυθμίσεις στο πρόσθετο μηχανικής μάθησης (OpenSearch ML plugin), μέσα από τη δημιουργία των συνδετήρων μηχανικής μάθησης (ML Connectors) και των απομακρυσμένων μοντέλων (REMOTE models). Με αυτό τον τρόπο, το OpenSearch μπορεί να καλεί τα εκπαιδευόμενα μοντέλα μέσω του πρωτοκόλλου HTTP. Επιπλέον, έχουν σχεδιαστεί κάποιοι αγωγοί εισαγωγής (ingest pipelines) και μερικοί δείκτες (indices) για την αποθήκευση τόσο των ροών όσο και των προβλέψεων, επιτρέποντας με αυτό τον τρόπο την ανάλυσή τους σε πραγματικό ή σχεδόν πραγματικό χρόνο (real-time / near real-time).

Κλείνοντας, στο κομμάτι της αξιολόγησης του συστήματος, ο στόχος της διπλωματικής είναι η κατασκευή μερικών πινάκων (dashboards) στο OpenSearch ώστε με αυτό τον τρόπο να οπτικοποιηθούν οι κινήσεις του δικτύου, οι κινήσεις των επιθέσεων αλλά και η συμπεριφορά των μοντέλων της Μηχανικής Μάθησης Random Forest και XGBoost. Ακόμη, γίνεται και ένας πίνακας σύγκρισης των δύο αυτών μοντέλων με βάση τα κοινά τους δεδομένα και τα ζεύγη των προβλέψεων (pairwise predictions). Παράλληλα, υλοποιείται και ο μηχανισμός των ειδοποιήσεων (alerting) στο OpenSearch, ο οποίος μέσω κάποιων παρακολουθητών (monitors) και των αγκίστρων ιστού (webhook) ενεργοποιεί τις ειδοποιήσεις όταν κάποια χρονική περίοδο εντοπίζεται αυξημένη κακόβουλη δραστηριότητα. Η κριτική ανάλυση αυτών των αποτελεσμάτων γίνεται με ιδιαίτερη έμφαση στη διαφορά μεταξύ των non-leaky και τον leaky μοντέλων αλλά και στις επιπτώσεις που έχει το data leakage στην αξιολόγηση ενός συστήματος ανίχνευσης των επιθέσεων. Με αυτό τον τρόπο, ολοκληρώνεται το πλαίσιο των στόχων της διπλωματικής. Κλείνοντας, η υλοποίηση ενός ολοκληρωμένου αγωγού επεξεργασίας ανίχνευσης επιθέσεων στο OpenSearch αποτελεί τον κεντρικό στόχο αυτής της διπλωματικής εργασίας και μπορεί να αποτελέσει μια καλή βάση για περαιτέρω έρευνα ή ακόμη και για μελλοντική ενσωμάτωση σε πραγματικά εταιρικά περιβάλλοντα με τις κατάλληλες παραμετροποιήσεις.

1.3 Επισκόπηση μεθοδολογίας

Αρχικά, η μεθοδολογία που ακολουθείται στη παρούσα διπλωματική εργασία είναι υλοποιημένη σε διαδοχικά στάδια, τα οποία συνδέουν τη θεωρητική διερεύνηση του αντικειμένου με τη πειραματική υλοποίηση και την αξιολόγηση ενός ολοκληρωμένου συστήματος Παρακολούθησης της Ασφάλειας του δικτύου (Cybersecurity Monitoring-CM) πάνω στη πλατφόρμα του OpenSearch. Πρώτα από όλα, γίνεται μια βιβλιογραφική έρευνα με στόχο τη συγκέντρωση αλλά και τη μελέτη επιστημονικών πηγών που αφορούν την κυβερνοασφάλεια (Cybersecurity), την ανίχνευση επιθέσεων δικτύου, τα παραδοσιακά συστήματα ανίχνευσης εισβολών (Intrusion Detection Systems – IDS), καθώς και την εξέλιξή τους στα πιο εξελιγμένα συστήματα Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM). Τα συγκεκριμένα συστήματα Παρακολούθησης της Ασφάλειας συνδυάζουν πολλαπλές πηγές δεδομένων και τις τεχνικές συσχέτισης των συμβάντων[4]. Επίσης, στο ίδιο πλαίσιο

εξετάζονται οι βασικές έννοιες και οι αλγόριθμοι της Μηχανικής Μάθησης (Machine Learning – ML), οι οποίοι είναι κατάλληλοι για την ταξινόμηση των ροών του δικτύου.

Στη συνέχεια, ακολουθεί η συλλογή και η προεπεξεργασία των δεδομένων του dataset CIC-DDoS2019 που αντιστοιχούν αποκλειστικά στην ημέρα 11/03/2019. Το σύνολο των δεδομένων που υπάρχουν επικεντρώνεται στις επιθέσεις τύπου DoS / DdoS, τα δεδομένα συγκεντρώνονται, ενώνονται και καθαρίζονται από ελλιπείς ή μη έγκυρες εγγραφές. Ενώ, τα αριθμητικά χαρακτηριστικά τυποποιούνται και επιλέγονται τα κατάλληλα γνωρίσματα. Σε αυτό το στάδιο, δημιουργούνται δύο εκδοχές των δεδομένων. Μια non-leaky, η οποία στηρίζεται σε ένα περιορισμένο και ασφαλές σύνολο χαρακτηριστικών και μια leaky, η οποία περιλαμβάνει χαρακτηριστικά με ισχυρή συσχέτιση προς την ετικέτα και έχει σαν στόχο την ανάδειξη των επιπτώσεων της διαρροής δεδομένων (data leakage).

Κατόπιν, ξεκινάει το στάδιο της εκπαίδευσης και της αξιολόγησης των μοντέλων της Μηχανικής Μάθησης (Machine Learning – ML). Οι αλγόριθμοι στους οποίους εστιάζει η διπλωματική είναι ο Random Forest και ο XGBoost, οι οποίοι εκπαιδεύονται στη non-leaky και στη leaky εκδοχή του dataset και αξιολογούνται με χρήση κατάλληλων μετρικών ταξινόμησης, όπως η ακρίβεια θετικών προβλέψεων (precision), η ανάκληση (recall), ο δείκτης F1 (F1-score) και ο πίνακας σύγχυσης (confusion matrix). Η σύγκριση μεταξύ των δύο μοντέλων και των δύο εκδοχών των δεδομένων επιτρέπει τη μελέτη πάνω στο θέμα της πραγματικής ικανότητας γενίκευσης των δεδομένων και της ψευδούς φαινομενικής βελτίωσης που μπορεί να προκαλέσει η διαρροή πληροφορίας.

Στο επόμενο στάδιο, τα εκπαιδευόμενα μοντέλα που αναφέραμε πιο πριν μετατρέπονται σε μορφή ONNX (Open Neural Network Exchange) ώστε να είναι φορητά και να μπορούν να χρησιμοποιηθούν από το χρήστη ανεξάρτητα από τη γλώσσα προγραμματισμού που χρησιμοποιείται στην εκάστοτε περίσταση. Πάνω σε αυτή τη βάση δημιουργείται και αναπτύσσεται ένας προγνωστικός εξυπηρετητής REST (REST predictor) σε Python/Flask. Ο οποίος αρχικά φορτώνει τα ONNX μοντέλα, δέχεται ως είσοδο κάποια διανύσματα χαρακτηριστικών και στη συνέχεια επιστρέφει τις προβλέψεις μέσω ενός σταθερά ορισμένου προγραμματιστικού διεπαφής εφαρμογών (Application Programming Interface – API). Τέλος, ο REST predictor λειτουργεί ως συνιστώσα backend του προτεινόμενου συστήματος Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM).

Ακολούθως, η μεθοδολογία συνεχίζεται με την ενσωμάτωση των μοντέλων στην πλατφόρμα του OpenSearch. Όπου, το πρόσθετο της Μηχανικής Μάθησης (OpenSearch ML plugin) τροποποιείται με τη δημιουργία των συνδετήρων Μηχανικής Μάθησης (ML Connectors) και των απομακρυσμένων μοντέλων (REMOTE models) με στόχο την επικοινωνία με τον REST predictor. Παράλληλα, σχεδιάζονται οι αγωγοί της εισαγωγής (ingest pipelines), οι οποίοι εφαρμόζουν τη διαδικασία του συμπερασμού (inference) πάνω στις ροές δεδομένων και αποθηκεύουν τις εγγραφές σε κατάλληλους δείκτες (indices), επιτρέποντας έτσι τη λειτουργία του συστήματος στα σενάρια Παρακολούθησης της Ασφάλειας (CM).

Κλείνοντας, ένα πολύ σημαντικό στοιχείο που χρησιμοποιήθηκε στη μεθοδολογία είναι η οπτικοποίηση και η δημιουργία ειδοποιήσεων. Για να επιτευχθεί αυτό, στο περιβάλλον του OpenSearch Dashboards κατασκευάζονται κάποιοι πίνακες οπτικοποίησης (dashboards) οι οποίοι επιτρέπουν την παρακολούθηση της κίνησης των επιθέσεων που έχουν ανιχνευτεί και της συμπεριφοράς τον μοντέλων Random Forest και XGBoost. Ενώ, παράλληλα ρυθμίζονται οι μηχανισμοί ειδοποίησης (alerting) με τους παρακολουθητές (monitors) και με τα άγκιστρα ιστού (webhooks), ώστε το σύστημα να προλαβαίνει να ενημερώνει τον χρήστη την σωστή χρονική περίοδο για ενδείξεις αυξημένης ή ύποπτης δραστηριότητας. Τέλος, αναλύονται τα πειραματικά αποτελέσματα, με στόχο την εξαγωγή

τεκμηριωμένων συμπερασμάτων ώστε μελλοντικά να γίνουν αρκετές βελτιώσεις στις επεκτάσεις του συστήματος Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM).

1.4 Δομή της διπλωματικής

Πρώτα από όλα, η συγκεκριμένη διπλωματική εργασία είναι οργανωμένη σε επτά κεφάλαια, ώστε ο αναγνώστης να κατανοήσει τη μετάβαση που γίνεται από το θεωρητικό μέρος, στην επεξεργασία των δεδομένων. Έπειτα, στην ανάπτυξη του συστήματος Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM) και τέλος, στην ανάλυση των τελικών αποτελεσμάτων και στα συμπεράσματα που προκύπτουν από αυτά.

Στο πρώτο κεφάλαιο, το οποίο είναι και η εισαγωγή της διπλωματικής παρουσιάζονται το αντικείμενο και η σημασία του θέματος. Ακόμη, διατυπώνονται οι στόχοι της εργασίας, περιγράφεται συνοπτικά η μεθοδολογία που ακολουθήθηκε και αποτυπώνεται η συνολική δομή του κειμένου.

Κατόπιν, το δεύτερο κεφάλαιο εστιάζει στη θεωρητική προσέγγιση της ανίχνευσης των επιθέσεων που γίνονται μέσα στο δίκτυο, με αναφορά και στις DoS / DDoS επιθέσεις που αποτελούν το αντικείμενο του πειραματικού μέρους. Αρχικά, γίνεται μια εισαγωγή των βασικών αρχών της κυβερνοασφάλειας (Cybersecurity) ώστε με αυτό το τρόπο να γίνει κατανοητό το πλαίσιο μέσα στο οποίο εντάσσονται τα συστήματα της ανίχνευσης. Στη συνέχεια, περιγράφονται και ταξινομούνται οι πιο σημαντικές κατηγορίες επιθέσεων που εμφανίζονται στα σύγχρονα δίκτυα. Ακόμη, ακολουθεί μια αναλυτική παρουσίαση των παλιών παραδοσιακών συστημάτων ανίχνευσης εισβολών (Intrusion Detection Systems – IDS), εστιάζοντας στη διάκριση ανάμεσα στην ανίχνευση που βασίζεται στις υπογραφές (signature-based) και στην ανίχνευση των ανωμαλιών (anomaly-based). Επίσης, τονίζονται οι περιορισμοί που υπάρχουν στις κλασικές προσεγγίσεις, τόσο ως προς την ικανότητα τους να καλύπτουν νέες επιθέσεις όσο και ως προς τη παραγωγή ψεύτικων θετικών συναγερμών. Κλείνοντας, το κεφάλαιο ολοκληρώνεται με τη μετάβαση από τα κλασικά συστήματα IDS στα πιο εξελιγμένα συστήματα Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM) και με την ανάγκη για αξιοποίηση των τεχνικών Μηχανικής Μάθησης (Machine Learning – ML) στη δικτυακή ασφάλεια, προκειμένου να βελτιωθεί η ακρίβεια και η προσαρμοστικότητα των μηχανισμών ανίχνευσης.

Στο τρίτο κεφάλαιο, αναλύονται οι βασικοί αλγόριθμοι Μηχανικής Μάθησης (Machine Learning – ML) που χρησιμοποιούνται για το προσδιορισμό και την ανίχνευση επιθέσεων. Εστιάζοντας, στους αλγορίθμους Random Forest και XGBoost, οι οποίοι είναι και αυτοί που χρησιμοποιήθηκαν σε αυτή την εργασία. Επιπρόσθετα, γίνεται μια συνοπτική αναφορά στους κλασικούς αλγόριθμους ταξινόμησης, δύο από αυτούς είναι η λογιστική παλινδρόμηση (Logistic Regression) και οι μηχανές των διανυσμάτων υποστήριξης (Support Vector Machines – SVM). Ακόμη, παρουσιάζονται σε μεγαλύτερο βάθος τα δέντρα απόφασης (Decision Trees) και αναλύονται πλήρως οι αλγόριθμοι Random Forest και XGBoost. Στο ίδιο κεφάλαιο, περιγράφονται και οι μετρικές αξιολόγησης οι οποίες χρησιμοποιούνται για την αποτίμηση της απόδοσης των μοντέλων αυτών, όπως η ακρίβεια (accuracy), η ακρίβεια θετικών προβλέψεων (precision), η ανάκληση (recall), ο δείκτης F1 (F1-score) και τέλος ο πίνακας σύγχυσης (confusion matrix).

Εν συνεχεία, το τέταρτο κεφάλαιο εστιάζει στο σύνολο των δεδομένων (dataset) CIC-DDoS2019 και στην προεπεξεργασία των δεδομένων με τα πειραματικά δεδομένα να αφορούν την ημέρα 11/03/2019 που χρησιμοποιήθηκε στο πρακτικό σκέλος. Πρώτα, περιγράφεται η δομή των ροών (flows) και οι κατηγορίες των επιθέσεων που υπάρχουν στο dataset. Στη συνέχεια, αναφέρονται τα προβλήματα που παρουσιάζει, όπως για παράδειγμα η ανισορροπία μεταξύ των κλάσεων, η ύπαρξη πλεονάζουσας πληροφορίας (redundancy) και η πιθανότητα της διαρροής δεδομένων (data leakage). Κατόπιν,

αναλύονται τα βήματα του καθαρισμού, της κανονικοποίησης και της επιλογής χαρακτηριστικών (feature selection) και παρουσιάζονται οι διαφορές ανάμεσα στα non-leaky και leaky χαρακτηριστικά, το οποίο είναι και ο κεντρικός στόχος της πειραματικής ανάλυσης.

Στο πέμπτο κεφάλαιο, αναφέρεται στη μελέτη περίπτωσης της εργασίας (case study) και στη μεθοδολογία υλοποίησης του συστήματος Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM) πάνω στη πλατφόρμα του OpenSearch. Αρχικά, αναλύονται τα βήματα που χρειάζονται για την εκπαίδευση των μοντέλων Random Forest και XGBoost και η εξαγωγή τους σε μορφή ONNX (Open Neural Network Exchange). Επιπλέον, εξηγείτε η ανάπτυξη του προγνωστικού εξυπηρετητή REST (REST predictor) και η παραμετροποίηση του πρόσθετου της Μηχανικής Μάθησης (OpenSearch ML plugin) με τη χρήση των συνδετήρων Μηχανικής Μάθησης (ML Connectors) και των απομακρυσμένων μοντέλων (REMOTE models). Καθώς επίσης και, η υλοποίηση των αγωγών εισαγωγής (ingest pipelines) και των δεικτών (indices) που υποστηρίζουν τη ροή επεξεργασίας των δεδομένων Παρακολούθησης της Ασφάλειας (CM pipeline). Τέλος, περιγράφονται οι πίνακες της οπτικοποίησης (dashboards) και ο μηχανισμός των ειδοποιήσεων (alerting) που υλοποιείται με τη χρήση των παρακολουθητών (monitors) και των αγκίστρων ιστού (webhooks).

Επιπρόσθετα, το έκτο κεφάλαιο εστιάζει στα πειραματικά αποτελέσματα. Αρχικά, παρουσιάζονται και αναλύονται τα αποτελέσματα των non leaky μοντέλων Random Forest και XGBoost. Ακόμη, γίνεται η συγκριτική τους αξιολόγηση και αναφέρεται η συμπεριφορά των αντίστοιχων leaky μοντέλων, με στόχο να φανεί η επίδραση που έχει η διαρροή δεδομένων (data leakage) στις μετρήσεις της απόδοσης που έγιναν. Εν κατακλείδι, αναλύονται τα αποτελέσματα τα οποία συνοδεύονται από κριτική ανάλυση και ερμηνεία.

Κλείνοντας, στο έβδομο κεφάλαιο συνοψίζονται τα βασικά συμπεράσματα της εργασίας και αναλύεται η επιστημονική και η πρακτική της συνεισφορά στο πεδίο των συστημάτων Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM). Ακόμη, επισημαίνονται οι περιορισμοί που υπάρχουν στη προτεινόμενη προσέγγιση της διπλωματικής και προτείνονται κάποιες κατευθύνσεις με στόχο την μελλοντική επέκταση της. Όπως για παράδειγμα, η διερεύνηση των μοντέλων βαθιάς μάθησης (Deep Learning), η εφαρμογή πάνω σε πραγματικά εταιρικά δεδομένα και η ανάπτυξη κατανεμημένων ή ομοσπονδιακών συστημάτων παρακολούθησης της ασφάλειας (distributed/federated CM systems).Στο τέλος της εργασίας ακολουθούν η βιβλιογραφία και τα παραρτήματα.

1.5 Επίλογος κεφαλαίου

Συνοψίζοντας, στο πρώτο κεφάλαιο παρουσιάστηκε το γενικό πλαίσιο της παρούσας διπλωματικής εργασίας, αναδείχθηκε το πόσο σημαντικό είναι να ανιχνεύονται οι επιθέσεις στο δίκτυο στα σύγχρονα περιβάλλοντα της κυβερνοασφάλειας (Cybersecurity) και διατυπώθηκαν οι κύριοι και επιμέρους στόχοι της. Παράλληλα, έγινε μια επισκόπηση της μεθοδολογίας που ακολουθείται. Ξεκινώντας, από τη βιβλιογραφική έρευνα και την προεπεξεργασία των δεδομένων και συνεχίζοντας, με την εκπαίδευση των μοντέλων Μηχανικής Μάθησης (Machine Learning – ML), την ενσωμάτωση τους σε ένα σύστημα Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM) με τη χρήση της πλατφόρμας του OpenSearch και την οπτικοποίηση και αξιολόγηση των αποτελεσμάτων. Τέλος, περιγράφεται η δομή της εργασίας ώστε ο αναγνώστης να έχει μια καθαρή εικόνα της πορείας που ακολουθεί.

Με αυτό το τρόπο δημιουργούνται οι βάσεις για τα επόμενα κεφάλαια. Έτσι, στο δεύτερο κεφάλαιο αναπτύσσεται το θεωρητικό υπόβαθρο της ανίχνευσης των επιθέσεων του δικτύου. Ακόμη, παρουσιάζονται οι βασικές έννοιες της κυβερνοασφάλειας (Cybersecurity) και αναλύονται οι προσεγγίσεις των συστημάτων ανίχνευσης των εισβολών (Intrusion Detection Systems – IDS) αλλά και

η εξέλιξη τους σε συστήματα Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM). Στη συνέχεια, στα επόμενα κεφάλαια, γίνεται αναφορά στα μοντέλα της Μηχανικής Μάθησης (Machine Learning – ML), στο σύνολο των δεδομένων (dataset) CIC-DDoS2019, στην πειραματική υλοποίηση των δεδομένων πάνω στη πλατφόρμα του OpenSearch αλλά και στη λεπτομερή ανάλυση των πειραματικών αποτελεσμάτων.

Κεφάλαιο 2ο: Ανίχευση επιθέσεων δικτύου

2.1 Εισαγωγή κεφαλαίου

Πρώτα από όλα, η ανίχνευση επιθέσεων στα δίκτυα υπολογιστών είναι ένα από τα πιο σημαντικά και πιο απαιτητικά ζητήματα στο πεδίο της ασφάλειας των δικτύων. Καθώς, είναι επιτακτική ανάγκη τα συστήματα ανίχνευσης των εισβολών στα δίκτυα (Network Intrusion Detection Systems – NIDS) να είναι αξιόπιστα και αποτελεσματικά[5]. Ιδιαίτερα στις μέρες μας που υπάρχουν σύγχρονα και πολύπλοκα περιβάλλοντα. Βέβαια, παρά τις σημαντικές προόδους στη τεχνολογία των συστημάτων ανίχνευσης των εισβολών στα δίκτυα (NIDS) μεγάλο μέρος των λύσεων που υπάρχουν εξακολουθεί να βασίζεται κυρίως στις τεχνικές ανίχνευσης που χρησιμοποιούν υπογραφές γνωστών επιθέσεων (signature-based detection), αντί να χρησιμοποιούν τις πιο ευέλικτες προσεγγίσεις της ανίχνευσης ανωμαλιών (anomaly-based detection), με αποτέλεσμα να περιορίζεται η ικανότητά τους στην αντιμετώπιση νέων απειλών, καθώς και επιθέσεων μεγάλης κλίμακας όπως οι DoS / DDoS[5]. Παράλληλα, η συνεχόμενη αύξηση που υπάρχει στον όγκο της δικτυακής κίνησης των δεδομένων, η μεγάλη ποικιλομορφία που υπάρχει στα πρωτόκολλα και στις υπηρεσίες καθώς, και συνεχής μεταβαλλόμενη φύση των σύγχρονων δικτύων κάνουν ολοένα και δυσκολότερο τον διαχωρισμό μεταξύ της κανονικής και της κακόβουλης συμπεριφοράς[5]. Έτσι, αυξάνονται ολοένα και περισσότερο οι απαιτήσεις από τα συστήματα ανίχνευσης των επιθέσεων. Βέβαια, παρότι στο παρόν κεφάλαιο παρουσιάζονται το γενικό θεωρητικό πλαίσιο της ανίχνευσης της κακόβουλης δικτυακής δραστηριότητας (NIDS / IDS) και η εξέλιξη του προς τις πλατφόρμες Παρακολούθησης της Ασφάλειας (CM), η πειραματική αξιολόγηση και η υλοποίηση της παρούσας διπλωματικής εξειδικεύονται στα σενάρια επιθέσεων άρνησης της εξυπηρέτησης (DoS / DDoS) σύμφωνα με το σύνολο δεδομένων CIC-DDoS2019.

Επιπλέον, στα παραδοσιακά συστήματα ανίχνευσης των εισβολών (Intrusion Detection Systems – IDS), η ανίχνευση βασίζεται κυρίως σε κανόνες και υπογραφές (signatures) ήδη γνωστών επιθέσεων. Αυτά τα δύο συστατικά πρέπει να ενημερώνονται διαρκώς ώστε το σύστημα να παραμένει αποτελεσματικό απέναντι στις νέες μορφές των απειλών[5]. Παρότι αυτή η προσέγγιση εξακολουθεί να χρησιμοποιείται πολύ συχνά, παρουσιάζει αρκετούς σοβαρούς περιορισμούς. Πρώτον, η εξάρτηση τους από προκαθορισμένα μοτίβα οδηγεί στη μειωμένη ικανότητα ανίχνευσης των άγνωστων ή παραλλαγμένων επιθέσεων[5]. Δεύτερον, η ανάγκη για ανθρώπινη παρέμβαση τόσο στη δημιουργία όσο και στη συντήρηση των κανόνων αυξάνει το κόστος, τον χρόνο απόκρισης και τον κίνδυνο για σφάλματα[5]. Οπότε, η συλλογή και η επιμέλεια των αξιόπιστων δεδομένων εκπαίδευσης καθώς και οι μεταβολές της συμπεριφοράς στο δίκτυο με τη πάροδο του χρόνου καθιστούν πολύ δύσκολη τη διατήρηση της σταθερής και παράλληλα αξιόπιστης απόδοσης μέσα στο χρόνο, στα παραδοσιακά συστήματα ανίχνευσης των εισβολών (IDS).

Ακολούθως, οι τωρινές εξελίξεις στην αρχιτεκτονική και τη χρήση των δικτύων έχουν ακόμη περισσότερες απαιτήσεις από τα συστήματα της ανίχνευσης των επιθέσεων. Αρχικά, λόγο της τεράστιας αύξησης του όγκου δεδομένων που διέρχονται μέσα στα δίκτυα σε συνδυασμό με την εξάπλωση του Διαδικτύου των Πραγμάτων (Internet of Things – IoT) και των υπηρεσιών του υπολογιστικού νέφους (cloud services) είναι σημαντικό να δημιουργηθούν περιβάλλοντα τα οποία αναλύουν τη κίνηση με μεγαλύτερη ταχύτητα, αποδοτικότητα και ακρίβεια. Επιπλέον, η αυξανόμενη ποικιλία των πρωτοκόλλων και των τύπων των κινήσεων, η δυναμική συμπεριφορά των συστημάτων και οι επιθέσεις χαμηλής συχνότητας (low-frequency attacks) δυσκολεύουν τον σαφή προσδιορισμό του φυσιολογικού προτύπου λειτουργίας και περιορίζουν την αποτελεσματικότητα των κλασικών τεχνικών ανίχνευσης[5].

Στο πλαίσιο αυτό, οι τεχνικές της Μηχανικής Μάθησης (Machine Learning – ML) και πλέον οι μέθοδοι της Βαθιάς Μάθησης (Deep Learning) φαίνονται ως οι πιο υποσχόμενες προσεγγίσεις, καθώς επιτρέπουν την αυτόματη εξαγωγή των σύνθετων μοτίβων από τα δεδομένα των δικτυακών ροών και μπορούν κάτω από κάποιες προϋποθέσεις να βελτιώσουν την ακρίβεια και την προσαρμοστικότητα των συστημάτων ανίχνευσης των επιθέσεων[5].

2.2 Θεμελιώδεις αρχές κυβερνοασφάλειας

Πρώτα από όλα, πριν προχωρήσουμε στην ανάλυση των επιμέρους εννοιών, πρέπει να σημειωθεί ότι η ενότητα αυτή οργανώνεται στις παρακάτω υποενότητες. Αρχικά, παρουσιάζονται οι βασικοί ορισμοί και οι κεντρικές έννοιες της κυβερνοασφάλειας. Επιπλέον, στην επόμενη υποενότητα αναλύεται η κλασική τριάδα Εμπιστευτικότητα, Ακεραιότητα, Διαθεσιμότητα (Confidentiality - Integrity - Availability, CIA triad). Στη συνέχεια, εξετάζονται οι πρόσθετες αρχές όπως η αυθεντικότητα και η λογοδοσία. Τέλος, στη τελευταία υποενότητα γίνεται αναφορά στις βασικές αρχές σχεδίασης των ασφαλών δικτυακών συστημάτων και στη σύνδεσή τους με την ανίχνευση των επιθέσεων στο δίκτυο.

2.2.1 Βασικές έννοιες

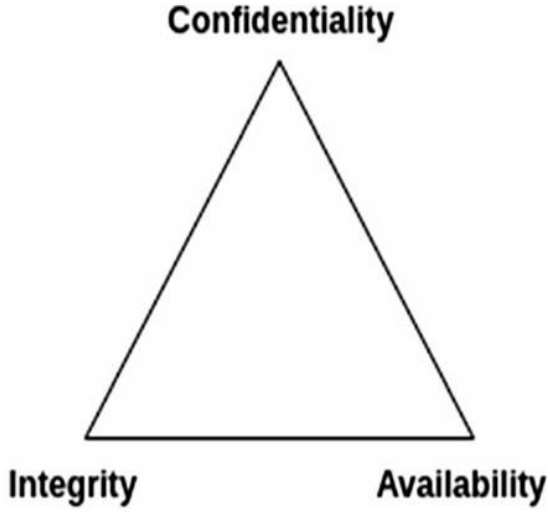
Η κυβερνοασφάλεια (cybersecurity) μπορεί να οριστεί ως ο συνδυασμός των τεχνικών, οργανωτικών και διαδικαστικών μέτρων που στοχεύουν στην προστασία των πληροφοριακών περιουσιακών στοιχείων (informations assets) ενός οργανισμού, δηλαδή των υποδομών, των εφαρμογών, των δεδομένων και των υπηρεσιών από μη εξουσιοδοτημένη πρόσβαση, αλλοίωση, διαρροή ή καταστροφή διασφαλίζοντας τη συνέχιση της λειτουργίας του[6]. Δεν περιορίζεται μόνο σε τεχνικές λύσεις όπως συστήματα τείχους προστασίας (firewalls) ή κρυπτογράφηση αλλά περιλαμβάνει πολιτικές ασφάλειας, διαδικασίες και ανθρώπινους ρόλους, πάνω στη συνεχόμενη προσπάθεια διαχείρισης των κινδύνων.

Πολύ σημαντικό ρόλο στη προσπάθεια αυτή έχουν οι έννοιες της απειλής (threat), της ευπάθειας (vulnerability) και του κινδύνου (risk). Αρχικά, ως απειλή θεωρείται κάθε γεγονός ή ενέργεια που μπορεί να προκαλέσει ανεπιθύμητο αντίκτυπο σε ένα πληροφοριακό περιουσιακό στοιχείο, ενώ ως ευπάθεια ορίζεται μια αδυναμία στο σχεδιασμό, στην υλοποίηση ή στη ρύθμιση ενός συστήματος, η οποία μπορεί να χρησιμοποιηθεί από μια απειλή[6]. Από την άλλη, ο κίνδυνος προκύπτει από τον συνδυασμό της πιθανότητας εκμετάλλευσης μιας ευπάθειας και της σοβαρότητας των συνεπειών για τον οργανισμό. Η διαχείριση των κινδύνων (risk management) περιλαμβάνει την αναγνώριση, την ανάλυση και την αντιμετώπιση αυτών κινδύνων με κατάλληλα μέτρα, ώστε να μειωθούν σε ένα αποδεκτό επίπεδο[6]. Τέλος, στα σύγχρονα περιβάλλοντα της εικονικοποίησης, των υπηρεσιών νέφους (cloud) και των κατανεμημένων υποδομών η επιφάνεια επίθεσης (attack surface) αυξάνεται ολοένα και περισσότερο, γεγονός που κάνει απαραίτητη τη χρήση των μηχανισμών της συνεχούς παρακολούθησης και ανίχνευσης των ανωμαλιών στη συμπεριφορά των συστημάτων και του δικτύου.

2.2.2 Η Τριάδα Εμπιστευτικότητα-Ακεραιότητα-Διαθεσιμότητα(CIA)

Επιπρόσθετα, η κλασική τριάδα Εμπιστευτικότητα, Ακεραιότητα, Διαθεσιμότητα (Confidentiality – Integrity - Availability, CIA) αποτελεί το βασικότερο στόχο προστασίας σε κάθε σύστημα ασφάλειας πληροφοριών και αναφέρεται ενδεικτικά στο Σχήμα 2.1. Πρώτα από όλα, η εμπιστευτικότητα αναφέρεται στην προστασία των πληροφοριών από μη εξουσιοδοτημένη αποκάλυψη, μέσα από μηχανισμούς όπως ο έλεγχος πρόσβασης (access control) και η κρυπτογράφηση (encryption)[6]. Η ακεραιότητα αφορά τη διασφάλιση στο κομμάτι των δεδομένων, δηλαδή ότι αυτά δεν τροποποιούνται η καταστρέφονται χωρίς εξουσιοδότηση, με τεχνικές όπως οι κρυπτογραφικές συναρτήσεις κατακερματισμού (hashing), οι ψηφιακές υπογραφές (digital signatures) και ο μηχανισμός ελέγχου

αλλαγών[6]. Τέλος, η διαθεσιμότητα διασφαλίζει ότι τα συστήματα και οι υπηρεσίες παραμένουν προσβάσιμα στους εξουσιοδοτημένους χρήστες ακόμη και κάτω από σφάλματα ή από επιθέσεις άρνησης υπηρεσίας (Denial of Service – DoS), μέσω των εφεδρικών υποδομών, των ανθεκτικών αρχιτεκτονικών και κατάλληλων μηχανισμών ανάκαμψης[6].



Σχήμα 2.1: The CIA triad[6]

2.2.3 Πρόσθετες αρχές ασφάλειας

Βέβαια, εκτός από την τριάδα CIA, τα σύγχρονα πλαίσια ασφαλείας αναφέρονται και σε κάποιες άλλες πρόσθετες αρχές που εξειδικεύουν τους στόχους προστασίας, οι σημαντικότερες από αυτές φαίνονται στο Σχήμα 2.2. Η αυθεντικοποίηση (authentication) εξασφαλίζει την αξιόπιστη ταυτοποίηση των χρηστών, των συσκευών ή των υπηρεσιών πριν τους δοθεί η πρόσβαση σε πόρους ενώ η εξουσιοδότηση (authorization) καθορίζει σε ποιους πόρους και με ποια δικαιώματα επιτρέπεται η πρόσβαση, σύμφωνα με την αρχή της ελάχιστης αναγκαίας πρόσβασης (least privilege)[6]. Ακόμη, η λογοδοσία (accountability) και η ιχνηλασιμότητα (traceability) προϋποθέτουν τη καταγραφή (logging) των κρίσιμων ενεργειών, ώστε αυτές να μπορούν να αποδοθούν σε συγκεκριμένες ταυτότητες και να αναλυθούν αν συμβεί κάποια περίπτωση περιστατικού ασφάλειας[6]. Επίσης, η μη αποποίηση της ευθύνης (non-repudiation) διασφαλίζει ότι τα εμπλεκόμενα μέρη δεν μπορούν να αρνηθούν εκ των υστέρων μια ενέργεια, όπως για παράδειγμα την αποστολή ενός μηνύματος ή την έγκριση μιας συναλλαγής[6]. Τέλος, η ιδιωτικότητα (privacy) και η προστασία των δεδομένων προσωπικού χαρακτήρα έχουν ιδιαίτερη σημασία σε περιβάλλοντα όπου εφαρμόζονται εκτεταμένοι μηχανισμοί παρακολούθησης και καταγραφής της κίνησης του δικτύου[6].

Σχήμα 2.2: CIAAA principle[6]

2.2.4 Αρχές σχεδίασης της ασφάλειας σε δίκτυα

Όλοι οι παραπάνω στόχοι και οι αρχές ασφάλειας στο επίπεδο των υποδομών υλοποιούνται μέσω συγκεκριμένων αρχών σχεδίασης των ασφαλών δικτυακών συστημάτων. Πρώτα, η άμυνα σε βάθος (defense in depth) προτείνει τη χρήση πολλαπλών, διαδοχικών επιπέδων ελέγχου, όπως για παράδειγμα τα τείχη της προστασίας (firewalls), τις ζώνες του απομονωμένου δικτύου (DMZ), τους μηχανισμοί ελέγχου της πρόσβασης και τα συστήματα ανίχνευσης και αποτροπής εισβολών (NIDS and NIPS)[6]. Ώστε, με αυτό το τρόπο η παραβίαση σε ένα επίπεδο να μην οδηγεί αμέσως στο πλήρη συμβιβασμό του συστήματος. Ακόμη, η αρχή της ελάχιστης αναγκαίας πρόσβασης (least privilege) εφαρμόζεται τόσο σε χρήστες ατομικά όσο και σε υπηρεσίες αλλά και σε συσκευές, έτσι περιορίζει σε μεγάλο βαθμό τη δυνατότητα της πλευρικής κίνησης (lateral movement) ενός επιτιθέμενου μέσα στο δίκτυο[6]. Επίσης, η τμηματοποίηση και ζωνοποίηση του δικτύου (network segmentationzoning) απομονώνει τις κρίσιμες υπηρεσίες σε ξεχωριστές ζώνες με αυστηρά ελεγχόμενη κίνηση μεταξύ τους, γεγονός που διευκολύνει τόσο στη πρόληψη όσο και στην ανίχνευση των επιθέσεων[6]. Τέλος, αρχές όπως η αποτυχία σε ασφαλή κατάσταση (fail-safe / fail-secure), ο διαχωρισμός των καθηκόντων (separation of duties) και πιο πρόσφατα η φιλοσοφία της μηδενικής εμπιστοσύνης (zero trust) αυξάνουν τη συνολική ανθεκτικότητα του δικτύου και δημιουργούν ένα περιβάλλον στο οποίο τα συστήματα ανίχνευσης των επιθέσεων και οι πλατφόρμες Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM), όπως είναι και το OpenSearch που χρησιμοποιήθηκε σε αυτήν την εργασία μπορούν να λειτουργήσουν αποτελεσματικά αξιοποιώντας σωστά όλα τα δομημένα δεδομένα της παρακολούθησης[6].

2.3 Προσδιορισμός και ταξινόμηση επιθέσεων σε σύγχρονα δίκτυα

Πρώτα από όλα, βασικό στοιχείο για τον αποτελεσματικό σχεδιασμό των συστημάτων ανίχνευσης των εισβολών (Network Intrusion Detection Systems – NIDS) και των πλατφορμών Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM) είναι η κατανόηση του τρόπου με τον οποίο εκδηλώνονται οι επιθέσεις στα σύγχρονα δίκτυα. Στις περισσότερες περιπτώσεις οι επιτιθέμενοι δεν κάνουν τυχαίες κινήσεις αλλά ακολουθούν συγκεκριμένα βήματα δράσης, τα οποία μπορούν να

περιγραφούν και να ομαδοποιηθούν σε κατηγορίες κακόβουλης κίνησης (malicious traffic) ώστε να διευκολύνεται η μελέτη αλλά και η ανίχνευση τους.

2.3.1 Στάδια εξέλιξης κακόβουλης δραστηριότητας

Γενικότερα, στη διεθνή βιβλιογραφία προτείνεται η ανάλυση της κακόβουλης δραστηριότητας σε διαδοχικά στάδια ελέγχου (control stages), τα οποία περιγράφουν τη πορεία μιας επίθεσης από την αρχική διερεύνηση μέχρι την υλοποίησή της. Αρχικά, στο πρώτο στάδιο δηλαδή στο στάδιο της αναγνώρισης (reconnaissance) ο επιτιθέμενος συλλέγει πληροφορίες για τον στόχο, αξιοποιώντας παθητικές τεχνικές, όπως για παράδειγμα η ανάλυση των διαθέσιμων δεδομένων που βρίσκονται δημόσια αλλά και πιο ενεργητικές τεχνικές, όπως οι δοκιμαστικές συνδέσεις και οι μετρήσεις χρονισμού[7]. Στη συνέχεια, ακολουθεί το στάδιο της σάρωσης (scanning) στο οποίο χρησιμοποιούνται πιο συστηματικές προσπάθειες εντοπισμού ανοικτών θυρών, εκτεθειμένων υπηρεσιών και πιθανών ευπαθειών με τη χρήση της σάρωσης θυρών (port scanning) ή της αποτύπωσης των λειτουργικών συστημάτων (application fingerprinting) και εφαρμογών[7]. Τέλος, στο τελικό στάδιο δηλαδή σε αυτό της επίθεσης (attack) γίνεται η ουσιαστική παραβίαση. Δηλαδή, ο επιτιθέμενος επιχειρεί να εκμεταλλευτεί τις αδυναμίες που έχει εντοπίσει στο σύστημα, να παρακάμψει τους μηχανισμούς της άμυνας και να πετύχει τους στόχους του, οι οποίοι είναι η μη εξουσιοδοτημένη πρόσβαση, η διαρροή ή αλλοίωση των δεδομένων καθώς και η διατάραξη της διαθεσιμότητας των υπηρεσιών[7].

2.3.2 Κατηγορίες κακόβουλης κίνησης

Βέβαια, πέρα από τα στάδια της εξέλιξης πολύ σημαντικό ρόλο παίζει και η ταξινόμηση των επιθέσεων σε κατηγορίες και αυτό γίνεται με βάση το επίπεδο του συστήματος που στοχεύουν και με βάση το τύπο της κακόβουλης ενέργειας. Μία προσέγγιση που συναντάται πολύ συχνά χωρίζει τις κακόβουλες κινήσεις σε πέντε βασικές κατηγορίες. Στις επιθέσεις δικτύου (network attacks), στις επιθέσεις σε υπολογιστικά συστήματα ή κόμβους (host-based attacks), στις επιθέσεις λογισμικού ή εφαρμογών (software or application attacks), στις φυσικές επιθέσεις (physical attacks) και στις επιθέσεις που σχετίζονται με τον ανθρώπινο παράγοντα και την κοινωνική μηχανική (social engineering)[7]. Στη παρούσα εργασία, το πειραματικό σκέλος εστιάζει στην υποκατηγορία των επιθέσεων δικτύου και συγκεκριμένα στις επιθέσεις DoS / DDoS καθώς αυτές αποτελούν τον πυρήνα του CIC-DDoS2019.

Αρχικά, οι επιθέσεις δικτύου μεταξύ κάποιων άλλων επιθέσεων περιλαμβάνουν τις επιθέσεις της άρνησης υπηρεσίας (Denial of Service – DOS and Destributed Denial of Service – DDOS), τις επιθέσεις με ενδιάμεσο εισβολέα (man-in-the-middle attacks), τη πλαστογράφηση ή ειδική κατασκευή πακέτων (packet crafting) και τις προηγμένες τεχνικές σάρωσης και εκμετάλλευσης των χαρακτηριστικών πρωτοκόλλων[7]. Ενώ, οι επιθέσεις σε υπολογιστικά συστήματα στοχεύουν σε συγκεκριμένους κόμβους ή συσκευές και εκδηλώνονται μέσα από κακόβουλα λογισμικά (malware), από δούρειους ίππους (trojans), από κλοπή η σπάσιμο κωδικών πρόσβασης (password cracking) και από καταχρήσεις προνομίων (privilege abuse) και αλλοίωσης των κρίσιμων ρυθμίσεων του συστήματος (modification of critical system configurations)[7].

Στη συνέχεια, στη κατηγορία των επιθέσεων λογισμικού ή εφαρμογών εντάσσονται οι επιθέσεις σε επίπεδο εφαρμογής όπως για παράδειγμα η έγχυση κώδικα σε βάσεις δεδομένων (SQL injection), οι επιθέσεις υπερχείλισης της ενδιάμεσης μνήμης (buffer overflow), καθώς και οι επιθέσεις στον φυλλομετρητή (browser attacks) όπως η επίθεση παραπλάνησης κλικ (clickjacking) και οι επιθέσεις τύπου άνθρωπος στον φυλλομετρητή (man-in-the-browser)[7]. Από την άλλη, οι φυσικές επιθέσεις αφορούν τις παρεμβάσεις σε συσκευές και αισθητήρες (physical attacks on devices and sensors), την

εγκατάσταση κρυφών μηχανισμών πρόσβασης στο υλικό (hardware backdoors), την αλλοίωση των μετρήσεων ή των δεδομένων προέλευσης (data provenance) και γενικότερα τις επιθέσεις που αξιοποιούν τη φυσική πρόσβαση σε υποδομές[7]. Τέλος, είναι και οι επιθέσεις κοινωνικής μηχανικής όπως το ηλεκτρονικό ψάρεμα (phishing) και το στοχευμένο ηλεκτρονικό ψάρεμα(spear-phishing), τα οποία εκμεταλλεύονται κυρίως τον ανθρώπινο παράγοντα πείθοντας τους χρήστες να αποκαλύψουν τα διαπιστευτήρια ή να εκτελέσουν εν αγνοία τους μερικές κακόβουλες ενέργειες[7].

2.3.3 Σύνδεση ταξινομίας με συστήματα ανίχνευσης επιθέσεων δικτύου (NIDS) και πλατφόρμες παρακολούθησης της ασφάλειας (CM)

Επιπρόσθετα, σε ένα σύστημα ανίχνευσης επιθέσεων του δικτύου (NIDS) και μία πλατφόρμα Παρακολούθησης της Ασφάλειας (CM) η ταξινόμηση που έγινε παραπάνω προσφέρει ένα χρήσιμο εννοιολογικό πλαίσιο. Έτσι, επιτρέπει να χαρτογραφηθούν συγκεκριμένοι τύποι κακόβουλης κίνησης σε αντίστοιχες κλάσεις ταξινόμησης, να σχεδιαστούν κατάλληλες μετρικές ανίχνευσης και να αξιολογηθεί κατά πόσο ένα dataset όπως το CIC-DDoS2019 που χρησιμοποιήθηκε στην παρούσα εργασία καλύπτει το εύρος των σεναρίων της κατανεμημένης άρνησης της υπηρεσίας (DDoS), απειλών που εξετάζει αυτή η εργασία[7]. Με αυτό το τρόπο, ο προσδιορισμός και η ταξινόμηση των επιθέσεων στα σύγχρονα δίκτυα δεν αποτελούν μόνο θεωρητική άσκηση αλλά συνδέονται άμεσα και με τον τρόπο που σχεδιάζονται, εκπαιδεύονται και αξιολογούνται τα μοντέλα της Μηχανικής Μάθησης και τα συστήματα ανίχνευσης των επιθέσεων στα πραγματικά περιβάλλοντα[7].

2.4 Παραδοσιακά Συστήματα Ανίχνευσης Εισβολών (Traditional IDS): Ανίχνευση βάσει υπογραφών (Signature Detection) και ανίχνευση ανωμαλιών (Anomaly Detection)

Γενικότερα, τα συστήματα ανίχνευσης εισβολών (IDS) χωρίζονται σε δύο βασικές προσεγγίσεις, την ανίχνευση με υπογραφές (signature-based detection) και την ανίχνευση των ανωμαλιών (anomaly-based detection). Ο διαχωρισμός αυτός βοηθάει στον συστηματικό χαρακτηρισμό των συστημάτων ανίχνευσης εισβολών δηλαδή, στη κατανόηση των δυνατοτήτων και των περιορισμών τους αλλά και στη σωστή επιλογή ή σχεδίαση των λύσεων ανάλογα με το περιβάλλον στο οποίο πρόκειται να εφαρμοστούν[8].

2.4.1 Ανίχνευση με υπογραφές

Πρώτα από όλα, στα συστήματα ανίχνευσης με υπογραφές το αν μία ροή ή ένα πακέτο είναι κακόβουλο βασίζεται στη σύγκριση των παρατηρούμενων μοτίβων της κίνησης μέσω των γνώσεων που υπάρχουν από παλιές επιθέσεις, οι οποίες περιγράφονται με κανόνες ή υπογραφές (signatures)[8]. Κάθε υπογραφή κωδικοποιεί τα χαρακτηριστικά γνωρίσματα μιας συγκεκριμένης επίθεσης, όπως για παράδειγμα τους συγκεκριμένους συνδυασμούς πεδίων στα πρωτόκολλα, τις ακολουθίες των πακέτων ή γνωστά μοτίβα στο περιεχόμενο (payload) και το σύστημα ενεργοποιεί κάποιο συναγερμό όταν εντοπίσει την αντιστοιχία της τρέχουσας κίνησης με κάποια από αυτές[8].

Καθώς, το μεγαλύτερο πλεονέκτημα της προσέγγισης με υπογραφές είναι ότι έχει υψηλή ακρίβεια και χαμηλά ποσοστά ψευδών θετικών αποτελεσμάτων (false positives) για επιθέσεις που έχουν ήδη μοντελοποιηθεί[9]. Αυτό, την κάνει μία πολύ καλή επιλογή για τα λειτουργικά περιβάλλοντα όπου η σταθερότητα των συναγερμών είναι κρίσιμη. Από την άλλη, η αποτελεσματικότητα αυτών των συστημάτων εξαρτάται άμεσα από την πληρότητα και την επικαιρότητα της βάσης υπογραφών, με

αποτέλεσμα να δυσκολεύονται να εντοπίσουν νέες άγνωστες ή τροποποιημένες επιθέσεις (zero-day attacks) και να απαιτούν συνεχή ενημέρωση και συντήρηση από εξειδικευμένο προσωπικό[8].

2.4.2 Ανίχνευση ανωμαλιών

Από την άλλη, τα συστήματα ανίχνευσης των ανωμαλιών ακολουθούν μια διαφορετική λογική. Καθώς, αντί να αναζητούν συγκεκριμένες γνωστές υπογραφές, προσπαθούν να εντοπίσουν αποκλίσεις από ένα μοντέλο κανονικής συμπεριφοράς του δικτύου ή του συστήματος[8]. Το μοντέλο αυτό προκύπτει τις περισσότερες φορές μέσα από στατιστικές τεχνικές, μεθόδους Μηχανικής Μάθησης ή άλλες προσεγγίσεις και έχει εκπαιδευτεί πάνω σε δεδομένα που θεωρούνται αντιπροσωπευτικά της φυσιολογικής λειτουργίας του δικτύου[8].

Επιπλέον, στη φάση λειτουργίας τα συστήματα ανίχνευσης των εισβολών σηματοδοτούν ως ύποπτες τις ροές ή τα γεγονότα που αποκλίνουν πολύ από τη καθιερωμένη κανονικότητα ακόμη και αν δεν αντιστοιχούν σε κάποια γνωστή υπογραφή επίθεσης[9]. Με αυτό το τρόπο, η ανίχνευση ανωμαλιών μπορεί να εντοπίσει νέες, εξελισσόμενες επιθέσεις ή επιθέσεις με χαμηλή ένταση που δεν είναι εύκολο να περιγραφούν με στατικούς κανόνες[9]. Ωστόσο, η ακρίβεια αυτών των συστημάτων επηρεάζεται σε μεγάλο βαθμό από την ποιότητα του μοντέλου κανονικότητας και από τη διαθεσιμότητα των αξιόπιστων δεδομένων εκπαίδευσης, ενώ στα δυναμικά περιβάλλοντα είναι συχνό φαινόμενο τα αυξημένα ποσοστά των ψευδών θετικών συναγερμών[9].

2.4.3 Σύγκριση προσεγγίσεων και υβριδικές αρχιτεκτονικές συστημάτων ανίχνευσης εισβολών

Αρχικά όπως φαίνεται και στο Πίνακα 2.1, η επιλογή ανάμεσα στην ανίχνευση με υπογραφές και την ανίχνευση ανωμαλιών στην πραγματικότητα είναι ένας συμβιβασμός ανάμεσα στη σταθερή απόδοση για τις γνωστές επιθέσεις, την ικανότητα εντοπισμού των νέων απειλών και το επίπεδο των λανθασμένων θετικών συναγερμών που μπορούν να ανεχθούν. Καθώς, τα συστήματα που βασίζονται σε υπογραφές προσφέρουν πολύ μεγάλη ειδικότητα και προβλέψιμη συμπεριφορά σε σχέση με τις τεκμηριωμένες επιθέσεις αλλά έχουν περιορισμένη ικανότητα κάλυψης του άγνωστου χώρου απειλών (the unknown space of threats)[8]. Αντίθετα, τα συστήματα ανίχνευσης των ανωμαλιών μπορούν και προσαρμόζονται σε νέα μοτίβα κακόβουλης δραστηριότητας και σε ετερογενή περιβάλλοντα, με αντάλλαγμα όμως την αυξημένη πολυπλοκότητα της σχεδίασης και τη διαχείριση του μεγάλου αριθμού των λανθασμένων θετικών συναγερμών[9].

Πίνακας 2.1: Σύγκριση Signature-based VS Anomaly-based IDS

Προσέγγιση	Βασική ιδέα	Πλεονεκτήματα	Μειονεκτήματα	Τυπικές χρήσεις
Signature-based	Σύγκριση με γνωστές υπογραφές επιθέσεων	Υψηλή ακρίβεια σε γνωστές επιθέσεις, λίγα false positives	Δεν βλέπει zero-day, θέλει συνεχή ενημέρωση	Παραδοσιακά NIDS/IDS, antivirus
Anomaly-based	Εντοπισμός απόκλισης από «κανονική» συμπεριφορά	Μπορεί να δει νέες/άγνωστες επιθέσεις	Πολλά false positives, δύσκολη ρύθμιση	Προηγμένα NIDS, ML-based IDS

Για τον λόγο αυτό, προτείνονται συχνά οι υβριδικές αρχιτεκτονικές των συστημάτων ανίχνευσης των εισβολών (Hybrid IDS architectures), στις οποίες συνδυάζονται οι δύο παραπάνω προσεγγίσεις ώστε με αυτό το τρόπο να αξιοποιούνται τα πλεονεκτήματα και των δύο και να μετριάζονται οι αδυναμίες της καθεμίας. Σε τέτοια συστήματα, η μονάδα ανίχνευσης με υπογραφές είναι υπεύθυνη για τον πιο

αποδοτικό εντοπισμό των ήδη γνωστών επιθέσεων, με περιορισμένες υπολογιστικές απαιτήσεις[9]. Ενώ, η μονάδα ανίχνευσης ανωμαλιών συχνά βασίζεται σε τεχνικές Μηχανικής Μάθησης και εστιάζει στην ανακάλυψη νέων και ύπουλων επιθέσεων που δεν έχουν ακόμη καταγραφεί[9]. Αυτή η υβριδική λογική συνδέεται άμεσα με τη μετάβαση από τα παραδοσιακά συστήματα ανίχνευσης εισβολών (IDS) στις πιο ευέλικτες πλατφόρμες Παρακολούθησης της Ασφάλειας (CM), όπου η συσχέτιση των πολλαπλών πηγών δεδομένων και η χρήση των προηγμένων μοντέλων ανάλυσης αποτελούν τα κεντρικά στοιχεία του σχεδιασμού[9].

2.4.4 Παραδοσιακές προσεγγίσεις ανίχνευσης DDoS επιθέσεων (Traditional DDoS detection approaches)

Οι παραδοσιακές προσεγγίσεις ανίχνευσης και άμυνας απέναντι σε επιθέσεις DDoS στηρίζονται κυρίως στις τεχνικές φιλτραρίσματος (filtering) και περιορισμού του ρυθμού (rate limiting) της ύποπτης κίνησης, με κριτήρια τα οποία προκύπτουν από κανόνες, ευρετικές και ενδείξεις της δικτυακής συμπεριφοράς[10]. Μια κλασική οπτική, τις ταξινομεί σε τρείς κατηγορίες, πρώτον στις τεχνικές της αντοχής (survival techniques), όπου ενισχύονται οι πόροι ή εφαρμόζεται πλεονασμός ώστε ο στόχος να συνεχίσει να λειτουργεί[10]. Δεύτερον, στις προληπτικές τεχνικές (proactive techniques), οι οποίες επιδιώκουν να εντοπίσουν και να περιορίσουν την επίθεση πριν επηρεάσει τον στόχο και τέλος, στις αντιδραστικές τεχνικές όπου η ανίχνευση γίνεται αφού ο στόχος έχει ήδη αρχίσει να επηρεάζεται και ακολουθεί, ο μετριασμός (mitigation) και όπου είναι εφικτό οι διαδικασίες ιχνηλάτησης (traceback)[10].

Ένα ακόμη βασικό χαρακτηριστικό των παραδοσιακών αμυνών είναι ότι μπορούν να αναπτυχθούν σε διαφορετικά σημεία του δικτύου, δηλαδή στο άκρο της πηγής, στο πυρήνα, στο άκρο του θύματος ή κατανεμημένα και κάθε επιλογή συνοδεύεται από διαφορετικά ανταλλάγματα. Για παράδειγμα, από το πυρήνα μέχρι το άκρο (core-end) η κίνηση εμφανίζεται πιο συσσωρευμένη (aggregated), κάτι το οποίο δυσκολεύει τη διάκριση της νόμιμης από τη κακόβουλη κίνηση και μπορεί να αυξήσει το κίνδυνο του λανθασμένου φιλτραρίσματος[10]. Στο πλαίσιο αυτό, μία από τις πιο κλασικές γραμμές άμυνας είναι το φιλτράρισμα της εισερχόμενης και εξερχόμενης κίνησης (ingress and egress filtering) στους δρομολογητές άκρου (edge routers), όπου εφαρμόζονται κανόνες που απορρίπτουν πακέτα με πηγαία διεύθυνση (source IP) που δεν αντιστοιχεί στο αναμενόμενο ή εκχωρημένο χώρο διευθύνσεων (address space), περιορίζοντας έτσι τη δυνατότητα παραποίησης της διεύθυνσης IP (IP spoofing)[10]. Παρότι η λογική αυτή είναι χρήσιμη, έχει μερικούς γνωστούς περιορισμούς, όπως το ότι ο επιτιθέμενος μπορεί να μιμηθεί τις διευθύνσεις εντός του σωστού υποδικτύου, ότι σε επιθέσεις του επιπέδου εφαρμογής (application-layer floods) συχνά δεν χρησιμοποιείται παραποίηση (spoofing), αλλά εμφανίζονται οι πραγματικές διευθύνσεις των μολυσμένων ή συμβιβασμένων συσκευών (bots) και ότι η διαχείριση και συντήρηση των πολιτικών αυξάνει το διοικητικό κόστος (administrative overhead)[10].

Παράλληλα, εντοπίζονται προσεγγίσεις που βασίζονται στη στατιστική παρακολούθηση της κίνησης και στην ανίχνευση των αποκλίσεων πολύ κοντά στη πηγή (source-end). Σε τέτοιες αρχιτεκτονικές, όταν εντοπίζονται μερικές ανωμαλίες σε σχέση με τα αναμενόμενα πρότυπα, εφαρμόζεται περιορισμός του ρυθμού (rate-limiting) προς τους προορισμούς που θεωρείται ότι δέχονται επίθεση, προσπαθώντας ταυτόχρονα να προστατευθεί η νόμιμη κίνηση προς τους άλλους προορισμούς[10]. Η λογική αυτή μπορεί να είναι αποτελεσματική στις έντονες ροές των δεδομένων, όμως εισάγει υπολογιστική επιβάρυνση (overhead) στα στοιχεία του άκρου και δεν είναι πάντα εύκολη η καθαρή διάκριση της νόμιμης από τη κακόβουλη κίνηση, κάτι το οποίο μπορεί να οδηγήσει είτε σε άσκοπο περιορισμό (false positives) είτε σε διαφυγές (false negatives)[10].

Επιπλέον, οι παραδοσιακές άμυνες αξιοποιούν ευρετικές που στοχεύουν στη παραποίηση και στις πρωτοκολλικές ιδιαιτερότητες. Ενδεικτικά, το φιλτράρισμα με βάση των αριθμό των αλμάτων (HCF) στο άκρο του θύματος (victim-end) αξιοποιεί το χρόνο ζωής του πακέτου (TTL), ώστε να εκτιμηθεί ο αριθμός των αλμάτων (hops) και να ελεγχθεί αν το προφίλ της διαδρομής ταιριάζει με τα αναμενόμενα προφίλ των νόμιμων πελατών (clients), απορρίπτοντας πακέτα με ύποπτα χαρακτηριστικά[10]. Ωστόσο, η αποτελεσματικότητά του επηρεάζεται σε περιβάλλοντα με δυναμικές διευθύνσεις (DHCP), σε χρήστες πίσω από το μηχανισμό NAT και σε νέους νόμιμους πελάτες που δεν υπάρχουν στο πίνακα αναφοράς[10]. Αντίστοιχα, για τις επιθέσεις τύπου πλημμύρας SYN (SYN flood), τα SYN cookies αποτελούν κλασικό μηχανισμό άμυνας, ώστε ο διακοσμητής να μην δεσμεύει κάποια κατάσταση πριν ολοκληρωθεί η χειραψία TCP (TCP handshake), μειώνοντας της εξάντληση των πόρων λόγω των ημι-ανοικτών συνδέσεων[10]. Παρότι είναι ιδιαίτερα χρήσιμα έναντι της εξάντλησης της κατάστασης (state exhaustion), δεν αντιμετωπίζουν επιθέσεις που εξαντλούν το διαθέσιμο εύρος ζώνης (bandwith) και συνοδεύονται από λειτουργικούς ή υπολογιστικούς περιορισμούς, όπως το κόστος υπολογισμού του cookie και ο χειρισμός των απωλειών SYN/ACP[10].

Τέλος, ένα σημαντικό συμπληρωματικό σκέλος των παραδοσιακών αμυνών είναι οι μηχανισμοί ιχνηλάτησης της διαδρομής (traceback) και οι πρακτικές του μετριασμού ανάντη (upstrean mitigation), με στόχο ο μετριασμός να εφαρμόζεται όσο το δυνατόν πιο νωρίς στην αλυσίδα της μεταφοράς[10]. Οικογένειες τεχνικών όπως η σήμανση των πακέτων (packet marking), στοχαστική (probabilistic) ή ντετερμινιστική (deterministic), η καταγραφή των πακέτων (packet logging) και η τεχνική pushback επιδιώκουν να υποστηρίξουν την ανακατασκευή της διαδρομής και τον εντοπισμό των πηγών της επίθεσης[10]. Ωστόσο, στη πράξη η υλοποίηση της ιχνηλάτησης της διαδρομής παραμένει δύσκολη λόγω της πλαστογράφησης των διευθύνσεων IP (IP spoofing), του άνευ κατάστασης χαρακτήρα της δρομολόγησης IP (stateless IP routing) και της πολυπλοκότητας των σύγχρονων σεναρίων της επίθεσης[10].

Συνολικά, οι παραδοσιακές προσεγγίσεις αποτελούν ένα πρώτο στρώμα άμυνας το οποίο στηρίζεται σε κανόνες και ευρετικές (heuristics), σε έλεγχο του ρυθμού και σε προσπάθειες μετατόπισης της άμυνας προς τα άναντη δίκτυα (upstream)[10]. Η αποτελεσματικότητα τους εξαρτάται πάρα πολύ από το σημείο της ανάπτυξης, δηλαδή τη πηγή (source), το πυρήνα (core), το θύμα (victim) ή τα κατανεμημένα δίκτυα (distributed) και από το πόσο δύσκολο καθίσταται να διακριθεί η κακόβουλη από τη νόμιμη κίνηση όταν η επίθεση μιμείται τη φυσιολογική συμπεριφορά, κάτι το οποίο επηρεάζει άμεσα τα ποσοστά των ψευδών θετικών και των ψευδών αρνητικών συναγερμών (false positives / false negatives)[10].

2.5 Περιορισμοί των κλασικών Συστήματα Ανίχνευσης Εισβολών

Αρχικά, τα κλασικά συστήματα ανίχνευσης των εισβολών βασίζονται κυρίως σε τρεις μεθοδολογίες. Πρώτον, στην ανίχνευση με υπογραφές (signature-based detection), στην ανίχνευση των ανωμαλιών (anomaly-based detection) και στην ανάλυση των πρωτοκόλλων με διατήρηση της κατάστασης (stateful protocol analysis).Τα βασικά πλεονεκτήματα και μειονεκτήματα των τριών αυτών προσεγγίσεων συνοψίζονται στο Πίνακα 2.2. Γενικότερα, παρότι αυτές οι προσεγγίσεις αποτελούν το κεντρικό κομμάτι της παραδοσιακής ανίχνευσης παρουσιάζουν αρκετούς περιορισμούς τόσο στο κομμάτι της κάλυψης του φάσματος των επιθέσεων όσο και στο κομμάτι της ακρίβειας και της αποδοτικότητας τους στα πραγματικά περιβάλλοντα. Ορισμένη από τους περιορισμούς αυτούς φαίνονται συγκριτικά και στο Πίνακα 2.2.

Πίνακας 2.2: Pros and cons of intrusion detection methodologies[11]

Signature-based (knowledge-based)	Anomaly-based (behavior-based)	Stateful protocol analysis (specification-based)
Pros • Simplest and effective method to detect known attacks. • Detail contextual analysis.	• Effective to detect new and unforeseen vulnerabilities. • Less dependent on OS. • Facilitate detections of privilege abuse.	• Know and trace the protocol states. • Distinguish unexpected sequences of commands.
Cons • Ineffective to detect unknown attacks, evasion attacks, and variants of known attacks. • Little understanding to states and protocols. • Hard to keep signatures/patterns up to date. • Time consuming to maintain the knowledge	• Weak profiles accuracy due to observed events being constantly changed. • Unavailable during rebuilding of behavior profiles. • Difficult to trigger alerts in right time.	• Resource consuming to protocol state tracing and examination. • Unable to inspect attacks looking like benign protocol behaviors. • Might incompatible to dedicated OSs or APs.

Πρώτα από όλα, θα αναλυθούν οι περιορισμοί της ανίχνευσης με υπογραφές. Ένας από τους βασικούς περιορισμούς είναι η αδυναμία εντοπισμού των άγνωστων μη καταγεγραμμένων επιθέσεων αλλά και παραλλαγών των γνωστών επιθέσεων, καθώς το σύστημα συγκρίνει την κίνηση που παρατηρεί με μοτίβα που έχουν ήδη κωδικοποιηθεί στη βάση γνώσης[11]. Επιπλέον, η βάση των υπογραφών πρέπει να ενημερώνεται συνεχώς ώστε να μπορέσει να ακολουθήσει την εμφάνιση των νέων τρωτών σημείων και εκμεταλλεύσεων, κάτι το οποίο αυξάνει σημαντικά το κόστος λειτουργίας και κάνει το σύστημα ευάλωτο σε περιόδους όπου η βάση δεν είναι πλήρως ενημερωμένη[11]. Στη συνέχεια, η ανάλυση των πρωτοκόλλων με διατήρηση της κατάστασης (stateful protocol analysis) στηρίζεται σε τυπικά μοντέλα πρωτοκόλλων και συχνά απαιτεί αυξημένους υπολογιστικούς πόρους για να παρακολουθεί την εξέλιξη των καταστάσεων. Ενώ, δυσκολεύεται να εντοπίσει τις επιθέσεις που μιμούνται πιστά τη κανονική χρήση των πρωτοκόλλων ή που αποκλίνουν ελάχιστα από τις προδιαγραφές. Αντίστοιχα, η ανάλυση των πρωτοκόλλων με τη διατήρηση της κατάστασης (stateful protocol analysis) βασίζεται σε ιδανικά ή τυπικά μοντέλα πρωτοκόλλων και χρειάζεται μεγάλο βαθμό υπολογιστικών πόρων για να παρακολουθεί την εξέλιξη των καταστάσεων, ενώ δυσκολεύεται να εντοπίσει επιθέσεις που μιμούνται πιστά την κανονική χρήση των πρωτοκόλλων ή αποκλίνουν ελάχιστα από τις προδιαγραφές[11].

Από την άλλη πλευρά, θα αναλυθούν οι περιορισμοί της ανίχνευσης των ανωμαλιών. Τα συστήματα της ανίχνευσης των ανωμαλιών αντιμετωπίζουν διαφορετικού τύπου περιορισμούς. Η ποιότητα της ανίχνευσης σε μεγάλο βαθμό εξαρτάται από την ακρίβεια των προφίλ της κανονικής συμπεριφοράς, τα οποία είναι σημαντικό να κατασκευάζονται από ιστορικά δεδομένα και να ανανεώνονται όταν αλλάζει ο τρόπος χρήσης των συστημάτων του δικτύου[11]. Σε περιβάλλοντα όπου η συμπεριφορά των χρηστών και των εφαρμογών αλλάζει συχνά, τα προφίλ αυτά μπορεί να παλιώσουν σε πολύ σύντομο χρονικό διάστημα, έτσι θα οδηγήσουν σε υψηλά ποσοστά στους ψευδείς θετικούς συναγερμούς(false positives) ή αντίστροφα, στους ψευδείς αρνητικούς συναγερμούς (false negatives), όταν μια επίθεση κρύβεται μέσα σε νέα και θεωρητικά νόμιμα μοτίβα κίνησης[11]. Επιπλέον, η διαδικασία της εκπαίδευσης και την ενημέρωσης των μοντέλων ανωμαλιών είναι στις περισσότερες περιπτώσεις υπολογιστικά απαιτητική και αυτό το γεγονός δυσκολεύει την εφαρμογή τους σε συστήματα με περιορισμένους πόρους ή σε περιβάλλοντα με πολύ υψηλούς ρυθμούς κίνησης[11].

