Εθνικό Μετσόβιο Πολυτεχνείο Αθηνών
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Ανίχνευση επιθέσεων DDoS σε υποδομές cloud και edge με χρήση τεχνικών Μηχανικής Μάθησης

Διπλωματική Εργασία

ΕΛΕΥΘΈΡΙΟΣ ΓΑΛΑΤΆΣ

Επιβλέπων: Εμμανουήλ Βαρβαρίγος
Καθηγητής, ΕΜΠ

Αθήνα, Ιούλιος 2023

Εθνικό Μετσόβιο Πολυτεχνείο Αθηνών
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών
Τομέας Επικοινωνιών, Ηλεκτρονικής και Συστημάτων Πληροφορικής

Ανίχνευση επιθέσεων DDoS σε υποδομές cloud και edge με χρήση τεχνικών Μηχανικής Μάθησης

Διπλωματική Εργασία

ΕΛΕΥΘΈΡΙΟΣ ΓΑΛΑΤΆΣ

Επιβλέπων: Εμμανουήλ Βαρβαρίγος
Καθηγητής, ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 1η Νοεμβρίου, 2023

.....	.....	.....
Εμμανουήλ Βαρβαρίγος Καθηγητής Ε.Μ.Π	Θεοδώρα Βαρβαρίγου Καθηγήτρια Ε.Μ.Π	Ηρακλής Αβραμόπουλος Καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2023

(Υπογραφή)

.....
Ελευθέριος Γαλατάς
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Περίληψη

Η αυξανόμενη ανάγκη σε χρήση υποδομών cloud και edge έχει οδηγήσει σε έντονη ανησυχία για την ασφάλειά τους έναντι Επιθέσεων Κατανεμημένης Άρνησης Υπηρεσίας (DDoS). Στο πλαίσιο αυτής της διπλωματικής, προτείνουμε μια ολοκληρωμένη προσέγγιση για την ασφάλεια των υποδομών cloud και edge μέσω της χρήσης τεχνικών ανίχνευσης και αντιμετώπισης τέτοιου είδους επιθέσεων, που βασίζονται στη Μηχανική Μάθηση.

Η διπλωματική διερευνά την χρήση διαφόρων αλγορίθμων ML για την ανίχνευση επιθέσεων DDoS, αναλύοντας την αποτελεσματικότητά τους στον εντοπισμό και την ταξινόμηση κακόβουλων μοτίβων διαδικτυακής κίνησης. Μέσα από εκτενείς αξιολογήσεις και συγκρίσεις απόδοσης, εντοπίζουμε τους καταλληλότερους αλγόριθμους ML για ανίχνευση DDoS σε περιβάλλοντα cloud και edge.

Για την αντιμετώπιση της υπολογιστικής επιβάρυνσης στα ML μοντέλα ανίχνευσης σε περιβάλλοντα με περιορισμένους πόρους, εισάγουμε μια ιδέα για την επιτάχυνση των λειτουργιών τους, απόλυτα συμβατή με υποδομές cloud και edge. Αυτή η τεχνική συνδυάζεται σαν προέκταση, βελτιώνοντας σημαντικά την υπολογιστική πολυπλοκότητα, διατηρώντας την υψηλή απόδοση των μοντέλων ML και επιτρέποντας την ταχεία και αποτελεσματική ανίχνευση σε πραγματικό χρόνο επιθέσεων DDoS. Μέσω αξιολογήσεων και προσομοιώσεων, αποδεικνύουμε την αποτελεσματικότητα της προτεινόμενης προσέγγισης στον εντοπισμό των επιθέσεων DDoS σε υποδομές cloud και edge.

Τέλος, προτείνουμε μια αρχιτεκτονική μοντέλου ML, αξιοποιώντας τεχνικές βαθιάς μάθησης, για να περαιτέρω βελτίωση της ακρίβειας ανίχνευσης DDoS. Αυτό το μοντέλο συνδυάζει εντοπισμό μοτίβων διαδικτυακής κίνησης και ανωμαλιών συμπεριφοράς, για να παρέχει μια ολοκληρωμένη εικόνα των πιθανών επιθέσεων και να ελαχιστοποιεί τα ψευδώς θετικά (False Positives).

Στόχος αυτής της διπλωματικής είναι να συμβάλλει στον τομέα της ασφάλειας του cloud και edge, παρέχοντας ένα ολοκληρωμένο πλαίσιο για τον εντοπισμό και την αντιμετώπιση επιθέσεων DDoS, αξιοποιώντας τη δύναμη της Μηχανικής Μάθησης. Οι προτεινόμενες τεχνικές ακουμπούν στις θεμελιωμένες μεθόδους ασφαλείας cloud και edge, αλλά ανοίγουν επίσης το δρόμο για εξελίξεις στην προστασία κρίσιμων συστημάτων από τις απειλές στον κυβερνοχώρο.

.

Λέξεις Κλειδιά

Υποδομές cloud και edge, Μηχανική Μάθηση, Επιθέσεις DDoS, Ανίχνευση και αντιμετώπιση, Αξιολόγηση απόδοσης, Περιορισμένοι υπολογιστικοί πόροι, Τεχνικές επιτάχυνσης, Βαθιά μάθηση, Ανωμαλίες Συμπεριφοράς, Ανίχνευση Πραγματικού Χρόνου

Abstract

The increasing adoption of cloud and edge computing infrastructures has led to a growing concern for their security against Distributed Denial of Service (DDoS) attacks. In this thesis, we propose a comprehensive approach to enhance the security of cloud and edge infrastructures through the use of Machine Learning-based detection and mitigation techniques.

The thesis investigates various ML algorithms for DDoS attack detection, analyzing their effectiveness in identifying and classifying malicious traffic patterns. Through extensive performance evaluations and comparisons, we identify the most suitable ML algorithms for DDoS detection in cloud and edge environments.

To address the computational challenge of ML detection models in resource-constrained environments, we introduce an idea for accelerating these operations, fully compatible with cloud and edge infrastructures. This technique serves as an extension, optimizing the computational complexity while maintaining high ML model performance, enabling rapid and efficient real-time detection of DDoS attacks. Through evaluations and simulations, we demonstrate the effectiveness of the proposed approach in DDoS attack detection within cloud and edge infrastructures.

Furthermore, we propose an ML model architecture leveraging deep learning techniques to further enhance the accuracy of DDoS detection. This model combines the detection of malicious traffic patterns and behavioral anomalies to provide a comprehensive view of potential attacks and minimize false positives.

The objective of this thesis is to contribute to the field of cloud and edge security by providing a comprehensive framework for DDoS attack detection and mitigation, leveraging the power of Machine Learning. The proposed techniques build upon established cloud and edge security methods while paving the way for advancements in safeguarding critical systems from cyber threats.

Keywords

Cloud computing, Edge computing, Security, Distributed Denial of Service (DDoS), Machine Learning, Detection, Mitigation, Traffic patterns, Performance evaluations, Resource-constrained environments, Acceleration techniques, Real-time detection, Deep learning, Cyber threats

Ευχαριστίες

Θα ήθελα καταρχάς να ευχαριστήσω τον καθηγητή κ. Εμμανουήλ Βαρβαρίγο για την επίβλεψη αυτής της διπλωματικής εργασίας. Επίσης, ευχαριστώ ιδιαίτερα τον υποψήφιο διδάκτορα Ιπποκράτη Σαρτζετάκη για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε καθ' όλη τη διάρκεια της διπλωματικής αυτής. Τέλος, θα ήθελα να ευχαριστήσω γονείς, συγγενείς και φίλους για την ηθική συμπαράσταση και βοήθεια που μου προσέφεραν όλα αυτά τα χρόνια και συνέβαλαν στην διεκπεραίωση των στόχων μου.

Ελευθέριος Γαλατάς

Αθήνα, 2023

Περιεχόμενα

Εισαγωγή

Αντικείμενο της εργασίας

Κατά την εποχή της ψηφιακής μετάβασης, το cloud και το edge έχουν εμφανιστεί ως κεντρικοί πυλώνες στον κόσμο της τεχνολογίας. Ο τρόπος με τον οποίον βλέπουμε τον κόσμο εδώ και δεκαετίες αλλάζει δραστικά στην εποχή της ψηφιοποίησης. Η ζωή μας εξαρτάται πλέον από υπολογιστές, το διαδίκτυο και διάφορες συσκευές Internet of Things (IoT), μέσα που προσφέρουν πληθώρα πολυσύνθετων και πολυδιάστατων δεδομένων, δημιουργώντας προκλήσεις στην διαχείριση τους, στην αποθήκευσή τους αλλά καθώς και στην ανάγνωση μέσα στον όγκο τους, πολύπλοκων μοτίβων.

Τα συστήματα υπολογιστικού νέφους (cloud) παίζουν καθοριστικό ρόλο στην κάλυψη αυτής της ολοένα και γρηγορότερα αυξανόμενης τεχνολογικής ζήτησης. Οι υπηρεσίες cloud προσφέρονται από απομονωμένους διακομιστές και αξιοποιούνται από πελάτες μέσω διαδικτύου. Με κυβερνήσεις και εταιρείες ανά τον κόσμο, να μεταφέρουν τις εγκαταστάσεις τους στο cloud, η ζήτηση σε τέτοιου είδους υπηρεσίες γίνεται ολοένα και πιο μεγάλη. Η αξία της αγοράς των δημόσιων υπηρεσιών cloud το 2019 ήταν περίπου 228 δισεκατομμύρια δολάρια παγκοσμίως και προβλέπεται να φτάσει τα 355 δισεκατομμύρια δολάρια μέχρι το 2022, με αύξηση 24% ετησίως.

Ένα σύστημα cloud αποτελείται από έναν τεράστιο πλήθος στοιχείων υλικού (hardware) και λογισμικού (software). Συνεπώς, η παρακολούθηση της «υγειούς» λειτουργίας αυτών των στοιχείων και η άμεση ανίχνευση οποιασδήποτε ανωμαλίας, είναι κρίσιμης σημασίας στην εξασφάλιση αξιόπιστης και αδιάλειπτης λειτουργίας των υπηρεσιών cloud, που αποτελεί και το βασικό χαρακτηριστικό της υπηρεσίας όσον αφορά την πλευρά των πελατών. Η παρακολούθηση γίνεται μέσω της συλλογής στοιχείων τηλεμετρίας σε μορφή καταγραφών (logs)[1], ίχνη εκτέλεσης, μετρήσεις κ.λπ. Συνήθως, οι καταγραφές παράγονται και συλλέγονται συνεχώς (24/7). Οι πηγές τους, ωστόσο, διαφέρουν (π.χ. καταγραφές υλικού, καταγραφές λειτουργικού συστήματος και καταγραφές εφαρμογών), αντικατοπτρίζοντας, έτσι, την ιδιομορφία των δεδομένων.

Η αυξανόμενη πολυπλοκότητα, ταχύτητα, ποικιλία και όγκος των καταγραφών που παράγονται από τα διάφορα στοιχεία του cloud καθιστούν την ανάλυση τους μια πρόκληση Μεγάλων Δεδομένων (Big Data challenge). Σε ένα περιβάλλον cloud, οι τεχνικές ανίχνευσης ανωμαλιών (anomaly detection), που εξαρτώνται από σχεδόν πλήρως στατιστικές και ευρετικές μετρικές, είναι λιγότερο αποτελεσματικές λόγω της κλίμακας των πλατφόρμων cloud και της μεταβαλλόμενης φύσης των φόρτων εργασίας που εκτελούνται σε αυτές. Εδώ, υπεισέρχεται και η ανάγκη για την εφαρμογή τεχνικών Μηχανικής Μάθησης (ML) αντί για κλασικές μαθηματικές μεθόδους, η οποίες αποδεικνύονται ιδιαίτερα χρήσιμες.

Ένα σύστημα διαχείρισης δικτύου με χρήση τεχνικών ML μπορεί να υλοποιηθεί σε ένα κεντρικό σύστημα στο cloud, το οποίο σημαίνει ότι τόσο η κατασκευή μοντέλου όσο και η εξαγωγή συμπερασμάτων γίνονται σε μια μοναδική τοποθεσία, όπως ένα κέντρο δεδομένων ή ένα site νέφους. Ωστόσο, η χρήση του κεντρικού νέφους για τον εντοπισμό ανωμαλιών μπορεί να προκαλέσει μεγάλη καθυστέρηση (latency), καθ’ ότι σε αυτή την προσέγγιση, τα δεδομένα αρχικά, συλλέγονται από διάφορους αισθητήρες συνόλου του δικτύου σε μια κεντρική τοποθεσία, κάτι που απαιτεί μεγάλο εύρους ζώνης δικτύου (bandwidth), και εν συνεχεία, μετά την επεξεργασία τους στην κεντρική τοποθεσία cloud, το σήμα ελέγχου πρέπει να σταλεί πάλι πίσω στις τελικές συσκευές.

Έτσι γεννάται η ιδέα υλοποίησης ενός συστήματος διαχείρισης δικτύου με χρήση ML σε μια αρχιτεκτονική υπολογισμού στην άκρη (edge computing). Σε αυτήν την τεχνική edge intelligence, ενώ η κατασκευή μοντέλου πραγματοποιείται σε μια κεντρική τοποθεσία στο cloud, σε αντίθεση με την κεντρική προσέγγιση, η απόδοση γίνεται στις συσκευές άκρης (edge) αντί της κεντρικής τοποθεσίας. Το αποτέλεσμα είναι η μείωση καθυστέρησης μετάδοσης, δίνοντας την δυνατότητα σε επεξεργασία πραγματικού χρόνου για εφαρμογές για ανίχνευση και απόκριση σε ανωμαλίες.

Συνοψίζοντας, στο πλαίσιο αυτής της εργασίας τα μοντέλα μηχανικής μάθησης που θα εξεταστούν, έχουν σκοπό την ένταξη τους σε μία αρχιτεκτονική προσανατολισμένη προς την μεριά του edge, με σκοπό την γρήγορη και αποτελεσματική ανίχνευση ανωμαλιών σε real time περιβάλλοντα.

Δομή της εργασίας

1. Cloud & Edge computing

Σε αυτήν την ενότητα, θα περιγράφουν συνοπτικά οι τεχνολογίες Cloud Computing και Edge Computing. Θα αναφερθούν, ακόμα, οι βασικές αρχές, τα πλεονεκτήματα και οι περιορισμοί της κάθε τεχνολογίας, καθώς και παραδείγματα για την χρήση τους. Επίσης, θα εξηγηθούν οι διαφορές μεταξύ τους, καθώς όμως και η συνεργασία τους για την παροχή αποτελεσματικών υπηρεσιών.

2. DDoS σε Cloud & Edge υποδομές

Σε αυτήν την ενότητα, θα εξεταστούν οι DDoS επιθέσεις (Distributed Denial of Service) και ο τρόπος με τον οποίον επηρεάζουν τις υποδομές του Cloud & Edge Computing. Θα αναλυθούν οι κύριοι τύποι επιθέσεων σε δομές cloud και edge και οι αντίστοιχες επιπτώσεις τους. Ακόμα, θα αναφερθούν παραδείγματα πραγματικών επιθέσεων που έχουν συμβεί σε μεγάλες εταιρείες και υπηρεσίες, και πώς οι DDoS επιθέσεις μπορούν να προκαλέσουν σοβαρά προβλήματα ασφάλειας και αποδοτικότητας.

3. Αλγόριθμοι για ανίχνευση DDoS επιθέσεων

Σε αυτήν την ενότητα, θα εξετάσουμε διάφορους αλγορίθμους που χρησιμοποιούνται για την ανίχνευση των επιθέσεων DDoS σε υποδομές Cloud & Edge. Θα περιγράφουν οι διάφορες κατηγορίες μεθόδων μηχανικές μάθησης (supervised / unsupervised), καθώς και οι επικρατέστερες μέθοδοι για την ανίχνευση τέτοιου είδους επιθέσεων, Ακόμα, θα αναλυθεί η αποτελεσματικότητα των αλγορίθμων και οι προκλήσεις που αντιμετωπίζουν κατά την αντιμετώπιση επιθέσεων DDoS.

4. Υλοποίηση – Προσομοιώσεις – Συγκρίσεις

Σε αυτήν την ενότητα, θα παρουσιάσουμε διάφορες υλοποιήσεις αλγορίθμων ανίχνευσης DDoS σε υποδομές Cloud & Edge Computing. Θα περιγράψουμε τον τρόπο υλοποίησης αυτών των αλγόριθμων, και εν συνέχεια, θα

παρουσιάσουμε τα αποτελέσματα από τις υλοποιήσεις, παρέχοντας στατιστικά δεδομένα και γραφήματα που δείχνουν την αποτελεσματικότητα τους στην ανίχνευση DDoS επιθέσεων. Θα συγκρίνουμε τα αποτελέσματα μεταξύ διαφορετικών αλγορίθμων και θα αναλύσουμε τις επιδόσεις τους σε συγκεκριμένα σενάρια και φορτία εργασίας. Στο τμήμα αυτό, θα εστιάσουμε στις διαφορές και τα πλεονεκτήματα κάθε υλοποίησης, προκειμένου να κατανοήσουμε τις πρακτικές επιπτώσεις της επιλογής αλγορίθμων ανίχνευσης DDoS σε υποδομές Cloud & Edge Computing.

5. Συμπεράσματα

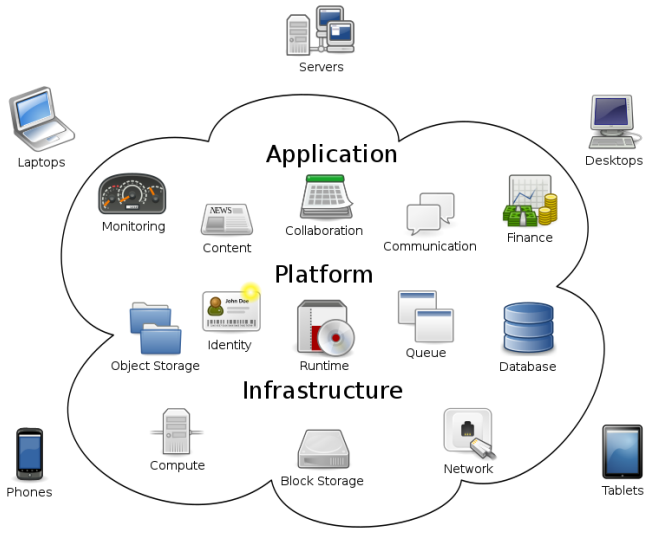
Στο τελευταίο μέρος, θα παρουσιάσουμε τα συμπεράσματα που προκύπτουν από την ανάλυση που έχουμε διεκπεραιώσει. Επισημαίνονται οι προκλήσεις και οι πιθανές βελτιώσεις στον τομέα της ανίχνευσης DDoS επιθέσεων σε υποδομές cloud και edge computing και πώς οι αλγόριθμοι ανίχνευσης μπορούν να συνδράμουν στην ενίσχυση της ασφάλειας. Θα συζητήσουμε πιθανές μελλοντικές εξελίξεις και βελτιώσεις για την προστασία από DDoS επιθέσεις, και πώς η συνεργασία μεταξύ Cloud & Edge μπορεί να βελτιώσει την απόδοση και την ασφάλεια των υπηρεσιών.

Κεφάλαιο 1

Cloud και Edge computing

1.1 Cloud computing

Το cloud computing αποτελεί μία σύγχρονη τεχνολογία της οποίας ο ορισμός περιλαμβάνει την παροχή υπολογιστικών υπηρεσιών μέσω του διαδικτύου. Έτσι, αντί για την αποθήκευση και χρήση, εν ολίγοις «φιλοξενία», εφαρμογών, βάσεων δεδομένων ή υπολογιστικών πόρων σε τοπικούς server ή σε προσωπικές συσκευές, το cloud computing προσφέρει την δυνατότητα στους χρήστες να έχουν πρόσβαση και δυνατότητα χρήσης των παραπάνω υπηρεσιών, εξ αποστάσεως μέσω του διαδικτύου.



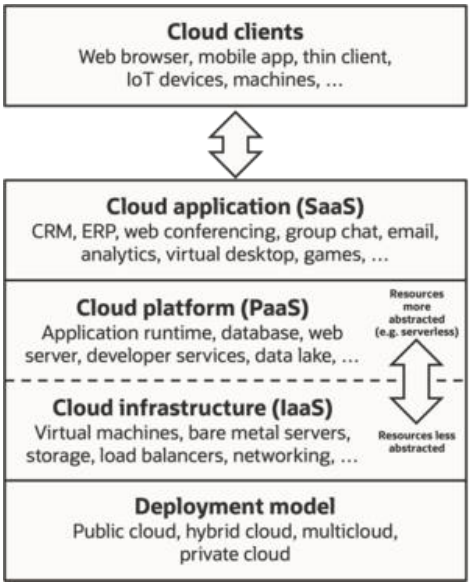
Εικόνα 1 Μια "μεταφορική" απεικόνιση του cloud

Το cloud computing σαν ιδέα, πάει πίσω αρκετές δεκαετίες, αλλά ο ορισμός του, όπως το ξέρουμε σήμερα και όπως ξεκίνησε να γίνεται ευρέως γνωστό, τοποθετείται στα μέσα της δεκαετίας 2000 με 2010. Τις δεκαετίες του ’60 και ’70, συνελήφθη η ιδέα του utility computing, ο «πρόγονος» του cloud computing, με τους ερευνητές να επιδιώκουν την παροχή υπολογιστικών υπηρεσιών σαν δημόσια υπηρεσία, όπως το ρεύμα. Την δεκαετία του ’90, οι εταιρίες τηλεπικοινωνιών άρχισαν να προσφέρουν VPNs (Virtual Private Networks) και εικονικούς ιδιωτικούς server, θέτοντας τα πραγματικά θεμέλια για το cloud computing. Το 2002. η Amazon παρουσίασε το Amazon Web Services (AWS), το οποίο προσέφερε αποθηκευτικό χώρο και υπολογιστικές υπηρεσίες σε περιβάλλον νέφους (cloud). Με την δημοτικότητα του cloud να αυξάνεται, το 2006 η AWS παρουσίασε το Elastic Compute Cloud (EC2), επιτρέποντας πλέον στους χρήστες να νοικιάζουν κατ’ απαίτηση εικονικές μηχανές. Μέχρι το τέλος της δεκαετίας του 2000, και άλλοι τεχνολογικοί τιτάνες εισήλθαν στην αγορά του cloud computing, όπως η Google και η Microsoft, με τις cloud υπηρεσίες Google App Engine και Microsoft Azure, αντίστοιχα. Από το 2010 έως και σήμερα, το cloud computing έχει αναπτυχθεί ραγδαία, με περισσότερους προμηθευτές και διαρκώς εξελισσόμενες υπηρεσίες και τεχνολογίες, κάνοντας το αναπόσπαστο κομμάτι των πρακτικών στις επιχειρήσεις και της σύγχρονης τεχνολογίας[2].

1.2 Μοντέλα cloud computing

Το cloud computing χωρίζεται τυπικά σε τρία βασικά μοντέλα υπηρεσιών:

1. **Software-as-a-Service (SaaS):** Το μοντέλο «Λογισμικό σαν Υπηρεσία» περιγράφει την δυνατότητα παροχής πρόσβασης στους χρήστες σε εφαρμογές λογισμικού στο cloud. Έτσι, οι εκάστοτε χρήστες, μπορούν να χρησιμοποιούν αυτές τις εφαρμογές μέσω του διαδικτύου (web browser), χωρίς να υπάρχει η ανάγκη σε κατέβασμα, εγκατάσταση και διατήρηση του λογισμικού τοπικά. Κάποια σημαντικά παραδείγματα είναι το Google Workspace, το Microsoft 365, το Salesforce, και το Dropbox. Τα πλεονεκτήματα του μοντέλου έχουν να κάνουν με την εύκολη προσβασιμότητα, όπως εξηγήθηκε παραπάνω από την ίδια την φύση του SaaS, αλλά και τις αυτόματες ενημερώσεις (updates), αφού οι ενημερώσεις λογισμικού και η συντήρηση των εφαρμογών είναι στα χέρια του προμηθευτή των υπηρεσιών (provider), εξασφαλίζοντας τα πιο σύγχρονα χαρακτηριστικά για την εφαρμογής και ενημερωμένες ασφάλειες. Τέλος, υπάρχει οικονομική αποδοτικότητα, αφού οι άδειες λογισμικού για κάθε χρήστη δεν είναι απαραίτητες και οι ανάγκες για IT συντήρηση είναι λιγότερες.



Εικόνα 2 Τα μοντέλα cloud computing τοποθετημένα σε στοίβα

2. **Platform-as-a-Service (PaaS**): Το μοντέλο «Πλατφόρμα σαν Υπηρεσία» προσφέρει την χρήση πλατφόρμας και περιβάλλοντος εργασίας για προγραμματιστές, ώστε να χτίσουν, να εφαρμόσουν και να διαχειριστούν εφαρμογές λογισμικού χωρίς, όμως, να υπάρχει η πολύπλοκη ανάγκη αντιμετώπισης υποκείμενης υποδομής. Αυτό επιτρέπει στους προγραμματιστές το γράψιμο κώδικα απρόσκοπτα, καθώς ο προμηθευτής αναλαμβάνει την διαχείριση του περιβάλλοντος εκτέλεσης, του λειτουργικού και του hardware. Γνωστά PaaS είναι τα Google App Engine, Microsoft Azure App Service, και Heroku. Από τα παραπάνω, εύκολα συμπεραίνεται πως ο χρόνος ανάπτυξης εφαρμογών μπορεί να μειωθεί δραματικά. Ακόμα, όσον αφορά το PaaS, οι πλατφόρμες έχουν την δυνατότητα να «επεκτείνουν» αυτόματα εφαρμογές κατ' απαίτηση, χωρίς ανάγκη για παρέμβαση.

3. **Infrastructure-as-a-Service (IaaS):** Το μοντέλο «Υποδομή σαν Υπηρεσία» προσφέρει την δυνατότητα ενοικίασης εικονικών μηχανών, χώρου αποθήκευσης και στοιχείων δικτύωσης κατ' απαίτηση, δίνοντας στους χρήστες περισσότερο έλεγχο στην υποκείμενη υποδομή, καθιστώντας το μοντέλο ιδανικό για επιχειρήσεις με συγκεκριμένες ανάγκες IT. Γνωστά IaaS είναι τα Amazon Web Services (AWS), Microsoft Azure και Google Cloud Platform. Τα οφέλη του μοντέλου IaaS, ειδικά για τις επιχειρήσεις, είναι ο πλήρης έλεγχος της ανάγκης σε επίπεδο υποδομής, το οποίο μεταφράζεται σε μείωση κόστους, με την μείωση ή αύξηση τους, ανάλογα με το φόρτο δουλειάς και την άρση της υποχρεωτικότητας αγοράς και διατήρησης φυσικού εξοπλισμού (hardware).

1.3 Βασικά πλεονεκτήματα του cloud

Το κάθε μοντέλο cloud computing προσφέρει διάφορα επίπεδα ελέγχου και αφορά συγκεκριμένα παραδείγματα περιπτώσεων χρήσης και χρηστών. Ωστόσο, το cloud computing πέρα από τις εξειδικευμένες υπηρεσίες που προσφέρει, έχει κάποια βασικά πλεονεκτήματα[3] που βοήθησαν στην διάδοση του.

- **Κλιμάκωση:** Είναι ιδιαίτερα χρήσιμη η ρύθμιση των αναγκών σε υπολογιστικούς πόρους, περισσοτέρους ή λιγότερους ανάλογα με το αν οι ανάγκες είναι αυξημένες ή μειωμένες, μειώνοντας το συνολικό κόστος, γλυτώνοντας την διαρκή απασχόληση των μέγιστων δυνατόν πόρων.

- **Οικονομία:** Δεν είναι πλέον απαραίτητη η κατάθεση μεγάλων κεφαλαίων, για την απόκτηση ακριβού εξοπλισμού hardware καθώς και των υψηλών κοστών συντήρησης και διατήρησης αυτών. Το μοντέλο με την διαρκή πληρωμή τύπου συνδρομή, βοηθάει στην διατήρηση κάλου προϋπολογισμού, ειδικότερα για μικρές εταιρίες ή startups.

- **Ελαστικότητα και Προσβασιμότητα:** Υπάρχει πάντα η δυνατότητα πρόσβασης από οποιαδήποτε συσκευή η οποία έχει πρόσβαση στο διαδίκτυο. Εξασφαλίζεται έτσι η ευκολία στην εργασία εξ αποστάσεως στην συνεργασία αλλά και στην πρόσβαση σε δεδομένα και εφαρμογές από παντού, δίνοντας μεγάλη «κινητικότητα».

- **Συντήρηση και Ενημερώσεις:** Οι προμηθευτές των υπηρεσιών cloud αναλαμβάνουν εξολοκλήρου τις ενημερώσεις, τις διορθώσεις και τις εργασίες συντήρησης, παίρνοντας το βάρος αυτό από τους «ώμους» των χρηστών, οι οποίοι μπορούν να χρησιμοποιούν την υπηρεσία απρόσκοπτα.

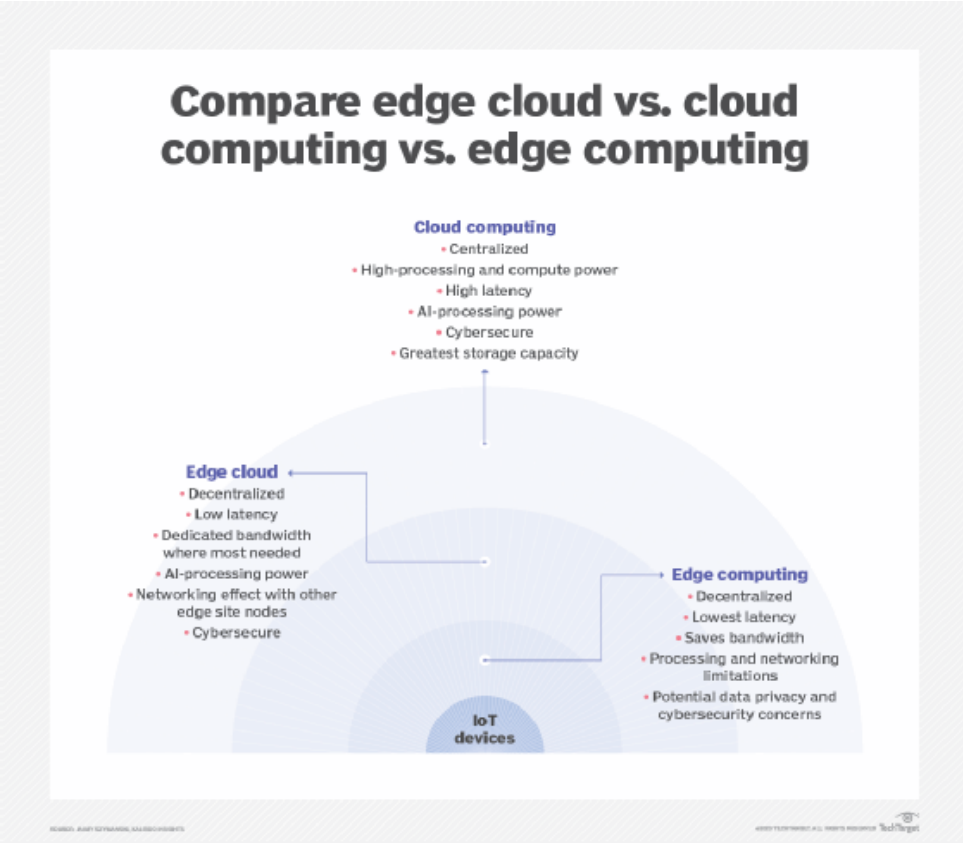
- **Ασφάλεια:** Οι προμηθευτές υπηρεσιών cloud πληρώνουν αδρά σε μέτρα ασφάλειας, όπως κρυπτογραφήσεις, τείχη ασφαλείας και πολύ-επίπεδες μεθόδους αυθεντικοποίησης χρηστών, για να προστατεύουν τα ευαίσθητα δεδομένα και εφαρμογές από απειλές. Οι ειδικεύσεις και οι πόροι που χρησιμοποιούν συχνά ξεπερνούν και τα πρότυπα των εξειδικευμένων οργανισμών.

- **Περιβαλλοντικό αντίκτυπο:** Η βελτιστοποιημένη χρήση και η δυνατότητα αποδοτικής «κοινοχρησίας» υπολογιστικών πόρων, μειώνει τις ανάγκες σε χρήση ενέργειας και αφήνει μικρότερο αποτύπωμα διοξειδίου άνθρακα σε σχέση με τις παραδοσιακές μεθόδους υλικών κέντρων δεδομένων (data centers).

1.4 Edge computing

Η ιδέα του edge computing αφορά, ουσιαστικά, την ανάπτυξη υπολογιστικών και αποθηκευτικών πόρων κοντά στην τοποθεσία δημιουργίας των δεδομένων. Πιο συγκεκριμένα,

είναι ένα κατανεμημένο υπολογιστικό πρότυπο, που τοποθετεί την επεξεργασία και αποθήκευση δεδομένων πιο κοντά στην πηγή τους, ουσιαστικά στην «άκρη» του δικτύου, σε αντίθεση με το πιο σύνηθες "κεντρικά προσανατολισμένο προς το cloud" πρότυπο. Παρ ότι η ιδέα των κατανεμημένων υπολογιστικών μοντέλων, δεν είναι καινούργια, η εγκατάλειψη του παραδοσιακού προτύπου, με κέντρο το cloud, αποφεύγεται. Ο κύριος λόγος, είναι ο φόβος της ανάγκης, για ενδελεχή μελέτη και παρακολούθηση.



Εικόνα 3 Μια σύγκριση των cloud edge, cloud computing και edge computing σε σχέση με συσκευές IoT

Παρ' ότι το edge computing , σαν έννοια όχι μόνο είναι κοντά στο cloud και στο fog computing, αλλά υπάρχουν και σημεία που οι έννοιες επικαλύπτονται, εν τούτοις είναι διαφορετικές έννοιες και δεν θα πρέπει να συγχέονται. Παρακάτω, θα γίνει εκτενέστερη σύγκριση των edge και cloud, αλλά προς το παρόν θα δούμε μόνο κάποια σημαντικά, για το edge, στοιχεία. Αρχικά, και οι τρεις έννοιες μιλάνε για ένα κατανεμημένο υπολογιστικό σύστημα, και επικεντρώνονται στο μέρος στο οποίο υπάρχουν οι μονάδες επεξεργασίας και αποθήκευσης σε σχέση με τα δεδομένα που παράγονται. Η διαφορά έγκειται στο «που» τοποθετούνται αυτές οι δομές. Με λίγα λόγια, στο edge computing, οι δομές είναι κοντά στα δεδομένα και μακριά από τις κεντρικές δομές, στο fog computing, οι δομές είναι ανάμεσα, ούτε πιο κοντά, ούτε πιο μακριά από την πηγή, και τέλος το cloud computing, που μπαίνει στον ρόλο της κεντρικών δομών[4].

Η ανάπτυξη δομών επεξεργασίας και αποθήκευσης δεδομένων εκεί που δημιουργούνται τα δεδομένα, το edge computing μπορεί να διαχειριστεί πολλές συσκευές σε ένα πολύ μικρότερο και πιο γρήγορο LAN (Local Area Network), στο οποίο όλο το bandwidth χρησιμοποιείται αποκλειστικά από τα τοπικές συσκευές παραγωγής δεδομένων, καθιστώντας τα latency και congestion πρακτικά ανύπαρκτα. Τα ωμά δεδομένα, αποθηκεύονται και προστατεύονται, και αν χρειαστεί προ-επεξεργάζονται, προκειμένου να παρθούν αποφάσεις πραγματικού χρόνου πριν αποσταλούν τα αποτελέσματα προς το κεντρικό cloud.

Η δημοτικότητα του edge computing αυξάνεται διαρκώς, γιατί προσφέρει μια αποδοτική λύση στα προβλήματα που γεννώνται με την μετακίνηση τεράστιων όγκων δεδομένων, που οι σύγχρονοι οργανισμοί παράγουν και καταναλώνουν. Επιπλέον, πέρα από το πρόβλημα του όγκου, υπάρχει και το θέμα της ταχύτητας, με τις εφαρμογές να είναι εξαρτημένες όλο και περισσότερο με τους χρόνους επεξεργασίας και απόκρισης. Οι συσκευές edge (π.χ. συσκευές IoT, αισθητήρες, πύλες), έχουν την δυνατότητα συλλογής και επεξεργασίας δεδομένων τοπικά, αν και συχνά απαιτούν κάτι παραπάνω από ένα απλό εξοπλισμό για να λειτουργήσουν σε ένα LAN. Συχνά, ο υπολογιστικός εξοπλισμός είναι θωρακισμένος για να προστατεύεται από θερμοκρασίες, υγρασία και άλλους εξωτερικούς παράγοντες. Όσον αφορά την υπολογιστική ικανότητα, μπορούν να σηκώσουν μικρο-εφαρμογές και να κάνουν ένα αρχικό φιλτράρισμα, όπως και κάποια κανονικοποίηση στην ροή των δεδομένων . Εν συνεχεία, μόνο αυτά που έχουν σημασία ή έχουν ήδη προ-επεξεργαστεί θα αποσταλούν στο κεντρικό σύστημα cloud, για περαιτέρω ανάλυση ή αποθήκευση.