Βέβαια, οι περιορισμοί δεν αφορούν μόνο τη μεθοδολογία της ανίχνευσης αλλά και τον τρόπο ανάπτυξης και τοποθέτησης των κλασικών συστημάτων ανίχνευσης των εισβολών στο δίκτυο. Γενικά, στα συστήματα ανίχνευσης των εισβολών που είναι βασισμένα σε κεντρικό υπολογιστή (Host-based IDS – HIDS) και τα οποία λειτουργούν πάνω στον ίδιο τον εξυπηρετητή ή στο τερματικό, η ανίχνευση γίνεται με περιορισμένη γνώση του ευρύτερου πλαισίου, καθώς το σύστημα βλέπει μόνο τα τοπικά γεγονότα και όχι τη συνολική εικόνα του δικτύου ενώ την ίδια στιγμή καταναλώνει τους πόρους του

ίδιου του συστήματος που προστατεύει[11]. Η σύγκριση των βασικών χαρακτηριστικών και περιορισμών των συστημάτων ανίχνευσης των εισβολών που είναι βασισμένα σε κεντρικό υπολογιστή (HIDS), των συστημάτων ανίχνευσης των εισβολών τα οποία είναι βασισμένα στο δίκτυο (NIDS),των συστημάτων ανίχνευσης των εισβολών για ασύρματα δίκτυα (WIDS) και των συστημάτων ανάλυσης της συμπεριφοράς κίνησης (Network Behavior Analysis – NBA) παρουσιάζονται συνοπτικά στο Πίνακα 2.3. Τα συστήματα ανίχνευσης των εισβολών τα οποία είναι βασισμένα στο δίκτυο (Network-based IDS – NIDS), παρακολουθούν την κίνηση στα τμήματα του δικτύου και δυσκολεύονται να αναλύσουν πλήρως την κίνηση σε πολύ υψηλές ταχύτητες, αντιμετωπίζουν μεγάλα ποσοστά ψευδών θετικών και αρνητικών συναγερμών σε πολύπλοκα περιβάλλοντα και δεν μπορούν εύκολα να επιθεωρήσουν την πλήρως κρυπτογραφημένη κίνηση[11]. Αντίστοιχα, τα συστήματα ανίχνευσης των εισβολών για ασύρματα δίκτυα (Wireless IDS – WIDS) είναι πολύ ευάλωτα στις επιθέσεις του φυσικού επιπέδου, όπως το επιτηδευμένο πνίξιμο του ασύρματου καναλιού (jamming), ενώ τα συστήματα της ανάλυσης συμπεριφοράς της κίνησης (Network Behavior Analysis – NBA) συχνά λειτουργούν με δεδομένα των ροών που συλλέγονται και στέλνονται σε τμήματα, κάτι το οποίο προκαλεί καθυστέρηση στην ανίχνευση[11].

Πίνακας 2.3: Comparisons of IDS technology types[11]

Item	Technology			
	HIDS	**NIDS**	**WIDS**	**NBA**
Components[a]	Agent: software (inline) MS: 1∼*n* DS: 1∼*n* (option)	Sensor: *n* (inline/passive) MS: 1∼*n* DS: 1∼*n* (option)	Sensor: *n* (passive) MS: 1∼*n* DS: 1∼*n* (option)	Sensor: *n* (most passive) MS: 1∼*n* (option) DS: optional
Detection scope of sensor/agent	Single host	Network subnet: *n* Host: *n*	WLAN: *n* WLAN client: *n*	Network subnet: *n* Host: *n*
Architecture[b]	MN or SN	MN	MN or SN	MN or SN
Strengths	Only HIDS can analyze end-to-end encrypted communications' activity.	Capable to analyze the broadest scopes of AP protocols	WIDS is more accurate due to its narrow focus. Only WIDS can supervise wireless protocol activity.	Superior detection powers at reconnaissance scanning, reconstruct malware infections and DoS attacks
Technology limitations[c]	• More challenging in detection accuracy due to a lack of context knowledge • Delays in alert generation and centralized reporting • Consume host resources • Conflict with existing security controls	• Cannot monitor wireless protocols • High false positive and false negative rates • Cannot detect attacks within encrypted traffic • No full analysis support under high loads.	• Cannot monitor AL, TL and NL protocol activities. • Cannot avoid evasion techniques. • Sensors are susceptible to physical jamming attacks. • Cannot compensate for insecure wireless protocols	• The major limitation is the delay in detection attacks, caused by transferring flow data to NBA in batches, but not in real time.
Security capabilities				
Information gathering	Network traffic, system calls, file system activity.	Hosts, OSs, APs, network traffic.	WLAN, devices (e.g., APs, clients).	Hosts, OS, services (IP, TCP, UDP, etc).
Logging	Reference (Stavroulakis and Stamp, 2010)	Reference (Stavroulakis and Stamp, 2010)	Reference (Stavroulakis and Stamp, 2010)	Reference (Stavroulakis and Stamp, 2010)
Detection methodology[d]	SD and AD (combined)	SD (major), AD and SPA	AD (major), SD and SPA	AD (major), SPA
Type of suspicious events detected	AL, TL and NL network traffic, event logs (e.g., application activities, file system activities), system logs (e.g., configurations, OS activity)	AL, TL, NL and HW reconnaissance and attacks, unexpected AP services, policy violations	Wireless protocol activity, insecure WLAN and devices, DoS attacks, network scanning, policy violations	AL, TL, NL anomalous traffic flows (DoS attacks, malware) unexpected AP services, network scanning, policy violations

[a] Components: management server (MS), database server (DS).
[b] Network architecture: managed networks (MN), standard networks (SN).
[c] Technology limitations: application (AP), application layer (AL), transport layer (TL), network layer (NL), hardware (HW), operating system (OS).
[d] Detection methodology: signature-based (SD), anomaly-based (AD), stateful protocol analysis (SPA).

Οι παραπάνω περιορισμοί γίνονται ιδιαίτερα εμφανείς στα σενάρια των DDoS, όπου η αξιολόγηση μιας άμυνας συνδέεται άμεσα με τη μείωση των ψευδών θετικών συναγερμών (false positives) και των ψευδών αρνητικών (false negatives)[12]. Σε πολλές περιπτώσεις, κάποιες απλές προσεγγίσεις βασισμένες σε ρυθμό ή όρια (rate-limiting / threshholding) ή σε στατικά μοτίβα (static patterns) μπορεί να οδηγήσουν είτε σε αυξημένους ψευδείς συναγερμούς κατά τη διάρκεια των νόμιμων αιχμών της κίνησης (flash crowds) είτε σε μειωμένη ανίχνευση όταν η επίθεση προσαρμόζει τη συμπεριφορά της (adaptive behavior)[12]. Επιπλέον, ειδικά στις επιθέσεις DDoS στο επίπεδο της εφαρμογής (application layer DDoS) η κίνηση μπορεί να μιμείται νόμιμα αιτήματα (legitimate requests), γεγονός το οποίο

καθιστά τη διάκριση από φυσιολογική δραστηριότητα (normal activity) πολύ δύσκολη και αυξάνει την ανάγκη για πιο ευέλικτες και συνδυαστικές στρατηγικές ανίχνευσης και άμυνας[12].

Συνολικά, επισημαίνεται ότι κανένας από τους κλασικούς τύπους των συστημάτων ανίχνευσης των εισβολών δεν μπορεί να προσφέρει την ίδια στιγμή πλήρη κάλυψη σε ετερογενή σενάρια απειλών, χαμηλό ποσοστό ψευδών συναγερμών και υψηλή αποδοτικότητα σε κλίμακα ιδιαίτερα σε περιβάλλοντα με κρυπτογραφημένη τη κίνηση, σε ασύρματες υποδομές και σε δυναμικά πρότυπα χρήσης. Ωστόσο, όλοι αυτοί οι περιορισμοί δίνουν κίνητρο για την εξέλιξη των συστημάτων ανίχνευσης των εισβολών με νέες πιο ευέλικτες, υβριδικές προσεγγίσεις οι οποίες συνδυάζουν διαφορετικές μεθοδολογίες της ανίχνευσης και αξιοποιούν τις προηγμένες τεχνικές της Μηχανικής Μάθησης χρησιμοποιώντας και τις πλατφόρμες Παρακολούθησης της Ασφάλειας.

2.6 Από τα κλασικά Συστήματα Ανίχνευσης Εισβολών (IDS) στα συστήματα παρακολούθησης ασφάλειας (CM)

Βέβαια, όσο εξελίσσονται οι απειλές και κάνοντας χρήση στοχευμένων επιθέσεων, παραβιάζοντας μακροχρόνια τα συστήματα (APT), χρησιμοποιώντας κρυπτογράφηση και κατανεμημένες υποδομές έχουν κάνει τα κλασικά εργαλεία ασφάλειας, όπως τα μεμονωμένα συστήματα ανίχνευσης εισβολών (IDS), τα τείχη προστασίας (firewalls) και τα λογισμικά προστασίας από ιούς (antivirus), να είναι ανεπαρκή όταν λειτουργούν απομονωμένα. Για αυτό το λόγο, υπάρχει η ανάγκη για νέες ολοκληρωμένες πλατφόρμες (Enterprise Security Analytics) οι οποίες δεν περιορίζονται σε μία απλή ανίχνευση με υπογραφές αλλά μαζεύουν δεδομένα από πολλές πηγές, όπως για παράδειγμα αρχεία καταγραφής συστημάτων, ροές δικτύων, γεγονότα από εφαρμογές και υποδομές υπολογιστικού νέφους και εφαρμόζουν προηγμένες τεχνικές ανάλυσης[13]. Σε αυτή τη κατηγορία εντάσσονται τα συστήματα Παρακολούθησης της Ασφάλειας (CM), τα οποία μόνιμα παρατηρούν και αξιολογούν τους κινδύνους σε πραγματικό ή σχεδόν πραγματικό χρόνο ώστε να διατηρήσουν την ασφάλεια του δικτύου.

Οπότε, τα Συστήματα Παρακολούθησης της Ασφάλειας (CM) μπορούν να θεωρηθούν η φυσική εξέλιξη των κλασικών συστημάτων ανίχνευσης των εισβολών (IDS) καθώς εξελίσσονται στις πιο καινούργιες πλατφόρμες Παρακολούθησης της Ασφάλειας, αντίστοιχες με τα σύγχρονα συστήματα διαχείρισης των συμβάντων και των πληροφοριών ασφάλειας (Security Information and Event Management – SIEM) και τις πλατφόρμες της αναλυτικής ασφάλειας σε επιχειρησιακό επίπεδο (enterprise security analytics)[13]. Τα συστήματα Παρακολούθησης της Ασφάλειας, αντί να επεξεργάζονται μόνο πακέτα ή γεγονότα σε επίπεδο δικτύου (host), συγκεντρώνουν και αποθηκεύουν μεγάλο όγκο από ετερογενή δεδομένα, τα οποία είναι αρχεία καταγραφής από λειτουργικά συστήματα, εφαρμογές και βάσεις δεδομένων ή ροές δικτύου (flows) και γεγονότα που προέρχονται από συσκευές ασφαλείας (firewall, IDS/IPS), καθώς και πληροφορίες από τη διαχείριση των ταυτοτήτων και των προβλέψεων[13].

Στη διπλωματική αυτή τα συστήματα Παρακολούθησης της Ασφάλειας (CM) αξιοποιούνται κυρίως για την παρακολούθηση και τον έγκαιρο εντοπισμό των εξάρσεων DoS / DDoS στη δικτυακή κίνηση, με τη χρήση των μοντέλων ταξινόμησης και των μηχανισμών ειδοποίησης (alerting). Έτσι, τα συστήματα παρακολούθησης της ασφάλειας αναλύουν και ενοποιούν αυτές τις πηγές ώστε να μπορέσουν να δέσουν μεταξύ τους συμβάντα που αρχικά φαίνονται άσχετα και με αυτό το τρόπο να μπορέσουν να αποκαλυφθούν πιο σύνθετα σενάρια επίθεσης, τα οποία ένα μεμονωμένο σύστημα ανίχνευσης εισβολών (IDS) θα ήταν πολύ δύσκολο να εντοπίσει.

Παράλληλα, τα σύγχρονα συστήματα Παρακολούθησης της Ασφάλειας, ενσωματώνουν όλο και πιο συχνά στις λειτουργίες τους τεχνικές Μηχανικής Μάθησης (Machine Learning) και προηγμένης αναλυτικής (security analytics), με στόχο να μπορούν να ανιχνεύουν τα πρότυπα συμπεριφοράς των

χρηστών και των συστημάτων, τις αποκλίσεις από την κανονικότητα και τις αργά εξελισσόμενες επιθέσεις[13]. Γενικότερα, η εστίαση μετατοπίζεται από τις απλές, κανονο-βασισμένες προσεγγίσεις σε πιο έξυπνα μοντέλα που πατάνε πάνω σε υποδομές μεγάλου όγκου δεδομένων (big data), αλγορίθμους ταξινόμησης και ανίχνευσης των ανωμαλιών αλλά και σε σύγχρονες τεχνικές οπτικοποίησης, ώστε να διευκολύνεται όσο το δυνατόν περισσότερο η δουλειά των αναλυτών ασφαλείας[13]. Έτσι, τα σύγχρονα συστήματα Παρακολούθησης της Ασφάλειας (CM) δεν περιορίζονται απλώς στο να κοιτάνε τι συνέβη εκ των υστέρων αλλά προσπαθούν να προσφέρουν μια συνολική εικόνα της κατάστασης επίγνωσης (situational awareness) για το τι συμβαίνει αυτή τη στιγμή στο κάθε οργανισμό.

Οπότε, σε σχέση με τα κλασικά συστήματα ανίχνευσης των εισβολών (IDS) που αναλύθηκαν στις προηγούμενες ενότητες τα σύγχρονα Συστήματα Παρακολούθησης της Ασφάλειας (CM) αντιμετωπίζουν τα κλασικά συστήματα ανίχνευσης εισβολών ως μια από τις πολλές πηγές πληροφορίας και όχι ως μια αυτόνομη λύση. Οι συναγερμοί (alerts) από τα συστήματα ανίχνευσης των εισβολών στο επίπεδο του δικτύου (network-based IDS) και στο επίπεδο του κεντρικού υπολογιστή (host-based IDS), τα αρχεία καταγραφής του τείχους προστασίας (firewall logs), οι καταγραφές του ελέγχου πρόσβασης (access control logs) και οι μετρήσεις από τις υποδομές του υπολογιστικού νέφους και της εικονικοποίησης (cloud and virtualization infrastructures) συγκεντρώνονται σε ένα κεντρικό περιβάλλον ανάλυσης όπου μπορούν να συσχετιστούν με τις επιχειρησιακές πληροφορίες (business information), όπως τα κρίσιμα πληροφοριακά περιουσιακά στοιχεία (critical assets), οι κατηγορίες των χρηστών (user categories) και οι επιχειρησιακές διαδικασίες (business processes)[13].

Με αυτό τον τρόπο, η μετάβαση από τα κλασικά Συστήματα Ανίχνευσης των Εισβολών (IDS) στα συστήματα Παρακολούθησης της Ασφάλειας (CM) δεν είναι απλώς μια τεχνική αναβάθμιση αλλά μία στροφή από την αποσπασματική ανίχνευση των μεμονωμένων τεχνικών συμβάντων προς τη συνεχή και ολιστική παρακολούθηση της ασφάλειας[13]. Αυτή η λογική αποτελεί και το βασικό θεμέλιο πάνω στο οποίο σχεδιάζεται και τελικά υλοποιείται η πλατφόρμα Παρακολούθησης της Κυβερνοασφάλειας (Cybersecurity Monitoring platform) στη συγκεκριμένη διπλωματική.

2.7 Ανάγκη χρήσης Μηχανικής Μάθησης (ML) στη δικτυακή ασφάλεια

Αρχικά, όπως φάνηκε και στις προηγούμενες ενότητες, τα κλασικά εργαλεία της ασφάλειας όπως για παράδειγμα τα τείχη προστασίας (firewalls), τα συστήματα ανίχνευσης των εισβολών βάση υπογραφών και τα παραδοσιακά λογισμικά προστασίας από ιούς (antivirus) δυσκολεύονται ολοένα και περισσότερο να ανταπεξέλθουν στις σύγχρονες επιθέσεις. Καθώς, οι επιτιθέμενοι αξιοποιούν τις τρωτότητες μηδενικής ημέρας (zero-day vulnerabilities), το πολυμορφικό και μεταμορφικό κακόβουλο λογισμικό (polymorphic and metamorphic malware), τις στοχευμένες επιθέσεις σε κρίσιμες υποδομές (Critical Infrastructures – CI), αλλά και τις τεχνικές απόκρυψης μέσα σε κρυπτογραφημένη κίνηση (hiding techniques within encrypted traffic) με αποτέλεσμα τα στατιστικά, βασισμένα σε κανόνες συστήματα (rule-based systems) να μην επαρκούν για την έγκαιρη αλλά και αξιόπιστη ανίχνευση των νέων η ελαφρώς τροποποιημένων μορφών γνωστών επιθέσεων[14]. Βέβαια, παρότι οι παραπάνω προκλήσεις αφορούν συνολικά την δικτυακή ασφάλεια, η πειραματική αξιολόγηση της εργασίας επικεντρώνεται σε σενάρια DoS / DDoS, όπου ο όγκος και η ένταση της κίνησης αποτελούν τα κυρίαρχα χαρακτηριστικά.

Σε αυτό το κομμάτι, η Μηχανική Μάθηση (ML) αποτελεί μια φυσική εξέλιξη ως ένα επόμενο στάδιο στο κομμάτι της ενίσχυσης της δικτυακής ασφάλειας. Σε αντίθεση με τα παλαιότερα συστήματα ανίχνευσης που στηρίζονται αποκλειστικά σε χειροποίητους κανόνες και υπογραφές, τα μοντέλα της Μηχανικής Μάθησης μπορούν να μαθαίνουν από δεδομένα, είτε αυτά είναι ετικετοποιημένα (supervised learning) είτε όχι (unsupervised learning) και να ανακαλύπτουν περίπλοκα μοτίβα τα οποία

διαχωρίζουν την κανονική από την κακόβουλη συμπεριφορά[14]. Έτσι, είναι δυνατό να εντοπίζονται οι μη γνωστές επιθέσεις ή οι παραλλαγές γνωστών επιθέσεων, οι οποίες δεν έχουν ακόμη καταγραφεί σε μία βάση υπογραφών.

Επομένως, η ανάγκη για τέτοιες προσεγγίσεις μεγαλώνει διότι τα σύγχρονα δίκτυα και οι κρίσιμες υποδομές παράγουν ολοένα και μεγαλύτερους όγκους ετερογενών δεδομένων. Μέσα από τις ροές του δικτύου (network flows), τα αρχεία καταγραφής των συστημάτων (system logs), τη τηλεμετρία από τα βιομηχανικά συστήματα ελέγχου (Industrial Control Systems – ICS, Supervisory Control and Data Acquisition – SCADA), τα δεδομένα των αισθητήρων από τα έξυπνα δίκτυα ενέργειας (smart grids) καθώς και δεδομένα από τα οχήματα και τις υποδομές των μεταφορών (Vehicular Ad hoc Networks – VANETs)[14]. Έτσι, σε έναν τέτοιο όγκο πολύπλοκων δεδομένων η χρήση των στατικών κανόνων ή η χειροκίνητη ανάλυση είναι πρακτικά αδύνατη, ενώ είναι πολύ πιο εύκολο με αυτό το τρόπο να χαθούν και κρίσιμα σήματα της επίθεσης μέσα σε όλων αυτό το θόρυβο[14].

Σε πρόσφατες αναλύσεις, η Μηχανική Μάθηση παρουσιάζεται ως η πιο σημαντική τεχνολογία σε αρκετές εφαρμογές ασφάλειας, όπως για παράδειγμα στην ανίχνευση των εισβολών σε δίκτυα κρίσιμων υποδομών (intrusion detection in critical infrastructure networks), στη προστασία των βιομηχανικών συστημάτων ελέγχου (Industrial Control Systems – ICS) και στα συστήματα εποπτικού ελέγχου και συλλογής Δεδομένων (Supervisory Control and Data Aqcuisition – SCADA), στην ασφάλεια στα προσωρινά αυτο-οργανωμένα δίκτυα οχημάτων (Vehicular Ad hoc Networks – VANETs) και στην ανάλυση των κακόβουλων λογισμικών (malware analysis)[14]. Σε όλες αυτές τις περιπτώσεις, ο ρόλος της Μηχανικής Μάθησης είναι να εκπαιδεύει τους ταξινομητές και ανιχνευτές των ανωμαλιών (classifiers and anomaly detectors) πάνω σε μεγάλα σύνολα πραγματικών δεδομένων της κίνησης και των συμβάντων, ώστε να μπορεί να εντοπίζει τις πιο διακριτικές και μικρές αποκλίσεις από τη φυσιολογική λειτουργία, τις οποίες ένα κλασικό σύστημα ανίχνευσης των εισβολών (IDS) πιθανότατα θα προσπερνούσε χωρίς να τις θεωρήσει ύποπτες.

Βέβαια, η Μηχανική Μάθηση έχει και αυτή προβλήματα. Τα μοντέλα της ειδικά όταν χρησιμοποιούνται σε περιβάλλοντα με έξυπνους επιτιθέμενους, οι οποίοι μπορούν και προσαρμόζονται στις αλλαγές των καταστάσεων, μπορούν να γίνουν τα ίδια στόχος αντιπαραθετικής μάθησης (adversarial machine learning) μέσα από επιθέσεις όπως η δηλητηρίαση των δεδομένων εκπαίδευσης (data poisoning) ή η σκόπιμη παραμόρφωση των εισόδων (evasion attacks), με στόχο τη παραπλάνηση του ταξινομητή[14]. Επιπλέον, για να αξιοποιηθούν αυτά τα μοντέλα προϋποθέτει πως τα σύνολα εκπαίδευσης είναι μεγάλα και αντιπροσωπευτικά και πως σε αυτά γίνεται συχνή επανεκπαίδευση καθώς τα πρότυπα της κίνησης αλλάζουν (concept drift). Ακόμη, χρειάζονται εξειδικευμένες γνώσεις τόσο στη Μηχανική Μάθηση όσο και στην κυβερνοασφάλεια, με αυτό το τρόπο τα δεδομένα θα σχεδιάζονται, θα ελέγχονται και θα ερμηνεύονται σωστά[14].

Κλείνοντας, αυτό που προκύπτει είναι πως η συνεχόμενη αύξηση στη πολυπλοκότητα των επιθέσεων, ο τεράστιος όγκος των δεδομένων της ασφάλειας και οι περιορισμοί των καθαρά μηχανιστικών προσεγγίσεων που βασίζονται μόνο σε υπογραφές κάνουν τη Μηχανική Μάθηση όχι απλώς μια καλή επιλογή αλλά ένα απαραίτητο εργαλείο για τη σύγχρονη δικτυακή ασφάλεια. Στο πλαίσιο αυτό εντάσσεται και η συγκεκριμένη διπλωματική, η οποία αξιοποιεί τους αλγορίθμους της Μηχανικής Μάθησης πάνω στα δεδομένα των ροών του δικτύου, με στόχο την υλοποίηση μια πρακτικής πλατφόρμας Παρακολούθησης της Ασφάλειας.

2.8 Επίλογος κεφαλαίου

Συνοψίζοντας, στο κεφάλαιο αυτό ετοιμάστηκε το θεωρητικό υπόβαθρο της ανίχνευσης των επιθέσεων στα σύγχρονα δίκτυα. Πρώτα, αναλύθηκαν οι θεμελιώδεις αρχές της κυβερνοασφάλειας, οι οποίες είναι η τριάδα Εμπιστευτικότητα, Ακεραιότητα και Διαθεσιμότητα (Condidentiality, Integrity, Availability - CIA), η αυθεντικοποίηση, η λογοδοσία, η ιδιωτικότητα και οι βασικές αρχές της σχεδίασης των ασφαλών δικτύων δηλαδή η άμυνα σε βάθος, η ελάχιστη αναγκαία πρόσβαση και η τμηματοποίηση του δικτύου, ώστε να γίνει σαφές τι προσπαθούμε να προστατέψουμε και με ποιο τρόπο. Στη συνέχεια, αναφέρθηκαν τα στάδια εξέλιξης μιας επίθεσης, δηλαδή η φάση αναγνώρισης (reconnaissance), η φάση της σάρωσης (scanning) και η φάση της επίθεσης (attack) και οι βασικές κατηγορίες της κακόβουλης δραστηριότητας, δείχνοντας με αυτό το τρόπο πως αυτή η ταξινόμηση συνδέεται άμεσα με το τρόπο που σχεδιάζονται και αξιολογούνται τα δικτυακά συστήματα ανίχνευσης των εισβολών (NIDS) και οι πλατφόρμες Παρακολούθησης της Ασφάλειας (CM) καθώς και με τη χρήση των συνόλων δεδομένων (datasets) όπως το CIC-DDoS2019 που χρησιμοποιήθηκε στην παρούσα εργασία.

Ακολούθως, έγινε αναλυτική αναφορά στα παραδοσιακά Συστήματα Ανίχνευσης των Εισβολών (IDS), στις δύο κλασικές προσεγγίσεις ανίχνευσης, δηλαδή στην ανίχνευσης με υπογραφές και στην ανίχνευση των ανωμαλιών, στις υβριδικές αρχιτεκτονικές και κυρίως στους πρακτικούς περιορισμούς τους στα πραγματικά και δυναμικά περιβάλλοντα. Πάνω σε αυτό το πλαίσιο παρουσιάστηκε η μετάβαση από τα μεμονωμένα σύστημα ανίχνευσης των εισβολών (IDS) προς τις πιο ολοκληρωμένες λύσεις δηλαδή σε πλατφόρμες Παρακολούθησης της Ασφάλειας (CM), οι οποίες ενοποιούν ετερογενείς πηγές δεδομένων και αξιοποιούν προηγμένες τεχνικές ανάλυσης προσεγγίζοντας με αυτό τον τρόπο την λογική των σύγχρονων συστημάτων διαχείρισης των πληροφοριών και των συμβάντων της ασφάλειας (SIEM) και της αναλυτικής επεξεργασίας των δεδομένων ασφάλειας σε επιχειρησιακό επίπεδο (enterprise security analytics). Τέλος, τεκμηριώθηκε το πόσο σημαντικό είναι να ενσωματωθεί η Μηχανική Μάθηση στη δικτυακή ασφάλεια, τόσο για να διαχειριστεί τον όγκο και την πολυπλοκότητα των δεδομένων, όσο και για να ανιχνεύσει τις νέες, εξελισσόμενες απειλές. Πάνω σε αυτό το κομμάτι στηρίζεται το επόμενο κεφάλαιο όπου παρουσιάζονται οι αλγόριθμοι της Μηχανικής Μάθησης που αξιοποιούνται στη παρούσα διπλωματική.

Κεφάλαιο 3ο: Αλγόριθμοι Μηχανικής Μάθησης για προσδιορισμό και ανίχνευση επιθέσεων

3.1 Εισαγωγή κεφαλαίου

Αρχικά, στο προηγούμενο κεφάλαιο παρουσιάστηκαν οι βασικές έννοιες της κυβερνοασφάλειας, τα στάδια εξέλιξης μιας επίθεσης, οι κατηγορίες της κακόβουλης κίνησης, καθώς και οι δυνατότητες κα οι περιορισμοί των παραδοσιακών συστημάτων ανίχνευσης εισβολών (IDS) και των πλατφορμών Παρακολούθησης της Ασφάλειας (CM). Επίσης, το βασικό συμπέρασμα που προκύπτει είναι ότι στα σύγχρονα και δυναμικά περιβάλλοντα δικτύων τα κλασικά, βασισμένα σε κανόνες εργαλεία δεν επαρκούν από μόνα τους για να μπορέσουν να ανιχνεύσουν έγκαιρα και αξιόπιστα τις σύνθετες και εξελισσόμενες επιθέσεις. Σε αυτό το κομμάτι, συνεισφέρει η Μηχανική Μάθηση (ML) η οποία εμφανίζεται ως μία από τις βασικές τεχνολογίες ενίσχυσης των συστημάτων ανίχνευσης των επιθέσεων και των πλατφορμών Παρακολούθησης της Ασφάλειας. Ωστόσο, παρότι στο παρόν κεφάλαιο παρουσιάζονται γενικά οι επιβλεπόμενοι αλγόριθμοι ταξινόμησης για την ανίχνευση της κακόβουλης δραστηριότητας, η πειραματική αξιολόγηση και η υλοποίηση της συγκεκριμένης διπλωματικής εξειδικεύονται σε σενάρια DoS / DDoS, σύμφωνα με το σύνολο δεδομένων CIC-DDoS2019.

Σκοπός του κεφαλαίου αυτού είναι να παρουσιάσει με αναλυτικό τρόπο τους βασικούς επιβλεπόμενους αλγόριθμους ταξινόμησης που χρησιμοποιούνται για την ανίχνευση της κακόβουλης δραστηριότητας πάνω σε δεδομένα ροών του δικτύου, με έμφαση στα σενάρια των επιθέσεων άρνησης της εξυπηρέτησης (DoS / DDoS) και στους αλγορίθμους που αξιοποιούνται πρακτικά στη συγκεκριμένη διπλωματική εργασία. Αρχικά, γίνεται μια σύντομη αναφορά σε δύο κλασικούς αλγορίθμους της επιβλεπόμενης μάθησης, τη λογιστική παλινδρόμηση (Logistic Regression) και τις μηχανές των διανυσμάτων στήριξης (Support Vector Machines – SVM) προκειμένου να διαμορφωθεί ένα θεωρητικό υπόβαθρο, ώστε να γίνει σαφές το πώς δημιουργείται το πρόβλημα της ανίχνευσης επιθέσεων και πώς αυτό μπορεί αυτό να διατυπωθεί και να αντιμετωπιστεί ως πρόβλημα ταξινόμησης.

Στη συνέχεια, το κεφάλαιο εστιάζει στους τρεις βασικούς αλγορίθμους των δέντρων και των συνόλων των δέντρων δηλαδή, στα Δέντρα Απόφασης (Decision Trees), στα Τυχαία Δάση (Random Forest) και στον αλγόριθμο Extreme Gradient Boosting (XGBoost). Οι δύο τελευταίοι αποτελούν και τους κύριους αλγόριθμους τους παρούσας διπλωματικής, καθώς χρησιμοποιούνται για την εκπαίδευση των μοντέλων ανίχνευσης των επιθέσεων πάνω στα δεδομένα των ροών του δικτύου. Για κάθε αλγόριθμο παρουσιάζονται η βασική αρχή της λειτουργίας του, τα πλεονεκτήματα και οι αδυναμίες του, καθώς και οι λόγοι για τους οποίους είναι κατάλληλος ή όχι για χρήση στα πλαίσια των πλατφορμών Παρακολούθησης της Ασφάλειας (CM). Κλείνοντας, περιγράφονται οι μετρικές της αξιολόγησης δηλαδή, η ακρίβεια (Accuracy), η ακρίβεια των θετικών προβλέψεων (Precision), η ανάκληση (Recall), ο δείκτης F1 (F1-score),το εμβαδόν κάτω από την καμπύλη ROC (ROC-AUC) και τέλος ο πίνακας σύγχυσης (Confusion Matrix), τα οποία χρησιμοποιούνται για να υπολογισθεί η απόδοση των μοντέλων, προετοιμάζοντας με αυτό το τρόπο την κατάσταση για τα πειραματικά αποτελέσματα που θα αναλυθούν στα επόμενα κεφάλαια.

3.2 Βασικοί επιβλεπόμενοι αλγόριθμοι ταξινόμησης

Αρχικά, όπως αναφέρθηκε και στο εισαγωγικό κομμάτι του κεφαλαίου, το πρόβλημα της ανίχνευσης των επιθέσεων στα δεδομένα των ροών του δικτύου διατυπώνεται συνήθως ως πρόβλημα ταξινόμησης (classification). Καθώς, για κάθε ροή ή για κάθε εγγραφή της κίνησης στόχος είναι να αποφασιστεί αν

πρόκειται για κανονική δραστηριότητα ή αν είναι κάποια μορφή επίθεσης, όπως για παράδειγμα η δυαδική ταξινόμηση (binary classification) ή μπορεί να είναι και μία πιο εξειδικευμένη κατηγορία επίθεσης όπως η πολυ-κλασσική ταξινόμηση (multi-class classification). Στην εργασία αυτή, το πειραματικό σκέλος εξετάζει τόσο τη δυαδική διάκριση (benign vs DDoS) όσο και τη πολυ-κλασσική ταξινόμηση ανά τύπο DDoS, ανάλογα με τη διαμόρφωση των ετικετών (labels) στο CIC-DDoS2019.

Επίσης, στα επιβλεπόμενα μοντέλα Μηχανικής Μάθησης (supervised learning) η εκπαίδευση βασίζεται σε ετικετοποιημένα δεδομένα (labeled data), δηλαδή για κάθε δείγμα είναι γνωστό εκ των προτέρων αν είναι κανονικό (benign) ή κακόβουλο (malicious). Έτσι, η διαδικασία αυτή οργανώνεται τυπικά σε κάποιες φάσεις εκπαίδευσης και εξαγωγής συμπερασμάτων (training and inference) όπου γίνεται διαχωρισμός των δεδομένων σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου, ώστε ο αλγόριθμος να μάθει μια συνάρτηση προσέγγισης (function approximation) και να ελεγχθεί η ικανότητά του να γενικεύει πάνω σε νέα δεδομένα[15].

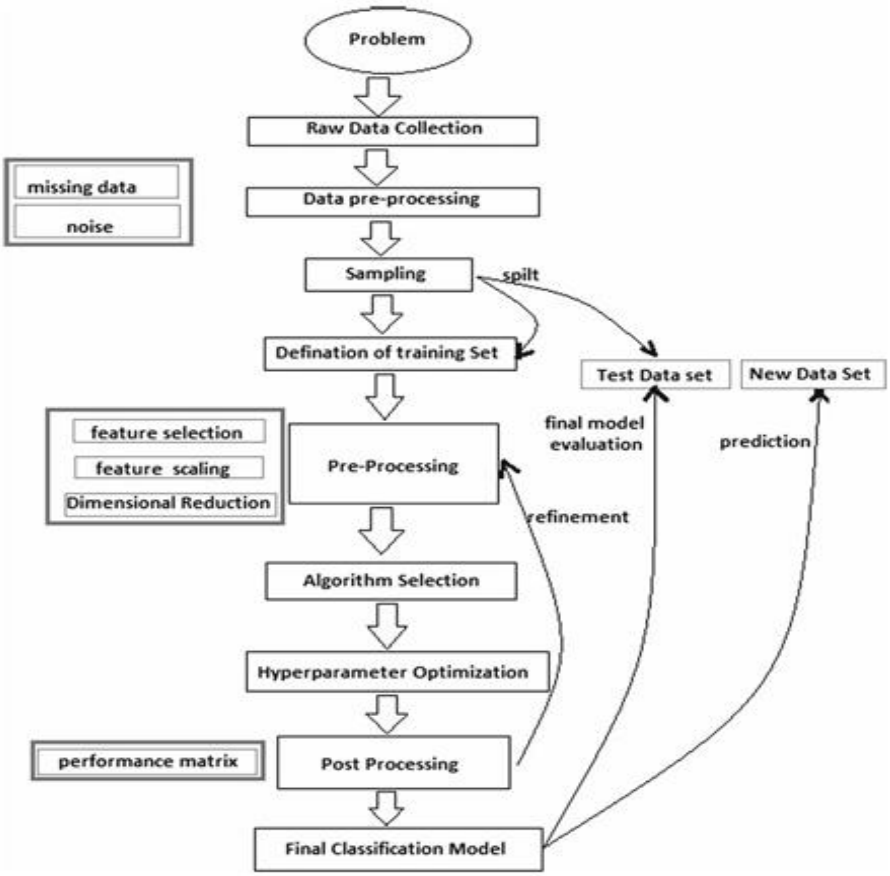
Ειδικότερα, υπάρχει ένα πολύ μεγάλο εύρος επιβλεπόμενων αλγορίθμων της Μηχανικής Μάθησης (supervised Machine Learning) οι οποίοι έχουν χρησιμοποιηθεί για ανίχνευση των εισβολών, του κακόβουλου λογισμικού (malware), της ανεπιθύμητης αλληλογραφίας (spam) και άλλων μορφών επιθέσεων. Στην πλευρά της ταξινόμησης (classification), χρησιμοποιούνται οι κλασικοί ρηχοί αλγόριθμοι, όπως για παράδειγμα ο Ναιβός Μπέυζ (Naive Bayes), οι Μηχανές Διανυσμάτων Στήριξης (Support Vector Machines - SVM) και ο αλγόριθμος k-Πλησιέστερων Γειτόνων (k-Nearest Neighbors – k-NN)[15]. Ενώ, για τα προβλήματα της παλινδρόμησης (regression) χρησιμοποιούνται μεταξύ άλλων αντίστοιχων εκδοχών αλγορίθμων η Λογιστική Παλινδρόμηση (Logistic Regression – LR) και τα Τυχαία Δάση (Random Forest – RF)[15].

Τέλος, στο κεφάλαιο αυτό πριν αναλυθούν οι βασικοί αλγόριθμοι δέντρων (Decision Trees, Random Forest, XGBoost) παρουσιάζονται συνοπτικά οι δύο χαρακτηριστικοί επιβλεπόμενοι ταξινομητές που χρησιμοποιούνται πιο συχνά δηλαδή, η Λογιστική Παλινδρόμηση (LR) και οι Μηχανές Διανυσμάτων Στήριξης (SVM). Με αυτό το τρόπο, αναφέρονται κάποιες βασικές έννοιες όπως το γραμμικό και μη γραμμικό όριο απόφασης (decision boundary), η πιθανότητα, το περιθώριο διαχωρισμού (margin) αλλά και ζητήματα όπως η υπερπροσαρμογή (overfitting) και η ικανότητα γενίκευσης (generalization)[15].

3.2.1 Λογιστική Παλινδρόμηση (Logistic Regression)

Πρώτα από όλα, η Λογιστική Παλινδρόμηση (Logistic Regression) είναι ένας από τους πιο κλασικούς επιβλεπόμενους αλγόριθμους ταξινόμησης (supervised classification algorithms) και χρησιμοποιείται όταν η μεταβλητή στόχος είναι δυαδική, δηλαδή μπορεί να πάρει δύο τιμές, για κανονική κίνηση το 0 και για μια επίθεση το 1. Σε αντίθεση με τη Γραμμική Παλινδρόμηση (Linear Regression), η οποία προβλέπει μια συνεχόμενη τιμή, η Λογιστική Παλινδρόμηση μοντελοποιεί κατευθείαν τη πιθανότητα ένα δείγμα να ανήκει στη θετική κλάση χρησιμοποιώντας μια σιγμοειδή (logistic) συνάρτηση για να συμπιέσει τις τιμές στο διάστημα μεταξύ του 0 και του 1[16].

Γενικότερα, το πλαίσιο λειτουργίας ενός επιβλεπόμενου μοντέλου ταξινόμησης όπως η Λογιστική Παλινδρόμηση παρουσιάζεται ενδεικτικά στο Σχήμα 3.1. Αρχικά, ξεκινώντας από τον ορισμό του προβλήματος, τα δεδομένα συλλέγονται και προ-επεξεργάζονται, δηλαδή αντιμετωπίζονται οι ελλιπείς τιμές, ο θόρυβος , κ.α. Στην συνέχεια, ακολουθεί η δειγματοληψία και ο διαχωρισμός στο σύνολο εκπαίδευσης και δοκιμής, η επιλογή και η κλιμάκωση των χαρακτηριστικών, η επιλογή του αλγόριθμου και ο προσδιορισμός των υπερπαραμέτρων. Τέλος, το μοντέλο εκπαιδεύεται σε ένα σύνολο εκπαίδευσης (training set) και αξιολογείται σε ένα ανεξάρτητο σύνολο ελέγχου (independent test set) και εφόσον κριθεί ικανοποιητικό, χρησιμοποιείται για τη πρόβλεψη σε νέα δεδομένα[16].



Σχήμα 3.1: Working flowchart of supervised classification model[16]

Πρακτικά, ο αλγόριθμος αυτός υπολογίζει ένα γραμμικό συνδυασμό των χαρακτηριστικών εισόδου (features) και στη συνέχεια εφαρμόζει τη σιγμοειδή συνάρτηση σε αυτό το άθροισμα. Έτσι, η έξοδος μπορεί να ερμηνευτεί ως πιθανότητα, δηλαδή αν η πιθανότητα είναι πάνω από ένα προκαθορισμένο κατώφλι (threshold) η ροή χαρακτηρίζεται ως επίθεση διαφορετικά είναι μια κανονική κίνηση στο δίκτυο. Αυτό κάνει τη Λογιστική Παλινδρόμηση ιδιαίτερα εύκολη στην ερμηνεία, καθώς κάθε συντελεστής (weight) αποτυπώνει το πόσο και προς ποια κατεύθυνση το αντίστοιχο χαρακτηριστικό έχει τη πιθανότητα να ταξινομηθεί ως κακόβουλο[16]

Η λογική πίσω από την λήψη της απόφασης σε έναν δυαδικό ταξινομητή μπορεί να αποτυπωθεί σχηματικά και με ένα απλό δέντρο αποφάσεων όπως αυτό του Σχήματος 3.2. Βέβαια παρότι είναι ένα φαινομενικά απλό παράδειγμα, αποτυπώνει πολύ καθαρά τη διαδικασία των διαδοχικών ερωτημάτων και χαρακτηριστικών, τα οποία οδηγούν στις τελικές αποφάσεις. Αντίστοιχα, η Λογιστική Παλινδρόμηση υλοποιεί μια γραμμική αλλά μαθηματικά πιο οργανωμένη εκδοχή ενός τέτοιου ορίου απόφασης, στην οποία το τελικό αποτέλεσμα βασίζεται σε έναν γραμμικό συνδυασμό χαρακτηριστικών και σε ένα κατώφλι πιθανότητας[16].

Σχήμα 3.2: A decision tree to decide how to have food[16]

Ειδικότερα, η Λογιστική Παλινδρόμηση περιγράφεται ως ένα μοντέλο ταξινόμησης το οποίο εκτιμά τις πιθανότητες κλάσης και παρότι περιλαμβάνει τον όρο παλινδρόμηση δεν χρησιμοποιείται για την πρόβλεψη των συνεχών τιμών όπως η γραμμική παλινδρόμηση[16]. Η βασική της ιδιότητα είναι, ότι στηρίζεται σε μια στατιστική περιγραφή της σχέσης μεταξύ των χαρακτηριστικών εισόδου και της μεταβλητής στόχου και μπορεί να χειριστεί πολλαπλές αριθμητικές και κατηγορικές μεταβλητές (continuous and categorical features) εφόσον οι τελευταίες κωδικοποιηθούν κατάλληλα, για παράδειγμα με κωδικοποίηση one-hot (one-hot encording)[16].

Ένα βασικό πλεονέκτημα που φαίνεται και στην ανασκόπηση είναι ότι η Λογιστική Παλινδρόμηση δεν χρειάζεται απαραίτητα τεράστια σύνολα εκπαίδευσης για να λειτουργήσει ικανοποιητικά και η απόδοση της δεν εξαρτάται τόσο πολύ από το μέγεθος του συνόλου εκπαίδευσης (training set), σε αντίθεση με κάποιους πιο βαριούς αλγορίθμους[16]. Επίσης, είναι σχετικά γρήγορη στην εκπαίδευση, απλή στην υλοποίηση και δίνει ένα καθαρό γραμμικό όριο απόφασης (linear decision boundary), το οποίο μπορεί να αναλυθεί από τον αναλυτή ασφαλείας.

Από την άλλη πλευρά, ο βασικός περιορισμός της Λογιστικής Παλινδρόμησης είναι ότι θεωρεί πως υπάρχει μια γραμμική σχέση μεταξύ των χαρακτηριστικών και του λογάριθμου των πιθανοτήτων (log-odds). Αυτό σημαίνει ότι, όταν η πραγματική δομή των δεδομένων είναι έντονα μη γραμμική, το μοντέλο μπορεί να μην αποδώσει καλά, εκτός αν πριν γίνει μια σημαντική μη γραμμική προεπεξεργασία ή μηχανική χαρακτηριστικών (feature engineering)[16]. Όμως, σε προβλήματα όπως η ανίχνευση των επιθέσεων σε δεδομένα ροών του δικτύου, όπου τα πρότυπα της κακόβουλης συμπεριφοράς μπορεί να είναι δύσκολα, σύνθετα και μη γραμμικά, η Λογιστική Παλινδρόμηση χρησιμοποιείται συχνά ως το βασικό σημείο αναφοράς (baseline classifier) και όχι ως ο πιο ισχυρός τελικός ταξινομητής[16]. Παρόλα αυτά, είναι πολύ χρήσιμη καθώς είναι απλή, εύκολη στην ερμηνεία της και παράγει πιθανότητες οι οποίες μπορούν να χρησιμοποιηθούν μέσα σε μεγαλύτερες πλατφόρμες Παρακολούθησης της Ασφάλειας (CM), ενισχύοντας την διαδικασία λήψης των αποφάσεων.

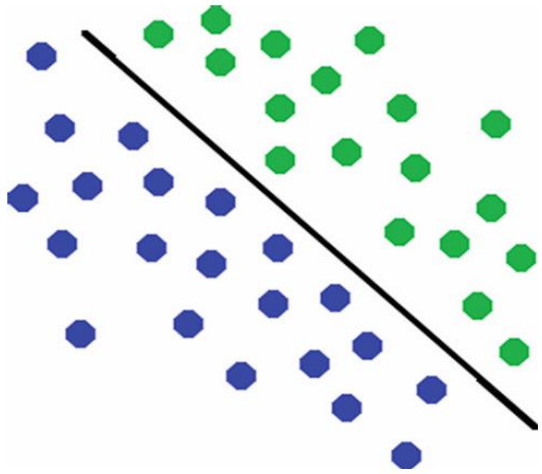
3.2.2 Μηχανές Διανυσμάτων Στήριξης (Support Vector Machines)

Αρχικά, οι Μηχανές των Διανυσμάτων Στήριξης (SVM) αποτελούν μια πιο προχωρημένη κατηγορία επιβλεπόμενων αλγορίθμων που μπορούν να χρησιμοποιηθούν τόσο για ταξινόμηση (classification) όσο και για παλινδρόμηση (regression), με την κύρια τους χρήση όμως να είναι στην ταξινόμηση. Η βασική ιδέα είναι ότι κάθε δείγμα αναπαρίσταται ως ένα σημείο σε ένα χώρο n διαστάσεων (n-dimensional feature space), όπου κάθε διάσταση αντιστοιχεί σε ένα χαρακτηριστικό (feature)[16]. Ο στόχος των

Μηχανών των Διανυσμάτων Στήριξης, είναι να βρει ένα υπερεπίπεδο (hyperplane) το οποίο θα διαχωρίζει τα σημεία των διαφορετικών κλάσεων με όσο το δυνατόν μεγαλύτερο περιθώριο (margin), δηλαδή να αφήνει χώρο ασφαλείας ανάμεσα στις κλάσεις[16].

Η λογική αυτή αποτυπώνεται χαρακτηριστικά στο Σχήμα 3.3, όπου παρουσιάζεται ένα δισδιάστατο παράδειγμα των Μηχανών των Διανυσμάτων Στήριξης. Το παράδειγμα αποτελείται από τα σημεία των δύο κλάσεων τα οποία κατανέμονται σε ένα χώρο δυο χαρακτηριστικών και σε ένα γραμμικό υπερεπίπεδο, δηλαδή στην ευθεία που διαχωρίζει τις δύο κλάσεις. Ακόμη, σε ένα πλήρες μοντέλο των Μηχανών των Διανυσμάτων Στήριξης η ακριβής θέση και ο προσανατολισμός αυτής της γραμμής καθορίζονται από τα διανύσματα της στήριξης (support vectors) και το μέγιστο περιθώριο (margin) μεταξύ των κλάσεων[16].

Επίσης, τα σημεία αυτά τα οποία βρίσκονται πιο κοντά στο υπερεπίπεδο και ουσιαστικά ορίζουν τη θέση του ονομάζονται διανύσματα στήριξης (support vectors). Αυτά είναι και τα πιο σημαντικά δείγματα, καθώς αν αλλάξουν αλλάζει και το όριο της απόφασης. Βέβαια, όταν τα δεδομένα μπορούν να διαχωριστούν γραμμικά, το πρόβλημα είναι σχετικά απλό, δηλαδή οι Μηχανές των Διανυσμάτων Στήριξης αναζητούν το υπερεπίπεδο που μεγιστοποιεί το περιθώριο μεταξύ των δύο κλάσεων. Ωστόσο, σε πολλές πραγματικές εφαρμογές, όπως και στη δικτυακή ασφάλεια τα δεδομένα δεν είναι γραμμικά διαχωρισμένα στον αρχικό χώρο των χαρακτηριστικών. Για αυτή την περίπτωση, οι Μηχανές των Διανυσμάτων Στήριξης χρησιμοποιούν τη τεχνική των πυρήνων (kernel trick), δηλαδή κάποιες συναρτήσεις πυρήνα (kernel functions) που μετασχηματίζουν τα αρχικά δεδομένα σε έναν χώρο υψηλότερων διαστάσεων, όπου εκεί είναι πιο εύκολο να βρεθεί ένα γραμμικό υπερεπίπεδο διαχωρισμού[16].



Σχήμα 3.3: A depiction of SVM classification with hyperplane[16]

Όπως αναφέρεται γενικότερα, οι Μηχανές των Διανυσμάτων Στήριξης έχουν ορισμένα σημαντικά πλεονεκτήματα. Καθώς, αποδίδουν πολύ καλά σε δεδομένα υψηλής διάστασης (high-dimensional data), ακόμα και όταν ο αριθμός των χαρακτηριστικών είναι μεγαλύτερος από τον αριθμό των δειγμάτων είναι αρκετά αποδοτικές στη μνήμη, διότι η συνάρτηση της απόφασης εξαρτάται μόνο από ένα υποσύνολο των δειγμάτων, δηλαδή από τα διανύσματα της στήριξης και όταν το υπερεπίπεδο και το περιθώριο έχουν οριστεί σωστά έχουν σαν αποτέλεσμα μια πολύ καλή ικανότητα γενίκευσης (generalization performance)[16]. Αυτά τα στοιχεία κάνουν τις Μηχανές των Διανυσμάτων Στήριξης μια πολύ καλή επιλογή σε προβλήματα όπου τα δεδομένα είναι πλούσια σε χαρακτηριστικά, όπως για παράδειγμα τα χαρακτηριστικά των ροών του δικτύου.

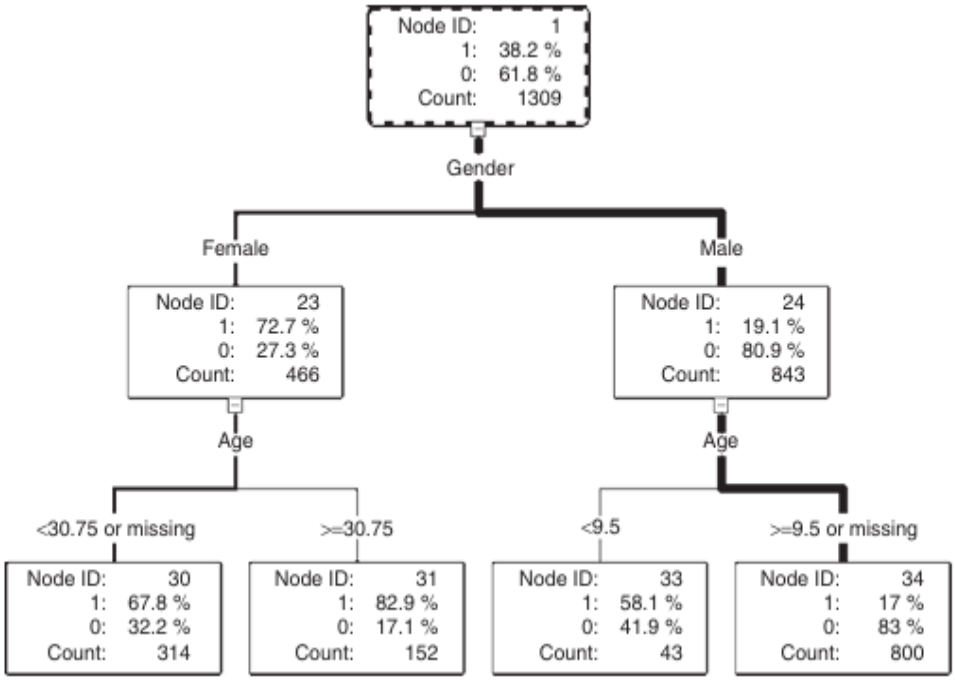
Βέβαια, παρά τα πλεονεκτήματα τους, οι Μηχανές των Διανυσμάτων Στήριξης παρουσιάζουν και ορισμένα μειονεκτήματα. Ένα βασικό ζήτημα είναι ο χρόνος εκπαίδευσης (training time), ο οποίος μπορεί να γίνει πολύ μεγάλος όταν το σύνολο δεδομένων (dataset) είναι μεγάλο και αυτό επηρεάζει αρνητικά τη δυνατότητα χρήσης τους σε συστήματα ανίχνευσης σε πραγματικό ή σχεδόν πραγματικό χρόνο (real-time or near-real-time)[16]. Επίσης, όταν οι κλάσεις επικαλύπτονται έντονα ή όταν τα δεδομένα περιέχουν πολύ θόρυβο, δηλαδή παρουσιάζουν θορυβώδεις και αλληλεπικαλυπτόμενες κλάσεις (noisy, overlapping classes), η απόδοση των Μηχανών των Διανυσμάτων Στήριξης μπορεί να μειωθεί. Τέλος, για να μπορέσουν οι Μηχανές των Διανυσμάτων Στήριξης να βγάλουν αξιόπιστες εκτιμήσεις πιθανοτήτων και όχι μόνο σκληρές αποφάσεις μίας κλάσης χρειάζονται συνήθως μερικές επιπλέον διαδικασίες, όπως οι τεχνικές της διασταυρωμένης επικύρωσης τύπου n-fold (n-fold cross-validation) οι οποίες αυξάνουν παραπάνω το υπολογιστικό κόστος[16].

Παρόλα αυτά, οι Μηχανές των Διανυσμάτων Στήριξης έχουν χρησιμοποιηθεί αρκετά σε μια μεγάλη γκάμα εφαρμογών, όπως στην ανάλυση των χρηματοοικονομικών δεδομένων, στη βιοπληροφορική, στην αναγνώριση του προσώπου και γενικότερα στην ταξινόμηση των εικόνες και των μοτίβων. Η ικανότητα τους να χειρίζονται πολυδιάστατα δεδομένα και να κατασκευάζουν ισχυρά μη γραμμικά όρια απόφασης μέσω των πυρήνων τις καθιστά μια πολύ καλή λύση για τα προβλήματα της ανίχνευσης των επιθέσεων σε ροές του δικτύου, καθώς εκεί οι επιθέσεις συχνά κρύβονται μέσα σε περίπλοκα πρότυπα της κίνησης[16]. Στο πλαίσιο της συγκεκριμένης διπλωματικής, οι Μηχανές των Διανυσμάτων Στήριξης δεν παρουσιάζονται ως το βασικό εργαλείο της υλοποίησης, αλλά ως ένας χαρακτηριστικός, κλασικός μη γραμμικός ταξινομητής. Η παρουσίασή τους βοηθάει στο να γίνουν πιο κατανοητές κάποιες έννοιες όπως για παράδειγμα το υπερεπίπεδο, το περιθώριο και η χρήση των πυρήνων, πριν φτάσουμε στις μεθόδους που βασίζονται στα δέντρα και στα σύνολα δέντρων (Random Forest, XGBoost) οι οποίες αποτελούν και τον βασικό κορμό της υλοποίησης.

3.3 Δέντρα Απόφασης (Decision Trees)

Πρώτα από όλα, τα Δέντρα Απόφασης (Decision Trees – DT) αποτελούν τους μηχανισμούς πρόβλεψης και ταξινόμησης του γενικού σκοπού και χρησιμοποιούνται κυρίως στη στατιστική, στη Μηχανική Μάθηση και στα συστήματα της τεχνικής νοημοσύνης[17]. Βασικό τους χαρακτηριστικό είναι ότι παράγουν αποτελέσματα που είναι ιδιαίτερα εύκολα στην ερμηνεία και προσφέρουν μια δενδροειδής, βήμα προς βήμα αναπαράσταση η οποία είναι διαισθητική και βοηθάει σημαντικά τόσο στην κατανόηση όσο και στη διάχυση των συμπερασμάτων του μοντέλου[17].

Επιπλέον, στο πυρήνα τους τα Δέντρα Απόφασης υλοποιούν μια διαδικασία αναδρομικής κατάτμησης (recursive subsetting) του συνόλου δεδομένων. Δηλαδή, το αρχικό σύνολο (root node) υποδιαιρείται σε υποσύνολα με βάση τις τιμές ενός ή περισσοτέρων από τα πεδία εισόδου (inputs / predictors), δημιουργώντας έτσι κόμβους (nodes / leaves) που σε κάθε επίπεδο του δέντρου είναι όσο γίνεται πιο ομοιογενείς ως προς τη μεταβλητή-στόχο και ταυτόχρονα πιο ανόμοιοι μεταξύ τους[17]. Στη συνέχεια, παρουσιάζεται ένα χαρακτηριστικό παράδειγμα το οποίο είναι η ανάλυση της επιβίωσης των επιβατών του Τιτανικού (Titanic) και απεικονίζεται ενδεικτικά στο Σχήμα 3.4. Αρχικά, ο κόμβος-ρίζα περιέχει τη συνολική κατανομή της επιβίωσης (δηλαδή το ποσοστό των επιζώντων και των μη επιζώντων στο σύνολο των 1.309 επιβατών), ενώ οι διαδοχικές διασπάσεις με βάση το φύλο (gender) και την ηλικία (age) οδηγούν σε κάποιες υποομάδες επιβατών με διαφορετικά ποσοστά επιβίωσης. Με αυτό το τρόπο, φαίνονται καθαρά τα συμφραζόμενα εφέ (contextual effects) δηλαδή ότι αλλάζει η πιθανότητα επιβίωσης από ομάδα σε ομάδα, για παράδειγμα το δέντρο δείχνει ότι οι γυναίκες και οι νεότεροι επιβάτες είχαν μεγαλύτερες πιθανότητες να σωθούν σε σχέση με τις άλλες κατηγορίες επιβατών[17].



Σχήμα 3.4: A decision tree illustrating analysis of survival in Titanic sinking[17]

Στο τυπικό σχήμα ενός Δέντρου Απόφασης, όπως φαίνεται και στο Σχήμα 3.4, κάθε εσωτερικός κόμβος αντιστοιχεί σε μια συνθήκη διαχωρισμού (partition) πάνω σε ένα πεδίο εισόδου, κάθε κλαδί αντιστοιχεί σε μία περιοχή τιμών του πεδίου αυτού και κάθε κόμβος-φύλλο αντιστοιχεί σε μία τελική κατηγορία ή τιμή της μεταβλητής στόχου. Τα δέντρα μπορούν να χειριστούν φυσικά όλους τους τύπους μεταβλητών όπως για παράδειγμα ονομαστικές, διατεταγμένες και συνεχείς μεταβλητές τόσο ως μεταβλητές στόχους (target) όσο και ως μεταβλητές εισόδου (inputs), ενώ υποστηρίζουν δυαδικούς (binary) αλλά και πολυκλαδικούς (multi-way) διαχωρισμούς ανάλογα με το πως ορίζονται τα κριτήρια της ομοιογένειας στους επιμέρους κόμβους[17]. Επιπλέον, προσφέρουν εύκολους τρόπους χειρισμού των ελλιπών τιμών (missing values), οι οποίες μπορούν είτε να ομαδοποιηθούν με παρόμοιες τιμές ως προς τη σχέση τους με τη μεταβλητή στόχο είτε να αντιμετωπιστούν σαν ξεχωριστή κατηγορία.

Στη συνέχεια, ένα από τα βασικά πλεονεκτήματα των δέντρων απόφασης είναι η ερμηνευσιμότητα και η δυνατότητα εξαγωγής των κανόνων. Κάθε μονοπάτι από τη ρίζα μέχρι ένα φύλλο μπορεί να μεταφραστεί σε έναν κατανοητό κανόνα της μορφής, αν (συνθήκες πάνω σε χαρακτηριστικά) τότε (πρόβλεψη) και το γεγονός αυτό κάνει τα μοντέλα των δέντρων μία πολύ καλή λύση όταν απαιτείται διαφάνεια στη λήψη των αποφάσεων[17]. Στο πλαίσιο των συστημάτων ανάλυσης και της ανίχνευσης των συμβάντων, δηλαδή αυτά που αφορούν τις ροές του δικτύου και τις επιθέσεις, η δυνατότητα να παρουσιαστεί η απόφαση του μοντέλου ως ένα σύνολο των λογικών και εύκολα κατανοητών στη πράξη κανόνων είναι πολύ σημαντική γιατί κάνει τα αποτελέσματα πιο αποδεκτά και πιο αξιοποιήσιμα τόσο από αναλυτές όσο και από απλούς χρήστες χωρίς κάποιο τεχνικό υπόβαθρο[17].

Βέβαια, παρά τα σημαντικά πλεονεκτήματά τους, τα μοναδικά (single) δέντρα είναι επιρρεπή στην υπερπροσαρμογή (overfitting) αν δεν τεθούν τα κατάλληλα κριτήρια τερματισμού και αν δεν γίνει ο έλεγχος της πολυπλοκότητας[17]. Για τον λόγο αυτό, έχουν αναπτυχθεί τεχνικές όπως οι κανόνες διακοπής (stopping rules), οι διαδικασίες επικύρωσης (validation) και οι μέθοδοι κλαδέματος (pruning), ώστε να εξασφαλίζεται ότι το δέντρο γενικεύει σωστά όταν βρίσκεται μπροστά σε νέα δεδομένα[17]. Η ίδια λογική χρησιμοποιείται και στα πολλαπλά δέντρα (multitree methods) όπου πολλά δέντρα

εκπαιδεύονται σε επαναδειγματοληπτημένα σύνολα και οι προβλέψεις τους συνδυάζονται. Τέτοιες προσεγγίσεις, όπως για παράδειγμα οι τεχνικές ενίσχυσης των μοντέλων (boosting) και τα Τυχαία Δάση (Random Forests) έχουν βελτιώσει πάρα πολύ την πρακτική απόδοση των δέντρων απόφασης και αποτελούν τη βάση για τους σύγχρονους, πιο ισχυρούς ταξινομητές που χρησιμοποιούνται ευρέως στα δεδομένα μεγάλης κλίμακας[17]. Στις επόμενες ενότητες, αυτές οι μέθοδοι αξιοποιούνται ως δομικό στοιχειό για την ανάπτυξη των πιο σύνθετων μοντέλων που μελετώνται στη συγκεκριμένη διπλωματική.

3.4 Τυχαία Δάση (Random Forests)

Πρώτα από όλα, τα Τυχαία Δάση (Random Forest – RF) ανήκουν στην οικογένεια των μεθόδων συνόλων (ensemble methods) και ειδικότερα των πολυδένδρινων μοντέλων (multitree methods), όπου η τελική πρόβλεψη προκύπτει από τον συνδυασμό πολλών δέντρων απόφασης αντί από ένα μόνο δέντρο[18]. Η βασική ιδέα είναι πώς, αντί να στηριζόμαστε σε ένα μοναδικό Δέντρο Απόφασης (DT) το οποίο μπορεί να είναι ασταθές, εκπαιδεύουμε ένα δάσος από δέντρα σε τυχαία τροποποιημένες εκδοχές των δεδομένων και στη συνέχεια συνδυάζουμε τις αποφάσεις μέσω της πλειοψηφικής ψήφου (majority vote)[18]. Με αυτό το τρόπο, μειώνεται η διασπορά (variance) του ταξινομητή και βελτιώνεται η ικανότητα γενίκευσης πάνω σε νέα δεδομένα.

3.4.1 Θεωρητικό υπόβαθρο και μαθηματική διατύπωση

Τυπικά, ένα μοντέλο Τυχαίου Δάσους αποτελείται από B δέντρα απόφασης $\{h_b(x)\}_{b=1}^{B}$ όπου κάθε δέντρο $h_b(x)$ εκπαιδεύεται σε διαφορετικό δείγμα (bootstrap) του συνόλου εκπαίδευσης και με τυχαίο υποσύνολο χαρακτηριστικών σε κάθε κόμβο[18]. Για ένα νέο δείγμα x , η τελική πρόβλεψη προκύπτει μέσω της πλειοψηφικής ψήφου πάνω στις προβλέψεις των επιμέρους δέντρων. Στη δυαδική ταξινόμηση με κλάσεις $c \in \{0,1\}$ η απόφαση του δάσους μπορεί να γραφεί όπως φαίνεται στη σχέση (3.1):

$$\widehat{\{y\}}(x) = \arg\max_{c \in \{0,1\}} \sum_{b=1}^{B} I(h_b(x) = c) \qquad (3.1)$$

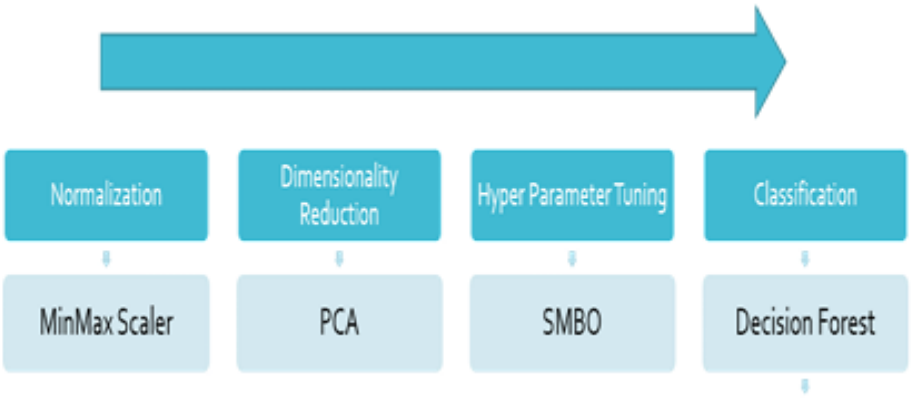
Όπου $I(\cdot)$ είναι η συνάρτηση δείκτης, η οποία παίρνει τιμή 1 όταν η συνθήκη ισχύει και 0 όταν δεν ισχύει. Με άλλα λόγια, κάθε δέντρο ψηφίζει για μια κλάση και ο αλγόριθμος Τυχαίο Δάσος (RF) επιλέγει την κλάση που συγκεντρώνει τις περισσότερες ψήφους. Αυτή, η απλή αλλά ισχυρή ιδέα βρίσκεται στον πυρήνα της λειτουργίας των Τυχαίων Δασών[18].

3.4.2 Αλγοριθμική λειτουργία και ενδεικτικό μοντέλο ανίχνευσης των εισβολών

Σε ότι έχει να κάνει με το αλγοριθμικό επίπεδο τα Τυχαία Δάση (RF) λειτουργούν με τον εξής τρόπο. Αρχικά, για κάθε δέντρο δημιουργείται ένα νέο σύνολο εκπαίδευσης με δειγματοληψία και με επανατοποθέτηση (bootstrap sample) από το αρχικό σύνολο δεδομένων (dataset). Στη συνέχεια, στην ανάπτυξη του δέντρου, σε κάθε κόμβο δεν εξετάζονται όλα τα διαθέσιμα χαρακτηριστικά αλλά επιλέγεται ένα μικρό υποσύνολο χαρακτηριστικών (k) από το σύνολο όλων των διαθέσιμων χαρακτηριστικών (m), οπότε και ισχύει (k < m)[18]. Έτσι, ο βέλτιστος διαχωρισμός υπολογίζεται μόνο πάνω σε αυτό το υποσύνολο χαρακτηριστικών (k) με βάση κάποιο κριτήριο καθαρότητας (entropy ή Gini) και ο κόμβος σπάει σε θυγατρικούς κόμβους. Η διαδικασία συνεχίζεται μέχρι να ικανοποιηθεί κάποιο κριτήριο τερματισμού, όπως το μέγιστο βάθος, ο ελάχιστος αριθμός δειγμάτων ανά φύλλο κ.α. Το δάσος προκύπτει επαναλαμβάνοντας τη διαδικασία για πολλά δέντρα (n) και η τελική απόφαση για κάθε δείγμα προκύπτει από την κλάση με τις περισσότερες ψήφους στα επιμέρους δέντρα[18].

Γενικότερα, προτείνεται ένα μοντέλο ανίχνευσης των εισβολών στην κυβερνοασφάλεια που βασίζεται στον αλγόριθμο Τυχαία Δάση (RF). Ο προεπεξεργαστικός σωλήνας (pre-processing pipeline) φαίνεται

γραφικά στο Σχήμα 3.5 ως μια ακολουθία τεσσάρων βασικών σταδίων, της Κανονικοποίησης (Normalization), της Μείωσης της Διαστασιμότητας (Dimensionality Reduction), της Βελτιστοποίησης των Υπερπαραμέτρων (Hyperparameter Tuning) και της Ταξινόμησης (Classification). Στο πρώτο στάδιο, δηλαδή στη Κανονικοποίηση εφαρμόζεται ο μετασχηματισμός ελαχίστου-μεγίστου (Min-Max Scaler), ώστε όλα τα αριθμητικά χαρακτηριστικά να κλιμακωθούν σε ένα κοινό εύρος τιμών. Με αυτό το τρόπο, αποφεύγονται τα φαινόμενα κυριαρχίας από πεδία με πολύ μεγάλες ή πολύ μικρές κλίμακες[18]. Στο δεύτερο στάδιο, δηλαδή στη Μείωση της Διαστασιμότητας χρησιμοποιείται η Ανάλυση των Κύριων Συνιστωσών (Principal Component Analysis – PCA) με στόχο τη μείωση της διαστασιμότητας καθώς, οι αρχικές μεταβλητές προβάλλονται σε ένα μικρότερο αριθμό των κύριων συνιστωσών που διατηρούν το μεγαλύτερο μέρος της πληροφορίας[18]. Ακολουθεί το στάδιο της Βελτιστοποίησης των Υπερπαραμέτρων, όπου εφαρμόζεται η Βελτιστοποίηση η οποία είναι βασισμένη σε διαδοχικά μοντέλα (Sequential Model Based Optimization – SMBO) για τη συστηματική διερεύνηση των διαφορετικών συνδυασμών των υπερπαραμέτρων του ταξινομητή με βάση την απόδοση τους σε ένα σωστά διαμορφωμένο σύνολο αξιολόγησης[18]. Τέλος, στο στάδιο της Ταξινόμησης βρίσκεται ο αλγόριθμος του Δάσους Αποφάσεων (Decision Forest), δηλαδή ο ταξινομητής Τυχαίο Δάσος (RF) ο οποίος εκπαιδεύεται πάνω στα κανονικοποιημένα και μειωμένης διαστασιμότητας δεδομένα και παράγει την τελική ετικέτα απόφασης για κάθε ροή, δηλαδή αν είναι κανονική (normal) ή ανωμαλία (anomaly)[18]. Έτσι, το Σχήμα 3.5 δείχνει τη ροή των δεδομένων μέσα από τα βασικά βήματα της προεπεξεργασίας και της ταξινόμησης που δημιουργούν το συγκεκριμένο μοντέλο.

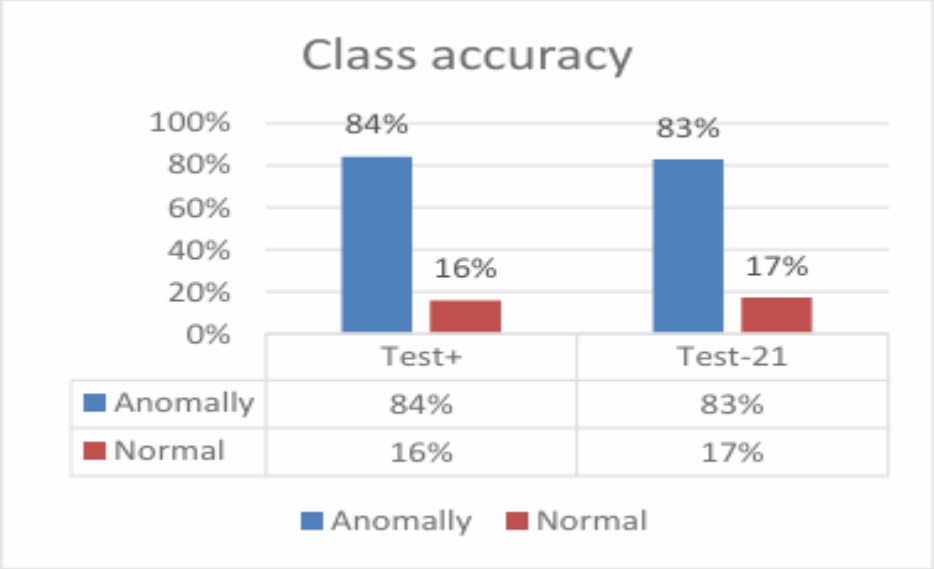


Σχήμα 3.5: Model Overview[18]

Στο συγκεκριμένο παράδειγμα του Σχήματος 3.6 για την εκπαίδευση του αλγορίθμου Τυχαίο Δάσος (RF) χρησιμοποιείται το σύνολο δεδομένων NSL-KDD (dataset NSL-KDD) το οποίο περιγράφει κάθε σύνδεση με σαράντα ένα χαρακτηριστικά και παρέχει κάποια διακριτά σύνολα (Train, Test+ και Test-21)[18]. Τα δύο τελευταία, λειτουργούν ως ανεξάρτητα σύνολα ελέγχου με το Test-21 να θεωρείται πολύ απαιτητικό λόγω της σύνθεσης κατανομής του και του υψηλότερου επιπέδου δυσκολίας των δειγμάτων του[18]. Η εφαρμογή την Ανάλυσης των Κύριων Συνιστωσών (Principal Component Analysis – PCA) πριν από την εκπαίδευση του ταξινομητή επιτρέπει στο μοντέλο να δουλεύει σε πιο συμπαγή διανύσματα χαρακτηριστικών, μειώνοντας τον θόρυβο και την υπερβολική συσχέτιση ανάμεσα στα αρχικά πεδία, ενώ ταυτόχρονα συγκρατεί το υπολογιστικό κόστος της εκπαίδευσης[18].

Τα αποτελέσματα των πειραμάτων που φαίνονται στο Σχήμα 3.6, το οποίο είναι ένα ενδεικτικό παράδειγμα μέσα από τη βιβλιογραφία που χρησιμοποιήθηκε, δείχνουν ότι το μοντέλο παρουσιάζει υψηλή απόδοση στο σύνολο της εκπαίδευσης, με την ακρίβεια διασταυρωμένης επικύρωσης (cross-validation accuracy) να είναι περίπου 99%. Στα ανεξάρτητα σύνολα ελέγχου Test+ και Test-21, η συνολική ακρίβεια μειώνεται καθώς στην πρώτη περίπτωση είναι περίπου 78% και στη δεύτερη 45%

αντίστοιχα, βέβαια η συμπεριφορά ανά κλάση είναι πολύ σημαντική. Όπως φαίνεται και στο Σχήμα 3.6, το μοντέλο ταξινομεί μόνο ένα σχετικά μικρό ποσοστό της κλάσης Κανονικό (Normal) δηλαδή 16% για το Test+ και 17% για το Test-21, ενώ αντίθετα πετυχαίνει πολύ υψηλά ποσοστά ταξινόμησης για τη κλάση Ανωμαλία (Anomaly) δηλαδή 84% για το Test+ και 83% για το Test-21. Η ανάλυση αυτή δείχνει, ότι ο ταξινομητής έχει ρυθμιστεί ώστε να ευνοεί την ανίχνευση των ανωμαλιών ακόμη και με το κόστος περισσότερων ψευδών θετικών προβλέψεων (false positives), μία επιλογή που είναι συχνά επιθυμητή σε εφαρμογές κυβερνοασφάλειας όπου είναι πολύ σημαντικό να μην χαθούν οι πραγματικές επιθέσεις[18]. Οι αντίστοιχες καμπύλες ROC(Receiver Operating Characteristic) και οι τιμές Εμβαδού κάτω από τη καμπύλη (Area Under the Curve – AUC), που κυμαίνονται περίπου στο 85% για το Test+ και 61% για το Test-21 επιβεβαιώνουν ότι το μοντέλο είναι ικανό να διακρίνει αποτελεσματικά ανάμεσα στη κανονική και στη κακόβουλη κίνηση, ιδίως στο λιγότερο δύσκολο σύνολο[18].



Σχήμα 3.6: Evaluation (Class Accuracy)[18]

3.4.3 Συνολική αξιολόγηση και ο ρόλος του αλγορίθμου στη διπλωματική

Συνολικά, επιβεβαιώνεται ότι ο αλγόριθμος Τυχαίο Δάσος (Random Forest) είναι ένας πολύ καλός ταξινομητής ιδιαίτερα στα σενάρια ανίχνευσης των εισβολών, καθώς συνδυάζει την ανθεκτικότητα των δέντρων απόφασης σε μη γραμμικά και υψηλής διαστασιμότητας δεδομένα με τα πλεονεκτήματα ενός συνόλου μοντέλων και με αυτό τον τρόπο προσφέρει υψηλή απόδοση, αντοχή στο θόρυβο και έχει τη δυνατότητα να προσαρμόζει τη συμπεριφορά του με κατάλληλες ρυθμίσεις στις υπερπαραμέτρους του[18]. Στη παρούσα διπλωματική, ο αλγόριθμος Τυχαίο Δάσος (RF) χρησιμοποιείται ως ένας από τους δύο βασικούς αλγόριθμους ταξινόμησης πάνω στη προτεινόμενη πλατφόρμα Παρακολούθησης της Ασφάλειας, ο ίδιος εκπαιδεύεται σε δεδομένα των ροών του δικτύου με κατάλληλη προεπεξεργασία και επιλογή χαρακτηριστικών και οι προβλέψεις του ενσωματώνονται σε διαδραστικούς πίνακες ελέγχου (dashboards), σε δείκτες και σε μηχανισμούς ειδοποίησης και έχουν ως στόχο την έγκαιρη και αξιόπιστη επισήμανση της ύποπτης κίνησης σε πραγματικό ή σχεδόν πραγματικό χρόνο.

3.5 Αλγόριθμος Extreme Gradient Boosting (XGBoost)

3.5.1 Θεωρητική περιγραφή και ο μηχανισμός ενίσχυσης (boosting)

Αρχικά, ο αλγόριθμος Extreme Gradient Boosting (XGBoost) ανήκει στην οικογένεια των μεθόδων ενίσχυσης (boosting) με δέντρα απόφασης και υλοποιεί στην πράξη τον αλγόριθμο των Δέντρων απόφασης με ενίσχυση κλίσης (Gradient Boosting Decision Trees – GBDT). Στον πυρήνα της, η ενίσχυση (boosting) ξεκινάει από ασθενείς ταξινομητές (weak learners), δηλαδή συνήθως ρηχά δέντρα απόφασης και τους συνδυάζει διαδοχικά με σκοπό να προκύψει ένας ισχυρός ταξινομητής[19]. Κάθε νέο δέντρο, εκπαιδεύεται πάνω σε μία τροποποιημένη εκδοχή του αρχικού συνόλου των δεδομένων και εστιάζει στα παραδείγματα που μέχρι εκείνη την στιγμή ταξινομήθηκαν δύσκολα, δηλαδή είχαν υψηλό σφάλμα[19].

Στο πλαίσιο της ενίσχυσης κλίσης (gradient boosting) η διαδικασία αυτή αναφέρεται ως πρόβλημα της αριθμητικής βελτιστοποίησης, δηλαδή σε κάθε επανάληψη προσθέτουμε ένα νέο δέντρο προς την κατεύθυνση κλίσης (gradient) της συνάρτησης κόστους, ώστε να μειωθεί σταδιακά το συνολικό σφάλμα του μοντέλου. Ο αλγόριθμος που χρησιμοποιείται (XGBoost) είναι μια βελτιστοποιημένη, παράλληλη και επεκτάσιμη υλοποίηση αυτής της ιδέας και έχει σχεδιαστεί για μεγάλους όγκους δεδομένων και υψηλές διαστασιμότητες με αποδεδειγμένη υψηλή ακρίβεια και σε εφαρμογές κυβερνοασφάλειας[19].

Γενικά, σε τυπική μορφή ο αλγόριθμος XGBoost μοντελοποιεί την πρόβλεψη ως άθροισμα από Κ δέντρα απόφασης όπως αναφέρετε στη σχέση (3.2):

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i)\,,\ f_k \in F \qquad (3.2)$$

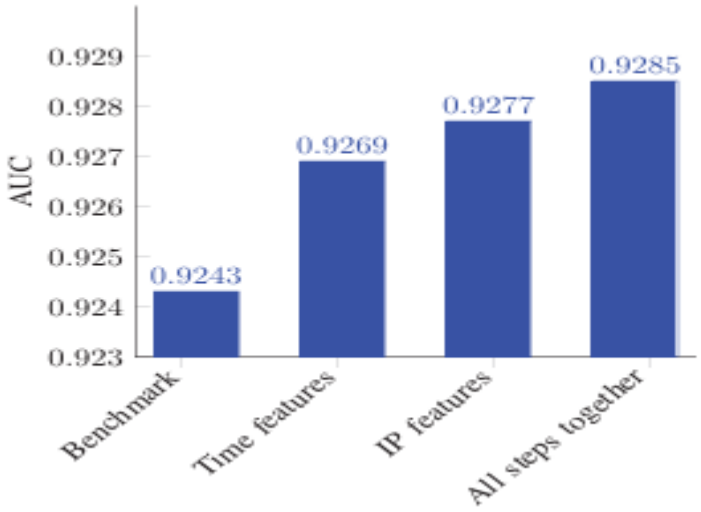
Όπου κάθε fk είναι ένα δέντρο απόφασης. Η εκπαίδευση γίνεται μέσω της ελαχιστοποίησης μιας κανονικοποιημένης συνάρτησης κόστους, η οποία συνδυάζει την εμπειρική απώλεια, για παράδειγμα την λογιστική συνάρτηση απώλειας (logistic loss) για δυαδική ταξινόμηση με έναν όρο κανονικοποίησης (regularization) ο οποίος τιμωρεί την πολυπλοκότητα των δέντρων δηλαδή τον αριθμό των φύλλων, το μέγεθος των βαρών, κ.α. και έτσι περιορίζει την υπερπροσαρμογή[19]. Πιο συγκεκριμένα, η συνάρτηση κόστους προσεγγίζεται με την ανάπτυξη κατά Taylor δεύτερης τάξης (second-order Taylor expansion), με αυτό το τρόπο ο υπολογισμός του κέρδους (gain) κάθε υποψηφίου διαχωρισμού γίνεται πιο αποτελεσματικός κατά την ανάπτυξη ενός δέντρου. Επιπλέον, χρησιμοποιείται ο μηχανισμός της ενίσχυσης με βήμα μάθησης (shrinkage / learning rate), όπου η συνεισφορά κάθε δέντρου πολλαπλασιάζεται με έναν συντελεστή n , ώστε το μοντέλο να μαθαίνει πιο σταδιακά και να μειώνεται ο κίνδυνος της υπερπροσαρμογής[19].

3.5.2 Υλοποίηση και τα πρακτικά χαρακτηριστικά του αλγορίθμου

Στο επίπεδο της υλοποίησης, ο αλγόριθμος XGBoost έχει σχεδιαστεί με έντονη έμφαση στην αποδοτικότητα (efficiency) και την επεκτασιμότητα (scalability). Ο ίδιος υποστηρίζει αραιές αναπαραστάσεις των δεδομένων (sparse data representations), ενσωματώνει έναν ενήμερο για την αραιότητα αλγόριθμο εύρεσης των διαχωρισμών (sparsity-aware split finding) που χειρίζεται με φυσικό τρόπο τις ελλιπείς τιμές ενώ την ίδια στιγμή προσφέρει υποδειγματοληψία των δειγμάτων και των χαρακτηριστικών (subsampling σε γραμμές και στήλες) ως έναν επιπλέον μηχανισμό κανονικοποίησης (regularization)[19]. Επιπλέον, επιτρέπει την παράλληλη και κατανεμημένη εκπαίδευση ακόμη και σε υπολογιστικές μονάδες γραφικών (Graphic Processing Units – GPU) γεγονός που τον κάνει μια πολύ καλή επιλογή στις εφαρμογές μεγάλης κλίμακας (large-scale learning)[19]. Τέλος, ο συνδυασμός αυτών ιδιοτήτων έχει κάνει τον αλγόριθμο XGBoost μια δεδομένη επιλογή σε ένα κομμάτι εφαρμογών οι οποίες σχετίζονται με την ανάλυση του μεγάλου όγκου δεδομένων.

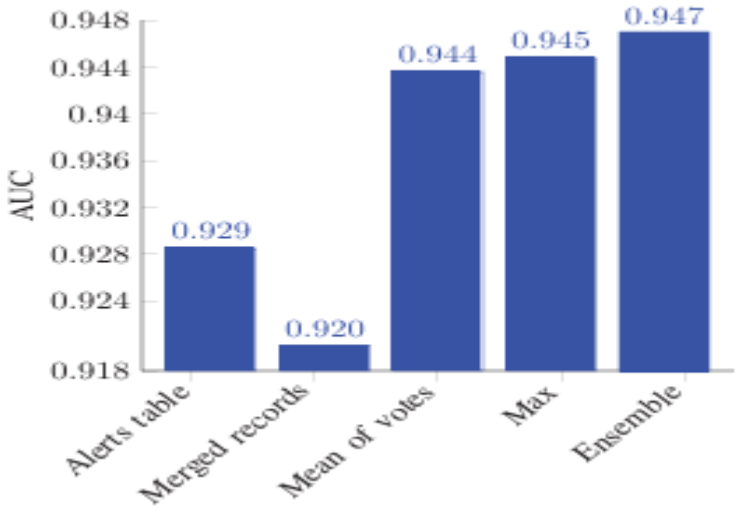
Γενικά, ο αλγόριθμος XGBoost χρησιμοποιείται ως ένας βασικός ταξινομητής (classifier) για την ανίχνευση των ύποπτων γεγονότων στη δικτυακή κίνηση μεγάλης κλίμακας. Δίνεται μεγάλη βαρύτητα

στη μηχανική των χαρακτηριστικών (feature engineering), καθώς το μεγαλύτερο μέρος των γνωρισμάτων είναι κατηγορικά (categorical features), το οποίο ταιριάζει πολύ καλά με τις μεθόδους που είναι βασισμένες σε δέντρα (tree-based methods)[19]. Για κάθε κατηγορικό γνώρισμα εφαρμόζεται κωδικοποίηση one-hot (one-hot encoding), ώστε με αυτό το τρόπο τα δεδομένα να μετατραπούν σε αριθμητική μορφή και να μπορούν να αξιοποιηθούν από το μοντέλο. Η βελτιστοποίηση του συνόλου των χαρακτηριστικών γίνεται επαναληπτικά με χρήση της πενταπλής διασταυρούμενης επικύρωσης (5-fold cross-validation), όπου σε κάθε επανάληψη αξιολογείται η μεταβολή της επιφάνειας κάτω από την καμπύλη ROC (Area Under the Roc Curve – AUC)[19]. Ξεκινώντας από ένα αρχικό μοντέλο αναφοράς (benchmark model), το οποίο πετυχαίνει AUC ≈ 0,9243, προστίθενται πρώτα τα χρονικά χαρακτηριστικά (time features) αυξάνοντας την AUC ≈ 0,9269. Στη συνέχεια, τα χαρακτηριστικά που κωδικοποιούν την πληροφορία IP (IP features) με AUC ≈ 0,9277 και τέλος, ο πλήρης συνδυασμός όλων των βημάτων (feature engineering), ο οποίος φτάνει την AUC ≈ 0,9285. Η προοδευτική αυτή βελτίωση φαίνεται καθαρά στο Σχήμα 3.7, όπου αναγνωρίζεται και η επίδραση της κάθε ομάδας χαρακτηριστικών στην τελική απόδοση του μοντέλου.



Σχήμα 3.7: Improvement of the AUC results by feature engineering steps[19]

Ένα ακόμη σημαντικό στοιχείο είναι η χρήση των στρατηγικών συνόλων (ensemble strategies) πάνω από τον ίδιο τον αλγόριθμο XGBoost με στόχο την καλύτερη ισορροπία μεταξύ της υποπροσαρμογής (underfitting) και της υπερπροσαρμογής (overfitting) σε διαφορετικές εκδοχές του συνόλου δεδομένων. Αρχικά, εξετάζεται ένα απλό σενάριο όπου ο ταξινομητής εκπαιδεύεται μόνο στα συνοπτικά χαρακτηριστικά του επιπέδου ειδοποίησης (alerts table) πετυχαίνοντας AUC ≈ 0,929. Στη συνέχεια, χρησιμοποιούνται οι συγχωνευμένες εγγραφές (merged records) από διαφορετικούς πίνακες, το οποίο χωρίς το σωστό χειρισμό οδηγεί σε πτώση της AUC ≈ 0,920. Για να χρησιμοποιηθεί σωστά η πληροφορία, δοκιμάζονται οι συναρτήσεις συνάθροισης (aggregation functions) πάνω σε πολλαπλές προβλέψεις και ο μέσος όρος των ψήφων δίνει AUC ≈ 0,944, ενώ η χρήση του μέγιστου (max) ανεβάζει την AUC ≈ 0,945. Τέλος, με την κατασκευή ενός συνόλου ταξινομητών (ensemble) από είκοσι διαφορετικά μοντέλα XGBoost, τα οποία διαφέρουν ως προς τον τρόπο επιλογής, αφαίρεσης εγγραφών και ως προς τον αρχικό σπόρο τυχαιοποίησης (random seed), η AUC ≈ 0,947 πετυχαίνοντας και την καλύτερη συνολική επίδοση[19]. Η σύγκριση αυτών των διαμορφώσεων παρουσιάζεται στο Σχήμα 3.8.



Σχήμα 3.8: Experimental results - different configurations[19]

3.5.3 Συνοπτική αξιολόγηση και ο ρόλος του αλγορίθμου στη διπλωματική

Συνοψίζοντας ο αλγόριθμος XGBoost προσφέρει μια ισχυρή και ευέλικτη προσέγγιση για την ανίχνευση των επιθέσεων στις ροές του δικτύου. Μπορεί να μοντελοποιήσει πολύπλοκες, μη γραμμικές αλληλεπιδράσεις μεταξύ των χαρακτηριστικών, να αντιμετωπίσει αποτελεσματικά την ανισορροπία των κλάσεων (class imblance) μέσα από την κατάλληλη στάθμιση των δειγμάτων και να παράγει τέτοιες βαθμολογίες ικανότητας (class probabilities) που μπορούν να ενσωματώνονται εύκολα σε συστήματα ειδοποιήσεων (alerting) και σε πίνακες ελέγχου (dashboards). Τα παραπάνω αποτελέσματα δείχνουν ότι όταν ο αλγόριθμος αυτός συνδυάζεται με προσεγμένο σχεδιασμό και μετασχηματισμό χαρακτηριστικών και όπου χρειάζεται με τη χρήση στρατηγικών συνόλων (ensembles), μπορεί να πετύχει πολύ υψηλές τιμές AUC σε ρεαλιστικά σενάρια της κυβερνοασφάλειας ακόμη και σε δεδομένα μεγάλου όγκου και υψηλής διαστασιμότητας[19].

Στο πλαίσιο της συγκεκριμένης διπλωματικής, ο αλγόριθμος XGBoost χρησιμοποιείται ως ο δεύτερος βασικός ταξινομητής σε συνδυασμό με τον αλγόριθμο Random Forest. Και οι δύο αλγόριθμοι εκπαιδεύονται πάνω σε επιλεγμένα, μη διαρρέοντα (non-leaky) χαρακτηριστικά των ροών του δικτύου μέσα από σύνολα με δεδομένα επιθέσεων, ενώ τα εκπαιδευμένα μοντέλα μετατρέπονται σε μορφή ONNX και εισάγονται στη πλατφόρμα Παρακολούθησης της Κυβερνοασφάλειας OpenSearch. Με αυτό το τρόπο, είναι δυνατή τόσο η συγκριτική αξιολόγηση των δύο αλγορίθμων όσο και η παράλληλη αξιοποίηση τους μέσα στο ίδιο περιβάλλον λειτουργίας, με στόχο την πιο αξιόπιστη ανίχνευση των επιθέσεων σε πραγματικό χρόνο.

3.6 Μετρικές αξιολόγησης (Confusion Matrix, ROC – AUC, Accuracy, Precision, Recall, F1-score)

Πρώτα από όλα, η αξιολόγηση των μοντέλων της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης (AI and ML) είναι ένα κρίσιμο βήμα σε κάθε σύστημα ανίχνευσης των επιθέσεων, καθώς καθορίζει κατά πόσο ένα προτεινόμενο σχήμα μπορεί να χρησιμοποιηθεί με ασφάλεια σε πραγματικές υποδομές. Γενικότερα, αναφέρεται ότι για την αξιολόγηση της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης στη δικτυακή ασφάλεια χρησιμοποιούνται κάποιες κατάλληλες μετρικές, οι οποίες είναι απαραίτητες τόσο για την εκτίμηση της ικανότητας ανίχνευσης (threat detection efficacy) όσο και για τη μέτρηση της επιχειρησιακής ωριμότητας μίας λύσης, δηλαδή το κατά πόσο μπορεί η ίδια να

ενσωματωθεί στις υπάρχουσες αρχιτεκτονικές ασφάλειας χωρίς να επιβαρύνει αρνητικά τους πόρους ή να παράγει υπερβολικούς ψευδείς συναγερμούς[20].

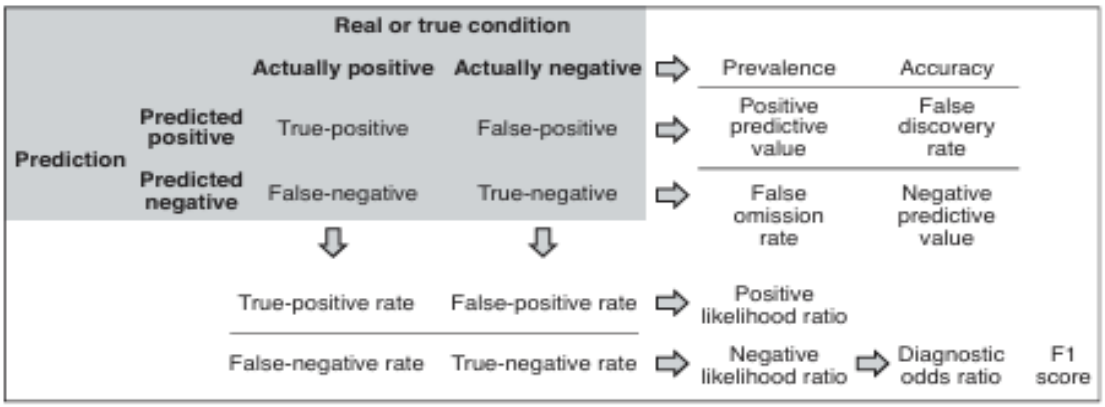
Ακόμη, είναι σημαντικό να αναφερθεί ότι η υιοθέτηση της Τεχνητής Νοημοσύνης και της Μηχανικής Μάθησης στη δικτυακή ασφάλεια χρειάζεται συστηματική χρήση των δεικτών που αποτυπώνουν την ισορροπία ανάμεσα στην ανίχνευση και στους ψευδείς συναγερμούς, την ικανότητα γενίκευσης σε νέα σενάρια καθώς και τις επιπτώσεις στη συνολική κυβερνοασφάλεια ενός οργανισμού (cybersecurity posture)[20]. Στο πλαίσιο αυτό, φαίνονται ως βασικά εργαλεία ο πίνακας σύγχυσης (confusion matrix), η καμπύλη ROC και η αντίστοιχη επιφάνεια ROC - AUC καθώς και οι παράγωγες μετρικές, όπως η ακρίβεια (accuracy), η ακρίβεια των θετικών προβλέψεων (precision), η ανάκληση (recall) και ο δείκτης F1 (F1-score)

Στις επόμενες υποενότητες αναλύεται πρώτα ο πίνακας σύγχυσης (confusion matrix) και η καμπύλη ROC – AUC, τα οποία είναι και η μαθηματική βάση πάνω στην οποία ορίζονται όλες οι υπόλοιπες μετρικές. Στην συνέχεια, γίνεται αναφορά στις υπόλοιπες μετρικές και συγκεκριμένα στην ακρίβεια (accuracy), στην ακρίβεια των θετικών προβλέψεων (precision), στην ανάκληση (recall) και στο δείκτη F1 (F1-score) και εξηγείται ο ρόλος τους στην αξιολόγηση των ταξινομητών που χρησιμοποιούνται σε αυτή την διπλωματική.

3.6.1 Πίνακας Σύγχυσης (Confusion Matrix)

Ο πίνακας σύγχυσης (confusion matrix) είναι το βασικό εργαλείο για να δούμε πως αποδίδει ένας δυαδικός ταξινομητής στη πράξη. Στο πλαίσιο ενός συστήματος Παρακολούθησης της Ασφάλειας χρησιμοποιείται για να διακρίνει αν μια κίνηση είναι κανονική η θεωρείτε επίθεση. Οι πραγματικές κλάσεις (actually positive / actually negative) τοποθετούνται συνήθως στις γραμμές και οι προβλέψεις του μοντέλου (predicted positive / predicted negative) στις στήλες με αποτέλεσμα να προκύπτουν τέσσερα είδη εκβάσεων[21]. Τα οποία είναι, σωστά εντοπισμένες επιθέσεις (true positives), κανονικές ροές που χαρακτηρίζονται λανθασμένα ως επιθέσεις (false positives), επιθέσεις που ξέφυγαν από το σύστημα (false negatives) και σωστά αναγνωρισμένες κανονικές κινήσεις (true negatives)[21]. Από αυτές τις τέσσερις ποσότητες προκύπτουν όλες οι κλασικές μετρικές, δηλαδή η ακρίβεια (accuracy), η ακρίβεια των θετικών προβλέψεων (precision), η ανάκληση (recall) και ο δείκτης F1 (F1-score).

Στο Σχήμα 3.9 ο πίνακας σύγχυσης (confusion matrix) εμφανίζεται στο αριστερό τμήμα του διαγράμματος όπου οι τέσσερις περιπτώσεις που αναφέραμε πριν (true positive, false positive, false negative, true negative) τοποθετούνται στα αντίστοιχα κελιά. Στην δεξιά πλευρά του Σχήματος 3.9 με βέλη, φαίνεται το πώς προκύπτουν οι διάφορες μετρικές, δηλαδή στη πρώτη γραμμή εμφανίζονται δείκτες όπως η επικράτηση (prevalance) και η συνολική ακρίβεια (accuracy). Στη δεύτερη γραμμή, φαίνονται τα θετικά και αρνητικά προγνωστικά ποσοστά (positive and negative predictive values – PPV / NPV), ενώ χαμηλότερα απεικονίζονται οι ρυθμοί αληθώς θετικών και ψευδών θετικών αποτελεσμάτων (true positive / false positive rates) καθώς και οι λόγοι πιθανοφάνειας (likelihood ratios), ο λόγος των διαγνωστικών odds (diagnostic odds ratio) και ο δείκτης F1 (F1-score)[21].



Σχήμα 3.9: Components of confusion matrix[21]

Στο πλαίσιο αυτής της διπλωματικής, ο πίνακας σύγχυσης (confusion matrix) και η οπτική αναπαράσταση του Σχήματος 3.9 είναι πολύ κρίσιμα εργαλεία, καθώς επιτρέπουν να φανεί καθαρά αν ένα μοντέλο κλίνει προς πολλούς ψευδείς συναγερμούς (false positivies) ή προς πολλές μη ανιχνεύσιμες επιθέσεις (false negatives) και αποτελούν τη βάση πάνω στην οποία στη συνέχεια θα οριστούν οι μαθηματικοί τύποι για την ακρίβεια (accuracy), την ακρίβεια των θετικών προβλέψεων (precision), την ανάκληση (recall) και το δείκτη F1 (F1-score), ώστε η σύγκριση μεταξύ των αλγορίθμων Random Forest και XGBoost να είναι δίκαιη και συνεπής.

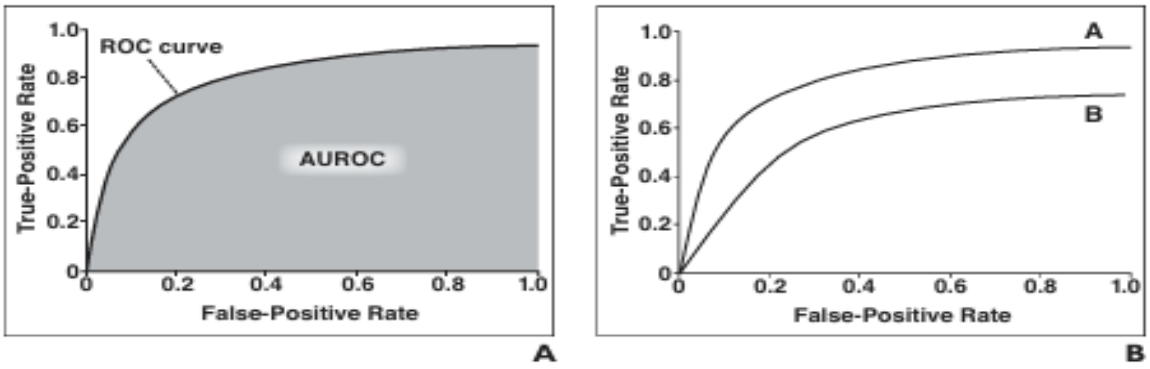
3.6.2 Καμπύλη ROC και Επιφάνεια ROC – AUC

Αρχικά, η καμπύλη ROC (Receiver Operating Characteristic) περιγράφει τη συμπεριφορά ενός ταξινομητή όταν αλλάζει το κατώφλι απόφασης (threshold) πάνω στις προβλεπόμενες πιθανότητες. Για κάθε κατώφλι απόφασης, από τον αντίστοιχο πίνακα σύγχυσης μπορούμε να υπολογίσουμε το ρυθμό των αληθώς θετικών (true positive rate – TPR ή recall) και τον ρυθμό τον ψευδών θετικών (false positive rate – FPR)[21]. Αν επιλεχθεί χαμηλό κατώφλι απόφασης, το σύστημα γίνεται πιο ευαίσθητο (υψηλό TPR) αλλά παράγει περισσότερους ψευδείς συναγερμούς (υψηλό FPR), το αντίστροφο συμβαίνει αν επιλέξουμε υψηλότερο κατώφλι απόφασης (threshold)[21]. Η καμπύλη ROC είναι το σύνολο των σημείων FPR και TPR για όλα τα δυνατά κατώφλια απόφασης και αποτυπώνει γραφικά αυτή τη σχέση συμβιβασμού (trade-off).

Στη συνέχεια, στο Σχήμα 3.10, το αριστερό μέρος (A) δείχνει μια χαρακτηριστική ROC καμπύλη με την περιοχή κάτω από αυτήν σκιασμένη και επισημασμένη ως AUROC (Area Under the ROC Curve). Η σκιασμένη αυτή επιφάνεια είναι ο συνοπτικός δείκτης απόδοσης, δηλαδή όσο μεγαλύτερη είναι η AUROC τόσο καλύτερα διαχωρίζει ο ταξινομητής τα θετικά από τα αρνητικά δείγματα[21]. Μαθηματικά η AUC ορίζεται ως το ολοκλήρωμα της TPR ως προς την FPR το οποίο παρουσιάζεται στη σχέση (3.3):

$$AUC = \int_0^1 TPR(FPR)d(FPR) \qquad (3.3)$$

Στο δεξί μέρος (B) του σχήματος εμφανίζονται δύο καμπύλες ROC (μοντέλα A και B) στο ίδιο γράφημα. Η ανώτερη καμπύλη (μοντέλο A) βρίσκεται συστηματικά πιο κοντά στην πάνω αριστερή γωνία, άρα έχει μεγαλύτερη AUC και σαφώς καλύτερη συνολική διακριτική ικανότητα σε σχέση με το άλλο μοντέλο (μοντέλο B)[21].



Σχήμα 3.10: Area under ROC curve (AUROC) [21]

Γενικότερα, για εφαρμογές που έχουν να κάνουν με την ανίχνευση των επιθέσεων, η ROC – AUC είναι ιδιαίτερα χρήσιμη διότι επιτρέπει τη δίκαιη σύγκριση διαφορετικών αλγορίθμων ανεξάρτητα από το κατώφλι απόφασης (threshold) που θα επιλεγεί επιχειρησιακά, για παράδειγμα σε περιόδους απειλής μπορεί να γίνει αποδεκτό υψηλότερο FPR για να μεγιστοποιηθεί το TPR[21]. Στη εργασία αυτή, η ROC – AUC θα χρησιμοποιηθεί ως βασικός συνοπτικός δείκτης για τη σύγκριση των μοντέλων Random Forest και XGBoost, συμπληρώνοντας τις υπόλοιπες μετρικές που προκύπτουν από το πίνακα σύγχυσης (confusion matrix).

3.6.3 Ακρίβεια (Accuracy)

Η ακρίβεια (accuracy) είναι ίσως η πιο άμεση και διαισθητική μετρική καθώς εκφράζει το ποσοστό των δειγμάτων που ταξινομήθηκαν σωστά από το μοντέλο. Με βάση τον πίνακα σύγχυσης αν ορίσουμε ότι είναι οι σωστά εντοπισμένες επιθέσεις (true positives - TP), οι σωστά αναγνωρισμένες κανονικές κινήσεις (true negatives - TN), οι κανονικές ροές που χαρακτηρίζονται λανθασμένα ως επιθέσεις (false positives - FP), οι επιθέσεις που ξέφυγαν από το σύστημα (flase negatives – FN) τότε η ακρίβεια ορίζεται όπως φαίνεται στη σχέση (3.4):[22]

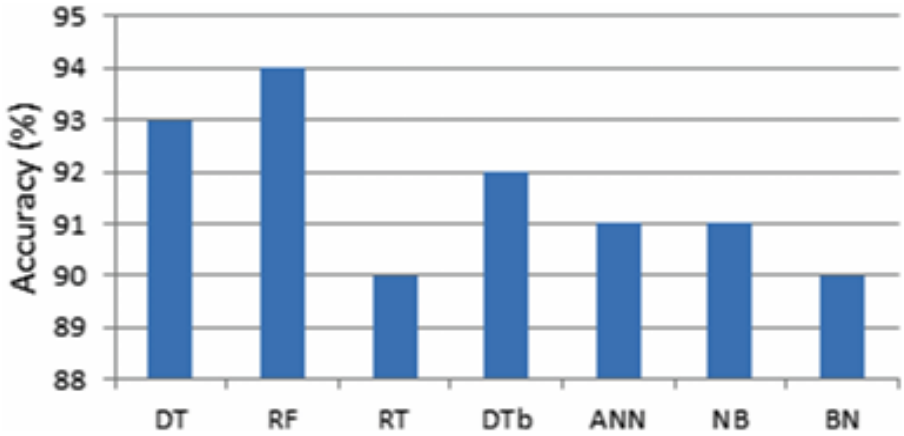
$$Accuracy = \frac{TP + TN}{TP+TN+FP+FN} \qquad (3.4)$$

Στο πλαίσιο ενός συστήματος Παρακολούθησης της Ασφάλειας (CM), η ακρίβεια απαντάει στο ερώτημα, σε ποιο ποσοστό των ροών ή των συμβάντων το σύστημα πήρε τη σωστή απόφαση, είτε ως κανονικό είτε ως ύποπτο. Για να θεωρηθεί ένα μοντέλο συνολικά αξιόπιστο χρειάζεται ένα υψηλό ποσοστό ακρίβειας, ειδικά όταν ενσωματώνεται σε μια λειτουργική πλατφόρμα Παρακολούθησης της Ασφάλειας.

Παρόλα αυτά, όταν τα δεδομένα είναι πολύ ανισόρροπα, δηλαδή έχουν πολύ περισσότερη κανονική κίνηση από επιθέσεις, η ακρίβεια μπορεί να γίνει παραπλανητική. Ένα μοντέλο που χαρακτηρίζει κανονική σχεδόν κάθε ροή μπορεί να εμφανίζει υψηλό ποσοστό ακρίβειας αλλά να αποτυγχάνει στη πράξη να εντοπίσει μεγάλο ποσοστό των επιθέσεων[22]. Για το λόγο αυτό, στη συγκεκριμένη εργασία η ακρίβεια (accuracy) χρησιμοποιείται κυρίως συμπληρωματικά πάντα σε συνδυασμό με την ακρίβεια των θετικών προβλέψεων (precision), την ανάκληση (recall) και το δείκτη F1 (F1-score), ώστε να αποδίδει πιο ρεαλιστικά τη συμπεριφορά του συστήματος ως προς τη κλάση επίθεση.

Ένα χαρακτηριστικό παράδειγμα φαίνεται στο Σχήμα 3.11, όπου συγκρίνονται διάφοροι ταξινομητές της Μηχανικής Μάθησης στο περιβάλλον της ανίχνευσης των επιθέσεων. Ο Random Forest επιτυγχάνει την υψηλότερη ακρίβεια σε σχέση με τους εναλλακτικούς αλγόριθμους, επιβεβαιώνοντας την

καταλληλότητά του για προβλήματα της κυβερνοασφάλειας, γεγονός που συνδέει άμεσα την επιλογή του ως ένα εκ των δύο βασικών μοντέλων αυτής της διπλωματικής[22].



Σχήμα 3.11: Performance comparison results with respect to accuracy for numerous machine learning based IDS models[22]

3.6.4 Ακρίβεια θετικών προβλέψεων (Precision)

Ακόμη, η ακρίβεια των θετικών προβλέψεων (Precision – positive predictive value – PPV) μετράει καθαρά πόσες είναι οι θετικές ειδοποιήσεις (alerts) του συστήματος. Ορίζεται ως το ποσοστό των προβλέψεων επίθεση που αντιστοιχούν πράγματι σε επιθέσεις, με τον παρακάτω τύπο ο οποίος παρουσιάζεται στο (3.5):[22]

$$Precision = \frac{TP}{TP+FP} \qquad (3.5)$$

Με πιο απλά λόγια, αν το σύστημα Παρακολούθησης της Ασφάλειας ενεργοποιήσει 100 συναγερμούς τότε η ακρίβεια των θετικών προβλέψεων δείχνει σε πόσες από αυτές τις περιπτώσεις υπήρξε πραγματική απειλή. Όταν η ακρίβεια των θετικών προβλέψεων είναι υψηλή σημαίνει λίγους ψευδούς θετικούς συναγερμούς (false positives) και συνεπώς μειωμένο θόρυβο για τους αναλυτές της ασφάλειας[22]. Αυτό είναι ιδιαίτερα σημαντικό στα επιχειρησιακά περιβάλλοντα, καθώς η συνεχής ροή μη χρήσιμων ειδοποιήσεων (alerts) μπορεί να οδηγήσει σε κόπωση των συναγερμών (alert fatigue) και τελικά σε μείωση της εμπιστοσύνης στο σύστημα[22].

Στο κομμάτι αυτής της διπλωματικής, η ακρίβεια των θετικών προβλέψεων (precision) αποτελεί βασική μετρική για την αξιολόγηση των ταξινομητών, ιδιαίτερα σε σενάρια με μεγάλο όγκο καθημερινών γεγονότων. Γενικότερα, ένα μοντέλο με σημαντικά υψηλότερη ακρίβεια θετικών προβλέψεων θεωρείται πρακτικά πιο αξιοποιήσιμο, ακόμη και όταν η συνολική ακρίβεια (accuracy) είναι συγκρίσιμη με αυτή των άλλων αλγόριθμων.

3.6.5 Ανάκληση (Recall)

Επιπρόσθετα, η ανάκληση (recall - true positive rate - TPR) μετράει το ποσοστό των πραγματικών επιθέσεων που καταφέρνει να εντοπίσει το μοντέλο. Με βάση τον πίνακα σύγχυσης (confusion matrix) ορίζεται ως η μαθηματική σχέση η οποία αναφέρεται στο (3.6):[22]

$$Recall = TPR = \frac{TP}{TP+FN} \qquad (3.6)$$

Με άλλα λόγια, η ανάκληση απαντάει στο ερώτημα, από όλες τις επιθέσεις που υπάρχουν στο σύστημα πόσες ανιχνεύει το σύστημα Παρακολούθησης της Ασφάλειας (CM). Όταν η ανάκληση έχει υψηλή τιμή σημαίνει ότι υπάρχουν λίγες μη ανιχνευμένες επιθέσεις (false negatives), οι οποίες από επιχειρησιακή μεριά αποτελούν συνήθως τα πιο κρίσιμα και επικίνδυνα σφάλματα. Μια επίθεση που περνά απαρατήρητη μπορεί να οδηγήσει σε διαρροή των δεδομένων, διακοπή ή υποβάθμιση κρίσιμων υπηρεσιών καθώς και σε πλευρική κίνηση του επιτιθέμενου στο εσωτερικό δίκτυο[22].

Για τον λόγο αυτό, σε πολλά συστήματα της κυβερνοασφάλειας προτιμάται συχνά ένα μοντέλο με ελαφρώς χαμηλότερη ακρίβεια των θετικών προβλέψεων (precision) και υψηλότερη ανάκληση (recall), ειδικά στις κρίσιμες ζώνες όπως οι εξυπηρετητές παραγωγής, ο δικτυακός πυρήνας, κ.α. Η τελική επιλογή του κατωφλίου (threshold) είναι πάντα θέμα συμβιβασμού μεταξύ του recall και του precision, όπως αναλύεται και μέσω της καμπύλης ROC-AUC στην προηγούμενη υποενότητα[22].

3.6.6 Δείκτης F1 (F1-Score)

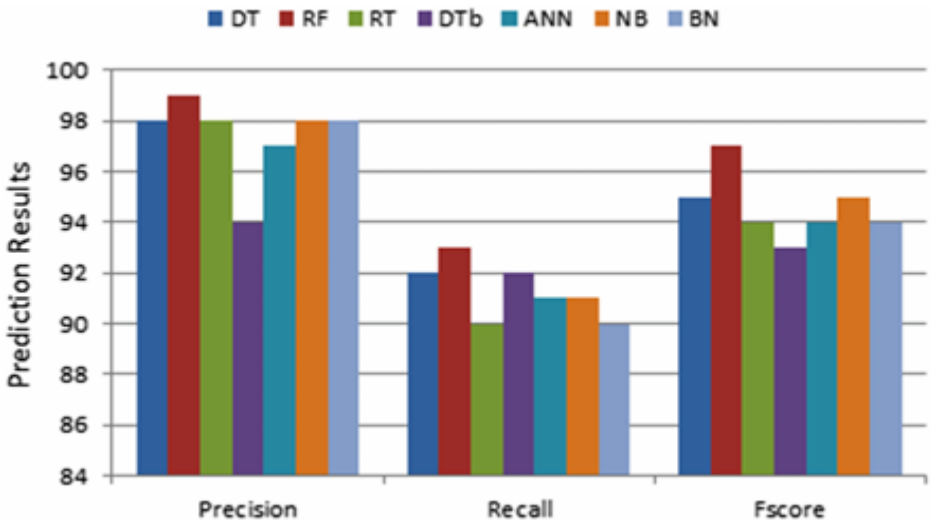
Επιπλέον, ο δείκτης F1 (F1- score) συνδυάζει την ακρίβεια των θετικών προβλέψεων (precision) και την ανάκληση (recall) σε έναν ενιαίο αριθμό, λαμβάνοντας υπόψη και τις δύο πτυχές της απόδοσης. Ορίζεται ως αρμονικός μέσος (harmonic mean) των δύο και αναλύεται στη σχέση (3.7):[22]

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad (3.7)$$

Επειδή ο δείκτης F1 χρησιμοποιεί αρμονικό και όχι αριθμητικό μέσο, τιμωρεί τις ανισορροπίες, καθώς εάν ένα μοντέλο έχει πολύ υψηλή ακρίβεια θετικών προβλέψεων (precision) αλλά χαμηλή ανάκληση (recall) ή το αντίστροφο τότε η τιμή F1 παραμένει σχετικά χαμηλή. Με αυτό τον τρόπο, ο δείκτης F1 δεν επιτρέπει να κρυφτεί ένα σοβαρό πρόβλημα, όπως για παράδειγμα πολλές μη ανιχνεύσιμες επιθέσεις ή πάρα πολλοί ψευδείς συναγερμοί, πίσω από μία μόνο εντυπωσιακή μετρική[22].

Γενικά, στα ανισόρροπα σύνολα δεδομένων (datasets) κυβερνοασφάλειας όπου τα κανονικά δείγματα είναι κατά πολύ περισσότερα από τις επιθέσεις, ο δείκτης F1 θεωρείται συχνά πιο αντιπροσωπευτικός συνολικός δείκτης σε σχέση με τη σκέτη ακρίβεια (accuracy)[22].Για αυτό το λόγο, ο δείκτης F1 αξιοποιείται σε σχετικές εργασίες στον χώρο της ανίχνευσης εισβολών για να συγκρίνει διαφορετικούς ταξινομητές πάνω στο ίδιο σύνολο δεδομένων. Στη διπλωματική αυτή, ο δείκτης F1 χρησιμοποιείται ως ένας βασικός συνολικός δείκτης απόδοσης για τη σύγκριση των Random Forest και XGBoost, καθώς αποτυπώνει κατά πόσο το μοντέλο καταφέρνει να διατηρεί ταυτόχρονα υψηλή ακρίβεια θετικών προβλέψεων (precision) και υψηλή ανάκληση (recall).

Κλείνοντας, η σημασία του δείκτη F1 σε συνδυασμό με την ακρίβεια των θετικών προβλέψεων και την ανάκληση αποτυπώνεται στο Σχήμα 3.12, όπου παρουσιάζονται σε σύγκριση οι τιμές των τριών αυτών μετρικών για διαφορετικούς ταξινομητές σε ένα κλασικό σενάριο ανίχνευσης των επιθέσεων. Σε αυτό το Σχήμα παρατηρείται ότι ο Random Forest εμφανίζει τις υψηλότερες τιμές και στις τρεις μετρικές, κάτι το οποίο τονίζει την ισορροπημένη του συμπεριφορά και εξηγεί το λόγο για τον οποίο χρησιμοποιείται τόσο συχνά σε εφαρμογές της ασφάλειας[22].



Σχήμα 3.12: Performance comparison results with respect to precision, recall, f1-score for numerous machine learning classification based IDS models[22]

3.6.7 Σχόλια για τη χρήση των Μετρικών Αξιολόγησης στην ανίχνευση επιθέσεων

Όλες οι παραπάνω μετρικές, δεν είναι ανταγωνιστικές η μία απέναντι στην άλλη, αλλά συμπληρωματικές. Κάθε μία αναδεικνύει μια διαφορετική πλευρά της συμπεριφοράς του μοντέλου και η σωστή τους ερμηνεία είναι πολύ σημαντική όταν τα αποτελέσματα της ταξινόμησης επηρεάζουν τις αποφάσεις της ασφάλειας. Σε ένα σύστημα Παρακολούθησης της Ασφάλειας (CM), το ζητούμενο δεν είναι απλώς ένα υψηλό ποσοστό σε μία μόνο μετρική, αλλά μια ισορροπία ανάμεσα στην αποτελεσματική ανίχνευση, δηλαδή την υψηλή ανάκληση (recall) και στην επιχειρησιακή διαχειρισιμότητα, δηλαδή την υψηλή ακρίβεια θετικών προβλέψεων (precision) ώστε οι αναλυτές να μην χάνονται μέσα στους ψευδείς συναγερμούς.

Γενικότερα, επισημαίνεται ότι η αποκλειστική χρήση της συνολικής ακρίβειας (accuracy) μπορεί να οδηγήσει σε πολύ αισιόδοξες εκτιμήσεις, ιδιαίτερα σε μη ισορροπημένα σύνολα δεδομένων (imbalanced datasets) όπου η πλειονότητα των δειγμάτων ανήκει στην κανονική κλάση[22]. Για τον λόγο αυτό, προτείνονται συχνά πιο σύνθετοι δείκτες, όπως ο δείκτης F1 (F1-score) και το εμβαδόν κάτω από την καμπύλη ROC (ROC-AUC), οι οποίοι λαμβάνουν υπόψη τόσο τα ψευδώς θετικά όσο και τα ψευδώς αρνητικά (false positives and false negatives) αποτελέσματα και επιτρέπουν μια πιο σταθερή σύγκριση μεταξύ των διαφορετικών μοντέλων και των διαφορετικών κατωφλίων απόφασης (thresholds)[22].

Στο πλαίσιο της παρούσας διπλωματικής, οι παραπάνω μετρικές χρησιμοποιούνται για την αξιολόγηση των δύο βασικών ταξινομητών που υλοποιούνται πρακτικά, δηλαδή των Random Forest και XGBoost. Ο πίνακας σύγχυσης (confusion matrix) και οι μετρικές ακρίβεια (accuracy), δηλαδή η ακρίβεια των θετικών προβλέψεων (precision), η ανάκληση (recall) και ο δείκτης F1 (F1-score) αξιοποιούνται για την αναλυτική αποτύπωση της συμπεριφοράς του κάθε μοντέλου ανά κλάση (κανονική κίνηση / επίθεση), ενώ το εμβαδόν κάτω από την καμπύλη ROC (ROC-AUC) λειτουργεί ως ένας συνοπτικός δείκτης της διαχωριστικής ικανότητας. Με αυτό τον τρόπο, τα συμπεράσματα σχετικά με την καταλληλότητα του κάθε μοντέλου στο προτεινόμενα περιβάλλον Παρακολούθησης της Ασφάλειας (CM) στηρίζονται σε μια στιβαρή και με πολλές διαστάσεις αξιολόγηση και όχι σε έναν μεμονωμένο αριθμητικό δείκτη.

3.7 Επίλογος Κεφαλαίου

Εν κατακλείδι, στο τρίτο κεφάλαιο αναλύθηκε το θεωρητικό υπόβαθρο των επιβλεπόμενων αλγορίθμων ταξινόμησης (supervised classification algorithms) που αξιοποιούνται στη διπλωματική αυτή για την ανίχνευση των επιθέσεων στις ροές του δικτύου και για την υλοποίηση ενός συστήματος Παρακολούθησης της Ασφάλειας (CM). Αρχικά, εξετάστηκαν δύο κλασικοί αλγόριθμοι , η Λογιστική Παλινδρόμηση (LR) και οι Μηχανές των Διανυσμάτων Στήριξης (SVM), οι οποίοι λειτουργούν ως αντιπροσωπευτικά παραδείγματα των γραμμικών και μη γραμμικών ταξινομητών. Μέσα από την ανάλυση των ιδιοτήτων, των πλεονεκτημάτων και των περιορισμών τους αναδείχθηκε ο ρόλος τους κυρίως ως σημείο αναφοράς (baseline classifiers) για την αξιολόγηση των πιο σύνθετων μεθόδων.

Στη συνέχεια, το κεφάλαιο εστίασε στου αλγόριθμους η οποίοι είναι βασισμένοι στα δέντρα απόφασης (decision trees), οι οποίοι αποτελούν τον πυρήνα των μοντέλων που υλοποιούνται πρακτικά στη διπλωματική. Αρχικά, παρουσιάστηκαν τα απλά Δέντρα Απόφασης, τα οποία προσφέρουν υψηλή ερμηνευσιμότητα και δυνατότητα χειρισμού των ετερογενών χαρακτηριστικών αλλά από την άλλη είναι ευάλωτα στην υπερπροσαρμογή (overfitting). Πάνω σε αυτή τη βάση, αναλύθηκαν οι δύο ισχυρές μέθοδοι συνόλων, ο Random Forest ο οποίος αξιοποιεί την τυχαία δειγματοληψία των δειγμάτων και των χαρακτηριστικών (bagging and feature subsampling) για να μειώσει τη διασπορά και για να βελτιώσει τη γενίκευση και ο XGBoost, ο οποίος υλοποιεί έναν βελτιστοποιημένο αλγόριθμο ενίσχυσης την κλίσης (gradient boosting) με μεγάλη έμφαση στην αποδοτικότητα (efficiency), τη κανονικοποίηση (regularization) και την κλιμακωσιμότητα (scalability). Η βιβλιογραφία που εξετάστηκε ανέδειξε τον Random Forest και τον XGBoost ως δύο από τους πιο αποτελεσματικούς αλγόριθμους για τα προβλήματα μεγάλης κλίμακας της κυβερνοασφάλειας. Έτσι, δικαιολογείται και η επιλογή τους ως βασική ταξινομητές συγκεκριμένη εργασία.

Τέλος, το κεφάλαιο ολοκληρώθηκε με μία αναλυτική παρουσίαση των μετρικών αξιολόγησης που θα χρησιμοποιηθούν στα πειραματικά αποτελέσματα δηλαδή, ο πίνακας σύγχυσης (confusion matrix) και οι παράγωγες μετρικές όπως, η ακρίβεια (accuracy), η ακρίβεια των θετικών προβλέψεων (precision), η ανάκληση (recall), και το εμβαδόν κάτω από την καμπύλη ROC (ROC- AUC). Ακόμη, τονίστηκε ότι ειδικά στα μη ισορροπημένα σύνολα δεδομένων της κυβερνοασφάλειας, η αποκλειστική εστίαση στην ακρίβεια είναι ανεπαρκής και ότι χρειάζεται μια πολυδιάστατη αξιολόγηση που να λαμβάνει υπόψη τον συμβιβασμό (trade-off) μεταξύ των ψευδών θετικών (false positives) και των ψευδών αρνητικών (false negatives) αποτελεσμάτων. Στο κομμάτι αυτό, οι μετρικές που αναφέρθηκαν πριν θα χρησιμοποιηθούν στο πειραματικό μέρος για τη δίκαιη και συνεπή σύγκριση των μοντέλων Random Forest και XGBoost πάνω σε πραγματικά δεδομένα της δικτυακής κίνησης.

Στο επόμενο κεφάλαιο, αξιοποιείται το θεωρητικό πλαίσιο που διαμορφώθηκε εδώ, παρουσιάζονται το σύνολο δεδομένων (dataset) των ροών του δικτύου που χρησιμοποιείται , τα σενάρια της επίθεσης καθώς και τα βήματα της προεπεξεργασίας (pre-processing) και της επιλογής των χαρακτηριστικών (feature selection) που απαιτούνται, ώστε οι αλγόριθμοι Random Forest και XGBoost να εφαρμοστούν με σωστό και ορθά συγκρίσιμο τρόπο στο περιβάλλον της Παρακολούθησης της Ασφάλειας που χρησιμοποιήθηκε (OpenSearch dashboards), το οποίο χρησιμοποιείται για την οπτικοποίηση, τη συσχέτιση των συμβάντων και τη λειτουργική παρακολούθηση του monitoring και του alerting.

Κεφάλαιο 4ο: Προπεξεργασία των δεδομένων (Data Preprocessing) και το Σύνολο δεδομένων CIC-DDoS2019 (CIC-DDoS2019 Dataset)

4.1 Εισαγωγή κεφαλαίου

Στο προηγούμενο κεφάλαιο παρουσιάστηκαν οι βασικοί επιβλεπόμενοι αλγόριθμοι ταξινόμησης, δηλαδή οι Logistic Regression, Support Vector Machines, Decision Trees, Random Forest και XGBoost και οι μετρικές αξιολόγησης που θα χρησιμοποιηθούν αργότερα για την αξιολόγηση της απόδοσης των δύο μοντέλων που χρησιμοποιήθηκαν στο πρακτικό κομμάτι, δηλαδή του Random Forest και του XGBoost. Βέβαια, η εφαρμογή ακόμη και των πιο ισχυρών αλγόριθμων σε δεδομένα δικτυακής κίνησης δεν μπορεί να γίνει απευθείας πάνω σε μη επεξεργασμένα δεδομένα. Τα πραγματικά ίχνη της δικτυακής κίνησης (traces) περιέχουν θόρυβο, ελλιπείς τιμές, πλεονάζουσα πληροφορία, ανισορροπία κλάσεων και σε κάποιες περιπτώσεις γνωρίσματα τα οποία μπορεί να οδηγήσουν σε φαινόμενα διαρροής της πληροφορίας (data leakage). Για το λόγο αυτό, η προεπεξεργασία (data preprocessing) και η προσεκτική κατανόηση του συνόλου δεδομένων (dataset) αποτελούν τα σημαντικότερα βήματα πριν από οποιαδήποτε διαδικασία εκπαίδευσης των μοντέλων της Μηχανικής Μάθησης. Στην διπλωματική αυτή, το σύνολο δεδομένων και η προεπεξεργασία οργανώνονται ειδικά για την ανίχνευση DoS / DDoS, όπως αυτά αποτυπώνονται στο CIC-DDoS2019. Σημειώνεται ότι, για το πρακτικό σκέλος της διπλωματικής, αξιοποιήθηκαν τα δεδομένα αποκλειστικά από μια ημέρα καταγραφής (11-03-2019). Όποτε, η εκπαίδευση και η αξιολόγηση που ακολουθούν πραγματοποιούνται εντός της ίδια ημέρας.

Στο κομμάτι της συγκεκριμένης διπλωματικής χρησιμοποιήθηκε το δημόσιο σύνολο δεδομένων CIC-DDoS2019, το οποίο αξιοποιήθηκε αποκλειστικά για σενάρια DoS / DDoS και οι ετικέτες, κατηγορίες που αναλύονται στο κεφάλαιο αυτό αφορούν τους αντίστοιχους τύπους των DDoS επιθέσεων. Ενώ, στο πρακτικό κομμάτι χρησιμοποιήθηκε αποκλειστικά η ημέρα 11/03/2019. Ακόμη, το συγκεκριμένο σύνολο δεδομένων έχει σχεδιαστεί για να προσομοιώνει τη ρεαλιστική δικτυακή κίνηση με συνδυασμό μερικών κανονικών ροών αλλά και ποικιλία σεναρίων DoS / DDoS επιθέσεων. Το συγκεκριμένο σύνολο δεδομένων (dataset), οργανώνεται σε επίπεδα ροών (flows) και περιλαμβάνει τόσο χαρακτηριστικά χαμηλού επιπέδου, για παράδειγμα αριθμούς πακέτων, bytes και χρονισμούς όσο και πιο σύνθετα στατιστικά γνωρίσματα, μαζί με ετικέτες (labels) που δηλώνουν αν μια ροή είναι κανονική ή αν αντιστοιχεί σε ένα συγκεκριμένο τύπο επίθεσης. Ωστόσο, παρά τα πλεονεκτήματα του το CIC-DDoS2019 δεν είναι έτοιμο για χρήση κατευθείαν καθώς, παρουσιάζει αρκετά σημαντικά ζητήματα όπως η ανισορροπία μεταξύ της κανονικής και της κακόβουλης κίνησης, οι διπλότυπες ή σχεδόν ταυτόσημες καταγραφές (redundancy) καθώς και τα χαρακτηριστικά που μπορούν να οδηγήσουν σε υπερεκτίμηση της απόδοσης των μοντέλων αν δεν αντιμετωπιστούν με τον κατάλληλο τρόπο.

Γενικότερα, στόχος του συγκεκριμένου κεφαλαίου είναι να περιγράψει αναλυτικά το σύνολο δεδομένων CIC-DDoS2019, το οποίο αφορά την ημέρα 11/03/2019 όπως χρησιμοποιείται στη συγκεκριμένη εργασία αλλά και τα βήματα της προεπεξεργασίας που εφαρμόζονται ώστε να δημιουργηθούν δύο διακριτά σύνολα χαρακτηριστικών. Ένα σύνολο χωρίς διαρροή πληροφορίας (non-leaky), το οποίο επιτρέπει τη δίκαιη και ρεαλιστική αξιολόγηση των μοντέλων και ένα σύνολο με διαρροή πληροφορίας (leaky), το οποίο χρησιμοποιείται αποκλειστικά για τον ελεγχόμενο πειραματισμό ή για τη σύγκριση, ώστε να αναδειχθεί η μεροληψία (bias) και η υπερεκτίμηση των μετρικών που προκαλεί η διαρροή πληροφορίας και όχι ως ρεαλιστική διαμόρφωση της παραγωγής. Πιο συγκεκριμένα, στην επόμενη υποενότητα παρουσιάζεται το ίδιο το σύνολο δεδομένων, η δομή των

ροών του, οι κατηγορίες των επιθέσεων και τα βασικά προβλήματα που το χαρακτηρίζουν, δηλαδή η ανισορροπία, οι διπλότυπες ή σχεδόν ταυτόσημες καταγραφές και η διαρροή των δεδομένων. Στη συνέχεια, περιγράφεται βήμα προς βήμα η προεπεξεργασία δηλαδή, ο καθαρισμός των δεδομένων, η κανονικοποίηση, η επιλογή των χαρακτηριστικών και ο διαχωρισμός μεταξύ των γνωρισμάτων με διαρροή ή χωρίς. Τέλος, γίνεται ένας σύντομος επίλογος και συνδέονται τα αποτελέσματα της προεπεξεργασίας με τα επόμενα κεφάλαια όπου υλοποιούνται και αξιολογούνται τα μοντέλα Random Forest και XGBoost πάνω στα τελικά επεξεργασμένα σύνολα δεδομένων.

4.2 Το σύνολο δεδομένων CIC-DDoS2019 (DATASET CIC-DDoS2019)

4.2.1 Δομή και αναπαράσταση των ροών (flow-based representation)

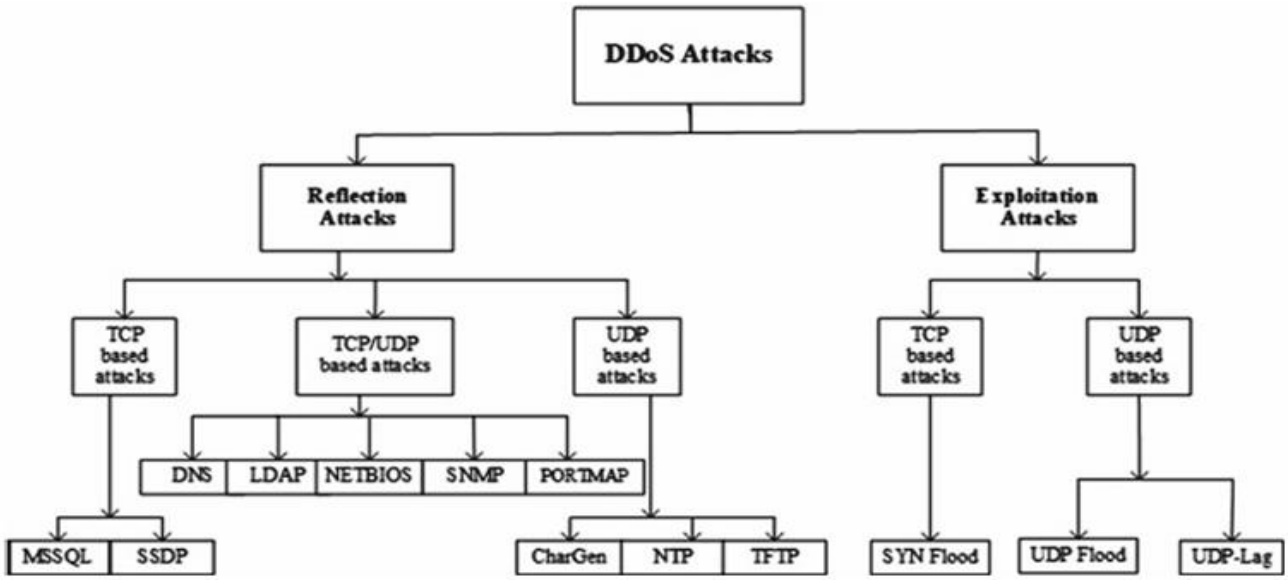
Αρχικά, το CIC-DDoS2019 είναι ένα σύνολο δεδομένων σχεδιασμένο για την υποστήριξη της ανάλυσης και της ανίχνευσης των επιθέσεων της άρνησης εξυπηρέτησης (DoS) και της κατανεμημένης άρνησης εξυπηρέτησης (DDoS) στο επίπεδο ροής (flow-based), παρέχοντας καταγραφές οργανωμένες ανά ημέρα πειραματισμού. Για κάθε ημέρα διατίθενται τόσο κάποια ακατέργαστα δεδομένα της δικτυακής κίνησης σε μορφή αρχείων καταγραφής πακέτων (PCAP), όσο και μερικές συμπληρωματικές καταγραφές γεγονότων (event logs) ανά μηχάνημα, ώστε να είναι δυνατή η αναπαραγωγή και η επαλήθευση της διαδικασίας εξαγωγής των χαρακτηριστικών και της αντιστοίχισης των ετικετών (labels) στα δεδομένα[23].