1.5 Βασικά πλεονεκτήματα του edge

Τα πιο σημαντικά πλεονεκτήματα του edge computing συνοψίζονται παρακάτω[5]:

- **Bandwidth (Εύρος ζώνης):** Η ποσότητα δεδομένων που μπορεί να διακινήσει ένα δίκτυο, εκφραζόμενη σε bits/sec. Όλα τα δίκτυα έχουν περιορισμό στο bandwidth, και ειδικά όταν μιλάμε για ασύρματα δίκτυα, όπου και το πρόβλημα είναι ακόμα πιο έντονο. Συνεπώς, υπάρχει όριο για το πόσα δεδομένα μπορούν να ‘‘επικοινωνηθούν’’ σε ένα δίκτυο.

- **Latency (Καθυστέρηση):** Ο χρόνος που χρειάζεται για να σταλθούν δεδομένα μεταξύ δύο σημείων σε ένα δίκτυο. Ακόμα και αν τα σήματα, ταξιδεύουν με την ταχύτητα του φωτός, μεγάλες φυσικές αποστάσεις σε συνδυασμό με την συμφόρηση, μπορούν να δημιουργήσουν καθυστερήσεις. Αυτές οι καθυστερήσεις, οδηγούν σε πιο αργές αναλύσεις και ακόμα πιο αργές αποφάσεις, μειώνοντας την δυνατότητα απόκρισης ενός συστήματος σε πραγματικό χρόνο.

- **Congestion (Συμφόρηση):** Το internet αποτελεί ένα παγκόσμιο δίκτυο δικτύων. Αν και έχει εξελιχτεί με τρόπο που να μπορεί να προσφέρει πολύ καλές γενικού τύπου ανταλλαγές δεδομένων, όπως ένα αρχείο ή κάποιο streaming, όταν μιλάμε για δισεκατομμύρια συσκευών, η ποσότητα των δεδομένων μπορεί να κατακλύσει το internet και να δημιουργήσει μεγάλη συμφόρηση, δημιουργώντας τεράστιες καθυστερήσεις.

- **Αυτονομία**: Σε τοποθεσίες όπως εξέδρες άντλησης πετρελαίου, πλοία στην θάλασσα, απομονωμένες φάρμες ή άλλες απομακρυσμένες τοποθεσίες, η αξιόπιστη σύνδεση και το μεγάλο bandwidth, δεν είναι δυνατά. Το edge computing, μπορεί να κάνει τους απαραίτητους υπολογισμούς, ακόμα και στις συσκευές του edge, ή και να αποθηκεύσει τα δεδομένα μέχρι να αποκατασταθεί μια σταθερή σύνδεση. Με την προ επεξεργασία των δεδομένων, το τελικό πλήθος που μπορεί να αποσταλεί μειώνεται δραστικά και χρειάζεται λιγότερο χρόνο σύνδεσης και μικρότερο bandwidth.

- **Κυριότητα δεδομένων:** Πέρα από το πρόβλημα του όγκου στην μεταφορά, πολλές φορές τα δεδομένα ταξιδεύοντας σε άλλα έθνη και περιοχές, δημιουργούν και προβλήματα ασφάλειας, ιδιωτικότητας και άλλων ειδών νομικά ζητήματα. Το edge μπορεί να χρησιμοποιηθεί για την διατήρηση των δεδομένων κοντά στην πηγή τους, τηρώντας τους νόμους της κυριαρχίας των δεδομένων, όπως το GDPR της ΕΕ, που καθορίζει το πως τα δεδομένα πρέπει να αποθηκεύονται, επεξεργάζονται και δημοσιεύονται. Έτσι, τα ωμά δεδομένα, μπορούν να επεξεργάζονται πρώτα τοπικά, ασφαλίζοντας τα ευαίσθητα σημεία, προτού γίνουν διαθέσιμα στο cloud ή σε κάποιο άλλο πρωταρχικό κέντρο δεδομένων, το οποίο μπορεί να είναι άλλης δικαιοδοσίας.

- **Ασφάλεια:** Προφανώς, η ένταξη ενός ακόμα σταθμού ανάμεσα στην πηγή των δεδομένων και τον όποιο προορισμό τους, δίνει την δυνατότητα για μία ακόμα εφαρμογή μέτρων ασφαλείας. Το edge μπορεί να προσφέρει κρυπτογράφηση ασφαλίζοντας όλα τα δεδομένα που κινούνται προς τον cloud και μπορεί και ίδιος να θωρακιστεί με καλύτερες άμυνες ενάντια σε hackers ή άλλες επιθέσεις.

1.6 Cloud και Edge

Το μέλλον του cloud computing σε συνεργασία με το edge computing διαγράφεται λαμπρό και αναμένεται να φέρει επανάσταση στον τρόπο με τον οποίο τα δεδομένα επεξεργάζονται και διαχειρίζονται. Παρακάτω αναφέρονται κάποια σημεία-κλειδιά όσον αφορά το αποτύπωμα σύγκλισης αυτών των δύο τεχνολογιών:

1. **Κατανεμημένη νοημοσύνη:** Η ενσωμάτωση του edge και του cloud θα οδηγήσει σε ακόμα πιο κατανεμημένα και έξυπνα συστήματα. Οι συσκευές edge θα γίνουν πιο δυνατές, ικανές να διεκπεραιώνουν προχωρημένη επεξεργασία και λήψη αποφάσεων τοπικά, ενώ το cloud θα αναλαμβάνει πολυσύνθετες αναλύσεις και εργασίες μηχανικής μάθησης.

2. **Επίγνωση πραγματικού χρόνου:** Η συνεργασία των edge και cloud θα επιτρέπει στους οργανισμούς να αποκτούν πληροφορία από τα δεδομένα σε πραγματικό χρόνο. Το edge θα επεξεργάζεται τα δεδομένα από την πηγή τους, δίνοντας άμεση πληροφόρηση, ενώ το cloud θα κάνει εκτενέστερη ανάλυση για απόσπαση χρήσιμων και πολύπλοκων μοτίβων.

3. **Αυτόνομα συστήματα:** Το edge computing είναι απαραίτητο για την ανάπτυξη αυτόνομων συστημάτων, όπως τα αυτοοδηγούμενα αυτοκίνητα. Τέτοια συστήματα, χρειάζονται την άμεση λήψη αποφάσεων και τη δυνατότητα επεξεργασίας δεδομένων σε πραγματικό χρόνο, που προσφέρει το edge, με το cloud να αναλαμβάνει την μάθηση και την βελτιστοποίηση τους.

4. **Μεταμόρφωση του IoT:** Ο συνδυασμός του cloud/edge θα είναι η κινητήριος δύναμη στην άνθιση του IoT, επιτρέποντας αδιάκοπη επικοινωνία, επεξεργασία και διαχείριση συσκευών IoT, με αποτέλεσμα πιο αποδοτικά και καλύτερα συνδεδεμένα οικοσυστήματα.

5. **5G και Edge:** Η ανάπτυξη των δικτύων 5G θα αποτελέσει συνεταίρο του edge computing με τις μεγάλες ταχύτητες internet και την συνδεσιμότητα με μικρό latency. Ο νέος αυτός συνδυασμός θα επιτρέψει νέες εφαρμογές σε πεδία όπως η επαυξημένη πραγματικότητα ή η εικονική πραγματικότητα.

6. **Ασφάλεια δικτύων:** Με την ανάλυση πραγματικού χρόνου της κίνησης στο δίκτυο και τα μειωμένα latency, το edge θα μπορεί να εντοπίζει άμεσα επιθέσεις κι ύποπτες συμπεριφορές και να τις αναχαιτίζει τοπικά. Το cloud, απ' την άλλη, μπορεί να αναλαμβάνει την εκπαίδευση μοντέλων μηχανικής μάθησης που είναι υπολογιστικά απαιτητικές, για εντοπισμό τέτοιων επιθέσεων και έχοντας πρόσβαση σε παγκόσμιας κλίμακας δεδομένα, να τα επανεκπαιδεύει, δημιουργώντας μια νοημοσύνη πραγματικού χρόνου ενάντια σε τέτοιες απειλές.

Κεφάλαιο 2

DDoS σε Cloud και Edge υποδομές

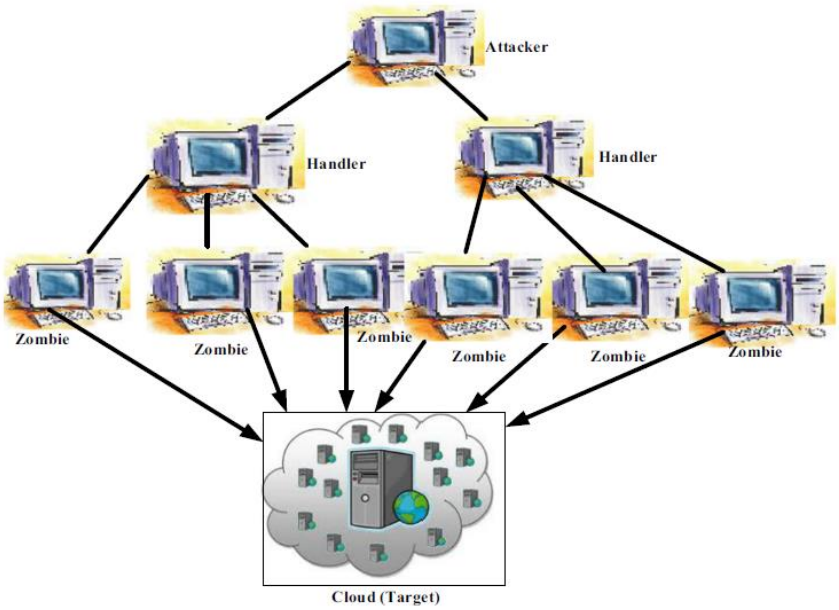
2.1 Επίθεση DDoS

Μια επίθεση DoS (Denial of Service) είναι μια κακόβουλη προσπάθεια από ένα άτομο ή μια ομάδα ατόμων να προκαλέσει στο θύμα, τον ιστότοπο ή σε κάποιο κόμβο να ‘‘αρνηθεί’’ την υπηρεσία στους πελάτες του. Όταν αυτή η προσπάθεια προέρχεται από ένα μόνο υπολογιστή του δικτύου, αποτελεί μια επίθεση DoS. Από την άλλη, είναι δυνατόν πολλοί κακόβουλοι υπολογιστές να συντονίζονται για να κατακλύσουν τον στόχο με πληθώρα πακέτων επίθεσης, ώστε η επίθεση να λαμβάνει χώρα ταυτόχρονα από πολλά σημεία. Αυτός ο τύπος επίθεσης, είναι αυτός που ονομάζεται Κατανεμημένη Επίθεση Άρνησης Υπηρεσίας ή DDoS (Distributed Deny of Service).

Μια επίθεση DDoS μπορεί να πραγματοποιηθεί χρησιμοποιώντας αυτοματοποιημένα εργαλεία επίθεσης. Ορισμένα εργαλεία επίθεσης είναι τα:

-*βασισμένα σε «πράκτορες*», με τον συνδυασμό πρακτόρων και ελεγκτών να γνωρίζουν ο ένας την ταυτότητα του άλλου

-*βασισμένα σε IRC (Internet Relay Chat*), με την επικοινωνία γίνεται έμμεσα, χωρίς να γνωρίζονται μεταξύ τους.



Εικόνα 4 Προσομοίωση επίθεσης DDoS σε cloud

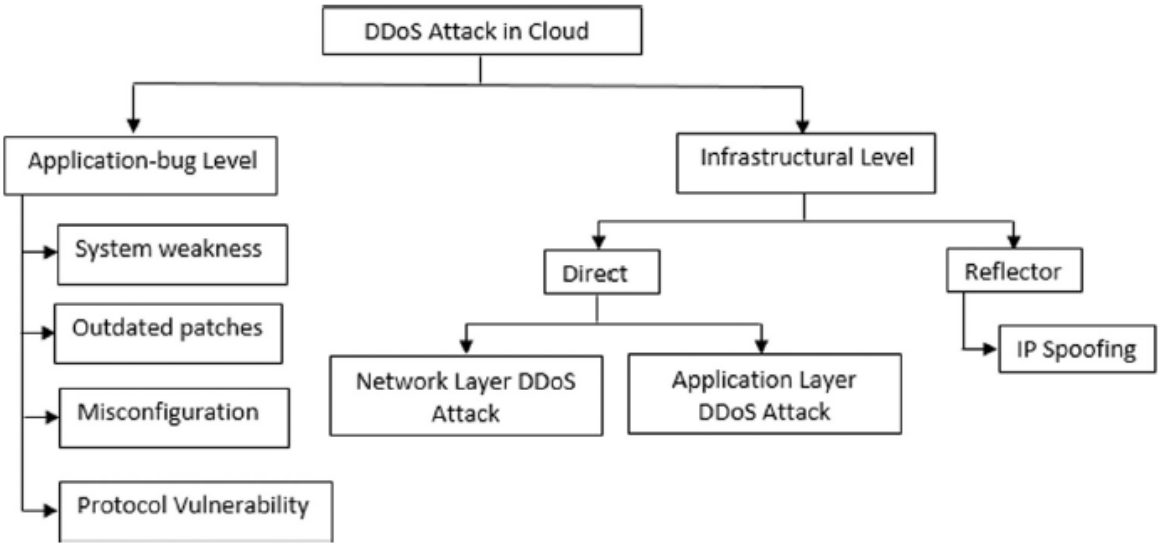
Πρόσφατα, οι επιθέσεις στις εφαρμογές ιστού έγιναν κύριος στόχος, καθώς οι αντίστοιχες λύσεις DDoS ανταποκρίνονταν αποτελεσματικά στις υπάρχουσες επιθέσεις DDoS. Οι επιθέσεις σε επίπεδο εφαρμογής προσομοιώνουν την ίδια σύνταξη αιτήματος και τα χαρακτηριστικά δικτύου όπως αυτά των νόμιμων πελατών, καθιστώντας τις επιθέσεις πολύ δυσκολότερο να ανιχνευθούν και να αντιμετωπιστούν, αφού δύσκολο διακρίνονται από την κανονική κίνηση δικτύου. Επίσης, το σύστημα-στόχος μπορεί να πληγεί ανεξάρτητα από την απόδοση του υλικού, διότι μπορεί να καταστραφεί απλά από πολυπληθείς μικρές συνδέσεις και ενέργειες [6].

2.2 Κατηγοριοποίηση επιθέσεων DDoS

Παρά την βαρύγδουπη σημασία των επιθέσεων DDoS, οι απόπειρες για μια ταξινόμηση αυτού του είδους των επιθέσεων στο cloud, είναι αρκετά περιορισμένες. Οι Deshmukh και Devadkar (2015)[7], χωρίσανε τις επιθέσεις DDoS σε επιθέσεις με σκοπό την εξάντληση του εύρους ζώνης και με σκοπό την εξάντληση των πόρων. Στο Cha και Kim (2011), οι επιθέσεις DDoS που στοχεύουν τις web-υπηρεσίες του cloud, κατηγοριοποιήθηκαν σε επιθέσεις υπερμεγέθους φορτίου, σε αναγκαστικής ανάλυσης και επιθέσεις πλημμύρας, ενώ οι (Wong και Tan, 2014; Bhuyan κ.ά., 2015)[8] κατηγοριοποίησαν τις επιθέσεις DDoS σε επιθέσεις σε επίπεδο υποδομής (OSI Επίπεδα 3 και 4) και επιθέσεις σε επίπεδο εφαρμογής (OSI Επίπεδο 7), όπου OSI (Open Systems Interconnection), είναι ένα μοντέλο που σπάει την επικοινωνία ενός υπολογιστικού συστήματος σε 7 επίπεδα:

1) Φυσικό *2) Συνδέσμου Δεδομένων* *3) Δικτύου* *4) Μετάδοσης*
5) Συνεδρίας *6) Παρουσίασης* *7) Εφαρμογής*.

Στο πλαίσιο αυτής της διπλωματικής, θα ακολουθήσουμε την κατηγοριοποίηση του Osanaiye, Choo και Dlodlo (2016)[9], με την λογική ταξινόμησης σε επίπεδο σφάλματος εφαρμογής και επιθέσεις σε επίπεδο υποδομής.



Εικόνα 5 Ταξινόμηση επιθέσεων DDoS στο cloud.

2.2.1 Επιθέσεις DDoS σε επίπεδο εφαρμογής

Αυτού του είδους οι επιθέσεις εκμεταλλεύονται τρωτά σημεία ή αδυναμίες του συστήματος, με σκοπό να καταστήσουν μη διαθέσιμους τους πόρους του cloud για τους χρήστες. Οι επιτιθέμενοι χρησιμοποιούν διάφορες μεθόδους για να υπερφορτώσουν εφαρμογές και να τις προκαλέσουν να καταρρεύσουν. Ανάμεσα στις πιο συνήθεις κατευθύνσεις, είναι ευπαθή πρωτόκολλα, αδυναμίες στο σύστημα, παλιές ενημερώσεις και λάθος ρυθμίσεις. Ένα παράδειγμα επίθεσης ευπαθούς πρωτοκόλλου, είναι η αποστολή πακέτου, από τους επιτιθεμένους στον στόχο, ειδικά δημιουργημένου για την υπερφόρτωση της εφαρμογής και τελικώς την κατάρρευση της. Οι Beitollahi και τον Deconinck (2012)[10], ή αλλιώς το "ping of death", περιγράφουν την χρήση ενός πακέτου ping με μέγεθος 65535 bytes, το οποίο υπερβαίνει το επιτρεπτό μέγιστο του τύπου IPv4. Σε αυτή την περίπτωση τα πιο πολλά μοντέρνα λειτουργικά συστήματα, μόλις προσπαθήσουν να διαχειριστούν τέτοια πακέτα, παγώνουν, κρασσάρουν και κάνουν επανεκκίνηση, λόγω υπερχείλισης.

2.2.2 Επιθέσεις DDoS σε επίπεδο υποδομής

Οι αλλιώς γνωστές, επιθέσεις πλημμύρας, στοχεύουν σε στοιχεία του cloud, όπως τον αποθηκευτικό χώρο, το εύρος ζώνης δικτύου, τους κύκλους του CPU και τους TCP buffers, για να τα καταστήσουν μη διαθέσιμα για τους χρήστες του cloud. Σε αντίθεση με τις προηγούμενες επιθέσεις, αυτές επικεντρώνονται στην υπερφόρτωση των υποδομών, με στόχο την διακοπή στην παροχή υπηρεσιών. Ένα ακόμα στοιχείο ευκολίας σε αυτήν την επίθεση είναι πως οι επιτιθέμενοι, χρειάζονται μόνο τη διεύθυνση IP του συστήματος-στόχου, χωρίς την ύπαρξη ανάγκης για εύρεση κάποιας αδυναμίας στο σύστημα. Αυτές οι επιθέσεις χωρίζονται σε δύο κύριες μορφές:

- ***Απευθείας επίθεση (Direct Attack):*** Σε μια απευθείας επίθεση, χρησιμοποιούνται θύματα με ελεγχόμενους υπολογιστές-ζόμπι για να στείλουν μια μαζική ποσότητα κακόβουλων πακέτων με στόχο να υπερφορτώσουν τους πόρους του συστήματος-στόχου. Αυτό έχει ως αποτέλεσμα το σύστημα να μην είναι πλέον διαθέσιμο για νόμιμους χρήστες.

Οι επιθέσεις απευθείας μπορούν να χωριστούν περαιτέρω σε επιθέσεις επιπέδου δικτύου DDoS και επιθέσεις επιπέδου εφαρμογής DDoS.

- ***Επίθεση μέσω ανακλαστήρα (Reflector Attack):*** Σε αυτό το είδος επίθεσης, ο επιτιθέμενος, υποκλέπτει μια διεύθυνση IP και στέλνει το αίτημα σε ένα μεγάλο αριθμό κόμβων ανάκλασης. Όταν τα αιτήματα ληφθούν, οι ανακλαστήρες στέλνουν την απάντηση στον στόχο IP, με αποτέλεσμα το πλημμύρισμα του (Bhuyan κ.ά., 2014)[11]. Ένα παράδειγμα αυτής της επίθεσης είναι μια επίθεση smurf, η οποία διεξάγεται στέλνοντας ένα αίτημα ICMP echo ως μήνυμα εκπομπής προς κόμβους στο διαδίκτυο με μια πειραγμένη διεύθυνση IP (την IP του στόχου) (Darwish κ.ά., 2013)[12]. Οι κόμβοι ενισχύουν την επίθεση στέλνοντας απαντήσεις ping προς τον στόχο. Άλλα παραδείγματα τέτοιων επιθέσεων είναι οι SYN ACK RST flood και DNS flood (Bhuyan κ.ά., 2014).

➢ Απευθείας επίθεση DDoS σε επίπεδο δικτύου

Για την πραγματοποίηση μιας επίθεσης σε επίπεδο δικτύου, οι έρευνες έχουν δείξει πως τα πρωτόκολλα που υπάρχουν στο επίπεδο δικτύου και μεταφοράς, μπορούν να αξιοποιηθούν για το πλημμύρισμα του στόχου. Κοινά παραδείγματα τέτοιων επιθέσεων περιλαμβάνουν:

- ***Επίθεση TCP SYN flooding:*** Το TCP είναι ένα πρωτόκολλο επικοινωνίας στο επίπεδο μεταφοράς του μοντέλου στοίβας TCP/IP. Το κύριο χαρακτηριστικό του είναι μια τριμερής χειραψία πριν την αποστολή πακέτων μεταξύ των host, προκειμένου να εδραιωθεί η επικοινωνία. Κατά την διάρκεια της χειραψίας, ο host επικοινωνίας στέλνει ένα μήνυμα SYN, στο άλλο host, ο οποίο απαντά με ένα μήνυμα SYN-ACK. Η χειραψία ολοκληρώνεται με ένα ακόμα ACK, από τον host επικοινωνίας. Οι επιτιθέμενοι εκμεταλλεύονται αυτό το χαρακτηριστικό επικοινωνίας, στέλνοντας ένα μεγάλο αριθμό μηνυμάτων SYN χωρίς όμως να επιτρέπουν την ολοκλήρωση της χειραψίας, προσκαλώντας ημι-ανοιχτές συνδέσεις, οι οποίες εξαντλούν τη μνήμη πυρήνα δημιουργώντας πολλαπλές αναθέσεις μπλοκ μετάδοσης (Wong και Tan, 2014). Αυτό μπορεί να επιτευχθεί καταλαμβάνοντας πολλαπλούς κόμβους στο διαδίκτυο για την διεξαγωγή συντονισμένης επίθεση. Αυτές οι επιθέσεις DDoS μπορούν να γίνουν και χρησιμοποιώντας πειραγμένες διευθύνσεις IP, κατά τη διάρκεια των οποίων, το τελικό ACK που απαιτείται για να ολοκληρωθεί η χειραψία δεν θα αποσταλεί, καθώς ο host με την πειραγμένη IP θα απαντήσει με σημαία RST ή ενδέχεται να μην υπάρχει καν.

- ***Επίθεση UDP flooding:*** Το UDP είναι επίσης ένα πρωτόκολλο μεταφοράς, που χρησιμοποιείται συχνά όταν η αξιοπιστία της μετάδοσης πακέτων δεν είναι υποχρεωτική. Ένα παράδειγμα είναι κατά τη διάρκεια της μεταφοράς εφαρμογών πραγματικού χρόνου, όπως φωνής και βίντεο. Το πρωτόκολλο UDP, μπορεί να εκμεταλλευτεί για να ξεκινήσει επιθέσεις DDoS, δημιουργώντας υπεράριθμα πακέτα UDP προς τυχαίες θύρες του στόχου cloud (Wong και Tan, 2014). Η επίθεση εκμεταλλεύεται τα χαρακτηριστικά έλλειψης σύνδεσης host-peer και αξιοπιστίας του UD, στέλνοντας ογκώδη κακόβουλη κυκλοφορίας προς τον στόχο, γεμίζοντας την ουρά αναμονής, εμποδίζοντας έτσι τις απαντήσεις προς τους χρήστες (Rui κ.ά., 2009)[13]. Το χαρακτηριστικό έλλειψης αξιοπιστίας στο UDP δεν επιτρέπει στο σύστημα στόχο να ρυθμίσει το ρυθμό αποστολής πακέτων των επιτιθεμένων (Wong και Tan, 2014).

- ***Επίθεση ICMP flooding:*** Το ICMP είναι ένα πρωτόκολλο IP που μπορεί να χρησιμοποιηθεί για τον έλεγχο της τρέχουσας κατάστασης της δικτύωσης ενός host. Οι επιτιθέμενοι έχουν χρησιμοποιήσει το ICMP για επιθέσεις DDoS στη μορφή smurf και ping flood (Wong και Tan, 2014), οι οποίες διεξάγονται κατευθύνοντας τεράστια πακέτα ICMP προς έναν στόχο στην προσπάθεια κατανάλωσης του εύρους ζώνης. Και πάλι το αποτέλεσμα είναι ο στόχος να μην μπορεί να ανταποκριθεί σε εισερχόμενα αιτήματα από χρήστες.

➢ Απευθείας επίθεση DDoS σε επίπεδο εφαρμογής

Οι επιθέσεις DDoS σε επίπεδο εφαρμογής στο cloud, αυξάνονται συνέχεια κατά την πάροδο του χρόνου, τόσο σε πλήθος όσο και σε πολυπλοκότητα. Αυτές οι επιθέσεις μειώνουν την παραγωγικότητα, την ποιότητα παροχής υπηρεσιών, την ποιότητα της εμπειρίας, το κύρος και, τελικώς, έσοδα του παρόχου cloud. Αυτές οι επιθέσεις, στοχεύουν στις υπηρεσίες cloud χρησιμοποιώντας το πλημμύρισμα με πακέτα και συνήθως HTTP floods, με ψηλούς ρυθμούς για να κατακλύσουν τους διακομιστές που φιλοξενούνται στο cloud, με σκοπό την παρεμπόδιση παροχής υπηρεσιών στους χρήστες. Τέτοιου είδους επιθέσεις, είναι δύσκολες στην αντιμετώπιση τους, αφού καταναλώνουν μικρό εύρος ζώνης και είναι πιο κρυφές στη φύση τους. Κοινά παραδείγματα τέτοιων επιθέσεων περιλαμβάνουν:

- ***Επίθεση HTTP flood:*** Οι (γνωστές και ως H-DoS, σχεδιάστηκαν για να πλημμυρίζουν τους διακομιστές ιστού και τις εφαρμογές του cloud, χρησιμοποιώντας πειραγμένα πακέτα HTTP (Choi κ.ά., 2014)[14], χωρίς απαραίτητα υψηλό ρυθμό ροής κυκλοφορίας. Παράδειγμα, μια επίθεση HTTP GET μπορεί να διεκπεραιωθεί καταλαμβάνοντας, αρκετούς κόμβους στο δίκτυο για να δημιουργήσει πολλαπλές συνεδρίες αιτημάτων στο θύμα προκειμένου να το κατακλύσει. Μία πρόσφατη αναφορά για παγκόσμιες επιθέσεις DDoS, αποκαλύπτει ότι περίπου ένα τέταρτο των τρεχουσών επιθέσεων DDoS στοχεύουν στο επίπεδο εφαρμογής (Wong και Tan, 2014), και το ένα πέμπτο των επιθέσεων DDoS HTTP είναι κατακλυσμοί HTTP GET.

- ***Επίθεση XML flood:*** Κατά την αίτηση πόρων, οι χρήστες και οι παρόχοι cloud χρησιμοποιούν μηνύματα SOAP. Τα μηνύματα SOAP χρησιμοποιούν HTTP πρωτόκολλο και είναι γραμμένα σε XML, διότι είναι μια καθολικά αποδεκτή γλώσσα που λειτουργεί σε οποιαδήποτε πλατφόρμα (Karnwal κ.ά., 2012)[15]. Οι X-DoS, όπως λέγονται, πραγματοποιούνται με λιγότερο εξελιγμένα εργαλεία, λόγω της ευκολίας υλοποίησής του. Η κατανεμημένη έκδοση του X-DoS είναι γνωστή ως DX-DoS. Στην επίθεση "wrapping" XML στις υπηρεσίες Amazon EC2, από τους Gruschka και Iacono (2009)[16], οι επαληθεύσεις αιτημάτων SOAP εκμεταλλεύονται αλλάζοντας τις ετικέτες XML, επιτρέποντας την πρόσβαση μη εξουσιοδοτημένων χρηστών στις υπηρεσίες Amazon EC2, τις οποίες μπορεί να καταχραστεί ο επιτιθέμενος.

2.3 Εγκατάσταση άμυνας κατά επιθέσεων DDoS

Τις τελευταίες δύο δεκαετίες, έχουν γίνει αρκετές προτάσεις στην άμυνα από επιθέσεις DDoS, και οι πρώτες σχεδιάστηκαν για την αντιμετώπιση επιθέσεων εναντίον ενός μόνο

υπολογιστή. Ένας αριθμός από άμυνες για DDoS στο cloud, που προτάθηκαν πρόσφατα βασίστηκαν σε Λογισμικά Ορισμένο Δίκτυο (SDN). Οι Wang κ.ά. (2014) εξέτασαν την επίδραση στην ασφάλεια από DDoS επιθέσεις σε ένα εταιρικό δίκτυο, που περιλάμβανε SDN και cloud. Οι άμυνες από DDoS για cloud υπηρεσίες μπορούν να εγκατασταθούν σε τέσσερις βασικές τοποθεσίες: στο άκρο της πηγής, σε σημεία πρόσβασης, στο ενδιάμεσο δίκτυο και στην κατανεμημένη άμυνα.

1. **Εγκατάσταση στο άκρο της πηγής**

Το πλεονεκτήματα αυτής της υλοποίησης περιλαμβάνει την αποτελεσματικότερη προστασία των πόρων του δικτύου και της εύρους ζώνης. Για παράδειγμα, οι άμυνες που εγκαθίστανται στην πηγή μιας πιθανής επίθεσης χρησιμοποιούν ένα στοιχείο περιορισμού στον ρυθμό των εξερχόμενων πακέτων κατά τη διάρκεια επιθέσεων DDoS (Bhuyan κ.ά., 2013)[17], προστατεύοντας τους πόρους τόσο του ενδιάμεσου δικτύου όσο και του στόχου.

2. **Εγκατάσταση σε σημεία πρόσβασης**

Η εγκατάσταση στο σημείο πρόσβασης συνήθως γίνεται στο front-end, στο back-end ή σε κάθε εικονική μηχανή (VM) στο περιβάλλον του cloud. Το front-end αποτελεί το «πρόσωπο» της υπηρεσίας cloud και λειτουργεί σαν διεπαφή, μεταξύ των χρήστη-cloud και των στοιχείων-cloud. Οι άμυνες για DDoS, που εγκαθίστανται σε σημεία πρόσβασης διαχωρίζουν τα κανονικά πακέτα από τα κακόβουλα πακέτα πριν δώσουν πρόσβαση στους πόρους και τις υπηρεσίες cloud. Ένας βασικός περιορισμός, είναι ότι τα σημεία πρόσβασης δεν αποτελούν κατάλληλη τοποθεσία για φιλτράρισμα ή περιορισμό του ρυθμού κυκλοφορίας, καθώς το εύρος ζώνης μπορεί να είναι κορεσμένο. Εν τούτοις, αυτή η προσέγγιση είναι η πιο κοινή λόγω της ευκολίας εγκατάστασης.

3. **Εγκατάσταση στο ενδιάμεσο δίκτυο**

Αυτές οι άμυνες εγκαθίστανται σε κόμβους του δικτύου για να περιορίσουν την επιτυχία των επιθέσεων DDoS στο δίκτυο πριν προλάβουν να επηρεάσουν τον στόχο. Αυτό γίνεται με την επιβολή ορίων στον ρυθμό κυκλοφορίας που στους κόμβους, μέσω σύγκρισης με κυκλοφορία κανονικού ρυθμού (Bhuyan κ.ά., 2013). Παρά την αποτελεσματικότητα της η πρακτικότητα είναι εμπόδιο ειδικά σε cloud περιβάλλοντα καθώς οι κόμβοι δεν ελέγχονται από τον ίδιο πάροχο. Έτσι, αυτή η εγκατάσταση μειώνει το πεδίο εφαρμογής της, ίσως σε μία εγκατάσταση ιδιωτικού cloud.

4. **Κατανεμημένη άμυνα**

Η κατανεμημένη άμυνα αποτελεί ένα υβριδικό μοντέλο που περιλαμβάνει τον συνδυασμό όλων των παραπάνω, την εγκατάσταση στο άκρο της πηγής, στα σημεία πρόσβασης και/ή στο ενδιάμεσο δίκτυο, επιτυγχάνοντας πολύ υψηλά ποσοστά ανίχνευσης επιθέσεων DDoS. Το MTF (Iyengar κ.ά., 2014)[18] είναι ένα παράδειγμα διανεμημένης ανάπτυξης άμυνας.

2.4 Ανίχνευση επιθέσεων DDoS

2.4.1 Τυπικές τεχνικές ανίχνευσης DDoS επιθέσεων

Οι τυπικές τεχνικές ανίχνευσης DDoS διακρίνουν την κυκλοφορία πακέτων ως κανονική ή κακόβουλη και μπορούν να κατηγοριοποιηθούν ευρέως σε βασισμένες σε υπογραφές, βασισμένες σε ανωμαλίες και υβριδικές.

Ανίχνευση βασισμένη σε υπογραφές (Signature based detection)

Οι τεχνικές ανίχνευσης βασισμένες σε υπογραφές χρησιμοποιούν ένα σύνολο κανόνων και γνωστά μοτίβα επιθέσεων(υπογραφές), που είναι αποθηκευμένα σε μία βάση δεδομένων. Τα μοτίβα κυκλοφορίας παρακολουθούνται και συγκρίνονται με τις υπάρχουσες υπογραφές με σκοπό την ανίχνευση κακόβουλης κυκλοφορίας. Αυτού του είδους οι τεχνικές ανίχνευσης είναι γνωστές για την ακρίβειά τους στην ανίχνευση γνωστών επιθέσεων, με προϋπόθεση η βάση να παραμένει ενημερωμένη. Αναλόγως, το μεγάλο μειονέκτημα της είναι η αδυναμία της στο να ανιχνεύει καινούργιες επιθέσεις ή παραλλαγές υπογραφών από γνωστές επιθέσεις, οδηγώντας σε υψηλά ποσοστά ψευδώς θετικών.

Πλεονεκτήματα:

- Ακρίβεια στην ανίχνευση υπογραφών ήδη γνωστών επιθέσεων με χαμηλό ποσοστό ψευδών θετικών.
- Η παρουσία ετικετών επίθεσης DDoS επιτρέπει στον διαχειριστή του συστήματος να καθορίσει τον ακριβή τύπο της επίθεσης.

Μειονεκτήματα:

- Η διατήρηση ενημερωμένων υπογραφών είναι μια πολυδάπανη, εάν όχι αδύνατη, διαδικασία.
- Η παραποίηση των υπογραφών θα έχει ως αποτέλεσμα υψηλό ποσοστό ψευδών αρνητικών.
- Αδυναμία ανίχνευσης άγνωστων και επιθέσεων zero-day(επιθέσεων που στοχεύουν αδυναμίες στο σύστημα που δεν έχουν παρατηρηθεί προγενέστερα).

Ανίχνευση βασισμένη σε ανωμαλίες (Anomaly based detection)

Η ανίχνευση βασισμένη σε ανωμαλίες ή κατηγοριοποίησης συμπεριφοράς περιλαμβάνει τη συλλογή ενός προφίλ συμπεριφοράς της κυκλοφορίας κανονικής κυκλοφορίας κατά τη διάρκεια ενός χρονικού διαστήματος. Ο στόχος είναι η ανίχνευση υπολειπόμενων μοτίβων που αποκλίνουν από μια αναμενόμενη συμπεριφορά. Οι Chandola κ.ά. (2009)[19] ομαδοποιούν τις ανωμαλίες σε τρεις κύριες κατηγορίες.