Ακόμη, η εξαγωγή των χαρακτηριστικών από τα ακατέργαστα δεδομένα στο σύνολο δεδομένων πραγματοποιείται με τη χρήση του εργαλείου CICFlowMeter-V3, το οποίο παράγει ένα πλούσιο πίνακα χαρακτηριστικών ροής (flow features), με αποτέλεσμα τη δημιουργία αρχείων CSV ανά μηχάνημα, τα οποία είναι κατάλληλα για εφαρμογές Τεχνητής Νοημοσύνης (AI) και Μηχανικής Μάθησης (ML)[23]. Έτσι, η αναπαράσταση αυτή επιτρέπει την εκπαίδευση και την αξιολόγηση των ταξινομητών (classifiers) πάνω σε συνοπτικά συστατικά της ροής, περιορίζοντας με αυτό το τρόπο την ανάγκη επεξεργασίας σε επίπεδο πακέτου και μειώνοντας σημαντικά το υπολογιστικό κόστος όταν το πλήθος των καταγραφών είναι μεγάλο[23].

4.2.2 Κατηγορίες DoS / DDoS επιθέσεων και ετικέτες (labels)

Γενικότερα, το σύνολο δεδομένων CIC-DDoS2019 περιλαμβάνει κάποιες ανακλαστικές επιθέσεις τύπου DDoS που καλύπτουν διαφορετικά πρωτόκολλα και διαφορετικούς μηχανισμούς ενίσχυσης (amplification), όπως για παράδειγμα τα PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS και SNMP[23]. Η ποικιλία αυτή επιτρέπει την αξιολόγηση των μοντέλων σε πολλές διαφορετικές μορφές DoS/DDoS, όπου η κακόβουλη κίνηση μπορεί να βασίζεται είτε σε καταχρηστική χρήση των πρωτοκόλλων UDP και των μηχανισμών της αντανάκλασης (reflection) είτε σε τεχνικές εξάντλησης των πόρων μέσω TCP / UDP ροών. Η λογική της ομαδοποίησης και της ταξινόμησης των DDoS επιθέσεων μπορεί να αποτυπωθεί σχηματικά μέσω μίας ταξινομίας, όπου οι επιθέσεις διαχωρίζονται σε κατηγορίες όπως οι αντανακλαστικές και ενισχυτικές επιθέσεις (reflection and amplification attacks) εναντίον των επιθέσεων ευπαθειών (exploitation attacks), καθώς και σε υποκατηγορίες ανά πρωτόκολλο ή μηχανισμό υλοποίησης όπως παρουσιάζεται και ενδεικτικά στο Σχήμα 4.1[24]. Στο οποίο παρουσιάζονται οι επιθέσεις που σχετίζονται με DNS, NTP, SSDP, SNMP

(στο πλαίσιο των reflection /amplification) αλλά και σενάρια όπως SYN flood, UDP flood και UDP-Lag, τα οποία συνδέονται με την εξάντληση των πόρων στο επίπεδο TCP / UDP[24].



Σχήμα 4.1: Taxonomy of DDoS attack[24]

Επιπρόσθετα, οι επιθέσεις εκτελούνται σε διακριτά χρονικά διαστήματα μέσα στην ημέρα καταγραφής ώστε να διαχωρίζονται κατανοητά οι περίοδοι του καλοήθους υπόβαθρου κίνησης (benign, background traffic) από τις περιόδους επίθεσης και να διευκολύνεται η αντιστοίχιση των ετικετών (labels) στα αντίστοιχα τμήματα της κίνησης[23]. Επίσης, σύμφωνα με την επίσημη περιγραφή του συνόλου δεδομένων κατά την ημέρα της εκπαίδευσης (training day) εκτελέστηκαν δώδεκα σενάρια επιθέσεων (attack scenarios) (NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag. WebDDoS, SYN, TFTP), ενώ κατά την ημέρα ελέγχου (testing day) εκτελέστηκαν επτά σενάρια (PortScan, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN)[23]. Ακόμη, επισημαίνεται ότι ο όγκος της κίνησης για το WebDDoS είναι ιδιαίτερα χαμηλός, ενώ το PortScan εκτελείται μόνο στην ημέρα ελέγχου (testing day), ώστε να λειτουργεί ως άγνωστο μοτίβο (unseen pattern) κατά την αξιολόγηση και να αποτυπώνεται πιο αληθινά η ικανότητα της γενίκευσης (generalization) των μοντέλων στα μη παρατηρημένα σενάρια[23].

4.2.3 Βασικά ζητήματα ποιότητας δεδομένων (Ανισορροπία, διπλότυπες ή σχεδόν ταυτόσημες καταγραφές, διαρροή)

Επιπρόσθετα, η επίσημη τεκμηρίωση του CIC-DDoS2019 δίνει έμφαση στη δημιουργία ρεαλιστικής καλοήθους υποβάθρου κίνησης (benign background), αναφέροντας ότι αξιοποιήθηκε το σύστημα B-Profile για τη προσομοίωση των φυσιολογικών αλληλεπιδράσεων των χρηστών στο δοκιμαστικό περιβάλλον (testbed)[23]. Στο πλαίσιο αυτό, περιγράφονται οι δραστηριότητες που σχετίζονται με πρωτόκολλα και υπηρεσίες όπως τα HTTP/HTTPS, FTP, SSH και το ηλεκτρονικό ταχυδρομείο (email), με στόχο οι περίοδοι της κανονικής λειτουργίας να προσεγγίζουν πιο ρεαλιστικά τα πραγματικά περιβάλλοντα[23].

Βέβαια, στο πλαίσιο της συγκεκριμένης διπλωματικής εκτός από την επίσημη περιγραφή του συνόλου δεδομένων, η προεπεξεργασία που ακολουθεί στη συνέχεια θεωρείται μία πολύ κρίσιμη προϋπόθεση ώστε τα δεδομένα να είναι κατάλληλα για εκπαίδευση και για δίκαιη αξιολόγηση των μοντέλων.

Ειδικότερα, στα σύνολα δεδομένων μεγάλης κλίμακας (large scale datasets) τα οποία βασίζονται σε ροές (flow-based) συναντώνται κάποια πρακτικά ζητήματα όπως η ανισορροπία των κλάσεων (class imbalance) μεταξύ κανονικής (benign) και επιθετικής κίνησης (attack), η πλεονάζουσα πληροφορία (redundancy) και τα χαρακτηριστικά που ενδέχεται να προκαλέσουν μια υπερεκτίμηση της απόδοσης λόγω διαρροής της πληροφορίας (data leakage)[23]. Για το λόγο αυτό, στην εργασία αυτή εφαρμόζονται μερικά βήματα καθαρισμού (data cleaning), κανονικοποίησης (normalization) και επιλογής των χαρακτηριστικών (feature selection), με ρητή διάκριση μεταξύ των συνόλων χωρίς διαρροή (non-leaky) και με διαρροή (leaky) σύμφωνα με τη μεθοδολογία που χρησιμοποιείται.

4.3 Προεπεξεργασία του συνόλου δεδομένων

Αρχικά, η προεπεξεργασία του CIC-DDoS2019 είναι ένα αναγκαίο στάδιο πριν από την εκπαίδευση των μοντέλων, τόσο λόγω της κλίμακας του, διότι δέχεται δεκάδες εκατομμύρια ροές και δεκάδες γνωρίσματα ανά ροή, όσο και λόγω της έντονης ανισορροπίας μεταξύ των καλοήθων (benign) και των κακόβουλων ροών (malicious), η οποία μπορεί να οδηγήσει σε παραπλανητικές μετρικές απόδοσης αν δεν αντιμετωπιστεί συχνά. Ενδεικτικά, η κατανομή των ροών ανά κατηγορία εμφανίζει μεγάλες αποκλίσεις στο πλήθος των εγγραφών μεταξύ των κλάσεων όπως αποτυπώνεται και στο Πίνακα 4.1, όπου ορισμένοι τύποι DDoS συγκεντρώνουν πολύ μεγαλύτερο όγκο ροών σε σχέση με άλλους[24]. Επιπλέον, στα σύνολα δεδομένων τα οποία είναι βασισμένα σε ροές (flow-based datasets) η ίδια η διαδικασία της εξαγωγής και της σήμανσης (labeling) των ροών μπορεί να εισάγει γνωρίσματα ή μεταπληροφορία που συνδέονται σε μεγάλο βαθμό με την ετικέτα, αυξάνοντας με αυτό το τρόπο τον κίνδυνο διαρροής της πληροφορίας (data leakage)[24].

Πίνακας 4.1: Attack types in CICDDoS2019 dataset[24]

Attack type	Flow count
Benign	56,863
DDoS DNS	5,071,011
DDoS LDAP	2,179,930
DDoS MSSQL	4,522,492
DDoS NetBIOS	4,093,279
DDoS NTP	1,202,642
DDoS SNMP	5,159,870
DDoS SSDP	2,610,611
DDoS SYN	1,582,289
DDoS TFTP	20,082,580
DDoS UDP	3,134,645
DDoS UDP-Lag	366,461

4.3.1 Καθαρισμός και χειρισμός ελλιπών και άκυρων τιμών

Πρώτα από όλα, στο αρχικό στάδιο γίνεται η ενοποίηση των επιμέρους αρχείων, για παράδειγμα ανά ημέρα, μηχάνημα και σενάριο, σε έναν ενιαίο πίνακα με ομογενοποίηση των ονομάτων στηλών και των τύπων δεδομένων, ώστε όλα τα δείγματα να περιγράφονται με σωστό τρόπο. Ακολούθως, εφαρμόζεται ένας έλεγχος πληρότητας και εγκυρότητας, με στόχο την απομάκρυνση των γραμμών που περιέχουν μη ορισμένες τιμές (missing / NaN), άπειρες τιμές (+ / - Inf) ή τιμές που προκύπτουν ως αριθμητικά τεχνουργήματα[24]. Σε σύνολα που προέρχονται από εργαλεία εξαγωγής των ροών (flow extraction tools), τέτοιου είδους περιπτώσεις εμφανίζονται συχνά σε χαρακτηριστικά των ρυθμών ή των

αναλογιών (πχ bytes/s ή pkt/s), όταν ο παρονομαστής είναι ίσος με το μηδέν, δηλαδή έχει μηδενική διάρκεια ροής ή όταν υπάρχουν ακραίες χρονικές τιμές. Σκοπός του είναι να διασφαλιστεί ότι το τελικό σύνολο της εκπαίδευσης δεν περιέχει τιμές που να αλλοιώνουν τη κλιμάκωση, τις στατιστικές ιδιότητες και τη συμπεριφορά του αλγορίθμου της μάθησης[24].

Παράλληλα, εξετάζεται η ύπαρξη της πλεονάζουσας πληροφορίας μέσα από τα διπλότυπα ή τις σχεδόν ταυτόσημες εγγραφές. Η απομάκρυνση τέτοιων εγγραφών είναι σημαντική καθώς μπορεί να ευνοήσει τεχνητά την απόδοση, ιδιαίτερα όταν πολύ παρόμοια δείγματα καταλήγουν ταυτόχρονα σε σύνολο εκπαίδευσης (training set) και σε σύνολο ελέγχου (test set), αλλά και να ενισχύσει περισσότερο την επίδραση της ήδη έντονης ανισορροπίας που παρατηρείται στη σύνθεση του συνόλου δεδομένων (dataset)[24].

4.3.2 Κανονικοποίηση και κλιμάκωση χαρακτηριστικών

Στη συνέχεια, μετά τον καθαρισμό εφαρμόζεται η κανονικοποίηση και η κλιμάκωση στα αριθμητικά γνωρίσματα, ώστε τα χαρακτηριστικά να βρίσκονται σε συγκρίσιμο αριθμητικό εύρος και να αποφεύγεται η κυριαρχία των γνωρισμάτων με πολύ μεγάλες τιμές, για παράδειγμα με μέγεθος κάποια bytes έναντι γνωρισμάτων μικρότερης κλίμακας δηλαδή μετρήσεις IAT κ.ά. Παρότι, τα μοντέλα τα οποία είναι βασισμένα στα δέντρα (tree-based models) Random Forest και XGBoost είναι γενικά λιγότερο ευαίσθητα σε διαφορετικές κλίμακες η κλιμάκωση παραμένει χρήσιμη για την τυποποίηση της ροής της επεξεργασίας (pipeline), για σταθερότερη αριθμητική συμπεριφορά και για τη δυνατότητα της άμεσης σύγκρισης με τους εναλλακτικούς ταξινομητές[24].

Γενικά, η κλιμάκωση πραγματοποιείται με τρόπο που δεν εισάγει διαρροή από το σύνολο ελέγχου, καθώς οι παράμετροι της κανονικοποίησης, για παράδειγμα η μέση τιμή, η τυπική απόκλιση και το ελάχιστο-μέγιστο, υπολογίζονται αποκλειστικά στο σύνολο εκπαίδευσης (training set) και στη συνέχεια εφαρμόζονται αυτούσιοι στο σύνολο επικύρωσης (validation set) και στο σύνολο ελέγχου και δοκιμής (test set), εξασφαλίζοντας την ορθότητα της πειραματικής διαδικασίας[24].

4.3.3 Επιλογή των χαρακτηριστικών (feature selection)

Η επιλογή των χαρακτηριστικών έχει ως στόχο την μείωση της διαστατικότητας, την απομάκρυνση μη ενημερωτικών ή πλεονάζοντων γνωρισμάτων και τη βελτίωση της γενίκευσης. Σχετικές προσεγγίσεις στη βιβλιογραφία αξιοποιούν κάποιες μεθόδους κατάταξης και σημαντικότητας των χαρακτηριστικών, όπως η εκτίμηση της συνεισφοράς των γνωρισμάτων με σχήματα βασισμένα στον Random Forest, για τον εντοπισμό των πλέον διακριτικών χαρακτηριστικών ανά σενάριο επίθεσης[24].

Στην εργασία αυτή, η επιλογή των χαρακτηριστικών δεν αντιμετωπίζεται μόνο ως πρόβλημα βελτιστοποίησης της απόδοσης αλλά και ως μηχανισμός ελέγχου της εγκυρότητας, διότι αφαιρούνται χαρακτηριστικά που δεν είναι κατάλληλα για τη ρεαλιστική ανίχνευση, όπως τα καθαρά αναγνωριστικά πεδία αλλά και τα χαρακτηριστικά που δημιουργούν αυξημένο κίνδυνο διαρροής των δεδομένων. Με αυτό το τρόπο, το τελικό σύνολο των χαρακτηριστικών αντανακλά τη πληροφορία που θα ήταν διαθέσιμη και σε πραγματικές συνθήκες παρακολούθησης της δικτυακής κίνησης, χωρίς έμμεση αποκάλυψη της ετικέτας[24].

4.3.4 Διαχωρισμός των χαρακτηριστικών με διαρροή εναντίον των χαρακτηριστικών χωρίς διαρροή (leaky vs non-leaky)

Ιδιαίτερη έμφαση δίνεται στον διαχωρισμό των συνόλων χαρακτηριστικών με διαρροή (leaky) και των συνόλων χαρακτηριστικών χωρίς διαρροή (non leaky), ώστε να αποτυπωθεί πειραματικά η επίδραση της διαρροής της πληροφορίας στις μετρικές αξιολόγησης. Η ανάγκη για αυτό το διαχωρισμό τεκμηριώνεται από το γεγονός ότι σε αναπαραστάσεις βασισμένες σε ροές, σε μεταδεδομένα όπως οι διευθύνσεις IP, σε θύρες, σε πρωτόκολλα και σε χρονική σήμανση μπορούν να συνδεθούν άμεσα με το τρόπο που οργανώθηκαν τα σενάρια των επιθέσεων και η επισήμανση των ροών[24]. Στο Πίνακα 4.2 παρουσιάζεται ότι οι επιθέσεις αντιστοιχίζονται σε συγκεκριμένα χρονικά παράθυρα (detection times) ανά ημέρα, γεγονός που κάνει τη χρονική πληροφορία και τα συναφή μεταδεδομένα ιδιαίτερα ισχυρά ως προς τη πρόβλεψη της κλάσης, άρα και υποψήφια για διαρροή αν χρησιμοποιηθούν άκριτα στο κομμάτι της εκπαίδευσης[24]

Πίνακας 4.2: Day-wise attacks were detected with the time of detection[24]

Days	Attacks (detection time)
First day (01-11-2019)	PortMap (9:43–9:51)
	NetBIOS (10:00–10:09)
	LDAP (10:21–1030)
	MSSQL (10:33–10:42)
	UDP (10:53–11:03)
	UDP-Lag (11:14–11:24)
	SYN (11:28–17:35)
	NTP (10:35–10:45)
	DNS (10:52–11:05)
	MAP (11:22–11:32)
	MSSQL (11:36–11:45)
Second day (03-11-2019)	NetBIOS (11:50–12:00)
	SKIMP (12:12–12:23)
	SSDP (12:27–12:37)
	UDP (12:45–13:09)
	UDP-Lag (13.11–13:15)
	WebDDoS (13:18–13:29)
	SYN (13:29–13:34)
	TFTP (13:35–17:15)

Με βάση όσα αναφέρθηκαν παραπάνω, στα σύνολα των χαρακτηριστικών χωρίς διαρροή (non leaky) διατηρούνται χαρακτηριστικά που περιγράφουν τη κίνηση με όρους στατιστικών και χρονικών μεγεθών της ροής, ενώ αφαιρούνται ή ελέγχονται αυστηρά τα πεδία τα οποία λειτουργούν ως αναγνωριστές ή ως συντομεύσεις προς την ετικέτα[24]. Από την άλλη μεριά, στο σύνολο των χαρακτηριστικών με διαρροή (leaky) διατηρούνται ελεγχόμενα και τέτοιου τύπου πεδία και γνωρίσματα, ώστε να αποτυπωθεί πώς ένα μοντέλο μπορεί να εμφανίσει φαινομενικά εξαιρετική απόδοση, η οποία όμως δεν αντανακλά την πραγματική ικανότητα της γενίκευσης[24]. Το αποτέλεσμα της διαδικασίας είναι δύο παράλληλα, συγκρίσιμα σύνολα δεδομένων, τα οποία τροφοδοτούνται στα ίδια μοντέλα και αξιολογούνται με κοινή μεθοδολογία στα επόμενα στάδια.

4.4 Επίλογος κεφαλαίου

Στο κεφάλαιο αυτό παρουσιάστηκε το σύνολο δεδομένων CIC-DDoS2019 και τεκμηριώθηκε ο ρόλος του ως βασική πηγή δεδομένων για τη πειραματική υλοποίηση της εργασίας πάνω στη ημέρα 11/03/2019 του dataset σε σενάρια επιθέσεων της άρνησης εξυπηρέτησης (DoS/DDoS). Αρχικά, αναλύθηκε η δομή του συνόλου δεδομένων (dataset) και η λογική της αναπαράστασης στο επίπεδο των ροών (flow based), ώστε να γίνει σαφές πώς από την με επεξεργασμένη δικτυακή κίνηση προκύπτουν μαζεμένα κάποια χαρακτηριστικά τα οποία είναι κατάλληλα για να αξιοποιηθούν από αλγορίθμους της Μηχανικής Μάθησης. Παράλληλα, παρουσιάστηκαν οι κατηγορίες των επιθέσεων και ο τρόπος με τον

οποίο η κίνηση επισημαίνεται χρονικά με ετικέτες, επιτρέποντας να διακριθούν οι περίοδοι της κανονικής λειτουργίας από τις περιόδους επίθεσης.

Στη συνέχεια, δόθηκε έμφαση στα βασικά πρακτικά ζητήματα της ποιότητας των δεδομένων που επηρεάζουν άμεσα την εγκυρότητα των πειραμάτων, όπως η έντονη ανισορροπία μεταξύ της κανονικής και της κακόβουλης κίνησης, η πιθανή ύπαρξη της πλεονάζουσας και της διπλότυπης πληροφορίας και ο κίνδυνος της διαρροής πληροφορίας μέσω των χαρακτηριστικών, τα οποία λειτουργούν ως έμμεσοι αναγνωριστές των σεναρίων. Με βάση αυτές τις παρατηρήσεις, αναλύθηκε ένα σαφής και τεκμηριωμένος αγωγός προεπεξεργασίας των δεδομένων (preprocessing pipeline), ο οποίος περιλαμβάνει τον καθαρισμό, τον χειρισμό των άκυρων ή ελλιπών τιμών, την κανονικοποίηση και τη κλιμάκωση όπου αυτό χρειάζεται, καθώς και την επιλογή των χαρακτηριστικών με στόχο την αξιόπιστη εκπαίδευση και τη δίκαιη αξιολόγηση.

Ακόμη, ένα πολύ σημαντικό αποτέλεσμα του κεφαλαίου είναι ότι η προεπεξεργασία δεν αντιμετωπίζεται ως ένα τυπικό τεχνικό βήμα, αλλά ως μια προϋπόθεση για να βγουν κάποια συμπεράσματα με πραγματική σημασία. Για το λόγο αυτό, υιοθετήθηκε ένας σημαντικός διαχωρισμός μεταξύ των συνόλων των χαρακτηριστικών χωρίς διαρροή (non-leaky) και των χαρακτηριστικών με διαρροή (leaky), έτσι ώστε να δειχθεί πειραματικά πώς μπορεί να προκύψει φαινομενικά υψηλή απόδοση όταν το μοντέλο αξιοποιεί την πληροφορία που δεν θα είναι διαθέσιμη ή σταθερή στις ρεαλιστικές συνθήκες λειτουργίας. Με αυτό το τρόπο, διαμορφώνονται δύο βάσεις σύγκρισης οι οποίες υποστηρίζουν τόσο τη ρεαλιστική αξιολόγηση όσο και τη κριτική ερμηνεία των αποτελεσμάτων.

Κλείνοντας, στο επόμενο κεφάλαιο αξιοποιούνται τα τελικά επεξεργασμένα σύνολα των δεδομένων και εφαρμόζονται οι αλγόριθμοι Random Forest και XGBoost, ώστε να παρουσιαστούν τα πειραματικά αποτελέσματα και να γίνει η συγκριτική ανάλυση της απόδοσης, τόσο συνολικά όσο και κάτω από την επίδραση που έχει η επιλογή των χαρακτηριστικών, δηλαδή τα χαρακτηριστικά χωρίς διαρροή (non-leaky) εναντίον των χαρακτηριστικών με διαρροή (leaky) στη συμπεριφορά των μοντέλων.

Κεφάλαιο 5ο: Μελέτη Περίπτωσης: Υλοποίηση και Ενσωμάτωση Μοντέλων Μηχανικής Μάθησης σε Σύστημα Παρακολούθησης της Ασφάλειας (CM) με OpenSearch

5.1 Εισαγωγή κεφαλαίου

Το συγκεκριμένο κεφάλαιο παρουσιάζει το πρακτικό σκέλος της διπλωματικής υπό τη μορφή της μελέτης περίπτωσης (case study) και περιγράφει αναλυτικά τη μεθοδολογία της υλοποίησης ενός συστήματος Παρακολούθησης της Ασφάλειας (CM) πάνω στην πλατφόρμα του OpenSearch. Η υλοποίηση και τα πειράματα του πρακτικού σκέλους βασίστηκαν σε δεδομένα του CIC-DDoS2019 που αντιστοιχούν αποκλειστικά στην ημέρα 11/03/2019. Επομένως η αξιολόγηση που προκύπτει από τα πειράματα του κεφαλαίου ερμηνεύεται ως ενδοημερήσια διάσπαση της εκπαίδευσης και του ελέγχου (within-day holdout) . Στόχος είναι η μεταφορά των αποτελεσμάτων της Μηχανικής Μάθησης από το πειραματικό στάδιο της εκπαίδευσης σε μια ολοκληρωμένη ροή λειτουργίας, όπου η δικτυακή κίνηση που εισέρχεται εμπλουτίζεται αυτόματα με προβλέψεις, αποθηκεύεται σε κατάλληλες δομές δεδομένων, οπτικοποιείται σε πίνακες ελέγχου (dashboards) και ενεργοποιεί τους μηχανισμούς ειδοποίησης όταν εντοπίζεται ύποπτη ή κακόβουλη δραστηριότητα.

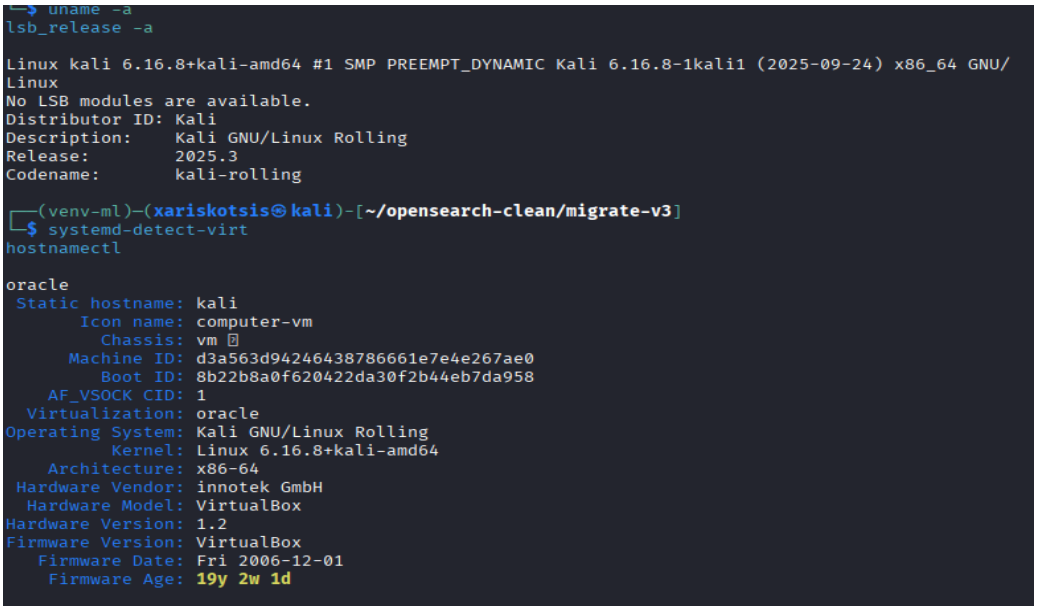
Γενικά, η υλοποίηση βασίζεται σε δύο ταξινομητές, τον Random Forest και τον XGBoost, οι οποίοι έχουν εκπαιδευτεί πάνω σε επεξεργασμένες ροές του συνόλου δεδομένων CIC-DDoS2019, αποκλειστικά από την ημέρα 11/03/2019. Η διαδικασία σχεδιάστηκε ώστε να επιτρέπει τόσο τη ρεαλιστική αξιολόγηση μέσω των συνόλων των χαρακτηριστικών χωρίς διαρροή της πληροφορίας (non-leaky), όσο και τον ελεγχόμενο πειραματισμό με σύνολα που διατηρούν τα γνωρίσματα του υψηλού κινδύνου της διαρροής (leaky), προκειμένου να φανεί πρακτικά η επίδραση του φαινομένου της διαρροής πληροφορίας (data leakage) στις μετρικές απόδοσης και στη συμπεριφορά ενός συστήματος Παρακολούθησης της Ασφάλειας (CM). Στο πλαίσιο αυτό, δίνεται μεγάλη έμφαση στην αναπαραγωγιμότητα και στη διαλειτουργικότητα, καθώς τα εκπαιδευόμενα μοντέλα εξάγονται σε μορφή ONNX (Open Neural Network Exchange) και αξιοποιούνται μέσω ενός ανεξάρτητου προγνωστικού εξυπηρετητή REST (REST predictor), ο οποίος λειτουργεί ως υπηρεσία πρόβλεψης στο υποσύστημα της υποστήριξης (backend prediction service).

Στη συνέχεια, παρουσιάζονται η ενσωμάτωση του προγνωστικού εξυπηρετητή (predictor) στο OpenSearch μέσω των μηχανισμών των συνδετήρων της Μηχανικής Μάθησης (ML connectors) και της απομακρυσμένης κλήσης του μοντέλου (remote model invocation), ώστε οι προβλέψεις να εκτελούνται από το ίδιο το οικοσύστημα του OpenSearch και να εντάσσονται σε έναν αγωγό εμπλουτισμού (enrichment pipeline) της Μηχανικής Μάθησης κατά την εισαγωγή των δεδομένων (data ingestion). Με αυτό το τρόπο, οι ροές της κίνησης αποκτούν επιπλέον πεδία όπως η προβλεπόμενη κλάση, η πιθανότητα ή βαθμολογία της εμπιστοσύνης και τα αναγνωριστικά του μοντέλου, τα οποία αποθηκεύονται σε κάποιους ειδικούς δείκτες (indices) για βαθύτερη ανάλυση. Η πρακτική αξία της προσέγγισης αποτυπώνεται σε τρία επίπεδα, πρώτον στη δυνατότητα παρακολούθησης της εξέλιξης των συμβάντων και της κατανομής των προβλέψεων σε πραγματικό ή ημιπραγματικό χρόνο. Δεύτερον, στη συγκριτική οπτικοποίηση των αποτελεσμάτων μεταξύ των Random Forest και XGBoost και τρίτον στην ενεργοποίηση των μηχανισμών ειδοποίησης (alerting) μέσω του webhook, που υποστηρίζουν την επιχειρησιακή χρήση του συστήματος ως ένα σύστημα έγκαιρης προειδοποίησης.

Δομικά, το κεφάλαιο ξεκινάει με τη περιγραφή της εκπαίδευσης των μοντέλων και της εξαγωγής τους σε μορφή ONNX, συνεχίζει με την ανάπτυξη του προγνωστικού εξυπηρετητή REST και με την οριοθέτηση της μορφής εισόδου και εξόδου των προβλέψεων και μετά παρουσιάζει τη διασύνδεση με το OpenSearch μέσω των συνδετήρων (connectors) και των απομακρυσμένων προβλέψεων (remote predictions). Ακολούθως, αναλύονται οι αγωγοί της εισαγωγής (ingest pipelines) και οι δείκτες (indices) που υποστηρίζουν τον εμπλουτισμό και την αποθήκευση των αποτελεσμάτων, καθώς και οι πίνακες ελέγχου (dashboards) που χρησιμοποιούνται για την οπτικοποίηση και την επιχειρησιακή παρακολούθηση. Τέλος, παρουσιάζεται ο μηχανισμός της ειδοποίησης (alerting) και συνοψίζεται πώς τα επιμέρους δομικά στοιχεία συνθέτουν ένα πλήρες αγωγό Παρακολούθησης της Ασφάλειας (CM pipeline) πάνω στη πλατφόρμα του OpenSearch, το οποίο μπορεί να υποστηρίξει τη συνεχή επιτήρηση της δικτυακής κίνησης με χρήση των μοντέλων της Μηχανικής Μάθησης.

5.2 Περιβάλλον πειραματισμού και αρχιτεκτονική συστήματος Παρακολούθησης της Ασφάλειας (CM)

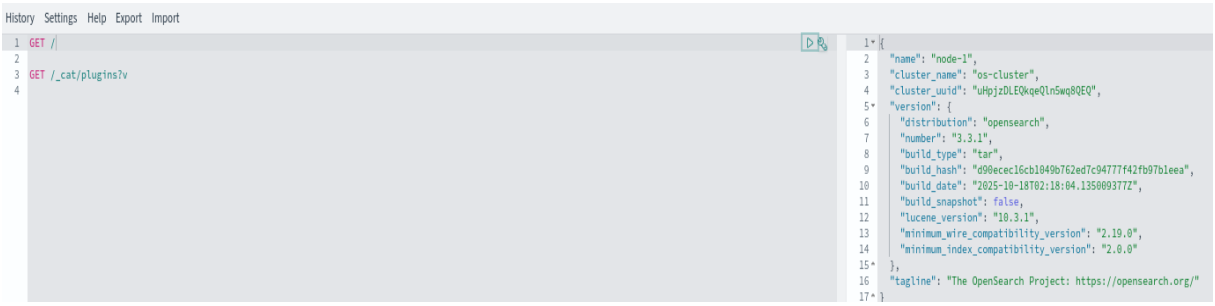
Το πρακτικό σκέλος αυτής της εργασίας υλοποιήθηκε σε απομονωμένο περιβάλλον εικονικοποίησης (virtualized test environment) με στόχο τη σταθερή εκτέλεση των πειραμάτων και την επαναληψιμότητα (reproducibility) των αποτελεσμάτων. Η πειραματική πλατφόρμα εκτελεί το λειτουργικό Kali GNU / Linux μέσα στο Oracle VM VirtualBox, αυτό αποδεικνύει ότι η ανάπτυξη και η λειτουργία του συστήματος πραγματοποιείται στο επίπεδο της εικονικής μηχανής (virtual machine – VM), το οποίο φαίνεται καθαρά και στην Εικόνα 5.1. Η συγκεκριμένη επιλογή επιτρέπει στο χρήστη να έχει καλύτερο έλεγχο των πόρων και ένα απομονωμένο περιβάλλον εκτέλεσης, μειώνοντας έτσι τις εξωτερικές παρεμβολές και τις διακυμάνσεις που θα μπορούσαν να επηρεάσουν τα πειραματικά αποτελέσματα.



Εικόνα 5.1: Περιβάλλον εκτέλεσης

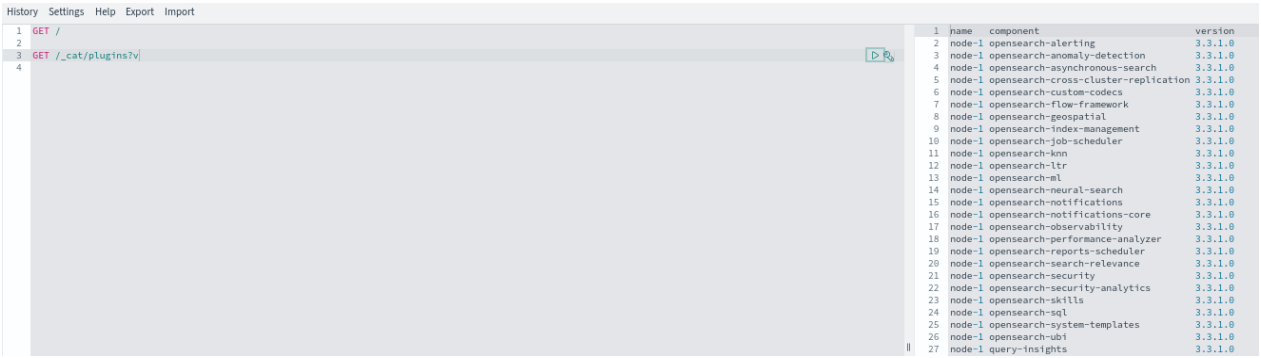
Επιπλέον, η υποδομή Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring) υλοποιήθηκε με ανάπτυξη σε δοχεία (containerized deployment) μέσω του Docker Compose, ώστε οι βασικές υπηρεσίες, όπως η αποθήκευση και η αναζήτηση, η οπτικοποίηση και η απομακρυσμένη πρόβλεψη να αναπτύσσονται με διαχωρισμό των ρόλων και με ελεγχόμενα σημεία πρόσβασης (endpoints). Συγκεκριμένα, στο αρχείο docker_compose.yml ορίζονται οι υπηρεσίες του OpenSearch και του OpenSearch Dashboards, μαζί με τις βασικές παραμέτρους λειτουργίας όπως η ανάπτυξη ενός κόμβου

(single-node deployment), οι θύρες πρόσβασης (ports) και οι τόμοι αποθήκευσης (volumes) για την επίμονη αποθήκευση (persistence). Με αυτό το τρόπο, η αρχιτεκτονική περιγράφεται δηλωτικά (declaratively) και παραμένει εύκολα επαναλήψιμη (reproducible) σε διαφορετικό υπολογιστή (host), χωρίς να χρειαστεί να γίνουν χειροκίνητες ρυθμίσεις (ad-hoc configuration).



Εικόνα 5.2: Επιβεβαίωση λειτουργίας και έκδοσης OpenSearch μέσω API

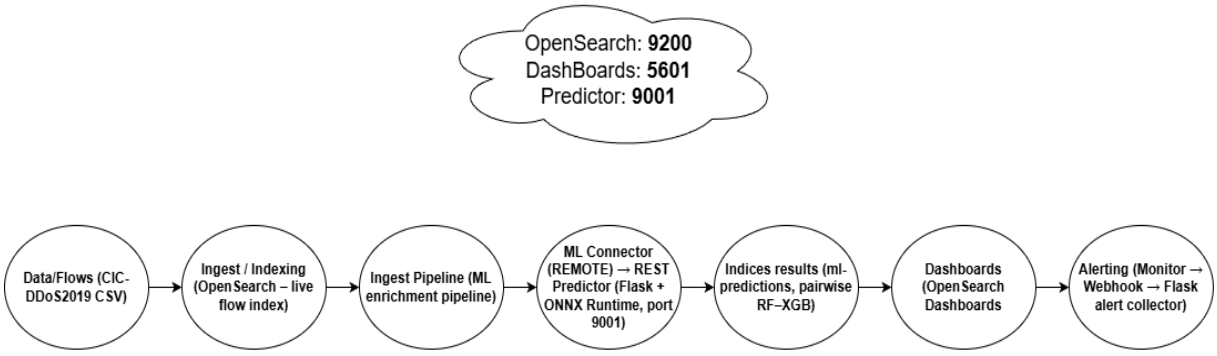
Επίσης, η ορθή λειτουργία του OpenSearch επιβεβαιώνεται στο επίπεδο της προγραμματιστικής διεπαφής (API), καθώς η απόκριση του συμπλέγματος (cluster) επιτρέπει την ταυτοποίηση της εγκατάστασης και της έκδοσης, κάτι που φαίνεται και στην Εικόνα 5.2. Παράλληλα, η διαθεσιμότητα των κρίσιμων δυνατοτήτων για την υλοποίηση ενός συστήματος Παρακολούθησης της Ασφάλειας, όπως η Μηχανική Μάθηση και οι μηχανισμοί της ειδοποίησης, αποδεικνύεται μέσω της λίστας των εγκατεστημένων πρόσθετων (plugins), τα οποία φαίνονται και στην Εικόνα 5.3. Άρα, τα στοιχεία των Εικόνων 5.1 ως 5.3 αποτελούν σαφή ένδειξη ότι το περιβάλλον έχει τις απαραίτητες υποδομές για την εισαγωγή των δεδομένων (ingest), την αναζήτηση και την αποθήκευση (search and storage), τον εμπλουτισμό (enrichment) με τη χρήση της Μηχανικής Μάθησης και των μηχανισμών ειδοποίησης. Έτσι, υπάρχει όλο το υπόβαθρο για ένα πλήρως λειτουργικό σύστημα Παρακολούθησης της Ασφάλειας.



Εικόνα 5.3: Επιβεβαίωση διαθέσιμων plugins

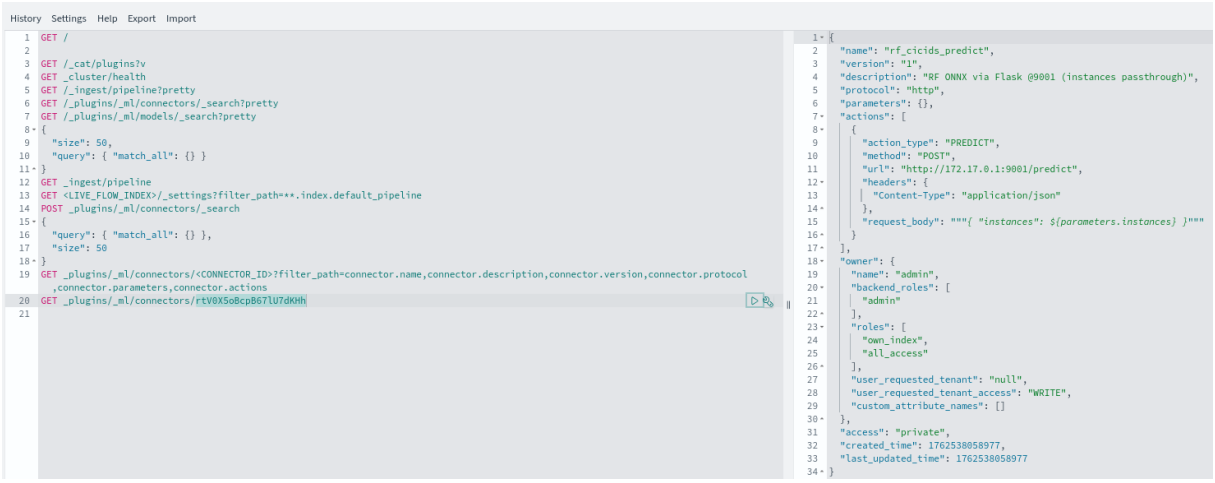
Στο επίπεδο της αρχιτεκτονικής, η ροή των δεδομένων υλοποιείται ως αγωγός της Παρακολούθησης της Ασφάλειας (CM pipeline), ο οποίος ξεκινάει από την εισαγωγή των εγγραφών της ροής (flow records) και καταλήγει στη παραγωγή των προβλέψεων, οι οποίες μετά είναι διαθέσιμες για οπτικοποίηση και για ενεργοποίηση των μηχανισμών ειδοποίησης. Η συνολική λογική παρουσιάζεται σχηματικά στο Σχήμα 5.1, όπου αποτυπώνεται η ακολουθία των βασικών σταδίων ξεκινώντας από την εισαγωγή και την ευρετηρίαση των δεδομένων (ingest / indexing) στο OpenSearch, μετά περνάμε στην εφαρμογή του αγωγού της εισαγωγής (ingest pipeline) με εμπλουτισμό μέσω της Μηχανικής Μάθησης (ML enrichment). Στη συνέχεια, έρχεται η εκτέλεση των προβλέψεων μέσω της απομακρυσμένης κλήσης του μοντέλου (remote inference) και τέλος η αποθήκευση των αποτελεσμάτων σε κατάλληλους δείκτες (indices), ώστε να υποστηρίζεται τόσο η επιχειρησιακή παρακολούθηση μέσω των πινάκων

ελέγχου (dashboards) όσο και η ενεργοποίηση των παρακολουθητών (monitors) και των καναλιών της ειδοποίησης (alerting). Ακόμη, η διάταξη του Σχήματος 5.1 προβάλλει ότι η προτεινόμενη λύση δεν περιορίζεται στις εκτός σύνδεσης διαδικασίες αλλά είναι ένα ολοκληρωμένο σύστημα από άκρη σε άκρη καθώς συνδυάζει την αποθήκευση και την αναζήτηση, τον εμπλουτισμό με τις τεχνικές της Μηχανικής Μάθησης, την επιχειρησιακή απεικόνιση και τους μηχανισμούς της έγκαιρης ειδοποίησης.

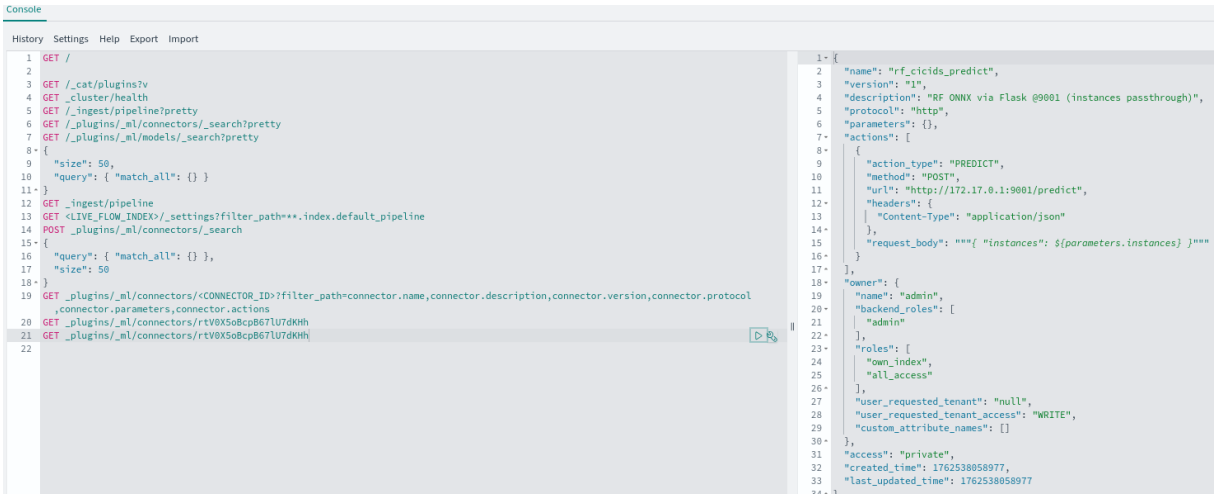


Σχήμα 5.1: .Συνολική αρχιτεκτονική CM pipeline

Γενικά, ο κρίσιμος σύνδεσμος μεταξύ του OpenSearch και του προγνωστικού εξυπηρετητή (predictor) υλοποιείται με τη χρήση των συνδετήρων της Μηχανικής Μάθησης (ML connectors) σε απομακρυσμένη λειτουργία, ώστε οι προβλέψεις να εκτελούνται εκτός της πλατφόρμας σε υπηρεσία Flask με χρόνο εκτέλεσης (Runtime) ONNX αλλά να καλούνται με τυποποιημένο και ενοποιημένο τρόπο από το οικοσύστημα του OpenSearch. Στην υλοποίηση έχουν οριστεί διακριτοί συνδετήρες για τα δύο μοντέλα, το Random Forest και το XGBoost, οι οποίοι δρομολογούν τα αιτήματα προς το αντίστοιχο σημείο πρόσβασης (endpoint) του predictor. Στην Εικόνα 5.4 παρουσιάζεται ο συνδετήρας που αντιστοιχεί στο μοντέλο Random Forest, ενώ στην Εικόνα 5.5 παρουσιάζεται ο αντίστοιχος συνδετήρας για το μοντέλο XGBoost. Έτσι, η επιλογή αυτή υποστηρίζει τον διαχωρισμό των αρμοδιοτήτων, το OpenSearch αναλαμβάνει την εισαγωγή των δεδομένων (ingest), τους αγωγούς εισαγωγής (pipelines), την αποθήκευση (storage) και τη παρακολούθηση (monitoring), ενώ ο predictor αναλαμβάνει την εκτέλεση του συμπερασμού ή της πρόβλεψης (inference) σε μορφή ONNX. Με αυτό το τρόπο, γίνεται πιο εύκολη τόσο η ανάπτυξη όσο και η μελλοντική αντικατάσταση ή η αναβάθμιση των μοντέλων χωρίς να απαιτούνται αλλαγές στο πυρήνα του αγωγού της Παρακολούθησης της Ασφάλειας (CM pipeline).



Εικόνα 5.4: Ανάκτηση ρυθμίσεων ML Connector για Random Forest



Εικόνα 5.5: Ανάκτηση ρυθμίσεων ML Connector για XGBoost

5.3 Εκπαίδευση των μοντέλων Random Forest και XGBoost

5.3.1 Πειραματικός Σχεδιασμός (split / validation)

Επιπρόσθετα, ο πειραματικός σχεδιασμός της εκπαίδευσης οργανώθηκε με στόχο αφενός την επαναληψιμότητα των αποτελεσμάτων και αφετέρου τη δίκαιη σύγκριση των μοντέλων Random Forest και XGBoost κάτω από ίδιες συνθήκες των δεδομένων και της αξιολόγησης. Η διαδικασία βασίστηκε στη προσέγγιση του διαχωρισμού της εκπαίδευσης-ελέγχου (hold-out evaluation), με αναλογία 80/20 (train_size=0.8) εντός της ημέρας 11/03/2019. Ο διαχωρισμός υλοποιήθηκε με τυχαία δειγματοληψία και στρωματοποιημένη κατατομή (stratified split) ως προς την ετικέτα (stratify=y), ώστε η αναλογία των κλάσεων να παραμένει κατά προσέγγιση ίδια στο σύνολο της εκπαίδευσης και στο σύνολο του ελέγχου. Επιπλέον, επειδή το πρακτικό σκέλος βασίζεται μόνο στη ημέρα 11/03/2019, η αξιολόγηση που προκύπτει ερμηνεύεται ως ενδοημερήσια διάσπαση της εκπαίδευσης και του ελέγχου (within-day holdout).

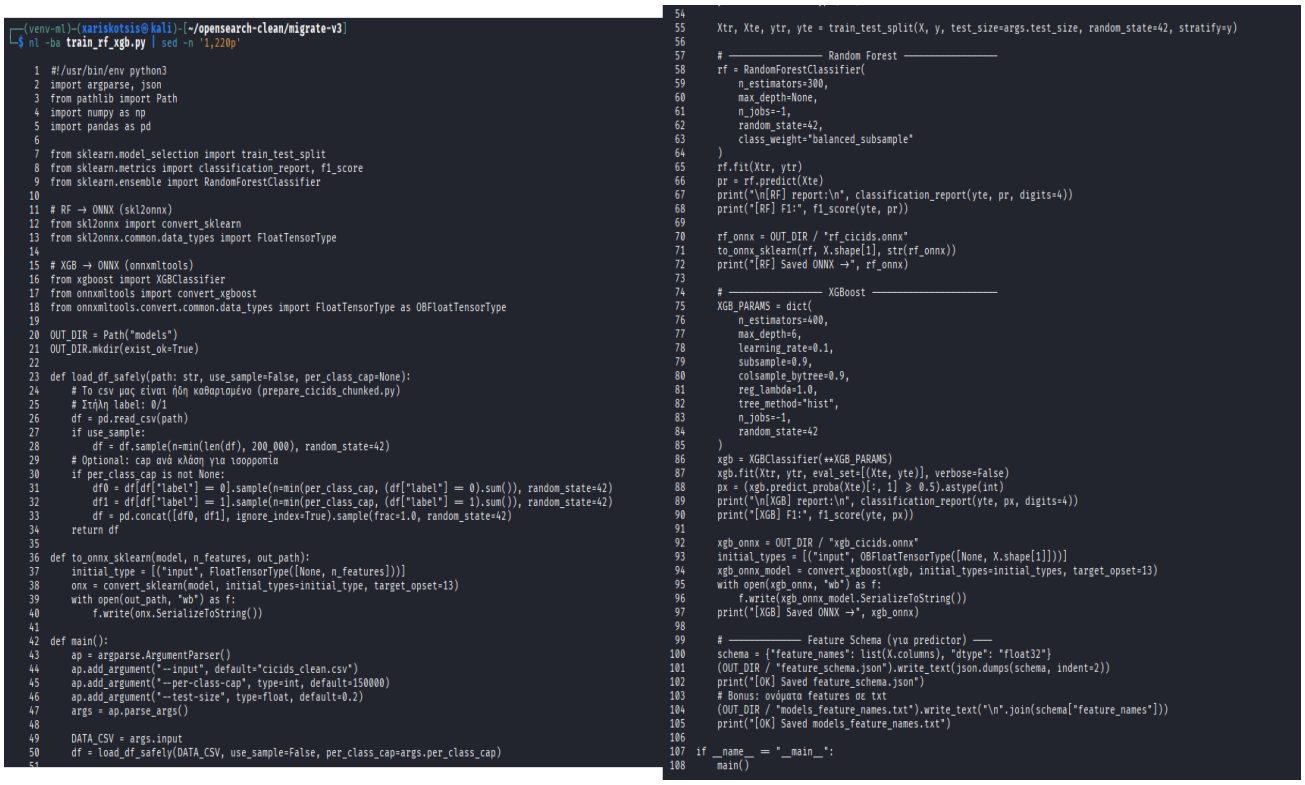
Ακόμη, ένα κρίσιμο στοιχείο της σχεδίασης είναι ότι το σύνολο δεδομένων παρουσιάζει ισχυρή ανισορροπία κλάσεων, όπως φαίνεται και από την κατανομή της ετικέτας στην Εικόνα 5.6. Για το λόγο αυτό, ο διαχωρισμός υλοποιήθηκε με στρωματοποιημένη δειγματοληψία (stratified split, stratify=y),

ώστε η αναλογία των κλάσεων να παραμένει κατά προσέγγιση ίδια τόσο στο σύνολο της εκπαίδευσης όσο και στο σύνολο του ελέγχου. Με το τρόπο αυτό, μειώνονται οι πιθανές στρεβλώσεις (bias) στην αξιολόγηση, όπως η τεχνική αύξησης της ακρίβειας (accuracy) όταν η πλειοψηφική κλάση εμφανίζεται σε πολύ μεγαλύτερο ποσοστό στο σύνολο ελέγχου από ότι στο αρχικό σύνολο των δεδομένων.

```
python - <<'PY'
import pandas as pd

paths = [
  "cicids_clean.csv",
  "data/CICIDS2019_13f_norm_clean.csv",
  "data/CICIDS2019_leaky_norm_clean.csv",
]

for path in paths:
    try:
        hdr = pd.read_csv(path, nrows=0)
    except FileNotFoundError:
        continue

    print("\n=== FILE:", path, "===")
    print("Columns:", len(hdr.columns))
    label_col = next((c for c in ["Label","label","Class","class"] if c in hdr.columns), None)
    print("Label column:", label_col)

    if label_col:
        s = pd.read_csv(path, usecols=[label_col])[label_col]
        print("Top classes:")
        print(s.value_counts().head(10))
PY


=== FILE: cicids_clean.csv ===
Columns: 83
Label column: label
Top classes:
label
1     8063823
0       38924
Name: count, dtype: int64

=== FILE: data/CICIDS2019_13f_norm_clean.csv ===
Columns: 14
Label column: Label
Top classes:
Label
1     8063823
0       38924
Name: count, dtype: int64

=== FILE: data/CICIDS2019_leaky_norm_clean.csv ===
Columns: 17
Label column: Label
Top classes:
Label
1     8063823
0       38924
Name: count, dtype: int64
```

Εικόνα 5.6: Έλεγχος χαρακτηριστικών και κατανομής κλάσεων (label) στα αρχεία CSV του CIC-DDoS2019

Για να διασφαλιστεί η πλήρης επαναληψιμότητα, η διαδικασία της τυχαιοποίησης κλειδώθηκε με σταθερό σπόρο τυχαιότητας (seed) ίσο με 42 μέσω της παραμέτρου του σπόρου τυχαιότητας (random_state), ώστε τα αποτελέσματα να είναι άμεσα συγκρίσιμα τόσο μεταξύ των διαφορετικών αλγορίθμων όσο και μεταξύ των διαφορετικών παραλλαγών της προεπεξεργασίας, όπως για παράδειγμα το σύνολο χωρίς διαρροή έναντι του συνόλου με διαρροή χαρακτηριστικών τα οποία και αναλύονται στις επόμενες υποενότητες. Η υλοποίηση του διαχωρισμού μαζί με τη δήλωση των παραμέτρων του μεγέθους συνόλου ελέγχου (test_size), του σπόρου τυχαιότητας (random_state) και της στρωματοποίησης (stratify) αποτυπώνεται στο σενάριο εκπαίδευσης (script) που φαίνεται στην Εικόνα 5.7.



```
┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ nl -ba train_rf_xgb.py | sed -n '1,220p'

     1  #!/usr/bin/env python3
     2  import argparse, json
     3  from pathlib import Path
     4  import numpy as np
     5  import pandas as pd
     6
     7  from sklearn.model_selection import train_test_split
     8  from sklearn.metrics import classification_report, f1_score
     9  from sklearn.ensemble import RandomForestClassifier
    10
    11  # RF → ONNX (skl2onnx)
    12  from skl2onnx import convert_sklearn
    13  from skl2onnx.common.data_types import FloatTensorType
    14
    15  # XGB → ONNX (onnxmltools)
    16  from xgboost import XGBClassifier
    17  from onnxmltools import convert_xgboost
    18  from onnxmltools.convert.common.data_types import FloatTensorType as OBFloatTensorType
    19
    20  OUT_DIR = Path("models")
    21  OUT_DIR.mkdir(exist_ok=True)
    22
    23  def load_df_safely(path: str, use_sample=False, per_class_cap=None):
    24      # Το csv μας είναι ήδη καθαρισμένο (prepare_cicids_chunked.py)
    25      # Στήλη label: 0/1
    26      df = pd.read_csv(path)
    27      if use_sample:
    28          df = df.sample(n=min(len(df), 200_000), random_state=42)
    29      # Optional: cap ανά κλάση για ισορροπία
    30      if per_class_cap is not None:
    31          df0 = df[df["label"] == 0].sample(n=min(per_class_cap, (df["label"] == 0).sum()), random_state=42)
    32          df1 = df[df["label"] == 1].sample(n=min(per_class_cap, (df["label"] == 1).sum()), random_state=42)
    33          df = pd.concat([df0, df1], ignore_index=True).sample(frac=1.0, random_state=42)
    34      return df
    35
    36  def to_onnx_sklearn(model, n_features, out_path):
    37      initial_type = [("input", FloatTensorType([None, n_features]))]
    38      onx = convert_sklearn(model, initial_types=initial_type, target_opset=13)
    39      with open(out_path, "wb") as f:
    40          f.write(onx.SerializeToString())
    41
    42  def main():
    43      ap = argparse.ArgumentParser()
    44      ap.add_argument("--input", default="cicids_clean.csv")
    45      ap.add_argument("--per-class-cap", type=int, default=150000)
    46      ap.add_argument("--test-size", type=float, default=0.2)
    47      args = ap.parse_args()
    48
    49      DATA_CSV = args.input
    50      df = load_df_safely(DATA_CSV, use_sample=False, per_class_cap=args.per_class_cap)
    51
    54
    55      Xtr, Xte, ytr, yte = train_test_split(X, y, test_size=args.test_size, random_state=42, stratify=y)
    56
    57      # ──────────── Random Forest ────────────
    58      rf = RandomForestClassifier(
    59          n_estimators=300,
    60          max_depth=None,
    61          n_jobs=-1,
    62          random_state=42,
    63          class_weight="balanced_subsample"
    64      )
    65      rf.fit(Xtr, ytr)
    66      pr = rf.predict(Xte)
    67      print("\n[RF] report:\n", classification_report(yte, pr, digits=4))
    68      print("[RF] F1:", f1_score(yte, pr))
    69
    70      rf_onnx = OUT_DIR / "rf_cicids.onnx"
    71      to_onnx_sklearn(rf, X.shape[1], str(rf_onnx))
    72      print("[RF] Saved ONNX →", rf_onnx)
    73
    74      # ──────────── XGBoost ────────────
    75      XGB_PARAMS = dict(
    76          n_estimators=400,
    77          max_depth=6,
    78          learning_rate=0.1,
    79          subsample=0.9,
    80          colsample_bytree=0.9,
    81          reg_lambda=1.0,
    82          tree_method="hist",
    83          n_jobs=-1,
    84          random_state=42
    85      )
    86      xgb = XGBClassifier(**XGB_PARAMS)
    87      xgb.fit(Xtr, ytr, eval_set=[(Xte, yte)], verbose=False)
    88      px = (xgb.predict_proba(Xte)[:, 1] ≥ 0.5).astype(int)
    89      print("\n[XGB] report:\n", classification_report(yte, px, digits=4))
    90      print("[XGB] F1:", f1_score(yte, px))
    91
    92      xgb_onnx = OUT_DIR / "xgb_cicids.onnx"
    93      initial_types = [("input", OBFloatTensorType([None, X.shape[1]]))]
    94      xgb_onnx_model = convert_xgboost(xgb, initial_types=initial_types, target_opset=13)
    95      with open(xgb_onnx, "wb") as f:
    96          f.write(xgb_onnx_model.SerializeToString())
    97      print("[XGB] Saved ONNX →", xgb_onnx)
    98
    99      # ──────────── Feature Schema (για predictor) ──
   100      schema = {"feature_names": list(X.columns), "dtype": "float32"}
   101      (OUT_DIR / "feature_schema.json").write_text(json.dumps(schema, indent=2))
   102      print("[OK] Saved feature_schema.json")
   103      # Bonus: ονόματα features σε txt
   104      (OUT_DIR / "models_feature_names.txt").write_text("\n".join(schema["feature_names"]))
   105      print("[OK] Saved models_feature_names.txt")
   106
   107  if __name__ == "__main__":
   108      main()
```

Εικόνα 5.7: Script εκπαίδευσης Random Forest και XGBoost και εξαγωγής μοντέλων σε ONNX

Τέλος, η αξιολόγηση των μοντέλων πραγματοποιείται στο σύνολο ελέγχου (hold-out test set) με χρήση των μετρικών ταξινόμησης (classification metrics), με έμφαση στις μετρικές που είναι πιο κατάλληλες για ανισόρροπα δεδομένα (imbalanced data), όπως για παράδειγμα ο δείκτης F1. Με αυτό το τρόπο, αποτυπώνεται πιο σωστά η ικανότητα ανίχνευσης της μειοψηφικής κλάσης (minority class) και αποφεύγοντα τα λάθος συμπεράσματα που μπορεί να προκύψουν από την αποκλειστική χρήση της ακρίβειας (accuracy). Με βάση αυτό το ενιαίο πλαίσιο του διαχωρισμού και της αξιολόγησης, οι επόμενες ενότητες παρουσιάζουν ξεχωριστά την εκπαίδευση σε δεδομένα χωρίς διαρροή και με διαρροή αλλά και την εξαγωγή των τελικών μοντέλων σε μορφή ONNX.

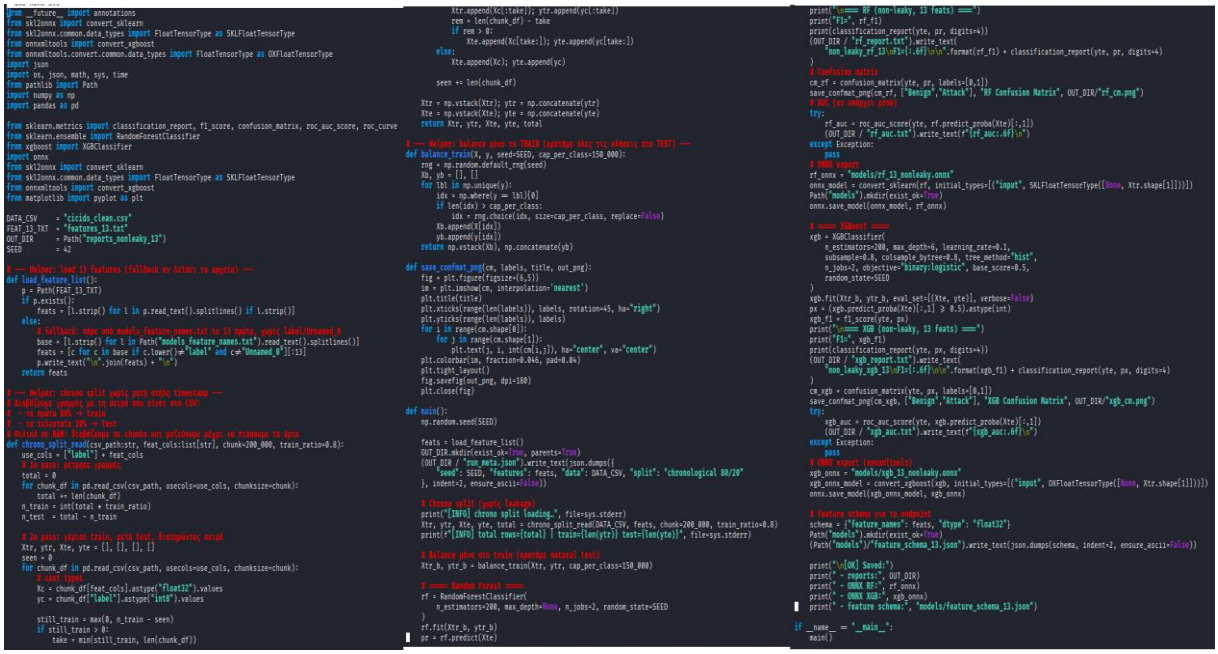
5.3.2 Εκπαίδευση των χαρακτηριστικών χωρίς διαρροή (non-leaky models)

Στα δεδομένα χωρίς διαρροή, η εκπαίδευση πραγματοποιήθηκε σε ένα σύνολο δεδομένων όπου έχουν αφαιρεθεί ή αποφευχθεί κάποια χαρακτηριστικά που θα μπορούσαν να προκαλέσουν διαρροή πληροφορίας και να οδηγήσουν λανθασμένα σε πολύ υψηλή απόδοση. Η υλοποίηση βασίστηκε στο σενάριο εκπαίδευσης train_non_leaky_13.py, με είσοδο το cicids_clean.csv και με περιορισμό του διανύσματος εισόδου στα δεκατρία επιλεγμένα χαρακτηριστικά. Τα χαρακτηριστικά αυτά φορτώνονται από ένα αρχείο λίστας για να παραμένει σταθερό το σύνολο των χαρακτηριστικών (feature set) μεταξύ της εκπαίδευσης και της φάσης πρόβλεψης ή του συμπερασμού (inference).

Ο διαχωρισμός της εκπαίδευσης και του ελέγχου (train and test split) υλοποιήθηκε ως χρονολογική διάσπαση 80/20 χωρίς να είναι απαραίτητη η ξεχωριστή στήλη της χρονικής σήμανσης (timestamp), καθώς τα δεδομένα διαβάζονται με τη σειρά που εμφανίζονται στο αρχείο και το πρώτο 80% χρησιμοποιείται για εκπαίδευση ενώ το τελευταίο 20% για έλεγχο. Έτσι, διατηρείται η χρονική ακολουθία και προσεγγίζεται πιο ρεαλιστικά ένα σενάριο συνεχόμενης ροής των γεγονότων σε ένα περιβάλλον Παρακολούθησης της Ασφάλειας (CM). Παράλληλα, λόγω της έντονης ανισορροπίας των κλάσεων εφαρμόστηκε η εξισορρόπηση μόνο στο σύνολο εκπαίδευσης μέσω ενός περιορισμού ανά

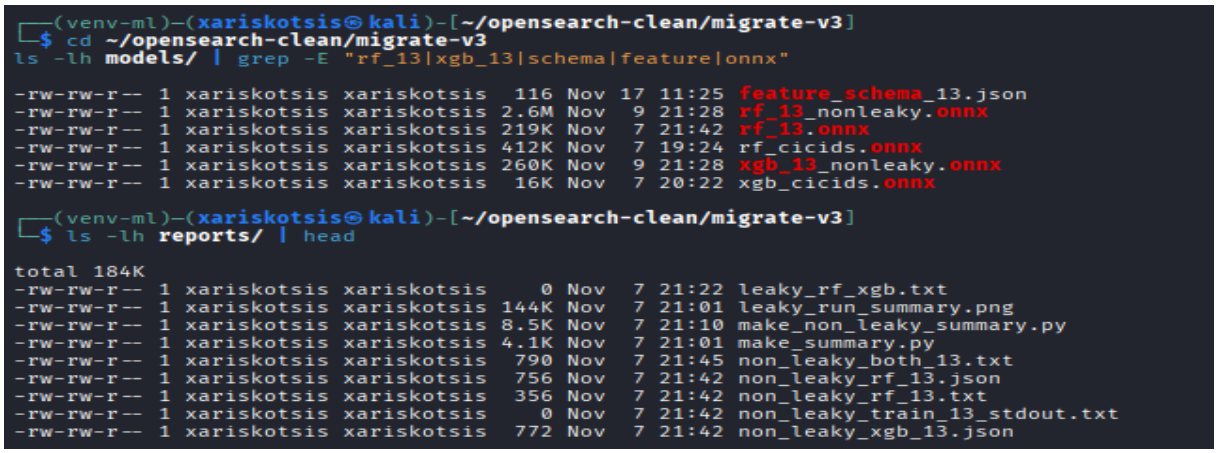
κλάση (cap per class), ενώ το σύνολο ελέγχου διατηρεί τη κανονική του κατανομή ώστε η αξιολόγηση να είναι ίδια με τις επιχειρησιακές συνθήκες.

Στο ίδιο πλαίσιο δεδομένων χωρίς διαρροή εκπαιδεύτηκαν δύο μοντέλα ταξινόμησης, Random Forest και XGBoost και υπολογίστηκαν οι βασικές μετρικές απόδοσης, όπως ο δείκτης F1 και η αναφορά ταξινόμησης αλλά και τα πρόσθετα τεκμήρια αξιολόγησης, όπως οι πίνακες σύγχυσης και η καμπύλη ROC-AUC, όπου ήταν εφικτό. Επιπλέον, στο τέλος της διαδικασίας πραγματοποιήθηκε εξαγωγή των μοντέλων σε μορφή ONNX και δημιουργήθηκε το σχήμα των χαρακτηριστικών (feature schema), το οποίο περιλαμβάνει τύπους δεδομένων, σειρά και τα ονόματα των χαρακτηριστικών για να διασφαλιστεί η συμβατότητα με τη φάση του συμπερασμού (inference) μέσω του προγνωστικού εξυπηρετητή REST. Η συνολική ροή του πειράματος δηλαδή ο χρονολογικός διαχωρισμός (chrono split) 80/20, η εξισορρόπηση μόνο στο σύνολο εκπαίδευσης, η εκπαίδευση των ταξινομητών Random Forest και XGBoost, ο υπολογισμός των μετρικών και η εξαγωγή αποτυπώνονται όλα μαζί στην Εικόνα 5.8.



Εικόνα 5.8: Κύρια ροή εκπαίδευσης σε non-leaky ρύθμιση με χρονολογικό split 13 χαρακτηριστικών

Ενώ, στο επίπεδο των αποτελεσμάτων και των παραγόμενων τεχνουργημάτων (artifacts) τα τελικά αρχεία που αξιοποιούνται αργότερα και από τον αγωγό του συστήματος Παρακολούθησης της Ασφάλειας (CM pipeline) είναι τα μοντέλα .onnx, το σχήμα των χαρακτηριστικών και οι αναφορές της αξιολόγησης στους φακέλους models/ και reports/, όλα αυτά τεκμηριώνονται στην Εικόνα 5.9.



Εικόνα 5.9: Παραγόμενα artefacts εκπαίδευσης για τα non-leaky πειράματα

5.3.3 Εκπαίδευση των χαρακτηριστικών με διαρροή (leaky models)

Για λόγους σύγκρισης και για να φανεί η επίδραση της διαρροής πληροφορίας πραγματοποιήθηκε ξεχωριστή εκπαίδευση και αξιολόγηση των μοντέλων πάνω σε ένα σύνολο δεδομένων με διαρροή (leaky dataset). Σε αυτό το σύνολο, υπάρχουν χαρακτηριστικά που συσχετίζονται άμεσα ή έμμεσα με τη μεταβλητή-στόχο (target variable) και αυτό έχει σαν αποτέλεσμα αυτά τα χαρακτηριστικά σε κάποιες περιπτώσεις να οδηγήσουν σε μη ρεαλιστικές υψηλές αποδόσεις. Η διαδικασία υλοποιήθηκε μέσω του σεναρίου αξιολόγησης (eval_leaky_models.py) το οποίο φορτώνει το αρχείο data/CICIDS2019_leaky_norm_clean.csv, εκτελεί διαχωρισμό εκπαίδευσης και ελέγχου (train and split set) και παράγει μετρικές για τον RandomForest και τον XGBoost. Όπως αποδεικνύεται από την εκτέλεση του σεναρίου, το συγκεκριμένο σύνολο δεδομένων διαβάζεται με περιορισμό πλήθους (nrows=200.000), σχηματίζεται πίνακας εισόδου δεκαέξι χαρακτηριστικών και ο διαχωρισμός οδηγεί σε train size ίσο με 140.000 και test size ίσο με 60.000, τα οποία φαίνονται ξεκάθαρα και στην Εικόνα 5.10. Στο ίδιο στιγμιότυπο καταγράφεται ότι και τα δύο μοντέλα που εξετάστηκαν εμφανίζουν τέλεια επίδοση, με ακρίβεια (accuracy), ακρίβεια θετικών προβλέψεων (precision), ανάκληση (recall) και δείκτη F1 (F1-score) ίσους με ένα καθώς και πίνακες σύγχυσης (confusion matrices) χωρίς σφάλματα. Η συμπεριφορά αυτή αποτελεί τυπική ένδειξη ύπαρξης διαρροής, διότι η απόδοση δεν αντανακλά τη ρεαλιστική ικανότητα γενίκευσης αλλά ένα τεχνητά εύκολο πρόβλημα λόγω των χαρακτηριστικών που ουσιαστικά προδίδουν την κλάση.

```
┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ cd ~/opensearch-clean/migrate-v3
source venv-ml/bin/activate
python eval_leaky_models.py | tee leaky_metrics_run.txt

Διαβάζω leaky CSV: data/CICIDS2019_leaky_norm_clean.csv (nrows=200000)
Φόρτωσα 200000 γραμμές
Στήλες: ['Flow Duration', 'Tot Fwd Pkts', 'Tot Bwd Pkts', 'TotLen Fwd Pkts', 'TotLen Bwd Pkts',
 'Flow Byts/s', 'Flow Pkts/s', 'Fwd IAT Mean', 'Bwd IAT Mean', 'Fwd Pkts/s', 'Bwd Pkts/s', 'Sub
flow Fwd Pkts', 'Subflow Bwd Pkts', 'Dest Port', 'Src Port', 'Protocol', 'Label']
Χρησιμοποιώ label column: Label
X shape: (200000, 16), y shape: (200000,)
Train size: 140000, Test size: 60000

[RF] Εκπαίδευση leaky Random Forest...

=== Random Forest (LEAKY) metrics (class=attack=1) ===
tp: 59951
fp: 0
tn: 49
fn: 0
accuracy: 1.0
precision: 1.0
recall: 1.0
f1: 1.0
support: 60000

Confusion matrix [ [tn, fp], [fn, tp] ]:
[[   49     0]
 [    0 59951]]

Classification report:
              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000        49
           1     1.0000    1.0000    1.0000     59951

    accuracy                         1.0000     60000
   macro avg     1.0000    1.0000    1.0000     60000
weighted avg     1.0000    1.0000    1.0000     60000
```

```
[XGB] Εκπαίδευση leaky XGBoost...

=== XGBoost (LEAKY) metrics (class=attack=1) ===
tp: 59951
fp: 0
tn: 49
fn: 0
accuracy: 1.0
precision: 1.0
recall: 1.0
f1: 1.0
support: 60000

Confusion matrix [ [tn, fp], [fn, tp] ]:
[[   49     0]
 [    0 59951]]

Classification report:
              precision    recall  f1-score   support

           0     1.0000    1.0000    1.0000        49
           1     1.0000    1.0000    1.0000     59951

    accuracy                         1.0000     60000
   macro avg     1.0000    1.0000    1.0000     60000
weighted avg     1.0000    1.0000    1.0000     60000
```

Εικόνα 5.10: Αποτελέσματα αξιολόγησης leaky μοντέλων, RF και XGB

Παράλληλα, η παραγωγή των σχετικών αναφορών καταγράφεται στο φάκελο reports/ όπου εμφανίζονται τα αρχεία που αντιστοιχούν στα πειράματα διαρροής τα οποία παρουσιάζονται στην Εικόνα 5.11, επιβεβαιώνοντας την ολοκλήρωση της ροής αξιολόγησης και την αποθήκευση των αποτελεσμάτων. Συνολικά, το σενάριο διαρροής (leaky) χρησιμοποιείται αποκλειστικά ως αντιπαράδειγμα ώστε να τεκμηριωθεί εμπειρικά ότι όταν υπάρχει διαρροή πληροφορίας μπορούν να δημιουργηθούν παραπλανητικές μετρήσεις της απόδοσης, οι οποίες δεν πρέπει να ερμηνεύονται σαν πραγματική επιχειρησιακή ικανότητα ανίχνευσης των επιθέσεων.

```
┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ ls -lh reports/ | grep -i leaky
sed -n '1,200p' reports/leaky_rf_xgb.txt

-rw-rw-r-- 1 xariskotsis xariskotsis    0 Nov  7 21:22 leaky_rf_xgb.txt
-rw-rw-r-- 1 xariskotsis xariskotsis 144K Nov  7 21:01 leaky_run_summary.png
-rw-rw-r-- 1 xariskotsis xariskotsis 8.5K Nov  7 21:10 make_non_leaky_summary.py
-rw-rw-r-- 1 xariskotsis xariskotsis  790 Nov  7 21:45 non_leaky_both_13.txt
-rw-rw-r-- 1 xariskotsis xariskotsis  756 Nov  7 21:42 non_leaky_rf_13.json
-rw-rw-r-- 1 xariskotsis xariskotsis  356 Nov  7 21:42 non_leaky_rf_13.txt
-rw-rw-r-- 1 xariskotsis xariskotsis    0 Nov  7 21:42 non_leaky_train_13_stdout.txt
-rw-rw-r-- 1 xariskotsis xariskotsis  772 Nov  7 21:42 non_leaky_xgb_13.json
-rw-rw-r-- 1 xariskotsis xariskotsis  357 Nov  7 21:42 non_leaky_xgb_13.txt
```

Εικόνα 5.11: Παραγόμενα artefacts αξιολόγησης για τα leaky πειράματα

5.3.4 Εξαγωγή των αποτελεσμάτων σε μορφή ONNX

Μετά την ολοκλήρωση της εκπαίδευσης των μοντέλων Random Forest και XGBoost, υλοποιήθηκε η φάση της εξαγωγής (export) των τελικών μοντέλων σε μορφή ONNX με σκοπό τη μεταφορά τους σε κάποιο περιβάλλον πρόβλεψης και συμπερασμού (inference) ανεξάρτητο από της γλώσσα και τις βιβλιοθήκες της εκπαίδευσης. Η επιλογή του ONNX επιτρέπει να αποθηκευτεί το μοντέλο σε μία κοινή, τυποποιημένη μορφή ώστε να μπορεί να εκτελείται εύκολα μέσω του χρόνου εκτέλεσης ONNX στον απομακρυσμένο προγνωστικό εξυπηρετητή (remote predictor) του συστήματος Παρακολούθησης της

Ασφάλειας (CM) χωρίς να χρειάζεται να είναι εγκατεστημένες στο περιβάλλον της πρόβλεψης οι βιβλιοθήκες που χρησιμοποιήθηκαν στην εκπαίδευση.

Η διαδικασίας της εξαγωγής πραγματοποιήθηκε μέσα από το σενάριο της εκπαίδευσης, αμέσως μετά την εκπαίδευση και την αξιολόγηση. Για τον Random Forest, πραγματοποιήθηκε μετατροπή σε ONNX με τη συνάρτηση convert_sklearn, με ορισμό του τενσόρ εισόδου (input tensor) ως FloatTensorType και του σχήματος εισόδου (input shape) και στη συνέχεια, αποθήκευση του μοντέλου στο αρχείο rf_13.onnx. Αντίστοιχα, για τον XGBoost πραγματοποιήθηκε μετατροπή σε ONNX και το αποτέλεσμα αποθηκεύτηκε ως xgb_13.onnx. Παράλληλα, δημιουργήθηκε το αρχείο σχήματος των χαρακτηριστικών (features_schema_13.onnx) το οποίο αποτυπώνει τα ονόματα και τη σειρά των δεκατριών χαρακτηριστικών, καθώς και τον τύπο των δεδομένων (float32). Η συγκεκριμένη λογική εξαγωγής, δηλαδή η δημιουργία ONNX μοντέλων και schema αποτυπώνεται στην Εικόνα 5.12.

```
┌──(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ cd ~/opensearch-clean/migrate-v3
nl -ba train_13_rf_xgb.py | sed -n '35,90p'

    35      Xtr, Xte, ytr, yte = train_test_split(
    36          X, y, test_size=0.2, random_state=42, stratify=y
    37      )
    38
    39      # ——— RandomForest (εύκολο ONNX) ———
    40      rf = RandomForestClassifier(n_estimators=150, n_jobs=2, random_state=42)
    41      rf.fit(Xtr, ytr)
    42      pr = rf.predict(Xte)
    43      save_report("non_leaky_rf_13", yte, pr,
    44                  {"features": feats, "n_features": len(feats)})
    45
    46      # ONNX RF
    47      initial_types = [("input", FloatTensorType([None, X.shape[1]]))]
    48      rf_onnx = convert_sklearn(rf, initial_types=initial_types, target_opset=13)
    49      onnx.save_model(rf_onnx, str(OUT_DIR / "rf_13.onnx"))
    50
    51      # ——— XGBoost (light) ———
    52      xgb = XGBClassifier(
    53          n_estimators=150, max_depth=6, learning_rate=0.1,
    54          subsample=0.8, colsample_bytree=0.8, tree_method="hist",
    55          n_jobs=2, objective="binary:logistic", base_score=0.5
    56      )
    57      xgb.fit(Xtr, ytr)
    58      px = (xgb.predict_proba(Xte)[:,1] ≥ 0.5).astype("int32")
    59      save_report("non_leaky_xgb_13", yte, px,
    60                  {"features": feats, "n_features": len(feats)})
    61
    62      # XGB σε ONNX (μέσω skl2onnx—δουλεύει για πολλά tree models)
    63      xgb_onnx = convert_sklearn(xgb, initial_types=initial_types, target_opset=13)
    64      onnx.save_model(xgb_onnx, str(OUT_DIR / "xgb_13.onnx"))
    65
    66      # feature schema για inference
    67      schema = {"feature_names": feats, "dtype": "float32"}
    68      (OUT_DIR / "feature_schema_13.json").write_text(json.dumps(schema, indent=2))
    69
    70  if __name__ == "__main__":
    71      main()
```

Εικόνα 5.12: Κώδικας εξαγωγής RF και XGBoost σε ONNX και δημιουργία feature_schema_13.json για συνεπή είσοδο στο inference

Η επιτυχής παραγωγή των τελικών τεχνουργημάτων (artifacts) επιβεβαιώνεται από τα περιεχόμενα του φακέλου models/ όπου εμφανίζονται τα αρχεία ONNX των δύο μοντέλων καθώς και το feature_schema_13.json. Επιπλέον, η χρήση της επιλογής -lh στην εντολή ls τεκμηριώνει και τα μεγέθη των παραγόμενων αρχείων τα οποία λειτουργούν ως πρακτική ένδειξη ότι πρόκειται για πλήρως αποθηκευμένα (serialized) μοντέλα και όχι για ημιτελείς εξαγωγές (partial exports). Η επιβεβαίωση της επιτυχίας παραγωγής των αρχείων αποτυπώνεται στην Εικόνα 5.13. Με αυτό το τρόπο, ολοκληρώνεται η σύνδεση του αγωγού εκπαίδευσης με το στάδιο της ανάπτυξης και λειτουργίας του συστήματος Παρακολούθησης της Ασφάλειας καθώς τα μοντέλα διατίθενται σε τυποποιημένη μορφή, έτοιμη για φόρτωση και εκτέλεση από τον προγνωστικό εξυπηρετητή REST.

```
┌──(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ ls -lh models/ | grep -E "rf_13|xgb_13|feature_schema_13"

-rw-rw-r-- 1 xariskotsis xariskotsis  116 Nov 17 11:25 feature_schema_13.json
-rw-rw-r-- 1 xariskotsis xariskotsis 2.6M Nov  9 21:28 rf_13_nonleaky.onnx
-rw-rw-r-- 1 xariskotsis xariskotsis 219K Nov  7 21:42 rf_13.onnx
-rw-rw-r-- 1 xariskotsis xariskotsis 260K Nov  9 21:28 xgb_13_nonleaky.onnx
```

Εικόνα 5.13: Επιβεβαίωση ONNX αρχείων και feature_schema_13.json στο models/ με μεγέθη αρχείων

5.4 Ανάπτυξη προγνωστικού εξυπηρετητή REST (CM backend)

5.4.1 Διεπαφή Προγραμματισμού Εφαρμογών Flask (Flask API)

Η ανάπτυξη του προγνωστικού εξυπηρετητή REST (REST predictor) υλοποιήθηκε ως ανεξάρτητη υπηρεσία υποστήριξης (backend service), η οποία εκτελεί απομακρυσμένο συμπερασμό και πρόβλεψη (remote inference) και ενσωματώνεται λειτουργικά στον αγωγό Παρακολούθησης της Ασφάλειας ως εξωτερικό σημείο πρόβλεψης (remote inference). Ο ρόλος της υπηρεσίας είναι να δέχεται τα τυποποιημένα αιτήματα πρόβλεψης από το σύστημα του OpenSearch, μέσω των απομακρυσμένων συνδετήρων, να καταλήγει σε συμπερασμό πάνω στο μοντέλο ONNX και να επιστρέφει το αποτέλεσμα σε μορφή JSON. Με αυτό το τρόπο, διατηρείται ο σαφής διαχωρισμός των ευθυνών, διότι το περιβάλλον του OpenSearch αναλαμβάνει την εισαγωγή των δεδομένων, την αποθήκευση, τους αγωγούς επεξεργασίας και τη παρακολούθηση, ενώ ο προγνωστικός εξυπηρετητής (predictor) αναλαμβάνει αποκλειστικά την εκτέλεση του μοντέλου.

Στο επίπεδο της υλοποίησης, η υπηρεσία βασίζεται στο Flask και εκθέτει τα δύο βασικά σημεία πρόσβασης (endpoints), το /health για τον έλεγχο της διαθεσιμότητας ή της ετοιμότητας (readiness) και το /predict για πρόβλεψη μέσω των HTTP και POST. Η λογική των σημείων πρόσβασης, οι βασικές παραμετροποιήσεις όπως για παράδειγμα το μονοπάτι του μοντέλου ONNX και ο αριθμός των αναμενόμενων χαρακτηριστικών αλλά και ο τρόπος φόρτωσης του μοντέλου τεκμηριώνονται στην Εικόνα 5.14. Ειδικότερα, στο /predict η είσοδος δίνεται ως φορτίο δεδομένων JSON (JSON payload) και περιλαμβάνει υποχρεωτικά το πεδίο instances , το οποίο μετατρέπεται σε δισδιάστατο πίνακα (ndim=2) τύπου float32. Παράλληλα, εφαρμόζεται έλεγχος της συμβατότητας ως προς το πλήθος των χαρακτηριστικών (EXPECTED_FEATURES = 13) ώστε να αποφεύγονται τα σφάλματα εκτέλεσης λόγω της ασυμφωνίας του σχήματος εισόδου (schema mismatch) και να διασφαλίζεται η συνέπεια με το σχήμα των χαρακτηριστικών (feature schema) που προέκυψε από το στάδιο της εκπαίδευσης και της εξαγωγής.

```
┌──(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ nl -ba predictor_onnx.py | sed -n '1,120p'

     1  from flask import Flask, request, jsonify
     2  import onnxruntime as ort
     3  import numpy as np
     4  import os, traceback
     5
     6  # Προσαρμόζεται αν άλλαξες θέση αρχείου
     7  ONNX_PATH = "/home/xariskotsis/opensearch-clean/migrate-v3/models/rf_13_nonleaky.onnx"
     8  EXPECTED_FEATURES = 13
     9  PROVIDERS = ["CPUExecutionProvider"]
    10
    11  app = Flask(__name__)
    12
    13  def load_session(path):
    14      if not os.path.exists(path):
    15          raise FileNotFoundError(f"ONNX not found: {path}")
    16      sess = ort.InferenceSession(path, providers=PROVIDERS)
    17      in_name = sess.get_inputs()[0].name
    18      return sess, in_name
    19
    20  sess, input_name = load_session(ONNX_PATH)
    21
    22  @app.route("/health", methods=["GET"])
    23  def health():
    24      return {"ok": True, "input_name": input_name, "expected_features": EXPECTED_FEATURES}, 200
    25
    26  @app.route("/predict", methods=["POST"])
    27  def predict():
    28      try:
    29          payload = request.get_json(force=True, silent=False)
    30          if not payload or "instances" not in payload:
    31              return jsonify(error="Missing 'instances'"), 400
    32
    33          X = np.asarray(payload["instances"], dtype=np.float32)
    34          if X.ndim ≠ 2:
    35              return jsonify(error=f"instances ndim={X.ndim}, expected 2"), 400
    36          if X.shape[1] ≠ EXPECTED_FEATURES:
    37              return jsonify(error=f"instances have {X.shape[1]} features, expected {EXPECTED_FEATURES}"), 400
    38
    39          outputs = sess.run(None, {input_name: X})
    40          out_json = {}
    41          for i, out in enumerate(outputs):
    42              out_json[f"output_{i}"] = out.tolist() if hasattr(out, "tolist") else out
    43
    44          preds = out_json.get("output_0")
    45          return jsonify(predictions=preds, raw=out_json), 200
    46      except Exception as e:
    47          app.logger.exception("predict error")
    48          return jsonify(error=str(e), traceback=traceback.format_exc()), 500
    49
    50  if __name__ == "__main__":
    51      app.run(host="0.0.0.0", port=9001)
```

Εικόνα 5.14: Υλοποίηση Flask API του REST Predictor

Επιπρόσθετα, η διαχείριση των σφαλμάτων υλοποιείται με τους σαφείς κωδικούς της κατάστασης HTTP (HTTP status code), δηλαδή επιστρέφεται 400 σε περιπτώσεις ελλιπούς ή μη έγκυρης εισόδου και 500 σε απρόβλεπτες εξαιρέσεις κατά την εκτέλεση του συμπερασμού. Παράλληλα, ενεργοποιείται η καταγραφή των συμβάντων (logging) μέσω του app.logger.exception(..), ενισχύοντας με αυτό το τρόπο την ιχνηλασιμότητα (traceability) και τη δυνατότητα της διάγνωσης στα σενάρια λειτουργίας της Παρακολούθησης της Ασφάλειας, όπου απαιτείται ο γρήγορος εντοπισμός των αιτιών αστοχίας (observability). Τέλος, το σημείο πρόσβασης επιστρέφει το κύριο αποτέλεσμα του μοντέλου ως predictions (primary output tensor) χωρίς να μπορεί να ερμηνευτεί ως πιθανότητες ή κλάσεις καθώς αυτό δεν προκύπτει μόνο από το ίδιο το γράφημα ONNX. Η σημασιολογία της εξόδου (output) καθορίζεται από το μοντέλο που έχει εξαχθεί και από το αντίστοιχο συμβόλαιο συμπερασμού (inference contract).

5.4.2 Χρόνος εκτέλεσης ONNX (ONNX Runtime)

Γενικά, η εκτέλεση των μοντέλων στον προγνωστικό εξυπηρετητή REST (REST predictor) υλοποιήθηκε με τη χρήση του χρόνου εκτέλεσης ONNX, ώστε η πρόβλεψη (inference) να γίνεται απευθείας πάνω σε μια φορητή αναπαράσταση του μοντέλου ONNX και αυτό το κομμάτι να είναι ανεξάρτητο από το αρχικό πλαίσιο εκπαίδευσης (training framework). Στο πλαίσιο αυτό, η υπηρεσία φορτώνει το αντίστοιχο αρχείο ONNX κατά την εκκίνηση και δημιουργεί μία συνεδρία εκτέλεσης

(InferenceSession), το οποίο παραμένει ενεργό για όλα τα επόμενα αιτήματα της πρόβλεψης. Με αυτή την επιλογή αποφεύγεται η επαναφόρτωση του μοντέλου σε κάθε αίτημα (request) και μειώνεται ο χρόνος απόκρισης (latency).

Επίσης, η φόρτωση και η εκτέλεση του μοντέλου ακολουθούν τη ροή που υλοποιείται στον κώδικα του προγνωστικού εξυπηρετητή (predictor), όπου και καθορίζεται η διαδρομή του μοντέλου (ONNX_PATH), δημιουργείται η συνεδρία της εκτέλεσης (InferenceSession) με ort.InferenceSession και στη συνέχεια ανακτάται με δυναμικό τρόπο το όνομα της εισόδου του γραφήματος (input tensor name). Η δυναμική ανάκτηση του ονόματος της εισόδου είναι κρίσιμη για τη σωστή κλήση του sess.run(..), επειδή οι διαφορετικές εξαγωγές του ONNX μπορεί να παράγουν διαφορετικές ονομασίες εισόδου, ενώ το OpenSearch και οι συνδετήρες του αποστέλλουν τα δεδομένα χωρίς να βασίζονται σε αυτή την εσωτερική λεπτομέρεια.

Για λόγους συμβατότητας και σταθερότητας της εκτέλεσης στο περιβάλλον πειραματισμού, ο προγνωστικός εξυπηρετητής χρησιμοποιεί ως πάροχο εκτέλεσης (execution provider) τον CPUExecutionProvider. Η είσοδος μετατρέπεται σε έναν πίνακα NumPy τύπου float32, ο οποίος είναι η συνηθέστερη και η πιο ασφαλής επιλογή τύπου δεδομένων για τα ONNX μοντέλα σε πινακοποιημένα δεδομένα (tabular data), μειώνοντας με αυτό το τρόπο τη πιθανότητα των ασυμβατοτήτων τύπων κατά την εκτέλεση. Η πρόβλεψη εκτελείται με sess.run(None,{input_name: X}) επιστρέφοντας τα τενσόρ εισόδου (output tensors) του μοντέλου. Στη συνέχεια, το κύριο αποτέλεσμα μετατρέπεται σε λίστα Python και σειριοποιείται (serialized) σε JSON, ώστε να μεταφέρεται μέσω HTML και να καταναλώνεται από τα επόμενα δομικά στοιχεία του αγωγού Παρακολούθησης της Ασφάλειας.

Τέλος, η χρήση του χρόνου εκτέλεσης του ONNX στην υπηρεσία της υποστήριξης εξυπηρετεί δύο βασικούς στόχους του συστήματος Παρακολούθησης της Ασφάλειας. Πρώτον, ένα σταθερό συμβόλαιο πρόβλεψης (inference contract), δηλαδή ίδια μορφή εισόδου και εξόδου ανεξάρτητα από την υλοποίηση της εκπαίδευσης και δεύτερον, ευκολότερη αντικατάσταση ή αναβάθμιση των μοντέλων καθώς η αλλαγή περιορίζεται στο τεχνούργημα ONNX (ONNX artifact), χωρίς να απαιτείται αναδιάρθρωση της υπηρεσίας ή του συνολικού συστήματος του OpenSearch.

5.4.3 Μορφή εισόδου/εξόδου αιτημάτων (instances)

Ακόμη, η διεπαφή του προγνωστικού εξυπηρετητή REST σχεδιάστηκε για να λειτουργεί ως ένα σαφώς ορισμένο σημείο ενοποίησης (integration point) ανάμεσα στον αγωγό Παρακολούθησης της Ασφάλειας και στην υπηρεσία της απομακρυσμένης πρόβλεψης. Η είσοδος δίνεται μέσω αιτήματος τύπου HTTP POST προς το σημείο πρόσβασης (endpoint) και μεταφέρεται ως φορτίο δεδομένων JSON (JSON payload) με βασικό πεδίο το instances. Το instances αναπαριστά έναν δισδιάστατο πίνακα N x d, όπου κάθε γραμμή αντιστοιχεί σε ένα δείγμα προς αξιολόγηση, για παράδειγμα σε μία εγγραφή ροής μετά τον μετασχηματισμό ή τον εμπλουτισμό και κάθε στήλη αντιστοιχεί σε ένα αριθμητικό χαρακτηριστικό. Η επιλογή της δισδιάστατης αναπαράστασης είναι σκόπιμη διότι αφενός επιτρέπει τη μαζική αξιολόγηση (batch evaluation) πολλών παραδειγμάτων στο ίδιο αίτημα, αφετέρου ευθυγραμμίζεται με τη συνηθισμένη μορφή της εισόδου που χρησιμοποιούν τα γραφήματα ONNX για τα πινακοποιημένα μοντέλα (tabular models). Στο πειραματικό πλαίσιο, το πλήθος των χαρακτηριστικών θεωρείται σταθερό και συμφωνεί με το σχήμα των χαρακτηριστικών που δημιουργήθηκε κατά την εκπαίδευση, ώστε να διασφαλίζεται ότι η διάσταση d παραμένει συνεπής ανάμεσα στην εκπαίδευση (training) και στη πρόβλεψη (inference).

Στο επίπεδο της επικύρωσης της εισόδου (input validation), ο προγνωστικός εξυπηρετητής εφαρμόζει τους ελέγχους σχήματος, όπως για παράδειγμα ότι το φορτίο είναι έγκυρο JSON, ότι υπάρχει το πεδίο

instances, ότι τα δεδομένα μπορούν να μετατραπούν σε αριθμητικό πίνακα και ότι η είσοδος είναι αυστηρά δισδιάστατη και με σωστό πλήθος στηλών πριν εκτελέσει τη συνεδρία εκτέλεσης (inference session) του ONNX. Οι έλεγχοι αυτοί είναι απαραίτητοι σε επιχειρησιακή χρήση επειδή δέχονται τα ασαφή σφάλματα του χρόνου εκτέλεσης του ONNX και μετατρέπουν τις ασυμβατότητες σε ρητές και εύκολα αναγνωρίσιμες απορρίψεις αιτημάτων.

Η έξοδος επιστρέφει επίσης σε μορφή JSON και έχει στο πεδίο predictions την τελική πρόβλεψη ταξινόμησης, για παράδειγμα τις τιμές 0 ή 1, όπως προκύπτει από τη λογική της μεταεπεξεργασίας (post processing) του μοντέλου. Ενώ, στο πεδίο raw αποτυπώνονται στα ακατέργαστα αποτελέσματα (raw outputs) του γραφήματος ONNX, όπως για παράδειγμα τα logits και οι πιθανότητες ανά κλάση ανάλογα με το γράφημα και το τρόπο της εξαγωγής. Η συνύπαρξη της τελικής πρόβλεψης και των ακατέργαστων εξόδων βοηθάει στο να γίνονται άμεσα οι βασικοί έλεγχοι της ορθότητας και να εξετάζεται πιο αναλυτικά η συμπεριφορά του μηχανισμού της πρόβλεψης, το οποίο είναι πολύ χρήσιμο κατά τη φάση της ανάπτυξης και της αξιολόγησης. Μία ενδεικτική κλήση του curl, ένα παράδειγμα του πεδίου instances και η αντίστοιχη απόκριση του προγνωστικού εξυπηρετητή παρουσιάζονται στην Εικόνα 5.15, τεκμηριώνοντας και τη μορφή της εισόδου και της εξόδου που χρησιμοποιείται στο υποσύστημα υποστήριξης (backend) του συστήματος Παρακολούθησης της Ασφάλειας.

```
┌──(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ curl -i -s -X POST "http://127.0.0.1:9001/predict" \
  -H "Content-Type: application/json" \
  -d '{"instances":[[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0,1.1,1.2,1.3]]}'

HTTP/1.1 200 OK
Server: Werkzeug/3.1.3 Python/3.13.7
Date: Sat, 20 Dec 2025 15:56:37 GMT
Content-Type: application/json
Content-Length: 103
Connection: close

{"predictions":[0],"raw":{"output_0":[0],"output_1":[{"0":0.6950000524520874,"1":0.304999977350235}]}}
```

Εικόνα 5.15: Παράδειγμα αιτήματος POST /predict με JSON instances προς τον REST Predictor

5.4.4 Έλεγχοι λειτουργίας και καταγραφή συμβάντων (health checks & logging)

Για να ενταχθεί ο προγνωστικός εξυπηρετητής ως αξιόπιστο υποσύστημα σε ένα ολοκληρωμένο περιβάλλον Παρακολούθησης της Ασφάλειας (CM) είναι απαραίτητοι τόσο οι μηχανισμοί του ελέγχου της διαθεσιμότητας όσο και η βασική ιχνηλασιμότητα του. Στο πλαίσιο αυτό, υλοποιήθηκε το σημείο πρόσβασης, HTTP GET /health, το οποίο λειτουργεί ως ένας έλεγχος υγείας και δίνει την άμεση ένδειξη ότι η υπηρεσία είναι ενεργή και βρίσκεται σε συνεπή κατάσταση.

Η απόκριση του /health δεν περιορίζεται σε μία απλή απάντηση OK αλλά επιστρέφει δομημένες πληροφορίες σχετικές με την εκτέλεση του μοντέλου, όπως το όνομα της εισόδου (input_name) που ανακτάται από το γράφημα του ONNX αλλά και τον αναμενόμενο αριθμό των χαρακτηριστικών (expected_features). Αυτό είναι σημαντικό, διότι δείχνει ότι ο προγνωστικός εξυπηρετητής δεν είναι απλώς σε λειτουργία αλλά ότι έχει ολοκληρώσει επιτυχώς το κρίσιμο στάδιο της φόρτωσης και της αρχικοποίησης της συνεδρίας της εκτέλεσης. Στην πράξη, αυτό το στάδιο είναι και η συχνότερη πηγή προβλημάτων σε αναπτύξεις όπως η λανθασμένη διαδρομή του αρχείου, η ασυμβατότητα του γραφήματος και η ασυμφωνία των παροχών της εκτέλεσης. Έτσι, ο αγωγός του συστήματος Παρακολούθησης της Ασφάλειας και οι μηχανισμοί της ενορχήστρωσης μπορούν να πραγματοποιούν ένα προληπτικό έλεγχο πριν σταλούν τα μαζικά αιτήματα της πρόβλεψης.

Παράλληλα, η καταγραφή των συμβάντων (logging) χρησιμοποιείται ώστε να υπάρχει μία ξεκάθαρη εικόνα για τη κανονική ροή λειτουργίας αλλά και για τη διάγνωση των σφαλμάτων σε πραγματικό χρόνο. Η υπηρεσία αυτή καταγράφει τα γεγονότα της εκκίνησης καθώς και τα εισερχόμενα αιτήματα προς το /predict και το /health μαζί με τους κωδικούς της κατάστασης HTTP, παρέχοντας με αυτό το τρόπο ένα βασικό αλλά χρήσιμο ίχνος ελέγχου (audit trail) για την αλληλεπίδραση με τον προγνωστικό εξυπηρετητή. Επιπλέον, όταν προκύπτουν εξαιρέσεις όπως για παράδειγμα, το άκυρο φορτίο εισόδου, η λάθος διάσταση και το σφάλμα της πρόβλεψης, καταγράφονται οι σχετικές πληροφορίες ώστε το πρόβλημα να μπορεί να αναπαραχθεί και να συσχετιστεί με ένα συγκεκριμένο σημείο πρόσβασης και μία συγκεκριμένη χρονική στιγμη.

Τέλος, η επιτυχής κλήση του /health και το ενδεικτικό απόσπασμα από την καταγραφή της λειτουργίας της υπηρεσίας παρουσιάζονται στην Εικόνα 5.16 και έτσι επιβεβαιώνεται ότι ο προγνωστικός εξυπηρετητής διαθέτει κάποιους λειτουργικούς μηχανισμούς ελέγχου της ετοιμότητας και της βασικής παρακολούθησης, οι οποίοι είναι απαραίτητοι για τη σταθερή ενσωμάτωση στο υποσύστημα της υποστήριξης του συστήματος Παρακολούθησης της Ασφάλειας.

```
┌──(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ curl -i -s "http://127.0.0.1:9001/health"
echo "─── tail /tmp/predictor_onnx.log ───"
tail -n 30 /tmp/predictor_onnx.log

HTTP/1.1 200 OK
Server: Werkzeug/3.1.3 Python/3.13.7
Date: Sat, 20 Dec 2025 15:57:16 GMT
Content-Type: application/json
Content-Length: 56
Connection: close

{"expected_features":13,"input_name":"input","ok":true}
─── tail /tmp/predictor_onnx.log ───
 * Serving Flask app 'predictor_onnx'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server
instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:9001
 * Running on http://192.168.2.11:9001
Press CTRL+C to quit
127.0.0.1 - - [17/Dec/2025 19:18:38] "POST /predict HTTP/1.1" 200 -
172.18.0.2 - - [17/Dec/2025 19:18:44] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [20/Dec/2025 17:56:37] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [20/Dec/2025 17:57:16] "GET /health HTTP/1.1" 200 -
```

Εικόνα 5.16: Έλεγχος υγείας GET /health και ενδεικτικά logs λειτουργίας του REST Predictor

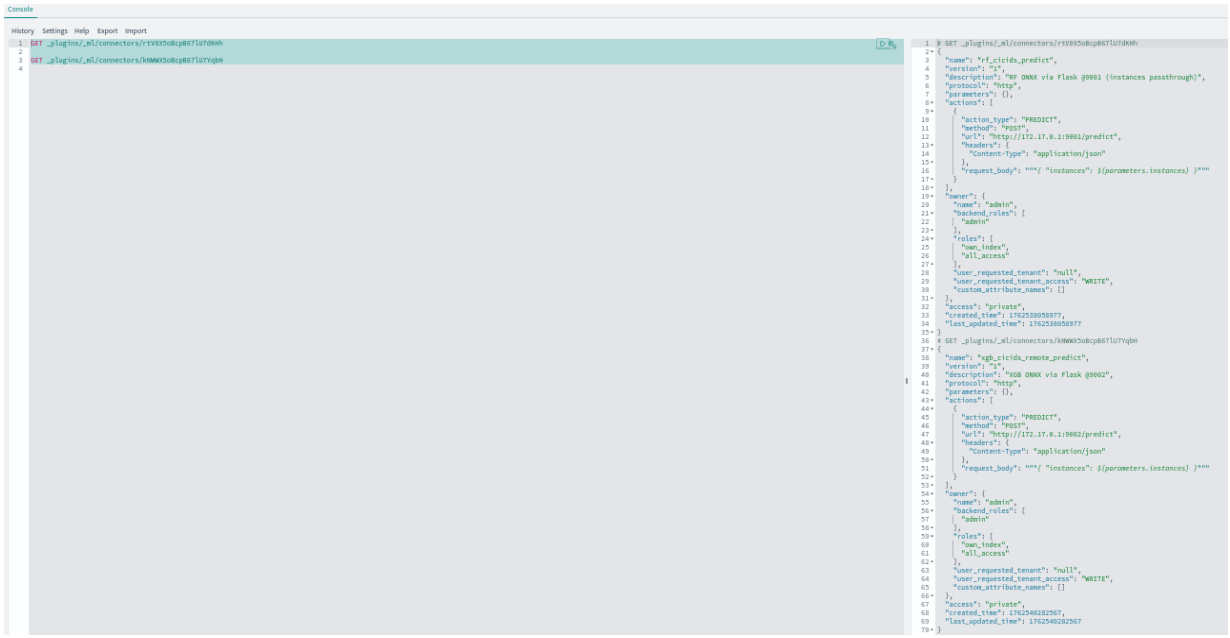
5.5 Ενσωμάτωση στη πλατφόρμα του OpenSearch και υλοποίηση του αγωγού του συστήματος Παρακολούθησης της Ασφάλειας (CM pipeline)

5.5.1 Συνδετήρες της Μηχανικής Μάθησης σε απομακρυσμένη λειτουργία (ML Connectors –REMOTE)

Για να ενσωματωθεί ο εξωτερικός προγνωστικός εξυπηρετητής REST στο πλαίσιο της Μηχανικής Μάθησης της πλατφόρμας του OpenSearch χρησιμοποιούνται οι συνδετήρες της Μηχανικής Μάθησης (ML Connectors) σε λειτουργία REMOTE. Οι συνδετήρες αυτοί λειτουργούν ως γέφυρα ανάμεσα στο OpenSearch και στην υπηρεσία της πρόβλεψης του υποσυστήματος υποστήριξης (backend inference service), η οποία υλοποιείται με χρήση του Flask και του χρόνου της εκτέλεσης ONNX (ONNX Runtime). Πρακτικά, κάθε συνδετήρας (connector) ορίζει με κατανοητό τρόπο το πρωτόκολλο επικοινωνίας (HTTP), τη μέθοδο της κλήσης (POST) και το σημείο της πρόσβασης (endpoint) στο οποίο θα αποστέλλονται τα αιτήματα της πρόβλεψης. Με αυτό το μηχανισμό, ο συμπερασμός ή πρόβλεψη (inference) εκτελείται εκτός του πυρήνα του OpenSearch, χωρίς όμως να σπάει την ενιαία λογική του οικοσυστήματος, καθώς το OpenSearch συνεχίζει να καλεί το μοντέλο με τυποποιημένο τρόπο μέσα από το πρόσθετο του ML (ML plugin). Αυτό βοηθάει πολύ στην επεκτασιμότητα (scalability) και

κυρίως επιτρέπει τη μελλοντική αντικατάσταση ή αναβάθμιση του προγνωστικού εξυπηρετητή χωρίς να απαιτούνται αλλαγές στα στάδια της εισαγωγής των δεδομένων (ingest) και της επίβλεψης (monitoring) του συστήματος Παρακολούθησης της Ασφάλειας.

Στην Εικόνα 5.17 παραθέτεται η ρύθμιση δύο συνδετήρων ενός για το μοντέλο Random Forest, όπου τα αιτήματα της πρόβλεψης δρομολογούνται προς το σημείο της πρόσβασης /predict της απομακρυσμένης θύρας 9001 και ενός για το XGBoost, όπου τα αιτήματα της πρόβλεψης δρομολογούνται προς το σημείο της πρόσβασης /predict της απομακρυσμένης θύρας 9002. Ακόμη, πολύ σημαντικό είναι το πρότυπο του σώματος αιτήματος (request_body template), το οποίο αποστέλλει τα δεδομένα εισόδου στη μορφή ({“instances” : ${parameters.instances}}). Με αυτό το τρόπο, οι τιμές των χαρακτηριστικών περνούν στο προγνωστικό εξυπηρετητή ακριβώς στη δομή που τις αναμένει το διεπαφικό πρόγραμμα των εφαρμογών (API) του Flask, χωρίς να απαιτείται πρόσθετη προσαρμογή ή ειδική σειριοποίηση σε κάθε κλήση. Τέλος, τα πεδία διαχείρισης της πρόσβασης, όπως η ιδιοκτησία (owner), οι ρόλοι (roles) και η πρόσβαση (access=private), δείχνουν ότι ο συνδετήρας (connector) αντιμετωπίζεται ως διαχειριζόμενος πόρος (managed artifact) μέσα στο OpenSearch. Αυτό είναι πολύ σημαντικό σε ένα περιβάλλον Παρακολούθησης της Ασφάλειας όπου απαιτείται η ελεγχόμενη χρήση και η ξεκάθαρη οριοθέτηση των δικαιωμάτων για τα στοιχεία τα οποία εκτελούν τις προβλέψεις.



Εικόνα 5.17: Ρύθμιση ML Connector (REMOTE) για δρομολόγηση αιτημάτων πρόβλεψης προς τον Flask predictor

5.5.2 Καταχώριση και διαχείριση απομακρυσμένων μοντέλων (REMOTE Models – RF & XGBoost)

Μετά από τη δημιουργία των συνδετήρων της Μηχανικής Μάθησης (ML connectors) στη λειτουργία της απομακρυσμένης κλήσης προς το προγνωστικό εξυπηρετητή REST, στο OpenSearch ακολούθησε η καταχώρηση (register) και η ενεργοποίηση (deploy) των αντίστοιχων απομακρυσμένων μοντέλων (REMOTE models). Με αυτό το τρόπο, η πρόβλεψη γίνεται διαθέσιμη μέσα από το πρόσθετο της Μηχανικής Μάθησης ως τυποποιημένο σημείο μοντέλου και μπορεί να αξιοποιηθεί από τα επόμενα στάδια του αγωγού Παρακολούθησης της Ασφάλειας, όπως ο μηχανισμός πρόβλεψης (inference), ο εμπλουτισμός (enrichment) και οι ειδοποιήσεις (alerting). Στο πλαίσιο της υλοποίησης,

δημιουργήθηκαν δύο ξεχωριστά μοντέλα μέσα στο OpenSearch, ένα για το Random Forest και ένα για το XGBoost. Η επιλογή αυτή έγινε για να είναι η υλοποίηση πιο καθαρή και διαχειρίσιμη. Αρχικά, επιτρέπει σε κάθε μοντέλο να διατηρεί τη δική του ρύθμιση, την έκδοση και τη κατάσταση της διάθεσης (deployment state) χωρίς να επηρεάζει το άλλο, κάτι που είναι πολύ χρήσιμο όταν γίνονται αλλαγές, ενημερώσεις ή επαναεκπαιδεύσεις. Παράλληλα, εξασφαλίζει ότι τα αιτήματα της πρόβλεψης ακολουθούν διακριτές ροές για κάθε αλγόριθμο, ώστε η σύγκριση των Random Forest και XGBoost να βασίζεται σε σαφή και συνεπή διαδικασία, χωρίς αμφιβολίες για το ποιο σημείο πρόσβασης ή ποια δρομολόγηση εξυπηρέτησε κάθε αποτέλεσμα.

Η κατάσταση της λειτουργίας και η αντιστοίχιση των απομακρυσμένων μοντέλων επαληθεύεται μέσω των αντίστοιχων κλήσεων του προγραμματιστικού περιβάλλοντος (API) και του πρόσθετου της Μηχανικής Μάθησης (ML plugin). Όπως φαίνεται στην Εικόνα 5.18, και τα δύο μοντέλα δηλώνονται ως απομακρυσμένα (REMOTE) και εμφανίζονται πως είναι σε ενεργή κατάσταση και διαθέσιμα για αιτήματα πρόβλεψης (DEPLOYED). Επιπλέον, στο πεδίο connector_id φαίνεται ποιος συνδετήρας είναι δεμένος με το μοντέλο δηλαδή ποιο κανάλι επικοινωνίας χρησιμοποιείται για να δρομολογούνται οι προβλέψεις προς τον εξωτερικό προγνωστικό εξυπηρετητή. Συγκεκριμένα, το μοντέλο Random Forest (rf_cicids_remote_predict) αντιστοιχίζεται στο συνδετήρα που εξυπηρετεί το σημείο πρόσβασης του RF, στη θύρα 9001, ενώ το μοντέλο XGBoost (xgb_cicids_remote_predict) αντιστοιχίζεται στο συνδετήρα που εξυπηρετεί το σημείο πρόσβασης του XGB, στη θύρα 9002. Με αυτό το τρόπο, το OpenSearch διατηρεί τον έλεγχο της ορχήστρωσης, δηλαδή τη καταχώριση του μοντέλου, τη παρακολούθηση της κατάστασης και τη τυποποιημένη διαπαφή της κλήσης, ενώ η πραγματική εκτέλεση της πρόβλεψης πραγματοποιείται εκτός του πυρήνα του OpenSeach στον εξωτερικό προγνωστικό εξυπηρετητή.

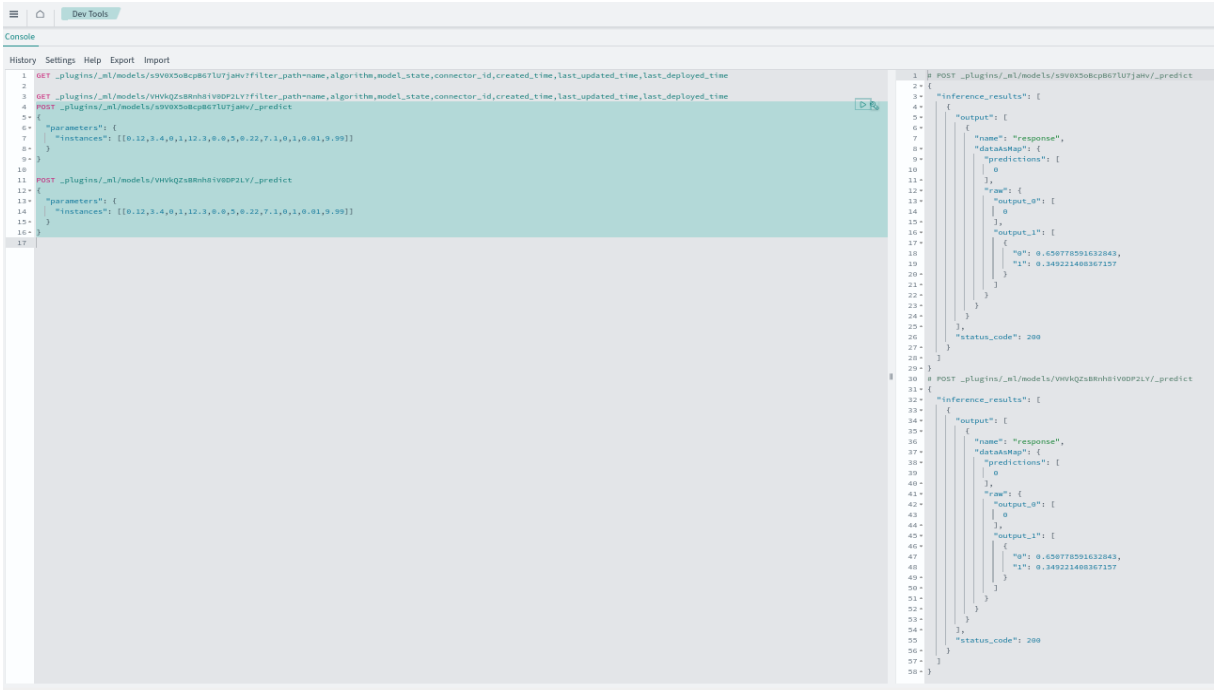
Τέλος, η επιλογή των δύο ξεχωριστών συνδετήρων και των δύο ξεχωριστών απομακρυσμένων μοντέλων δεν έγινε μόνο για τυπικούς λόγους αλλά επειδή ταιριάζει με το τρόπο που δουλεύει στη πράξη ένα σύστημα Παρακολούθησης της Ασφάλειας. Κάθε αλγόριθμος αντιμετωπίζεται ως ανεξάρτητο υποσύστημα, διότι έχει τη δική του διαδρομή κλήσης, το δικό του σημείο πρόσβασης και τη δική του κατάσταση λειτουργίας. Αυτό κάνει την υλοποίηση πιο ομαλή στη καθημερινή χρήση, διότι μπορεί να γίνει συντήρηση ή αντικατάσταση του ενός μοντέλου χωρίς να επηρεάζεται το άλλο, ενώ αν χρειαστεί περισσότερη υπολογιστική ισχύ μπορεί να ενισχυθεί (scale) μόνο ο προγνωστικός εξυπηρετητής που το απαιτεί. Ταυτόχρονα, ο διαχωρισμός αυτός αποφεύγει τη σύγχυση στη δρομολόγηση και κρατά τη σύγκριση του Random Forest και του XGBoost σαφή και επαναλήψιμη στα επόμενα στάδια της πειραματικής αξιολόγησης.

Εικόνα 5.18: Κατάσταση και μεταδεδομένα των REMOTE μοντέλων RF και XGB στο OpenSearch ML plugin

5.5.3 Εκτέλεση πρόβλεψης μέσω του API _ml/models/(id)/_predict (Model Inference API)

Επιπλέον, η επιχειρησιακή αξιοποίηση των εκπαιδευόμενων μοντέλων μέσα στον αγωγό Παρακολούθησης της Ασφάλειας έγινε μέσω του Model Inference API του πρόσθετου της Μηχανικής Μάθησης του OpenSearch, χρησιμοποιώντας τη κλήση _predict πάνω στο αντίστοιχο σημείο του μοντέλου. Σε αυτό το στάδιο, το σύστημα του OpenSearch λειτουργεί ως ένα ενιαίο σημείο πρόβλεψης (inference interface) στο οποίο ο χρήστης στέλνει το αίτημα της πρόβλεψης προς το μοντέλο και λαμβάνει το αποτέλεσμα της ταξινόμησης χωρίς να χρειάζεται να επικοινωνήσει απευθείας με τον εξωτερικό προγνωστικό εξυπηρετητή Flask. Αυτή η προσέγγιση είναι σημαντική για τη συνοχή του συστήματος Παρακολούθησης της Ασφάλειας, ενώ παραμένει εφικτή η εύκολη επέκταση και η σταθερή αναπαραγωγή των πειραμάτων.

Στην Εικόνα 5.19 παρουσιάζεται ενδεικτικά η εκτέλεση της πρόβλεψης μέσα από το Dev Tools του Opensearch. Εκεί πραγματοποιούνται κλήσεις τύπου POST προς _plugins/_ml/models/{id}/_predict για δύο διαφορετικά αναγνωριστικά μοντέλων (model IDs). Το αίτημα μεταφέρει τις τιμές εισόδου στο πεδίο parameters.instances, δηλαδή ως ένα πίνακα τιμών των χαρακτηριστικών, στην ίδια μορφή που έχει ακολουθηθεί και από τον προγνωστικό εξυπηρετητή REST. Η απόκριση επιστρέφεται στο πεδίο inference_results και περιλαμβάνει τη τελική πρόβλεψη (predictions) καθώς και επιμέρους ακατέργαστες εξόδους (raw outpouts). Αυτές οι έξοδοι αντιστοιχούν στα αποτελέσματα του γραφήματος ONNX και ανάλογα με το πώς έγινε η εξαγωγή του μοντέλου μπορούν να εκφράζουν βαθμολογίες, τιμές ενεργοποίησης ή πιθανότητες. Τέλος, η επιτυχής ολοκλήρωση της κλήσης επιβεβαιώνεται από το κωδικό status_code: 200, ο οποίος τεκμηριώνει ότι το OpenSearch επικοινώνησε σωστά με το απομακρυσμένο μοντέλο και έλαβε έγκυρο αποτέλεσμα πρόβλεψης.



Εικόνα 5.19: Εκτέλεση πρόβλεψης (inference) μέσω OpenSearch ML plugin σε REMOTE μοντέλα, με επιστροφή αποτελεσμάτων απόκρισης

5.5.4 Έλεγχος πρόσβασης και βασικές ρυθμίσεις ασφάλειας (Authentication / Authorization, TLS, Roles)

Γενικότερα, η ασφάλεια πρόσβασης στο περιβάλλον OpenSearch του συστήματος Παρακολούθησης της Κυβερνοασφάλειας βασίστηκε στον μηχανισμό OpenSearch Security, ο οποίος εφαρμόζει τον έλεγχο της ταυτότητας (authentication) και της εξουσιοδότησης (authorization) στο επίπεδο της διεπαφής του προγραμματισμού των εφαρμογών (API). Στη πράξη, αυτό σημαίνει ότι το σύστημα δεν επιτρέπει τη πρόσβαση στις λειτουργίες του χωρίς να υπάρχουν τα έγκυρα στοιχεία της σύνδεσης. Η συμπεριφορά αυτή αποδεικνύεται στην Εικόνα 5.20 από την απάντηση HTTP 401 (Unauthorized), όταν γίνεται μια προσπάθεια πρόσβασης στο βασικό σημείο πρόσβασης (endpoint) του cluster χωρίς διαπιστευτήρια, ενώ στο ίδιο μήνυμα εμφανίζεται και η ένδειξη Basic realm= “OpenSearch Security”, που επιβεβαιώνει ότι είναι ενεργοποιημένη η βασική μέθοδος του ελέγχου της πρόσβασης. Η συγκεκριμένη προστασία, είναι πολύ σημαντική σε ένα σύστημα Παρακολούθησης της Ασφάλειας επειδή αποτρέπει τις μη εξουσιοδοτημένες ενέργειες στα κρίσιμα τμήματα του συστήματος, όπως η εισαγωγή των δεδομένων (ingest), η πρόβλεψη μέσω των μοντέλων της Μηχανικής Μάθησης (ML inference) και η διαχείριση των μηχανισμών της ειδοποίησης (alerting), οι οποίες μπορούν να επηρεάσουν τόσο την ακεραιότητα των δεδομένων όσο και τη συνολική αξιοπιστία του αγωγού.

```
┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ curl -k -i https://localhost:9200/ | head -n 12

  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100    12  100    12    0     0   1577      0 --:--:-- --:--:-- --:--:--  1714
HTTP/2 401
www-authenticate: Basic realm="OpenSearch Security"
x-opensearch-version: OpenSearch/3.3.1 (opensearch)
content-type: text/plain; charset=UTF-8
content-length: 12

Unauthorized

┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ curl -k -i -u admin https://localhost:9200/ | head -n 20

Enter host password for user 'admin':
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   557  100   557    0     0  48739      0 --:--:-- --:--:-- --:--:-- 50636
HTTP/2 200
x-opensearch-version: OpenSearch/3.3.1 (opensearch)
content-type: application/json; charset=UTF-8
content-length: 557

{
  "name" : "node-1",
  "cluster_name" : "os-cluster",
  "cluster_uuid" : "uHpjzDLEQkqeQln5wq8QEQ",
  "version" : {
    "distribution" : "opensearch",
    "number" : "3.3.1",
    "build_type" : "tar",
    "build_hash" : "d90ecec16cb1049b762ed7c94777f42fb97b1eea",
    "build_date" : "2025-10-18T02:18:04.135009377Z",
    "build_snapshot" : false,
    "lucene_version" : "10.3.1",
    "minimum_wire_compatibility_version" : "2.19.0",
    "minimum_index_compatibility_version" : "2.0.0"
  },

┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ curl -k -i -u admin https://localhost:9200/_plugins/_security/api/account | head -n 80

Enter host password for user 'admin':
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100   266  100   266    0     0  15891      0 --:--:-- --:--:-- --:--:-- 16625
HTTP/2 200
x-opensearch-version: OpenSearch/3.3.1 (opensearch)
content-type: application/json; charset=UTF-8
content-length: 266

{"user_name":"admin","is_reserved":true,"is_hidden":false,"is_internal_user":true,"user_requeste
d_tenant":null,"backend_roles":["admin"],"custom_attribute_names":[],"tenants":{"global_tenant":
true,"admin_tenant":true,"admin":true},"roles":["own_index","all_access"]}
```

Εικόνα 5.20: Έλεγχος πρόσβασης στο OpenSearch μέσω HTTPS/TLS και Basic Authentication

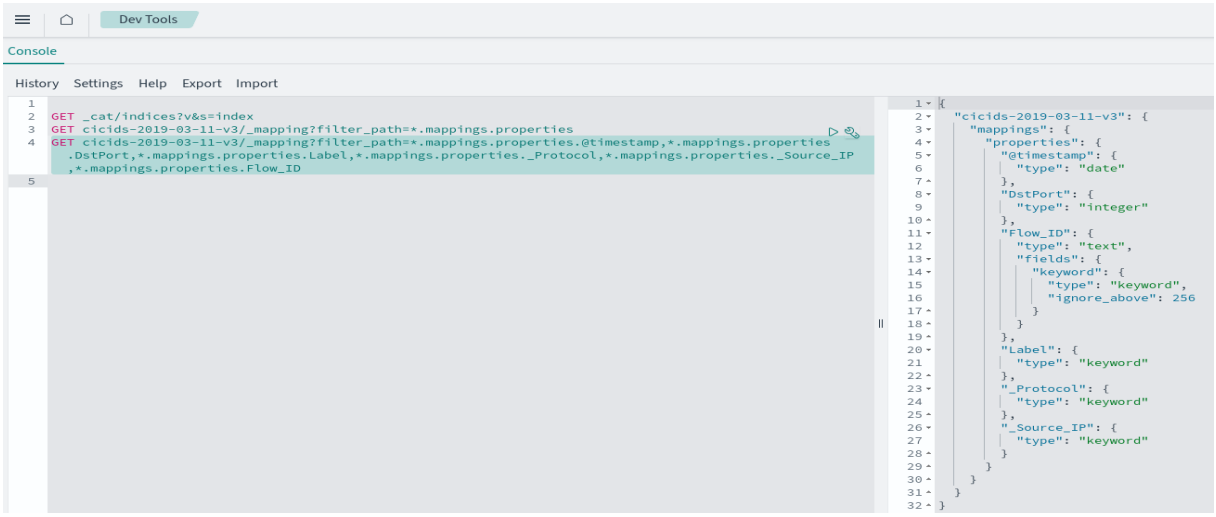
Την ίδια στιγμή η επικοινωνία με το cluster πραγματοποιείται μέσω του HTTPS με κρυπτογράφηση μεταφοράς (Transport Layer Security). Στο ίδιο πλαίσιο, στην Εικόνα 5.20 φαίνεται η χρήση της παραμέτρου curl –k, η οποία συνηθίζεται στις εργαστηριακές εγκαταστάσεις όταν το σύστημα χρησιμοποιεί αυτο-υπογεγραμμένο πιστοποιητικό (self-signed certificate). Με άλλα λόγια, ο πελάτης (client) παρακάμπτει τον έλεγχο της εμπιστοσύνης του πιστοποιητικού για σκοπούς δοκιμών αλλά η σύνδεση εξακολουθεί να πραγματοποιείται μέσω του HTTPS, άρα η μεταφορά των δεδομένων και των διαπιστευτηρίων γίνεται σε κρυπτογραφημένο κανάλι. Ακόμη, μετά από τον επιτυχή έλεγχο της ταυτότητας (authentication) σαν χρήστης admin, επιβεβαιώνεται και η εξουσιοδότηση (authorization) μέσω του Security API, όπως φάινεται και στην Εικόνα 5.20. Εκεί γίνεται η ανάκτηση των στοιχείων του λογαριασμού και εμφανίζονται οι ρόλοι της υποστήριξης και οι ενεργοί ρόλοι. Η συγκεκριμένη Εικόνα δείχνει πρακτικά ότι οι δυνατότητες του χρήστη δεν είναι ανοιχτές αλλά καθορίζονται από ρόλους, δηλαδή εφαρμόζεται ένας έλεγχος πρόσβασης με βάση του ρόλους (role-based access control – RBAC). Αυτό είναι σημαντικό σε ένα περιβάλλον Παρακολούθησης της Ασφάλειας, διότι επιτρέπει τον καθορισμό των επιτρεπόμενων ενεργειών ανά χρήστη ή ρόλο, για παράδειγμα τη διαχείριση των δεικτών, την εκτέλεση των προβλέψεων και τη διαμόρφωση των μηχανισμών της ειδοποίησης, εξασφαλίζοντας ότι οι κρίσιμες λειτουργίες του συστήματος είναι προσβάσιμες μόνο από τους χρήστες που έχουν την άδεια.

5.6 Αγωγοί εισαγωγής και δείκτες (Ingest Pipelines & Indices)

5.6.1 Σχεδίαση χαρτογραφήσεων και τύπων πεδίων (Schema / Mappings)

Αρχικά, η σχεδίαση του σχήματος (schema) και η σωστή χαρτογράφηση των πεδίων (mappings) είναι βασικό βήμα για ένα σύστημα Παρακολούθησης της Ασφάλειας, γιατί από αυτό εξαρτάται τόσο το πως αντιλαμβάνεται το σύστημα τα δεδομένα, όσο και το πόσο εύκολα μπορούν να αναζητηθούν, να φιλτραριστούν και να αναλυθούν. Στη συγκεκριμένη υλοποίηση, οι δείκτες του OpenSearch ορίστηκαν με σαφείς τύπους πεδίων, ώστε τα αριθμητικά χαρακτηριστικά, τα χρονικά στοιχεία και οι κατηγορικές μεταβλητές να αποθηκεύονται σε κατάλληλη μορφή και να υποστηρίζουν αξιόπιστα τόσο τις συναθροίσεις (aggregations) όσο και τα φίλτρα (filters) στο Dashboard και στους μηχανισμούς της ειδοποίησης.

Ενδεικτικά, στην Εικόνα 5.21 παρουσιάζεται η σωστή χαρτογράφηση των πεδίων του δείκτη cicids-2019-03-11-v3. Εκεί, το πεδίο timestamp έχει οριστεί ως ημερομηνία, κάτι που επιτρέπει τα χρονικά ερωτήματα και τις οπτικοποιήσεις με άξονα το χρόνο. Αντίστοιχα, πεδία όπως το DstPort ορίζονται ως ακέραιοι αριθμοί ώστε να είναι δυνατές οι αριθμητικές συγκρίσεις και οι στατιστικές συναθροίσεις. Οι κατηγορικές μεταβλητές όπως τα Label, Protocol και Source_IP, χαρτογραφούνται ως λέξεις κλειδιά, η επιλογή αυτή βοηθάει στις ακριβείς αντιστοιχίσεις (exact match), στις ομαδοποιήσεις (group-by) και στο φιλτράρισμα χωρίς να χρειάζεται η επεξεργασία του κειμένου (text analysis). Τέλος, το Flow_ID εμφανίζεται ως κείμενο με υποπεδίο τη λέξη κλειδί, ώστε όταν χρειάζεται να υποστηρίζονται και οι αναζητήσεις του τύπου κειμένου αλλά και η ακριβής ταυτοποίηση και συσχέτιση των ροών μέσω της λέξης κλειδί του υποπεδίου.



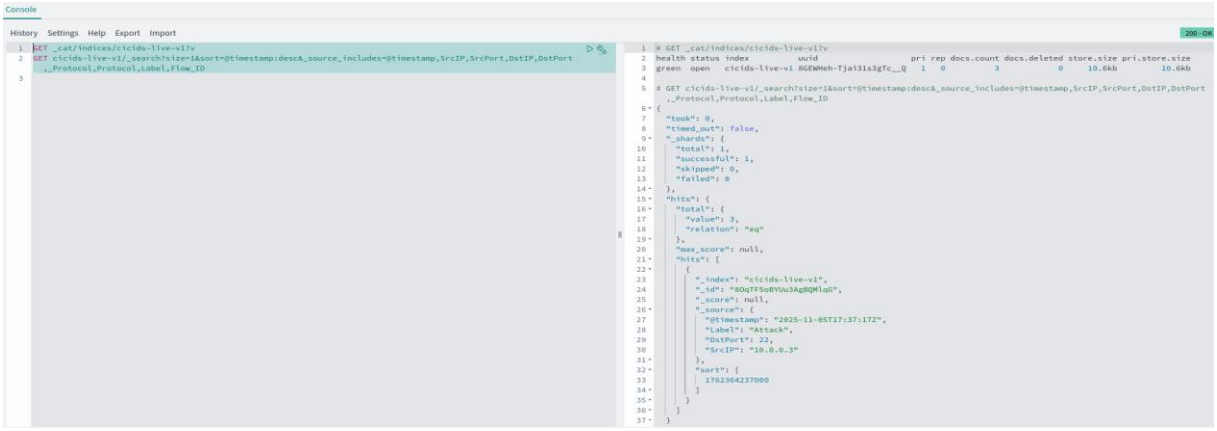
Εικόνα 5.21: Ενδεικτικό mapping (schema) του δείκτη cicids-2019-03-11-v3, με ορισμό τύπων πεδίων

Με αυτή τη χαρτογράφηση, τα δεδομένα των ροών του δικτύου αποθηκεύονται με μία δομή που ταιριάζει στις ανάγκες του αγωγού του συστήματος Παρακολούθησης της Ασφάλειας, δηλαδή στη σωστή χρονική ανάλυση, στο καθαρό διαχωρισμό των αριθμητικών και των κατηγορικών χαρακτηριστικών και στη δυνατότητα των σύνθετων ερωτημάτων που υποστηρίζουν τόσο τη παρακολούθηση όσο και τα επόμενα στάδια του εμπλουτισμού και της πρόβλεψης με χρήση της Μηχανικής Μάθησης.

5.6.2 Δείκτης ζωντανών ροών δικτύου (Live Flow Index)

Επιπλέον, ο δείκτης των ζωντανών ροών του δικτύου (live flow index) είναι ουσιαστικά το πρώτο σημείο όπου καταλήγουν τα δεδομένα μόλις εισαχθούν στη πλατφόρμα του OpenSearch. Ο δείκτης αυτός, λειτουργεί ως πρωτογενές αποθετήριο (landing index) για τις καταγραφές της ροής, πριν γίνει οποιαδήποτε επιπλέον επεξεργασία ή εμπλουτισμός. Στην υλοποίηση αυτή, το ρόλο αυτόν τον αναλαμβάνει ο δείκτης cicids-live-v1, ώστε οι εισερχόμενες εγγραφές να είναι άμεσα διαθέσιμες για αναζήτηση, για έλεγχο και για τροφοδότηση των επόμενων σταδίων του αγωγού Παρακολούθησης της Κυβερνοασφάλειας.

Η σωστή δημιουργία και η συνεχής ενημέρωση του δείκτη αναλύονται στην Εικόνα 5.22. Αρχικά, με την εντολή _cat/indices επιβεβαιώνεται ότι ο δείκτης είναι ενεργός (open) και σε καλή κατάσταση (green), ενώ την ίδια στιγμή φαίνεται και ο αριθμός των εγγράφων (docs.count) ο οποίος δείχνει ότι ο δείκτης περιέχει δεδομένα και δεν είναι κενός. Στη συνέχεια, εκτελείται ένα ερώτημα αναζήτησης με size=1 και η ταξινόμηση @timestamp:desc, ώστε να επιστραφεί η πιο πρόσφατη καταγραφή της ροής. Η απόκριση εμφανίζει τα χαρακτηριστικά πεδία μιας δικτυακής ροής, όπως για παράδειγμα τη χρονική σήμανση (@timestamp), τα βασικά στοιχεία της διευθυνσιοδότησης και μεταφοράς (SrcIP, DstPort) και την ετικέτα της κλάσης (Label). Με αυτό το τρόπο, επιβεβαιώνεται ότι το περιεχόμενο του δείκτη αντιστοιχεί πραγματικά στα δεδομένα της ροής όπως απαιτείται για τη παρακολούθηση και τη ταξινόμηση των συμβάντων στο πλαίσιο της Παρακολούθησης της Ασφάλειας (CM).