- Η ***ανωμαλία σημείου*** συμβαίνει όταν ένα συγκεκριμένο στιγμιότυπο δεδομένων θεωρείται ανωμαλία σε σχέση με τα υπόλοιπα δεδομένα.
- Μια ***συμφραζόμενη ανωμαλία***, χαρακτηρίζεται όταν τα δεδομένα είναι ανώμαλα σε ένα συγκεκριμένο πλαίσιο, αλλά όχι σε κάποιο άλλο πλαίσιο. Αυτό καθορίζεται κυρίως από τη δομή του συνόλου δεδομένων.
- Σε μια ***συλλογική ανωμαλία***, μια ομάδα στιγμιότυπων δεδομένων είναι ανώμαλη σε σχέση με το σύνολο των δεδομένων. Ένα παράδειγμα είναι οι επιθέσεις πλημμύρας DDoS, όπου μόνο κάποια μεμονωμένα στιγμιότυπα δεδομένων γίνονται ανώμαλα και βλάπτουν τον συνολικό συντονισμό.

Η προσέγγιση της ανίχνευσης ανωμαλιών υλοποιείται συνήθως σε δύο φάσεις:

- **Φάση εκπαίδευσης:** Η αποτελεσματικότητα της ανίχνευσης ανωμαλιών εξαρτάται από τη φύση των εισερχόμενων δεδομένων κατά τη φάση εκπαίδευσης. Η είσοδος αποτελείται από μία συλλογή στιγμιότυπων δεδομένων σε μορφή μοτίβων, δειγμάτων και παρατηρήσεων που περιγράφονται από ένα σύνολο χαρακτηριστικών, σε δυαδική μορφή, σε μορφή κατηγοριών ή σε μορφή αριθμών. Κάθε στιγμιότυπο δεδομένων μπορεί να αποτελείται από ένα χαρακτηριστικό (univariate) ή πολλά χαρακτηριστικά (multivariate) (Chandola κ.ά., 2009). Οι ετικέτες δεδομένων χρησιμοποιούνται για να καθορίσουν εάν ένα συγκεκριμένο στιγμιότυπο είναι κανονικό ή ανωμαλία. Υπάρχουν σύνολα δεδομένων για έρευνα, που περιλαμβάνουν διάφορες ετικέτες ανωμαλίας επιθέσεων και κανονικές περιπτώσεις δεδομένων. Ένα τέτοιο δημοφιλές παράδειγμα είναι το KDD'99, που περιλαμβάνει περίπου 4.900.000 διανύσματα μοναδικής σύνδεσης, με κάθε ένα από τα διανύσματα περιλαμβάνει 41 χαρακτηριστικά και χαρακτηρίζεται ως επίθεση ή κανονική. Οι επιθέσεις ανήκουν σε τέσσερις κατηγορίες: άρνησης υπηρεσίας (DoS), χρήστης-προς-ρίζα (U2R), απομακρυσμένος-προς-τοπικός (R2L) και ανίχνευσης ή Probing (Tavallaee κ.ά., 2009)[20].

- **Φάση ανίχνευσης:** Η ανίχνευση ανωμαλιών έχει τρεις λειτουργίες βάσει της ποσότητας των ετικετών που είναι διαθέσιμες, δηλαδή με επίβλεψη(supervised), με ημι-επίβλεψη(semi-supervised) και χωρίς επίβλεψη(unsupervised).

  a) Στο ***μοντέλο με επίβλεψη*** θεωρείται δεδομένη η διαθεσιμότητα στιγμιότυπων με ετικέτες στα σύνολα δεδομένων για εκπαίδευση με κανονικές και ανώμαλες κλάσεις. Η προσέγγιση αυτή χρησιμοποιείται για την κατασκευή μοντέλου πρόβλεψης κανονικών έναντι ανώμαλων κλάσεων. Στιγμιότυπα δεδομένων άγνωστα μέχρι τώρα, συγκρίνονται, για να καθοριστεί η κλάση στην οποία ανήκουν. Δύο είναι τα βασικά ζητήματα με την ανίχνευση ανωμαλιών με επίβλεψη. Καταρχάς, οι περιπτώσεις ανωμαλιών είναι πολύ λιγότερες σε σύγκριση με τις κανονικές περιπτώσεις στα δεδομένα εκπαίδευσης. Δεύτερον, το πρόβλημα διάκρισης των κλάσεων ανωμαλιών είναι με πλήρως αντιπροσωπευτικές ετικέτες (Bhuyan κ.ά., 2014).

  b) Το ***μοντέλο με ημι-επίβλεψη*** υποθέτει ότι τα δεδομένα εκπαίδευσης έχουν ετικέτες μόνο για την κανονική κλάση. Είναι πολύ πιο πρακτικές σε σύγκριση με τις τεχνικές, με επίβλεψη, αφού ετικέτες για την κλάση ανωμαλιών δεν είναι απαραίτητες (Chandola κ.ά., 2009).

  c) Στην ***μοντέλο χωρίς επίβλεψη***, δεν απαιτούνται καθόλου δεδομένα εκπαίδευσης. Επομένως, είναι μία από τις πιο ευρέως χρησιμοποιούμενες τεχνικές (Bhuyan κ.ά., 2014). Το μοντέλο αυτό, κάνει την υπόθεση πως οι κανονικές περιπτώσεις είναι σημαντικά περισσότερες από τις ανώμαλες περιπτώσεις σε ένα τυπικό σύνολο δεδομένων. Εάν αυτή η υπόθεση δεν είναι αληθής, υπάρχει πρόβλημα με υψηλό ποσοστό λάθος συναγερμών.

Πολύ σημαντικό στην ανίχνευση ανωμαλιών, είναι η αναφορά των ανωμαλιών που ανιχνεύονται. Τα δύο κυριότερα στοιχεία αυτών, είναι οι βαθμολογίες και οι ετικέτες. Η χρήση βαθμολογιών αφορά την ανάθεση βαθμολογίας της ανωμαλίας σε κάθε στιγμιότυπο δεδομένων για να δείξει το κατά πόσο είναι ανώμαλο. Έτσι, θεσπίζεται ένα κατώφλι (threshold), που καθορίζει την αποδοχή ή απόρριψη του στιγμιότυπου δεδομένων. Οι

ετικέτες, απ' την άλλη, περιλαμβάνουν την ανάθεση μιας ετικέτας σε κάθε στιγμιότυπο, σαν κανονικό ή με ανωμαλίες.

2.4.2 Τεχνικές ανίχνευσης ανωμαλιών DDoS επιθέσεων

Σε αυτή την κατηγοριοποίηση, βασιζόμαστε στους αλγόριθμους στους οποίους χρησιμοποιούνται από κάθε τεχνική.

1. **Στατιστική ανίχνευση ανωμαλιών (Statistical anomaly detection)**

Στη στατιστική ανίχνευση ανωμαλιών, συλλέγονται τα στατιστικά χαρακτηριστικά φυσιολογικής κυκλοφορίας με σκοπό την δημιουργία μοτίβου φυσιολογικής κυκλοφορίας, το οποίο θα συγκρίνεται με την εισερχόμενη κυκλοφορία για την ανίχνευση ανώμαλων πακέτων. Δύο γνωστά παραδείγματα στατιστικής ανίχνευσης ανωμαλιών είναι, των Vissers κ.ά. (2014)[21], με την χρήση ενός μοντέλου Gauss πολλών φάσεων, για άμυνα εναντίον επιθέσεων DDoS στο επίπεδο εφαρμογής. Το πρώτο στάδιο, περιλαμβάνει έλεγχο της επικεφαλίδας HTTP για να αποτροπή πλημμύρας HTTP, έλεγχο SOAP και έλεγχο μεγέθους. Στην επόμενη φάση, επεξεργάζεται το περιεχόμενο XML πριν από τον έλεγχο αν το SOAP, έχει παραποιηθεί. Τελικώς, γίνεται ένας διαδικαστικός έλεγχος που αξιολογεί κάθε χαρακτηριστικό σε σχέση με το αντίστοιχο μοντέλο Gauss. Η αδυναμία αυτού του μοντέλου βρίσκεται ανίχνευση αιτημάτων που προκύπτουν από νέες τεχνικές DDoS επιθέσεων, χωρίς την υλοποίηση επιπλέον χαρακτηριστικών. Ακόμα, είναι το μοντέλο των Shamsolmoali και Zareapoor (2014)[22], το οποίο χρησιμοποιεί ένα σύστημα φιλτραρίσματος βασισμένο στη στατιστική, με δύο επίπεδα φιλτραρίσματος. Στο πρώτο, αφαιρείται το πεδίο επικεφαλίδας του εισερχόμενου πακέτου και συγκρίνεται η τιμή TTL(Time To Live) με την αποθηκευμένη τιμή στον πίνακα IP-to-hop count (IP2HC). Εάν αυτές οι τιμές δεν είναι ίσες, το πακέτο απορρίπτεται και κατηγοριοποιείται ως παραποιημένο. Το δεύτερο επίπεδο βασίζεται στο μαθηματικό μοντέλο απόκλισης Jensen-Shannon, χρησιμοποιώντας ένα αποθηκευμένο φυσιολογικό προφίλ για σύγκριση με τις των επικεφαλίδες των εισερχομένων πακέτων.

Πλεονεκτήματα:

- Οι προσεγγίσεις στατιστικής ανίχνευσης ανωμαλιών επιτρέπουν την αναγνώριση αναμενόμενης συμπεριφοράς χωρίς πρότερη γνώση των φυσιολογικών δραστηριοτήτων του συστήματος στόχου.
- Οι βαθμολογίες ανωμαλιών που συνδέονται με τη στατιστική ανίχνευση μπορούν να χρησιμοποιηθούν ως διάστημα εμπιστοσύνης κατά τη λήψη αποφάσεων.

Μειονεκτήματα:

- Ο ορισμός ενός ιδανικού κατωφλίου, χωρίς υπερβολικά ψευδή θετικά ή ψευδή αρνητικά αποτελεί μία ιδιαίτερη πρόκληση.
- Οι τεχνικές ανίχνευσης στατιστικών ανωμαλιών εμπεριέχουν στην βάση τους, υποθέσεις οι οποίες ένα δεν ικανοποιηθούν, οδηγούν σε υψηλό ποσοστό λάθος κατηγοριοποίησης.

2. **Εξόρυξη Δεδομένων (Data Mining)**

Η αύξηση διαδικτυακής κυκλοφορίας δυσχεραίνει ιδιαίτερα τις προσπάθειες εντοπισμού DDoS επιθέσεων μέσω της ανίχνευσης ανωμαλιών. Ένα παράδειγμα στην αντιμετώπιση αυτού του προβλήματος: οι Choi et al. (2014)μια προσέγγιση πάνω στην εξόρυξη δεδομένων με την χρήση του μοντέλου map reduce, για να μειώσει τις επιθέσεις DDoS τύπου HTTP

GET, σε επίπεδο εφαρμογής. Το map reduce είναι ένα μοντέλο παράλληλης επεξεργασίας που έχει χρησιμοποιηθεί για να απλοποιεί την διαχείριση μεγάλων συνόλων δεδομένων. Ένας αλγόριθμος map reduce, χρησιμοποιήθηκε για την αξιολόγηση του μοντέλου αυτού, μετρώντας τα ποσοστά μεταξύ του μοτίβου «κανόνα» και του χρόνου ανίχνευσης του προτεινόμενου συστήματος σε σύγκριση με νέες υπογραφές, με τα αποτελέσματα να δείχνουν ικανότητα ανίχνευσης νέων προφίλ επίθεσης, με μικρότερο χρόνο επεξεργασίας.

Πλεονεκτήματα:

- Σημαντική η ικανότητα επεξεργασίας μεγάλων βάσεων δεδομένων, με εξαγωγή συνόλων πληροφοριών και μετατροπής τους σε απλές δομές.
- Προσθέτει ένα ακόμα επίπεδο εστίασης που βοηθά στην βελτίωση ανίχνευσης ανωμαλιών για DDoS.
- Ενισχύει τη δυνατότητα του διαχειριστή δικτύου να διακρίνει μεταξύ επιθέσεων και φυσιολογικής κίνησης, βάζοντας όρια που θεσπίζουν την κανονική δραστηριότητα δικτύου.

Μειονεκτήματα:

- Η απουσία ή οι κακές τιμές στο σύνολο δεδομένων μειώνει την αποτελεσματικότητα.
- Δύσκολο έργο η επιλογή σωστών χαρακτηριστικών μπορεί να είναι πρόβλημα για μεγάλα σύνολα δεδομένων, καθώς η επιλογή όλων επιδεινώνει την απόδοση.

3. Τεχνητή Νοημοσύνη (Artificial Intelligence)

Η προσέγγιση με χρήση τεχνητής νοημοσύνης, έχει σαν βασική απαίτηση μια συνεχή διαδικασία μάθησης, προκειμένου να είναι αποτελεσματική στην ανίχνευση νέων ανωμαλιών. Οι Joshi και Joshi (2012)[23], για παράδειγμα, προτείνουν ένα Cloud TraceBack (CTB) μοντέλο και έναν cloud προστάτη για την αντιμετώπιση επιθέσεων DDoS, χρησιμοποιώντας ένα αναδρομικό νευρωνικό δίκτυο, υλοποιημένο κοντά στο νέφος και χρησιμοποιεί έναν αλγόριθμο για να σημειώσει τα πεδία σημαίας και ID της κεφαλίδας IP των πακέτων. Η εξάλειψη της επίθεσης, αναλαμβάνεται από τον cloud προστάτη.
Οι Huang κ.ά. (2013)[24] προτείνουν ένα σύστημα ανίχνευσης πολλών σταδίων και έλεγχο κειμένου για την αντιμετώπιση επιθέσεων HTTP flooding. Το σύστημα αποτελείται από πέντε στάδια, δηλαδή έλεγχο πηγής, καταμέτρηση, ανίχνευση επίθεσης, δοκιμή Turing και δημιουργίας ερωτήσεων και έχει υλοποιηθεί στον πυρήνα του Linux και στους χώρους χρήστη, πάντα ανάλογα με τις απαιτήσεις, με τις δοκιμές απόδοσης να δείχνουν ψηλή αποτελεσματικότητα.

Πλεονεκτήματα:

- Η χρήση νευρωνικών δικτύων για μη επιβλεπόμενη μάθηση μπορεί να είναι σχετικά αποτελεσματική στην ανίχνευση πακέτων επιθέσεων DDoS.
- Η φύση των τεχνικών τεχνητής νοημοσύνης επιτρέπει την εκπαίδευση με στιγμιότυπα με συσσωρευτικό τρόπο.

Μειονεκτήματα:

- Δύσκολα κλιμακούμενες.
- Κατά τη φάση εκπαίδευσης, μπορεί να συμβεί overfitting(υπερβολική εκπαίδευση)
- Έλλειψη σε επαρκώς μεγάλο πλήθος δεδομένων κανονικής κίνησης, χωρίς τα οποία μειώνεται η αποτελεσματικότητα και η αποδοτικότητά.

4. Ταξινομητές (Classifiers)

Σε αυτή την τεχνική, ταξινομητές μαθαίνουν από ένα σύνολο στιγμιότυπων δεδομένων με ετικέτες, προκειμένου να ταξινομήσουν μια περίπτωση σε μία από τις κατηγορίες. Αυτές οι τεχνικές, αποτελούνται από την φάση εκπαίδευσης και στη φάση δοκιμής. Στη φάση εκπαίδευσης, ο ταξινομητής εκπαιδεύεται από τις διαθέσιμες ετικέτες στα δεδομένα και κατά τη φάση δοκιμής, ταξινομεί μια δοκιμαστική περίπτωση είτε ως κανονική είτε ως ανωμαλία(Chandola κ.ά., 2009). Οι Chonka και Abawajy (2012)[25] και Chonka κ.ά. (2011)[26] προτείνουν τεχνική ταξινόμησης με δέντρο αποφάσεων για την ανίχνευση και αντιμετώπιση επιθέσεων HX-DoS εναντίον υπηρεσιών νέφους. Το HX-DoS είναι μια επίθεση στο επίπεδο εφαρμογής που συνδυάζει τόσο HTTP όσο και XML μηνύματα για να πλημμυρίσει τους πόρους του cloud.

Οι Lonea κ.ά. (2013)[27] προτείνουν μία μέθοδο βασισμένη σε IDS(Intrusion Detection System). Το IDS υλοποιείται στις εικονικές μηχανές (VMs) του cloud, με τον front-end διακομιστή, να εφαρμόζει μία μεθοδολογία συγχώνευσης δεδομένων. Κατά την ανίχνευση απειλής, οι ειδοποιήσεις που δημιουργούνται στα VM αποθηκεύονται σε μια βάση δεδομένων MySQL και η ανάλυση τους γίνεται από τα IDS του κάθε VM, τα οποία χρησιμοποιούν ταξινομητές ποσοτικής λύσης. Οι αξιολογήσεις υποδεικνύουν ότι η προτεινόμενη λύση μπορεί να μειώσει το ρυθμό των ψευδών αρνητικών και να αυξήσει το ρυθμό ανίχνευσης, χωρίς αύξηση της πολυπλοκότητας.

Ο μηχανισμός πολυεπίπεδου φιλτραρίσματος Multilevel Thrust Filtration(MTF) των Iyengar κ.ά. (2014) περιλαμβάνει τέσσερις επίπεδα ανίχνευσης και πρόληψης, σχεδιασμένα για να προστατεύουν από την πρόσβαση των επιτεθέντων στο περιβάλλον του cloud. Τα επίπεδα είναι ανάλυσης κυκλοφορίας, ανίχνευσης ανωμαλιών, κατηγοριοποίηση ανωμαλιών με ταξινομητές και πρόληψης επιθέσεων. Το MTF λειτουργεί επαληθεύοντας τα εισερχόμενα πακέτα και ανιχνεύοντας τέσσερις τύπους κίνησης (ψεύτικη επίθεση, επίθεση DDoS, flash crowd και επιθετική κανονική κίνηση). Το μοντέλο χρησιμοποιεί τεχνικές ανίχνευσης βασισμένες στα τερματικά και στους δρομολογητές, με σκοπό για την ανίχνευση επιθέσεων στα πρώιμα στάδια, προκειμένου να μην επιτραπεί η είσοδος κακόβουλης κίνησης στο κέντρο δεδομένων του cloud.

Πλεονεκτήματα:

- Οι τεχνικές ανίχνευσης ανωμαλιών με ταξινομητές έχουν υψηλό ποσοστό ανίχνευσης, λόγω μεγάλης ακριβείας στις ρυθμίσεις των κατωφλιών(thresholds).
- Χαρακτηρίζονται από υψηλό ρυθμό προσαρμογής για την ενημέρωση των στρατηγικών ανίχνευσης.

Μειονεκτήματα:

- Απαιτούν επαρκώς μεγάλο αριθμό πληροφοριών εκπαίδευσης για την ανίχνευση άγνωστων επιθέσεων.
- Η κατανάλωση πόρων είναι υψηλή σε σύγκριση με άλλες τεχνικές.

5. Μηχανική Μάθηση (Machine Learning - ML)

Η εφαρμογή μηχανικής μάθησης για την ανίχνευση επιθέσεων DDoS στο cloud, συμπληρώνει τεχνικές όπως στατιστικές και εξόρυξης δεδομένων, έχοντας όμως μία ουσιώδη διαφορά. Οι τελευταίες απαιτούν την κατανόηση της διαδικασίας στην δημιουργία των δεδομένων, ενώ η μηχανική μάθηση έχει να κάνει με τη δημιουργία ενός συστήματος που έχει σκοπό την βελτίωση της απόδοσης βάσει προηγούμενων αποτελεσμάτων. Οι Gupta κ.ά.

(2013)[28] προτείνουν ένα σύστημα ανίχνευσης και πρόληψης εισβολών, το οποίο ασφαλίζει το νέφος από κακόβουλους εντός και εκτός του δικτύου. Συνδυάζει την ανάλυση λεπτομερών δεδομένων και την προσέγγιση με βάση την τεχνική Bayes, ενός γνωστού αλγορίθμου μηχανικής μάθησης, για την ανίχνευση επιθέσεων DDoS, με στόχο την ανίχνευση επιθέσεων στο δίκτυο, όπως το TCP SYN flooding.

Μια άλλη γενικότερη προσέγγιση, από τους Palmieri κ.ά. (2014)[29], επικεντρώνεται στην ανίχνευση μέσω ανάλυσης μεμονωμένων συνιστωσών. Το κατανεμημένο σύστημα αποτελείται από δύο φάσεις μηχανικές μάθησης, ενός Blind Source Separation(BSS), συγγενικού μοντέλου του Principal Component Analysis(PCA) για σήματα πηγής και ενός ταξινομητή βασισμένο σε κανόνες, για επιθέσεις zero-day, που αλλοιώνουν τα χαρακτηριστικά και το ρυθμό της κυκλοφορίας. Η BSS εξάγει χαρακτηριστικά κυκλοφορίας από κατανεμημένους αισθητήρες, για τη δημιουργία ενός προφίλ κανονικής ροής μέσω ενός ταξινομητή δέντρου αποφάσεως .

Πλεονεκτήματα:

- Ψηλή αποτελεσματικότητα στην ανίχνευση μοτίβου επιθέσεων DDoS.
- Δυνατότητα αλλαγής της εκτέλεσής τους, με νεοαποκτηθέντες πληροφορίες.

Μειονεκτήματα:

- Μεγάλη απαίτηση σε υπολογιστικούς πόρους κατά τη φάση εκπαίδευσης και δοκιμής.
- Επιδείνωση απόδοσης του συστήματος, σε περίπτωση κορεσμού, λόγω υπερβολικής κίνησης.

2.4.3 Άλλες τεχνικές ανίχνευσης DDoS επιθέσεων

- **Υβριδική ανίχνευση**

Η υβριδική προσέγγιση συνδυάζει την χρήση των τεχνικών που βασίζονται σε υπογραφές και ανωμαλίες, χρησιμοποιώντας τα συμπληρωματικά χαρακτηριστικά και των δύο για καλύτερα ποσοστά ανίχνευσης. Για παράδειγμα, οι Krishnan και Chatterjee (2012)[30] προτείνουν ένα κατανεμημένο Σύστημα Ανίχνευσης Εισβολών(Intrusion Detection System - IDS) που συνδυάζει τεχνικές βασισμένες σε ανωμαλίες και τεχνικές γνώσεων ενάντια DDoS επιθέσεων στο νέφος. Η λύση έχει έναν πράκτορα υπηρεσίας, έναν πράκτορα ειδοποίησης και έναν πράκτορα αποθήκευσης, οι οποίοι επικοινωνούν μεταξύ τους και με τα ζευγάρια κόμβων, με το σύστημα υλοποιεί επίσης έναν αναλυτή συναγερμού, βοηθώντας τους κόμβους να διαφοροποιούν μεταξύ λάθος συναγερμών και κακόβουλων κόμβων.

Οι Cha και ο Kim (2011) προτείνουν ανίχνευση ανωμαλιών με τρία στάδια. Το πρώτο στάδιο είναι παρακολούθησης, βασισμένο σε κανόνες και προ-επεξεργάζεται γνωστά μοτίβα επιθέσεων DDoS, το δεύτερο στάδιο, κάνει μία «ελαφρά» χρήση ανίχνευσης ανωμαλιών, προβλέποντας το φορτίο κίνησης σε κάθε διεπαφή χρήστη, διαχωρίζοντας την κυκλοφορία στο δίκτυο βάσει όγκου κατά μήκος του άξονα του χρόνου και αναλύοντας με χρήση της Bayesian τεχνικής. Το τελευταίο στάδιο χρησιμοποιεί αλγόριθμο μάθησης χωρίς επίβλεψη ανίχνευση γνωστών και άγνωστων μοτίβων επιθέσεων DDoS.

Οι Modi κ.ά. (2012)[31] σχεδίασαν ένα υβριδικό μοντέλο βασισμένο στο δίκτυο. Το Snort, μια open source μέθοδος ανίχνευσης, που βασίζεται σε υπογραφές και αποθηκεύει γνωστά μοτίβα επιθέσεων DDoS, και ο Bayesian ταξινομητής, ένας στατιστικός ταξινομητής που προβλέπει την πιθανότητα ένα γεγονός στο δίκτυο, να ανήκει σε μια κατηγορία του κανονικού ή του κακόβουλου, χρησιμοποιήθηκαν σε αυτήν τη λύση, προσφέροντας μεγάλη ακρίβεια.

Ενώ μια υβριδική προσέγγιση εκμεταλλεύεται τα πλεονεκτήματα που προσφέρουν τα συστήματα που βασίζονται τόσο σε υπογραφές όσο και σε ανωμαλίες, εντούτοις συνδέεται με επιβραδύνσεις και πολυπλοκότητα στο θέμα διασφάλισης αποτελεσματικής λειτουργίας σε περιπτώσεις διαφορετικών αλγορίθμων.

- **Ανίχνευση Ιχνών και παραποιημένων IP**

Με την ανίχνευση ιχνών, είναι δυνατός ο εντοπισμός της πραγματικής πηγής των επιθέσεων DDoS, καθώς αυτές έχουν την τάση να παραποιούν τις διευθύνσεις τους (π.χ. επίθεση μεσω ανακλαστήρα). Στο πλαίσιο της προτεινόμενης στρατηγικής αμυντικής για επιθέσεις επιπέδου εφαρμογής DDoS εναντίον υπηρεσιών νέφους, οι Yang κ.ά. (2012)[32] προτείνουν μια τεχνική βασισμένη σε υπηρεσίες (SOA) που ονομάζεται SOA-Based Traceback Approach (SBTA) και ένα φίλτρο cloud. Το SBTA, χρησιμοποιεί προηγμένες τεχνικές μαρκαρίσματος πακέτων με σκοπό τη ανακατασκευή των μονοπατιών των πακέτων και το φίλτρο νέφους, δρα σαν μηχανισμός ελέγχου για φιλτράρισμα και περιορισμό του ρυθμού. Το φίλτρο συλλέγει τις ετικέτες και τις διευθύνσεις IP πηγής κατά τη διάρκεια της επίθεσης και χρησιμοποιεί τη βάση δεδομένων για το φιλτράρισμα πακέτων με παραποιημένες διευθύνσεις IP. Ένα σημαντικό μειονέκτημα είναι η ψηλή συχνότητα ψευδών αρνητικών.

Άμυνες σε επιθέσεις DDoS στο cloud με παραποιημένες IP , έχουν επίσης προταθεί. Στο Jeyanthi κ.ά. (2013b)[33], προτείνεται μια τεχνική για την αναγνώριση ψευδών IP στις επιθέσεις DDoS. Οι συγγραφείς προτείνουν έναν αλγόριθμο, ο οποίος ενεργοποιείται όταν υπάρχει ξαφνική αύξηση στην κυκλοφορία πακέτων μεγαλύτερη από ένα προκαθορισμένο όριο. Ακόμη, περιλαμβάνει σύστημα αυθεντικοποίησης cloud (Cloud Authentication System - CAS), που επαληθεύει τη νομιμότητα των συνδέσεων των χρηστών cloud. Το CAS, διατηρεί δύο πίνακες έναν με ψευδείς διευθύνσεις και έναν με τις τρέχουσες συνδέσεις και χρησιμοποιώντας μαζί έλεγχο πλημμύρας, έλεγχο πακέτων και έναν τελικό έλεγχο, προσδιορίζει εάν η κυκλοφορία αποτελεί επίθεση πλημμύρας.

Στο Osanaiye (2015)[34], με κίνητρο το γεγονός ότι οι περισσότερες επιθέσεις DDoS χαρακτηρίζονται από την ψευδείς διευθύνσεις IP, προτείνεται μια τεχνική λειτουργικού συστήματος σαν αποτύπωμα, που παρακολουθεί τα πακέτα που εισέρχονται στο cloud για να προσδιορίσει το λειτουργικό σύστημά από το οποίο προέρχονται. Ο αλγόριθμος διαθέτει δύο στάδια, ένα ενεργητικό και ένα παθητικό στάδιο. Στο παθητικό, συλλέγονται και αναλύονται οι επικεφαλίδες των εισερχομένων πακέτων για τον προσδιορισμό του λειτουργικού συστήματος τους. Στο ενεργητικό, στέλνονται ειδικά δημιουργημένα πακέτα στην πηγή IP του συνδεδεμένου πακέτου για πραγματοποίηση σύγκρισης μεταξύ των δύο σταδίων. Εάν τα δύο λειτουργικά συστήματα δεν είναι τα ίδια, τα πακέτα θεωρούνται παραποιημένα και απορρίπτονται.

2.5 Το μέλλον στην άμυνα ενάντια σε επιθέσεις DDoS

Έχοντας κατηγοριοποιήσει τις επιθέσεις DDoS σε επίπεδο εφαρμογής και επίπεδο υποδομής, οι περισσότερες έρευνες φαίνεται να επικεντρώνονται κυρίως στο δεύτερο. Ο λόγος, όπως έχει ήδη αναφερθεί, η ευκολία με την οποία μπορούν να πραγματοποιηθούν τέτοιου είδους επιθέσεις DDoS σε επίπεδο υποδομής. Δεν χρειάζεται να υπάρχει αδυναμία ή τρωτό σημείο στο cloud, μόνο κίνηση κακόβουλων πακέτων προς τον στόχο για να καταναλώσουν τους πόρους σε βάρος των χρηστών. Από την άλλη πλευρά, έχουν αναφερθεί επίσης επιθέσεις DDoS σε επίπεδο εφαρμογών, στοχεύοντας αδυναμίες του συστήματος πραγματοποίηση επίθεσης, όπως ελλιπή διαμόρφωση, παλαιές ενημερώσεις, ευπάθειες πρωτοκόλλων. Ωστόσο, οι δημοσιεύσεις για τη μείωση αυτού του είδους των επιθέσεων DDoS είναι ελάχιστες.

Η πιο κοινή τοποθεσία τοποθέτησης άμυνας για επιθέσεις DDoS, είναι το σημείο πρόσβασης, με την κατανεμημένη τοποθέτηση άμυνων να έχει προταθεί για αποδοτικότητα. Η ανίχνευση στην πηγή, είναι η πιο ιδανική τοποθεσία, αλλά είναι δύσκολο να επιβληθεί μία γενική πολιτική σε όλους τους υπολογιστές στο διαδίκτυο. Επίσης, φαίνεται πως οι πρώιμες τεχνικές κλίνουν προς την τεχνική ανίχνευσης βασισμένη σε υπογραφές για την αναγνώριση γνωστών επιθέσεων DDoS, το οποίο παρότι είναι αποτελεσματικό για την αντιμετώπιση γνωστών επιθέσεων, γίνεται ολοένα και πιο αχρείαστο στο σημερινό τοπίο απειλών λόγω της ανικανότητάς τους στην ανίχνευση νέων, με τα διαθέσιμα εργαλεία που μπορούν να χρησιμοποιηθούν για τη δημιουργία επιθέσεων DDoS, να είναι πολυπληθή.

Οι λύσεις ανίχνευσης ανωμαλιών γίνονται όλο και πιο δημοφιλείς, καθώς αυτές οι προσεγγίσεις είναι αποτελεσματικές τόσο κατά των άγνωστων όσο και των παραγώγων γνωστών προτύπων επιθέσεων, με το κανονικό μοτίβο της κυκλοφορίας να μοντελοποιείται για την δημιουργία ενός φυσιολογικού προφίλ, περιλαμβάνοντας την εξαγωγή χαρακτηριστικών πακέτων, ακόμα και σε περιόδους χωρίς επίθεση.

Εικόνα 6 Τάσεις στις τεχνικές ανίχνευσης επιθέσεων DDoS στο cloud μεταξύ του Γενάρη του 2010 και του Δεκέμβρη 2015

Όσον αφορά την αξιολόγηση των προτεινόμενων λύσεων, υπάρχει ένα σοβαρό πρόβλημα αξιοπιστίας, προφανώς λόγω της έλλειψης ενημερωμένων και πραγματικού-κόσμου συνόλων δεδομένων για εκπαίδευση. Τα πιο γνωστά, ωστόσο, σύνολα δεδομένων περιλαμβάνουν ονομαστικά, το UNBISCX2012, το CAIDA DDoS 2007, το DARPA 2000LL-DDoS από το Lincoln Laboratory, το MIT και το KDD'99. Ένα ακόμα καίριο ζήτημα, είναι η έλλειψη διαθέσιμων συνόλων δεδομένων με ετικέτες, με το KDD'99 να είναι ένα από τα λίγα δημόσια διαθέσιμα σύνολα δεδομένων με ετικέτες που χρησιμοποιούνται από τους ερευνητές σήμερα.

Η αναγνώριση και ο διαχωρισμός των μοτίβων, αν είναι κανονικό ή επίθεση, αποτελεί την πιο κρίσιμη μετρική, όσον αφορά την χρησιμότητα και της αποδοτικότητα των προτεινόμενων τεχνικών. Άλλες μετρικές είναι:

- *Ρυθμός ανίχνευσης*: Η ακρίβεια της τεχνικής άμυνας, όσον αφορά την αναγνώριση των προτύπων επίθεσης σε μια ροή κυκλοφορίας.

- *Μέσος χρόνος απόκρισης:* Ο μέσος χρόνος που απαιτείται για έναν χρήστη cloud να ζητήσει και να λάβει υπηρεσίες νέφους, κατά τη διάρκεια μιας επίθεσης.

- *Λόγος επιβίωσης κανονικών πακέτων:* Ο λόγος του συνολικού αριθμού των κανονικών πακέτων, που επιτυγχάνουν την πρόσβαση στο cloud έναντι του συνολικού αριθμού των πακέτων που αποπειράθηκαν την πρόσβαση.

Παρά το μεγάλο πλήθος σε απόπειρες ερευνών σε αυτό τον τομέα, οι προκλήσεις που υπάρχουν είναι ακόμα αρκετές και πρέπει να αντιμετωπιστούν. Παράδειγμα, απαιτείται μια λύση άμυνας που να μπορεί να ανιχνεύει επιθέσεις τόσο σε επίπεδο εφαρμογής όσο και σε επίπεδο υποδομής, αφού τα σύγχρονα εργαλεία επίθεσης DDoS είναι ικανά να εκτελούν στοχεύουν διάφορα στοιχεία του cloud. Επιπλέον, υπάρχει η ανάγκη για αποτελεσματική προσέγγιση όσον αφορά τις βέλτιστες τιμές κατωφλιών για την καθορισμό των προτύπων, σε υπάρχουσες και μελλοντικές τεχνικές άμυνας. Τέλος, για την σωστή εκπαίδευση και αξιολόγηση, χρειάζεται δημιουργία συνόλων δεδομένων με επιλογή βέλτιστων χαρακτηριστικών σύμφωνα με τα τρέχοντα πρότυπα επιθέσεων DDoS, διασφαλίζοντας τη διαθεσιμότητα και την επικαιρότητα τέτοιων συνόλων για εκπαίδευση και δοκιμή.

Κεφάλαιο 3

Αλγόριθμοι Μηχανικής Μάθησης για ανίχνευση DDoS επιθέσεων

3.1 Η Μηχανική Μάθηση στις επιθέσεις DDoS

Στον συνεχώς εξελισσόμενο χώρο της κυβερνοασφάλειας, οι επιθέσεις Κατανεμημένης Άρνησης Υπηρεσίας (DDoS), συνεχίζουν να αποτελούν σημαντική απειλή για τα διαδικτυακά συστήματα και υπηρεσίες. Αυτού του είδους οι επιθέσεις περιλαμβάνουν τον υπερφόρτωση ενός στόχου με υπερβολικό όγκο κυκλοφορίας, καθιστώντας τον μη προσβάσιμο για νόμιμους χρήστες. Με την τεχνολογία να τρέχει, οι επιτιθέμενοι γίνονται πιο εξελιγμένοι και οι παραδοσιακές μέθοδοι ανίχνευσης και αντιμετώπισης των επιθέσεων DDoS αποδεικνύονται ανεπαρκείς. Εδώ είναι όπου εμφανίζεται η εφαρμογή τεχνικών Μηχανικής Μάθησης (ML).

Η Μηχανική Μάθηση προσφέρει έναν ελπιδοφόρο δρόμο για τη βελτίωση των ικανοτήτων ανίχνευσης και αντίδρασης στις επιθέσεις DDoS. Η ΜΜ εκμεταλλεύεται αλγόριθμους και στατιστικά μοντέλα για να επιτρέψει στα συστήματα να μαθαίνουν από δεδομένα, να αναγνωρίζουν πρότυπα και να λαμβάνουν ενημερωμένες αποφάσεις χωρίς ρητό προγραμματισμό. Αυτή η ικανότητα προσαρμογής και εξέλιξης βάσει νέων δεδομένων και νέων τεχνικών επιθέσεων είναι ιδιαίτερα πλεονεκτική στο πλαίσιο των επιθέσεων DDoS, οι οποίες μπορούν να ποικίλλουν σε μεγάλο βαθμό όσον αφορά τους διανυσματικούς προσανατολισμούς, τα πρότυπα κυκλοφορίας και την ένταση.

Οι παραδοσιακές μέθοδοι ανίχνευσης DDoS συχνά βασίζονται σε προκαθορισμένους κανόνες και υπογραφές που προσπαθούν να ταυτίσουν την εισερχόμενη κυκλοφορία με γνωστά πρότυπα επιθέσεων. Ενώ είναι αποτελεσματικές ενάντια σε καλά γνωστές και τεκμηριωμένες μεθόδους επίθεσης, αυτές οι προσεγγίσεις δυσκολεύονται να ανιχνεύσουν νέες και εξελιγμένες επιθέσεις που δεν ταιριάζουν σε προκαθορισμένα προφίλ. Επιπλέον, μπορούν να οδηγήσουν σε υψηλούς ρυθμούς ψευδών θετικών, με αποτέλεσμα να επισημαίνουν τη νόμιμη κυκλοφορία ως κακόβουλη και να προκαλούν διαταραχές στην κανονική λειτουργία.

Τα σημαντικά οφέλη της ΜΜ στην ανίχνευση DDoS είναι:

- ***Προσαρμοστική Μάθηση:*** Τα μοντέλα ΜΜ μπορούν να προσαρμοστούν σε αλλαγές στερεοτυπικών τεχνικών επίθεσης και προτύπων κυκλοφορίας μέσω της συνεχούς μάθησης από νέα δεδομένα. Αυτή η προσαρμοστικότητα είναι κρίσιμη στην εξέλιξη των στρατηγικών επίθεσης. Οι επιθέσεις DDoS δεν περιορίζονται πλέον σε απλές όγκο-κεντρικές επιθέσεις. Οι επιτιθέμενοι χρησιμοποιούν μια πληθώρα στρατηγικών, από

τεχνικές χαμηλής και αργής κυκλοφορίας μέχρι προηγμένες πολυ-διανυσματικές επιθέσεις. Αυτό το δυναμικό περιβάλλον απαιτεί μεθόδους ανίχνευσης που μπορούν να αναγνωρίσουν γρήγορα αναδυόμενα πρότυπα, ακόμη και αυτά που δεν έχουν αντιμετωπιστεί ποτέ προηγουμένως. Η ΜΜ διαπρέπει σε τέτοιες καταστάσεις λόγω της ικανότητάς της να προσαρμόζεται και να μαθαίνει από δεδομένα, επιτρέποντάς της να ανιχνεύει ανωμαλίες και πρότυπα που τα στατικά συστήματα βασισμένα σε κανόνες ενδέχεται να παραβλέπουν.

- ***Ανίχνευση Ανωμαλιών****:* Τα μοντέλα ΜΜ ξεχωρίζουν στην αναγνώριση ανωμαλιών σε μεγάλα σύνολα δεδομένων. Οι επιθέσεις DDoS συχνά οδηγούν σε ανώμαλες αυξήσεις στον όγκο της κυκλοφορίας, τη συχνότητα ή τα πρότυπα. Οι αλγόριθμοι ΜΜ μπορούν να ανιχνεύσουν αυτές τις ανωμαλίες και να εκδίδουν ειδοποιήσεις, ακόμη και όταν αντιμετωπίζουν προηγουμένως άγνωστους προσανατολισμούς επίθεσης.

- ***Αναγνώριση Προτύπων:*** Οι αλγόριθμοι της Μηχανικής Μάθησης μπορούν να αναγνωρίσουν διακριτικά πρότυπα που θα μπορούσαν να περάσουν απαρατήρητα από παραδοσιακές μεθόδους. Αυτά τα πρότυπα θα μπορούσαν να έχουν νέους προσανατολισμούς επίθεσης ή μικροσκοπικές παραλλαγές στους υπάρχοντες. Οι αλγόριθμοι ΜΜ μπορούν να αναλύσουν πολλά χαρακτηριστικά και τις σύνθετες σχέσεις μεταξύ τους, επιτρέποντάς τους να κατανοήσουν πολύπλοκες αλληλεπιδράσεις προτύπων κυκλοφορίας, διακρίνοντας κακόβουλα πρότυπα, ακόμη και όταν προσπαθούν να κρυφτούν μέσα στην κανονική κυκλοφορία.

- ***Μείωση Ψευδών Θετικών*****:** Λαμβάνοντας υπόψη μια ευρύτερη σειρά παραγόντων και συμπεριφορών, τα μοντέλα ΜΜ μπορούν να ελαχιστοποιήσουν τα ψευδή θετικά, μειώνοντας τον αντίκτυπο στους νόμιμους χρήστες και τους πόρους. Τα ψευδώς θετικά μπορούν να έχουν σοβαρές επιπτώσεις στις λειτουργίες και τους πόρους μιας οργάνωσης. Η εξάρτηση μόνο από προσεγγίσεις βασισμένες σε κανόνες μπορεί να «κατηγορεί» την νόμιμη κυκλοφορία, προκαλώντας περιττές διαταραχές και περιορίζοντας την εμπειρία των χρηστών. Τα μοντέλα ΜΜ, λαμβάνοντας υπόψη τη συνολική κυκλοφορία του δικτύου και τις πολλαπλές διαστάσεις, μπορούν σημαντικά να μειώσουν τους ψευδείς θετικούς. Αυτή η ακρίβεια βελτιώνει όχι μόνο την λειτουργική αποδοτικότητα, αλλά μειώνει επίσης τον κίνδυνο του ακούσιου αποκλεισμού των νόμιμων χρηστών.

- ***Αντίδραση σε Πραγματικό Χρόνο****:* Η φύση των επιθέσεων DDoS απαιτεί γρήγορη ανίχνευση και αντίδραση για να μειωθούν οι ζημίες. Τα συστήματα ανίχνευσης DDoS που τροφοδοτούνται από την ΜΜ μπορούν να παρέχουν ανάλυση πραγματικού χρόνου, επιτρέποντας την ταχύτερη αναγνώριση και αντιμετώπιση των επιθέσεων, κάτι το απαραίτητο στην ελαχιστοποίηση των ζημιών. Η δυνατότητα της ΜΜ για ανάλυση πραγματικού χρόνου επιτρέπει την ανίχνευση ανωμαλιών και κακόβουλων πρότυπων καθώς αυτά εμφανίζονται. Αυτή η ταχύτητα είναι κρίσιμη για την παύση της προόδου μιας επίθεσης και για την αποτροπή παρατεταμένης διακοπής λειτουργίας ή διακοπής υπηρεσιών.

Η προσαρμοστικότητα της ΜΜ σημαίνει ότι μπορεί να εξελιχθεί παράλληλα με νέες τεχνικές επίθεσης. Η σημασία της χρήσης της ΜΜ για τη διάκριση μεταξύ φυσιολογικής και επιθετικής κυκλοφορίας στην ανίχνευση DDoS δεν μπορεί να υπερτιμηθεί. Σε ένα περιβάλλον χαρακτηρισμένο από ποικίλα διανυσματικά επιθέσεων και διαρκή εξέλιξη, οι παραδοσιακές

μέθοδοι συχνά αποδυναμώνονται. Η επιδεξιότητα της ΜΜ στην κατανόηση πολύπλοκων προτύπων κυκλοφορίας, τη μείωση των ψευδών θετικών, τη δυνατότητα για γρήγορη ανταπόκριση και την προσαρμοστικότητα στην ασφάλεια υπογραμμίζει την δυνατότητα της να εξελίσσεται, ενισχύοντας την κυβερνοασφάλεια έναντι του συνεχώς αυξανόμενου απειλητικού περιβάλλοντος των επιθέσεων DDoS.

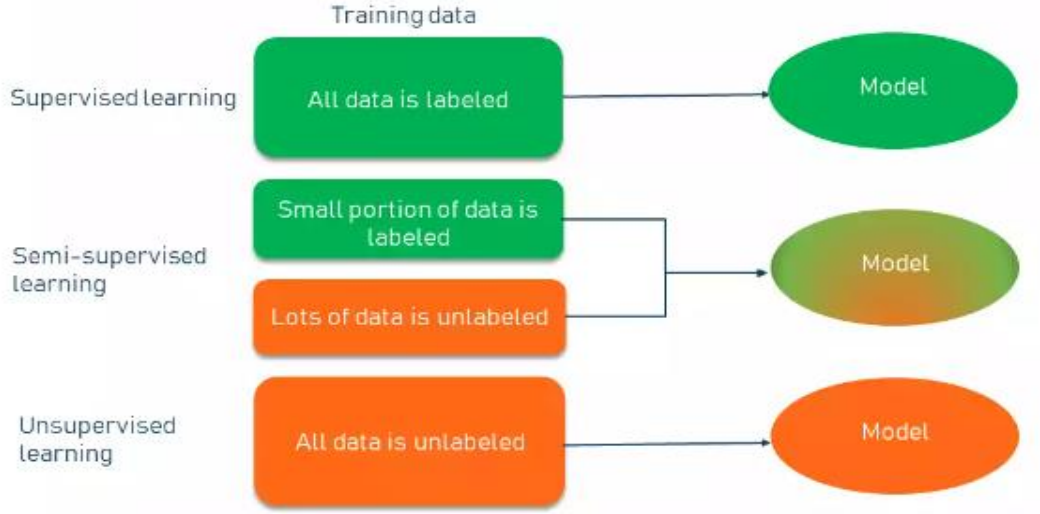
3.2 Τύποι αλγορίθμων Μηχανικής Μάθησης έναντι επιθέσεων DDoS

3.2.1 Μάθηση με επίβλεψη – Supervised learning

Η μάθηση με επίβλεψη (supervised learning) είναι μια προσέγγιση της μηχανικής μάθησης που ορίζεται από τη χρήση συνόλων δεδομένων με ετικέτες. Αυτά τα σύνολα δεδομένων σχεδιάζονται για να εκπαιδεύουν τους αλγορίθμους, ώστε να κατηγοριοποιούν τα δεδομένα ή να προβλέπουν αποτελέσματα με ακρίβεια. Χρησιμοποιώντας εισόδους και εξόδους με ετικέτες, το μοντέλο μπορεί να μετρήσει την ακρίβειά του και να μάθει με την πάροδο του χρόνου.
Η μάθηση με επίβλεψη μπορεί να χωριστεί σε δύο τύπους προβλημάτων:

- Τα προβλήματα **κατηγοριοποίησης (Classification)** χρησιμοποιούν έναν αλγόριθμο για να αντιστοιχίσουν με δεδομένα δοκιμής (test data) σε συγκεκριμένες κατηγορίες, όπως μία διάκριση μεταξύ μήλων και πορτοκαλιών. Στον πραγματικό κόσμο, αυτοί οι αλγόριθμοι μπορούν να χρησιμοποιηθούν για να ταξινομήσουν τα ανεπιθύμητα μηνύματα σε ξεχωριστό φάκελο από τα εισερχόμενα του ηλεκτρονικού σας ταχυδρομείου. Γραμμικοί ταξινομητές (Linear classifiers), Support Vector Machines (SVM), δέντρα απόφασης (Decision Tree) και τυχαίου δάσους ταξινομητές (Random Forest) είναι όλοι κοινοί τύποι αλγορίθμων κατηγοριοποίησης.

- Η **παλινδρόμηση (Regression)** είναι ένας άλλος τύπος μάθησης με επίβλεψη, που χρησιμοποιεί έναν αλγόριθμο για να κατανοήσει τη σχέση μεταξύ εξαρτημένων και ανεξάρτητων μεταβλητών. Τα μοντέλα παλινδρόμησης είναι χρήσιμα για την πρόβλεψη αριθμητικών τιμών με βάση διάφορα σημεία δεδομένων, όπως οι προβλέψεις των εσόδων από πωλήσεις για μια δεδομένη επιχείρηση. Ορισμένοι δημοφιλείς αλγόριθμοι παλινδρόμησης είναι η γραμμική παλινδρόμηση (Linear Regression), η λογιστική παλινδρόμηση (Logistic Regression) και η πολυωνυμική παλινδρόμηση (Polynomial Regression).



Εικόνα 7 Μια σύνοψη της μάθησης με επίβλεψη, χωρίς επίβλεψη και με ημι-επίβλεψη

3.2.2 Μάθηση χωρίς επίβλεψη – Unsupervised learning

Η μάθηση χωρίς επίβλεψη (unsupervised learning) χρησιμοποιεί αλγόριθμους μηχανικής μάθησης για να αναλύσει και να ομαδοποιήσει σύνολα δεδομένων χωρίς ετικέτες. Αυτοί οι αλγόριθμοι ανακαλύπτουν κρυμμένα μοτίβα στα δεδομένα χωρίς την ανάγκη ανθρώπινης παρέμβασης (γι' αυτό και "χωρίς επίβλεψη").
Τα μοντέλα μάθησης χωρίς επίβλεψη χρησιμοποιούνται για τρεις βασικές εργασίες:

- Η **ομαδοποίηση (Clustering)** είναι μια τεχνική εξόρυξης δεδομένων, και αφορά την ομαδοποίηση δεδομένων χωρίς ετικέτες, με βάση τις ομοιότητές ή τις διαφορές τους. Για παράδειγμα, οι αλγόριθμοι ομαδοποίησης K-means, αντιστοιχίζουν παρόμοια δείγματα σε ομάδες, με την τιμή K να αντιπροσωπεύει το πλήθος της ομαδοποίησης. Αυτή η τεχνική είναι χρήσιμη για την κατηγοριοποίηση της αγοράς, συμπίεση εικόνων κ.ά.

- Η **συσχέτιση (Association)** είναι άλλος τύπος μεθόδου χωρίς επίβλεψη, που χρησιμοποιεί κανόνες για να βρει σχέσεις μεταξύ μεταβλητών σε ένα δεδομένο σύνολο. Αυτές οι μέθοδοι χρησιμοποιούνται συχνά για την ανάλυση του καλαθιού αγορών και τα συστήματα συστάσεων.

- Η **μείωση διαστάσεων (Dimensionality Reduction)** είναι μια σημαντική τεχνική μάθησης που χρησιμοποιείται όταν ο αριθμός των χαρακτηριστικών (ή διαστάσεων) σε ένα δεδομένο σύνολο είναι πολύ υψηλός. Μειώνει τον αριθμό των εισόδων δεδομένων σε ένα πιο διαχειρίσιμο μέγεθος ενώ παράλληλα διατηρεί την ακεραιότητα των δεδομένων. Συχνά, αυτή η τεχνική χρησιμοποιείται σαν στάδιο προ-επεξεργασίας δεδομένων, όπως όταν αυτό-κωδικοποιητές, αφαιρούν θόρυβο από οπτικά δεδομένα για την βελτίωση ποιότητας των εικόνων.

- ***Με επίβλεψη εναντίον χωρίς επίβλεψη***

Ο βασικός διαχωρισμός μεταξύ των δύο προσεγγίσεων είναι η χρήση ή όχι ετικετών στα σύνολα δεδομένων. Η μάθηση με επίβλεψη χρησιμοποιεί εισόδους και εξόδους δεδομένων με ετικέτες, ενώ ένας αλγόριθμος μάθησης χωρίς επίβλεψη δεν το κάνει. Στην μάθηση χωρίς επίβλεψη, ο αλγόριθμος "μαθαίνει" από το σύνολο εκπαίδευσης κάνοντας επαναλαμβανόμενες προβλέψεις για τα δεδομένα και προσαρμόζοντας την σωστή απάντηση, ενώ τα μοντέλα μάθησης με επίβλεψη, τείνουν να είναι πιο ακριβή, αλλά απαιτούν ανθρώπινη παρέμβαση εκ των προτέρων για να δώσουν ετικέτες στα δεδομένα με κατάλληλο τρόπο.

Τα μοντέλα μάθησης χωρίς επίβλεψη, αντίθετα, λειτουργούν μόνα τους για να ανακαλύψουν την εγγενή δομή των δεδομένων, χωρίς αυτά να έχουν ετικέτες. Ωστόσο, πρέπει να σημειωθεί πως ακόμα και αυτά τα δεδομένα, χρειάζονται κάποια ανθρώπινη παρέμβαση για την επικύρωση των μεταβλητών εξόδου. Για παράδειγμα, ένα μοντέλο μη επιβλεπόμενης μάθησης μπορεί να αναγνωρίσει ότι οι online αγοραστές συχνά αγοράζουν ομάδες προϊόντων ταυτόχρονα. Ωστόσο, ένας αναλυτής δεδομένων θα πρέπει να επικυρώσει ότι έχει λογικό νόημα η ομαδοποίηση από τον αλγόριθμο.

Στην μάθηση χωρίς επίβλεψη, ο στόχος είναι να αποκτηθούν εισαγωγές από μεγάλο όγκο νέων δεδομένων και επαφίεται στο μοντέλο, να καθορίζει τι είναι διαφορετικό ή ενδιαφέρον από το σύνολο δεδομένων. Η μάθηση με επίβλεψη, είναι μια απλή μέθοδος για χρήση μηχανικής

μάθησης, συνήθως με χρήση της R ή Python, με την μάθηση χωρίς επίβλεψη, να χρειάζονται ισχυρά εργαλεία για την εργασία με μεγάλα μη ταξινομημένα σύνολα δεδομένων, με τα μοντέλα να είναι υπολογιστικά πολύπλοκα, καθώς χρειάζονται μεγάλο σύνολο εκπαίδευσης για να παράγουν τα επιθυμητά αποτελέσματα.

3.2.3 Μάθηση με ημι-επίβλεψη – Semi-supervised learning

Η μάθηση με ημι-επίβλεψη γεφυρώνει τις τεχνικές μάθησης με επίβλεψη και χωρίς επίβλεψη, για να αντιμετωπίσει τις βασικές προκλήσεις και των δύο. Με αυτήν τη μέθοδο, ένα αρχικό μοντέλο εκπαιδεύεται με κάποια δείγματα που έχουν ετικέτες, και στη συνέχεια, εφαρμόζετε επαναληπτικά το μοντέλο αυτό σε ένα μεγαλύτερο αριθμό δεδομένων χωρίς ετικέτες. Αντίθετα με τη μη επιβλεπόμενη μάθηση, η ημι-επιβλεπόμενη μάθηση λειτουργεί για μια ποικιλία προβλημάτων από ταξινόμηση έως ομαδοποίηση και συσχετισμό. Η χρήση μικρού όγκου δεδομένων με ετικέτα και μεγάλων ποσοτήτων χωρίς, μειώνει τα έξοδα και τον χρόνο προετοιμασίας των δεδομένων, με τα δεδομένα χωρίς ετικέτα, να είναι πλούσια, εύκολα προσβάσιμα και φθηνά.[35]

Δύο χαρακτηριστικά παραδείγματα είναι η αυτο-εκπαίδευση και η συν-εκπαίδευση.
Η διαδικασία στην αυτο-εκπαίδευση, είναι η λήψη οποιασδήποτε μεθόδου μάθησης με επίβλεψη, για ταξινόμηση ή παλινδρόμηση, και στην συνέχεια, η τροποποίηση της για να λειτουργεί με ημι-επιβλεπόμενο τρόπο, εκμεταλλευόμενο δεδομένα με ετικέτες και χωρίς. Η ροή εργασίας είναι ως εξής. Επιλέγεται μια μικρή ποσότητα δεδομένων με ετικέτες, για παράδειγμα, εικόνες που δείχνουν γάτες και σκύλους με τις αντίστοιχες ετικέτες τους, και με αυτά τα δεδομένα εκπαιδεύεται ένα βασικό μοντέλο συνηθισμένων μεθόδων μάθησης με επίβλεψη. Στη συνέχεια, εφαρμόζετε μία διαδικασία, γνωστή ως ψευδο-ετικετοποίηση, με το εκπαιδευμένο -εν μέρει- μοντέλο, να χρησιμοποιείται για να κάνετε προβλέψεις στα υπόλοιπα δεδομένα που δεν έχουν ακόμα ετικέτες. Από αυτό το σημείο, παίρνονται οι πιο βέβαιες προβλέψεις που έκανε το μοντέλο, προστίθενται στο σύνολο δεδομένων με ετικέτες και δημιουργείτε ένα νέο σύνολο, συνδυασμένων εισόδων για εκπαίδευση ενός βελτιωμένου μοντέλου.

Η συν-εκπαίδευση, αποτελεί τη βελτιωμένη έκδοση της αυτό-εκπαίδευσης και χρησιμοποιείται όταν υπάρχει μόνο μια μικρή ποσότητα δεδομένων με ετικέτες. Αντίθετα με την τυπική διαδικασία, η συν-εκπαίδευση εκπαιδεύει δύο ξεχωριστούς ταξινομητές βάσει δύο όψεων των δεδομένων. Οι όψεις είναι βασικά διαφορετικά σύνολα χαρακτηριστικών που παρέχουν επιπλέον πληροφορίες για κάθε παράδειγμα, πράγμα που σημαίνει ότι είναι ανεξάρτητες ως προς την κλάση. Επίσης, κάθε όψη είναι επαρκής - η κλάση δείγματος δεδομένων μπορεί να προβλεφθεί με ακρίβεια από κάθε σύνολο χαρακτηριστικών από μόνο του. Η διαδικασία, και πάλι, αρχικά εκπαιδεύει κάθε ταξινομητή σε κάθε όψη, χρησιμοποιώντας μια μικρή ποσότητα από τα δεδομένα με ετικέτες. Στη συνέχεια, προστίθεται το μεγαλύτερο σύνολο δεδομένων χωρίς ετικέτες, για να παραχθούν οι ψευδο-ετικέτες. Οι ταξινομητές συν-εκπαιδεύουν ο ένας τον άλλο χρησιμοποιώντας τις ψευδο-ετικέτες, με το υψηλότερο επίπεδο βεβαιότητας. Εάν ο πρώτος ταξινομητής προβλέπει με βεβαιότητα την πραγματική ετικέτα για ένα δείγμα δεδομένων και ο δεύτερος κάνει λάθος πρόβλεψη, τότε τα δεδομένα με τις βέβαιες ψευδο-ετικέτες που έχουν ανατεθεί από τον πρώτο ταξινομητή, ενημερώνουν τον δεύτερο ταξινομητή και αντιστρόφως. Το τελευταίο βήμα περιλαμβάνει τη συνδυασμένη χρήση των προβλέψεων από τους δύο ενημερωμένους ταξινομητές, για το τελικό αποτέλεσμα ταξινόμησης.

Ενώ η μάθηση με επίβλεψη έχει τα πλεονεκτήματά της, οι τεχνικές μάθησης με επίβλεψη και χωρίς επίβλεψη, χρησιμοποιούνται πιο συχνά στην ανίχνευση επιθέσεων DDoS. Οι μέθοδοι με επίβλεψη, εκμεταλλεύονται δεδομένα με ετικέτες επίθεσης και κανονικών δεδομένων, ενώ οι μέθοδοι χωρίς επίβλεψη, επικεντρώνονται στην αναγνώριση ανωμαλιών στη δικτυακή κίνηση.

3.2.4 Ενισχυτική μάθηση – Reinforcement learning

Η ενισχυτική μάθηση αφορά έναν πράκτορα που εξερευνά ένα άγνωστο περιβάλλον για να επιτύχει ένα στόχο. Η ενισχυτική μάθηση βασίζεται στην υπόθεση ότι όλοι οι στόχοι μπορούν να περιγράφουν από την μεγιστοποίηση της αναμενόμενης συσσωρευτική ανταμοιβής. Ο πράκτορας πρέπει να μάθει να αντιλαμβάνεται την κατάσταση του περιβάλλοντος, με σκοπό να πετύχει την μέγιστη ανταμοιβή. Το τυπικό πλαίσιο της Ενισχυτικής Μάθησης αντλεί από το πρόβλημα του βέλτιστου ελέγχου των Διαδικασιών Λήψης Αποφάσεων του Markov (Markov Decision Processes - MDP)[36].

Μια χρήσιμη μετρική για το άμεσο όφελος, είναι η συνάρτηση αξίας, η οποία αποτυπώνει τη συσσωρευτική ανταμοιβή που αναμένεται να συλλεχθεί από εκείνη την κατάσταση και μετά, στο διάστημα που ακολουθεί. Ο στόχος ενός αλγορίθμου ενισχυτικής μάθησης, είναι να ανακαλύψει την πολιτική ενεργειών, που θα μεγιστοποιήσει την μέση τιμή της συνάρτησης αξίας που μπορεί να εξαχθεί από κάθε κατάσταση του συστήματος.

Οι αλγόριθμοι ενισχυτικής μάθησης μπορούν να κατηγοριοποιηθούν σε αλγορίθμους χωρίς μοντέλο (model-free) και με μοντέλο (model-based). Οι αλγόριθμοι χωρίς μοντέλο, δεν κατασκευάζουν ένα ρητό μοντέλο περιβάλλοντος (δηλαδή MDP), αλλά είναι πιο κοντά σε αλγορίθμους δοκιμής-λάθους που εκτελούν πειράματα στο περιβάλλον και παράγουν την βέλτιστη πολιτική ενεργειών.

Οι αλγόριθμοι χωρίς μοντέλο είναι είτε βασισμένοι στην αξία είτε στην πολιτική. Οι αλγόριθμοι βασισμένοι στην αξία θεωρούν ότι η βέλτιστη πολιτική αποτελεί αποτέλεσμα της εκτίμησης της συνάρτησης αξίας κάθε κατάστασης, με ακρίβεια. Χρησιμοποιώντας μια αναδρομική σχέση, ο πράκτορας αλληλοεπιδρά με το περιβάλλον για να δειγματοληπτήσει μονοπάτια καταστάσεων και ανταμοιβών. Δεδομένου αρκετών μονοπατιών, μπορεί να εκτιμηθεί η συνάρτηση αξίας του MDP. Μόλις γνωρίζεται η συνάρτηση αξίας, η ανακάλυψη της βέλτιστης πολιτικής είναι απλώς θέμα κριτηρίου απληστότητας σε σχέση με την συνάρτηση αξίας σε κάθε κατάσταση της διαδικασίας.

Από την άλλη πλευρά, οι αλγόριθμοι βασισμένοι στην πολιτική εκτιμούν απευθείας την βέλτιστη πολιτική, χωρίς να μοντελοποιούν τη συνάρτηση αξίας. Με το να βρίσκουν την πολιτική απευθείας χρησιμοποιώντας βάρη που εκπαιδεύονται, μετατρέπουν το πρόβλημα μάθησης σε ένα πρόβλημα βελτιστοποίησης. Όπως και με τους αλγορίθμους βασισμένους στην αξία, ο πράκτορας δειγματοληπτεί μονοπάτια καταστάσεων και ανταμοιβών, όμως τώρα, ο στόχος είναι η απευθείας βελτίωση της πολιτική, με απώτερο σκοπό τη μεγιστοποίηση της μέσης τιμής της συνάρτησης αξίας σε όλες τις καταστάσεις.

Οι αλγόριθμοι ενισχυτικής μάθησης με μοντέλο (model-based), κατασκευάζουν ένα μοντέλο του περιβάλλοντος δειγματοληπτώντας τις καταστάσεις, παίρνοντας ενέργειες και παρατηρώντας τις ανταμοιβές. Για κάθε κατάσταση και δυνατή ενέργεια, το μοντέλο προβλέπει την αναμενόμενη ανταμοιβή και την αναμενόμενη μελλοντική κατάσταση. Η πρώτη είναι ένα πρόβλημα παλινδρόμησης, ενώ η δεύτερη είναι ένα πρόβλημα εκτίμησης πυκνότητας. Δεδομένου ενός μοντέλου του περιβάλλοντος, ο πράκτορας μπορεί να σχεδιάσει τις ενέργειές του χωρίς να αλληλοεπιδρά άμεσα με το περιβάλλον, κάτι το οποίο θυμίζει την ανθρώπινη πορεία σκέψης στην προσπάθεια να επίλυσης ενός προβλήματος.

Ωστόσο, στο πλαίσιο των επιθέσεων DDoS, με το περιβάλλον ιδιαιτέρα πολύπλοκο, η δημιουργία ενός ακριβούς μοντέλου είναι δύσκολη. Ακόμα, οι επιθέσεις DDoS απαιτούν αντιδράσεις πραγματικού χρόνου, ενώ οι πράκτορες για να μάθουν, απαιτούν σημαντικό χρόνο

και με τις επιθέσεις DDoS να μπορούν να πάρουν πολλές μορφές, οι πράκτορες αντιμετωπίζουν δυσκολίες στο να προσαρμοστούν γρήγορα σε νέα και απρόβλεπτα πρότυπα επίθεσης, με το παραπάνω να ολοκληρώνει την αντίληψη για την αδύναμη παρουσία της ενισχυτικής μάθησης στον χώρο πρόληψης επιθέσεων DDoS.

3.3 Θεμελιώδεις αλγόριθμοι Μηχανικής Μάθησης με επίβλεψη

3.3.1 Logistic Regression (LR)

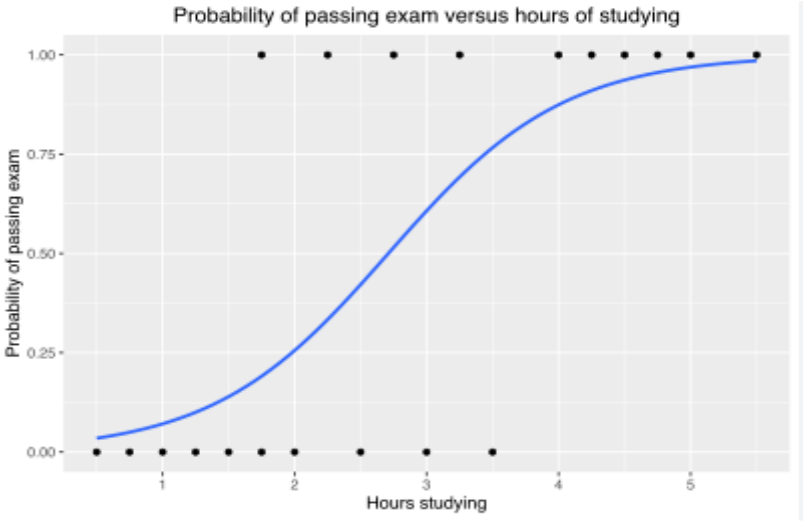
Η LR ανήκει στην οικογένεια των μοντέλων μηχανικής μάθησης με επίβλεψη. Σαν αλγόριθμος ταξινόμησης, χρησιμοποιείται συνήθως για την επίλυση προβλημάτων δυαδικής ταξινόμησης. Στη δυαδική ταξινόμηση, υπάρχουν δεδομένα με δύο πιθανές κλάσεις ή αποτελέσματα (συνήθως 0 και 1), και ο στόχος είναι η κατασκευή ενός μοντέλου που μπορεί να προβλέψει σε ποια κλάση ανήκουν νέα σημεία δεδομένων με βάση τα χαρακτηριστικά τους.

Η LR εκτιμά την πιθανότητα ενός συμβάντος να συμβεί, όπως ψήφισε ή δεν ψήφισε, βασισμένη σε ένα σύνολο δεδομένων ανεξάρτητων μεταβλητών. Εφόσον το αποτέλεσμα είναι μια πιθανότητα, η εξαρτώμενη μεταβλητή περιορίζεται μεταξύ του 0 και 1. Ο τρόπος με τον οποίο η LR λειτουργεί, είναι προσπαθώντας να μεγιστοποιήσει τη συνάρτηση της μέγιστης πιθανοφάνειας (maximum likelihood) $L_{\mathbf{w},b} \overset{\text{def}}{=} \prod_{i=1\ldots N} f_{\mathbf{w},b}(\mathbf{x}_i)^{y_i}(1 - f_{\mathbf{w},b}(\mathbf{x}_i))^{(1-y_i)}$

με $f_{\mathbf{w},b}(\mathbf{x}) \overset{\text{def}}{=} \frac{1}{1+e^{-(\mathbf{w}\mathbf{x}+b)}}$

για να προσδιορίσει τους συντελεστές του μοντέλου, δοκιμάζοντας διάφορες τιμές του βήτα μέσω πολλαπλών επαναλήψεων, ώστε να βρει την καλύτερη δυνατή εφαρμογή.

Αφού βρεθεί ο βέλτιστος συντελεστής (ή συντελεστές για περισσότερες από μία ανεξάρτητες μεταβλητές), μπορούν να υπολογιστούν οι όροι πιθανότητας για κάθε παρατήρηση και υπολογιστούν, μπορεί να προκύψει μια προβλεπόμενη πιθανότητα. Για δυαδική ταξινόμηση, μια πιθανότητα μικρότερη από .5 θα προβλέψει 0 ενώ μια πιθανότητα μεγαλύτερη από 0.5, θα προβλέψει 1.



Ξεκινώντας από το σύνολο δεδομένων που περιέχει παραδείγματα με ετικέτες, κάθε παράδειγμα έχει ένα σύνολο χαρακτηριστικών (ανεξάρτητες μεταβλητές) και μια αντίστοιχη ετικέτα κλάσης (0 ή 1). Η LR προσπαθεί να βρει την καλύτερη εφαρμογή μιας καμπύλης σχήματος S (καμπύλη σιγμοειδούς) που αντιστοιχεί τα χαρακτηριστικά στην πιθανότητα ανήκει στη θετική κλάση (κλάση 1). Ένα παράδειγμα αυτής της σιγμοειδούς καμπύλης δίνεται στην

εικόνα 8, με την κατηγοριοποίηση να έχει να κάνει με την πιθανότητα προβιβασμού από μια εξέταση ανάλογα με το διάβασμα.

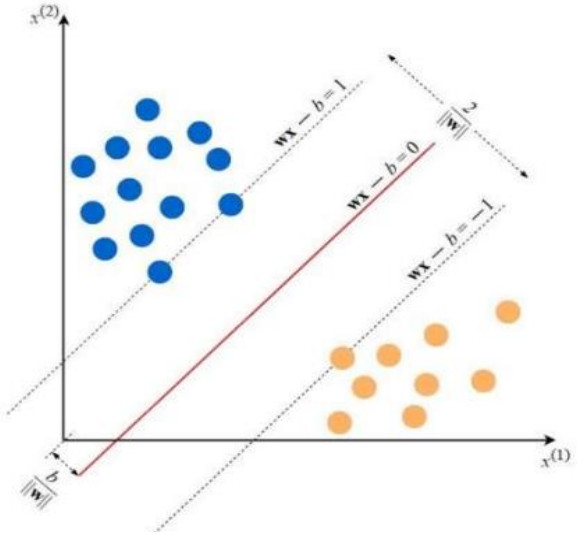
Ο αλγόριθμος προσαρμόζει τους συντελεστές που σχετίζονται με κάθε χαρακτηριστικό για να ελαχιστοποιήσει την συνάρτηση απώλειας, που μετρά τη διαφορά ανάμεσα στις προβλεπόμενες πιθανότητες και τις πραγματικές ετικέτες κλάσης. Για ένα συγκεκριμένο σύνολο χαρακτηριστικών εισόδου, υπολογίζεται ο γραμμικός συνδυασμός $wx+b$ και στη συνέχεια περνάει μέσω της συνάρτησης σιγμοειδούς, για να βγει η προβλεπόμενη πιθανότητα.[37]

3.3.2 Support Vector Machine (SVM)

Το Support Vector Machine (SVM) είναι ένας αλγόριθμος μηχανικής μάθησης με επίβλεψη, που χρησιμοποιείται κυρίως για κατηγοριοποίηση. Αποτελεί έναν ισχυρό αλγόριθμο, που βρίσκει ένα υπερεπίπεδο σε έναν χώρο υψηλής διάστασης για να χωρίσει βέλτιστα διάφορες κατηγορίες δεδομένων. Ο βασικός στόχος ενός SVM είναι να βρει το καλύτερο δυνατό υπερεπίπεδο που μεγιστοποιεί το περιθώριο μεταξύ των κατηγοριών, επιτρέποντας καλύτερη γενίκευση σε αδιάκριτα δεδομένα. Τα SVM είναι γνωστά για την ικανότητά τους να χειριστούν δεδομένα υψηλής διάστασης, την ανθεκτικότητά τους έναντι της υπερεκπαίδευσης και την ευελιξία τους μέσω της χρήσης διαφορετικών συναρτήσεων πυρήνα. Ωστόσο, μπορεί να είναι χρονοβόρα για μεγάλα σύνολα δεδομένων και απαιτούν προσεκτική ρύθμιση των παραμέτρων, όπως η παράμετρος κανονικοποίησης (C) και η επιλογή του πυρήνα.