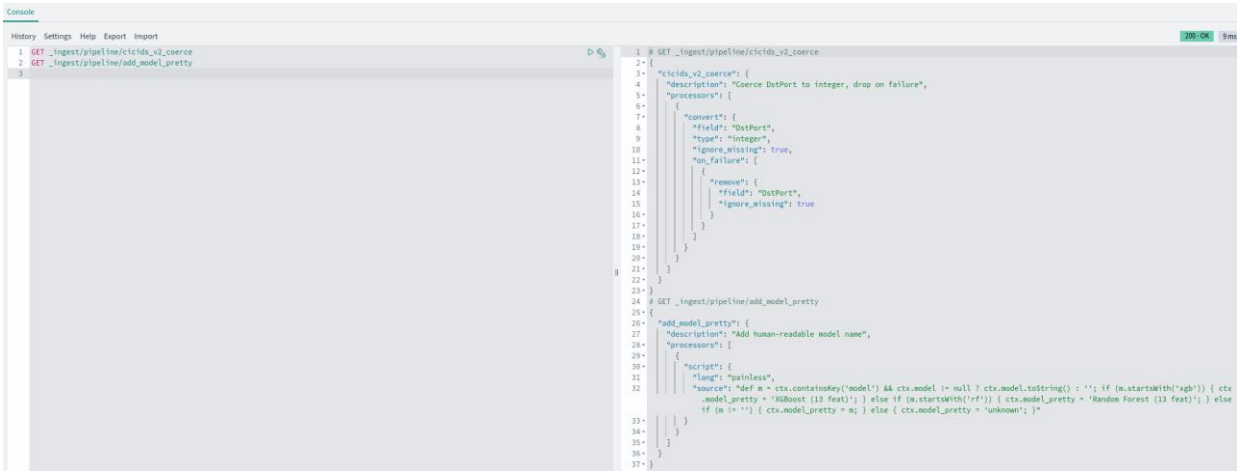


Εικόνα 5.22: Έλεγχος λειτουργίας του δείκτη ζωντανών ροών δικτύου cicids-live-v1

5.6.3 Αγωγοί εμπλουτισμού και παρακολούθησης κατά την εισαγωγή(Ingest preprocessing & Enrichment pipelines)

Στο πλαίσιο του αγωγού Παρακολούθησης της Ασφάλειας, αξιοποιήθηκαν οι αγωγοί της εισαγωγής (ingest pipelines), ώστε η προεπεξεργασία (preprocessing) και ο εμπλουτισμός (enrichment) των δεδομένων να πραγματοποιούνται συστηματικά στο σημείο της εισόδου προς το OpenSearch πριν από την αποθήκευση τους στους αντίστοιχους δείκτες. Η επιλογή αυτή επιτρέπει την ομοιογενή διαχείριση των εγγράφων των ροών του δικτύου και μειώνει τη πιθανότητα των ασυνεπειών κατά την αναζήτηση, την οπτικοποίηση και τη μετέπειτα αξιοποίηση των δεδομένων από τα επόμενα στάδια του συστήματος.

Όπως αποτυπώνεται και στην Εικόνα 5.23, υλοποιήθηκαν δύο χαρακτηριστικοί αγωγοί με συμπληρωματικούς ρόλους. Ο αγωγός cicids_v2_coerse εστιάζει στη τυποποίηση των τύπων των πεδίων (schema/type coercion), μετατρέποντας το πεδίο DstPort σε ακέραιο αριθμό μέσω του επεξεργαστή convert. Επιπλέον, έχει οριστεί η πολιτική on_failure έτσι ώστε σε περίπτωση που η μετατροπή αποτύχει να αφαιρεθεί το πεδίο, αποτρέποντας την εισαγωγή των τιμών που θα οδηγήσουν σε ασυμβατότητες με τη χαρτογράφηση των πεδίων (mapping) και αυτό μπορεί να αποφέρει σφάλματα ή στρεβλώσεις κατά την αναζήτηση και τις συναθροίσεις. Ο δεύτερος αγωγός add_model_pretty, υλοποιεί εμπλουτισμό με έναν επεξεργαστή σεναρίου σε γλώσσα Painless, δημιουργώντας ένα πιο εύχρηστο πεδίο model_pretty με βάση τη τιμή ενός αναγνωριστικού (model). Η προσθήκη αυτή δεν μεταβάλλει τη πρωτογενή πληροφορία, αλλά διευκολύνει τη σημασιολογική ερμηνεία των εγγράφων και την αποτελεσματικότερη ομαδοποίηση τους σε πίνακες ελέγχου (dashboards) και ερωτήματα.



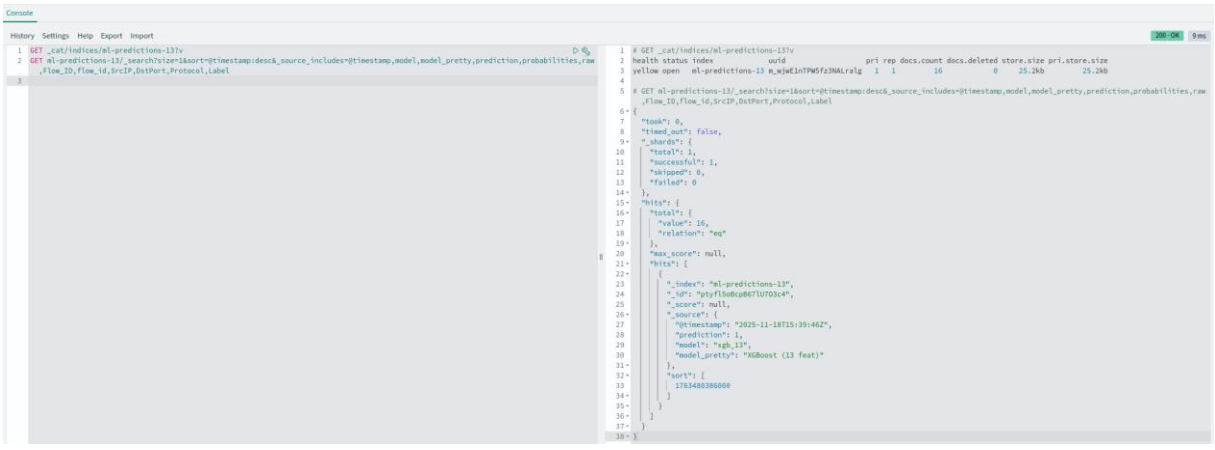
Εικόνα 5.23: Ορισμός ingest preprocessing & enrichment pipelines στο OpenSearch

Με το παραπάνω σχεδιασμό, οι αγωγοί της εισαγωγής λειτουργούν ως ένα ενδιάμεσο επίπεδο ελέγχου της ποιότητας (quality control) και του σημασιολογικού εμπλουτισμού (semantic enrichment) των δεδομένων των ροών του δικτύου. Έτσι, το σύστημα της Παρακολούθησης της Ασφάλειας αποκτά γενικά μια πιο συνεπή και προβλέψιμη συμπεριφορά στις συνθήκες της συνεχούς εισαγωγής, καθώς οι βασικοί κανόνες της κανονικοποίησης και τα βοηθητικά πεδία της παρακολούθησης εφαρμόζονται αυτόματα και με ενιαίο τρόπο σε κάθε νέα εγγραφή.

5.6.4 Δείκτης προβλέψεων Μηχανικής Μάθησης (ML-Predictions Index)

Πρώτα από όλα, ο δείκτης των προβλέψεων της Μηχανικής Μάθησης (ML-predictions index) αποτελεί το σημείο της συγκέντρωσης των αποτελεσμάτων της ταξινόμησης που παράγονται κατά την εισαγωγή (ingest-time inference) στο πλαίσιο του αγωγού Παρακολούθησης της Κυβερνοασφάλειας. Ο ml-predictions-13, αποθηκεύει την εμπλουτισμένη εκδοχή της πληροφορίας, δηλαδή την εκτίμηση της κλάσης (prediction) και τα αντίστοιχα μεταδεδομένα του μοντέλου στο οποίο πραγματοποιήθηκε η πρόβλεψη, σε αντίθεση με το δείκτη των ζωντανών ροών (live flow index), ο οποίος αποθηκεύει τις αρχικές καταγραφές των ροών του δικτύου. Με αυτό το τρόπο, επιτυγχάνεται ο σαφής διαχωρισμός μεταξύ των πρωτογενών δεδομένων και των αποτελεσμάτων της ανάλυσης διευκολύνοντας την αναζήτηση αλλά και την οπτικοποίηση στο επίπεδο των συμβάντων της ασφάλειας.

Επίσης, η σωστή ενημέρωση του δείκτη φαίνεται καθαρά στην Εικόνα 5.24. Αρχικά, μέσω της κλήσης _cat/indices/ml-predictions-13?v επιβεβαιώνεται ότι ο δείκτης είναι διαθέσιμος και περιέχει έγγραφα. Στη συνέχεια, με την αναζήτηση ml-predictions-13/_search και τη κατάλληλη ταξινόμηση με βάση το πεδίο @timestamp, εμφανίζεται η ενδεικτική εγγραφή στην οποία παρουσιάζονται η χρονική σήμανση, η τιμή της πρόβλεψης καθώς και τα πεδία ταυτοποίησης του μοντέλου, όπως για παράδειγμα το model: “xgb_13” και το model_pretty: “XGBoost (13 feat)”. Η συγκεκριμένη δομή των εγγράφων είναι η πρακτική βάση των επόμενων φάσεων, δηλαδή των πινάκων ελέγχου και των μηχανισμών της ειδοποίησης, καθώς επιτρέπει τη συσχέτιση των προβλέψεων με τη δικτυακή δραστηριότητα και τη σύνθεση των δεικτών της ασφάλειας (KPIs) που στηρίζονται στις εξόδους των μοντέλων.



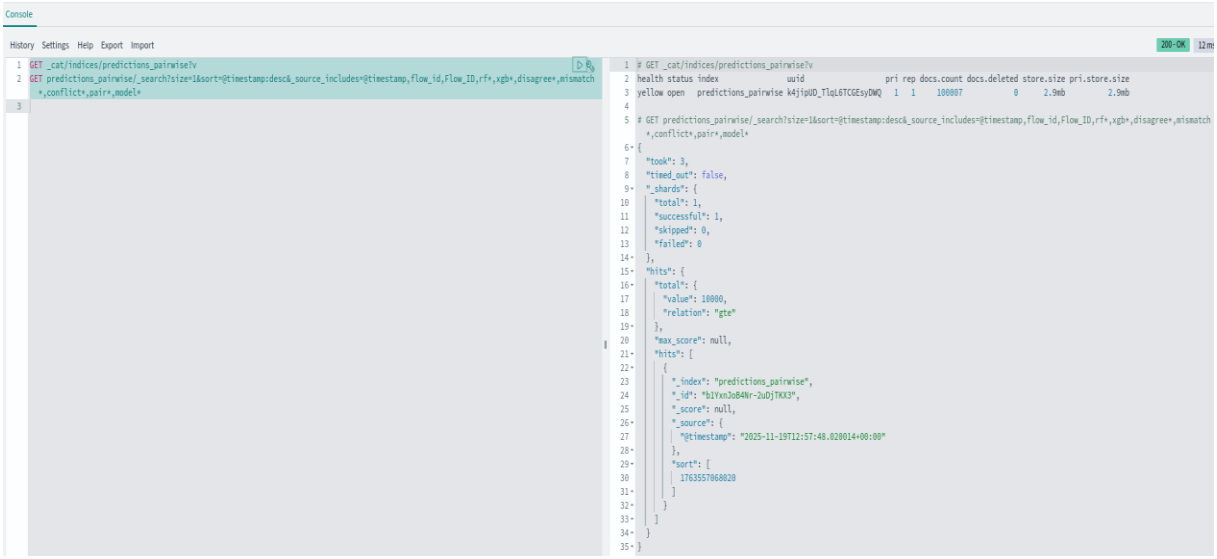
Εικόνα 5.24: Επιβεβαίωση λειτουργίας του δείκτη προβλέψεων Μηχανικής Μάθησης ml-predictions-13

5.6.5 Δείκτης συγκριτικών προβλέψεων RF - XGB (Pairwise RF – XGB Index)

Για τη συστηματική σύγκριση των δύο μοντέλων της ταξινόμησης, Random Forest και XGBoost υλοποιήθηκε ένας ξεχωριστός δείκτης των συγκριτικών προβλέψεων (pairwise index), ο οποίος σε μία εγγραφή καταγράφει τα αποτελέσματα και των δύο μοντέλων για το ίδιο συμβάν. Με αυτό το

σχεδιασμό, η ανάλυση δεν περιορίζεται μόνο στις προβλέψεις που δίνει κάθε μοντέλο ξεχωριστά αλλά επεκτείνεται στη ζευγαρωμένη αξιολόγηση (paired analysis), όπου μπορούν να εξεταστούν αμέσως οι περιπτώσεις της συμφωνίας και της διαφωνίας μεταξύ των δύο μοντέλων. Την ίδια στιγμή, δημιουργείται μια βάση για την επόμενη επιχειρησιακή λογική, όπως για παράδειγμα κάποιοι κανόνες ειδοποίησης όταν τα δύο μοντέλα αποκλίνουν.

Στη συνέχεια, στην Εικόνα 5.25 παρουσιάζεται η λειτουργία και η χρήση του δείκτη. Ειδικότερα, η κλήση _cat/indices επιβεβαιώνει την ύπαρξη του δείκτη prediction_pairwise και των αριθμό των αποθηκευμένων εγγράφων, ενώ η αναζήτηση _search βρίσκει το πιο πρόσφατο έγγραφο με βάση τη χρονική στιγμη, @timestamp. Επιπλέον, το ερώτημα περιορίζει σκόπιμα το αποτέλεσμα με τη χρήση του _source_includes, ώστε να επιστρέφονται μόνο τα χρήσιμα πεδία για τη σύγκριση. Έτσι, αντί να εμφανίζεται όλο το έγγραφο, προβάλλονται κυρίως τα πεδία των δύο μοντέλων, Random Forest και XGBoost, μαζί με τις αντίστοιχες ενδείξεις της απόκλισης, disagree*, mismatch* και conflict*. Η επιλογή αυτή, κάνει πιο εύκολη τη γρήγορη διερεύνηση (search) και τις συναθροίσεις (aggregation) πάνω στις περιπώσεις της διαφωνίας των μοντέλων χωρίς να χρειάζεται η συσχέτιση τύπου join ανάμεσα στους πολλαπλούς δείκτες, το οποίο είναι πολύ χρήσιμο στο περιβάλλον Παρακολούθησης της Ασφάλειας, όπου η ταχύτητα της διερεύνησης και η ιχνηλασιμότητα (traceability) είναι πολύ καθοριστικές.



Εικόνα 5.25: Δείκτης συγκριτικών προβλέψεων RF–XGB, predictions_pairwise

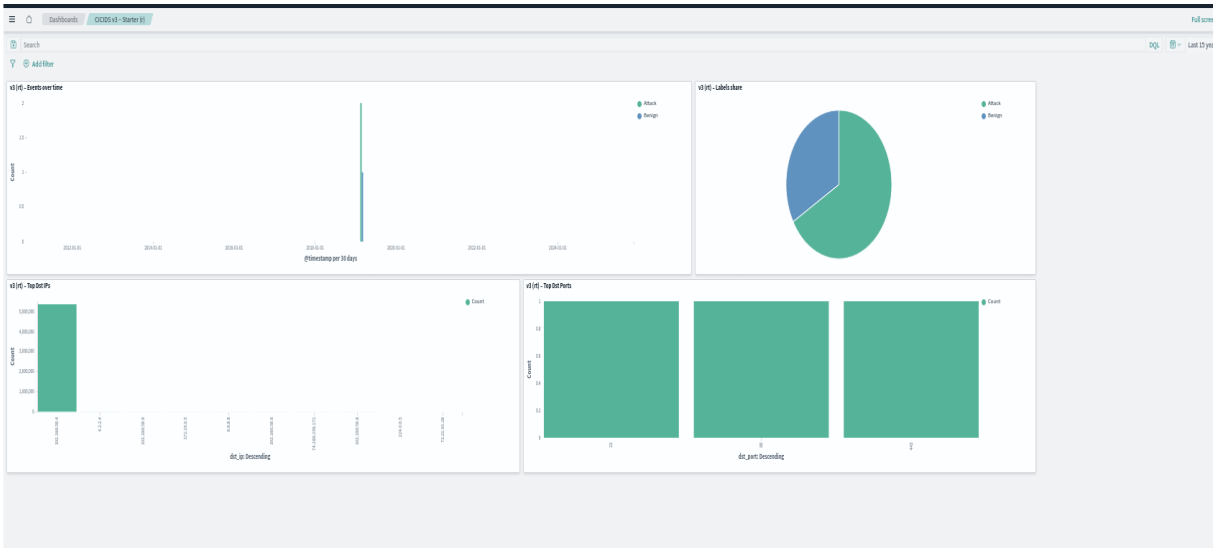
5.7 Οπτικοποίηση στο OpenSearch Dashboards (παρακολούθηση και ανάλυση CM)

5.7.1 Πίνακας συμβάντων (Events Dashboard)

Αρχικα, η οπτικοποίηση των συμβάντων στο OpenSearch Dashboards είναι ένα πολύ σημαντικό στοιχείο σε ένα σύστημα Παρακολούθησης της Ασφάλειας, γιατί μετατρέπει τις αποθηκευμένες ροές του δικτύου σε μια εικόνα της κατάστασης (situational awareness). Για το σκοπό αυτό δημιουργήθηκε ο Πίνακας των Συμβάντων (Events Dashboard), ο οποίος συγκεντρώνει τις βασικές όψεις της ροής των δεδομένων και υποστηρίζει τη γρήγορη επισκόπηση της χρονικής εξέλιξης, της αναλογίας μεταξύ της κανονικής και της κακόβουλης κίνησης, καθώς και τους συχνότερους προορισμούς (destination IPs) και τις συχνότερες θύρες προορισμού (destination ports). Έτσι, ο αναλυτής έχει την εικόνα της ροής

των δεδομένων και μπορεί να επιβεβαιώσει ότι η εισαγωγή (ingestion) εξελίσσεται ομαλά πριν προχωρήσει σε λεπτομερή ανάλυση.

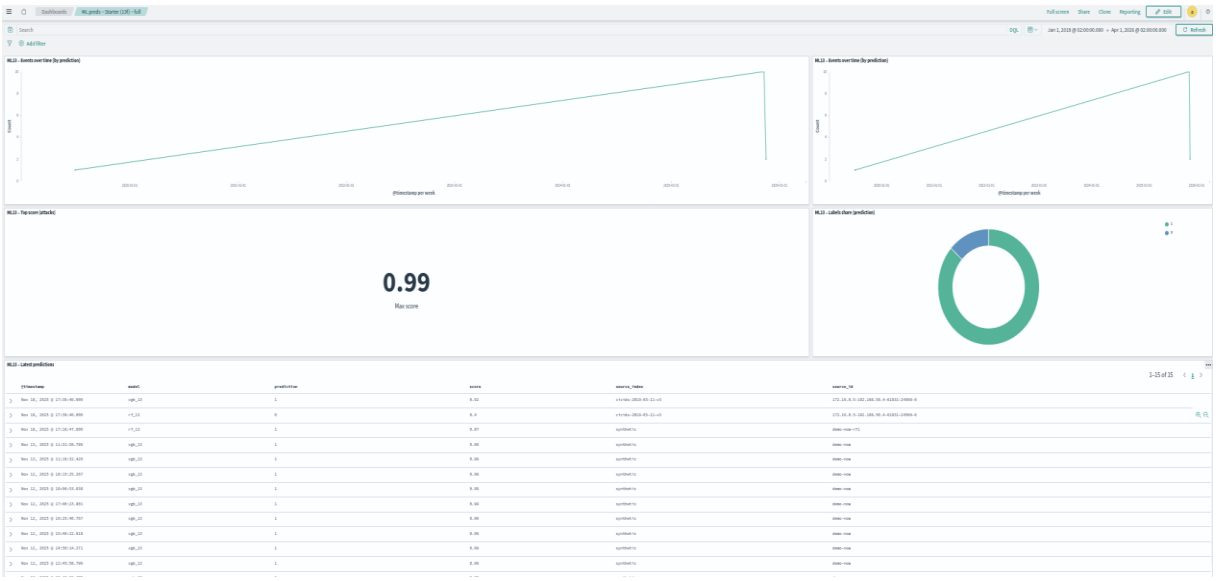
Στην Εικόνα 5.26 παρουσιάζεται ο πίνακας όπως υλοποιήθηκε στο πειραματικό περιβάλλον, για δεδομένα του CIC-DDoS2019 που αντιστοιχούν αποκλειστικά στην ημέρα 11/03/2019. Η απεικόνιση events over time, δείχνει τη χρονική εξέλιξη των συμβάντων και επιτρέπει τον εντοπισμό των αιχμών στο επιλεγμένο φίλτρο. Η οπτικοποίηση labels share δείχνει συνοπτικά την αναλογία benign έναντι attack στο εξεταζόμενο διάστημα. Η κατηγορία attack αντιστοιχεί στην ομαδοποίηση των επιμέρους DoS/DDoS labels το οποίο είναι χρήσιμο τόσο στη κατανόηση του μείγματος της κίνησης όσο και για την ερμηνεία των επόμενων ευρημάτων. Τέλος, οι οπτικοποιήσιες top destinatios IPs και top destination ports διευκολύνουν τον άμεσο εντοπισμό των επαναλαμβανόμενων στόχων ή των συγκεντρώσεων σε συγκεκριμένες θύρες, οι οποίες αποτελούν συχνά την αρχή για κάποια στοχευμένα ερωτήματα στο επίπεδο των εγγραφών (Discover).



Εικόνα 5.26: Πίνακας συμβάντων, Events Dashboard στο OpenSearch Dashboards

5.7.2 Πίνακας προβλέψεων Μηχανικής Μάθησης (ML Predictions Dashboard)

Στη συνέχεια, η οπτικοποίηση των αποτελεσμάτων της πρόβλεψης υλοποιήθηκε μέσω του ειδικού πίνακα στο OpenSearch Dashboards, ο οποίος συγκεντρώνει τις εγγραφές από το δείκτη των προβλέψεων, ml-predictions-13 για δεδομένα της ημέρας 11/03/2019 και τις παρουσιάζει σε κατάλληλη μορφή για επιχειρησιακή παρακολούθηση και γρήγορη διερεύνηση. Όπως φαίνεται και στην Εικόνα 5.27, ο πίνακας συνδυάζει τις χρονικές απεικονίσεις της ροής των προβλέψεων (events over time) με τους συνοπτικούς δείκτες, ώστε η συμπεριφορά του συστήματος να αξιολογείται άμεσα σε πραγματικό ή σχεδόν πραγματικό χρόνο, χωρίς να απαιτείται η εκτέλεση των εξωτερικών σεναρίων της επεξεργασίας (offline scripts).

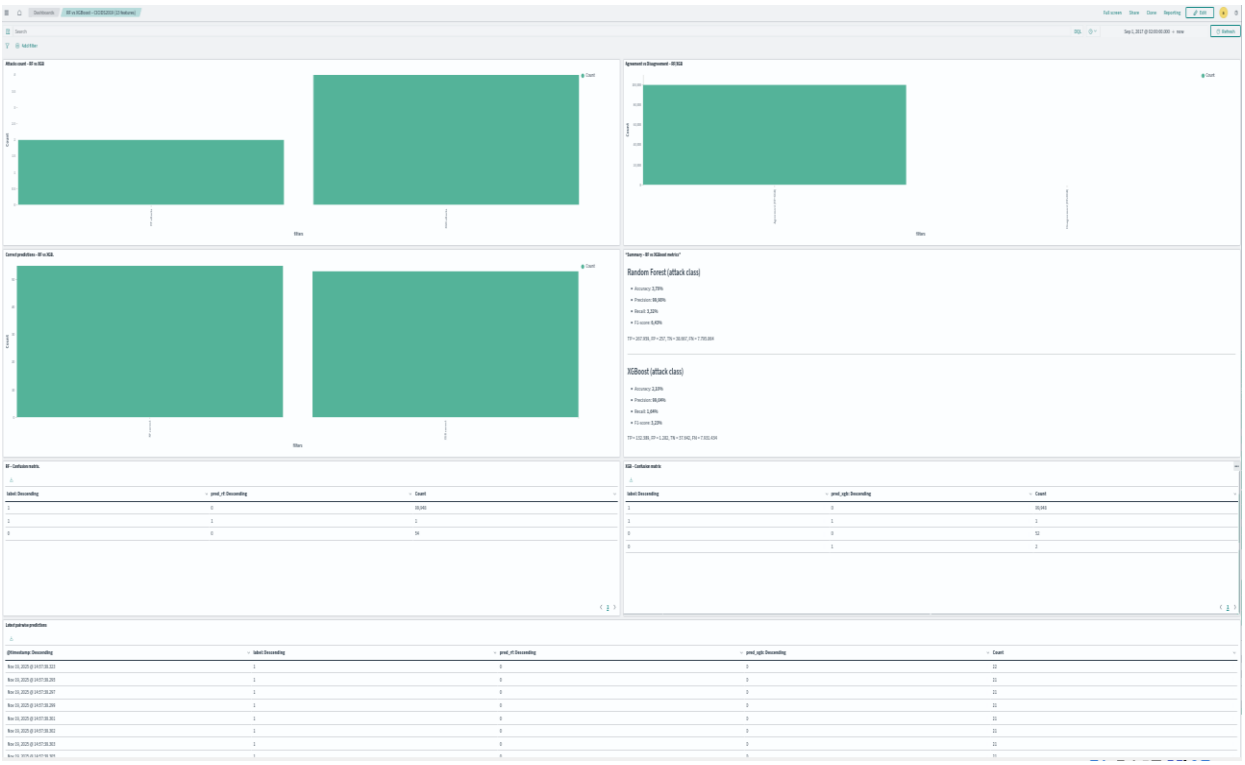


Εικόνα 5.27: Πίνακας προβλέψεων Μηχανικής Μάθηση, ML Predictions Dashboard στο OpenSearch Dashboards

Παράλληλα, αποτυπώνεται η κατανομή των προβλεπόμενων κλάσεων μέσω της οπτικοποίησης (Labels share – prediction), ενώ ο δείκτης Top score αποτυπώνει τη μέγιστη τιμή της εμπιστοσύνης (confidence score) που συνοδεύει τις καταγεγραμμένες προβλέψεις στο επιλεγμένο χρονικό παράθυρο. Επιπλέον, ο πίνακας Latest predictions υποστηρίζει τη διερεύνηση στο επίπεδο της εγγραφής, προβάλλοντας πεδία όπως η χρονική σήμανση (timestamp), το αναγνωριστικό του μοντέλου (model), η διάδικοί πρόβλεψη (prediction) και η βαθμολογία (score), καθώς και τα στοιχεία της συσχέτισης με την αρχική εγγραφή (source index / source id). Με αυτό το σχεδιασμό, το OpenSearch Dashboards λειτουργεί ως σημείο εποπτείας του επιπέδου της πρόβλεψης της Μηχανικής Μάθησης (ML inference), υποστηρίζοντας τόσο τη συνολική εικόνα της κατάστασης όσο και τη στοχευμένη διερεύνηση των μεμονωμένων περιστατικών.

5.7.3 Πίνακας συγκριτικής αξιολόγησης RF - XGB (Pairwise Comparison Dashboard)

Για τη συστηματική σύγκριση των δύο μοντέλων της ταξινόμησης, Random Forest και XGBoost στο επίπεδο της μεμονωμένης ροής (per-flow), αναπτύχθηκε ένας ειδικός πίνακας στο OpenSearch Dashboards, ο οποίος βασίζεται στον δείκτη των συγκριτικών προβλέψεων, pairwise index, και αφορά δεδομένα της ημέρας 11/03/2019. Στόχος του πίνακα είναι να συγκεντρώσει σε ένα σημείο τόσο τις περίπτωσεις της συμφωνίας (agreement) όσο και τις περιπτώσεις της απόκλισης (disagreement / mismatch) μεταξύ των δύο μοντέλων, ώστε η συμπεριφορά τους να μπορεί να παρακολουθείται και να ερμηνεύεται με τρόπο άμεσα αξιοποιήσιμο. Η συνολική διάταξη και οι βασικές οπτικοποιήσεις του πίνακα παρουσιάζονται στην Εικόνα 5.28.



Εικόνα 5.28: Πίνακας συγκριτικής αξιολόγησης RF–XGB, Pairwise Comparison Dashboard στο OpenSearch Dashboards

Ακόμη, όπως αποτυπώνεται στην Εικόνα 5.28, ο πίνακας περιλαμβάνει τις απεικονίσεις που δείχνουν τη συχνότητα και τη κατανομή των περιπτώσεων όπου τα δύο μοντέλα καταλήγουν στην ίδια απόφαση ή δίνουν διαφορετική πρόβλεψη. Παράλληλα, εμφανίζονται οι συνοπτικοί δείκτες ανά μοντέλο (accuracy, precision, recall, F1-score), ώστε να είναι δυνατή μία γρήγορη αποτίμηση της απόδοσης κάτω από τις ίδιες συνθήκες των δεδομένων και του χρονικού φίλτρου. Επιπλέον, ενσωματώνονται οι οπτικοποιήσεις του τύπου πίνακα της σύγχυσης (confusion matrix), με βάση τις προβλέψεις του κάθε μοντέλου σε σχέση με τη πραγματική ετικέτα (ground truth/label), προσφέροντας μία πιο άμεση εικόνα για το είδος των σφαλμάτων που παράγει το καθένα.

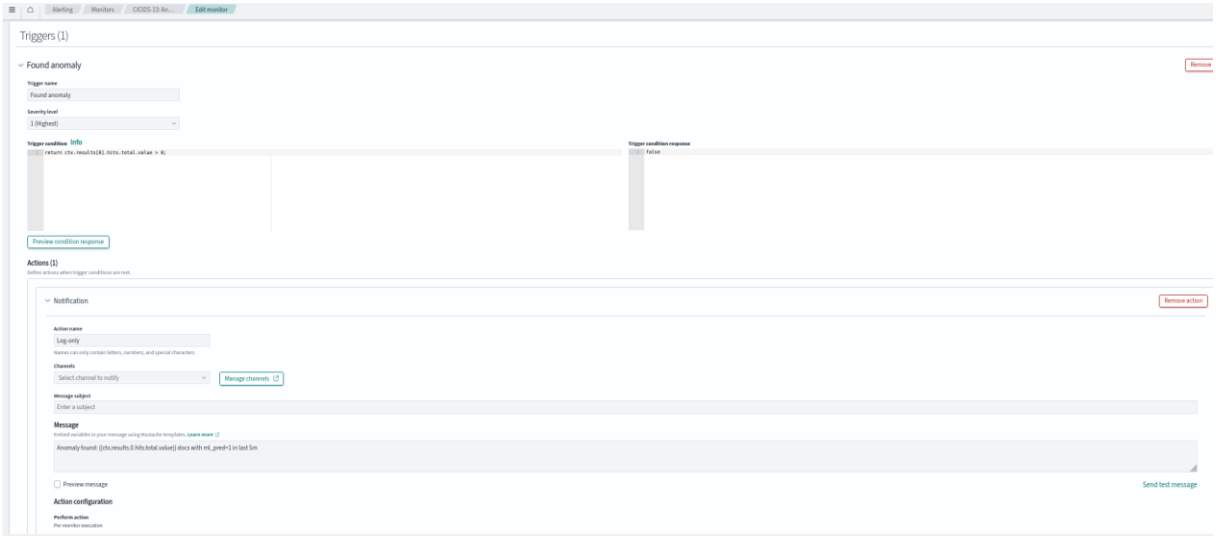
Τέλος, η ενότητα Latest pairwise predictions, επιτρέπει τον έλεγχο στο επίπεδο της εγγραφής, ώστε να εντοπίζονται τα πρόσφατα παραδείγματα της συμφωνιάς και διαφωνίας και να γίνεται στοχευμένη διερεύνηση. Με αυτό το τρόπο, η συγκριτική αξιολόγηση δεν εξαντλείται σε συγκεντρωτικά ποσοστά, αλλά υποστηρίζει τη πρακτική ανάλυση ανά περιστατικό, κάτι που είναι πολύ χρήσιμο σε ένα περιβάλλον Παρακολούθησης της Ασφάλειας, διότι απαιτείται τεκμηριωμένη και επιχειρησιακά αξιοποιήσιμη επεξήγηση των αποτελεσμάτων.

5.8 Ειδοποιήσεις στο OpenSearch μέσω Webhook (CM alerts with Webhook)

5.8.1 Παρακολουθητές ειδοποιήσεων (Monitor)

Αρχικά, στο πλαίσιο του αγωγού Παρακολούθησης της Ασφάλειας, υλοποιήθηκε ένας παρακολούθητης ειδοποιήσεων (monitor) στο OpenSearch Alerting, με στόχο να εντοπίζει έγκαιρα τα ύποπτα συμβάντα που προκύπτουν από το σύστημα των προβλέψεων. Η ενεργοποίηση βασίζεται σε ένα κανόνα (trigger) με μία συνθήκη, η οποία αξιολογείται πάνω στα αποτελέσματα του ερωτήματος (query) του παρακολουθητή των ειδοποιήσεων. Όπως φαίνεται και στην Εικόνα 5.29, η συνθήκη υλοποιείται με ένα κώδικα τύπου Painless (Painless script) και ελέγχει αν το ερώτημα επέστρεψε έστω και μία εγγραφή,

μέσω της έκφρασης ctx.results[0].hits.total.value > 0. Έτσι, κάθε χρονικό παράθυρο της αξιολόγησης (evaluation window) στο οποίο εντοπίζεται τουλάχιστον ένα σχετικό γεγονός οδηγεί σε ενεργοποίηση του trigger, με ένα απλό αλλά πρακτικό κριτήριο κατωφλίου (threshold-based detection).



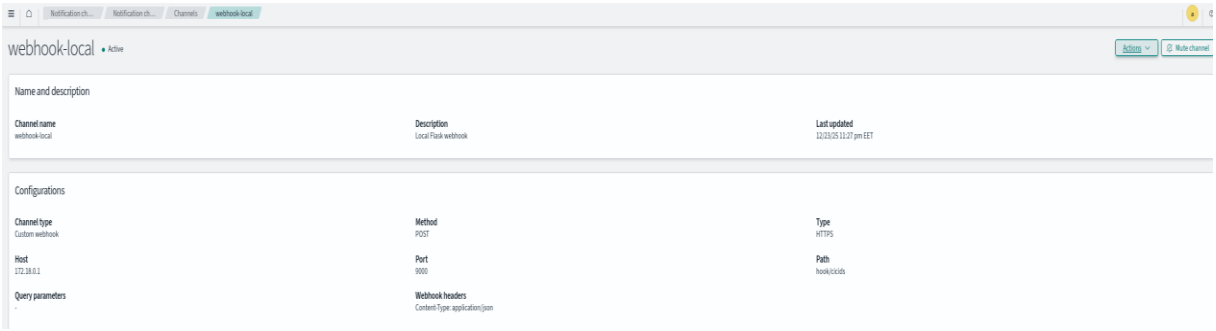
Εικόνα 5.29: Ορισμός trigger και ενέργειας ειδοποίησης σε Monitor του OpenSearch Alerting, Found anomaly

Οπότε, όταν ικανοποιηθεί η συνθήκη ο παρακολουθητής των ειδοποιήσεων (monitor) εκτελεί μια ενέργεια ειδοποίησης (action), στέλνοντας μήνυμα προς ένα προκαθορισμένο κανάλι. Στην ίδια Εικόνα αποτυπώνεται η ενότητα Actions, στην οποία ορίζεται το περιεχόμενο της ειδοποίησης με ένα πρότυπο Mustache, ώστε να ενσωματώνονται δυναμικά τα στοιχεία από τα αποτελέσματα του ερωτήματος. Ενδεικτικά, αξιοποιείται η μεταβλητή ctx.results.0.hits.total.value για να αναφερθεί το πόσες από τις εγγραφές ικανοποίησαν το κριτήριο στο τελευταίο χρονικό διάστημα. Με αυτό το τρόπο, η ειδοποίηση μεταφέρει συνοπτικά και μετρήσιμα στοιχεία για το τι ακριβώς ανιχνεύθηκε, κάνοντας πιο εύκολη την επιχειρησιακή εικόνα (operational awareness) όσο και την ιχνηλασιμότητα (traceability), ενώ στη συνέχεια η ίδια μπορεί να προωθηθεί μέσω του webhook και για καταγραφή κάποιον περιστατικών.

5.8.2 Κανάλι Webhook (Webhook Channel)

Ακόμη, στο πλαίσιο του μηχανισμού των ειδοποιήσεων του συστήματος Παρακολούθησης της Κυβερνοασφάλειας (CM), ορίστηκε ένα κανάλι webhook ως γέφυρα για την αποστολή των ειδοποιήσεων σε μία εξωτερική υπηρεσία συλλογής. Με την επιλογή αυτή, οι ειδοποιήσεις δεν μένουν μόνο μέσα στο OpenSearch, αλλά προωθούνται άμεσα σε έναν HTTP δέκτη, ώστε η τελική παράδοση και η περαιτέρω διαχείριση τους να γίνεται έξω από τη πλατφόρμα, με μεγαλύτερη ευελιξία στη μορφή και στην επεξεργασία του μηνύματος.

Έτσι, όπως φαίνεται και στην Εικόνα 5.30, το κανάλι ρυθμίστηκε ως custom webhook με τη μέθοδο HTTP POST, με καθορισμό του host, port και της διαδρομής (path) τα οποία αντιστοιχούν στο τελικό σημείο της πρόσβασης (endpoint) του τοπικού συλλέκτη. Παράλληλα, ορίστηκαν οι κεφαλίδες HTTP (headers) με Content-Type: application/json, ώστε το περιεχόμενο της ειδοποίησης να αποστέλλεται ως ένα φορτίο JSON. Η ρύθμιση αυτή διευκολύνει το δείκτη στο να διαβάζει και να αξιοποιεί άμεσα το μήνυμα, είτε για αποθήκευση και καταγραφή, είτε για περαιτέρω δρομολόγηση (routing) ή μετατροπή σε άλλα κανάλια ειδοποίησης.

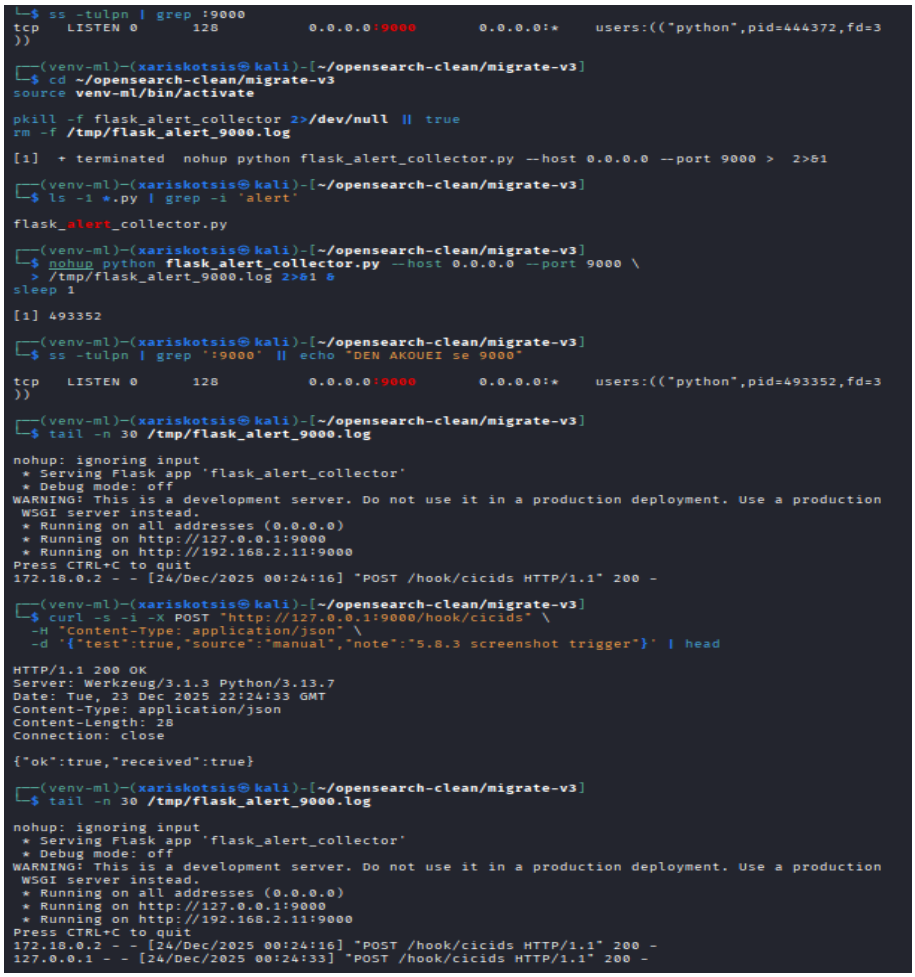


Εικόνα 5.30: Ρύθμιση καναλιού Webhook για αποστολή ειδοποιήσεων σε τοπικό συλλέκτη, Flask, με μέθοδο POST

5.8.3 Συλλέκτης ειδοποιήσεων σε Flask (Flask Alert Collector)

Τέλος, για τη παραλαβή και την αξιοποίηση των ειδοποιήσεων από το OpenSearch Alerting υλοποιήθηκε ένας ελαφρύς συλλέκτης ειδοποιήσεων με χρήση του Flask (Flask Alert Collector). Ο συλλέκτης λειτουργεί ως ένα σημείο πρόσβασης το οποίο δέχεται κάποιες κλήσεις τύπου POST από το κανάλι webhook του OpenSearch. Έτσι, ο μηχανισμός που εντοπίζει και ενεργοποιεί μια ειδοποίηση (monitor/trigger) παραμένει ανεξάρτητος από το τρόπο με τον οποίο η ειδοποίηση παραδίδεται και αξιοποιείται στη συνέχεια. Πρακτικά, αυτό επιτρέπει στις ειδοποιήσεις (alerts) να καταλήγουν σε κάποιες εξωτερικές ροές, όπως η αποθήκευση και η προώθηση σε πλατφόρμες μηνυμάτων χωρίς να χρειάζονται αλλαγές στη λογική του OpenSearch.

Η λειτουργία του συλλέκτη επιβεβαιώθηκε με μερικούς πρακτικούς ελέγχους εκτέλεσης και καταγραφής. Όπως, φαίνεται και στην Εικόνα 5.31, ο συλλεκτης των ειδοποήσεων Flask εκκινεί κανονικά και ακούει στη θύρα 9000 σε όλες τις διεπαφές, ενώ τα εισερχόμενα αιτήματα προς το σημείο της πρόσβασης /hook/cicids καταγράφονται στα αρχεία της καταγραφής (log). Επιπλέον, πραγματοποιήθηκε η δοκιμαστική αποστολή POST με χρήση του curl, η οποία επιστρέφει HTTP 200 OK, επιβεβαιώνοντας οτι ο συλλέκτης είναι προσβάσιμος και οτι ο κύκλος του αιτήματος (request) και της απόκρισης (response) ολοκληρώνεται σωστά. Η καταγραφή σε ξεχωριστό αρχείο βοηθάει στην παρακολούθηση και στη διάγνωση, διότι δίνει καθαρή εικόνα για το πότε ελήφθη μια ειδοποίηση, σε ποιο σημείο της πρόσβασης παραδόθηκε και αν ο χειρισμός της ολοκληρώθηκε επιτυχώς.



Εικόνα 5.31: Εκτέλεση και επαλήθευση λειτουργίας του Flask Alert Collector

5.9 Επίλογος κεφαλαίου

Εν κατακλείδι, στο κεφάλαιο αυτό παρουσιάστηκε η υλοποίηση ενός ολοκληρωμένου συστήματος Παρακολούθησης της Ασφάλειας (CM) πάνω στη πλατφόρμα του OpenSearch, με βασικό στόχο να φανεί καθαρά η μετάβαση από την εκτός σύνδεση εκπαίδευση των μοντέλων σε μια λειτουργική, επιχειρησιακή ροή από άκρη σε άκρη (edge-to-edge). Η μελέτη της περίπτωσης βασίστηκε στο CIC-DDoS2019 και αξιοποίησε τα δεδομένα αποκλειστικά από την ημέρα 11/03/2019. Για το λόγο αυτό, η αξιολόγηση αντιμετωπίζεται ως ενδοημερήσιος διαχωρισμός της εκπαίδευσης και του ελέγχου (within-day holdout). Παράλληλα, εξετάστηκαν τόσο τα σενάρια χωρίς διαρροή πληροφορίας (non-leaky) όσο και τα ελεγχόμενα σενάρια με διαρροή (leaky), ώστε να φανεί στη πράξη πως η διαρροή της πληροφορίας (data leakage) μπορεί να φουσκώσει τεχνητά τις μετρικές και να δώσει μία παραπλανητικά πολύ καλύτερη εικόνα σε σχέση με τη πραγματική συμπεριφορά ενός συστήματος Παρακολούθησης της Ασφάλειας.

Στο επίπεδο της υποδομής, παρουσιάστηκε ένα σταθερό και εύκολα επαναλήψιμο περιβάλλον πειραματισμού, αξιοποιώντας την εικονικοποίηση (virtualization) και την ανάπτυξη σε δοχεία (containerization) μέσω του Docker Compose. Η επιλογή αυτή επέτρεψε τον καθαρό διαχωρισμό των υπηρεσιών και τα ελεγχόμενα σημεία της πρόσβασης (endpoints), μειώνοντας έτσι την πιθανότητα των αστοχιών που σχετίζονται με τις χειροκίνητες ρυθμίσεις. Στη συνέχεια, τα μοντέλα Random Forest και XGBoost εκπαιδεύτηκαν, αξιολογήθηκαν και εξήχθησαν σε μορφή ONNX (Open Neural Network

Exchange), ώστε να είναι φορητά και να εκτελούνται ανεξάρτητα από το αρχικό πλαίσιο της εκπαίδευσης. Για την επιχειρησιακή αξιοποίηση τους, αναπτύχθηκε ένας ανεξάρτητος προγνωστικός εξυπηρετητής REST (REST predictor), ο οποίος εκτελεί τα μοντέλα με βάση το χρόνο εκτέλεσης ONNX, έχει μια συγκεκριμένη μορφή εισόδου και εξόδου (inference contract) και υποστηρίζει τους βασικούς ελέγχους της ετοιμότητας και της καταγραφής (health checks & logging).

Ακολούθως, επιτεύχθηκε η ενσωμάτωση των μοντέλων στο σύστημα του OpenSearch μέσω των συνδετήρων της Μηχανικής Μάθησης (ML connectors) και των απομακρυσμένων μοντέλων (REMOTE models). Με αυτό το τρόπο, ο συμπερασμός (remote inference) εκτελείται έξω από το cluster, αλλά καλείται με συγκεκριμένη μορφή και με ελεγχόμενο τρόπο από το ML plugin, σαν να είναι ενσωματωμένη λειτουργία της πλατφόρμας. Η ενσωμάτωση επεκτάθηκε και στο στάδιο της εισαγωγής, με τους αγωγούς της εισαγωγής (ingest pipelines) και με κατάλληλους δείκτες (indices) οι οποίοι υποστηρίζουν την αποθήκευση των πρωτογενών εγγραφών την ροής (flow records), τη συγκέντρωση των προβλέψεων (ML-predictions index) και τη ζευγαρωμένη σύγκριση (pairwise index) των αποτελεσμάτων Random Forest και XGBoost. Αυτός ο διαχωρισμός βοήθησε στο να παραμείνει καθαρή η γραμμή ανάμεσα στα δεδομένα της εισόδου και στα παραγόμενα αποτελέσματα διευκολύνοντας έτσι την αναζήτηση και την ανάλυσή τους.

Στη συνέχεια, στο επίπεδο της παρακολούθησης, το οικοσύστημα OpenSearch Dashboards χρησιμοποιήθηκε για να δοθεί μια πρακτική εικόνα της λειτουργίας των συστημάτων Παρακολούθησης της Ασφάλειας (CM). Για αυτό το λόγο, υλοποιήθηκαν τρεις συμπληρωματικές οπτικοποιήσεις. Πρώτον, ο πίνακας των συμβάντων για τη παρατήρηση της κίνησης και των ετικετών (events dashboard), δεύτερον ο πίνακας των προβλέψεων για τη παρακολούθηση των εξόδων των μοντέλων και των σχετικών δεικτών (ML predictions dashboard) και τρίτον ο πίνακας της ζευγαρωμένης σύγκρισης για το γρήγορο έλεγχο της συμφωνίας ή διαφωνίας ανάμεσα στα δύο μοντέλα, Random Forest και XGBoost (pairwise comparison dashboard). Τέλος, για να αποκτήσει το σύστημα και διάσταση έγκαιρης προειδοποίησης, υλοποιήθηκαν οι μηχανισμοί της ειδοποίησης (alerting) μέσω των παρακολουθητών (monitors) και των καναλίων webhook (webhook channels), με τελικό παραλήπτη έναν εξωτερικό συλλέκτη Flask (Flask alert collector). Έτσι, το σύστημα Παρακολούθησης της Ασφάλειας δεν περιορίζεται σε αναλύσεις των γεγονότων αφού συμβούν αλλά υποστηρίζει και την έγκαιρη παρακολούθηση και αντίδραση κατά τη ροή της λειτουργίας.

Κλείνοντας, στο κεφάλαιο συνολικά τεκμηριώθηκε μία πλήρης αλυσίδα Παρακολούθησης της Ασφάλειας (CM pipeline), ξεκινώντας από την εισαγωγή και την ευρετηρίαση (ingest/indexing), περνώντας στον εμπλουτισμό με τη Μηχανική Μάθηση (ML enrichment) και την αποθήκευση των αποτελεσμάτων και φτάνοντας μέχρι την οπτικοποίηση και την ενεργοποίηση των ειδοποιήσεων. Το βασικό συμπέρασμα είναι ότι η αξία της Μηχανικής Μάθησης στην κυβερνοασφάλεια δεν κρίνεται μόνο από τις μετρικές ενός πειράματος αλλά από το κατά πόσο το μοντέλο στέκεται μέσα σε μία σωστά δομημένη ροή. Δηλαδή, σε μία ροή με καθαρή προεπεξεργασία, αναπαραγώγιμη υλοποίηση, ελεγχόμενη πρόσβαση και με δυνατότητα να αξιοποιηθεί επιχειρησιακά. Στο επόμενο κεφάλαιο, οι δομές και τα δεδομένα που παράχθηκαν στο κεφάλαιο αυτό θα χρησιμοποιηθούν για τη παρουσίαση και την ερμηνεία των αποτελεσμάτων, καθώς και για τη συζήτηση των ευρημάτων και των περιορισμών της μελέτης.

Κεφάλαιο 6ο: Πειραματικά Αποτελέσματα και αξιολόγηση

6.1 Εισαγωγή κεφαλαίου

Στο κεφάλαιο αυτό, παρουσιάζονται τα πειραματικά αποτελέσματα και η αξιολόγηση των μοντέλων της Μηχανικής Μάθησης που αναπτύχθηκαν στο πλαίσιο του συστήματος Παρακολούθησης της Ασφάλειας (CM) στη πλατφόρμα του OpenSearch. Στόχος του κεφαλαίου είναι να φανεί η απόδοση των ταξινομητών μέσα από τις βασικές μετρικές τους και η πρακτική εικόνα που δίνουν όταν οι προβλέψεις εντάσσονται στη ροή των δεδομένων και αξιοποιούνται για παρακολούθηση και ανάλυση. Συγκεκριμένα, η μελέτη της περίπτωσης βασίζεται στο σύνολο δεδομένων CIC-DDoS2019 και χρησιμοποιεί τα δεδομένα αποκλειστικά από την ημέρα 11/03/2019. Επομένως, η αξιολόγηση αντιμετωπίζεται ως ενδοημερήσιος διαχωρισμός της εκπαίδευσης και του ελέγχου (within-day holdout), δηλαδή ως η διάσπαση του ίδιου του ημερήσιου χρονικού διαστήματος σε σύνολα εκπαίδευσης και ελέγχου.

Επίσης, η ανάλυση επικεντρώνεται κυρίως στα σενάρια χωρίς διαρροή πληροφορίας (non-leaky), όπου τα μοντέλα Random Forest και XGBoost αξιολογούνται σε συνθήκες που πλησιάζουν πιο πολύ στη ρεαλιστική ροή λειτουργίας. Επειδή τα δεδομένα εμφανίζουν μια ανισορροπία στις κλάσεις τους, δίνεται έμφαση στις μετρικές που αποτυπώνουν ουσιαστικότερα την ανίχνευση της μειοψηφικής κλάσης, κάποιες από αυτές είναι η ακρίβεια των θετικών προβλέψεων (precision), η ανάκληση (recall) και ιδιαίτερα ο δείκτης F1 (F1-score), που ισορροπεί τις δύο προηγούμενες. Παράλληλα, εξετάζεται το σενάριο με διαρροή (leaky) ως αντιπαράδειγμα, ώστε να φανεί στη πράξη ότι ορισμένα χαρακτηριστικά τα οποία αποκαλύπτουν άμεσα ή έμμεσα την ετικέτα μπορούν να ανεβάσουν τεχνητά τις μετρήσεις της απόδοσης και σε μια εικόνα αξιοπιστίας που δεν ανταποκρίνεται στη πραγματικότητα.

Ως προς τη δομή, το κεφάλαιο ξεκινάει με τα αποτελέσματα του Random Forest στα σενάρια χωρίς διαρροή πληροφορίας, συνεχίζει με τα αντίστοιχα αποτελέσματα του XGBoost και στη συνέχεια περνάει στη συγκριτική αξιολόγηση των δύο μοντέλων μέσα από τους πίνακες σύγχυσης και τις μετρικές, ακρίβεια των θετικών προβλέψεων, ανάκληση και δείκτη F1. Ακολούθως, γίνεται η ανάλυση των διαφωνιών (disagreement analysis), όπου εξετάζονται οι περιπτώσεις στις οποίες τα δύο μοντέλα δίνουν διαφορετική πρόβλεψη και παρουσιάζεται τι πρακτικά σημαίνει αυτό σε ένα περιβάλλον Παρακολούθησης της Ασφάλειας (CM). Τέλος, η σύγκριση των δεδομένων με διαρροή έναντι των δεδομένων χωρίς διαρροή συνοψίζει τον κίνδυνο της διαρροής πληροφορίας (data leakage) και στηρίζει με πιο καθαρό τρόπο τα συμπεράσματα που προκύπτουν για χρήση σε ένα περιβάλλον Παρακολούθησης της Ασφάλειας.

6.2 Αποτελέσματα μοντέλων χωρίς διαρροή (Non-Leaky Models)

6.2.1 Αποτελέσματα μοντέλου Random Forest χωρίς διαρροή (Non-Leaky Random Forest)

Αρχικά, στην υποενότητα αυτή παρουσιάζονται τα αποτελέσματα του μοντέλου Random Forest στο σενάριο χωρίς διαρροή πληροφορίας (non-leaky), με χρήση δεκατριών επιλεγμένων χαρακτηριστικών. Η εκπαίδευση και ο έλεγχος έγιναν με το σενάριο train_non_leaky_13.py, εφαρμόζοντας χρονολογικό διαχωρισμό 80/20 (chrono split) στα δεδομένα της ημέρας 11/03/2019. Όπως φαίνεται και στην Εικόνα 6.1, το σύνολο περιλαμβάνει 8.102.747 εγγραφές, από τις οποίες 6.482.197 χρησιμοποιούνται για την εκπαίδευση και 1.620.550 για τον έλεγχο.

```
┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ cd ~/opensearch-clean/migrate-v3
source venv-ml/bin/activate
python train_non_leaky_13.py 2>&1 | tee /tmp/nonleaky_13_6_2.txt

[INFO] chrono split loading…
[INFO] total rows=8102747 | train=6482197 test=1620550

==== RF (non-leaky, 13 feats) ====
F1= 0.9977652678531285
              precision    recall  f1-score   support

           0     0.3883    0.9574    0.5525      4647
           1     0.9999    0.9957    0.9978   1615903

    accuracy                         0.9956   1620550
   macro avg     0.6941    0.9765    0.7751   1620550
weighted avg     0.9981    0.9956    0.9965   1620550

Confusion Matrix (RF) [rows=true, cols=pred]:
[[    4449      198]
 [    7009  1608894]]

==== XGB (non-leaky, 13 feats) ====
F1= 0.9977770783882965
              precision    recall  f1-score   support

           0     0.3896    0.9572    0.5538      4647
           1     0.9999    0.9957    0.9978   1615903

    accuracy                         0.9956   1620550
   macro avg     0.6947    0.9764    0.7758   1620550
weighted avg     0.9981    0.9956    0.9965   1620550

Confusion Matrix (XGB) [rows=true, cols=pred]:
[[    4448      199]
 [    6970  1608933]]

[OK] Saved:
 - reports: reports_nonleaky_13
 - ONNX RF: models/rf_13_nonleaky.onnx
 - ONNX XGB: models/xgb_13_nonleaky.onnx
 - feature schema: models/feature_schema_13.json
```

Εικόνα 6.1: Έξοδος εκτέλεσης train_non_leaky_13.py, αποτελέσματα RF και XGB, non-leaky και με 13 χαρακτηριστικά

Γενικά, σε συνολικό επίπεδο το μοντέλο επιτυγχάνει ακρίβεια (accuracy) 0,9956 και πολύ υψηλό σταθμισμένο δείκτη F1 (weighted F1) 0,9965. Η εικόνα αυτή επηρεάζεται κυρίως από τη πλειοψηφική κλάση, για την οποία ο Random Forest εμφανίζει δείκτη F1 περίπου 0,9978, όπως φαίνεται και στην Εικόνα 6.1. Παράλληλα, η μακρομέση επίδοση F1 (macro F1) είναι αισθητά χαμηλότερη 0,7751, κάτι το οποίο είναι αναμενόμενο στα έντονα ανισόρροπα δεδομένα, καθώς η macro μέση τιμή ζυγίζει ισότιμα και τη μειοψηφική κλάση.

Πιο συγκεκριμένα, για τη μειοψηφική κλάση 0 (support 4.647), παρατηρείται πολύ υψηλή ανάκληση (recall) 0,9574 αλλά χαμηλή ακρίβεια των θετικών προβλέψεων (precision) 0,3883. Με απλά λόγια, το μοντέλο εντοπίζει τα περισσότερα πραγματικά δείγματα της κλάσης 0, όμως όταν προβλέπει 0 ένα σημαντικό μέρος αυτών των προβλέψεων αντιστοιχεί τελικά στην άλλη κλάση. Η συμπεριφορά αυτή αποτυπώνεται και στο πίνακα της σύγχυσης, όπου 7.009 δείγματα της κλάσης 1 ταξινομήθηκαν ως 0, μειώνοντας τη καθαρότητα των προβλέψεων για τη κλάση 0.

Αντίθετα, για τη πλειοψηφική κλάση 1 (support 1.615.903), οι επιδόσεις είναι σχεδόν άριστες, δηλαδή η ακρίβεια είναι 0,9999, η ανάκληση 0,9957, και ο δείκτης F1 περίπου 0,9978. Από τον πίνακα της σύγχυσης της Εικόνας 6.1 προκύπτει ότι τα σφάλματα είναι περιορισμένα, δηλαδή 198 δείγματα της κλάσης 0 ταξινομήθηκαν ως 1, ενώ 7.009 δείγματα της κλάσης 1 ταξινομήθηκαν ως 0. Συνολικά, το μοντέλο Random Forest αποδίδει πολύ ισχυρά στο σενάριο χωρίς διαρροή πληροφορίας (non-leaky), με τη βασική δυσκολία να εντοπίζεται στη καθαρότητα των προβλέψεων της μειοψηφικής κλάσης, όπως είναι συνηθισμένο στα έντονα ανισόρροπα σύνολα.

6.2.2 Αποτελέσματα μοντέλου XGBoost χωρίς διαρροή (Non-Leaky XGBoost)

Στις ίδιες ακριβώς πειραματικές συνθήκες, δηλαδή στο σενάριο χωρίς διαρροή πληροφορίας (non-leaky), με 13 επιλεγμένα χαρακτηριστικά και εφαρμόζοντας χρονικό διαχωρισμό 80/20 (chrono split)

αξιολογήθηκε και το μοντέλο XGBoost, ώστε η σύγκριση με το Random Forest να βασίζεται στο ίδιο σύνολο ελέγχου. Τα αποτελέσματα συνοψίζονται στην Εικόνα 6.1.

Σε συνολικό επίπεδο, ο XGBoost επιτυγχάνει ακρίβεια (accuracy) ίση με 0,9965 και σταθμισμένο δείκτη F1 (weighted F1) 0,9965, δηλαδή πρακτικά ίδια συνολική εικόνα με το Random Forest. Η μακρόμεση επίδοση F1 (macro F1) είναι οριακά υψηλότερη 0,7758, κάτι που υποδηλώνει μια μικρή βελτίωση στη συμπεριφορά ως προς τη μειοψηφική κλάση.

Για τη μειοψηφική κλάση 0 (support 4.647) το μοντέλο του XGBoost εμφανίζει ακρίβεια των θετικών προβλέψεων (precision) 0,3896, ανάκληση (recall) 0,9572 και δείκτη F1 (F1-score) 0,5538. Η εικόνα αυτή παραμένει παρόμοια και για τη κλάση 0, δηλαδή έχει υψηλή ανάκληση αλλά χαμηλή ακρίβεια θετικών προβλέψεων. Ο πίνακας της σύγχυσης δείχνει 6.970 περιπτώσεις από 1→ 0 και 199 περιπτώσεις από 0→ 1. Σε σχέση με το Random Forest, οι περιπτώσεις από 1→ 0 είναι ελαφρώς λιγότερες, 6.970 έναντι 7.009, κάτι που συμβαδίζει με τη μακρόμεση επίδοση F1.

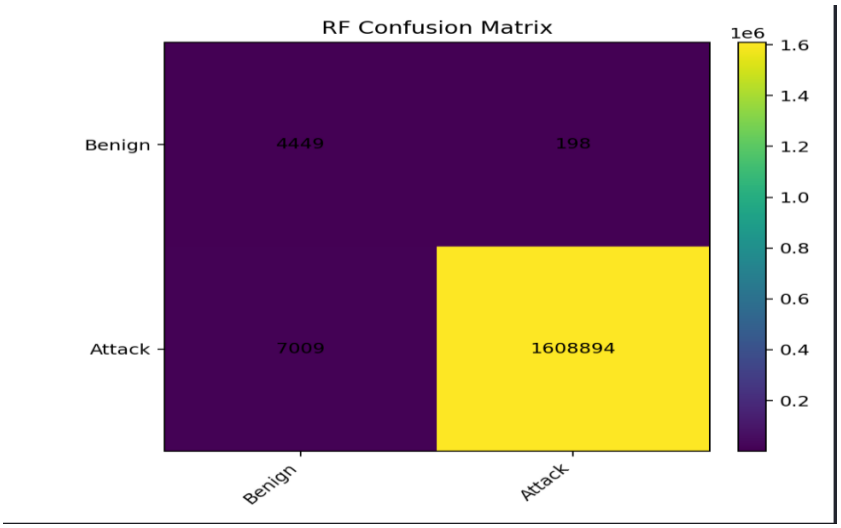
Για τη πλειοψηφική κλάση 1 (support 1.615.903), οι επιδόσεις παραμένουν εξαιρετικά υψηλές δηλαδή η ακρίβεια των θετικών προβλέψεων 0,9999, η ανάκληση 0,9957 και ο δείκτης F1 0,9978, επιβεβαιώνοντας ότι το μοντέλο XGBoost ταξινομεί πολύ αποτελεσματικά το κυρίαρχο τμήμα των δεδομένων, κάτι το οποίο φαίνεται στην Εικόνα 6.1. Συνολικά, στο σενάριο χωρίς διαρροή πληροφορίας (non-leaky), το μοντέλο XGBoost εμφανίζει σχεδόν ισοδύναμη συμπεριφορά με το Random Forest, με μία μικρή βελτίωση στα σφάλματα που σχετίζονται με τη μειοψηφική κλάση, η οποία θα αποτιμηθεί πιο καθαρά στη συγκριτική ανάλυση της επόμενης ενότητας.

6.3 Συγκριτική ανάλυση Random Forest έναντι XGBoost (RF vs XGB)

6.3.1 Πίνακες σύγχυσης (Confusion Matrices)

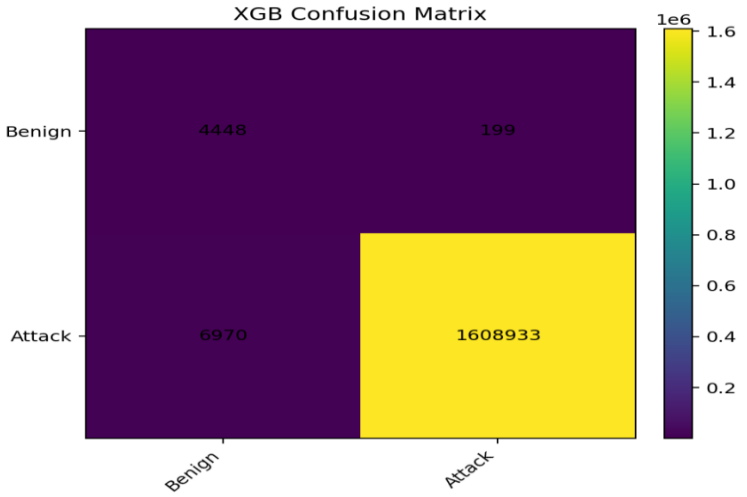
Αρχικά, για τη σύγκριση των μοντέλων Random Forest και XGBoost στο σενάριο χωρίς διαρροή πληροφορίας (non-leaky), χρησιμοποιήθηκαν οι πίνακες της σύγχυσης (confusion matrices). Οι πίνακες αυτοί δείχνουν καθαρά τι έκανε σωστά και τι λάθος το κάθε μοντέλο, δηλαδή ποια δείγματα ταξινομήθηκαν σωστά και σε ποιες περιπτώσεις μπερδεύτηκαν οι κλάσεις. Στο παρόν κεφάλαιο ακολουθείται η παρακάτω σύμβαση, δηλαδή οι γραμμές είναι ίσες με τη πραγματική κλάση (true label) και οι στήλες ίσες με τη προβλεπόμενη κλάση (predicted label), ώστε να είναι άμεσα ορατό το πώς κατανέμονται τα αποτελέσματα ανά κατηγορία.

Στην Εικόνα 6.2 παρουσιάζεται ο πίνακας σύγχυσης του Random Forest. Το μοντέλο ταξινομεί σωστά 4.449 δείγματα ως κανονική κίνηση (Benign), ενώ 198 φορές μια κανονική κίνηση σημειώνεται ως Επίθεση. Για τη κλάση Attack, ο Random Forest εντοπίζει 1.608.894 δείγματα ως επίθεση (Attack), αλλά υπάρχουν και 7.009 περιπτώσεις όπου μία επίθεση καταλήγει να προβλεφθεί σαν κανονική κίνηση (Benign). Με απλά λόγια, η συνολική εικόνα είναι πολύ καλή αλλά ένα μικρό μέρος των επιθέσεων ξεφεύγει και εμφανίζεται ως κανονική κίνηση.



Εικόνα 6.2: Πίνακας σύγχυσης RF σε non-leaky δεδομένα, με 13 χαρακτηριστικά

Στην Εικόνα 6.3 φαίνεται ο αντίστοιχος πίνακας σύγχυσης για το XGBoost. Η συμπεριφορά του είναι σχεδόν ίδια, καθώς 4.448 κανονικές κινήσεις (Benign) ταξινομούνται σωστά, 199 κανονικές κινήσεις χαρακτηρίζονται επιθέσεις, 1.608.933 επιθέσεις (Attack) ανιχνεύονται σωστά, ενώ 6.970 επιθέσεις προβλέπονται ως κανονικές κινήσεις (Benign). Σε σχέση με το Random Forest, ο XGBoost χάνει οριακά λιγότερες Επιθέσεις, 6.970 έναντι 7.009, με πρακτικά ίδιο επίπεδο ψευδών συναγερμών (false positives).



Εικόνα 6.3 : Πίνακας σύγχυσης XGB σε non-leaky δεδομένα, με 13 χαρακτηριστικά

Συνολικά, οι πίνακες της σύγχυσης δείχνουν ότι και τα δύο μοντέλα έχουν πολύ ισχυρή συμπεριφορά στο πλαίσιο του συνόλου δεδομένων χωρίς διαρροή (non-leaky). Όμως, αποδεικνύουν και κάτι σημαντικό για τη χρήση τους σε συστήματα Παρακολούθησης της Ασφάλειας, καθώς επειδή τα δεδομένα είναι έντονα ανισόρροπα ακόμη και ένα μικρό ποσοστό σφάλματος στη κλάση της επίθεσης (Attack) αντιστοιχεί σε χιλιάδες περιπτώσεις. Αυτές οι περιπτώσεις είναι επιχειρησιακά κρίσιμες, γιατί αντιπροσωπεύουν τις επιθέσεις που δεν θα εντοπιστούν. Έτσι, πέρα από τη συνολική ακρίβεια έχει πολύ μεγάλη σημασία γενικά να εξετάζονται προσεκτικά και οι δείκτες της ακρίβειας των θετικών

προβλέψεων (Precision), της ανάκλησης (Recall) και ο δείκτης F1 (F1-score), τα οποία αναλύονται στην επόμενη υποενότητα.