Κάθε σημείο δεδομένων αναπαρίσταται ως ένα διάνυσμα σε έναν χώρο υψηλών διαστάσεων. Ο στόχος είναι να βρεθεί ένα υπερεπίπεδο που να διαχωρίζει βέλτιστα τα σημεία δεδομένων διαφορετικών κλάσεων. Στην μηχανική μάθηση, το όριο που χωρίζει τα παραδείγματα διαφορετικών κλάσεων, λέγεται όριο απόφασης (decision boundary). Η εξίσωση του υπερεπιπέδου, δίνεται από δύο παραμέτρους, ένα διάνυσμα w ίδιων διαστάσεων με το χαρακτηριστικό διάνυσμα x και έναν πραγματικό αριθμό b: wx + b .

Ο αλγόριθμος προβλέπει την κλάση του παραδείγματος x μέσω της συνάρτησης: y = sign(wx − b), όπου sign είναι μαθηματική συνάρτηση που δέχεται ως είσοδο οποιαδήποτε τιμή και επιστρέφει +1 σε περίπτωση που η είσοδος είναι θετικός αριθμός, ενώ στην αντίθετη περίπτωση επιστρέφει -1. Στόχος του SVM αλγορίθμου είναι το νοητό υπερεπίπεδο, που χωρίζει τα θετικά από τα αρνητικά παραδείγματα να έχει όσο το δυνατόν μεγαλύτερο περιθώριο, την απόσταση των δύο πιο κοντινών παραδειγμάτων που ανήκουν σε κάθε κλάση. Αυτά τα πλησιέστερα σημεία δεδομένων ονομάζονται «υποστηρικτικά διανύσματα». Για να επιτευχθεί αυτό πρέπει να ελαχιστοποιηθεί η Ευκλείδεια νόρμα του w (συμβολίζεται ||w||) και δίνεται από τον τύπο $\sqrt{\sum_{j=1}^{D}(w|j)^2}$, όπου D το πλήθος των διαστάσεων.



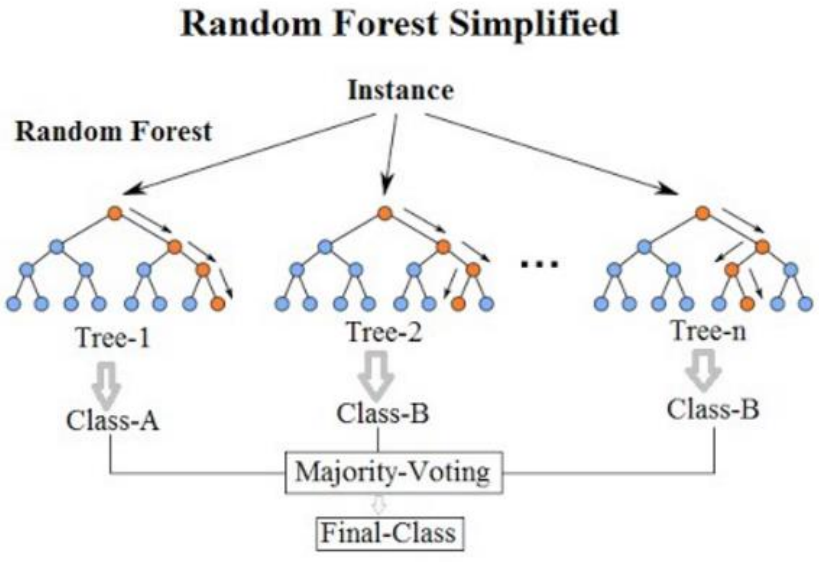
Εικόνα 9 Παράδειγμα διαχωρισμού δυο διαστάσεων με υπερεπίπεδο

Τα SVM επιτρέπουν ένα μαλακό περιθώριο, πράγμα που σημαίνει ότι επιτρέπουν λίγες λανθασμένες ταξινομήσεις προκειμένου να επιτευχθεί ένας καλύτερος συνολικός διαχωρισμός. Η ισορροπία μεταξύ της μεγιστοποίησης του περιθωρίου και της ανοχής λανθασμένων ταξινομήσεων ελέγχεται από την παράμετρο κανονικοποίησης C. Μπορούν να αντιμετωπίσουν αποτελεσματικά περιπτώσεις όπου τα δεδομένα δεν είναι γραμμικά διαχωρίσιμα στον αρχικό χώρο χαρακτηριστικών. Η τεχνική πυρήνα περιλαμβάνει τη μετατροπή των δεδομένων σε έναν χώρο υψηλότερων διαστάσεων χρησιμοποιώντας μια συνάρτηση πυρήνα (π.χ. πολυωνυμική, συνάρτηση βάσης βαθμίδας), καθιστώντας τα δεδομένα γραμμικά διαχωρίσιμα.

Μόλις το μοντέλο εκπαιδευτεί, εξαρτάται μόνο από ένα υποσύνολο των δεδομένων εκπαίδευσης που ονομάζονται υποστηρικτικά διανύσματα. Ο πίνακας πυρήνα, που περιλαμβάνει τα εσωτερικά γινόμενα μεταξύ όλων των ζευγαριών σημείων δεδομένων, μπορεί να προϋπολογιστεί εκ των προτέρων για να επιταχυνθούν οι προβλέψεις. Για να ταξινομήσει ένα νέο σημείο δεδομένων, το SVM αξιολογεί τη θέση του σε σχέση με το υπερεπίπεδο. Εάν βρίσκεται σε μία πλευρά, ταξινομείται ως μία κλάση, εάν βρίσκεται στην άλλη πλευρά, ταξινομείται ως η άλλη κλάση.[38]

3.3.3 Random Forest Classifier (RFC)

Το Random Forest Classifier είναι μια δημοφιλής μέθοδος μηχανικής μάθησης με επίβλεψη, που χρησιμοποιείται τόσο για ταξινόμηση όσο και για προβλήματα παλινδρόμησης. Αποτελεί μια επέκταση του αλγορίθμου Decision Tree Classifier και είναι γνωστός για την αποτελεσματικότητά και την ευελιξία του. Οι κυριότεροι άξονες λειτουργίας του ορίζονται παρακάτω. Αρχικά, ο αλγόριθμος ακολουθεί την διαδικασία του bootstrapping. Συγκεκριμένα, δημιουργεί πολλά δέντρα απόφασης (Decision Trees), για να εισαγάγει ποικιλία μεταξύ αυτών των δέντρων, χρησιμοποιώντας την τεχνική της τυχαίας δειγματοληψίας με αντικατάσταση. Δηλαδή για κάθε δέντρο, δημιουργείται ένα τυχαίο υποσύνολο του αρχικού συνόλου εκπαίδευσης. Αυτό το τυχαίο υποσύνολο περιέχει περίπου δύο τρίτα των αρχικών δεδομένων, και ορισμένα σημεία δεδομένων μπορεί να επαναληφθούν, ενώ άλλα μπορεί να παραλειφθούν. Κάθε δέντρο στο δάσος εκπαιδεύεται σε ένα από αυτά τα τυχαία υποσύνολα.



Εικόνα 10 Απλοποιημένη σχηματοποίηση του Random Forest Classifier

Στην συνέχεια, εκτός από τη δειγματοληψία δεδομένων, το RFC εισάγει επίσης τυχαιότητα στην επιλογή των χαρακτηριστικών κατά τη λήψη αποφάσεων σε κάθε κόμβο του δέντρου. Αντί να λαμβάνει υπόψη όλα τα χαρακτηριστικά, λαμβάνει υπόψη μόνο ένα τυχαίο υποσύνολο των χαρακτηριστικών για κάθε πιθανή διακλάδωση. Αυτή η διαδικασία ονομάζεται τυχαία επιλογή χαρακτηριστικών και ο στόχος της είναι να μειωθεί η

συσχέτιση μεταξύ των μεμονωμένων δέντρων στο δάσος, καθιστώντας τα πιο ποικίλα και λιγότερο πιθανά να υπερεκπαιδεύονται.

Κάθε δέντρο απόφασης στο RFC αναπτύσσεται ανεξάρτητα. Συνήθως, τα δέντρα απόφασης επιτρέπεται να αναπτυχθούν βαθιά μέχρι να φτάσουν σε ένα συγκεκριμένο βάθος ή μέχρι να πληρούν ένα κριτήριο διακλάδωσης, όπως ένα ελάχιστο αριθμό δειγμάτων που απαιτούνται για μια διακλάδωση ή μέχρι η περαιτέρω διακλάδωση να μην βελτιώνει την «καθαρότητα» (για ταξινόμηση) ή το μέσο τετραγωνικό σφάλμα (για παλινδρόμηση). Για τα προβλήματα ταξινόμησης, όταν έρχεται η στιγμή να γίνει μια πρόβλεψη, κάθε δέντρο στο δάσος ψηφίζει για την κλάση που προβλέπει. Η κλάση με τις περισσότερες ψήφους ανάμεσα σε όλα τα δέντρα γίνεται η τελική προβλεπόμενη κλάση. Αυτό ονομάζεται "πλειοψηφία ψήφων". Για προβλήματα παλινδρόμησης, κάθε δέντρο παρέχει μια αριθμητική πρόβλεψη. Η τελική πρόβλεψη για το RFC είναι ο μέσος (μέσος) όρος όλων των προβλέψεων που γίνονται από τα μεμονωμένα δέντρα.

Το τελικό αποτέλεσμα του RFC, είναι το συγκεντρωμένο αποτέλεσμα όλων των μεμονωμένων προβλέψεων των δέντρων. Αυτή η συγκέντρωση μειώνει τη διακύμανση και τείνει να παράγει πιο σταθερές και ακριβείς προβλέψεις σε σύγκριση με ένα μεμονωμένο δέντρο απόφασης.

Τα RFC ανήκουν στην οικογένεια του ensemble learning, η οποία περιλαμβάνει την χρήση πολλών μοντέλων για να πετύχει καλύτερες προβλέψεις. Συνδυάζοντας τις προβλέψεις πολλών δέντρων απόφασης που εκπαιδεύονται σε ελαφρώς διαφορετικά υποσύνολα δεδομένων και χαρακτηριστικών, τα RFC παρέχουν έναν ισχυρό και ευέλικτο αλγόριθμο μηχανικής μάθησης που μπορεί να αντιμετωπίσει μια ευρεία γκάμα εργασιών, ακόμα και με σύνολα δεδομένων με ανισορροπία και παρέχουν καλύτερη ακρίβεια από ένα μόνο δέντρο απόφασης, έχοντας ψηλή ακρίβεια και μικρό κίνδυνο υπερεκπαίδευσης[39].

3.3.4 Naïve Bayes (NB)

Ο Naïve Bayes είναι ένας πιθανοτικός αλγόριθμος μηχανικής μάθησης που χρησιμοποιείται κυρίως για εργασίες ταξινόμησης. Βασίζεται στο Θεώρημα του Bayes και ονομάζεται "αφελής" επειδή κάνει μια απλοποιημένη υπόθεση, ότι τα χαρακτηριστικά που χρησιμοποιούνται στην ταξινόμηση είναι υπό συνθήκη ανεξάρτητα. Αυτή η υπόθεση ονομάζεται "αφελής" και απλοποιεί σημαντικά τον υπολογισμό ενώ παράγει καλά αποτελέσματα, ιδιαίτερα στην ταξινόμηση κειμένου και στην ανίχνευση ανεπιθύμητης αλληλογραφίας (spam filtering). Η λειτουργία του συνοψίζεται στα εξής:

- **Το Θεώρημα του Bayes:** Ο NB βασίζεται στο Θεώρημα του Bayes, ένα θεμελιώδες θεώρημα πιθανοτικής θεωρίας που χρησιμοποιείται για προβλέψεις ή εικασίες βάσει των στοιχείων. Το Θεώρημα του Bayes διατυπώνεται ως εξής:

  $$P(A|B)= \frac{P(B \vee A) \cdot P(A)}{P(B)}$$

  - P(A|B): Η πιθανότητα του συμβάντος A να συμβεί δεδομένου ότι συνέβη το συμβάν B.
  - P(B|A): Η πιθανότητα του συμβάντος B να συμβεί δεδομένου ότι συνέβη το συμβάν A.
  - P(A): Η απόσταση πιθανότητα του συμβάντος A.
  - P(B): Η απόσταση πιθανότητα του συμβάντος B.

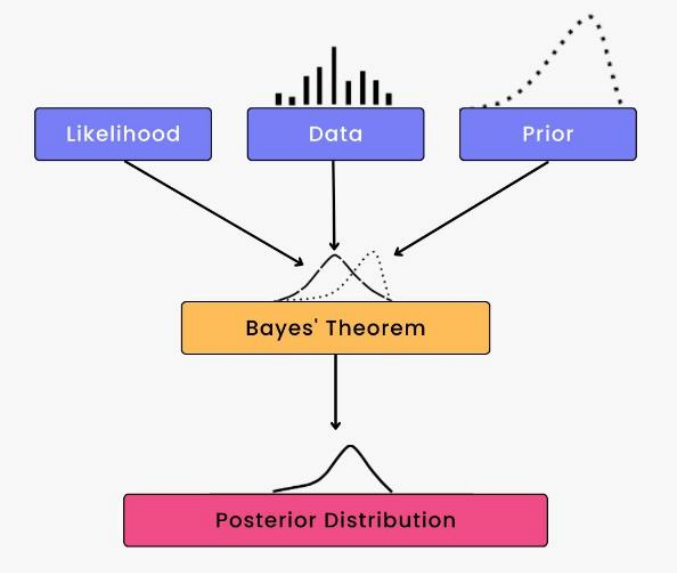
Στο πλαίσιο της ταξινόμησης με τον NB, το A αντιπροσωπεύει την ετικέτα της κλάσης ή την κατηγορία που θέλουμε να προβλέψουμε και το B αντιπροσωπεύει τα χαρακτηριστικά γενικά ή τα χαρακτηριστικά που συσχετίζονται με τα δεδομένα.

- **Η Αφελής Υπόθεση:** Η "αφελής" υπόθεση προέρχεται από την υπόθεση, ότι τα χαρακτηριστικά που χρησιμοποιούνται για την ταξινόμηση είναι υπό συνθήκη ανεξάρτητα, δεδομένης της ετικέτας της κλάσης. Αυτό σημαίνει ότι η παρουσία ή απουσία ενός χαρακτηριστικού δεν επηρεάζει την παρουσία ή απουσία άλλου χαρακτηριστικού. Μαθηματικά, αυτό μπορεί να γραφεί ως εξής:

$$P(X1,X2,\ldots,X_n|C) = P(X_1|C) \cdot P(X_2|C) \cdot \ldots \cdot P(X_n|C)$$

όπου P(Xi|C) είναι η πιθανότητα του χαρακτηριστικού *Xi* δεδομένη την κλάση *C*.

- **Εκπαίδευση του Ταξινομητή NB:** Για να εκπαιδευτεί ένας ταξινομητής NB, χρειάζεται ένα σύνολο δεδομένων με ετικέτες και παραδείγματα διάφορων κλάσεων, με τις αντίστοιχες τιμές των χαρακτηριστικών τους. Αυτό καθιστά τον NB, έναν αλγόριθμο μάθησης με επίβλεψη. Ο αλγόριθμος υπολογίζει τις ακόλουθες πιθανότητες για κάθε κλάση:
  - *Προεπιλεγμένη Πιθανότητα P(C):* Η πιθανότητα της κάθε κλάσης να εμφανίζεται στο σύνολο δεδομένων. Μπορεί να εκτιμηθεί ως το κλάσμα των παραδειγμάτων στο σύνολο εκπαίδευσης που ανήκουν σε κάθε κλάση.
  - *Πιθανότητα Συνθήκης Κλάσης $P(X_i/C)$:* Για κάθε χαρακτηριστικό Xi και κλάση C, η πιθανότητα παρατήρησης του Xi δεδομένης της κλάσης C. Υπολογίζεται με βάση τις εμφανίσεις του Xi μέσα στα παραδείγματα της κλάσης C στα δεδομένα εκπαίδευσης.



Εικόνα 11 Σχηματικά ο Naive Bayes

- **Πρόβλεψη:** Μόλις ο ταξινομητής εκπαιδευτεί, μπορεί να χρησιμοποιηθεί για προβλέψεις σε νέα δεδομένα. Δεδομένου ενός συνόλου χαρακτηριστικών X για ένα νέο σημείο δεδομένων, ο ταξινομητής υπολογίζει την πιθανότητα κάθε κλάσης C χρησιμοποιώντας το Θεώρημα του Bayes:

$$P(C|X) = \frac{P(X \vee C) \cdot P(C)}{P(X)}$$

με P(C|X), η πιθανότητα ότι το σημείο δεδομένων ανήκει στην κλάση C δεδομένων τα χαρακτηριστικά X, P(X|C) το γινόμενο των πιθανοτήτων συνθήκης των χαρακτηριστικών για κάθε χαρακτηριστικό και P(C) είναι η προεπιλεγμένη πιθανότητα της κλάσης C.
Ο ταξινομητής αναθέτει την ετικέτα στην κλάση με την μεγαλύτερη πιθανότητα:

$$\text{Προβλεπόμενη Κλάση} = argmax_C P(C|X)$$

- **Εξομάλυνση:** Στην πράξη, ορισμένες τιμές χαρακτηριστικών ενδέχεται να μην εμφανίζονται στα δεδομένα εκπαίδευσης για μια συγκεκριμένη κλάση, οδηγώντας σε μηδενικές πιθανότητες. Για την αντιμετώπιση αυτού του προβλήματος, συχνά χρησιμοποιούνται τεχνικές εξομάλυνσης όπως η εξομάλυνση Laplace (πρόσθετη εξομάλυνση) για να αποφευχθούν μηδενικές πιθανότητες και να γίνει ο ταξινομητής πιο ανθεκτικός.

Ο αλγόριθμος NB, είναι υπολογιστικά αποδοτικός και κλιμακώνεται καλά με μεγάλα σύνολα δεδομένων και χώρους χαρακτηριστικών υψηλών διαστάσεων και αντιμετωπίζει αρκετά

καλά δεδομένα με ανισορροπία, ειδικά όταν χρησιμοποιείται εξομάλυνση (smoothing). Ωστόσο, σε περίπλοκα προβλήματα, που παρουσιάζουν δεδομένα υψηλών διαστάσεων, όπως εικόνες, ήχος ή σενάρια όπου η ανεξαρτησία χαρακτηριστικών δεν ισχύει, ο NB μπορεί να είναι ανεπαρκής. Ένα χαρακτηριστικό παράδειγμα, είναι η ανάλυση της κυκλοφορίας δικτύου, όπου πολλά χαρακτηριστικά είναι συνδεδεμένα και εξαρτώνται μεταξύ τους. Η αρχή λειτουργίας του NB, υποθέτει ανεξαρτησία μεταξύ των χαρακτηριστικών, πράγμα που δυσχεραίνει την αντιμετώπιση περιπτώσεων με ισχυρές εξαρτήσεις μεταξύ τους (π.χ. επιθέσεις DDoS).

3.3.5 Decision Tree Classifier (DTC)

Το Decision Tree Classifier είναι ένας αλγόριθμος μάθησης με επίβλεψη, που κατασκευάζει μια δομή παρόμοια με ένα δέντρο, για να λαμβάνει αποφάσεις ή προβλέψεις βασισμένες στα χαρακτηριστικά εισόδου. Το δέντρο είναι, ουσιαστικά, ένας μη-περιοδικός γράφος που χρησιμοποιείται για τη λήψη αποφάσεων. Σε κάθε κόμβο διακλάδωσης του γράφου, εξετάζεται ένα συγκεκριμένο χαρακτηριστικό *j* του χαρακτηριστικού διανύσματος. Εάν η τιμή του χαρακτηριστικού είναι κάτω από ένα συγκεκριμένο κατώφλι που έχει οριστεί σύμφωνα με το πρόβλημα, τότε ακολουθείται η αριστερή διακλάδωση, ενώ σε αντίθετη περίπτωση η δεξιά. Φτάνοντας στον κόμβο του φύλλου (leaf node), έχει ληφθεί η απόφαση για την κλάση στην οποία ανήκει το χαρακτηριστικό. Στον αλγόριθμο DTC, τα δεδομένα εισόδου είναι παραδείγματα με κλάσεις, που ανήκουν στο σετ {0,1}.

Το δέντρο αποφάσεων ξεκινά ως ένας μόνο κόμβος που αποκαλείται «κόμβος ρίζα». Κάθε κόμβος στο δέντρο αντιπροσωπεύει μια απόφαση ή ένα τεστ σχετικά με ένα χαρακτηριστικό. Οι κόμβοι συνδέονται με κλαδιά, και οι τελικοί κόμβοι ονομάζονται "φύλλα" ή "τερματικοί κόμβοι," και περιέχουν την ετικέτα της κλάσης (στην περίπτωση της ταξινόμησης) ή την προβλεπόμενη τιμή (στην περίπτωση της παλινδρόμησης). Το δέντρο κατασκευάζεται αναδρομικά χρησιμοποιώντας μια διαδικασία που ονομάζεται «αναδρομικός διαμερισμός» (recursive partitioning). Σε κάθε κόμβο, ο αλγόριθμος επιλέγει το καλύτερο χαρακτηριστικό και τον καλύτερο διαχωριστική τιμή για το χαρακτηριστικό, αξιολογώντας διάφορα κριτήρια, όπως η μείωση του μέσου τετραγωνικού σφάλματος. Το επιλεγέν χαρακτηριστικό και σημείο διαχωρισμού, χρησιμοποιούνται για να διαχωρίσουν τα δεδομένα σε υποσύνολα, που είναι όσο το δυνατόν πιο «καθαρά» σε σχέση με τις ετικέτες των κλάσεων (για ταξινόμηση) ή όσο το δυνατόν πιο ομοιόμορφα (για παλινδρόμηση). Η διαδικασία αυτή, συνεχίζεται αναδρομικά για κάθε υποσύνολο μέχρι να συναντηθεί ένα κριτήριο διακοπής, όπως η επίτευξη του μέγιστου βάθους, ο ελάχιστος αριθμός δειγμάτων σε έναν κόμβο ή η επίτευξη ελάχιστου επιπέδου καθαρότητας.

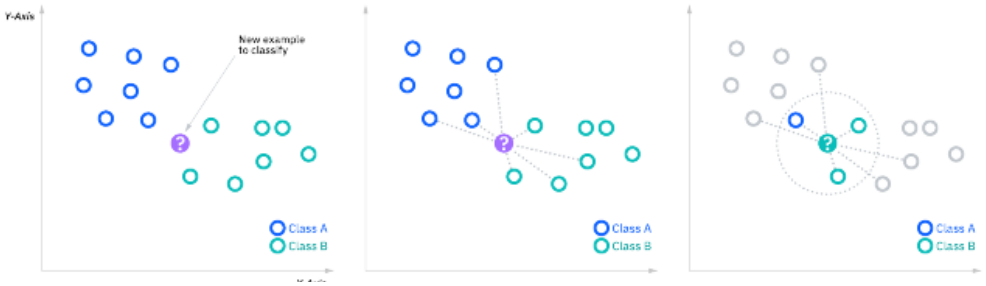
Για πρόβλεψη ή ταξινόμηση, ξεκινάει από τον ριζικό κόμβο και ακολουθούνται τα κλαδιά με βάση τις τιμές των χαρακτηριστικών εισόδου. Σε κάθε κόμβο, ελέγχεται το χαρακτηριστικό έναντι του επιλεγέντος διαχωριστικού σημείου. Ανάλογα με το αποτέλεσμα του ελέγχου, μετακινείται στον αριστερό ή δεξιό κόμβο-παιδί, έως ότου να φτάσει σε ένα φύλλο κόμβο, του οποίου, η ετικέτα της κλάσης ή η προβλεπόμενη τιμή αποτελεί την τελική πρόβλεψη.

Τα DTC, μπορούν να χειριστούν τόσο αριθμητικά όσο και κατηγορικά χαρακτηριστικά και εξαίρουν στην διαχείριση ελλείπων δεδομένων, μέσω αποφάσεων βασισμένων στα διαθέσιμα χαρακτηριστικά, δύο δυνατότητες ιδιαίτερα χρήσιμες στην ανίχνευση των DDoS. Επιπλέον, είναι πολύ απλά στην ερμηνεία τους, απαιτούν ελάχιστη προ-επεξεργασία δεδομένων (όπως κλιμάκωση και κανονικοποίηση) και μπορούν να εντοπίσουν μη γραμμικές σχέσεις στα δεδομένα. Ωστόσο, τα DTC, είναι επιρρεπή σε overfitting, ιδίως όταν το δέντρο είναι βαθύ και πολύπλοκο, είναι ευαίσθητα σε μικρές διακυμάνσεις στα δεδομένα και αν κλάσεις είναι μη ισορροπημένες, τα δέντρα που παράγει είναι παραμορφωμένα.

3.3.6 K-Nearest Neighbors (KNN)

Ο αλγόριθμος k-Nearest Neighbors (k-NN), γνωστός και ως k-NN, είναι ένας αλγόριθμος μη-παραμετρικής μάθησης με επίβλεψη, που χρησιμοποιεί την απόσταση, για να πραγματοποιεί ταξινομήσεις ή προβλέψεις σχετικά με την ομαδοποίηση ενός μεμονωμένου σημείου δεδομένων. Ενώ μπορεί να χρησιμοποιηθεί και για προβλήματα παλινδρόμησης (regression), συνήθως χρησιμοποιείται ως αλγόριθμος ταξινόμησης, λαμβάνοντας υπόψη ότι παρόμοια σημεία μπορούν να βρεθούν κοντά το ένα στο άλλο.

Για προβλήματα ταξινόμησης, ανατίθεται μια ετικέτα τάξης με βάση την πλειοψηφία των ψήφων - δηλαδή η ετικέτα που είναι πιο συχνά αντιπροσωπευμένη γύρω από ένα δεδομένο σημείο δεδομένων χρησιμοποιείται, με κύρια διαφορά με την παλινδρόμηση, να είναι ότι στην ταξινόμηση οι τιμές είναι διακριτές, ενώ στην παλινδρόμηση συνεχείς. Πριν χρησιμοποιηθεί ο k-NN, πρέπει να καθοριστεί η απόσταση μεταξύ των σημείων. Η πιο συνηθισμένη μετρική απόστασης που χρησιμοποιείται είναι η Ευκλείδεια απόσταση $d(x,y) = \sqrt{\sum_{i=1}^{n}(yi - xi)^2}$, αλλά χρησιμοποιούνται και άλλες μετρικές, όπως η Απόσταση του Manhattan ή η Απόσταση Minkowski, ανάλογα με τη φύση των δεδομένων.



Εικόνα 12 Παράδειγμα λειτουργίας k-NN για την ταξινόμηση του μωβ σημείου.

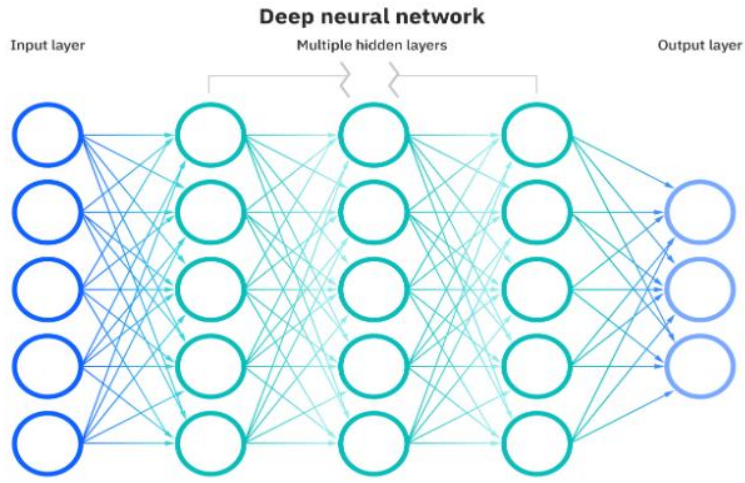
Ακόμα, υπάρχει και η επιλογή της τιμής του k, που αντιπροσωπεύει τον αριθμό των πλησιέστερων γειτόνων που θα ληφθούν υπόψη. Μια μικρή τιμή του k (π.χ., 1 ή 3) καθιστά το μοντέλο πιο ευαίσθητο στο θόρυβο στα δεδομένα αλλά μπορεί να οδηγήσει σε υπερ-προσαρμογή (overfitting). Μια μεγαλύτερη τιμή του k (π.χ., 5, 10 ή περισσότερο) παρέχει μια πιο ομαλή πρόβλεψη αλλά μπορεί να οδηγήσει σε υπο-προσαρμογή (underfittng) των δεδομένων, με την επιλογή του k να εξαρτάται από το εκάστοτε πρόβλημα και σύνολο δεδομένων.

Ο k-NN χρησιμοποιείται ευρέως λόγω της απλότητάς του, αλλά είναι σημαντικό να επιλεγεί προσεκτικά η τιμή του k και η μετρική απόστασης για βέλτιστη απόδοση, καθ' ότι μπορεί να είναι αποδοτικό σε μικρά ή μεσαία μεγέθη συνόλων δεδομένων, αλλά να μην λειτουργεί με υψηλότερες διαστάσεις δεδομένων. Επιπλέον, k-NN αποτελεί έναν lazy learning αλγόριθμο, που αποθηκεύει τα δεδομένα εκπαίδευσης και δεν τα υποβάλλει σε στάδιο εκπαίδευσης. Από την άλλη, πέρα από την χρονική του πολυπλοκότητα σε μεγάλα δεδομένα και την ευαισθησία του στην επιλογή του k, ο αλγόριθμος υποφέρει από την "κατάρα των διαστάσεων" όταν χρησιμοποιείται με υψηλές διαστάσεις δεδομένων, με την έννοια της απόστασης να γίνεται λιγότερο σημαντική και, τέλος, την απαίτηση για κανονικοποίηση προκειμένου να εξασφαλιστεί ότι όλα τα χαρακτηριστικά έχουν την ίδια επίδραση στον υπολογισμό της απόστασης.

3.3.7 Neural Networks (NN)

Το όνομά και η δομή τους είναι εμπνευσμένα από τον ανθρώπινο εγκέφαλο, μιμούμενα τον τρόπο με τον οποίο βιολογικοί νευρώνες μεταδίδουν σήματα μεταξύ τους. Αποτελούνται από στρώματα κόμβων, περιλαμβάνοντας ένα επίπεδο εισόδου, ένα ή περισσότερα κρυμμένα επίπεδα

και ένα επίπεδο εξόδου. Ο διαχωρισμός των NN, ανάλογα με το πλήθος των επιπέδων, χαρακτηρίζει τα NN με περισσότερα από τρία επίπεδα, σαν «βαθιά μάθηση» (deep learning), αλλιώς απλά «νευρωνικό δίκτυο». Κάθε κόμβος, ή νευρώνας, συνδέεται με έναν άλλον και έχει ένα συσχετισμένο βάρος και ένα κατώφλι (threshold). Εάν το αποτέλεσμα οποιουδήποτε ατομικού κόμβου βρίσκεται πάνω από την καθορισμένη τιμή κατωφλίου, ο συγκεκριμένος κόμβος ενεργοποιείται και στέλνει δεδομένα στο επόμενο επίπεδο του δικτύου. Διαφορετικά, δεν προωθείται κανένα δεδομένο στο επόμενο επίπεδο.



Εικόνα 13 Παράδειγμα NN deep learning

Τα νευρωνικά δίκτυα βασίζονται σε δεδομένα εκπαίδευσης για να μάθουν και να βελτιώνουν την ακρίβειά τους με την πάροδο του χρόνου. Ωστόσο, εφόσον έχουν ρυθμιστεί για ακρίβεια, αποτελούν ισχυρά εργαλεία τεχνητής νοημοσύνης, επιτρέποντάς ταξινόμηση και να ομαδοποίηση δεδομένων με υψηλή ταχύτητα.

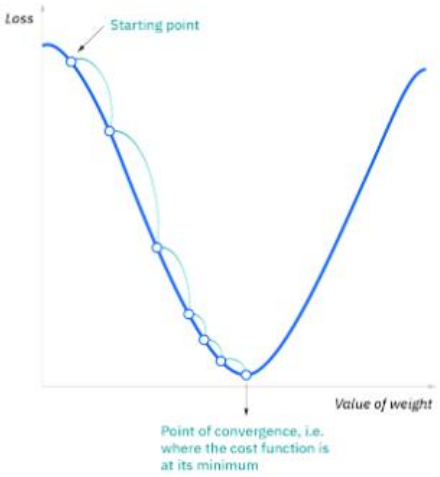
Πρέπει να επισημανθεί πως τα NN, μπορούν να χρησιμοποιηθούν και σαν μάθηση με επίβλεψη, αλλά και χωρίς επίβλεψη. Συγκεκριμένα, στην μάθηση με επίβλεψη, εκπαιδεύονται σε δεδομένα με ετικέτες, όπου κάθε είσοδος συσχετίζεται με μια αντίστοιχη επιθυμητή έξοδο. Το δίκτυο μαθαίνει να αντιστοιχίζει τις εισόδους στις εξόδους, προσαρμόζοντας τις παραμέτρους του κατά τη διάρκεια της εκπαίδευσης για την ελαχιστοποίηση ενός προκαθορισμένου σφάλματος ή συνάρτησης σφάλματος. Στην μάθηση χωρίς επίβλεψη, τα NN, εκπαιδεύονται σε δεδομένα χωρίς ετικέτες, με στόχο να ανακαλύψουν μοτίβα ή δομές μέσα στα δεδομένα, χωρίς επίβλεψη. Ένας γνωστός τύπος NN αυτής της κατηγορίας, είναι ο αυτοκωδικοποιητής (autoencoder), ο οποίος στοχεύει να μάθει μια συμπιεσμένη αναπαράσταση των εισόδων και στη συνέχεια να τις ανακατασκευάσει[40].

Πρακτικά, κάθε ατομικός κόμβος λειτουργεί σαν ένα μοντέλο linear regression, αποτελούμενο από δεδομένα εισόδου, βάρη, ένα κατώφλι και μια έξοδο. Με τον προσδιορισμό του επιπέδου εισόδου, ανατίθενται βάρη. Αυτά τα βάρη βοηθούν στον προσδιορισμό της σημασίας κάθε μεταβλητής, με τα μεγαλύτερα να συμβάλλουν περισσότερο στην έξοδο σε σύγκριση με άλλες εισόδους. Εν συνεχεία, όλες οι είσοδοι πολλαπλασιάζονται με τα αντίστοιχα βάρη τους και αθροίζονται. Το αποτέλεσμα περνά από μια συνάρτηση ενεργοποίησης, η οποία καθορίζει την έξοδο. Αυτή η διαδικασία περάσματος δεδομένων από ένα επίπεδο στο επόμενο ορίζει αυτό το νευρικό δίκτυο ως ένα δίκτυο feedforward. Τα περισσότερα νευρικά δίκτυα είναι feedforward, πράγμα που σημαίνει ότι ρέουν μόνο προς μία κατεύθυνση, από την είσοδο στην έξοδο. Ωστόσο, υπάρχουν και μοντέλα αντίθετης κατεύθυνσης διαδρομής (backpropagation), δηλαδή κίνησης από την έξοδο προς την είσοδο. Με το backpropagation, υπάρχει η δυνατότητα υπολογισμού σφάλματος σε κάθε νευρώνα, επιτρέποντάς την προσαρμογή των παραμέτρων του μοντέλου αναλόγως.

Η αξιολόγηση της ακρίβειας του NN, γίνεται μέσω μιας συνάρτησης κόστους, συνήθως, του μέσου τετραγωνικού σφάλματος (MSE). Στην παρακάτω εξίσωση, το **i** είναι το νούμερο του δείγματος, **yi΄** είναι το προβλεπόμενο αποτέλεσμα, **yi** είναι η πραγματική τιμή, και **m** είναι ο αριθμός των δειγμάτων.

$Cost\ Function = MSE = \frac{1}{2m}\sum_{i=1}^{m}(yi' - yi)^2$

με τελικό στόχο την ελαχιστοποίηση της. Καθώς το μοντέλο προσαρμόζει στα δεδομένα, χρησιμοποιεί τη συνάρτηση κόστους μάθησης για να φτάσει στο σημείο σύγκλισης ή τοπικό ελάχιστο.



Εικόνα 14 Σχηματική αναπαράσταση σύγκλισης σε τοπικό ελάχιστο

Τα feedforward NN, ή multi-layer perceptrons (MLPs), αποτελούνται από ένα επίπεδο εισόδου, ένα κρυφό επίπεδο ή περισσότερα, και ένα επίπεδο εξόδου. Είναι σημαντικό να σημειωθεί ότι αποτελούνται πραγματικά από σιγμοειδείς νευρώνες, καθώς τα περισσότερα πραγματικά προβλήματα είναι μη γραμμικά.

Τα Convolutional NNs (CNNs) είναι παρόμοια με τα feedforward δίκτυα, αλλά συνήθως χρησιμοποιούνται για την αναγνώριση εικόνων και στην όραση υπολογιστών. Εκμεταλλεύονται αρχές από τη γραμμική άλγεβρα, κυρίως τον πολλαπλασιασμό πινάκων, για να αναγνωρίσουν μοτίβα σε μια εικόνα.

Τα Recursive NNs (RNNs),αναγνωρίζονται από τα επαναληπτικά κυκλώματα τους, και χρησιμοποιούνται κυρίως με χρονοσειριακά δεδομένα, για πρόβλεψη μελλοντικών αποτελεσμάτων.

3.3.8 Gradient Boosting (GRB)

Το Gradient Boosting είναι μια τεχνική μηχανικής μάθησης με επίβλεψη, που χρησιμοποιείται τόσο για προβλήματα παλινδρόμησης όσο και για ταξινόμηση. Ανήκει στην οικογένεια του ensemble learning, όπου ο συνδυασμός πολλών μοντέλων δημιουργεί ένα πιο ισχυρό και ακριβές μοντέλο. Η λειτουργία του GRB, ξεκινά με ένα αρχικό μοντέλο, κατά κανόνα με δέντρο απόφασης (Decision Tree) σταθερού βάθους, το οποίο εν συνεχεία εκπαιδεύεται στο σύνολο εκπαίδευσης, και οι προβλέψεις του χρησιμοποιούνται ως αφετηρία.

Μετά, υπολογίζεται η διαφορά μεταξύ των πραγματικών τιμών-στόχων και των προβλέψεων που έγιναν από το αρχικό μοντέλο. Αυτές οι διαφορές ονομάζονται υπολείμματα ή σφάλματα.

Ένα νέο δέντρο απόφασης, εκπαιδεύεται για να προβλέπει αυτές τι διαφορές, αντί για τις αρχικές τιμές-στόχους και επικεντρώνεται στα σφάλματα που προκλήθηκαν από το αρχικό μοντέλο. Οι προβλέψεις που έγιναν από το αρχικό μοντέλο ενημερώνονται προσθέτοντας τις προβλέψεις που έγιναν από το νέο μοντέλο, με σκοπό να διορθώσει τα σφάλματα που προκλήθηκαν αρχικά. Η διαδικασία αυτή επαναλαμβάνεται για ένα προκαθορισμένο αριθμό επαναλήψεων ή μέχρι να επιτευχθεί ένα συγκεκριμένο επίπεδο ακρίβειας. Σε κάθε επανάληψη, εκπαιδεύεται ένα νέο μοντέλο στα υπολείμματα από την προηγούμενη επανάληψη.

Η τελική πρόβλεψη πραγματοποιείται συνδυάζοντας τις προβλέψεις όλων των μοντέλων. Για προβλήματα ταξινόμησης, που μας αφορούν, η πρόβλεψη περιλαμβάνει την λήψη της πλειοψηφίας (majority voting). Το βασικό στο GRB είναι ότι κάθε νέο μοντέλο, παρότι από μόνο του, αποτελεί έναν αδύναμο μαθητή (weak learner), προστίθεται στο σύνολο και εκπαιδεύεται για να αποτυπώσει τα σφάλματα ή υπολείμματα, που αφήνουν τα προηγούμενα μοντέλα. Αυτή η σειριακή διαδικασία μάθησης, βελτιώνει την απόδοσή με κάθε επανάληψη, καταλήγοντας σε ένα ισχυρό και ακριβές μοντέλο.

Η πιο δημοφιλής υλοποίηση του GRB, είναι το Gradient Boosting Machine (GBM), με άλλες γνωστές να είναι το XGBoost και το LightGBM, το καθένα με βελτιστοποιήσεις και πρόσθετα χαρακτηριστικά για πιο αποδοτική διαδικασία εκπαίδευσης.

3.4 Θεμελιώδεις αλγόριθμοι Μηχανικής Μάθησης χωρίς επίβλεψη

3.4.1 Isolation Forest (ISO FOR)

Ο αλγόριθμος Isolation Forest είναι ένας αλγόριθμος ανίχνευσης ανωμαλιών, που χρησιμοποιείται στη μηχανική μάθηση χωρίς επίβλεψη, για τον εντοπισμό ανωμαλιών σε ένα σύνολο δεδομένων. Είναι ιδιαίτερα αποτελεσματικός, ειδικά σε δεδομένα υψηλών διαστάσεων, καθιστώντας τον κατάλληλο για εργασίες όπως η ανίχνευση απάτης, η ασφάλεια δικτύων και ο εντοπισμός επιθέσεων DDoS.

Η κεντρική ιδέα είναι πως οι ανωμαλίες είναι συνήθως σπάνιες και διαφορετικές από την πλειοψηφία των σημείων δεδομένων. Είναι, έτσι, «απομονωμένες» περιπτώσεις στον χώρο χαρακτηριστικών. Ο αλγόριθμος αναδρομικά διαμερίζει το σύνολο δεδομένων με την τυχαία επιλογή ενός χαρακτηριστικού και μιας τυχαίας τιμής διαίρεσης μεταξύ των ελάχιστων και μέγιστων τιμών αυτού του χαρακτηριστικού. Αυτή η διαδικασία συνεχίζεται μέχρι να απομονώσει κάθε σημείο δεδομένων σε μοναδικά υποσύνολα.

Αυτό που αναμένεται, είναι ότι οι ανωμαλίες θα απομονωθούν πιο γρήγορα από τα κανονικά σημεία δεδομένων, αφού απαιτούν λιγότερες διαιρέσεις για να απομακρυνθούν από την πλειοψηφία των δεδομένων. Για να καθοριστεί αυτό, υπάρχει ένα σκορ ανωμαλίας σε κάθε σημείο δεδομένων με βάση το μέσο μήκος διαδρομής που απαιτείται για να το απομονώσει. Τα σημεία που απομονώνονται γρήγορα λαμβάνουν υψηλότερα σκορ ανωμαλίας. Στην συνέχεια, μένει μόνο να οριστεί ένα όριο (threshold), πάνω από το οποίο τα σημεία θα μαρκάρονται ως ανωμαλίες.

Παρόλο που το Isolation Forest είναι αποτελεσματικό στην ανίχνευση ανωμαλιών, ειδικά για δεδομένα υψηλής διάστασης και ικανό να χειριστεί μεγάλα σύνολα δεδομένων, δεν είναι τόσο καλό όταν οι ανωμαλίες είναι πυκνές ή συσσωρευμένες μαζί. Ακόμα, σαν αλγόριθμος χωρίς επίβλεψη, χρειάζεται, λογικά, σημαντικό μέγεθος συνόλου δεδομένων εκπαίδευσης, για να πετύχει καλά αποτελέσματα[41].

3.4.2 Local Outlier Factor (LOF)

Ο Local Outlier Factor (LOF) είναι ένα αλγόριθμος ανίχνευσης ανωμαλιών, που χρησιμοποιείται στην μηχανική μάθηση χωρίς επίβλεψη για τον εντοπισμό ανωμαλιών σε ένα σύνολο δεδομένων και είναι, και αυτός, ιδιαίτερα χρήσιμος σε δεδομένα υψηλών διαστάσεων όπου παραδοσιακές μέθοδοι όπως προσεγγίσεις με απόσταση ενδέχεται να μη λειτουργούν αποτελεσματικά.

Ο LOF ξεκινά με τον υπολογισμό της τοπικής πυκνότητας των σημείων δεδομένων. Για κάθε σημείο στο σύνολο δεδομένων, υπολογίζει την πυκνότητά του σε σχέση με τα γειτονικά σημεία δεδομένων. Η πυκνότητα είναι ουσιαστικά μια μέτρηση του πόσο κοντά βρίσκεται το σημείο στα γειτονικά του. Τα σημεία με υψηλότερη πυκνότητα θεωρούνται ότι βρίσκονται σε πυκνότερες περιοχές των δεδομένων, ενώ τα σημεία με χαμηλότερη πυκνότητα βρίσκονται σε αραιές περιοχές. Μετά από αυτόν τον υπολογισμό για κάθε σημείο δεδομένων, ο LOF συγκρίνει την πυκνότητα ενός σημείου με τις πυκνότητες των γειτονικών του σημείων. Συγκεκριμένα, εξετάζει πώς η πυκνότητα ενός σημείου συγκρίνεται με τις πυκνότητες των k-πλησιέστερων γειτόνων του, όπου το 'k' είναι μία παράμετρος που ορίζεται από τον χρήστη. Στόχος είναι να βρεθούν σημεία που έχουν σημαντικά χαμηλότερη πυκνότητα από τους γείτονές τους.

Το σκορ LOF για ένα σημείο δεδομένων υπολογίζεται ως ο μέσος όρος του λόγου της πυκνότητάς του προς τις πυκνότητες των k-πλησιέστερων γειτόνων του. Μαθηματικά, το σκορ LOF για ένα σημείο δεδομένων 'p' ορίζεται ως:

$LOF(p) = \frac{Πυκνότητα(p)}{ΜέσηΠυκνότητα(των k-πλησιέστερων γειτόνων του p)}$ με ένα ψηλό σκορ LOF, να υποδηλώνει ότι το σημείο δεδομένων 'p' βρίσκεται σε αραιή περιοχή σε σύγκριση με τους γείτονές του, καθιστώντας το δυνητικά ανώμαλο. Για τον εντοπισμό ανωμαλιών, ορίζεται ένα κατώφλι για τα σκορ LOF, πάνω από το οποίο θεωρούνται ανωμαλίες και από κάτω, κανονικά σημεία δεδομένων.

Ο LOF είναι ένας αλγόριθμος βασισμένος στην πυκνότητα, πράγμα που σημαίνει ότι είναι ευαίσθητος στην τοπική κατανομή των σημείων δεδομένων. Ακόμα, η επιλογή της παραμέτρου 'k' είναι κρίσιμη, με μία μικρή τιμή να φέρνει ευαισθησία στον τοπικό θόρυβο, ενώ μία μεγαλύτερη να τον καθιστά λιγότερο ευαίσθητο σε μικρά σύνολα ανωμαλιών. Τέλος, μπορεί να είναι υπολογιστικά ακριβός για μεγάλα σύνολα δεδομένων, καθώς απαιτεί τον υπολογισμό αποστάσεων μεταξύ των σημείων δεδομένων.

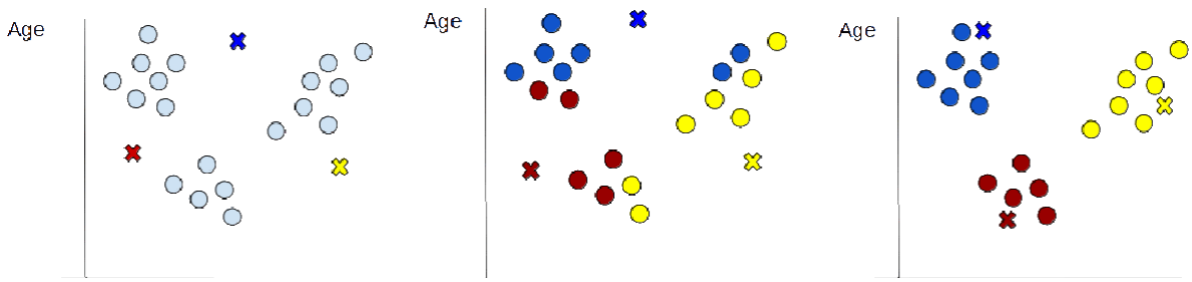
3.4.3 K-Means

Ο K-Means είναι ένας αλγόριθμος clustering (ομαδοποίησης) μηχανικής μάθησης χωρίς επίβλεψη και χρησιμοποιείται για την ομαδοποίηση ενός συνόλου σημείων δεδομένων σε συστάδες βάσει των ομοιοτήτων τους. Ο στόχος είναι να χωρίσει τα δεδομένα σε K συστάδες (clusters), όπου κάθε σημείο δεδομένων ανήκει στη συστάδα με το πλησιέστερο μέσο (centroid).

Αρχικά, επιλέγεται η τιμή του K, τα αρχικά centroids τυχαία από τα δεδομένα. Για κάθε σημείο δεδομένων, υπολογίζεται η Ευκλείδεια απόστασή του από όλα τα K centroids και ανατίθεται στη συστάδα του πλησιέστερου centroid. Αυτό το βήμα αναθέτει κάθε σημείο δεδομένων στην πλησιέστερη συστάδα. Στην συνέχεια, ξανα-υπολογίζονται οι θέσεις των centroids της κάθε συστάδας, βρίσκοντας την μέση τιμή όλων των σημείων δεδομένων που έχουν ανατεθεί στη συστάδα αυτή. Αυτά τα νέα centroids, αντιπροσωπεύουν το "κέντρο" της κάθε συστάδας.

Συνεχίζοντας την επανάληψη των βημάτων ανάθεσης και ενημέρωσης, ο αλγόριθμος συγκλίνει ελαχιστοποιώντας το άθροισμα των τετραγώνων των αποστάσεων μεταξύ των σημείων δεδομένων και των αντίστοιχων κέντρων τους το μοντέλο φτάνει στην σύγκλιση, και τελειώνει όταν τα κέντρα δεν αλλάζουν πλέον σημαντικά την θέση τους ή μετά από ορισμένο αριθμό επαναλήψεων. Όταν ο αλγόριθμος συγκλίνει, υπάρχουν K συστάδες, και κάθε σημείο δεδομένων ανήκει σε μία από αυτές τις συστάδες.

Τα βασικά θετικά του K-Means είναι η ευκολία στην χρήση του και η δυνατότητα του σε κλιμάκωση μεγάλων συνόλων δεδομένων. Ωστόσο, είναι ευαίσθητος στην αρχική επιλογή των centroid, με διαφορετικές αρχικοποιήσεις, να οδηγούν σε διαφορετικές τελικές αναθέσεις συστάδας, αν και αυτό αντιμετωπίζεται με πολλές εκτελέσεις με διαφορετικές αρχικοποιήσεις, και επιλογή του καλύτερου αποτελέσματος. Ακόμη, συχνά, ο K-Means, συγκλίνει σε τοπικά ακρότατα, πράγμα που σημαίνει ότι ενδέχεται να μην βρει την βέλτιστη λύση.

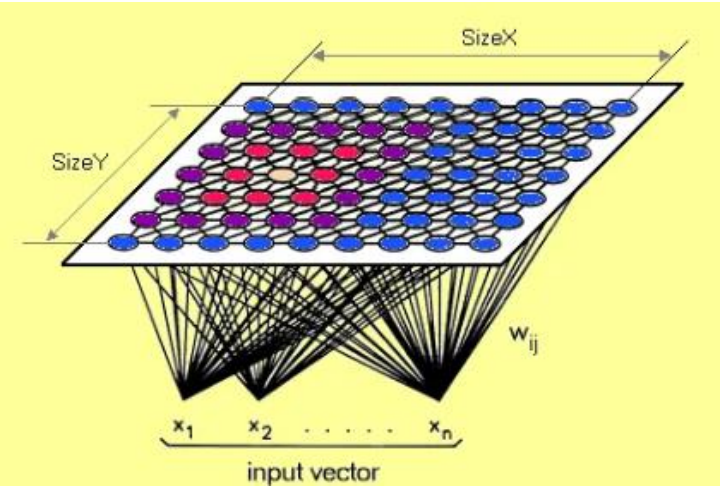


Εικόνα 15 Παράδειγμα χρήσης του K-Means με K=3 clusters

3.4.4 Self-Organizing Maps (SOMs)

Ένα Self-Organizing Map ή SOM, γνωστό και ως χάρτης Kohonen, είναι ένας τύπος τεχνητού νευρωνικού δικτύου (NN) που σχεδιάστηκε για μάθηση χωρίς επίβλεψη, οπτικοποίηση δεδομένων και μείωση της διαστάσεων δεδομένων (dimensionality reduction). Το τελευταίο, γίνεται μέσω διακριτής αποτύπωσης των δεδομένων εισόδου-εκπαίδευσης, συνήθως σε δύο διαστάσεις, εξ ου και το map (χάρτης). Τα SOMs, διαφέρουν από τα άλλα NN, αφού χρησιμοποιούν ανταγωνιστική μάθηση και όχι διόρθωσης σφαλμάτων(π.χ. Gradient Descent με backpropagation) και εισάγουν, στην ουσία, μία συνάρτηση γειτνίασης, για να διατηρήσουν τα τοπολογικά χαρακτηριστικά των δεδομένων εισόδου.

Η λειτουργία τους, συνάδει με το competitive learning. Η καρδιά ενός SOM αποτελείται από ένα πλέγμα νευρώνων, συνήθως οργανωμένων σε μία δισδιάστατη διάταξη, με κάθε νευρώνα στο πλέγμα, να αντιπροσωπεύει μια μοναδική θέση στον χώρο των εισόδων. Αυτή διάταξη των νευρώνων στο πλέγμα είναι που επιτρέπει στα SOMs να διατηρούν τις τοπολογικές σχέσεις των εισόδων. Κάθε νευρώνας στο πλέγμα έχει συσχετισμένα βάρη, τα οποία αποτελούν ουσιαστικά, ένα διάνυσμα τιμών. Το πλήθος των διαστάσεών αυτών των διανυσμάτων των βαρών είναι ίδιο με το πλήθος των διαστάσεων των εισόδων και, αρχικά, αυτά τα βάρη ορίζονται σε τυχαίες τιμές, που επιλέγονται από τα δεδομένα εισόδου. Ο ρυθμός μάθησης είναι μία παράμετρος που ελέγχει το μέγεθος των προσαρμογών των βαρών κατά τη διάρκεια της εκπαίδευσης. Αρχίζει με ένα σχετικά υψηλό επίπεδο και συνήθως μειώνεται σταδιακά κατά τη διάρκεια της εκπαίδευσης. Τέλος, η συνάρτηση γειτνίασης, καθορίζει τη χωρική έκταση επίδρασης που έχει ένας νικητής νευρώνας (Best Matching Unit - BMU) στους γειτονικούς του νευρώνες. Συνήθως αναπαρίσταται ως γκαουσιανή συνάρτηση με κέντρο το BMU.



Εικόνα 16 Αποτύπωση διανύσματος εισόδου σε χάρτη SOM

Πιο αναλυτικά, η αρχικοποίηση γίνεται με την δημιουργία του πλέγματος των νευρώνων και την αρχικοποίηση των βαρών τους. Η διαδικασία εκπαίδευσης είναι επαναληπτική και λειτουργεί ως εξής. Ένα σημείο δεδομένων επιλέγεται τυχαία από το σύνολο δεδομένων και παρουσιάζεται στο SOM. Το SOM καθορίζει ποιο νευρώνας έχει τα βάρη που είναι πιο παρόμοια με τα δεδομένα εισόδου. Αυτός ο νευρώνας ονομάζεται η BMU. Τα βάρη της BMU και των γειτονικών της νευρώνων προσαρμόζονται ώστε να γίνουν όσο το δυνατόν παρόμοια με τα δεδομένα εισόδου, με το ποσό της προσαρμογής εξαρτάται από τον ρυθμό μάθησης και τη συνάρτηση γειτνίασης. Η BMU λαμβάνει τη μεγαλύτερη προσαρμογή και η προσαρμογή μειώνεται με την απόσταση από αυτήν. Ο ρυθμός μάθησης μειώνεται σταδιακά, επιτρέποντας μεγαλύτερες προσαρμογές στα βάρη στα αρχικά στάδια της εκπαίδευσης και πιο λεπτές προσαρμογές αργότερα. Το ίδιο συμβαίνει και με την ακτίνα της γειτονίας (ελέγχεται από τη συνάρτηση γειτονίας), περιορίζοντας σταδιακά την επίδραση των πιο απομακρυσμένων νευρώνων. Τα παραπάνω επαναλαμβάνονται για κάποιον συγκεκριμένο αριθμό επαναλήψεων και μετά την εκπαίδευση, το πλέγμα νευρώνων του SOM οργανώνεται, ώστε παρόμοια σημεία δεδομένων να αντιστοιχούν σε κοντινούς νευρώνες, διατηρώντας τις τοπολογικές σχέσεις των δεδομένων. Μετά την δημιουργία του, ο χάρτης, μπορεί να χρησιμοποιηθεί σε πολλές τεχνικές εκ νέου, όπως σε αλγόριθμους clustering, regression η classification, με μειωμένα πλέον

χαρακτηριστικά σε σχέση με τα αρχικά και ακόμα για ανάλυση δεδομένων ή και visualization (οπτικοποίηση).

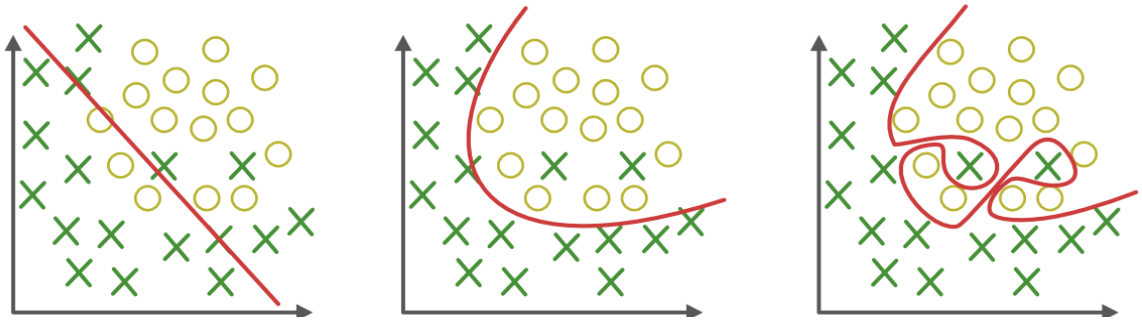
Η κυριότερη και πιο χρήσιμη δυνατότητα που προσφέρουν τα SOM είναι η μείωση διαστάσεων (dimensionality reduction), κυρίως λόγω της ικανότητας τους στην διατήρηση των τοπολογικών σχέσεων των δεδομένων εισόδου και της ισχυρής ικανότητας στην ανίχνευση μοτίβων[42].

3.5 Underfitting και Overfitting

Ένα μοντέλο έχει χαμηλό bias, όταν προβλέπει σωστά τις ετικέτες του training set. Όταν το μοντέλο κάνει αρκετά λάθη στις προβλέψεις του, τότε έχει υψηλό bias και προκύπτει το φαινόμενο του underfitting. Με απλά λόγια, το μοντέλο δεν είναι αρκετά σύνθετο για να ταιριάξει καλά με τα δεδομένα. Αυτό οδηγεί σε υψηλό σφάλμα εκπαίδευσης και υψηλό σφάλμα ελέγχου, υποδηλώνοντας ότι το μοντέλο δεν μπορεί να κάνει ακριβείς προβλέψεις τόσο στο training όσο και στο test set. Τα αδύναμα μοντέλα μπορεί να παραβλέπουν σημαντικά χαρακτηριστικά και εμφανίζουν ανεπαρκή γενίκευση. Λογικά, η λύση του underfitting είναι η αύξηση της πολυπλοκότητας του μοντέλου προσθέτοντας περισσότερες παραμέτρους ή χρησιμοποιώντας πιο πολύπλοκες αρχιτεκτονικές.

Το overfitting, από την άλλη, συμβαίνει όταν ένα μοντέλο είναι υπερβολικά πολύπλοκο και προσαρμόζεται υπερβολικά καλά στα δεδομένα εκπαίδευσης. Αυτό σημαίνει ότι το μοντέλο όχι μόνο αποτυπώνει τα βασικά μοτίβα στα δεδομένα, αλλά μαθαίνει επίσης το θόρυβο και την τυχαιότητα που υπάρχουν στα δεδομένα εκπαίδευσης. Έτσι, ενώ εξάγει πολύ σωστές προβλέψεις για το training set, δεν πραγματοποιεί τις ίδιες προβλέψεις στο test set. Πολλοί λόγοι μπορούν να οδηγήσουν σε overfitting, με τους κυριότερους να είναι είτε ένα ιδιαίτερα πολύπλοκο μοντέλο ως προς τη φύση των δεδομένων, είτε ένα dataset με πολλά χαρακτηριστικά για μικρό πλήθος δεδομένων. Επιπρόσθετα, ένας ακόμα λόγος που οδηγεί σε overfitting είναι και το μεγάλο variance (διακύμανση). Το variance είναι στατιστικός όρος και στην ουσία αφορά το σφάλμα του μοντέλου εξαιτίας των μικρών διακυμάνσεων που παρουσιάζουν οι τιμές του training set. Ως αποτέλεσμα, ένα υπερβολικά προσαρμοσμένο μοντέλο θα έχει χαμηλό σφάλμα εκπαίδευσης αλλά υψηλό σφάλμα ελέγχου, δείχνοντας ότι αποδίδει καλά στα δεδομένα εκπαίδευσης, αλλά κακά στα νέα, αφού δεν μπορεί να γενικεύσει σωστά σε νέα, μη διακριτικά δεδομένα. Αναλόγως, λύση για το overfitting είναι η χρήση, ίσως, ενός πιο απλού μοντέλου ή η μείωση των διαστάσεων του dataset. Άλλη μία γνωστή λύση, είναι η κανονικοποίηση, η προσθήκη όρων στην αντικειμενική συνάρτηση του μοντέλου, για να βοηθήσει στον έλεγχο του overfitting.

Η χρήση ενός Cross-Validation set, δηλαδή ενός ακόμα test set επιτρέπει την αξιολόγηση της απόδοσης του μοντέλου σε διαφορετικά υποσύνολα των δεδομένων, βοηθώντας στον εντοπισμό της κατάλληλης πολυπλοκότητας, συνεισφέροντας στην αντιμετώπιση και των δύο περιπτώσεων fitting.



Εικόνα 17 Παράδειγμα underfitting, καλού fit και overfitting

3.6 Τεχνικές Εκμάθησης Συνόλου (Ensemble)

Οι τεχνικές εκμάθησης συνόλου (ensemble techniques) είναι μέθοδοι μηχανικής μάθησης, που συνδυάζουν τις προβλέψεις από πολλά βασικά μοντέλα για να παράγουν μια πιο ακριβή και ανθεκτική τελική πρόβλεψη. Αντί να βασίζονται σε ένα μόνο μοντέλο, αξιοποιούν τη συλλογική σοφία πολλών μοντέλων για να βελτιώσουν την προβλεπτική απόδοση.
Οι τεχνικές ensemble, συνήθως παράγουν καλύτερη ακρίβεια από τα μεμονωμένα βασικά μοντέλα. Με τον συνδυασμό πολλών μοντέλων, μπορούν να ανιχνεύσουν διάφορα πρότυπα και να μειώσουν τον κίνδυνο overfitting, με αποτέλεσμα πιο αξιόπιστες προβλέψεις.

Επιπλέον, είναι λιγότερο ευαίσθητες στον θόρυβο, αφού μπορούν να απομακρύνουν τα σφάλματα ή τις εσφαλμένες προβλέψεις που κάνουν τα μεμονωμένα μοντέλα. Ορισμένα προβλήματα, μπορεί να έχουν πολύπλοκες σχέσεις που δεν μπορούν να καταγραφούν επαρκώς από ένα μόνο μοντέλο. Οι τεχνικές ensemble μπορούν να συνδυάσουν διάφορες προσεγγίσεις μοντελοποίησης για να αντιμετωπίσουν αυτό το ζήτημα αποτελεσματικά. Ακόμα, παράγουν πιο σταθερά μοντέλα, μειώνοντας την εξάρτηση από συγκεκριμένες τυχαίες αρχικές συνθήκες και, τέλος, είναι γνωστές για τη βελτίωση της δυνατότητας γενίκευσης από τα δεδομένα εκπαίδευσης σε νέα δεδομένα, προσφέροντας ευρεία γκάμα πραγματικών εφαρμογών.

Υπάρχουν διάφορες τεχνικές ensemble, συμπεριλαμβανομένων των Bagging, Boosting και Stacking, κάθε μία με τα δικά της πλεονεκτήματα και πεδία εφαρμογής.

- Το **Bagging** αναπτύσσει πολλαπλά παραδείγματα του ίδιου βασικού μοντέλου σε διάφορα υποσύνολα των δεδομένων εκπαίδευσης, συνήθως με επαναλαμβανόμενη επιλογή (bootstrap samples), με την τελική πρόβλεψη να προκύπτει από τη συγχώνευση των προβλέψεων των μεμονωμένων βασικών μοντέλων, συχνά μέσω πλειοψηφίας-majority voting (για ταξινόμηση) ή μέσου όρου (για παλινδρόμηση). Αλγόριθμοι τύπου bagging, περιλαμβάνουν το Random Forest ή τα Bagged Δέντρα Απόφασης.

- Το **Boosting** δημιουργεί πολλαπλά βασικά μοντέλα σε ακολουθία, όπου κάθε επόμενο μοντέλο επικεντρώνεται στη διόρθωση των σφαλμάτων που έγιναν από τα προηγούμενα. Η τελική πρόβλεψη είναι συνήθως ένας αριθμητικός συνδυασμός των ξεχωριστών βασικών μοντέλων, όπου το βάρος κάθε μοντέλου εξαρτάται από την απόδοσή του. Συνήθεις αλγόριθμοι Boosting περιλαμβάνουν το AdaBoost, το Gradient Boosting και το XGBoost.

- Το **Stacking** συνδυάζει προβλέψεις από πολλά διαφορετικά βασικά μοντέλα χρησιμοποιώντας ένα meta-model, τελικό μοντέλο. Τα βασικά μοντέλα εκπαιδεύονται στα ίδια δεδομένα και οι προβλέψεις τους χρησιμοποιούνται ως είσοδοι για το meta-model. Το Stacking επιτρέπει τον συνδυασμό των δυνάμεων διαφορετικών αλγορίθμων και μπορεί να οδηγήσει σε ιδιαίτερα βελτιωμένη απόδοση.

3.7 Εκτίμηση Απόδοσης Μοντέλου

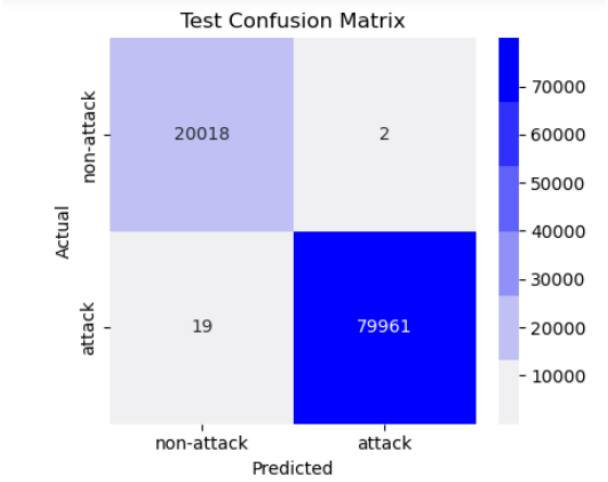
Μόλις ένας αλγόριθμος εκμάθησης δημιουργήσει το μοντέλο, ακολουθεί η αξιολόγηση του μοντέλου, με την χρήση ενός test set. Το test set περιέχει παραδείγματα που ο αλγόριθμος δεν έχει ξαναδεί, οπότε αν το μοντέλο αποδώσει καλά στις προβλέψεις του στο test set, θεωρούμε ότι το μοντέλο γενικεύει ικανοποιητικά. Για μεγαλύτερη ακρίβεια, υπάρχουν πολλά εργαλεία και

μετρικές για την αξιολόγηση αλγορίθμων μηχανικής μάθησης. Οι μετρικές που χρησιμοποιούνται πιο συχνά είναι:

- Ο πίνακας σύγχυσης (confusion matrix)
- Η ακρίβεια (accuracy)
- Η ακρίβεια/ανάκληση (precision/recall)
- Το F1-Score
- Η περιοχή κάτω από την καμπύλη ROC

Πίνακας Σύγχυσης (Confusion Matrix)

Ένας πίνακας που συνοψίζει την απόδοση ενός αλγορίθμου ταξινόμησης. Συνήθως χρησιμοποιείται σε προβλήματα δυαδικής ταξινόμησης (δύο κλάσεις: θετική και αρνητική) και παρέχει μια ανάλυση των προβλέψεων του μοντέλου[43]. Ο πίνακας αποτελείται από τέσσερις τιμές:



- ***Αληθείς Θετικοί (True Positives, TP):*** Ο αριθμός των σωστών θετικών προβλέψεων.
- ***Αληθείς Αρνητικοί (True Negatives, TN):*** Ο αριθμός των σωστών αρνητικών προβλέψεων.
- ***Ψευδείς Θετικοί (False Positives, FP):*** Ο αριθμός των εσφαλμένων θετικών προβλέψεων (Σφάλμα τύπου I).
- ***Ψευδείς Αρνητικοί (False Negatives, FN):*** Ο αριθμός των εσφαλμένων αρνητικών προβλέψεων (Σφάλμα τύπου II).

Εικόνα 18 Παράδειγμα Πίνακα Σύγχυσης

Ακρίβεια (Accuracy)

Το accuracy μετρά τη συνολική ορθότητα ενός μοντέλου ταξινόμησης. Είναι μια απλή και ευανάγνωστη μετρική που σας λέει το ποσοστό όλων των προβλέψεων που ήταν σωστές. Είναι ιδανική όταν υπάρχει ένα ισορροπημένο σύνολο δεδομένων με περίπου ίσο αριθμό θετικών και αρνητικών περιπτώσεων. Θεωρεί εξίσου τα ψευδή θετικά και τα ψευδή αρνητικά.
Υπολογίζεται ως: ***Accuracy = (TP + TN) / (TP + TN + FP + FN)***

Ακρίβεια και Ανάκληση (Precision και Recall)

Η ακρίβεια και η ανάκληση είναι δύο αλληλοσυμπληρούμενες μετρικές που χρησιμοποιούνται συχνά μαζί, ιδίως όταν ασχολούνται με μη ισορροπημένα σύνολα δεδομένων.
Η ακρίβεια επικεντρώνεται στην ελαχιστοποίηση των ψευδών θετικών και είναι ζωτικής σημασίας όταν το κόστος των ψευδών θετικών είναι υψηλό, και θέλουμε να βεβαιωθούμε, πως όταν το μοντέλο προβλέπει θετικό αποτέλεσμα, είναι πολύ πιθανό να είναι σωστό. Η ακρίβεια, μετρά το ποσοστό των σωστών θετικών προβλέψεων ανάμεσα σε όλες τις θετικές προβλέψεις.
Υπολογίζεται ως: ***Precision = TP / (TP + FP)***

Η ανάκληση επικεντρώνεται στην ελαχιστοποίηση των ψευδών αρνητικών και είναι απαραίτητη όταν το κόστος των ψευδών αρνητικών είναι υψηλό, και θέλετε να εξασφαλίσετε ότι αναγνωρίζονται όσο το δυνατόν περισσότερες πραγματικές θετικές περιπτώσεις. Μετρά το ποσοστό των σωστών θετικών προβλέψεων ανάμεσα σε όλα τα πραγματικά θετικά. Υπολογίζεται ως: ***Recall = TP / (TP + FN)***

F1-Score

Το F1-Score είναι ο αρμονικός μέσος της ακρίβειας και της ανάκλησης. Είναι μια μοναδική μετρική που ισορροπεί τόσο την ακρίβεια όσο και την ανάκληση. Είναι ιδιαίτερα χρήσιμο όταν χρειάζεται ένας συμβιβασμός μεταξύ της αποφυγής ψευδών θετικών και ψευδών αρνητικών, άρα μεταξύ ακρίβειας και ανάκλησης. Είναι ιδιαίτερα χρήσιμο σε καταστάσεις όπου υπάρχει ανισόρροπη κατανομή των κλάσεων ή όταν τόσο τα ψευδή θετικά όσο και τα ψευδή αρνητικά έχουν ίση σημασία.