6.3.2 Ακρίβεια θετικών προβλέψεων - Ανάκληση - Δείκτης F1 (Precision – Recall – F1)

Στο κομμάτι αυτό παρουσιάζονται οι βασικές μετρικές της ταξινόμησης, δηλαδή η ακρίβεια των θετικών προβλέψεων (precision), η ανάκληση (recall) και δείκτης F1 (F1-score), για τα δύο μοντέλα το Random Forest και το XGBoost στο σενάριο χωρίς διαρροή πληροφορίας (non-leaky) με δεκατρία χαρακτηριστικά. Οι αντίστοιχες αναφορές της ταξινόμησης (classification reports) παρουσιάζονται στην Εικόνα 6.4 για το Random Forest και στην Εικόνα 6.5 για το XGBoost.

Για το μοντέλο Random Forest, στην Εικόνα 6.4 παρουσιάζεται η ακρίβεια η οποία είναι 0,9956 και ο σταθμισμένος δείκτης F1 ο οποίος είναι 0,996. Στο επίπεδο των κλάσεων, για τη κλάση της επίθεσης (Attack, 1) το μοντέλο εμφανίζει ακρίβεια θετικών προβλέψεων (precision) ίση με 0,9999, ανάκληση (recall) ίση με 0,9957 και δείκτης F1 (F1-score) 0,9978, το γεγονός αυτό δείχνει ότι οι προβλέψεις της κλάσης αυτής είναι ιδιαίτερα αξιόπιστες και ότι χάνονται λίγες επιθέσεις. Για τη κλάση της κανονικής κίνησης (Benign, 0), η ανάκληση είναι πάλι υψηλή και ίση με 0,9574, όμως η ακρίβεια των θετικών προβλέψεων είναι αρκετά χαμηλότερη 0,3883 και ο δείκτης F1 είναι 0,5525. Γενικότερα, το μοντέλο βρίσκει τα περισσότερα πραγματικά δείγματα της κανονικής κίνησης αλλά όταν τα προβλέπει ένα σημαντικό μέρος αυτών των προβλέψεων τελικά αφορά την άλλη κλάση. Η συμπεριφορά αυτή είναι αναμενόμενη στα έντονα ανισόρροπα δεδομένα όπου οι συνολικές σταθμισμένες μετρικές επηρεάζονται κατά κύριο λόγο από τη πλειοψηφική κλάση.

```
┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ sed -n '1,120p' reports_nonleaky_13/rf_report.txt

non_leaky_rf_13
F1=0.997765

              precision    recall  f1-score   support

           0     0.3883    0.9574    0.5525      4647
           1     0.9999    0.9957    0.9978   1615903

    accuracy                         0.9956   1620550
   macro avg     0.6941    0.9765    0.7751   1620550
weighted avg     0.9981    0.9956    0.9965   1620550
```

Εικόνα 6.4: Αναφορά ταξινόμησης, precision–recall–F1 για τον RF σε non-leaky δεδομένα, με 13 χαρακτηριστικά

Αντίστοιχα, για το μοντέλο XGBoost η συνολική εικόνα η οποία παρουσιάζεται στην Εικόνα 6.5 είναι πρακτικά ίδια. Δηλαδή, η ακρίβεια είναι 0,9959 και ο σταθμισμένος δείκτης F1 είναι 0,9965. Για τη κλάση της επίθεσης (Attack, 1) καταγράφονται ξανά ακρίβεια θετικών προβλέψεων (precision) ίση με 0,9999, ανάκληση (recall) ίση με 0,9957 και δείκτης F1 (F1-score) 0,9978, δηλαδή ίδια συμπεριφορά με το μοντέλο Random Forest. Βέβαια, στη κλάση της κανονικής κίνησης (Benign, 0) το μοντέλο XGBoost εμφανίζει λίγο καλύτερη εικόνα με ακρίβεια θετικών προβλέψεων ίση με 0,3896 και δείκτης F1 ίσο με 0,5538. Η διαφορά είναι μικρή αλλά δείχνει μια ελάχιστα πιο καθαρή απόδοση στις προβλέψεις της κανονικής ροής.

```
┌──(venv-ml)─(xariskotsis㉿kali)-[~/opensearch-clean/migrate-v3]
└─$ sed -n '1,120p' reports_nonleaky_13/xgb_report.txt

non_leaky_xgb_13
F1=0.997777

              precision    recall  f1-score   support

           0     0.3896    0.9572    0.5538      4647
           1     0.9999    0.9957    0.9978   1615903

    accuracy                         0.9956   1620550
   macro avg     0.6947    0.9764    0.7758   1620550
weighted avg     0.9981    0.9956    0.9965   1620550
```

Εικόνα 6.5: Αναφορά ταξινόμησης, precision–recall–F1 για τον XGB σε non-leaky δεδομένα, με 13 χαρακτηριστικά

Τέλος, έχει σημασία να διαχωριστεί η ερμηνεία των μακρομεσαίων όρων (macro averages) και των σταθμισμένων μέσων όρων (weighted averages). Οι τιμές του μακρομεσαίου όρου δίνουν ίσο βάρος σε κάθε κλάση και επομένως δείχνουν τη δυσκολία της μικρότερης κλάσης. Αντίθετα, οι σταθμισμένες τιμές επηρεάζονται πιο πολύ από τη πλειοψηφική κλάση, επειδή οι συνεισφορές των κλάσεων υπολογίζονται ανάλογα με το πλήθος των δειγμάτων τους. Για την επιχειρησιακή ερμηνεία στο πλαίσιο του συστήματος Παρακολούθησης της Ασφάλειας, είναι χρήσιμο να εξετάζονται παράλληλα, οι δείκτες της κλάσης της επίθεσης (Attack), με έμφαση στην ανάκληση (recall) λόγω του κόστους των μη ανιχνευμένων επιθέσεων με τους δείκτες του μακρομεσαίου όρου (macro F1) ως μια πιο ισορροπημένη συνολική ένδειξη της συμπεριφοράς του μοντέλου.

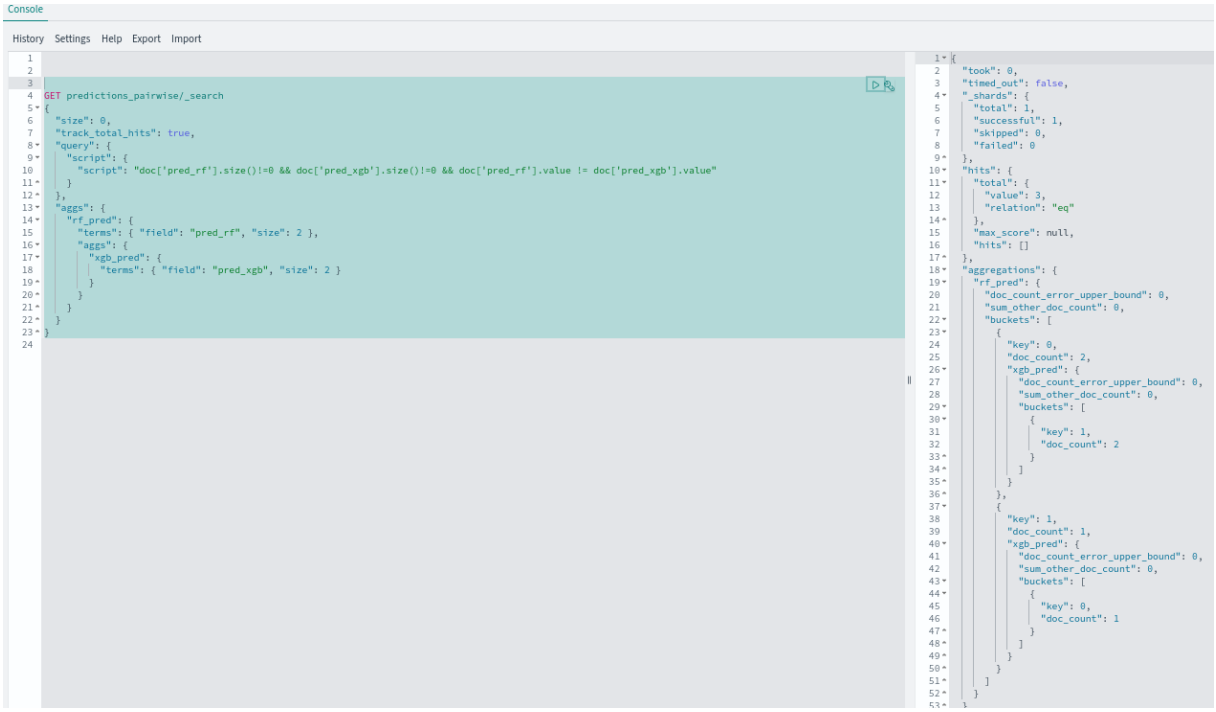
6.3.3 Ανάλυση διαφωνιών (Disagreement Analysis)

Στη παρούσα υποενότητα εξετάζεται το πόσο συχνά τα δύο μοντέλα, Random Forest και XGBoost καταλήγουν σε διαφορετική πρόβλεψη για την ίδια εγγραφή στον δείκτη predictions_pairwise. Ως διαφωνία ορίζεται η περίπτωση όπου υπάρχουν και οι δυο προβλέψεις, δηλαδή τα πεδία pred_rf και pred_xgb και οι τιμές τους δεν είναι ίδιες.

Για τον εντοπισμό των διαφωνιών εκτελέστηκε ένα ερώτημα με φίλτρο σεναρίου (script filter), το οποίο ελέγχει ότι τα δύο πεδία είναι διαθέσιμα και ότι το pred_rf είναι διαφορετικό του pred_xgb. Στη συνέχεια, με τη χρήση των συναθροίσεων (aggregations) υπολογίστηκε το πλήθος των εγγραφών που εξετάστηκαν και το πλήθος των εγγραφών που ικανοποιούν τη συνθήκη της διαφωνίας. Όπως φαίνεται και στην Εικόνα 6.6, στο εξεταζόμενο δείγμα, δηλαδή με N ίσο με 100.005 εγγραφές, εντοπίστηκαν μόλις τρεις περιπτώσεις διαφωνίας, το οποίο αποδεικνύει ότι τα μοντέλα έχουν πολύ υψηλή συμφωνία στη συγκεκριμένη ροή των δεδομένων.

Εικόνα 6.6: Υπολογισμός πλήθους εγγραφών και διαφωνιών, pred_rf ≠ pred_xgb στον δείκτη predictions_pairwise μέσω script filter και aggregations

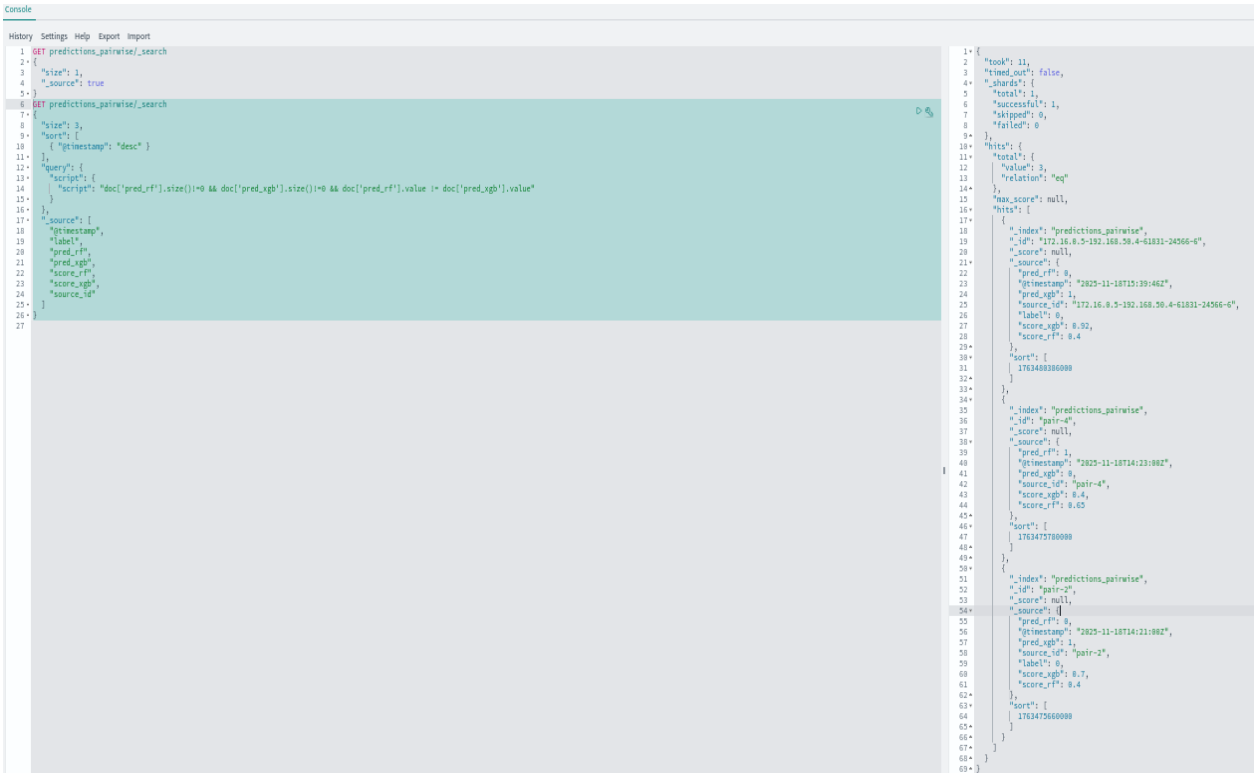
Για να αποτυπωθεί καλύτερα προς ποια κατεύθυνση εμφανίζονται οι διαφωνίες, ακολουθήθηκε ανάλυση με ομαδοποίηση των περιπτώσεων με βάσει τη τιμή του pred_rf και μέσα σε κάθε ομάδα με βάση τη τιμή του pred_xgb. Από την Εικόνα 6.7 προκύπτει ότι σε δύο περιπτώσεις ο Random Forest έδωσε 0, δηλαδή ότι είναι κανονική κίνηση (Benign) ενώ ο XGBoost έδωσε 1, δηλαδή ότι είναι επίθεση (Attack) και σε μία περίπτωση ο Random Forest έδωσε 1, ενώ ο XGBoost 0.



Εικόνα 6.7: Κατανομή διαφωνιών ανά συνδυασμό προβλέψεων, pred_rf → pred_xgb στον δείκτη predictions_pairwise

Τέλος, ανακτήθηκαν οι πιο πρόσφατες εγγραφές οι οποίες παρουσιάζουν διαφωνία, με ταξινόμηση ως προς το @timestamp και με εμφάνιση των βασικών πεδίων της σύγχυσης, δηλαδή τα, pred_rf, pred_xgb, label, score_rf, score_xgb, source_id. Στην Εικόνα 6.8, φαίνεται ότι στις δύο περιπτώσεις όπου το Random Forest είναι ίσο με 0 και το XGBoost είναι ίσο με 1, η πραγματική ετικέτα (label) είναι 0 δηλαδή κανονική κίνηση (Benign), κάτι το οποίο δηλώνει μεμονωμένα ψευδώς θετικά αποτελέσματα

(false positives) του μοντέλου XGBoost στο συγκεκριμένο δείγμα. Αντίστοιχα, στην περίπτωση όπου το Random Forest είναι ίσο με 1 και το XGBoost είναι ίσο με 0, η πραγματική ετικέτα (label) είναι 1 δηλαδή επίθεση (Attack), άρα πρόκειται για ψευδώς αρνητικό αποτέλεσμα (false negative) του μοντέλου XGBoost. Επιπλέον, οι τιμές των βαθμολογιών (scores) διαφοροποιούνται αισθητά, το οποίο δείχνει ότι τα δύο μοντέλα δεν βαθμονομούν με τον ίδιο τρόπο την εμπιστοσύνη τους και μπορεί να αντιδρούν διαφορετικά σε οριακά παραδείγματα.



Εικόνα 6.8: Ενδεικτικές πρόσφατες εγγραφές διαφωνίας, ταξινομημένες κατά @timestamp στον δείκτη predictions_pairwise

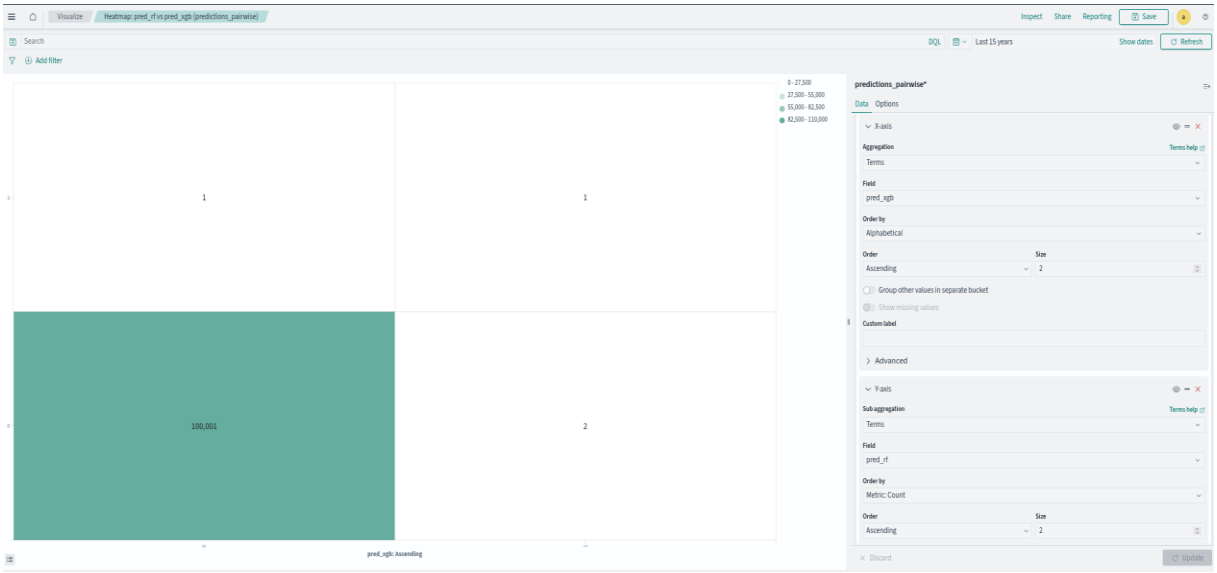
Συνολικά, η ανάλυση των διαφωνιών επιβεβαιώνει ότι οι δύο ταξινομητές δίνουν τις περισσότερες φορές την ίδια απόφαση στο συγκεκριμένο σύνολο. Παράλληλα, οι σπάνιες αποκλίσεις τους έχουν πρακτική αξία, επειδή μπορούν να λειτουργήσουν ως ενδείξεις της αβεβαιότητας (uncertainty flags) για κάποια στοχευμένη διερεύνηση στη κονσόλα της αναζήτησης (Discover) και για κάποιο ειδικό κανόνα της ειδοποίησης όταν τα μοντέλα δεν συμφωνούν.

6.3.4 Θερμικοί χάρτες και ζευγαρωμένα διαγράμματα (Heatmaps & Pairwise Plots)

Επιπρόσθετα, για να αποτυπωθεί με απλό και γρήγορο τρόπο η σχέση των δύο προβλέψεων των δύο ταξινομητών στην ίδια εγγραφή, δηλαδή ανά ροή (per-flow) χρησιμοποιήθηκε ένας θερμικός χάρτης 2x2 (heatmap), ο οποίος δείχνει πόσες φορές εμφανίζεται κάθε συνδυασμός των ζευγαρωμένων προβλέψεων pred_rf (Random Forest) και pred_xgb (XGBoost) στον δείκτη predictions_pairwise. Στο διάγραμμα αυτό, ο οριζόντιος άξονας αντιστοιχεί στις προβλέψεις του XGBoost και ο κατακόρυφος στις προβλέψεις του Random Forest, ενώ κάθε κελί απεικονίζει το πλήθος των εγγραφών που κατέληξαν στον αντίστοιχο συνδυασμό.

Γενικά, όπως φαίνεται και στην Εικόνα 6.9 οι περισσότερες παρατηρήσεις συγκεντρώνονται στη διαγώνιο του θερμικού χάρτη (heatmap), κάτι το οποίο σημαίνει ότι τα δύο μοντέλα καταλήγουν συνήθως στην ίδια απόφαση. Με άλλα λόγια, το κελί (0,0) συγκεντρώνει τις περιπτώσεις όπου και οι

δύο ταξινομητές προβλέπουν τη κανονική κίνηση (Benign), ενώ το κελί (1,1) τις περιπτώσεις όπου και οι δύο καταλήγουν σε επίθεση (Attack). Αντίθετα, τα κελιά που βρίσκονται εκτός της διαγωνίου (0,1) και (1,0) αντιστοιχούν στις περιπτώσεις της απόκλισης (disagreement), δηλαδή όταν το ένα μοντέλο δίνει κανονική κίνηση και το άλλο δίνει επίθεση. Στο εξεταζόμενο δείγμα, αυτές οι αποκλίσεις είναι ελάχιστες και συγκεκριμένα τρεις εγγραφές, γεγονός που συνδέεται και με τα ευρήματα της προηγούμενης ενότητας.



Εικόνα 6.9: Θερμικός χάρτης 2×2 των ζευγαρωμένων προβλέψεων pred_rf έναντι pred_xgb στον δείκτη predictions_pairwise

Συνολικά, ο θερμικός χάρτης λειτουργεί ως μία συνοπτική ζευγαρωμένη (pairwise) απεικόνιση για τις δυαδικές προβλέψεις, καθώς επιτρέπει να φανεί γρήγορα η κυρίαρχη εικόνα της συμφωνίας, να ξεχωρίσουν οι σπάνιες περιπτώσεις της διαφωνίας και να προσδιοριστεί το μέγεθος των αποκλίσεων, που σε ένα επιχειρησιακό σενάριο Παρακολούθησης της Ασφάλειας αποτελούν σημεία με πολύ ενδιαφέρον για στοχευμένη διερεύνηση.

6.3.5 Συγκριτική αξιολόγηση έναντι παραδοσιακών στατιστικών ανιχνευτών αναφοράς (Z-Score & MAD)

Αρχικά, για να τεκμηριωθεί εμπειρικά η διαφορά μεταξύ των μοντέλων της Μηχανικής Μάθησης και των παραδοσιακών προσεγγίσεων της ανίχνευσης, αξιολογήθηκαν δύο στατιστικοί ανιχνευτές αναφοράς (traditional baselines) χωρίς επίβλεψη, ένας ανιχνευτής Z-score και ένας ανιχνευτής Robust MAD (Median Absolute Deviation), όπως παρουσιάζεται συνοπτικά στην Εικόνα 6.10. Η αξιολόγηση πραγματοποιήθηκε στο ίδιο πλαίσιο χωρίς διαρροή πληροφορίας (non-leaky) με τα Random Forest και XGBoost, χρησιμοποιώντας την ίδια λίστα δεκατριών χαρακτηριστικών και χρονολογικό διαχωρισμό 80/20 (chronological split), ώστε παραμένει δίκαιη η σύγκριση κάτω από κοινές συνθήκες.

```
source venv-ml/bin/activate

python traditional_baselines_13.py | tee /tmp/traditional_baselines_13.txt

[INFO] Traditional baselines on non-leaky 13 features
[INFO] split: chronological 80/20
[INFO] features(13): ['_Active_Std', 'Idle_Mean', '_Fwd_IAT_Min', '_PSH_Flag_Count', '_Idle_Std
', 'Bwd_Packet_Length_Max', '_Fwd_Packet_Length_Max', '_Bwd_Avg_Packets_Bulk', '_Bwd_Packets_s'
, '_Init_Win_bytes_backward', '_Fwd_Avg_Bulk_Rate', '_Subflow_Fwd_Bytes', '_Bwd_PSH_Flags']
[INFO] total rows=8102747 | train=6482197 test=1620550
[INFO] label convention assumed: 0=benign, 1=attack

[Z-SCORE] threshold = 82.551132 (99th percentile of TRAIN benign)

=== BASELINE 1: Z-score anomaly detector ===
Confusion matrix [ [tn, fp], [fn, tp] ]:
[[   4622      25]
 [1615877      26]]

Classification report:
              precision    recall  f1-score   support

           0     0.0029    0.9946    0.0057      4647
           1     0.5098    0.0000    0.0000   1615903

    accuracy                         0.0029   1620550
   macro avg     0.2563    0.4973    0.0029   1620550
weighted avg     0.5084    0.0029    0.0000   1620550


Summary (positive class = 1 = attack):
accuracy=0.002868 | precision=0.509804 | recall=0.000016 | f1=0.000032

[MAD] threshold = 39601164.000000 (99th percentile of TRAIN benign)

=== BASELINE 2: Robust MAD anomaly detector ===
Confusion matrix [ [tn, fp], [fn, tp] ]:
[[   4594      53]
 [1615891      12]]

Classification report:
              precision    recall  f1-score   support

           0     0.0028    0.9886    0.0057      4647
           1     0.1846    0.0000    0.0000   1615903

    accuracy                         0.0028   1620550
   macro avg     0.0937    0.4943    0.0028   1620550
weighted avg     0.1841    0.0028    0.0000   1620550


Summary (positive class = 1 = attack):
accuracy=0.002842 | precision=0.184615 | recall=0.000007 | f1=0.000015
```

Εικόνα 6.10: Αποτελέσματα παραδοσιακών στατιστικών ανιχνευτών αναφοράς, Z-score και Robust MAD σε non-leaky πλαίσιο, με 13 χαρακτηριστικά

Ο πρώτος ανιχνευτής (Z-score) εκτιμά στο σύνολο της εκπαίδευσης το μέσο όρο και τη τυπική απόκλιση ανά χαρακτηριστικό και για κάθε εγγραφή στο σύνολο ελέγχου υπολογίζει το Z-score ανά χαρακτηριστικό. Ο τελικός δείκτης της ανωμαλίας ορίζεται ως η μέγιστη απόλυτη απόκλιση μεταξύ των χαρακτηριστικών, δηλαδή ως το μέγιστο απόλυτο z στις δεκατρείς διαστάσεις. Το κατώφλι δεν καθορίζεται αυθαίρετα, αλλά υπολογίζεται αποκλειστικά από τα δείγματα της κανονικής κίνησης (benign) του συνόλου εκπαίδευσης (training set), ίσο με το ενενηκοστό ένατο εκατοστημόριο (99th percentile) του δείκτη της ανωμαλίας (anomaly score), με στόχο τον έλεγχο των ψευδών θετικών ειδοποιήσεων (false positives), χωρίς χρήση των πληροφοριών του συνόλου ελέγχου (test set).

Επιπρόσθετα, στο σύνολο ελέγχου με κατώφλι Z-score ίσο με 82.551132 ο ανιχνευτής παρήγαγε αληθώς αρνητικά αποτελέσματα (True Negatives, TN) ίσα με 4.622 και ψευδώς θετικά (False Positives, FP) ίσα με 25, ενώ κατέγραψε ψευδώς αρνητικά (False Negatives, FN) 1.615.877 και αληθώς θετικά (True Positives, TP) ίσα με 26, όπως αποτυπώνεται και στην Εικόνα 6.10. Η εικόνα αυτή δείχνει ότι οι ψευδώς θετικές ανιχνεύσεις παραμένουν περιορισμένες, ωστόσο οι ψευδώς αρνητικές είναι πολύ αυξημένες, δηλαδή η μέθοδος αποτυγχάνει να εντοπίσει σχεδόν το σύνολο των επιθέσεων. Αντίστοιχα, για τη κλάση της επίθεσης (Attack=1) η ανάκληση (recall) είναι πραγματικά μηδενική, 0,000016 και ο δείκτης F1 (F1-score) αμελητέος, 0,000032, με συνολική ακρίβεια (accuracy) 0,002868.

Ο δεύτερος ανιχνευτής (Robust MAD) ακολουθεί μία αντίστοιχη λογική ανίχνευσης των ανωμαλιών αλλά χρησιμοποιεί τις ανθεκτικές στατιστικές εκτιμήσεις (robust statistics) της κεντρικής τάσης και διασποράς, δηλαδή τη διάμεσο (median) και τη MAD, ώστε να μειώνεται η ευαισθησία στις ακραίες τιμές (outliers) και στο θόρυβο (noise). Και εδώ, το κατώφλι επιλέγεται αποκλειστικά από τα δείγματα

της κανονικής κίνησης του συνόλου εκπαίδευσης, ως το ενενηκοστό ένατο εκατοστημόριο (99th percentile) του αντίστοιχου δείκτη της ανωμαλίας, ώστε η διαδικασία της ρύθμισης να παραμένει χωρίς διαρροή (non-leaky).

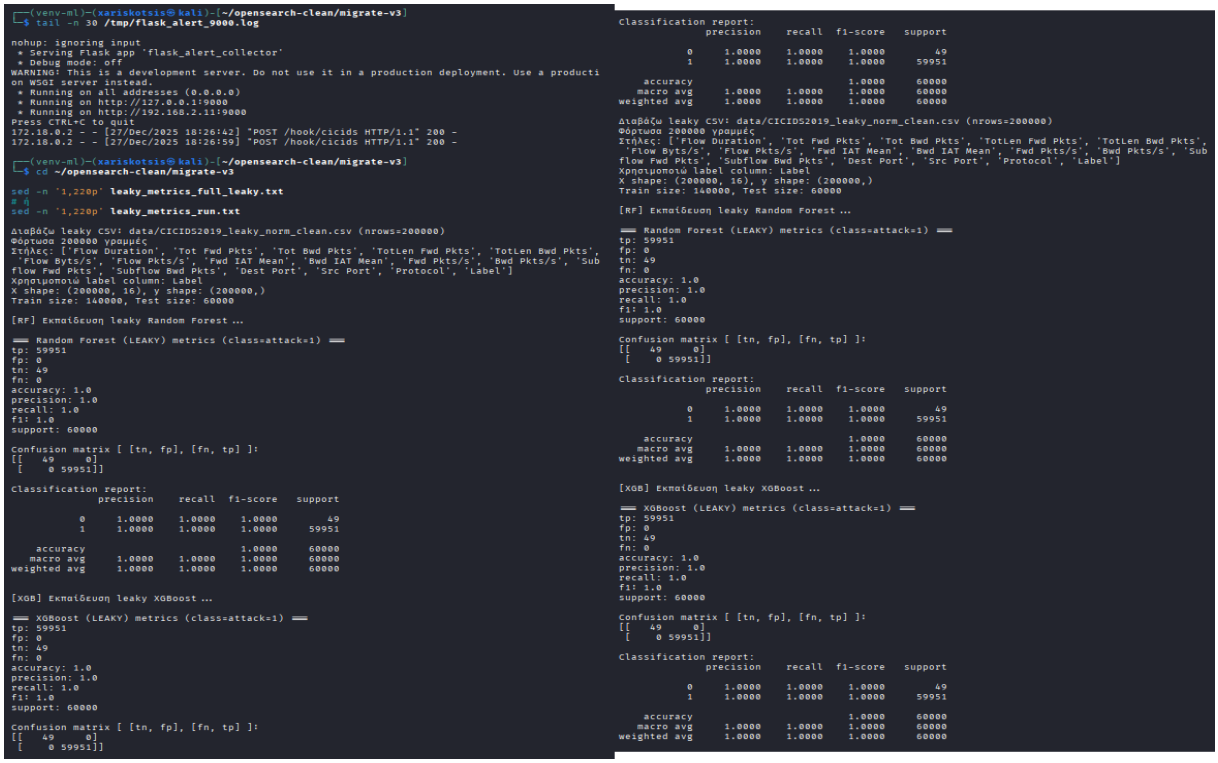
Στο σύνολο ελέγχου, με το κατώφλι MAD ίσο με 39.601.164, ο ανιχνευτής παρήγαγε αληθώς αρνητικά αποτελέσματα (TN) ίσα με 4.594 και ψευδώς θετικά (FP) ίσα με 53, ενώ κατέγραψε ψευδώς αρνητικά αποτελέσματα (FN) ίσα με 1.615.891 και αληθώς θετικά (TP) ίσα με 12, όπως φαίνεται καθαρά και στην Εικόνα 6.10. Και σε αυτή τη περίπτωση, η συμπεριφορά χαρακτηρίζεται από πολύ υψηλές ψευδώς αρνητικές προβλέψεις με ανάκληση (recall) για τη κλάση της επίθεσης, 0,000007 και δείκτη F1 (F1-score) 0,000015, ενώ η συνολική ακρίβεια (accuracy) είναι 0,002842.

Συνολικά, και οι δύο παραδοσιακοί στατιστικοί ανιχνευτές (statistical detectors) εμφανίζουν πολύ χαμηλότερη επιχειρησιακή απόδοση σε σχέση με τα μοντέλα Random Forest και XGBoost, κυρίως λόγω της σχεδόν μηδενικής ικανότητας εντοπισμού της κακόβουλης κλάσης. Το εύρημα είναι συμβατό με τους περιορισμούς των απλών μηχανισμών με βάση τα κατώφλια (threshold-based mechanisms) στις σύνθετες μορφές επιθέσεων DDoS, καθώς και σε δεδομένα όπου η επιθετική συμπεριφορά δεν εκδηλώνεται απαραίτητα ως ακραία απόκλιση σε ένα μόνο χαρακτηριστικό, αλλά ως συνδυαστικό μοτίβο σε πολλαπλές διαστάσεις. Συνεπώς, οι στατιστικές μέθοδοι μπορούν να λειτουργήσουν ως ένα σημείο αναφοράς χαμηλού κόστους, αλλά δεν επαρκούν μόνες τους για να υπάρξει αξιόπιστη και κλιμακούμενη ανίχνευση στο πλαίσιο του αγωγού του συστήματος Παρακολούθησης της Ασφάλειας (CM pipeline), γεγονός το οποίο δικαιολογεί την χρήση των προσεγγίσεων της Μηχανικής Μάθησης.

6.4 Σύγκριση δεδομένων με διαρροή και χωρίς διαρροή πληροφορίας (Leaky vs Non-Leaky)

Στην συγκεκριμένη ενότητα, συγκρίνεται η συμπεριφορά των μοντέλων στο σύνολο δεδομένων χωρίς διαρροή (non-leaky) και σε ένα σκόπιμα προβληματικό σύνολο δεδομένων με διαρροή (leaky), ώστε να φανεί στη πράξη πως η διαρροή της πληροφορίας (data leakage) μπορεί να αλλοιώσει τις μετρικές της αξιολόγησης. Η διάκριση αυτή είναι πολύ σημαντική στα συστήματα Παρακολούθησης της Ασφάλειας, γιατί μια πολύ υψηλή επίδοση έχει νόημα μόνο όταν προκύπτει από χαρακτηριστικά που είναι διαθέσιμα τη στιγμή που γίνεται η πρόβλεψη χωρίς να προδίδουν άμεσα η έμμεσα τη πραγματική ετικέτα της κλάσης.

Επιπρόσθετα, όπως φαίνεται και στην Εικόνα 6.11, στο σύνολο των δεδομένων με διαρροή (leaky) ο αλγόριθμος Random Forest και ο αλγόριθμος XGBoost εμφανίζουν τέλεια αποτελέσματα, δηλαδή η ακρίβεια, η ακρίβεια των θετικών προβλέψεων, η ανάκληση και ο δείκτης F1 είναι όλα ίσα με 1,0 και οι πίνακες σύγχυσης δεν εμφανίζουν κανένα σφάλμα. Αυτή η εικόνα είναι ένα κλασικό σημάδι έντονης διαρροής δεδομένων, διότι ουσιαστικά το πρόβλημα έχει γίνει τεχνητά εύκολο, επειδή κάποια χαρακτηριστικά σχετίζονται πολύ στενά με τη μεταβλητή-στόχο (label). Έτσι, οι μετρικές δεν αποτυπώνουν τη πραγματική ικανότητα της γενίκευσης αλλά την επίδοση που στηρίζεται στη πληροφορία η οποία δεν θα έπρεπε να θεωρείται διαθέσιμη σε πραγματικές συνθήκες λειτουργίας.



Εικόνα 6.11: Αποτελέσματα αξιολόγησης σε leaky σύνολο δεδομένων για RF και XGB

Αντίθετα, όπως φαίνεται και στην Εικόνα 6.12, στο σύνολο δεδομένων χωρίς διαρροή (non-leaky) η αξιολόγηση δίνει μία πιο ρεαλιστική εικόνα για το πώς λειτουργούν τα μοντέλα όταν περιορίζονται σε χαρακτηριστικά με χαμηλό κίνδυνο διαρροής δεδομένων. Παρότι οι συνολικές επιδόσεις είναι πάλι υψηλές, αυτό που έχει τη μεγαλύτερη σημασία είναι η κατανομή των σφαλμάτων και οι μετρικές ανά κλάση, ειδικά όταν τα δεδομένα είναι έντονα ανισόρροπα. Με βάση αυτά, γίνεται σαφές ότι για να χρησιμοποιηθεί κάτι σε έναν αγωγό ενός συστήματος Παρακολούθησης της Ασφάλειας (CM pipeline) το σημείο της αναφοράς πρέπει να είναι το σύνολο των δεδομένων χωρίς διαρροή, ενώ τα αποτελέσματα του συνόλου δεδομένων με διαρροή αξιοποιούνται ως αντιπαράδειγμα, ώστε να αποφεύγονται τα συμπεράσματα που φαίνονται εντυπωσιακά αλλά στη πράξη δεν είναι αξιόπιστα.

```
python train_non_leaky_13.py 2>&1 | tee /tmp/nonleaky_baseline_for_6_4.txt

[INFO] chrono split loading…
[INFO] total rows=8102747 | train=6482197 test=1620550

==== RF (non-leaky, 13 feats) ====
F1= 0.9977652678531285
              precision    recall  f1-score   support

           0     0.3883    0.9574    0.5525      4647
           1     0.9999    0.9957    0.9978   1615903

    accuracy                         0.9956   1620550
   macro avg     0.6941    0.9765    0.7751   1620550
weighted avg     0.9981    0.9956    0.9965   1620550

Confusion Matrix (RF) [rows=true, cols=pred]:
[[   4449     198]
 [   7009 1608894]]

==== XGB (non-leaky, 13 feats) ====
F1= 0.9977770783882965
              precision    recall  f1-score   support

           0     0.3896    0.9572    0.5538      4647
           1     0.9999    0.9957    0.9978   1615903

    accuracy                         0.9956   1620550
   macro avg     0.6947    0.9764    0.7758   1620550
weighted avg     0.9981    0.9956    0.9965   1620550

Confusion Matrix (XGB) [rows=true, cols=pred]:
[[   4448     199]
 [   6970 1608933]]

[OK] Saved:
 - reports: reports_nonleaky_13
 - ONNX RF: models/rf_13_nonleaky.onnx
 - ONNX XGB: models/xgb_13_nonleaky.onnx
 - feature schema: models/feature_schema_13.json
```

Εικόνα 6.12: Αποτελέσματα αξιολόγησης σε non-leaky σύνολο δεδομένων για RF και XGB

6.5 Συζήτηση και ερμηνεία των αποτελεσμάτων

Αρχικά, η ενότητα συνοψίζει και ερμηνεύει τα βασικά ευρήματα των προηγούμενων ενοτήτων, για να γίνει πιο ξεκάθαρο τι σημαίνουν τα αποτελέσματα που βρέθηκαν για ένα επιχειρησιακό σύστημα Παρακολούθησης της Ασφάλειας (CM). Η συζήτηση στηρίζεται κυρίως στα σενάρια χωρίς διαρροή, στη σύγκριση των μοντέλων Randon Forest και XGBoost, αλλά και στην αντιπαραβολή με το σενάριο της διαρροής, ώστε τα συμπεράσματα να μην μένουν μόνο στους αριθμούς αλλά και να συνδέονται με τη λειτουργία ενός αγωγού Παρακολούθησης της Ασφάλειας.

Πρώτα από όλα, είναι σημαντικό να ληφθεί υπόψη ότι το πρόβλημα της ταξινόμησης εξετάζεται σε έντονα ανισόρροπα δεδομένα. Αυτό πρακτικά σημαίνει ότι μια υψηλή συνολική ακρίβεια (accuracy) ή ένας υψηλός σταθμισμένος όρος (weighted average) μπορεί να δίνει μια πολύ καλή συνολική εικόνα ακόμη κι αν το μοντέλο δυσκολεύεται περισσότερο σε μία από τις δύο κλάσεις. Για αυτό το λόγο, η ερμηνεία δεν πρέπει να περιορίζεται και να σταματά στις συνολικές μετρικές, είναι σημαντικό να αναλύονται οι πίνακες της σύγχυσης (confusion matrix) αλλά και οι τιμές της ακρίβειας των θετικών προβλέψεων (precision) και της ανάκλησης (recall) για κάθε κλάση, για να φαίνεται καθαρά ποια σφάλματα κάνει το κάθε μοντέλο και πόσο συχνά. Στο περιβάλλον της κυβερνοασφάλειας αυτό έχει άμεση σημασία, διότι διαφορετικά σφάλματα έχουν και διαφορετικό κόστος, για παράδειγμα οι μη ανιχνεύσιμες επιθέσεις (false negatives) είναι πιο ακριβές επιχειρησιακά, ενώ οι ψευδείς συναγερμοί (false positives) αυξάνουν το θόρυβο και επιβαρύνουν τη διερεύνηση.

Στο σύνολο των δεδομένων χωρίς διαρροή, τα αποτελέσματα δείχνουν ότι και τα δύο μοντέλα έχουν πολύ υψηλή απόδοση στο σύνολο του ελέγχου. Αυτό δείχνει ότι το επιλεγμένο σύνολο των

χαρακτηριστικών, σε συνδυασμό με τη ροή της προεπεξεργασίας, παρέχει τη πληροφορία που χρειάζεται ώστε να διακρίνεται αποτελεσματικά η κανονική από τη κακόβουλη κίνηση. Παράλληλα, η ανάλυση ανά κλάση δείχνει πόσο δύσκολα είναι τα ανισόρροπα σύνολα, καθώς η μικρή ή πιο δύσκολη κλάση είναι εκείνη στην οποία φαίνονται πιο καθαρά οι περιορισμοί. Με άλλα λόγια, η υψηλή συνολική εικόνα δεν σημαίνει ότι όλα είναι εξίσου εύκολα, απλώς δείχνει ότι συνολικά το σύστημα λειτουργεί πολύ καλά, ενώ ουσιαστικά για να βγάλουμε ένα αποτέλεσμα πρέπει να δούμε σε ποια σημεία σκοντάφτει το μοντέλο, δηλαδή ποια λάθη κάνει, πόσο συχνά τα κάνει και σε ποία κλάση εμφανίζονται.

Στο επίπεδο της σύγκρισης των δύο μοντέλων, Random Forest και XGBoost, η γενική εικόνα είναι ότι τα δύο μοντέλα συμπεριφέρονται σχεδόν ισοδύναμα όταν αξιοποιούνται στις ίδιες συνθήκες, δηλαδή με το ίδιο σύνολο χαρακτηριστικών και το ίδιο σύνολο ελέγχου. Αυτό είναι ένα πολύ χρήσιμο συμπέρασμα, γιατί δείχνει ότι το αποτέλεσμα που παίρνουμε δεν εξαρτάται από ένα αλγόριθμο μόνο και από τα δύο μοντέλα προκύπτει παρόμοια συμπεριφορά. Βέβαια, οι μικρές διαφορές που εμφανίζονται στα επιμέρους σφάλματα ή στους δείκτες ανά κλάση δεν είναι αμελητέες, καθώς συνήθως αντιστοιχούν σε περιπτώσεις οι οποίες είναι οριακές και δεν ταιριάζουν απόλυτα με τα μοτίβα που έχει μάθει το κάθε μοντέλο. Σε ένα πλαίσιο Παρακολούθησης της Ασφάλειας, αυτά τα οριακά δείγματα είναι συχνά και τα πιο ενδιαφέροντα, επειδή είτε αποτελούν θόρυβο και ιδιαιτερότητες της κίνησης είτε δείχνουν συμπεριφορές που βρίσκονται κοντά στο όριο μεταξύ κακόβουλης και κανονικής κίνησης.

Ακόμη, η ανάλυση των διαφωνιών (disagreement analysis) έρχεται να δώσει ακριβώς αυτή τη πρακτική διάσταση. Όταν δύο ταξινομητές δεν συμφωνούν για την ίδια εγγραφή, η συγκεκριμένη εγγραφή αποκτά αυξημένη αξία για διερεύνηση. Επιχειρησιακά, οι διαφωνίες μπορούν να λειτουργήσουν σαν ένα σήμα αβεβαιότητας, δηλαδή ως ένδειξη ότι το περιστατικό αξίζει περισσότερη προσοχή. Επιπλέον, αν σε βάθος χρόνου αυξηθεί η συχνότητα των διαφωνιών, αυτό μπορεί να αποτελεί ένδειξη ότι η ροή των δεδομένων αλλάζει (drift) και ότι το σύστημα συναντά μοτίβα διαφορετικά από εκείνα στα οποία εκπαιδεύτηκε.

Η σύγκριση των δεδομένων με διαρροή (leaky) με αυτά χωρίς διαρροή (non-leaky) αναδεικνύει ένα από τα πιο ουσιαστικά συμπεράσματα. Δηλαδή, ότι η διαρροή της πληροφορίας μπορεί να δημιουργήσει μια εικόνα τέλειας απόδοσης που στη πραγματικότητα δεν θα μεταφερθεί στις πραγματικές συνθήκες λειτουργίας. Καθώς, όταν στο μοντέλο περνούν χαρακτηριστικά τα οποία κουβαλούν άμεσα ή έμμεσα πληροφορία της ετικέτας ή πληροφορία που δεν θα είναι διαθέσιμη τη στιγμή της πρόβλεψης, τότε το πρόβλημα γίνεται τεχνητά εύκολο και οι μετρικές διογκώνονται. Αυτό είναι πολύ κρίσιμο για τις λύσεις των συστημάτων Παρακολούθησης της Ασφάλειας, γιατί ένα μοντέλο το οποίο φαίνεται άριστο στην εκτός-σύνδεσης αξιολόγηση (offline) μπορεί να αποδειχθεί αναξιόπιστο όταν ενσωματωθεί σε ένα πραγματικό αγωγό εισαγωγής και πρόβλεψης. Για αυτό το λόγο, η αξιολόγηση χωρίς διαρροή είναι το σημείο αναφοράς για τα συμπεράσματα με επιχειρησιακή αξία, ενώ τα αποτελέσματα των δεδομένων με διαρροή χρησιμεύουν κυρίως ως αντιπαράδειγμα.

Τέλος, παρότι τα ευρήματα είναι ενθαρρυντικά, πρέπει να μελετηθούν με επίγνωση των ορίων της συγκεκριμένης μελέτης. Η αξιολόγηση έγινε ως ένας ενδοημερήσιος διαχωρισμός εκπαίδευσης και ελέγχου (within-day holdout), άρα δεν τεκμηριώνει πλήρως το πώς θα συμπεριφερθούν τα μοντέλα μέσα σε διαφορετικές ημέρες ή σε διαφορετικά περιβάλλοντα κίνησης. Βέβαια, το συνολικό αποτέλεσμα δείχνει ξεκάθαρα ότι η αξία της Μηχανικής Μάθησης στη κυβερνοασφάλεια δεν κρίνεται μόνο από τις μετρικές ενός πειράματος, αλλά από το αν η λύση εντάσσεται σωστά σε ένα συνεπή και ελεγχόμενο αγωγό Παρακολούθησης της Ασφάλειας, ώστε να παράγει τις προβλέψεις με σταθερό τρόπο, να υποστηρίζει τη παρακολούθηση και τη διερεύνηση και να μπορεί να τροφοδοτήσει τους πρακτικούς μηχανισμούς της έγκαιρης ειδοποίησης και απόκρισης.

6.6 Επίλογος κεφαλαίου

Εν κατακλείδι, στο παρόν κεφάλαιο συγκεντρώθηκαν τα πειραματικά αποτελέσματα και η αξιολόγηση της συγκεκριμένης προσέγγισης Παρακολούθησης της Ασφάλειας (CM), όπως αυτή υλοποιήθηκε και ενσωματώθηκε στο σύστημα του OpenSearch. Η ανάλυση βασίστηκε στο σύνολο των δεδομένων CIC-DDoS2019 και ειδικότερα στα δεδομένα αποκλειστικά απο την ημέρα 11/03/2019, άρα η αξιολόγηση αντιστοιχεί σε ένα ενδοημερήσιο διαχωρισμό της εκπαίδευσης και του ελέγχου (within-day holdout). Σε αυτό το κομμάτι, δόθηκε έμφαση στις επιδόσεις των μοντέλων σε σενάρια χωρίς διαρροή πληροφορίας (non-leaky) και στη σύγκριση της συμπεριφοράς των μοντέλων Random Forest και XGBoost κάτω από ίδιες συνθήκες δεδομένων και προεπεξεργασίας. Επιπλέον, για να τεκμηριωθεί η διαφορά από τις παραδοσιακές προσεγγίσεις της ανίχνευσης, συμπεριλήφθηκε και η συγκριτική αξιολόγηση έναντι δύο μη επιβλεπόμενων στατιστικών ανιχνευτών αναφοράς, του Z-score και του MAD στο ίδιο πλαίσιο χωρίς διαρροή και με τον ίδιο χρονολογικό διαχωρισμό.

Αρχικά, παρουσιάστηκαν τα αποτελέσματα των δύο ταξινομητών στο σενάριο χωρίς διαρροή πληροφορίας, με στόχο να φανεί καθαρά το πόσο αποτελεσματικά μπορούν να ξεχωρίσουν αυτά τα δύο μοντέλα τη κανονική από τη κακόβουλη κίνηση σε μία διαδικασία αξιολόγησης κάτω από ρεαλιστικές συνθήκες. Η αξιολόγηση δεν στηρίχθηκε σε μία μόνο τιμή, αλλά συνδύασε τους πίνακες της σύγχυσης και τους δείκτες της ακρίβειας των θετικών προβλέψεων, της ανάκλησης και του δείκτη F1, ώστε να εξηγηθεί καλύτερα τι σημαίνουν τα σφάλματα στη πράξη, ειδικά όταν τα δεδομένα είναι έντονα ανισόρροπα. Στη συνέχεια, η συγκριτική ανάλυση έδειξε ότι τα δύο μοντέλα κινούνται σε μεγάλο βαθμό με παρόμοια λογική απόφασης, ενώ οι λίγες περιπτώσεις στις οποίες διαφωνούν είναι ιδιαίτερα χρήσιμες καθώς αποτελούν σημεία τα οποία χρειάζονται μια παραπάνω διερεύνηση, είτε για πιο προσεκτική ποιοτική διερεύνηση είτε για την επιχειρησιακή ιεράρχηση των συμβάντων, όταν το ζητούμενο είναι η αξιόπιστη επιτήρηση και όχι απλώς ένας υψηλός συνολικός δείκτης της απόδοσης.

Ένα από τα πιο σημαντικά συμπεράσματα προέκυψε από τη σύγκριση των δεδομένων με διαρροή (leaky) έναντι των δεδομένων χωρίς διαρροή (non-leaky). Τα αποτελέσματα έδειξαν στη πράξη ότι η διαρροή μπορεί να φουσκώσει τις μετρικές και να δώσει μια πολύ αισιόδοξη εικόνα, σαν το μοντέλο να είναι σχεδόν τέλειο, χωρίς βέβαια αυτό να σημαίνει ότι θα σταθεί το ίδιο καλά όταν χρησιμοποιηθεί σε μία κανονική ροή λειτουργίας. Για αυτό το λόγο, η αξιολόγηση στο σύνολο των δεδομένων χωρίς διαρροή παραμένει το πιο ασφαλές και ουσιαστικό σημείο αναφοράς όταν ο στόχος είναι ένα σύστημα Παρακολούθησης της Ασφάλειας το οποίο δουλεύει σε συνθήκες συνεχούς εισαγωγής και ανάλυσης της δικτυακής κίνησης. Η θέση αυτή ενισχύεται από το γεγονός ότι οι στατιστικοί ανιχνευτές της αναφοράς (statistical baselines) εμφάνισαν πολύ περιορισμένη ικανότητα εντοπισμού της κακόβουλης κίνησης επιβεβαιώνοντας ότι οι απλές μέθοδοι οι οποίες είναι βασισμένες σε κατώφλια (threshold-based) δεν επαρκούν ως μοναδικός μηχανισμός ανίχνευσης στα σύνθετα σενάριο DDoS.

Συνολικά, το κεφάλαιο ανέδειξε ότι η αξία ενός μοντέλου στη κυβερνοασφάλεια δεν κρίνεται μόνο από τους μεγάλους αριθμούς, αλλά από το πώς ερμηνεύονται τα σφάλματα, το πόσο σταθερή είναι η συμπεριφορά του σε ρεαλιστικές συνθήκες και το πόσο ανθεκτική είναι η αξιολόγηση απέναντι σε παγίδες όπως η διαρροή των δεδομένων (data leakage). Στο επόμενο κεφάλαιο, τα ευρήματα αξιοποιούνται για μία πιο ολοκληρωμένη ερμηνεία, μια συζήτηση των περιορισμών και μια διατύπωση των συμπερασμάτων σχετικά με τη συνολική εφαρμογή της προτεινόμενης προσέγγισης.

Κεφάλαιο 7ο: Συμπεράσματα και μελλοντικές επεκτάσεις του συστήματος Παρακολούθησης της Ασφάλειας

7.1 Συνολικά συμπεράσματα της διπλωματικής

Στη συγκεκριμένη διπλωματική σχεδιάστηκε, υλοποιήθηκε και τεκμηριώθηκε ένα ολοκληρωμένο σύστημα Παρακολούθησης της Ασφάλειας (CM) πάνω στη πλατφόρμα του OpenSearch, με βασικό στόχο να φανεί στη πράξη πως περνάμε από την εκτός σύνδεσης εκπαίδευση των μοντέλων της Μηχανικής Μάθησης σε μια λειτουργική και επιχειρησιακά έτοιμη για αξιοποίηση ροή από άκρη σε άκρη. Η εργασία δεν αντιμετώπισε τα μοντέλα ως ένα μεμονωμένο πείραμα, αλλά ως μέρος ενός μεγαλύτερου μηχανισμού, ξεκινώντας από τη προεπεξεργασία και την ευρετηρίαση των δεδομένων, μέχρι τη παραγωγή των προβλέψεων, την αποθήκευση των αποτελεσμάτων, την οπτικοποίηση και την ενεργοποίηση των ειδοποιήσεων.

Η πειραματική αξιολόγηση βασίστηκε στο CIC-DDoS2019 και ειδικότερα στα δεδομένα της 11/03/2019, με ενδοημερήσιο διαχωρισμό της εκπαίδευσης και του ελέγχου (within-day holdout). Στο σενάριο χωρίς διαρροή πληροφορίας (non-leaky), ο Random Forest και ο XGBoost παρουσίασαν υψηλή συνολική απόδοση στο σύνολο του ελέγχου. Ταυτόχρονα, η ανάλυση ανά κλάση μαζί με τους πίνακες της σύγχυσης, ανέδειξε κάτι το οποίο ήταν αναμενόμενο αλλά παραμένει πολύ κρίσιμο, δηλαδή ότι στα ανισόρροπα δεδομένα η συνολική εικόνα μπορεί να φαίνεται πολύ καλή, αλλά ουσιαστικά χρειάζεται να εξετάζονται προσεκτικά τα είδη των σφαλμάτων και η κατανομή τους. Επιπλέον, η σύγκριση με το σενάριο της διαρροής (leaky) έδειξε καθαρά ότι η διαρροή της πληροφορίας μπορεί να φουσκώσει με λάθος τρόπο τις τιμές των μετρικών και να δώσει μια πολύ καλή εικόνα για την αξιοπιστία του μοντέλου. Για αυτό το λόγο, το σενάριο των δεδομένων χωρίς διαρροή αναδεικνύεται ως το ουσιαστικό σημείο αναφοράς όταν ο στόχος είναι η αξιολόγηση του μοντέλου με πραγματική επιχειρησιακή αξία. Τέλος, έγινε και μια σύγκριση με τους παραδοσιακούς στατιστικούς ανιχνευτές αναφοράς χωρίς επίβλεψη, Z-score και MAD στο ίδιο πειραματικό σενάριο, επιβεβαιώνοντας την προστιθέμενη αξία των επιβλεπόμενων μοντέλων Μηχανικής Μάθησης για την επιχειρησιακή ανίχνευση των επιθέσεων DDoS.

Στο επίπεδο της υλοποίησης, τεκμηριώθηκε μια υποδομή πειραματισμού με δυνατότητα εύκολης αναπαραγωγής σε ένα άλλο σύστημα με τη χρήση της εικονικοποίησης και την ανάπτυξη των επιμέρους υπηρεσιών σε δοχεία μέσω του Docker Compose, ώστε το σύστημα να έχει καθαρό διαχωρισμό των ρόλων, σαφώς ορισμένα σημεία της πρόσβασης και σταθερή συμπεριφορά στη λειτουργία του. Τα εκπαιδευόμενα μοντέλα εξήχθησαν σε μορφή ONNX και σε συνδυασμό με την ανάπτυξη ενός ανεξάρτητου προγνωστικού εξυπηρετητή τύπου REST, ενσωματώθηκαν στο OpenSearch μέσω των συνδετήρων της Μηχανικής Μάθησης και των απομακρυσμένων μοντέλων. Έτσι, η πρόβλεψη εκτελείται εκτός του συμπλέγματος (cluster), αλλά η κλήση της γίνεται με τυποποιημένο και ελεγχόμενο τρόπο, με σταθερή και συνεπή μορφή εισόδου και εξόδου. Τέλος, η αξιοποίηση των Πινάκων Ελέγχου του OpenSearch και του μηχανισμού των ειδοποιήσεων πρόσθεσε την απαραίτητη επιχειρησιακή διάσταση, δηλαδή τη συνεχή παρακολούθηση, τη γρήγορη διερεύνηση των συμβάντων και τη δυνατότητα της έγκαιρης ειδοποίησης όταν προκύπτουν κάποιες ύποπτες ενδείξεις.

Συνολικά, το βασικό συμπέρασμα της διπλωματικής είναι ότι η αξία της Μηχανικής Μάθησης στην κυβερνοασφάλεια δεν κρίνεται μόνο από τις υψηλές μετρικές σε μια μεμονωμένη αξιολόγηση. Αξιολογείται κυρίως από το αν το μοντέλο μπορεί να ενταχθεί σωστά σε μία ελεγχόμενη και

παρατηρήσιμη ροή της παρακολούθησης, η οποία μετατρέπει τις προβλέψεις σε μια πρακτική πληροφορία για ανάλυση, οπτικοποίηση, ειδοποίηση και τελικά επιχειρησιακή χρήση.

7.2 Επιστημονική και πρακτική συμβολή της προσέγγισης στο επίπεδο Παρακολούθησης της Ασφάλειας (CM)

Γενικότερα, η παρούσα εργασία έχει επιστημονική αλλά και πρακτική αξία, καθώς μεταφέρει το ενδιαφέρον από τη λογική ενός απλού IDS (Intrusion Detection System) σε μια πιο ολοκληρωμένη προσέγγιση αυτή της Παρακολούθησης της Ασφάλειας (Cybersecurity Monitoring – CM). Σε ένα κλασικό IDS το ζητούμενο είναι συνήθως μια απόφαση για το αν η κίνηση είναι κανονική ή επίθεση ή μια ειδοποίηση. Αντίθετα, εδώ το σύστημα Παρακολούθησης της Ασφάλειας αντιμετωπίζεται ως μια συνεχής επιχειρησιακή ροή, όπου η ανίχνευση είναι μόνο ένα βήμα μέσα σε μια μεγαλύτερη αλυσίδα, δηλαδή στη συλλογή των δεδομένων, στον εμπλουτισμό, στην αποθήκευση, στη παρακολούθηση, στη διερεύνηση και τελικά στην απόκριση.

Στο επιστημονικό επίπεδο, η εργασία δίνει έμφαση στο ότι η αξιολόγηση πρέπει να μοιάζει όσο γίνεται με πραγματική λειτουργία και όχι να μένει σε ένα στατικό πείραμα. Ο ενδοημερήσιος χρονολογικός διαχωρισμός (within-day holdout) και η καθαρή διάκριση μεταξύ των χαρακτηριστικών χωρίς διαρροή (non-leaky) και με διαρροή (leaky) λειτουργούν ως σημείο κλειδί στη μεθοδολογία, καθώς δείχνουν στη πράξη ότι οι μετρικές που εμφανίζονται ως τέλειες μπορεί να οφείλονται σε κάποια διαρροή της πληροφορίας και όχι στη πραγματική ικανότητα γενίκευσης. Έτσι, η συζήτηση δεν περιορίζεται στο ποιος αλγόριθμος είναι καλύτερος ανάμεσα στον Random Forest και στον XGBoost αλλά πηγαίνει σε ένα πιο σημαντικό ερώτημα, δηλαδή στο πόσο αξιόπιστα είναι τα συμπεράσματα όταν τα δεδομένα είναι ανισόρροπα και όταν το κόστος των σφαλμάτων δεν είναι συμμετρικό.

Στο πρακτικό επίπεδο, η βασική συνεισφορά είναι ότι υλοποιήθηκε μια πλήρης και με επαναληψιμότητα αρχιτεκτονική αγωγού Παρακολούθησης της Ασφάλειας (CM pipeline) πάνω στη πλατφόρμα του OpenSearch, όπου η Μηχανική Μάθηση εντάσσεται ως λειτουργικό υποσύστημα και όχι ως απομονωμένο πείραμα. Η εξαγωγή των μοντέλων σε μορφή ONNX, η ανάπτυξη του ανεξάρτητου προγνωστικού εξυπηρετητή REST (REST predictor) και η ενσωμάτωση μέσω των συνδετήρων της Μηχανικής Μάθησης (ML connectors) και των απομακρυσμένων μοντέλων (REMOTE models) διαμορφώνουν ένα ρεαλιστικό πρότυπο της αξιολόγησης, καθώς η πρόβλεψη εκτελείται εκτός του συμπλέγματος (cluster), αλλά η κλήση της παραμένει τυποποιημένη, ελεγχόμενη και εύκολα επεκτάσιμη. Παράλληλα, η αποθήκευση των αποτελεσμάτων σε ξεχωριστούς δείκτες δίνει μια σειρά στη διερεύνηση και επιτρέπει να απαντηθούν πρακτικά ερωτήματα όχι μόνο του τύπου τι προβλέφθηκε αλλά και πότε συνέβη, με ποιο μοντέλο, με τι βαθμολογία και αν συμφώνησαν τα μοντέλα μεταξύ τους.

Επιπλέον, η αξιοποίηση των Πινάκων ελέγχου του OpenSearch (OpenSearch Dashboards) και του μηχανισμού των ειδοποιήσεων (Alerting) μεταφέρει το έργο στο πεδίο της καθημερινής παρακολούθησης, δηλαδή οι προβλέψεις γίνονται ορατές, μπορούν να φιλτραριστούν και να συσχετιστούν χρονικά και να οδηγήσουν σε ειδοποιήσεις μέσω των webhooks προς τον εξωτερικό συλλέκτη. Αυτή η διάσταση αλλάζει ουσιαστικά την εργασία από μία κλασική υλοποίηση IDS, διότι το ζητούμενο δεν είναι μόνο να ανιχνεύσει αλλά να υποστηρίξει το σύστημα τη συνεχή επιτήρηση, τη ταχύτερη κατανόηση του τι συμβαίνει και τη πιο πρακτική ιεράρχηση των συμβάντων, για παράδειγμα να δίνεται προτεραιότητα σε περιπτώσεις όπου τα μοντέλα διαφωνούν, ως ένδειξη αυξημένης αβεβαιότητας.

Συνοψίζοντας, η εργασία δείχνει στη πράξη πως τα μοντέλα της Μηχανικής Μάθησης μπορούν να περάσουν από το στάδιο του πειραματισμού σε μία ολοκληρωμένη και τεκμηριωμένη λειτουργική ροή

Παρακολούθησης της Ασφάλειας μέσα στο OpenSearch, όπου η ανίχνευση, η παρακολούθηση και η ειδοποίηση συνδέονται σε μια ενιαία επιχειρησιακή διαδικασία και όχι σε μεμονωμένα αποτελέσματα ταξινόμησης.

7.3 Περιορισμοί και απειλές εγκυρότητας της μελέτης

Παρότι τα αποτελέσματα και η συνολική υλοποίηση του αγωγού Παρακολούθησης της Ασφάλειας είναι ενθαρρυντικά, η ερμηνεία τους πρέπει να γίνει με γνώση ορισμένων περιορισμών. Η ενότητα αυτή είναι σημαντική, γιατί βοηθάει ώστε να διαβαστούν σωστά τα συμπεράσματα και να ξεκαθαριστεί μέχρι ποιο σημείο μπορεί να μεταφερθεί χωρίς αλλαγές σε διαφορετικές συνθήκες η προτεινόμενη προσέγγιση.

Αρχικά, ο πρώτος και ο βασικός περιορισμός αφορά το χρονικό ορίζοντα των δεδομένων. Η μελέτη στηρίχθηκε αποκλειστικά στα δεδομένα μια ημέρας και συγκεκριμένα της 11/03/2019, άρα η αξιολόγηση αντιστοιχεί σε ενδοημερήσιο διαχωρισμό της εκπαίδευσης και του ελέγχου (within-day holdout) με χρονικό διαχωρισμό. Βέβαια, παρότι αυτός ο τρόπος διαχωρισμού είναι πιο κοντά στη λογική ης χρονικής ακολουθίας των δεδομένων, όπου η εκπαίδευση γίνεται σε παλαιότερες παρατηρήσεις και ο έλεγχος σε μεταγενέστερες, δεν αποδεικνύει το πώς θα συμπεριφερθούν τα μοντέλα σε άλλες ημέρες, σε διαφορετικά μοτίβα φόρτου ή σε ένα διαφορετικό περιβάλλον δικτύου. Με άλλα λόγια, δεν εξετάστηκε η διαημερήσια γενίκευση (cross-day generalization), ούτε η ανθεκτικότητα στις αλλαγές της συμπεριφορά της κίνησης με τη πάροδο του χρόνου. Με βάση τα προηγούμενα, υπάρχει περίπτωση η απόδοση να φαίνεται καλύτερη επειδή το σύνολο ελέγχου προέρχεται από το ίδιο πλαίσιο εκπαίδευσης. Ακόμη και χωρίς διαρροή πληροφορίας, όταν η εκπαίδευση και ο έλεγχος προέρχονται από την ίδια ημέρα είναι πιθανό να μοιράζονται παρόμοιους ρυθμούς, υπογραφές και μοτίβα της κίνησης. Αυτό δεν ακυρώνει τα αποτελέσματα, αλλά μειώνει τη βεβαιότητα με την οποία μπορούμε να πούμε ότι η ίδια συμπεριφορά θα εμφανιστεί σε διαφορετική κίνηση με το ίδιο αποτύπωμα.

Ένας δεύτερος περιορισμός είναι η ανισορροπία των κλάσεων. Στο επιλεγμένο υποσύνολο του CIC-DDoS2019 η κατανομή είναι έντονα άνιση, με αποτέλεσμα οι συνολικές μετρικές όπως η ακρίβεια (accuracy) και οι σταθμισμένοι μέσοι όροι (weighted averages) να επηρεάζονται κυρίως από την πλειοψηφική κλάση. Παρότι η ανάλυση βασίστηκε και στους πίνακες της σύγχυσης (confusion matrices) και στις μετρικές ανά κλάση (precision / recall / F1), παραμένει το γεγονός ότι ακόμη και οι μικρές μεταβολές στα σφάλματα της μειοψηφικής κλάσης μπορεί να έχουν μεγαλύτερο λειτουργικό αντίκτυπο από αυτό που υποδηλώνει η συνολική εικόνα. Επιπλέον, δεν εξετάστηκε ως ξεχωριστό θέμα η επιλογή του κατωφλίου της απόφασης (decision threshold) και η βαθμονόμηση των βαθμολογιών της εμπιστοσύνης, παράγοντες που στη πράξη διαμορφώνουν την ισορροπία μεταξύ των ψευδών συναγερμών (false positives) και των μη ανιχνευμένων επιθέσεων (false negatives).