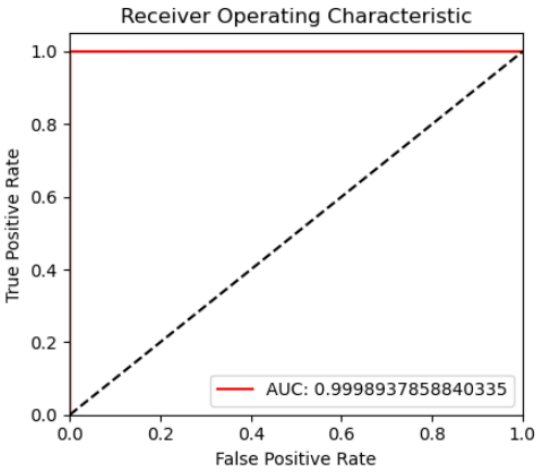
Ο τύπος του είναι: ***F1-Score = 2 * (Precision * Recall) / (Precision + Recall)***

Καμπύλη ROC και AUC

Η καμπύλη ROC (Receiver Operating Characteristic Curve) και AUC (Εμβαδόν κάτω από την καμπύλη ROC) είναι μια γραφική αναπαράσταση της απόδοσης ενός μοντέλου ταξινόμησης σε διάφορα κατώφλια (thresholds), για προβλήματα δυαδικής ταξινόμησης. Αναπαριστά τον Ρυθμό Αληθών Θετικών (Recall) έναντι του Ρυθμού Αληθών Αρνητικών (1 – Specificity) σε διάφορες ρυθμίσεις κατωφλιού. Ο Ρυθμός Ψευδών Θετικών ορίζεται ως:

1 – Specificity = FP / (FP + TN),

και προσδιορίζει πόσο καλά το μοντέλο αποφεύγει τις ψευδείς θετικές προβλέψεις.



Εικόνα 19 Παράδειγμα καμπύλης ROC και της τιμής AUC.

Ένα μεγάλο εμβαδόν κάτω από την καμπύλη ROC (AUC), υποδεικνύει καλύτερη απόδοση του μοντέλου και βοηθάει να κατανοήσουμε πόσο καλά το μοντέλο διακρίνει μεταξύ των θετικών και αρνητικών κλάσεων καθώς προσαρμόζεται το κατώφλι (threshold). Εν ολίγοις, στόχος της είναι η αξιολόγηση της δυνατότητας ενός μοντέλου να διακρίνει μεταξύ θετικών και αρνητικών κλάσεων σε διάφορα κατώφλια απόφασης και βοηθά στην σύγκριση της συνολικής απόδοσης διαφορετικών μοντέλων.

Κεφάλαιο 4

Υλοποίηση – Προσομοιώσεις – Συγκρίσεις

4.1 Εισαγωγή

Το ερευνητικό ενδιαφέρον της παρούσης διπλωματικής εργασίας επικεντρώθηκε στην σύγκριση των επιδόσεων γνωστών ML μοντέλων, με επίβλεψη και χωρίς, στην ανίχνευση DDoS. Πιο συγκεκριμένα, εφαρμόστηκαν μοντέλα ML, πάνω σε δύο από τα καλύτερα γνωστά dataset με κίνηση επιθέσεων DDoS. Τα δεδομένα προ επεξεργάστηκαν και οι επιδόσεις των μοντέλων, υπολογίστηκαν με την χρήση των μετρικών του confusion matrix, του F1-Score και της καμπύλης ROC (AUC).

4.2 Λογισμικό

Η παρούσα εργασία υλοποιήθηκε μέσω της γλώσσας προγραμματισμού Python και του Jupyter notebook. Η επιλογή της Python έγινε, για λόγους ευελιξίας της γλώσσας, αλλά και επειδή προσφέρει ένα πλούσιο οικοσύστημα βιβλιοθηκών και εργαλείων μηχανικής μάθησης, που έχουν βελτιστοποιηθεί για συσκευές της περιφέρειας. Αυτές οι βιβλιοθήκες επιτρέπουν την υλοποίηση μοντέλων μηχανικής μάθησης σε υλικό με περιορισμένους πόρους, κάτι το οποίο είναι κοντά στην μεριά του Edge Intelligence. Επίσης, οι δυνατότητες ανάλυσης δεδομένων της Python (π.χ., Pandas) είναι απαραίτητες για την εύκολη προ επεξεργασία και ανάλυση δεδομένων[44].

Το Jupyter notebook, επιλέχθηκε για την ευκολία στην δημιουργία πρωτοτύπων και πειραματισμών**,** καθώς επιτρέπει γρήγορες επαναλήψεις και προσαρμογές. Ακόμα, δίνει την δυνατότητα για ωραίες οπτικοποιήσεις για την παρακολούθηση και την απεικόνιση δεδομένων και αποτελεσμάτων , βοηθώντας στην κατανόηση και την αντιμετώπιση προβλημάτων[45].

Για τις προσομοιώσεις χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

- pandas και numpy για τη διαχείριση και επεξεργασία των dataset

- time για την χρονομέτρηση των ML μοντέλων
- scikit-learn για την εισαγωγή των μοντέλων και των μετρικών
- matplotlib και seaborn για την δημιουργία των γραφικών παραστάσεων
- prettytable για λόγους παρουσίασης αποτελεσμάτων

4.3 Κριτήρια Επιλογής

Το dataset UNSW (University of New South Wales)[46] είναι ευρέως γνωστό και αποτελεί σημείο κλειδί στον τομέα της κυβερνοασφάλειας. Προσφέρει μια πλήρη συλλογή από πραγματικά δεδομένα κυκλοφορίας στο δίκτυο, με ποικίλη γκάμα από σενάρια επιθέσεων. Με τις λεπτομερείς ετικέτες του και τον σημαντικό όγκο δεδομένων του, το σύνολο δεδομένων του UNSW έχει αποδειχθεί κρίσιμο για την ανάπτυξη και αξιολόγηση των συστημάτων ανίχνευσης διείσδυσης, καθιστώντας το θεμέλιο για την έρευνα στην κυβερνοασφάλεια.

Το dataset KDDCUP[47] αποτελεί έναν ακόμα θεμέλιο στον τομέα της ανίχνευσης επιθέσεων στο δίκτυο. Και πάλι είναι ένα μεγάλης κλίμακας σύνολο δεδομένων που περιλαμβάνει τόσο φυσιολογικά όσο και κακόβουλα πρότυπα κυκλοφορίας στο δίκτυο. Η χρήση του σε διαγωνισμούς όπως το KDDCUP, συνέβαλε στην ανάπτυξη διάφορων τεχνικών ανίχνευσης διείσδυσης και παραμένει ένα πρότυπο για την αξιολόγηση της αποτελεσματικότητας των μοντέλων ανίχνευσης επιθέσεων.

Τα δύο dataset επιλέχθηκαν για τα ρεαλιστικά δεδομένα που περιέχουν, αφού αποτελούνται από πραγματικά δεδομένα κίνησης δικτύου, αλλά και για την μεγάλη ποικιλία τους, σε συνδυασμό με αναλυτικές και ακριβείς ετικέτες, παράγοντας, έτσι, όσο το δυνατόν πιο κοντινά στην πραγματικότητα ML μοντέλα γίνεται.

Όσον αφορά την αξιολόγηση των μοντέλων και την όσο το δυνατόν ακριβέστερη εκτίμηση απόδοσης, ώστε η σύγκριση να είναι αξιοκρατική, χρησιμοποιήθηκαν για κάθε μοντέλο οι μετρικές του confusion matrix, του F1-Score και της καμπύλης ROC AUC, καθ’ ότι αυτές οι μετρικές εξυπηρετούν διάφορους σκοπούς στην αξιολόγηση ενός μοντέλου δυαδικής κατηγοριοποίησης:

- το confusion matrix, παρέχει μια καθαρή και συνοπτική περίληψη της απόδοσης του μοντέλου, διαχωρίζοντας τις προβλέψεις στις τέσσερις κατηγορίες και έτσι συμβάλει στην κατανόηση των τύπων σφαλμάτων που κάνει το μοντέλο, όπως ψευδή θετικά και ψευδή αρνητικά.
- το F1-Score, συνδυάζει και την μετρική του precision και του recall σε μία μετρική και είναι ιδιαίτερα χρήσιμο όταν η κατανομή των κλάσεων είναι μη ισορροπημένη.
- το ROC AUC βοηθά στην επιλογή ενός βέλτιστου κατωφλίου που ισορροπεί τον ρυθμό πραγματικών θετικών (ευαισθησία) και τον ρυθμό ψευδών θετικών (1 – Specificity) και πέρα απ’ την ανθεκτικότητα του στην ανισορροπία κλάσεων, επιτρέπει τη σύγκριση πολλαπλών μοντέλων με αξιολόγηση της συνολικής τους απόδοσης σε διάφορα κατώφλια πιθανοτήτων.

4.4 UNSW-15

4.4.1 Προ-επεξεργασία του dataset

Το dataset είναι διαθέσιμο στην ιστοσελίδα του UNSW, σε 4 αρχεία *UNSW-NB15_1.csv (169* MB*), UNSW-NB15_2.csv (165 MB), UNSW-NB15_1.csv (155 MB)* και *UNSW-NB15_1.csv (95 MB)* και ένα αρχείο *UNSW-NB15_features.csv (4.04 kB)*, με τα ονόματα των χαρακτηριστικών του dataset.
Το διάβασμα των αρχείων και το πέρασμα του σε dataframe, με την σωστή εισαγωγή των κεφαλίδων από το **features* στο υπόλοιπα δεδομένα, έγινε με την χρήση του παρακάτω κώδικα:

```
csv_files = ['UNSW-NB15_1.csv','UNSW-NB15_2.csv','UNSW-NB15_3.csv','UNSW-NB15_4.csv']
dfs=[]
for file in csv_files:
    df = pd.read_csv(file,header=None,low_memory=False)
    dfs.append(df)
unsw_dataset = pd.concat(dfs).reset_index(drop=True)
```

```
unsw_features = pd.read_csv('./NUSW-NB15_features.csv', encoding='ISO-8859-1')
unsw_features['Name'] = unsw_features['Name'].apply(lambda x: x.strip().replace('  ', ' ').lower())

unsw_dataset.columns = unsw_features['Name']
```

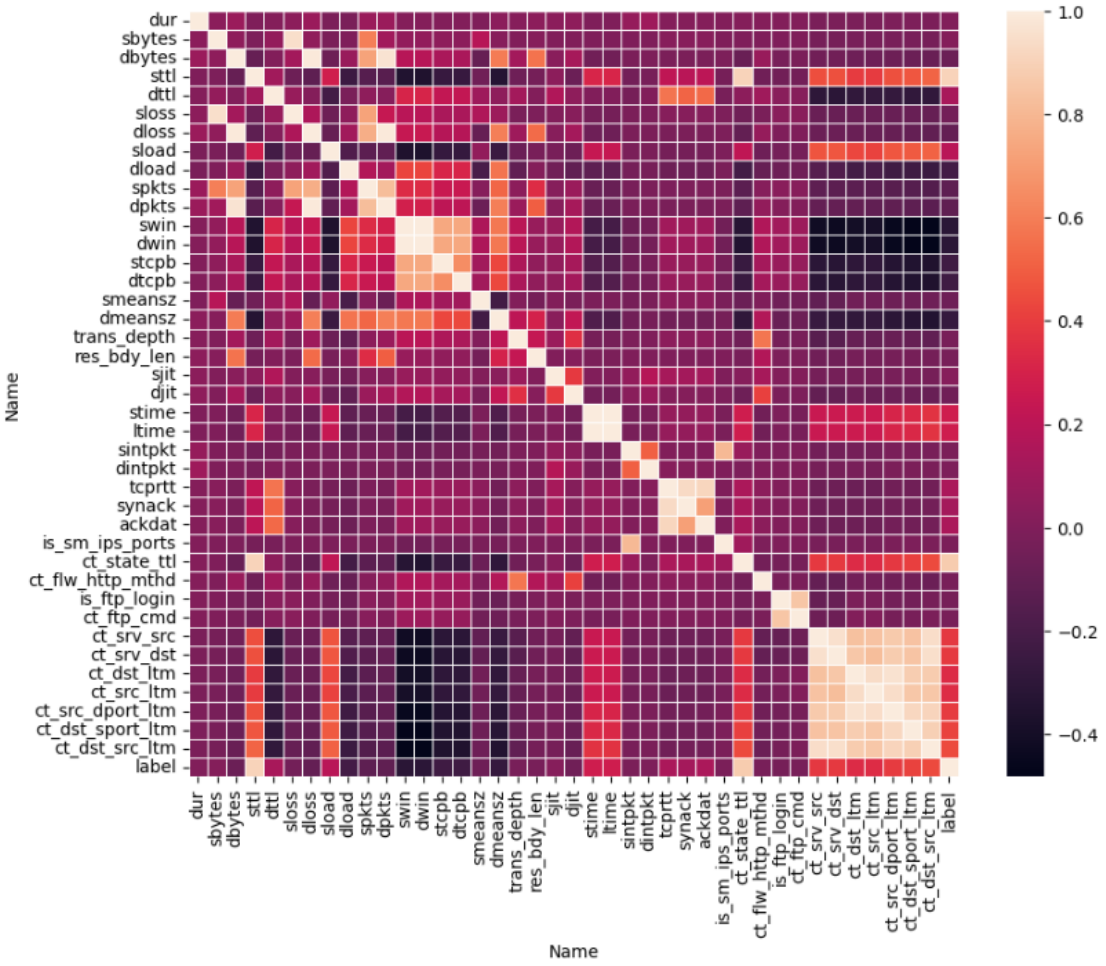
Εν συνεχεία, με την χρήση της εντολής *unsw_dataset.isna().sum(),* βρίσκουμε κατευθείαν πως τρία από τα συνολικά σαράντα εννέα attributes του dataset, περιέχουν τιμές NaN, τα οποία αντιμετωπίζονται με τρεις γραμμές κώδικα (η στήλη ‘attack_cat’, χρησιμοποιείται για multi-class προβλήματα):

```
ct_flw_http_mthd    1348145
is_ftp_login        1429879
ct_ftp_cmd                0
ct_srv_src                0
ct_srv_dst                0
ct_dst_ltm                0
ct_src_ltm                0
ct_src_dport_ltm          0
ct_dst_sport_ltm          0
ct_dst_src_ltm            0
attack_cat          2218764
```

```
#delete 'attack_cat'
unsw_dataset.drop('attack_cat', axis=1, inplace=True)

#replace NaN's with '0'
unsw_dataset['ct_flw_http_mthd'] = unsw_dataset.ct_flw_http_mthd.fillna(value=0)
unsw_dataset['is_ftp_login'] = (unsw_dataset.is_ftp_login.fillna(value=0)).astype(int)
```

Με την χρήση της εντολής .corr(), πάνω στο dataset και των γραφικών του seaborn, οπτικοποιούμε την συσχέτιση μεταξύ των διαφόρων attributes του dataset, με σκοπό τον εντοπισμό και, τελικά την αφαίρεση υψηλά συσχετιζόμενων features, τα οποία δυσχεραίνουν την εκπαίδευση των μοντέλων.



Αν και γραφικά, αποκτούμε μία ιδέα για το ποια features έχουν υψηλό συσχετισμό, αποτυπώνουμε τα ζευγάρια με τα υψηλότερα επίπεδα σχετικότητας, με την χρήση των εντολών:

```
# List of highly correlated feature pairs
correlated_pairs = []

# Iterate through the correlation matrix to find pairs of highly correlated features
for i in range(len(highly_correlated.columns)):
    for j in range(i):
        if highly_correlated.iloc[i, j]:
            correlated_pairs.append((highly_correlated.columns[i], highly_correlated.columns[j]))
```

```
Highly correlated feature pairs:
('sloss', 'sbytes')                ('ct_src_dport_ltm', 'ct_srv_src')
('dloss', 'dbytes')                ('ct_src_dport_ltm', 'ct_srv_dst')
('dpkts', 'dbytes')                ('ct_src_dport_ltm', 'ct_dst_ltm')
('dpkts', 'dloss')                 ('ct_src_dport_ltm', 'ct_src_ltm')
('dpkts', 'spkts')                 ('ct_dst_sport_ltm', 'ct_srv_src')
('dwin', 'swin')                   ('ct_dst_sport_ltm', 'ct_srv_dst')
('ltime', 'stime')                 ('ct_dst_sport_ltm', 'ct_dst_ltm')
('synack', 'tcprtt')               ('ct_dst_sport_ltm', 'ct_src_ltm')
('ackdat', 'tcprtt')               ('ct_dst_sport_ltm', 'ct_src_dport_ltm')
('is_sm_ips_ports', 'sintpkt')     ('ct_dst_src_ltm', 'ct_srv_src')
('ct_state_ttl', 'sttl')           ('ct_dst_src_ltm', 'ct_srv_dst')
('ct_ftp_cmd', 'is_ftp_login')     ('ct_dst_src_ltm', 'ct_dst_ltm')
('ct_srv_dst', 'ct_srv_src')       ('ct_dst_src_ltm', 'ct_src_ltm')
('ct_dst_ltm', 'ct_srv_src')       ('ct_dst_src_ltm', 'ct_src_dport_ltm')
('ct_dst_ltm', 'ct_srv_dst')       ('ct_dst_src_ltm', 'ct_dst_sport_ltm')
('ct_src_ltm', 'ct_srv_src')       ('label', 'sttl')
('ct_src_ltm', 'ct_srv_dst')       ('label', 'ct_state_ttl')
('ct_src_ltm', 'ct_dst_ltm')
```

Με μία απλή συνεπαγωγή μεταξύ των ζευγαριών, τα features που καταλήγουν να έχουν μεγάλη συσχέτιση είναι τα *['sloss', 'dloss', 'dpkts', 'dwin', 'ltime', 'ct_srv_dst', 'ct_src_dport_ltm', 'ct_dst_src_ltm', 'srcip', 'sport', 'dstip', 'dsport'],* τα οποία αφαιρούνται και, τέλος, τα *'dbytes'* και τα *'sbytes'* αντικαθίστανται σαν άθροισμα κίνησης από τα *'network_bytes'*.

Έτσι, καταλήγουμε στο dataset, με 34 features, εκ των οποίων τα 3 είναι τύπου object, τα οποία θα ονομάσουμε για λόγους ευκολίας **X** dataset, συν 1 για τις ετικέτες, το οποίο θα είναι το **y** dataset. Το αρχικό dataset είχε 48 features συν 1 για ετικέτες, εκ των οποίων τα 9 ήταν τύπου object.

4.4.2 Τελικά configurations και εκπαίδευση μοντέλων

Προκειμένου, το dataset, να μπορέσει να χρησιμοποιηθεί στην εκπαίδευση των μοντέλων, μένει τα τρία features τύπου object, 'proto', 'service' και 'state', να μετατραπούν και αυτά σε αριθμητικές τιμές. Αυτό γίνεται με την χρήση του OneHotEncoding, μίας τεχνικής για αναπαράσταση δεδομένων από κατηγορίες σε δυαδικά διανύσματα. Το dataset που δημιουργείται έχει πλέον 195 features, με μόνο αριθμητικούς τύπους και είναι έτοιμο να χρησιμοποιηθεί για εκπαίδευση.

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2540047 entries, 0 to 2540046
Columns: 195 entries, dur to state_no
dtypes: float64(195)
```

Το split του dataset, γίνεται σε 70% training data και 30% test data, ένα καλό split, για αρκετά δεδομένα εκπαίδευσης και εξίσου αρκετά δεδομένα για testing.

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

Προτού ξεκινήσει η εκπαίδευση του κάθε μοντέλου, εφαρμόζεται η συνάρτηση param_tuning, η οποία παίρνει ως ορίσματα το εκάστοτε μοντέλο, το εύρος των παραμέτρων που θα τεστάρει, ένα μέρος του dataset πάνω στο οποίο θα χρησιμοποιήσει τα simulations και τα metrics, για τον

υπολογισμό των καλύτερων δυνατόν παραμέτρων για το μοντέλο και μία μεταβλητή cv, η οποία ορίζεται ως default σε None, για την στρατηγική του cross validation. Η συνάρτηση ουσιαστικά χρησιμοποιείται για την αποθήκευση των αποτελεσμάτων, και εσωτερικά τρέχει το GridSearchCV, μιας τεχνικής για τον υπολογισμό καλύτερων παραμέτρων, τον οποίο επιτυγχάνει μέσω της αξιολόγησης του μοντέλου με διάφορους συνδυασμούς παραμέτρων.

```
def param_tuning(mlc, parameters, x, y, cv=None):

    scoring = {'auc':'roc_auc', 'f1':'f1'}

    tuning_clf = GridSearchCV( mlc, parameters, refit='auc',scoring=scoring, cv=cv, verbose=2,
    return_train_score=True)

    result = tuning_clf.fit(x, y)