Ο τρίτος περιορισμός έχει να κάνει με τη διαθεσιμότητα των χαρακτηριστικών στις συνθήκες της λειτουργίας. Η διάκριση ανάμεσα στα χαρακτηριστικά χωρίς διαρροή και με διαρροή είναι μεθοδολογικά σημαντική, ωστόσο στην πραγματική ροή των δεδομένων δεν είναι πάντα αυτονόητο ότι κάθε χαρακτηριστικό είναι διαθέσιμο ακριβώς τη στιγμή που απαιτείται η πρόβλεψη. Κάποια χαρακτηριστικά προϋποθέτουν την ολοκλήρωση της ροής (flow completion), το χρονικό παράθυρο της συγκέντρωσης (aggregation windows) ή τους υπολογισμούς που μπορεί να υλοποιούνται διαφορετικά ανά εργαλείο. Οπότε, ένα μέρος της εγκυρότητας εξαρτάται από το κατά πόσο η παραγωγή των χαρακτηριστικών και η προεπεξεργασία τους μπορούν να αναπαραχθούν χωρίς αλλαγές και σε άλλο περιβάλλον εισαγωγής.

Στο επίπεδο του συστήματος, πρέπει επίσης να σημειωθεί ότι η αρχιτεκτονική αξιολογήθηκε σε ένα ελεγχόμενο πειραματικό περιβάλλον και αναπτύχθηκε σε δοχεία (containerization) μέσω του Docker

Compose με σαφώς ορισμένες υπηρεσίες και σημεία πρόσβασης. Αυτό ευνοεί την αναπαραγωγιμότητα, όμως δεν ισοδυναμεί με τη πλήρη αξιολόγηση της παραγωγικής κλίμακας. Επίσης, δεν εξετάστηκαν τα συστηματικά ζητήματα όπως ο ρυθμός της εισαγωγής (ingest throughput), οι καθυστερήσεις του απομακρυσμένου συμπερασμού (remote inference latency), οι επιπτώσεις σε πόρους (CPU / RAM), η συμπεριφορά κάτω από έντονο φόρτο. Αλλά, ούτε τα σενάρια της αστοχίας (fault tolerance), όπως η προσωρινή μη διαθεσιμότητα του προγνωστικού εξυπηρετητή (predictor) ή οι αποτυχίες στη προώθηση των ειδοποιήσεων μέσω του webhook.

Τέλος, υπάρχει ένας περιορισμός στη μεταφορά των συμπερασμάτων σε άλλα δεδομένα ή σε πραγματικές (enterprise) εγκαταστάσεις. Η μελέτη βασίστηκε σε ένα συγκεκριμένο σύνολο αναφοράς (benchmark) και σε μία συγκεκριμένη μορφή των εγγραφών της ροής. Στη πράξη, τα δεδομένα μπορεί να προέρχονται από διαφορετικές πηγές, όπως Netflow, Zeek και αρχεία καταγραφής του τείχους προστασίας, να έχουν ελλείψεις, διαφορετική σήμανση και διαφορετικό τρόπο δειγματοληψίας. Άρα, παρότι ο αγωγός του συστήματος Παρακολούθησης της Ασφάλειας μεταφέρεται ως αρχιτεκτονικό μοτίβο, η εκπαίδευση και η αξιολόγηση των μοντέλων θα χρειαστούν προσαρμογή και νέα επικύρωση, ίσως και με διαφορετικούς δείκτες της επιτυχίας, όπως οι δείκτες της κόπωσης των ειδοποιήσεων ή ο χρόνος μέχρι τον εντοπισμό.

Συνολικά, οι παραπάνω περιορισμοί δεν ακυρώνουν τα ευρήματα αλλά ξεκαθαρίζουν σε ποιες συνθήκες μπορούμε να τα θεωρήσουμε αξιόπιστα. Η εργασία δείχνει στη πράξη το πώς μπορεί να ενσωματωθεί η Μηχανική Μάθηση σε έναν ολοκληρωμένο αγωγό Παρακολούθησης της Ασφάλειας και αναδεικνύει ξεκάθαρα τις σημαντικές παγίδες, όπως για παράδειγμα τη διαρροή της πληροφορίας. Ωστόσο, για να βγουν πιο καλά συμπεράσματα και να υπάρξει μεγαλύτερη βεβαιότητα ότι η ίδια εικόνα θα υπάρξει και εκτός του συγκεκριμένου πειράματος χρειάζεται μεγαλύτερος χρονικός ορίζοντας των δεδομένων, δοκιμές σε διαφορετικά περιβάλλοντα και σε άλλες ροές και περισσότερος πειραματισμός σε πρακτικά ζητήματα όπως, το σημείο του κατωφλίου της απόφασης, οι μεταβολές στη συμπεριφορά της κίνησης (drift) και η λειτουργία του συστήματος κάτω από αυξημένο φόρτο.

7.4 Προτάσεις για μελλοντικές χρήσεις της εργασίας και επεκτάσεις του συστήματος Παρακολούθησης της Ασφάλειας (CM)

7.4.1 Ενσωμάτωση μοντέλων Βαθιάς Μάθησης στο σύστημα Παρακολούθησης της Ασφάλειας (Deep Learning & LSTM / Autoencoders at CM)

Πρώτα από όλα, μια επέκταση του προτεινόμενου συστήματος Παρακολούθησης της Ασφάλειας είναι η ενσωμάτωση των μοντέλων της Βαθιάς Μάθησης (Deep Learning – DL). Ο στόχος δεν είναι απλά να αντικατασταθούν οι κλασικοί ταξινομητές, αλλά να εμπλουτιστεί η παρακολούθηση με μοντέλα τα οποία μπορούν να αποτυπώσουν πιο περίπλοκα μοτίβα και κατά κύριο λόγο τις χρονικές εξαρτήσεις στη δικτυακή κίνηση. Γενικότερα, οι προσεγγίσεις της Βαθιάς Μάθησης είναι χρήσιμες στη κυβερνοασφάλεια γιατί μαθαίνουν πιο πλούσιες αναπαραστάσεις από δεδομένα υψηλής διάστασης και αξιοποιούν την πληροφορία των ακολουθιών, κάτι το οποίο είναι πολύ σημαντικό όταν η ανίχνευση δεν βασίζεται μόνο στα στιγμιαία χαρακτηριστικά αλλά στην εξέλιξη της συμπεριφοράς μέσα στο χρόνο[25].

Στο τομέα του συστήματος Παρακολούθησης της Ασφάλειας, μεγάλο ενδιαφέρον έχουν τα μοντέλα των ακολουθιών, όπως τα Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks – RNNs) και οι παραλλαγές τους (GRU / LSTM), επειδή είναι σχεδιασμένα για να πιάνουν τις χρονικές συσχετίσεις που ξεδιπλώνονται στα διαδοχικά συμβάντα και στις ροές[25]. Αυτό επιτρέπει μια πιο συνολική εικόνα

της κίνησης, δηλαδή αντί το σύστημα να παίρνει τις αποφάσεις αποκλειστικά ανά ροή (per-flow), μπορεί να επεκταθεί σε συμπεράσματα ανά πηγή ή ανά χρονικό παράθυρο (window-based), όπου αξιολογούνται οι ακολουθίες από τις ροές και παράγεται μια συνολική εκτίμηση του κινδύνου που αντανακλά τη συμπεριφορά σε βάθος χρόνου.

Παράλληλα, για τα περιβάλλοντα της παρακολούθησης πολύ χρήσιμες είναι και οι μη επιβλεπόμενες η ημι-επιβλεπόμενες προσεγγίσεις, όπως οι Αυτοκωδικοποιητές (Autoencoders), οι οποίοι μπορούν να λειτουργήσουν ως αισθητήρες αναγνώρισης της ανωμαλίας (anomaly sensors). Ένας αυτοκωδικοποιητής μπορεί να εκπαιδευτεί στη κανονική κίνηση και κατά τη λειτουργία και να παράγει ένα σφάλμα ανακατασκευής (reconstruction error) ως σήμα της απόκλισης[25]. Τα πλεονεκτήματα είναι, ότι προσφέρει μια συμπληρωματική κάλυψη δίπλα στους επιβλεπόμενους ταξινομητές, ειδικά όταν η ετικετοποίηση είναι ατελής ή όταν εμφανίζονται νέα ή μεταλλαγμένα μοτίβα επιθέσεων[25]. Επίσης, δίνει μια έγκαιρη ένδειξη ότι κάτι αλλάζει στη ροή πριν αυτό αποτυπωθεί καθαρά στις κλασικές μετρικές, έτσι το σύστημα Παρακολούθησης της Ασφάλειας δεν μένει μόνο στις δυαδικές αποφάσεις, αλλά αποκτά και ένα συνεχές σήμα (score) το οποίο μπορεί να αξιοποιηθεί για την ιεράρχηση των συμβάντων και για πιο ευέλικτους κανόνες ειδοποίησης[25].

Ωστόσο, για να ενταχθεί η Βαθιά Μάθηση στο σύστημα Παρακολούθησης της Ασφάλειας πρέπει να σχεδιαστεί με καθαρούς επιχειρησιακούς όρους. Η Βαθιά Μάθηση συνήθως αυξάνει τις απαιτήσεις στους υπολογιστικούς πόρους και ανεβάζει τη πολυπλοκότητα στην εκπαίδευση, στη ρύθμιση και στη διαχείριση της εισόδου και της αναπαράστασης των δεδομένων. Συχνά, απαιτείται ένας πιο προσεκτικός ορισμός του σχήματος της εισόδου (feature schema) και στις ακολουθιακές προσεγγίσεις, ένας σαφής σχεδιασμός των χρονικών παραθύρων (windowing) ώστε η χρονική πληροφορία να αξιοποιείται σωστά[25]. Για να παραμείνει η λύση πάνω στα πρότυπα της συγκεκριμένης εργασίας, μια πολύ καλή επιλογή είναι η αξιοποίηση της Βαθιάς Μάθησης ως απομακρυσμένο μοντέλο (remote inference), με σαφή ορισμένη μορφή εισόδου και εξόδου (inference contract), ώστε να ενσωματώνονται χωρίς ρήξεις στην ίδια επιχειρησιακή ροή. Με αυτό το τρόπο, διατηρείται ο έλεγχος, η παρατηρησιμότητα και η αναπαραγωγιμότητα, ενώ παράλληλα ανοίγει ο δρόμος για τα μοντέλα τα οποία υπερέχουν όταν οι χρονικές εξαρτήσεις ή τα σήματα της ανωμαλίας δίνουν ουσιαστικό πλεονέκτημα[25].

Τέλος, η επιλογή των μοντέλων της Βαθιάς Μάθησης έχει νόημα όταν συνδέεται με ένα συγκεκριμένο στόχο παρακολούθησης, για παράδειγμα καλύτερη μοντελοποίηση των ακολουθιών με GRU / LSTM όταν δίνεται βάρος στη συμπεριφορά στο χρόνο, ανίχνευση των αποκλίσεων με αυτοκωδικοποιητές όταν χρειάζεται η έγκαιρη προειδοποίηση για τις άγνωστες συμπεριφορές και ο εμπλουτισμός των πινάκων ελέγχου και των ειδοποιήσεων με επιπλέον βαθμολογίες και σήματα αβεβαιότητας[25]. Με αυτή τη λογική, η Βαθιά Μάθηση δεν μπαίνει στο σύστημα Παρακολούθησης της Ασφάλειας ως αντικαταστάτης ενός ταξινομητή αλλά ως λειτουργική ενίσχυση της ίδιας της παρακολούθησης.

7.4.2 Ροές δεδομένων σε πραγματικό χρόνο με συνεχής επιτήρηση (Real-time streaming CM)

Μια ουσιαστική μελλοντική επέκταση του συστήματος Παρακολούθησης της Ασφάλειας είναι η μετάβαση από τη περιοδική επεξεργασία σε επεξεργασία ροής σε πραγματικό χρόνο (streaming). Η ανάγκη αυτή προκύπτει από ο γεγονός ότι τα εργαλεία της κυβερνοασφάλειας παράγουν πολύ μεγάλους όγκους δεδομένων οι οποίοι ρέουν συνεχώς προς μια κεντρική μονάδα, όπου ο στόχος είναι η γρήγορη ανταπόκριση στις μεταβολές της κατάστασης ασφάλειας[26]. Σε τέτοια σενάρια, η πλήρης επεξεργασία όλων των δεδομένων μαζί δεν είναι απαραίτητη, καθώς το αυτό που είναι το πιο σημαντικό στο επιχειρησιακό κομμάτι είναι η έγκαιρη επεξεργασία των τρεχουσών ροών[26].

Στο πλαίσιο αυτό, μια ρεαλιστική κατεύθυνση είναι η υιοθέτηση της αρχιτεκτονικής stream processing, όπου τα συμβάντα περνούν από τα διακριτά στάδια της επεξεργασίας και προωθούνται διαδοχικά στο επόμενο στάδιο μόλις ολοκληρωθεί ο υπολογισμός αυτού του βήματος. Η λογική οργανώνεται σε βήματα τα οποία λαμβάνουν συνεχώς γεγονότα, τα μετασχηματίζουν και στη συνέχεια παράγουν μια εμπλουτισμένη έξοδο η οποία μπορεί να οδηγήσει σε κάποια ειδοποίηση[26]. Με αυτό το τρόπο, η επεξεργασία γίνεται σε μικρό χρονικό διάστημα μετά τη λήψη των δεδομένων, συχνά από χιλιοστά του δευτερολέπτου έως λεπτά, ώστε να υποστηρίζεται πρακτικά η σχεδόν real-time επιτήρηση[26].

Ακόμη, κεντρικό στοιχείο στη ροϊκή Παρακολούθηση της Ασφάλειας (streaming CM) είναι τα χρονικά παράθυρα (time windows). Αντί να εξετάζεται όλη η διαδικασία, το σύστημα κρατά ένα πρόσφατο κομμάτι της, για παράδειγμα τα τελευταία δεκαπέντε λεπτά και μέσα σε αυτό το χρονικό διάστημα υπολογίζει συνοπτικά τα χαρακτηριστικά τα οποία είναι κατάλληλα για άμεση ανάλυση. Έτσι, τα εισερχόμενα συμβάντα μετατρέπονται σε μια δομημένη μορφή, ώστε να μπορούν να εφαρμοστούν οι σταθεροί και αποδοτικοί υπολογισμοί της συγκέντρωσης, οι οποίοι είναι δύσκολο να γίνουν κατευθείαν πάνω σε ακατέργαστα γεγονότα[26]. Στη συνέχεια, τα δεδομένα μπορούν να ομαδοποιούνται ανά οντότητα ενδιαφέροντος, ώστε κάθε οντότητα να εκπροσωπείται ανά παράθυρο από ένα συμπαγές διάνυσμα χαρακτηριστικών το οποίο περιγράφει τη συμπεριφορά της.

Πάνω σε αυτό το στιγμιότυπο του παραθύρου (aggregated snapshot) το επόμενο βήμα είναι η ταξινόμηση του κινδύνου σε πραγματικό χρόνο (real-time classification). Εκεί, κάθε νέο διάνυσμα χαρακτηριστικών μπορεί να ταξινομηθεί σε μία από τις ήδη ορισμένες κατηγορίες, ενώ γίνεται και ο χειρισμός των περιπτώσεων που δεν ταιριάζουν στα γνωστά πρότυπα[26]. Η ροή αυτή συνδέεται άμεσα με το σύστημα Παρακολούθησης της Ασφάλειας, διότι η έξοδος της ταξινόμησης μπορεί να τροφοδοτεί άμεσα τις εμπλουτισμένες ειδοποιήσεις (enriched alerts) ή τους μηχανισμούς της ειδοποίησης προς τους διαχειριστές, μειώνοντας σε μεγάλο βαθμό τη καθυστέρηση σε σχέση με τις αναφορές οι οποίες βασίζονται σε μεγάλα χρονικά διαστήματα[26]. Ένα πρόσθετο θέμα που προκύπτει στις πραγματικές ροές είναι η επικαιροποίηση του μοντέλου, καθώς η περιοδική ανανέωση μπορεί να είναι αναγκαία για να παραμένει το μοντέλο επίκαιρο, ενώ μπορούν να διερευνηθούν και οι δομικές ενημερώσεις (dynamic model update) κατά τη ροή, με προσεκτική αντιμετώπιση των διαφορών που προκύπτουν έναντι ενός εκτός σύνδεσης χαρακτηριστικού[26].

Τέλος, για τη πρακτική υλοποίηση, η τμηματοποιημένη σε στάδια αρχιτεκτονική (step-based), με τοπική διατήρηση του τρέχοντος παραθύρου και προώθηση μόνο των απαραίτητων ή αλλαγμένων δεδομένων προς τα επόμενα στάδια, μπορεί να κρατήσει την υπολογιστική επιβάρυνση σε ελεγχόμενα επίπεδα και να κάνει τα επιμέρους υποσυστήματα πιο ελαφριά. Επιπλέον, η επιλογή του μεγέθους του παραθύρου αποτελεί μια παράμετρο που επηρεάζει τόσο τη καθυστέρηση όσο και τη σταθερότητα των χαρακτηριστικών, οπότε αποτελεί σημαντικό άξονα του μελλοντικού πειραματισμού στα συστήματα Παρακολούθησης της Ασφάλειας[26]. Αντίθετα, η χρήση των κατωφλίων (thresholds) πάνω στα συγκεντρωτικά (aggregated) χαρακτηριστικά μπορεί να διερευνηθεί ως ένας μηχανισμός περιορισμού του θορύβου στο στάδιο της δημιουργίας των ειδοποιήσεων[26].

7.4.3 Εμπλουτισμός με εταιρικά δεδομένα (Enterprise CM)

Επιπρόσθετα, μια πολύ χρήσιμη μελλοντική εξέλιξη του συστήματος Παρακολούθησης της Ασφάλειας (CM) είναι να εμπλουτιστεί με εταιρικά δεδομένα, ώστε να περάσει από την καθαρά τεχνική παρακολούθηση, η οποία αποτελείται από πακέτα, ροές, δείκτες, και ειδοποιήσεις σε μία προσέγγιση που εξυπηρετεί πιο άμεσα τις ανάγκες ενός οργανισμού. Υπό αυτές τις συνθήκες, η κυβερνοασφάλεια

δεν αντιμετωπίζεται μόνο σαν θέμα του τμήματος πληροφορικής αλλά ως επιχειρησιακή λειτουργία η οποία πρέπει να συνδέεται με υπηρεσίες, κρίσιμες διαδικασίες και πραγματικές επιπτώσεις[27].

Στη πράξη, αυτό σημαίνει ότι η παρακολούθηση δεν σταματά όταν εντοπιστεί κάτι ύποπτο, αλλά προσπαθεί να βρει από τι επηρεάστηκε αυτό και πόσο σοβαρό είναι. Για παράδειγμα, η ίδια τεχνική ένδειξη μπορεί να έχει διαφορετική σημασία, ανάλογα με το αν αφορά ένα σταθμό εργασίας χαμηλής κρισιμότητας ή έναν εξυπηρετητή που υποστηρίζει μια κρίσιμη επιχειρησιακή υπηρεσία[27]. Άρα, οι ειδοποιήσεις, η συσχέτιση των γεγονότων και η διερεύνηση των περιστατικών αποκτούν μεγαλύτερη αξία όταν μπαίνουν πάνω στη πληροφορία για τα περιουσιακά στοιχεία (assets), τις υπηρεσίες, τους ρόλους, τη κρισιμότητα των εφαρμογών και τις επιχειρησιακές προτεραιότητες[27].

Ένας κεντρικός μηχανισμός για να γίνει αυτή η σύνδεση είναι η αξιοποίηση της επιχειρησιακής αρχιτεκτονικής (Enterprise Architecture – EA) ως ένα κοινό σημείο αναφοράς ανάμεσα στις επιχειρήσεις και στο τμήμα της Πληροφορικής. Η επιχειρησιακή αρχιτεκτονική μπορεί να λειτουργήσει ως οργανωμένη χαρτογράφηση του οργανισμού, ώστε η ανάλυση του κινδύνου να μην βασίζεται μόνο στα τεχνικά logs αλλά και στη κατανόηση της δομής των λειτουργιών και από τι επηρεάζονται όταν κάτι πάει στραβά[27].

Με αυτή τη λογική, το σύστημα Παρακολούθησης της Ασφάλειας μπορεί να τροφοδοτείται από τη τηλεμετρία του συστήματος αλλά και από την εταιρική γνώση ώστε να μπορέσει να εκτιμηθεί η επιχειρησιακή επίπτωση[27]. Για παράδειγμα, ένα περιστατικό μπορεί να βαθμολογείται (risk scoring) με βάση τη πιθανότητα της κακόβουλης ενέργειας και τη κρισιμότητα του περιουσιακού στοιχείου ή της υπηρεσίας που επηρεάζεται[27]. Έτσι, η πλατφόρμα δεν λειτουργεί μόνο ως μηχανισμός της ανίχνευσης, αλλά και ως πρακτικό εργαλείο της ιεράρχησης της σειράς επίβλεψης του αναλυτή.

Επιπλέον, μια υλοποίηση η οποία είναι προσανατολισμένη στο επιχειρησιακό περιβάλλον χρειάζεται μια συνεχόμενη διαδικασία διαχείρισης του κινδύνου (risk management), δηλαδή πρέπει να γίνεται εκτίμηση του κινδύνου, σχεδιασμός της απόκρισης, εφαρμογή κάποιων ελέγχων και παρακολούθηση του κατά πόσο οι έλεγχοι λειτουργούν όπως πρέπει[27]. Σε αυτή την οπτική, η συνεχόμενη παρακολούθηση δεν αποσκοπεί μόνο στην ανίχνευση των περιστατικών, αλλά και στο να διατηρείται ενημερωμένη η εικόνα στις νέες απειλές και ευπάθειες, στις αλλαγές στις υποδομές και στις μεταβολές στις διαδικασίες και στα περιουσιακά στοιχειά (assets) τα οποία αλλάζουν το συνολικό επίπεδο του κινδύνου[27].

Τέλος, όλο αυτό αποδεικνύει ότι η αξιολόγηση ενός επιχειρησιακού περιβάλλοντος Παρακολούθησης της Ασφάλειας δεν πρέπει να βασίζεται μόνο στο πόσο καλά γίνεται η ταξινόμηση αλλά και σε κάποιους όρους επιχειρησιακής χρησιμότητας, για παράδειγμα αν μειώνει το χρόνο του εντοπισμού ενός περιστατικού, αν υποστηρίζει καλύτερη τεκμηρίωση και αν συνδέει με τρόπο εύκολα εντοπίσιμο την επιχειρησιακή απαίτηση με το τεχνικό μέτρο της προστασίας και τη παρακολούθησή του[27].

7.4.4 Κατανεμημένοι κόμβοι των συστημάτων Παρακολούθησης της Ασφάλειας και ομοσπονδιακή λειτουργία (Distributed & Federated CM nodes)

Μια ακόμη μελλοντική επέκταση του συστήματος Παρακολούθησης της Ασφάλειας είναι η μετάβαση από μια καθαρά κεντρικοποιημένη λογική σε ένα κατανεμημένο σχήμα, όπου οι πολλαπλοί κόμβοι της επιτήρησης λειτουργούν τοπικά και συνεργάζονται ομοσπονδιακά[28]. Η κατεύθυνση αυτή στοχεύει στη μικρότερη καθυστέρηση και στην ανίχνευση, στη καλύτερη κλιμάκωση και στην ισχυρότερη προστασία της ιδιωτικότητας, ειδικά σε περιβάλλοντα με γεωγραφική διασπορά ή αυξημένες απαιτήσεις της συμμόρφωσης[28].

Η βασική ιδέα είναι ότι κάθε κόμβος συλλέγει και επεξεργάζεται τα δεδομένα του τοπικά, εκπαιδεύοντας ή προσαρμόζοντας ένα μοντέλο πάνω στη δική του κίνηση και στα δικά του συμβάντα. Στη συνέχεια, προς ένα κεντρικό σημείο μεταφέρονται μόνο οι ενημερώσεις του μοντέλου, όπως οι παράμετροι και τα βάρη και όχι τα πρωτογενή δεδομένα, όπως τα logs και οι ροές, ώστε η γνώση να διαμοιράζεται χωρίς να εκτίθενται οι ευαίσθητες πληροφορίες[28]. Με αυτό το τρόπο, η προσέγγιση ταιριάζει ιδιαίτερα στα επιχειρησιακά περιβάλλοντα, όπου η μεταφορά των ακατέργαστων δεδομένων προς ένα κεντρικό σύστημα μπορεί να είναι δύσκολη ή ανεπιθύμητη για λόγους ιδιωτικότητας και διακυβέρνησης των δεδομένων [28].

Στην ομοσπονδιακή μάθηση (Federated Learning), ο κεντρικός ρόλος της είναι να συνδυάζει τις ενημερώσεις από πολλούς κόμβους και να παράγει ένα παγκόσμιο μοντέλο το οποίο διανέμεται ξανά στους συμμετεχόντες. Ένας τυπικός μηχανισμός είναι το Federated Averaging (FedAvg), όπου οι τοπικές ενημερώσεις συντίθενται, συχνά με στάθμιση ως προς το πλήθος ή την ποιότητα των τοπικών δεδομένων, για να προκύψει το νέο παγκόσμιο μοντέλο[28]. Παράλληλα, επειδή στα πραγματικά δίκτυα οι κόμβοι σπάνια βλέπουν ίδιας φύσης δεδομένα (non-IID), χρειάζεται σχεδιασμός ώστε η ομοσπονδιακή διαδικασία να παραμένει σταθερή και χρήσιμη ακόμη και όταν τα προφίλ της κίνησης και το μίγμα των απειλών διαφέρουν αισθητά από ζώνη σε ζώνη[28].

Ένα επιπλέον πρακτικό ζήτημα στις κατανεμημένες αρχιτεκτονικές είναι η αξιοπιστία των ενημερώσεων, καθώς δεν είναι δεδομένο ότι κάθε ενημέρωση είναι ποιοτική και καλοπροαίρετη. Για αυτό το λόγο, προτείνονται οι μηχανισμοί του ελέγχου και της επικύρωσης των ενημερώσεων να γίνονται πριν ενσωματωθούν στο παγκόσμιο μοντέλο, ώστε να ενισχύεται η ανθεκτικότητα του συστήματος απέναντι στις θορυβώδεις ή στις ύποπτες συνεισφορές[28]. Σε μεγαλύτερη κλίμακα, μπορούν επίσης να αξιοποιηθούν οι τεχνικές της συμπίεσης των ενημερώσεων και οι ασφαλείς μέθοδοι της επικοινωνίας, ώστε να περιορίζεται το υπολογιστικό και το δικτυακό κόστος και να προστατεύεται η ακεραιότητα της ανταλλαγής[28].

Με βάση τα παραπάνω, μια σωστή κίνηση για την εξέλιξη της συγκεκριμένης διπλωματικής είναι η υλοποίηση των κατανεμημένων κόμβων του συστήματος Παρακολούθησης της Ασφάλειας (Distributed CM nodes), όπου κάθε κόμβος διατηρεί τοπικά τη ροή της εισαγωγής και της πρόβλεψης και την επιχειρησιακή επιτήρηση για τη περιοχή της ευθύνης του, ενώ σε δεύτερο επίπεδο συμμετέχει στην ομοσπονδιακή εκπαίδευση και ενημέρωση των μοντέλων[28]. Έτσι, μπορεί να μεταφέρεται σταδιακά η γνώση για τις απειλές ανάμεσα στις διαφορετικές ζώνες, χωρίς να απαιτείται η συγκέντρωση των πρωτογενών δεδομένων σε ένα σημείο[28].

Στο σενάριο αυτό, η αξία δεν περιορίζεται μόνο στους δείκτες της ταξινόμησης. Επεκτείνεται στη συνολική προσαρμοστικότητα και στην ανθεκτικότητα της λύσης, επειδή το σύστημα Παρακολούθησης της Ασφάλειας μπορεί να μαθαίνει από διαφορετικά περιβάλλοντα, να προσαρμόζεται πιο γρήγορα στις εξελισσόμενες απειλές και ταυτόχρονα να μειώνει το κίνδυνο ενός κεντρικού σημείου αποτυχίας (single point of failure), λόγω της αποκεντρωμένης λειτουργίας του[28].

7.5 Επίλογος κεφαλαίου

Εν κατακλείδι, στο κεφάλαιο αυτό συνοψίστηκαν τα βασικά συμπεράσματα της διπλωματικής και αναδείχθηκε η επιστημονική και η πρακτική αξία της προτεινόμενης προσέγγισης ως ένα σύστημα Παρακολούθησης της Ασφάλειας (CM) και όχι ως ένα μεμονωμένο σύστημα ανίχνευσης των εισβολών (IDS). Η εργασία έδειξε ότι η πραγματική χρησιμότητα της Μηχανικής Μάθησης στη κυβερνοασφάλεια δεν κρίνεται μόνο από τις υψηλές μετρικές στα απομονωμένα πειράματα, αλλά και από το κατά πόσο τα μοντέλα μπορούν να ενταχθούν σε μια συνεπή, ελεγχόμενη και επιχειρησιακά αξιοποιήσιμη ροή από

άκρη σε άκρη. Στο πλαίσιο της τεκμηρίωσης, η αξιολόγηση δεν περιορίστηκε μόνο στα επιβλεπόμενα μοντέλα, αλλά συμπληρώθηκε και με σύγκριση έναντι των παραδοσιακών στατιστικών ανιχνευτών αναφοράς, ώστε να αποσαφηνιστεί η διαφορά τους κάτω από ιδίες συνθήκες. Σε αυτή τη ροή, οι προβλέψεις δεν μένουν στην άκρη αλλά μεταφράζονται σε πρακτική επιτήρηση, διερεύνηση και έγκαιρη ειδοποίηση μέσα στο περιβάλλον της λειτουργίας.

Παράλληλα, αποσαφηνίστηκαν οι περιορισμοί και οι απειλές της εγκυρότητας που συνοδεύουν μια εφαρμοσμένη μελέτη αυτού του τύπου, με έμφαση στη γενικευσιμότητα, στη δυναμική μεταβολή της δικτυακής κίνησης και στο κίνδυνο των παραπλανητικών συμπερασμάτων όταν η αξιολόγηση επηρεάζεται από φαινόμενα όπως η διαρροή της πληροφορίας ή από συνθήκες οι οποίες δεν αντιπροσωπεύουν τη πραγματική λειτουργία. Η ανάδειξη αυτών των σημείων δεν μειώνει τα ευρήματα αντίθετα, τα τοποθετεί στο σωστό πλαίσιο ώστε να διαβαστούν με τεχνική ακρίβεια και επιχειρησιακό ρεαλισμό. Ιδιαίτερα, η χρήση του πλαισίου χωρίς διαρροή, του πειραματικού και του χρονολογικού διαχωρισμού, σε συνδυασμό με τη σύγκριση έναντι των μεθόδων της αναφοράς, Z-score και MAD ανέδειξε ότι οι απλές στατιστικές προσεγγίσεις οι οποίες είναι βασισμένες σε κατώφλια (threshold-based) εμφανίζουν αρκετά μειωμένη ικανότητα εντοπισμού της κακόβουλης κίνησης σε σχέση με τα μοντέλα της Μηχανικής Μάθησης, παρά τον περιορισμένο θόρυβο στις ψευδείς θετικές ενδείξεις.

Τέλος, παρουσιάστηκαν οι ρεαλιστικές κατευθύνσεις της μελλοντικής εξέλιξης οι οποίες μπορούν να ενισχύσουν τη βιωσιμότητα και την επεκτασιμότητα ενός συστήματος Παρακολούθησης της Ασφάλειας, δηλαδή πιο συγκεκριμένα η ενσωμάτωση πιο σύνθετων μοντέλων, η μετάβαση σε ροές πραγματικού χρόνου, ο εμπλουτισμός με εταιρικά δεδομένα και η υιοθέτηση των κατανεμημένων ή ομοσπονδιακών σχημάτων λειτουργίας. Οι κατευθύνσεις αυτές δείχνουν, πως το σύστημα Παρακολούθησης της Ασφάλειας μπορεί να εξελιχθεί από ένα αποδεικτικό πρωτότυπο σε μια πιο ώριμη λύση, η οποία είναι ικανή να προσαρμοστεί σε διαφορετικές πηγές δεδομένων, σε απαιτήσεις κλίμακας και σε πολιτικές διακυβέρνησης. Με τα παραπάνω ολοκληρώνεται η διπλωματική εργασία, συγκεντρώνοντας τα βασικά ευρήματα, τους περιορισμούς και τις προτεινόμενες προοπτικές εξέλιξης του συστήματος Παρακολούθησης της Ασφάλειας (CM).

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] A. S. Ashoor and S. Gore, “Importance of intrusion detection system (IDS),” *International Journal of Scientific and Engineering Research*, vol. 2, no. 1, pp. 1–4, Jan. 2011.

[2] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, Jul. 2015, doi: 10.1126/science.aaa8415.

[3] S. Kuppusamy, *Mastering OpenSearch: A Comprehensive Guide*. [Online]. Available: https://books.google.com/. Accessed: Nov. 23, 2025.

[4] L. Lapadula, *State of the Art in CyberSecurity Monitoring: A Supplement*, 2001.

[5] N. Shone *et al*., “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.

[6] G. Kaur *et al*., “Introduction to cybersecurity,” in *Understanding Cybersecurity Management in FinTech*, pp. 17–34, 2021, doi: 10.1007/978-3-030-79915-1_2.

[7] H. Hindy *et al*., “A taxonomy of malicious traffic for intrusion detection systems,” in *Proc. International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA)*, Jun. 2018, pp. 1–4, doi: 10.1109/CYBERSA.2018.8551386.

[8] Y. Otoum and A. Nayak, “AS-IDS: Anomaly and signature based IDS for the Internet of Things,” *Journal of Network and Systems Management*, vol. 29, no. 3, Mar. 2021, doi: 10.1007/s10922-021-09589-6.

[9] H. Debar *et al*., “Towards a taxonomy of intrusion-detection systems,” *Computer Networks*, vol. 31, no. 8, pp. 805–822, Apr. 1999, doi: 10.1016/S1389-1286(98)00017-6.

[10] M. Aamir and M. A. Zaidi, “A Survey on DDoS Attack and Defense Strategies: From Traditional Schemes to Current Techniques,” Interdisciplinary Information Sciences, vol. 19, no. 2, pp. 173–200, Nov. 2013, doi: 10.4036/iis.2013.173.

[11] H.-J. Liao *et al*., “Intrusion detection system: A comprehensive review,” *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, Jan. 2013, doi: 10.1016/j.jnca.2012.09.004.

[12] AAMIR, Muhammad, and Mustafa Ali ZAIDI. “A Survey on DDoS Attack and Defense Strategies: From Traditional Schemes to Current Techniques.” Interdisciplinary Information Sciences, vol. 19, no. 2, 2013, pp. 173–200, https://doi.org/10.4036/iis.2013.173. Accessed 25 Oct. 2019

[13] T. D. Le *et al*., “Cybersecurity analytics for the enterprise environment: A systematic literature review,” *Electronics*, vol. 14, no. 11, 2025, Art. no. 2252, doi: 10.3390/electronics14112252.

[14] A. Handa *et al*., “Machine learning in cybersecurity: A review,” *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 4, Feb. 2019, doi: 10.1002/widm.1306.

[15] R. Geetha and T. Thilagam, “A review on the effectiveness of machine learning and deep learning algorithms for cyber security,” *Archives of Computational Methods in Engineering*, Sep. 2020, doi: 10.1007/s11831-020-09478-2.

[16] P. C. Sen *et al*., "Supervised classification algorithms in machine learning: A survey and review," *Advances in Intelligent Systems and Computing*, vol. 937, pp. 99–111, Jul. 2019, doi: 10.1007/978-981-13-7403-6_11.

[17] B. de Ville, "Decision trees," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 6, pp. 448–455, Oct. 2013, doi: 10.1002/wics.1278.

[18] A. David and A. Adefolaju, "A model for intrusion detection in cybersecurity using random forest algorithm," *Afr. J. Comp. & ICT*, vol. 14, no. 2, pp. 46–51, 2021. [Online]. Available: africanjournalofcomputingict1.wordpress.com/wp-content/uploads/2025/01/d9e25-vol142sep21pap5pagenumb.pdf. Accessed: Dec. 8, 2025.

[19] L. Podlodowski and M. Kozlowski, "Application of XGBoost to the cyber-security problem of detecting suspicious network traffic events," in *Proc. IEEE International Conference on Big Data*, Dec. 2019, doi: 10.1109/BigData47090.2019.9006586.

[20] O. El Gharbaoui *et al*., "Evaluating AI and ML in network security: A comprehensive literature review," *Procedia Computer Science*, vol. 251, pp. 727–733, 2024, doi: 10.1016/j.procs.2024.11.176.

[21] G. S. Handelman *et al*., "Peering into the black box of artificial intelligence: Evaluation metrics of machine learning methods," *American Journal of Roentgenology*, vol. 212, no. 1, pp. 38–43, Jan. 2019, doi: 10.2214/AJR.18.20224.

[22] H. Alqahtani *et al*., "Cyber intrusion detection using machine learning classification techniques," *Communications in Computer and Information Science*, vol. 1235, pp. 121–131, 2020, doi: 10.1007/978-981-15-6648-6_10.

[23] Canadian Institute for Cybersecurity (CIC), University of New Brunswick, "DDoS 2019 (CICDDoS2019) dataset," *CIC Datasets*. [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html. Accessed: Dec. 14, 2025.

[24] C. Pentu *et al*., "A review of DDoS evaluation dataset: CICDDoS2019 dataset," *Lecture Notes in Electrical Engineering*, pp. 389–397, 2023, doi: 10.1007/978-981-99-3691-5_34.

[25] M. Alazab and M.-J. Tang, Eds., *Deep Learning Applications for Cyber Security*. Cham, Switzerland: Springer, 2019, doi: 10.1007/978-3-030-13057-2.

[26] P. Pekarcik *et al*., "Real-time processing of cybersecurity system data for attacker profiling," in *Proc. IEEE International Conference on Informatics*, Nov. 2019, doi: 10.1109/Informatics47936.2019.9119254.

[27] T. Chmielecki *et al*., "Enterprise-oriented cybersecurity management," in *Proc. Federated Conference on Computer Science and Information Systems*, vol. 2, Sep. 2014, doi: 10.15439/2014F38.

[28] P. Gupta *et al*., "Federated learning-driven intrusion detection for cybersecurity in smart distribution system," in *Proc. IEEE Workshop* (GCWKSHp), Dec. 2024, pp. 1–6, doi: 10.1109/gcwkshp64532.2024.11101136.

ΠΑΡΑΡΤΗΜΑ Α: docker-compose.yml

```
version: "3.8"


services:
  opensearch-node1:
    image: opensearchproject/opensearch:3.3.1
    container_name: opensearch-node1
    environment:
      - cluster.name=os-cluster
      - node.name=node-1
      - discovery.type=single-node
      - bootstrap.memory_lock=true
      - OPENSEARCH_JAVA_OPTS=-Xms1g -Xmx1g
      -
OPENSEARCH_INITIAL_ADMIN_PASSWORD=${OPENSEARCH_INITIAL_ADMIN_PASSW
ORD}
      - DISABLE_INSTALL_DEMO_CONFIG=false
      - DISABLE_SECURITY_PLUGIN=false
    ulimits:
      memlock:
        soft: -1
        hard: -1
    ports:
      - "9200:9200"
    volumes:
```

```
    - opensearch-data1:/usr/share/opensearch/data


  opensearch-dashboards:
    image: opensearchproject/opensearch-dashboards:3.3.0
    container_name: opensearch-dashboards
    depends_on:
      - opensearch-node1
    ports:
      - "5601:5601"
    environment:
      - OPENSEARCH_HOSTS=["https://opensearch-node1:9200"]
      - OPENSEARCH_USERNAME=admin
      - OPENSEARCH_PASSWORD=${OPENSEARCH_INITIAL_ADMIN_PASSWORD}
    volumes:
      - ./dashboards.yml:/usr/share/opensearch-dashboards/config/opensearch_dashboards.yml:ro


volumes:
  opensearch-data1:
OPENSEARCH_INITIAL_ADMIN_PASSWORD=********
```

ΠΑΡΑΡΤΗΜΑ Β: train_non_leaky_13.py

```
from future import annotations
import json import sys from pathlib
import Path
import numpy as np
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import ( classification_report, confusion_matrix, f1_score, roc_auc_score, )
```

```
from xgboost import XGBClassifier

import onnx from skl2onnx

import convert_sklearn from skl2onnx.common.data_types

import FloatTensorType as SKLFloatTensorType

from onnxmltools import convert_xgboost

from onnxmltools.convert.common.data_types import FloatTensorType as OXFloatTensorType

from matplotlib import pyplot as plt

DATA_CSV = "cicids_clean.csv" FEAT_13_TXT = "features_13.txt" OUT_DIR = Path("reports_nonleaky_13") SEED = 42

def load_feature_list() -> list[str]: """ Φορτώνει τα 13 feature names. - Αν υπάρχει features_13.txt -> το χρησιμοποιεί - Αλλιώς -> παίρνει 13 από models_feature_names.txt (χωρίς label/Unnamed_0) και γράφει features_13.txt """ p = Path(FEAT_13_TXT) if p.exists(): feats = [l.strip() for l in p.read_text().splitlines() if l.strip()] else: base = [ l.strip() for l in Path("models_feature_names.txt").read_text().splitlines() if l.strip() ] feats = [c for c in base if c.lower() != "label" and c != "Unnamed_0"][:13] p.write_text("\n".join(feats) + "\n")

if len(feats) != 13:

    raise RuntimeError(f"Expected 13 features, got {len(feats)}: {feats}")


return feats


def chrono_split_read( csv_path: str, feat_cols: list[str], chunk: int = 200_000, train_ratio: float = 0.8, ): """ Chronological split χωρίς timestamp: - 1ο pass: μέτρηση rows - 2ο pass: πρώτα 80% -> train, τελευταία 20% -> test """ use_cols = ["label"] + feat_cols

total = 0

for df in pd.read_csv(csv_path, usecols=use_cols, chunksize=chunk):

    total += len(df)


n_train = int(total * train_ratio)


Xtr, ytr, Xte, yte = [], [], [], []

seen = 0


for df in pd.read_csv(csv_path, usecols=use_cols, chunksize=chunk):

    Xc = df[feat_cols].astype("float32").values

    yc = df["label"].astype("int8").values
```

```
    still_train = max(0, n_train - seen)
    if still_train > 0:
        take = min(still_train, len(df))
        Xtr.append(Xc[:take])
        ytr.append(yc[:take])

        rem = len(df) - take
        if rem > 0:
            Xte.append(Xc[take:])
            yte.append(yc[take:])
    else:
        Xte.append(Xc)
        yte.append(yc)

    seen += len(df)

Xtr = np.vstack(Xtr) if Xtr else np.empty((0, len(feat_cols)), dtype=np.float32)
ytr = np.concatenate(ytr) if ytr else np.empty((0,), dtype=np.int8)
Xte = np.vstack(Xte) if Xte else np.empty((0, len(feat_cols)), dtype=np.float32)
yte = np.concatenate(yte) if yte else np.empty((0,), dtype=np.int8)

return Xtr, ytr, Xte, yte, total

def balance_train( X: np.ndarray, y: np.ndarray, seed: int = SEED, cap_per_class: int = 150_000, ):
"""Balance μόνο στο TRAIN: κρατάει έως cap_per_class δείγματα ανά κλάση."""" rng = np.random.default_rng(seed) Xb, yb = [], []
for lbl in np.unique(y):
    idx = np.where(y == lbl)[0]
    if len(idx) > cap_per_class:
        idx = rng.choice(idx, size=cap_per_class, replace=False)

    Xb.append(X[idx])
    yb.append(y[idx])
```

```
return np.vstack(Xb), np.concatenate(yb)


def save_confmat_png(cm: np.ndarray, labels: list[str], title: str, out_png: Path): fig = plt.figure(figsize=(6, 5)) im = plt.imshow(cm, interpolation="nearest") plt.title(title) plt.xticks(range(len(labels)), labels, rotation=45, ha="right") plt.yticks(range(len(labels)), labels)
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        plt.text(j, i, int(cm[i, j]), ha="center", va="center")


plt.colorbar(im, fraction=0.046, pad=0.04)
plt.tight_layout()
fig.savefig(out_png, dpi=180)
plt.close(fig)

def main(): np.random.seed(SEED)
feats = load_feature_list()
OUT_DIR.mkdir(exist_ok=True, parents=True)

(OUT_DIR / "run_meta.json").write_text(
    json.dumps(
        {"seed": SEED, "features": feats, "data": DATA_CSV, "split": "chronological 80/20"},
        indent=2,
        ensure_ascii=False,
    )
)

# Chrono split (χωρίς leakage)
print("[INFO] chrono split loading…", file=sys.stderr)
Xtr, ytr, Xte, yte, total = chrono_split_read(DATA_CSV, feats, chunk=200_000, train_ratio=0.8)
print(f"[INFO] total rows={total} | train={len(ytr)} test={len(yte)}", file=sys.stderr)

# Balance μόνο στο train (κρατάμε natural test)
Xtr_b, ytr_b = balance_train(Xtr, ytr, cap_per_class=150_000)
```

```
# ===== Random Forest =====
rf = RandomForestClassifier(n_estimators=200, max_depth=None, n_jobs=2, random_state=SEED)
rf.fit(Xtr_b, ytr_b)

pr = rf.predict(Xte)
rf_f1 = f1_score(yte, pr)

print("\n==== RF (non-leaky, 13 feats) ====")
print("F1=", rf_f1)
print(classification_report(yte, pr, digits=4))

(OUT_DIR / "rf_report.txt").write_text(
    "non_leaky_rf_13\nF1={:.6f}\n\n".format(rf_f1) + classification_report(yte, pr, digits=4)
)

cm_rf = confusion_matrix(yte, pr, labels=[0, 1])
print("Confusion Matrix (RF) [rows=true, cols=pred]:")
print(cm_rf)

save_confmat_png(cm_rf, ["Benign", "Attack"], "RF Confusion Matrix", OUT_DIR / "rf_cm.png")

try:
    rf_auc = roc_auc_score(yte, rf.predict_proba(Xte)[:, 1])
    (OUT_DIR / "rf_auc.txt").write_text(f"{rf_auc:.6f}\n")
except Exception:
    pass

# ONNX export (RF)
Path("models").mkdir(exist_ok=True)
rf_onnx="models/rf_13_nonleaky.onnx"
rf_onnx_model = convert_sklearn(
    rf, initial_types=[("input", SKLFloatTensorType([None, Xtr.shape[1]]))]
```

```
)
onnx.save_model(rf_onnx_model, rf_onnx)

# ===== XGBoost =====
xgb = XGBClassifier(
    n_estimators=200,
    max_depth=6,
    learning_rate=0.1,
    subsample=0.8,
    colsample_bytree=0.8,
    tree_method="hist",
    n_jobs=2,
    objective="binary:logistic",
    base_score=0.5,
    random_state=SEED,
    eval_metric="logloss",
)
xgb.fit(Xtr_b, ytr_b, eval_set=[(Xte, yte)], verbose=False)

px = (xgb.predict_proba(Xte)[:, 1] >= 0.5).astype(int)
xgb_f1 = f1_score(yte, px)

print("\n==== XGB (non-leaky, 13 feats) ====")
print("F1=", xgb_f1)
print(classification_report(yte, px, digits=4))

(OUT_DIR / "xgb_report.txt").write_text(
    "non_leaky_xgb_13\nF1={:.6f}\n\n".format(xgb_f1) + classification_report(yte, px, digits=4)
)

cm_xgb = confusion_matrix(yte, px, labels=[0, 1])
print("Confusion Matrix (XGB) [rows=true, cols=pred]:")
print(cm_xgb)
```

```
save_confmat_png(cm_xgb, ["Benign", "Attack"], "XGB Confusion Matrix", OUT_DIR / "xgb_cm.png")

try:
    xgb_auc = roc_auc_score(yte, xgb.predict_proba(Xte)[:, 1])
    (OUT_DIR / "xgb_auc.txt").write_text(f"{xgb_auc:.6f}\n")
except Exception:
    pass

# ONNX export (XGB)
xgb_onnx = "models/xgb_13_nonleaky.onnx"
xgb_onnx_model = convert_xgboost(
    xgb, initial_types=[("input", OXFloatTensorType([None, Xtr.shape[1]]))]
)
onnx.save_model(xgb_onnx_model, xgb_onnx)

# feature schema για endpoint
schema = {"feature_names": feats, "dtype": "float32"}
(Path("models") / "feature_schema_13.json").write_text(
    json.dumps(schema, indent=2, ensure_ascii=False)
)

print("\n[OK] Saved:")
print(" - reports:", OUT_DIR)
print(" - ONNX RF:", rf_onnx)
print(" - ONNX XGB:", xgb_onnx)
print(" - feature schema:", "models/feature_schema_13.json")

if name == "main": main()
```

ΠΑΡΑΡΤΗΜΑ Γ: predictor_onnx.py

```
from flask import Flask, request, jsonify
```

import onnxruntime as ort

import numpy as np import os

import traceback

import argparse

import socket

PROVIDERS = ["CPUExecutionProvider"]

app = Flask(name)

def load_session(path: str): if not path or not os.path.exists(path): raise FileNotFoundError(f"ONNX not found: {path}") sess = ort.InferenceSession(path, providers=PROVIDERS) in_name = sess.get_inputs()[0].name return sess, in_name

def as_list(x): return x.tolist() if hasattr(x, "tolist") else x

def infer(sess, input_name, X): outputs = sess.run(None, {input_name: X}) out_json = {} for i, out in enumerate(outputs): out_json[f"output_{i}"] = as_list(out) preds = out_json.get("output_0") return preds, out_json

@app.route("/health", methods=["GET"]) def health(): return jsonify({ "ok": True, "expected_features": app.config["EXPECTED_FEATURES"], "rf_loaded": app.config["RF_SESS"] is not None, "xgb_loaded": app.config["XGB_SESS"] is not None, "default_model": app.config["DEFAULT_MODEL"], "port": app.config["PORT"], "hostname": socket.gethostname(), }), 200

@app.route("/predict", methods=["POST"]) def predict(): try: payload = request.get_json(force=True, silent=False) if not payload or "instances" not in payload: return jsonify(error="Missing 'instances'"), 400

```
  # model selection
   model = payload.get("model")
   if model is None or str(model).strip() == "":
      model = app.config["DEFAULT_MODEL"]


   model = str(model).strip().lower()
   if model not in ("rf", "xgb"):
      return jsonify(error=f"Invalid model='{model}'. Use 'rf' or 'xgb'."), 400


   X = np.asarray(payload["instances"], dtype=np.float32)
   if X.ndim != 2:
      return jsonify(error=f"instances ndim={X.ndim}, expected 2"), 400


   exp = app.config["EXPECTED_FEATURES"]
```

```
    if X.shape[1] != exp:
        return jsonify(error=f"instances have {X.shape[1]} features, expected {exp}"), 400

    if model == "rf":
        sess = app.config["RF_SESS"]
        input_name = app.config["RF_IN"]
    else:
        sess = app.config["XGB_SESS"]
        input_name = app.config["XGB_IN"]

    if sess is None:
        return jsonify(error=f"Model '{model}' not loaded"), 500

    preds, raw = infer(sess, input_name, X)
    return jsonify(model=model, predictions=preds, raw=raw), 200

except Exception as e:
    return jsonify(error=str(e), traceback=traceback.format_exc()), 500

def main(): parser = argparse.ArgumentParser() parser.add_argument("--host", default="0.0.0.0") parser.add_argument("--port", type=int, required=True) parser.add_argument("--rf", required=True, help="Path to RF ONNX") parser.add_argument("--xgb", required=True, help="Path to XGB ONNX") parser.add_argument("--expected_features", type=int, default=13) args = parser.parse_args()
rf_sess, rf_in = load_session(args.rf)
xgb_sess, xgb_in = load_session(args.xgb)

# default model per port (backward compatible payload without "model")
if args.port == 9001:
    default_model = "rf"
elif args.port == 9002:
    default_model = "xgb"
else:
    default_model = "rf"
```

app.config["RF_SESS"] = rf_sess

app.config["RF_IN"] = rf_in

app.config["XGB_SESS"] = xgb_sess

app.config["XGB_IN"] = xgb_in

app.config["EXPECTED_FEATURES"] = args.expected_features

app.config["DEFAULT_MODEL"] = default_model

app.config["PORT"] = args.port

app.run(host=args.host, port=args.port)

if name == "main": main()

ΠΑΡΑΡΤΗΜΑ Δ: flask_alert_collector.py

from flask import Flask, request, jsonify import datetime import argparse

app = Flask(**name**)

@app.route("/health", methods=["GET"]) def health(): return jsonify(ok=True), 200

@app.route("/hook/cicids", methods=["POST"]) def hook_cicids(): payload = request.get_json(silent=True) ts = datetime.datetime.now().isoformat(timespec="seconds") print(f"[{ts}] webhook /hook/cicids -> {payload}") return jsonify(ok=True, received=True), 200

if **name** == "**main**": parser = argparse.ArgumentParser() parser.add_argument("--host", default="0.0.0.0") parser.add_argument("--port", type=int, default=9000) args = parser.parse_args() app.run(host=args.host, port=args.port)

ΠΑΡΑΡΤΗΜΑ Ε: batch_eval_pairwise.py

#!/usr/bin/env python3

-- coding: utf-8 --

import csv

import datetime as dt

import json import ssl from pathlib

import Path from typing

import Dict, List, Tuple, Optional

import numpy as np

import onnxruntime as ort

```
import requests
import urllib3
----------
0) ΒΑΣΙΚΕΣ ΡΥΘΜΙΣΕΙΣ (EDIT HERE)
----------
OpenSearch endpoint (self-signed -> verify=False)
OS_URL = "https://localhost:9200" OS_AUTH = ("admin", "CHANGE_ME") # Καλό: βάλε το από env/.env και όχι hard-coded INDEX_NAME = "predictions_pairwise"
Paths
CSV_PATH = "/home/xariskotsis/opensearch-clean/migrate-v3/data/CICIDS2019_13f_norm_clean.csv" LABEL_COL = "Label" # ή "label" ανάλογα με το CSV
RF_ONNX = "models/rf_13_nonleaky.onnx" XGB_ONNX = "models/xgb_13_nonleaky.onnx"
Bulk settings
BATCH_SIZE = 1000
Runtime providers
PROVIDERS = ["CPUExecutionProvider"]
----------
1) TLS / warnings (self-signed)
----------
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
ssl._create_default_https_context = ssl._create_unverified_context
session = requests.Session() session.verify = False session.auth = OS_AUTH
----------
2) OpenSearch helpers


def ensure_index_exists(index_name: str) -> None: """ Δημιουργεί τον index αν δεν υπάρχει. Χρησιμοποιεί ένα απλό mapping ώστε dashboards/queries να λειτουργούν ομαλά. """ r = session.get(f"{OS_URL}/{index_name}") if r.status_code == 200: return
mapping = {
  "settings": {
    "index": {
      "number_of_shards": 1,
      "number_of_replicas": 0
    }
```

```
    },
    "mappings": {
      "properties": {
        "@timestamp": {"type": "date"},
        "label": {"type": "integer"},
        "pred_rf": {"type": "integer"},
        "pred_xgb": {"type": "integer"},
        "score_rf": {"type": "float"},
        "score_xgb": {"type": "float"},
        "agree": {"type": "boolean"},
        "mismatch": {"type": "boolean"},
        "source_id": {"type": "keyword"},
        "source_index": {"type": "keyword"},
      }
    }
}

resp = session.put(
    f"{OS_URL}/{index_name}",
    headers={"Content-Type": "application/json"},
    data=json.dumps(mapping),
)
resp.raise_for_status()

def bulk_index(index_name: str, docs: List[dict]) -> None: """Αποστολή docs στο OpenSearch με _bulk (NDJSON)."""  if not docs: return
lines = []
for d in docs:
    lines.append(json.dumps({"index": {"_index": index_name}}))
    lines.append(json.dumps(d, ensure_ascii=False))

data = "\n".join(lines) + "\n"
resp = session.post(
```

```
    f"{OS_URL}/_bulk",
    data=data,
    headers={"Content-Type": "application/x-ndjson"},
)
resp.raise_for_status()


-----------------------------
3) ONNX helpers
-----------------------------
def load_model(path: str) -> Tuple[ort.InferenceSession, str]: p = Path(path) if not p.exists(): raise FileNotFoundError(f"ONNX model not found: {path}")
sess = ort.InferenceSession(str(p), providers=PROVIDERS)
input_name = sess.get_inputs()[0].name
return sess, input_name


def predict_one( sess: ort.InferenceSession, input_name: str, features_vec: List[float], ) -> Tuple[int, float]: """ features_vec: λίστα 13 floats. Επιστρέφει: (pred_label, prob_attack)
Σημείωση: Τα outputs μπορεί να διαφέρουν ανά export.
Χειριζόμαστε:
- predicted labels στο output_0
- probs είτε dict είτε array στο output_1
"""
x = np.array([features_vec], dtype=np.float32)
outputs = sess.run(None, {input_name: x})

# output_0: predicted label
pred = int(outputs[0][0])

# output_1: probabilities
prob_attack = 0.0
if len(outputs) > 1:
    probs = outputs[1][0]
    if isinstance(probs, dict):
        # keys μπορεί να είναι "1" ή 1
```

```
        prob_attack = float(probs.get("1", probs.get(1, 0.0)))
    else:
        # π.χ. array [p0, p1]
        try:
            prob_attack = float(probs[1])
        except Exception:
            prob_attack = 0.0

return pred, prob_attack


----------
4) Metrics helpers
----------
def update_conf(m: Dict[str, int], y_true: int, y_pred: int) -> None: if y_true == 1 and y_pred == 1: m["tp"] += 1 elif y_true == 0 and y_pred == 1: m["fp"] += 1 elif y_true == 0 and y_pred == 0: m["tn"] += 1 elif y_true == 1 and y_pred == 0: m["fn"] += 1
def compute_scores(m: Dict[str, int]) -> Dict[str, float]: tp, fp, tn, fn = m["tp"], m["fp"], m["tn"], m["fn"] total = tp + fp + tn + fn
acc = (tp + tn) / total if total > 0 else 0.0
prec = tp / (tp + fp) if (tp + fp) > 0 else 0.0
rec = tp / (tp + fn) if (tp + fn) > 0 else 0.0

if prec + rec > 0:
    f1 = 2 * prec * rec / (prec + rec)
else:
    f1 = 0.0

return {
    "tp": float(tp),
    "fp": float(fp),
    "tn": float(tn),
    "fn": float(fn),
    "accuracy": float(acc),
    "precision": float(prec),
```

```
    "recall": float(rec),
    "f1": float(f1),
    "support": float(total),
}


----------------------------
5) CSV helpers
----------------------------
```

def load_header(csv_path: str) -> List[str]: with open(csv_path, "r", newline="") as f: reader = csv.reader(f) header = next(reader) return header

def infer_feature_order(csv_path: str, label_col: str) -> List[str]: """ Παίρνουμε τη σειρά των feature columns κατευθείαν από το CSV: όλα τα πεδία εκτός από το LABEL_COL. Έτσι διασφαλίζουμε ότι η σειρά ταιριάζει με αυτή που χρησιμοποιήθηκε στο training. """ header = load_header(csv_path)

```
if label_col not in header:
    raise KeyError(
        f"Δεν βρέθηκε η στήλη label '{label_col}' στο CSV. "
        f"Βρέθηκαν: {header}"
    )


feats = [c for c in header if c != label_col]
if len(feats) != 13:
    # Αν δεν είναι 13, πιθανότατα έχεις extra columns (π.χ. Unnamed: 0)
    # ή δεν είναι το σωστό CSV.
    raise ValueError(
        f"Expected 13 feature columns, got {len(feats)}. "
        f"Features: {feats}"
    )
return feats


----------------------------
6) Main
----------------------------
```

def main() -> None: # 6.1 Validate paths if not Path(CSV_PATH).exists(): raise FileNotFoundError(f"CSV not found: {CSV_PATH}")

```
# 6.2 Ensure index
ensure_index_exists(INDEX_NAME)

# 6.3 Feature order from CSV
features_order = infer_feature_order(CSV_PATH, LABEL_COL)
print("Features order (CSV):", features_order)

# 6.4 Load ONNX models
rf_sess, rf_input = load_model(RF_ONNX)
xgb_sess, xgb_input = load_model(XGB_ONNX)

# 6.5 Metrics accumulators
metrics = {
    "rf": {"tp": 0, "fp": 0, "tn": 0, "fn": 0},
    "xgb": {"tp": 0, "fp": 0, "tn": 0, "fn": 0},
}

docs_buffer: List[dict] = []
total_rows = 0

# 6.6 Read CSV & predict
with open(CSV_PATH, "r", newline="") as f:
    reader = csv.DictReader(f)

    for row in reader:
        total_rows += 1

        if total_rows % 5000 == 0:
            print(f"Processed {total_rows} rows...", flush=True)

        # label: 0 ή 1
        try:
            y_true = int(row[LABEL_COL])
```

```
except KeyError:
    raise KeyError(
        f"Δεν βρέθηκε η στήλη label '{LABEL_COL}' στο CSV. "
        "Άλλαξέ την στο script."
    )
except ValueError:
    raise ValueError(
        f"Μη έγκυρη τιμή label στη γραμμή {total_rows}: {row.get(LABEL_COL)}"
    )

# features στην ίδια σειρά με features_order
try:
    feats = [float(row[name]) for name in features_order]
except KeyError as e:
    raise KeyError(
        f"Λείπει feature column {e} στη γραμμή {total_rows}. "
        f"Expected columns: {features_order}"
    )
except ValueError as e:
    raise ValueError(
        f"Μη έγκυρη τιμή feature στη γραμμή {total_rows}: {e}"
    )

# RF prediction
pred_rf, score_rf = predict_one(rf_sess, rf_input, feats)

# XGB prediction
pred_xgb, score_xgb = predict_one(xgb_sess, xgb_input, feats)

# metrics update
update_conf(metrics["rf"], y_true, pred_rf)
update_conf(metrics["xgb"], y_true, pred_xgb)
```

```
        agree = (pred_rf == pred_xgb)
        mismatch = (pred_rf != pred_xgb)

        # doc για OpenSearch
        doc = {
            "@timestamp": dt.datetime.utcnow().strftime("%Y-%m-%dT%H:%M:%S.%f")[:-3] + "Z",
            "label": y_true,
            "pred_rf": pred_rf,
            "pred_xgb": pred_xgb,
            "score_rf": float(score_rf),
            "score_xgb": float(score_xgb),
            "agree": bool(agree),
            "mismatch": bool(mismatch),
            # Προαιρετικά πεδία συσχέτισης (αν τα έχεις διαθέσιμα):
            "source_index": "cicids_2019_13f",
            "source_id": str(total_rows),
        }

        docs_buffer.append(doc)

        if len(docs_buffer) >= BATCH_SIZE:
            bulk_index(INDEX_NAME, docs_buffer)
            docs_buffer = []

# 6.7 Send remaining docs
if docs_buffer:
    bulk_index(INDEX_NAME, docs_buffer)

print(f"\nProcessed rows: {total_rows}")

# 6.8 Final scores
rf_scores = compute_scores(metrics["rf"])
xgb_scores = compute_scores(metrics["xgb"])
```

```
print("\n=== Random Forest metrics (class=attack=1) ===")
for k, v in rf_scores.items():
    print(f"{k}: {v}")

print("\n=== XGBoost metrics (class=attack=1) ===")
for k, v in xgb_scores.items():
    print(f"{k}: {v}")

if name == "main": main()
```

ΠΑΡΑΡΤΗΜΑ ΣΤ: traditional_baselines_13.py

```
from future import annotations
import numpy as np
import pandas as pd
from pathlib import Path
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score, precision_recall_fscore_support
DATA_CSV = "cicids_clean.csv" FEAT_13_TXT = "features_13.txt" SEED = 42
def load_feature_list():
p = Path(FEAT_13_TXT)
if p.exists():
feats = [l.strip() for l in p.read_text().splitlines() if l.strip()]
else:
base = [l.strip() for l in Path("models_feature_names.txt").read_text().splitlines()] feats = [c for c in base if c.lower() != "label" and c != "Unnamed_0"][:13] p.write_text("\n".join(feats) + "\n")
return feats
def chrono_split_read(csv_path: str, feat_cols: list[str], chunk=200_000, train_ratio=0.8):
use_cols = ["label"] + feat_cols
total = 0
    for chunk_df in pd.read_csv(csv_path, usecols=use_cols, chunksize=chunk):
        total += len(chunk_df)
    n_train = int(total * train_ratio)
```

```
    Xtr, ytr, Xte, yte = [], [], [], []
    seen = 0
    for chunk_df in pd.read_csv(csv_path, usecols=use_cols, chunksize=chunk):
        Xc = chunk_df[feat_cols].astype("float32").values
        yc = chunk_df["label"].astype("int8").values

         still_train = max(0, n_train - seen)
        if still_train > 0:
            take = min(still_train, len(chunk_df))
             Xtr.append(Xc[:take]); ytr.append(yc[:take])
            rem = len(chunk_df) - take
             if rem > 0:
                Xte.append(Xc[take:]); yte.append(yc[take:])
            else:
                Xte.append(Xc); yte.append(yc)

            seen += len(chunk_df)

    Xtr = np.vstack(Xtr); ytr = np.concatenate(ytr)
    Xte = np.vstack(Xte); yte = np.concatenate(yte)
    return Xtr, ytr, Xte, yte, total


def print_metrics(tag: str, y_true: np.ndarray, y_pred: np.ndarray):
print(f"\n==== {tag} ====") cm = confusion_matrix(y_true, y_pred, labels=[0, 1])
    print("Confusion matrix [ [tn, fp], [fn, tp] ]:") print(cm) print("\nClassification report:")
    print(classification_report(y_true, y_pred, digits=4))
acc = accuracy_score(y_true, y_pred)
    p, r, f, _ = precision_recall_fscore_support(
        y_true, y_pred, average="binary", pos_label=1, zero_division=0
    )
    print("\nSummary (positive class = 1 = attack):")
    print(f"accuracy={acc:.6f} | precision={p:.6f} | recall={r:.6f} | f1={f:.6f}")
```

```
def baseline_zscore(Xtr, ytr, Xte):
mu = Xtr.mean(axis=0)
sigma = Xtr.std(axis=0)
sigma[sigma == 0] = 1.0
Zte = (Xte - mu) / sigma
        score_te = np.max(np.abs(Zte), axis=1)

        Ztr = (Xtr - mu) / sigma
        score_tr = np.max(np.abs(Ztr), axis=1)

        benign_tr = score_tr[ytr == 0]
        thr = 3.0 if len(benign_tr) == 0 else float(np.quantile(benign_tr, 0.99))
        y_pred = (score_te > thr).astype(int)
        return thr, y_pred

def baseline_mad(Xtr, ytr, Xte):
        # Robust center/scale: median & MAD (scaled)
med = np.median(Xtr, axis=0)
mad = np.median(np.abs(Xtr - med), axis=0)
mad[mad == 0] = 1.0
scale = 1.4826 * mad # consistency constant
Zte = (Xte - med) / scale
        score_te = np.max(np.abs(Zte), axis=1)

        Ztr = (Xtr - med) / scale
        score_tr = np.max(np.abs(Ztr), axis=1)

        benign_tr = score_tr[ytr == 0]
        thr = 3.5 if len(benign_tr) == 0 else float(np.quantile(benign_tr, 0.99))
        y_pred = (score_te > thr).astype(int)
        return thr, y_pred

def main():
```

```
np.random.seed(SEED)
feats = load_feature_list()
print("[INFO] Traditional baselines on non-leaky 13 features")
        print("[INFO] split: chronological 80/20")
        print(f"[INFO] features({len(feats)}): {feats}")
        Xtr, ytr, Xte, yte, total = chrono_split_read(DATA_CSV, feats, train_ratio=0.8)
        print(f"[INFO] total rows={total} | train={len(ytr)} test={len(yte)}")
        print("[INFO] label convention assumed: 0=benign, 1=attack")

        thr_z, pred_z = baseline_zscore(Xtr, ytr, Xte)
        print(f"\n[Z-SCORE] threshold = {thr_z:.6f} (99th percentile of TRAIN benign)")
        print_metrics("BASELINE 1: Z-score anomaly detector", yte, pred_z)

        thr_m, pred_m = baseline_mad(Xtr, ytr, Xte)
        print(f"\n[MAD] threshold = {thr_m:.6f} (99th percentile of TRAIN benign)")
        print_metrics("BASELINE 2: Robust MAD anomaly detector", yte, pred_m)

if name == "main":
main()
```