    return result
```

Η συνάρτηση, εφαρμόζεται σε κάθε μοντέλο, σε ένα μικρό ποσοστό των training data, ώστε να παρθεί μία διαίσθηση για το ποιες παράμετροι είναι αυτές που θα βοηθήσουν στην καλύτερη επίδοση. Οι καλύτεροι υπερπαράμετροι, στην συνέχεια, χρησιμοποιούνται για την δημιουργία και fine tuning, προτού το μοντέλο εκπαιδευτεί στα training data. Έτσι, κανένα μοντέλο δεν εκπαιδεύεται με τυχαίες παραμέτρους, ώστε να συλλεχθούν τα καλύτερα αποτελέσματα σε κάθε περίπτωση και να γίνει η σύγκριση τους αξιοκρατικά.

Για την συγκέντρωση των αποτελεσμάτων από όλες τις προσομοιώσεις δημιουργούμε ένα dictionary, με πεδία για το όνομα του μοντέλου, τον τύπο του (supervised - unsupervised), τις μετρικές ROC/AUC και F1-Score και τον χρόνο εκπαίδευσης του κάθε μοντέλου.

Στην ίδια την εκπαίδευση χρησιμοποιούνται δύο υλοποιημένες και πάλι συναρτήσεις, οι evaluate_result_super και evaluate_result_unsuper.

- Μέσα τους και οι δύο υλοποιούν τις βασικές μεθόδους fit() και predict(), για την εκμάθηση και για την πρόβλεψη στα data. Κατά την κλήση τους, υλοποιούν δύο fit() – predict(), ένα για τα training data και ένα για τα test data. Στα training data, τα αποτελέσματα, δείχνουν το πόσο καλά το μοντέλο εκπαιδεύεται στα δεδομένα, σε τι βαθμό αναγνωρίζει patterns σε αυτά και για το αν το μοντέλο έχει κάνει overfitting ή underfitting, ενώ στα test data, το αποτέλεσμα δείχνει την πραγματική επίδοση του μοντέλου, σε νέα δεδομένα που δεν έχει ξαναδεί και αποτελεί την ουσιαστική μετρική για την αποτελεσματικότητα του εκάστοτε μοντέλου.
- Εν συνεχεία, υπολογίζουν τα metrics, κάνουν evaluate την απόδοση του μοντέλου και αποθηκεύουν τα αποτελέσματα του.
- Τέλος, με την χρήση του prettytable, του seaborn και του matplotlib, πλοτάρουν τα αποτελέσματα, το confusion matrix και την καμπύλη ROC/AUC.

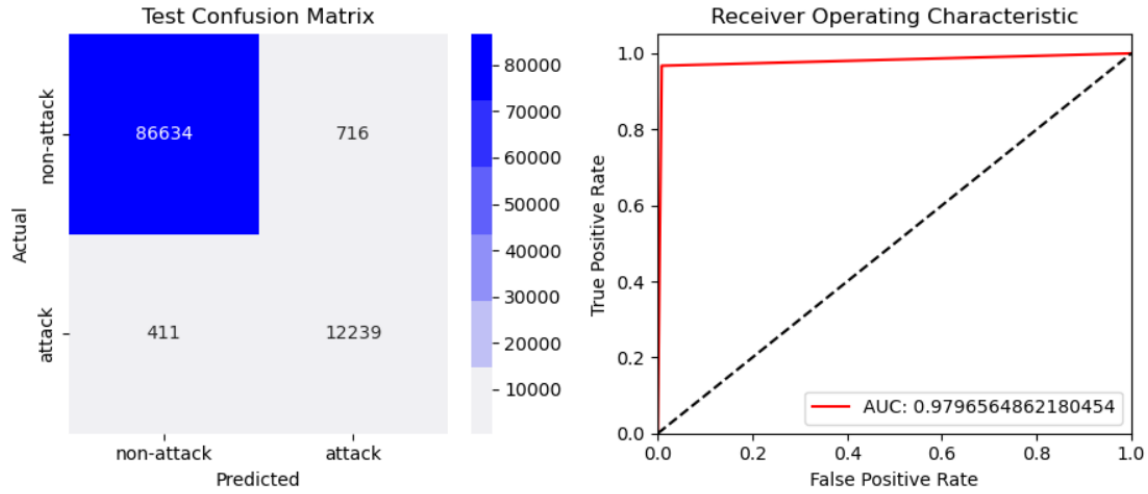
Η μοναδική διαφορά των δύο συναρτήσεων, είναι όσον αφορά το αν το μοντέλο είναι supervised ή unsupervised, με την πρώτη να εισάγει και την χρήση labels από το training set, ενώ η δεύτερη, εκπαιδεύει σε δεδομένα χωρίς ετικέτες.

4.4.3 Προσομοιώσεις Μοντέλων και Επιδόσεις

Logistic Regression (LR)

Το πρώτο μοντέλο που εκπαιδεύτηκε είναι αυτό του Logistic Regression, με μοναδική παράμετρο να παίζει ρόλο, το C=10, μία παράμετρος της οποίας η ισορροπία καθορίζει το πόσο καλά μπορεί να γενικεύσει το μοντέλο τη LR. Τα αποτελέσματα της εκπαίδευσης είναι:

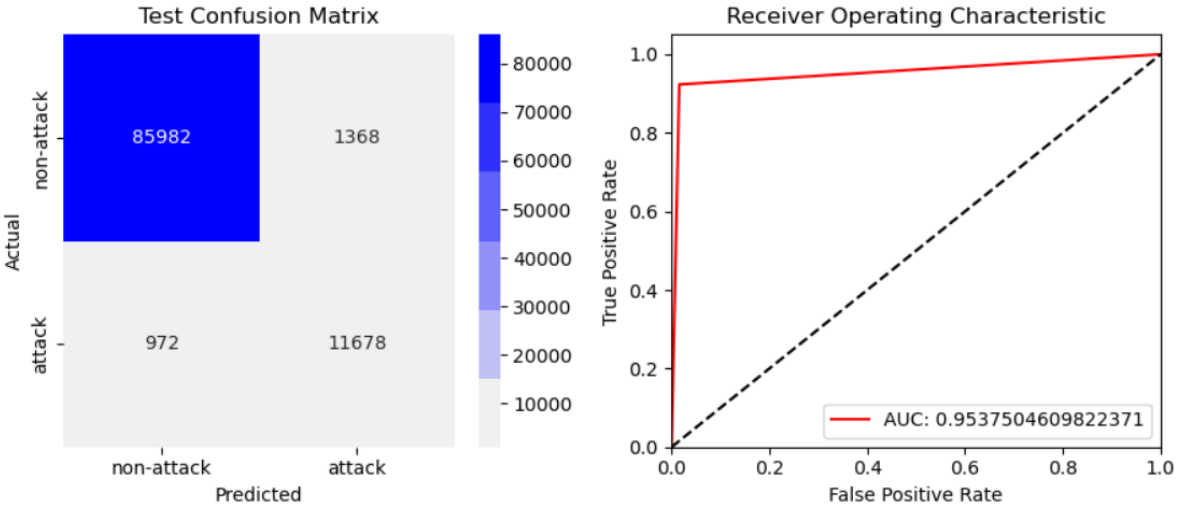
Dataset	Model	AUC	F1-score
Train	LR	0.9820111542602578	0.9588722794516302
Test	LR	0.9796564862180454	0.9559851591486037



Από το PrettyTable (πάνω αριστερά), βλέπουμε πως η LR απέδωσε πολύ καλά, και όσον αφορά το fit στα training data και όσον αφορά την γενίκευση στα test data. Ακόμα, το confusion matrix (κάτω αριστερά), δείχνει πως το ποσοστό των missed hits είναι πολύ μικρό, όπως και το αποτέλεσμα ROC/AUC (κάτω δεξιά) δείχνει ένα πολύ καλό classification. Ο λόγος που η LR πήγε τόσο καλά αποδίδεται στο ότι είναι πολύ καλό μοντέλο για binary classification προβλήματα, αλλά και είναι πολύ καλή στην γενίκευση σε νέα δεδομένα.

Support Vector Machine (SVM)

Dataset	Model	AUC	F1-score
Train	SVM	0.9531881046569157	0.9105230482694397
Test	SVM	0.9537504609822371	0.9089352428393525



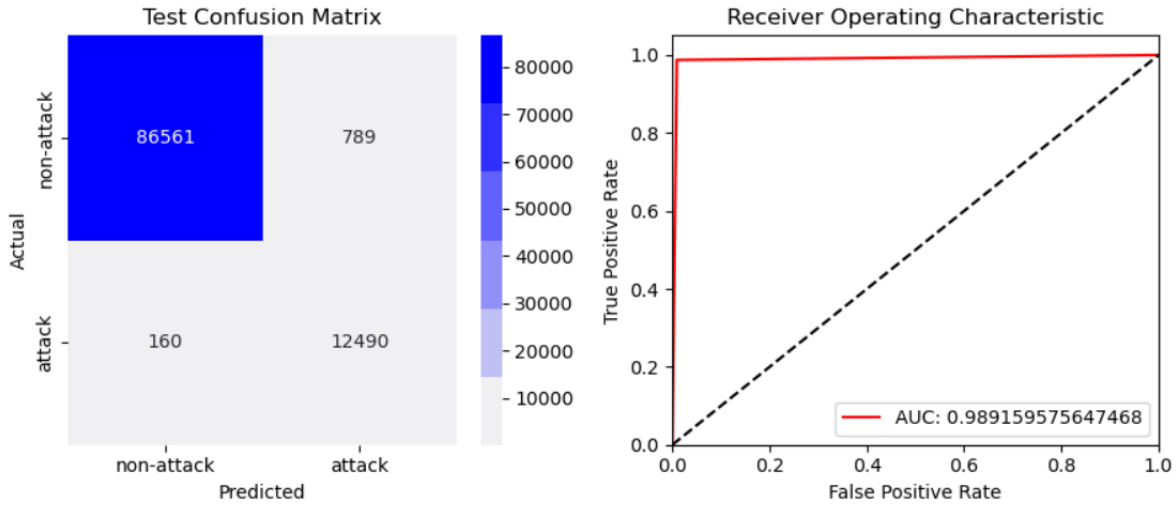
Για το μοντέλο SVC, χρησιμοποιήθηκε το SGDClassifier() του sklearn. Οι παράμετροι ορίστηκαν ως *loss='hinge'*, που έχει να κάνει με την λειτουργία σαν γραμμικός ταξινομητής. Το μοντέλο

πρακτικά, ψάχνει το καλύτερο hyperplane (υπερεπίπεδο), για να χωρίσει τις δύο κλάσεις. Ακόμα, το *alpha = 1*, με παρόμοια λειτουργία με το C του LR και το *penalty = l2*, που είναι το default, και προσθέτει έναν όρο κανονικοποίησης στην συνάρτηση κόστους, για την αποφυγή μεγάλων όρων στις μεταβλητές κατά την διάρκεια της εκπαίδευσης.
Με μία ματιά, το μοντέλο είχε πολύ καλές επιδόσεις σε όλους τους τομείς, ελαφρώς χειρότερα από το LR.

Radom Forest Classifier (RFC)

Η δημιουργία του μοντέλου έγινε ως εξής:
RandomForestClassifier(max_depth = 10, min_samples_split= 2, n_estimators= 100, n_jobs=-1)
Το *max_depth*, καθορίζει το μέγεθος των decision trees, με μεγαλύτερες τιμές να αυξάνουν την πολυπλοκότητα του μοντέλου, το *min_samples_split*, ορίζει ένα ελάχιστο threshold για το 'σπάσιμο' ενός εσωτερικό κόμβου σε παρακλάδια, ελέγχοντας έτσι την διαδικασία «δημιουργίας» του κάθε δέντρου, το *n_estimators*, ορίζει το μέγεθος του ensemble δίνοντας τον αριθμό των συνολικών δέντρων και τέλος το *n_jobs*, το οποίο ρυθμισμένο σε -1, κάνει την καλύτερη δυνατή παράλληλη επεξεργασία.

Dataset	Model	AUC	F1-score
Train	RFC	0.9918202391386405	0.9701068162606625
Test	RFC	0.989159575647468	0.9634000539935978



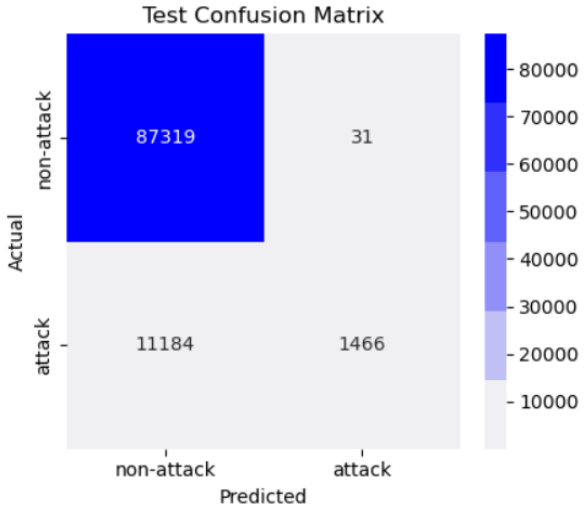
Εδώ, τα αποτελέσματα του μοντέλου συναντούν πολύ ψηλές επιδόσεις. Ο βασικός λόγος είναι ότι εκ φύσεως το RFC, αποτελεί ένα μοντέλο ensemble τεχνικής, συνυφασμένο με βελτιστοποίηση σε όλους τους τομείς σε σχέση με πιο απλά μοντέλα.

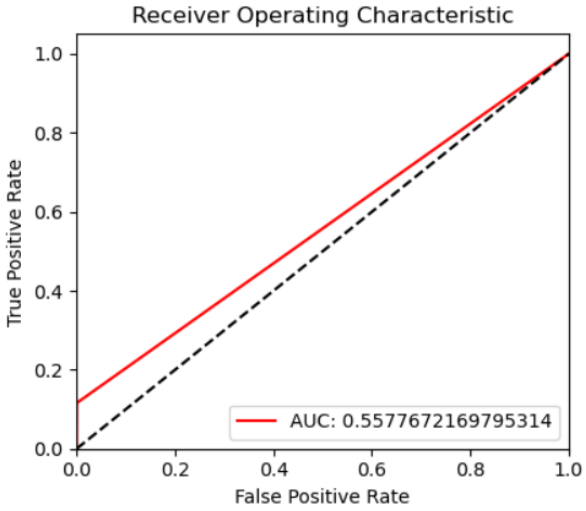
Naïve Bayes (N-B)

Η μοναδική παράμετρος είναι η priors, η οποία τέθηκε στην τιμή None, επιτρέποντας στο μοντέλο να βρει μόνο του τις πιθανότητες ανάθεσης σε κάθε κλάση.
Εδώ, τα αποτελέσματα έχουν ενδιαφέρον. Ενώ η μετρική του confusion matrix φαίνεται να υποδεικνύει καλές προβλέψεις, οι μετρικές του F1-Score και ROC/AUC, διαφωνούν. Η αντίφαση αυτή δείχνει ότι το μοντέλο δεν αποδίδει καλά και ο λόγος είναι ένας, η ανισορροπία κλάσεων. Το dataset έχει μικρό ποσοστό επιθέσεων σε σχέση με τον όγκο της κανονικής κυκλοφορίας, κάτι πολύ κοντά σε ένα real-world scenario. Αυτό το «παραφουσκωμένο» ποσοστό των True Negative, δηλώνει μεγάλο accuracy, με τον NB να κάνει σωστές προβλέψεις στην πλειοψηφία των κανονικών πακέτων. Ωστόσο, τα F1-Score δηλώνει πως στα ανώμαλα πακέτα οι προβλέψεις είναι

πολύ κακές, ενώ η κόκκινη γραμμή κοντά στην διχοτόμο, κάνει λόγο πιο πολύ για έναν «50-50» classifier (με λίγα λόγια, τυχαίο).

Dataset	Model	AUC	F1-score
Train	N-B	0.5639370247865877	0.2268213070352108
Test	N-B	0.5577672169795314	0.20725242100798758



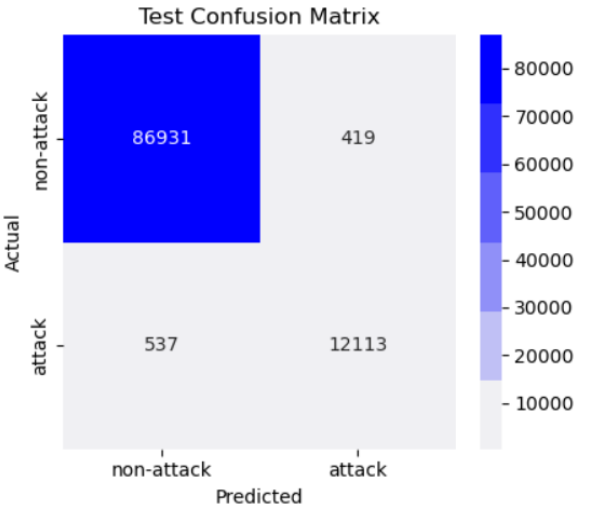


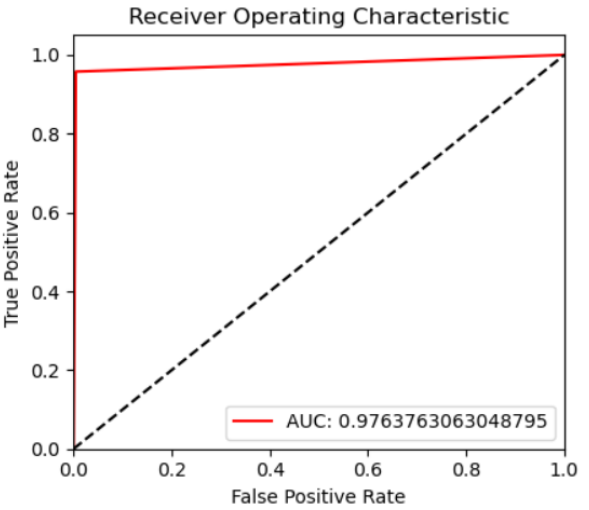
Decision Tree Classifier (DTC)

Κοντά στο RFC μοντέλο, η μοναδική διαφορετική παράμετρος του DTC, είναι η min_samples_leaf, η οποία ορίζει ένα ελάχιστο threshold για την δημιουργία ενός φύλλου, το οποίο περιέχει και την ετικέτα κλάσης. Σαν παράμετρος, ελέγχει την πολυπλοκότητα του δέντρου και μειώνει την τάση για overfitting.

Και πάλι, όπως και με το RFC, τα αποτελέσματα είναι πολύ καλά. Το μοντέλο έχει πολύ ψηλά score και στο train set , δείχνοντας την δυνατή του ικανότητα να κατανοήσει τα δεδομένα που διαβάζει σε μεγάλο βαθμό, αλλά και να γενικεύει σε νέα δεδομένα.

Dataset	Model	AUC	F1-score
Train	DTC	0.9918011154800157	0.987445393364556
Test	DTC	0.9763763063048795	0.9620363751886268

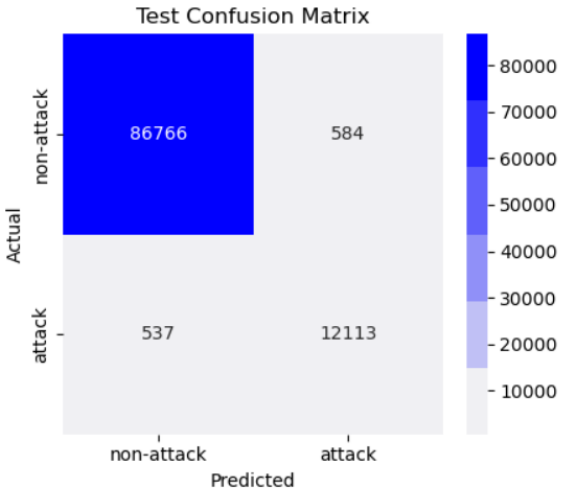


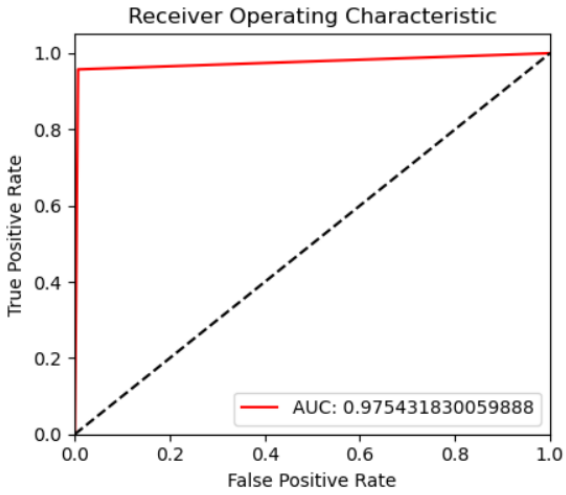


K Nearest Neighbors (KNN)

Το μοντέλο δημιουργήθηκε ως εξής: *KNeighborsClassifier(n_neighbors=11),* με την μεταβλητή *n_neighbors*, να είναι το βασικό component του KNN αλγορίθμου.

Dataset	Model	AUC	F1-score
Train	KNN	0.9811073954838966	0.9650743931088489
Test	KNN	0.975431830059888	0.9557738588393104





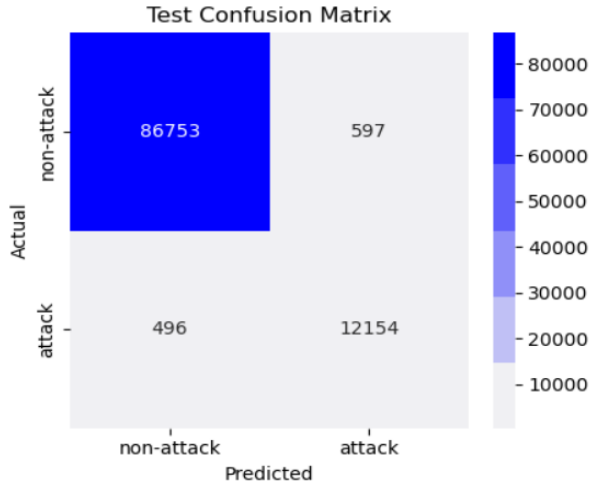
Neural Networks (NN)

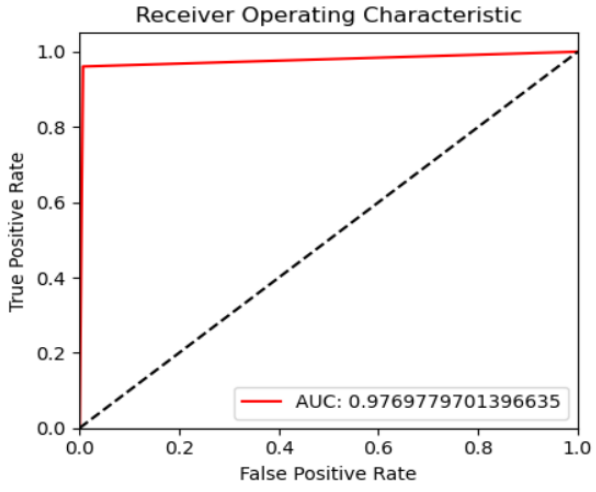
Το NN δημιουργήθηκε μέσω τη εντολής
MLPClassifier(hidden_layer_sizes=(128,64,32), activation='logistic', alpha= 0.01, batch_size= 64, max_iter= 100)

Οι μεταβλητές δηλώνουν,

- hidden_layer_sizes, τον αριθμό των νευρώνων σε κάθε επίπεδο του νευρωνικού
- activation, την συνάρτηση ενεργοποίησης σε κάθε νευρώνα, με logistic, περιορίζει το αποτέλεσμα μεταξύ {0,1}, καθιστώντας το ιδανικό για binary classification
- alpha, που είναι ένας πρόσθετος όρος ποινής στην συνάρτηση κόστους, μην επιτρέποντας τα ρυθμιζόμενα βάρη να ξεφύγουν σε τιμές

Dataset	Model	AUC	F1-score
Train	NN	0.9786891198800094	0.9594029617473527
Test	NN	0.9769779701396635	0.9569701980236998





- batch_size, που ορίζει κάθε πόσα δείγματα το μοντέλο θα ενημερώνει τα βάρη του.

Οι επιδόσεις του NN, είναι πολύ καλές, αλλά αξίζει να σημειωθεί ο σημαντικός χρόνος που χρειάστηκε για την εκπαίδευση του.

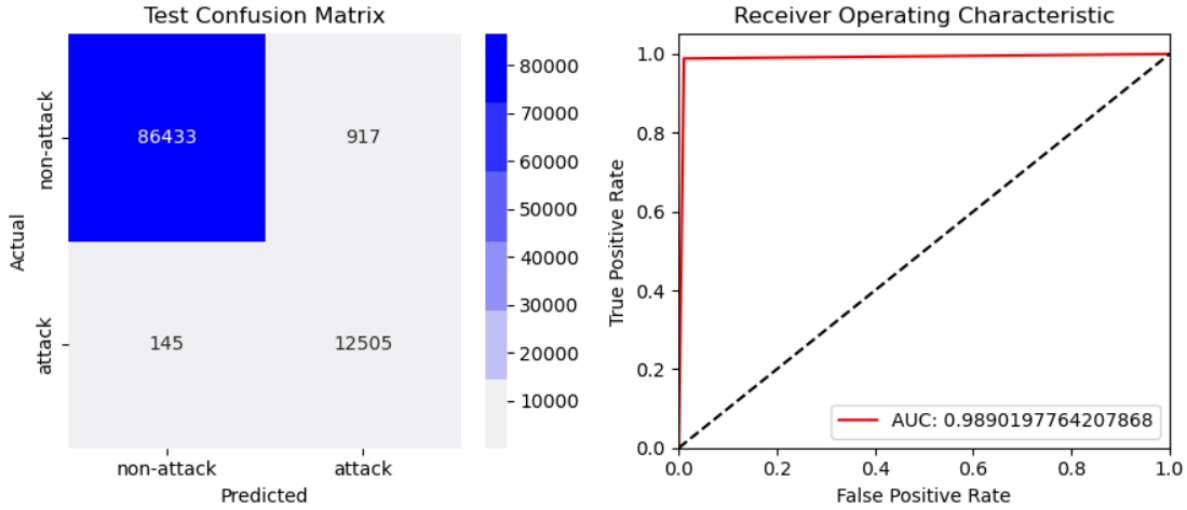
Gradient Boosting (GR-B)

Ο κώδικας για την υλοποίηση του μοντέλου με τις καλύτερες παραμέτρους, είναι:

xgb.XGBClassifier(objective='binary:logistic', random_state=101, n_estimators= 200, learning_rate= 0.01, max_depth= 4, subsample= 1, colsample_bytree= 0.8)

Η παράμετρος *objective* έχει οριστεί για πρόβλημα binary classification, η *random state*, σε κάθε επανάληψη στην μάθηση φροντίζει την ίδια αφετηρία, διατηρώντας τα ίδια αποτελέσματα, το *n_estimators*, καθορίζει το πόσα δέντρα θα κατασκευάσει το μοντέλο, το *learning_rate*, καθορίζει το πόσο κάθε δέντρο επηρεάζει την τελική πρόβλεψη, το *max_depth*, περιορίζει το βάθος του κάθε δέντρου, το *subsample*, καθορίζει το ποσοστό από τα training data για την ανάπτυξη του κάθε δέντρου(εδώ 1.0, άρα όλο το train) και τέλος, το *colsample_bytree*, που ορίζει το ποσοστό συμμετοχής των feature στην δημιουργία των δέντρων, εισάγοντας κάποια τυχαιότητα και μειώνοντας το overfitting.

Dataset	Model	AUC	F1-score
Train	GR-B	0.9901539410652529	0.9610725984611869
Test	GR-B	0.9890197764207868	0.9592666462104941

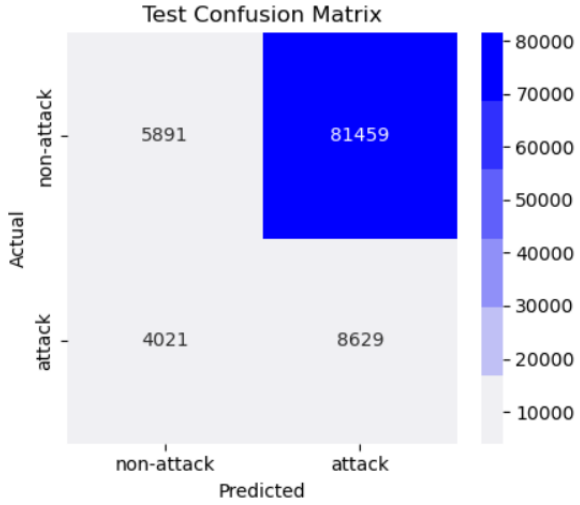


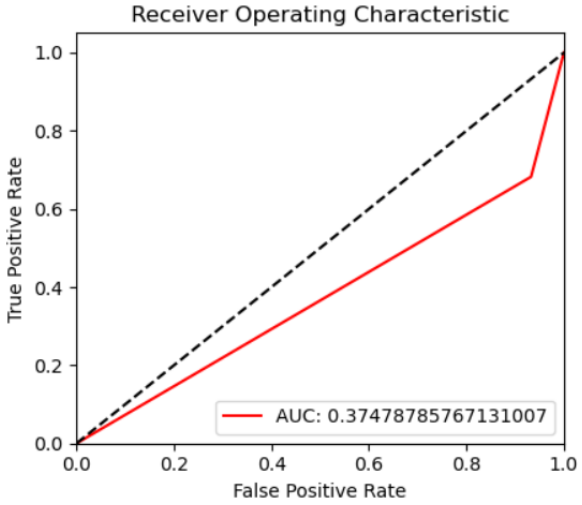
Και εδώ, το GR-B, είναι ένα μοντέλο ensemble τεχνικής, λίγο διαφορετικό από το RFC, ως προς την λειτουργία του, όπως έχει εξηγηθεί και εκτενέστερα σε προηγούμενο κεφάλαιο. Ωστόσο, τα αποτελέσματα είναι και εδώ, σημαντικά καλά τοποθετώντας το στα πιο ψηλά επίπεδα απόδοσης.

Isolation Forest (Iso_For)

Εδώ πλέον, κάνουμε λόγο για unsupervised μοντέλο. Συνεπώς, εκπαιδεύεται σε δεδομένα χωρίς ετικέτες. Οι παράμετροι που χρησιμοποιήθηκαν είναι οι contamination, που δίνει μια εκτίμηση για το ποσοστό ανωμαλιών στο dataset και n_estimators, που είναι το πλήθος των trees που χρησιμοποιεί το μοντέλο.

Dataset	Model	AUC	F1-score
Train	Iso_For	0.37269640729645076	0.16804399669051442
Test	Iso_For	0.37478785767131007	0.1679806887422375



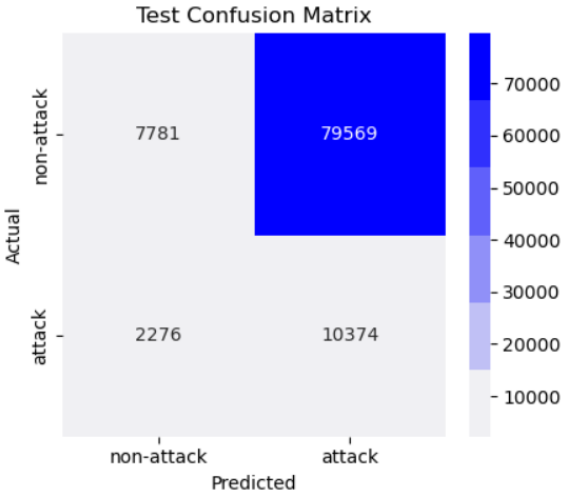


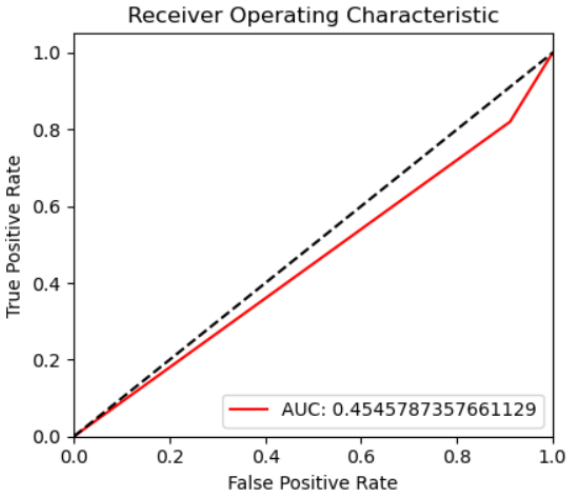
Όπως φαίνεται, το μοντέλο δεν τα πήγε καλά σε κανένα τομέα των metrics. Ο κυριότερος λόγος, είναι και πάλι η φύση του dataset, που είναι ιδιαίτερα unbalanced, με το Iso_For, να μην μπορεί να το διαχειριστεί. Άλλο ένα σημαντικό χαρακτηριστικό του μοντέλου, είναι ότι η κύρια ικανότητα του είναι η ανίχνευση ανωμαλιών σε ένα dataset και όχι η κατηγοριοποίηση τους, πράγμα που το καθιστά επίσης αρκετά αναξιόπιστο σε αυτό το binary classification πρόβλημα.

Local Outlier Factor (LOF)

Πάλι, στο ίδιο μοτίβο με το Iso_For, αυτό το μοντέλο δεν τα πηγαίνει καθόλου καλά. Οι υπερπαράμετροι για τον κώδικα του είναι ίδιοι, αλλά και πάλι τα αποτελέσματα δεν είναι ιδανικά. Για μία ακόμα φορά, η εξήγηση βασίζεται και πάλι στην φύση της λειτουργίας του μοντέλου, προσαρμοσμένο στην ανάθεση score ανωμαλίας σε data points ανάλογα με την τοπική πυκνότητα ή αλλιώς την εύρεση απομονωμένων instances μέσα σε ένα dataset, και όχι ο διαχωρισμός σε attack και non-attack, σε ένα ιδιαίτερα unbalanced dataset.

Dataset	Model	AUC	F1-score
Train	LOF	0.45658692386716504	0.20534785885315807
Test	LOF	0.4545787357661129	0.20223602000136462



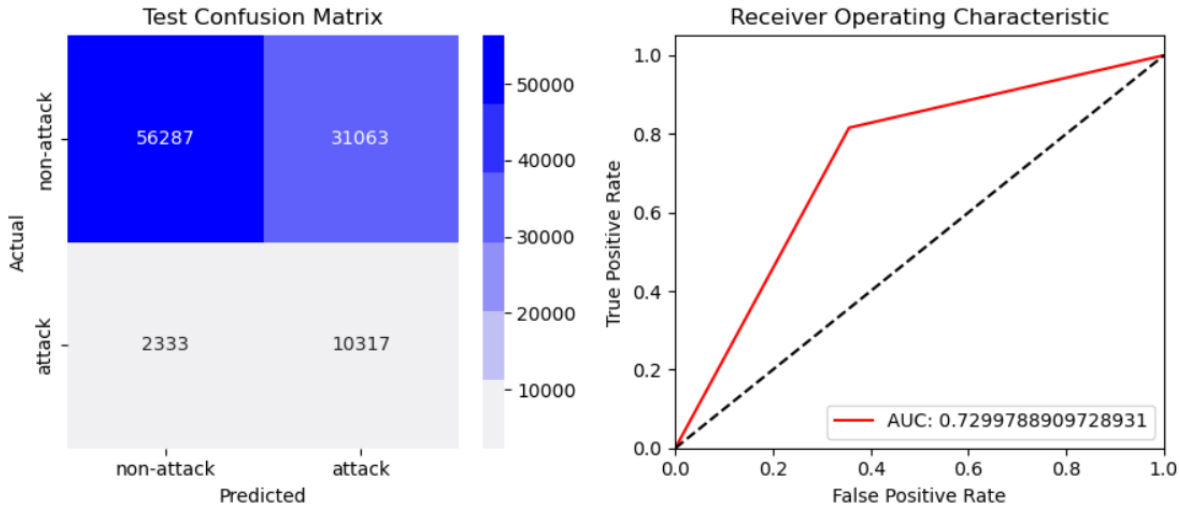


K- Means (KMEANS)

Κατά το δοκούν, και ο KMEANS, δεν αποδίδει σαν unsupervised. Οι λόγοι παραμένουν οι ίδιοι, με το μοντέλο να λειτουργεί για clustering, σε δεδομένα χωρίς ετικέτες, και παρ ότι η λειτουργία του μπορεί να πλησιάσει από την μάθηση ένα binary classification μοντέλο, εντούτοις δεν μπορεί να χρησιμοποιήσει την δύναμη που μπορεί να εκμεταλλευτεί ένας supervised αλγόριθμος με την χρήση ετικετών, πάνω στα ίδια δεδομένα.

Ένα ενδιαφέρον στα metrics, ωστόσο, του KMEANS, είναι η καμπύλη ROC/AUC, η οποία δίνει αρκετά καλό αποτέλεσμα. Και πάλι, ο συνδυασμός των metrics, μας επιτρέπει να δούμε πως το μοντέλο δεν αποδίδει καλά, και πιο συγκεκριμένα, η διαφορά στα metrics, οδηγεί και πάλι στην ανισορροπία του dataset.

Dataset	Model	AUC	F1-score
Train	KMEANS	0.732175194578896	0.38538954443890455
Test	KMEANS	0.7299788909728931	0.3818989450305386



4.4.4 Εφαρμογή dimensionality reduction με την χρήση SOM

Έχοντας δει πως αποδίδουν τα μοντέλα στο dataset, πλέον θα εισάγουμε την χρήση της τεχνικής SOM, για όσο το δυνατόν πιο καθαρό και αποδοτικό dimensionality reduction.

Η πρώτη εντολή:

som = MiniSom(x=10, y=10, input_len=X.shape[1], random_seed=101, learning_rate=0.5)

Εξηγώντας τις παραμέτρους, εξηγείται και η λειτουργία των SOM πάνω στο dataset. Αρχικά, οι παράμετροι *x* και *y* δημιουργούν το πλέγμα των SOM, σε διαστάσεις 10x10 στοιχείων. Η δεύτερη παράμετρος, *input_len*, παίρνει από το dataset το συνολικό πλήθος των features, με το SOM να χρειάζεται αυτή την τιμή, ώστε να μπορέσει να εισάγει κατάλληλα όλους τους κόμβους για την αποτύπωση του dataset. Το *random_seed* έχει να κάνει με την σταθερότητα των αποτελεσμάτων για διαφορετικά τρεξίματα, και τέλος το *learing_rate*, το οποίο ορίζει τον ρυθμό που τα βάρη των νευρώνων στο grid μεταβάλλονται κατά την εκπαίδευση, πρώτα γρήγορα και μετά πιο ομαλά.

Εν συνεχεία, ξεκινάει η εκπαίδευση του SOM:

som.train_random(X.values, 100000)

Το SOM, τρέχει για 100000 epochs ή επαναλήψεις, σε κάθε μία από τις οποίες λαμβάνει ένα τυχαίο δεδομένο και ενημερώνει τα βάρη (weights), των νευρώνων. Εδώ είναι το μέρος της διαδικασίας, στην οποία το mapping, θα καθορίσει τα features με την μεγαλύτερη επίδραση μέσα στο dataset. Αξίζει να σημειωθεί, πώς τα SOM, είναι unsupervised, οπότε το *X.values*, είναι αρκετό για την εκπαίδευση και δεν χρειάζεται η παροχή του y, με τις ετικέτες.

weights = som.get_weights()

Με τα βάρη των νευρώνων, να έχουν καθοριστεί πλέον από το dataset, με την εντολή *get_weights()*, παίρνουμε το βάρος του κάθε νευρώνα όπως αυτό έχει καθοριστεί μέσα στο grid, μετά την εκπαίδευση.

feature_importance = np.sum(weights, axis=(0, 1))
sorted_indices = np.argsort(feature_importance)[::-1]

Σε δύο εντολές παίρνουμε αυτό που θέλουμε, με το *feature_importance*, να περιέχει το άθροισμα των βαρών των νευρώνων που αντιστοιχούν σε κάθε feature (*axis=(0,1)*), και στην *sorted_indices*, να σορτάρονται ώστε να απορρίψουμε στην συνέχεια, τα features με τις μικρές εξαρτήσεις στο dataset. Αριστερά, βλέπουμε τα πρώτα στοιχεία του *feature_importance*, αυτά με την μεγαλύτερη βαρύτητα.

```
sjit ===> Importance 79.88972720302736
proto_tcp ===> Importance 72.12756084446269
state_FIN ===> Importance 68.49904656002387
service_None ===> Importance 57.9346679677669
sintpkt ===> Importance 51.74566700428816
dintpkt ===> Importance 49.27565938130978
djit ===> Importance 42.103106085804356
tcprtt ===> Importance 40.037702609652044
spkts ===> Importance 39.95178934341104
is_sm_ips_ports ===> Importance 38.969896573728
ackdat ===> Importance 38.59589847944606
dttl ===> Importance 36.81396270863704
synack ===> Importance 35.62048326337956
swin ===> Importance 26.96019825669865
network_bytes ===> Importance 26.574463499884317
```

Με ένα απλό τρέξιμο στο *sorted_indices*, παίρνουμε τα features που έχουν την μεγαλύτερη «επίδραση», και κόβουμε αυτά των οποίων η σημασία είναι μικρότερη από ένα μικρό ορισμένο threshold.

Το επεξεργασμένο με χρήση SOM dataset, αποτελείται πλέον από τον ίδιο αριθμό data (σειρών), όμως με μόλις 39 features, έναντι των 195. Όλες οι παραπάνω εντολές, εκτελέστηκαν αρκετά, γρήγορα, με ακόμα και το training , να ολοκληρώνεται σε μερικά δευτερόλεπτα, εξασφαλίζοντας έτσι ότι δεν υπάρχει αύξηση της χρονικής πολυπλοκότητας στον κώδικα.

```
X_som.shape
```

```
(2540047, 39)
```

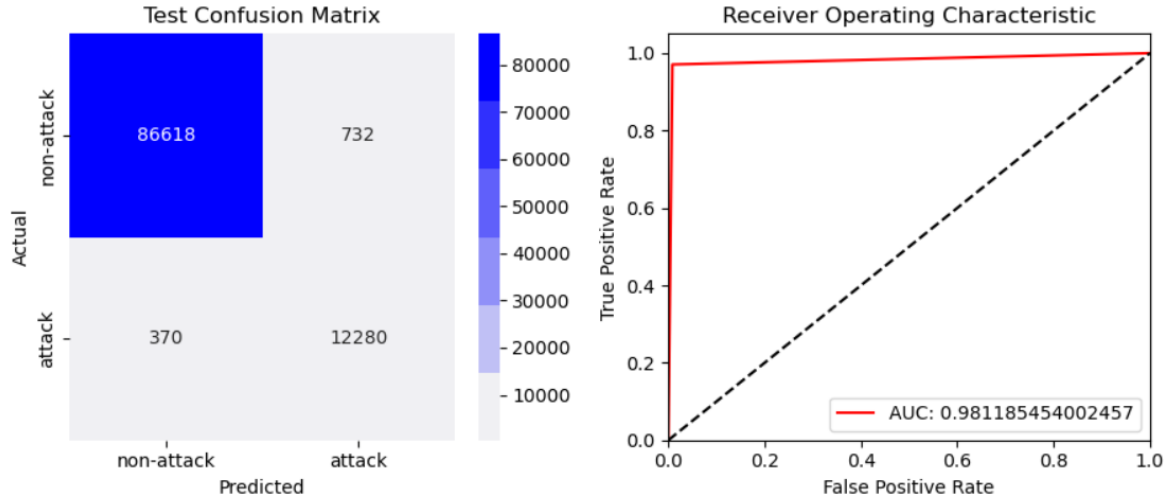
Μένει μόνο να δούμε εάν, πέρα προφανώς από την χρονική και υπολογιστική βελτίωση, αν τα αποτελέσματα στο τρέξιμο των μοντέλων, είναι το ίδιο αποδοτικά. Όπως και προηγουμένως, τα μοντέλα πρώτα μπαίνουν στην συνάρτηση *param_tuning()*, για την εύρεση των καλύτερων δυνατόν παραμέτρων. Εδώ, πλέον κοιτάμε την επίδοση των μοντέλων, να είναι όσο το δυνατόν πιο κοντά στην επίδοση χωρίς την χρήση του dimensionality reduction, το οποίο προφανώς θα έχει λιγότερα δεδομένα για το μοντέλο, σε σχέση με την χρήση όλων των features του dataset.

4.4.5 Προσομοιώσεις και Επιδόσεις με SOM

Logistic Regression (LR)

Το πρώτο μοντέλο, είναι το Logistic Regression, με την παράμετρο C=10, να ορίζεται πάλι στην ίδια τιμή

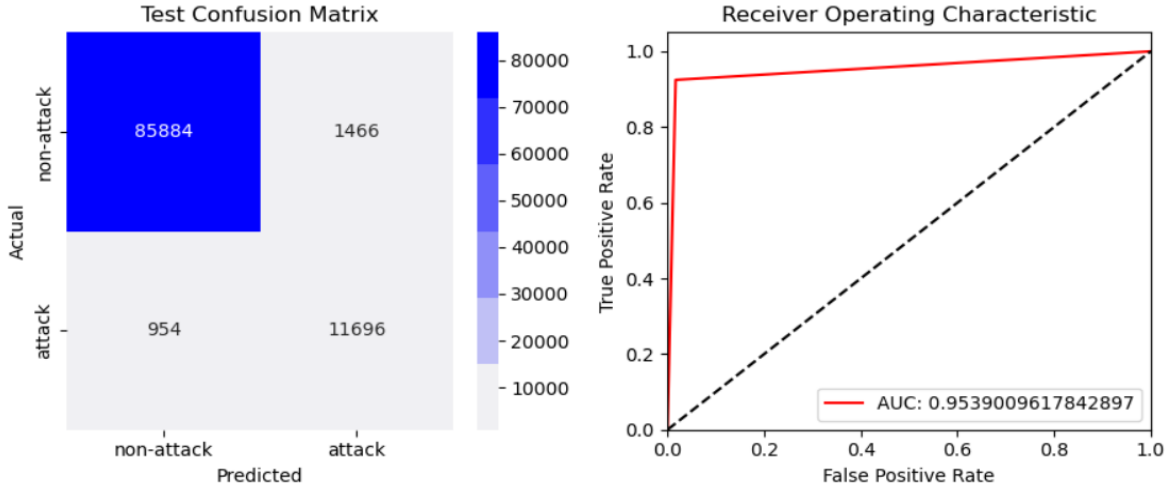
Dataset	Model	AUC	F1-score
Train	LR	0.9834882628629551	0.9594500656522746
Test	LR	0.981185454002457	0.9570571272698933



Ο χρόνος εκπαίδευσης θα παρουσιαστεί αργότερα, ωστόσο όχι μόνο παρατηρείται καλή επίδοση από το μοντέλο, αλλά ακόμα και μία ελαφρά αύξηση. Η απλή εξήγηση, βασίζεται στην λειτουργία του dimensionality reduction, με στοιχεία τα οποία έχουν υψηλή συσχέτιση με άλλα, να συγκεντρώνουν λιγότερη βαρύτητα και συνεπώς να θεωρούνται μη χρήσιμα. Η απλοποίηση του dataset, βοηθά επίσης το απλό σχετικά μοντέλο του LR, να αποδώσει καλύτερα αποφεύγοντας και το overfitting, που πιθανώς συνέβαινε σε μεγαλύτερο βαθμό με το προηγούμενο high dimensional dataset.

Support Vector Machine (SVM)

Dataset	Model	AUC	F1-score
Train	SVM	0.9535589227682015	0.9070368228262965
Test	SVM	0.9539009617842897	0.9062451572911824

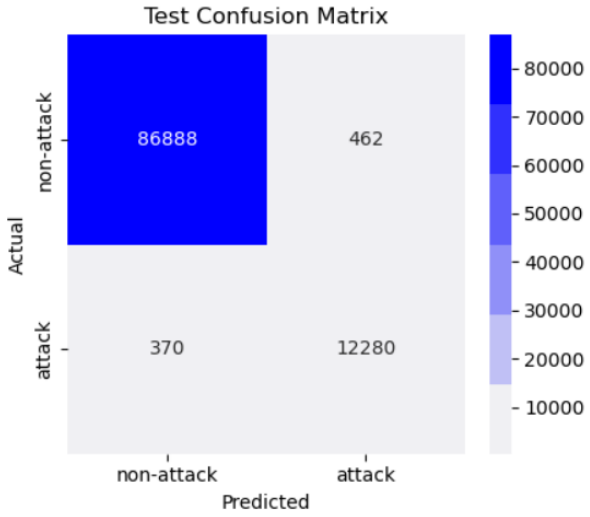


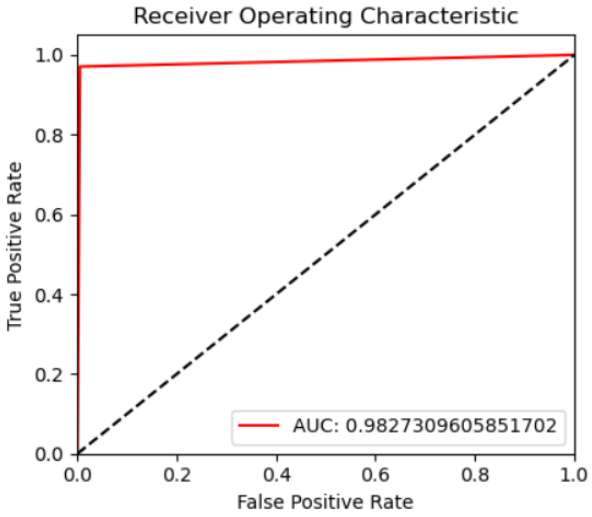
Για το μοντέλο SVC, η απόδοση μειώθηκε ελαφρώς. Για την ακρίβεια, στο F1-Score, έπεσε κατά 0.002, το οποίο σε σχέση με το AUC, που αυξήθηκε κατά 0.002, μας λέει ότι η απόδοση του μοντέλου δεν μεταβλήθηκε.

Radom Forest Classifier (RFC)

Όπως και στο SVM, οι επιδόσεις και εδώ παρέμειναν στο ίδιο πολύ ικανοποιητικό επίπεδο.

Dataset	Model	AUC	F1-score
Train	RFC	0.9884125610920582	0.9792075323656335
Test	RFC	0.9827309605851702	0.9672337744171392

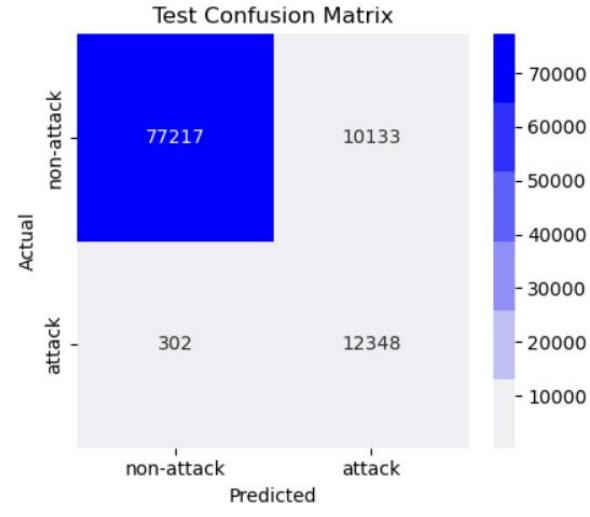


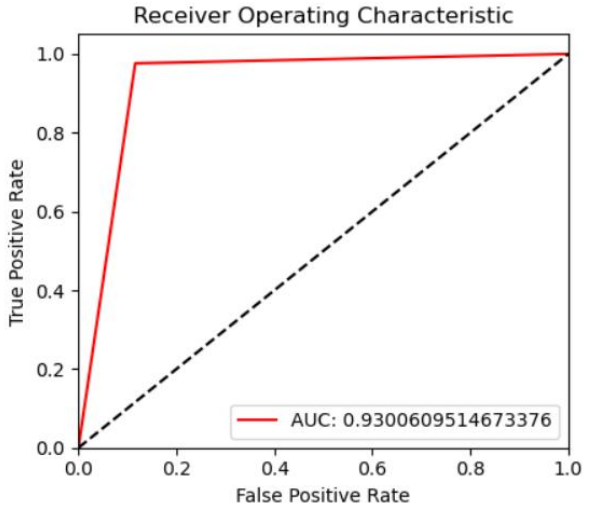


Naïve Bayes (N-B)

Η αύξηση στις επιδόσεις του N-B, είναι σημαντική. Παρ, ότι παραμένει ακόμα ένα από τα μοντέλα με τις χειρότερες επιδόσεις, σε σχέση με τα αποτελέσματα του με το προηγούμενο dataset, υπάρχει έντονη βελτίωση. Ο λόγος, είναι κυρίως και πάλι στην μείωση των πολλών συσχετισμένων features, το οποίο πηγαίνει κόντρα στην βασική υπόθεση του μοντέλου, πως όλα τα features είναι ανεξάρτητα. Εδώ, έχοντας διώξει τα περισσότερα συσχετισμένα μεταξύ τους «άχρηστα» features, ο N-B μπορεί πλέον να κάνει ακριβέστερες προβλέψεις.

Dataset	Model	AUC	F1-score
Train	N-B	0.9295037380217589	0.7021853294934778
Test	N-B	0.9300609514673376	0.7029688878768039



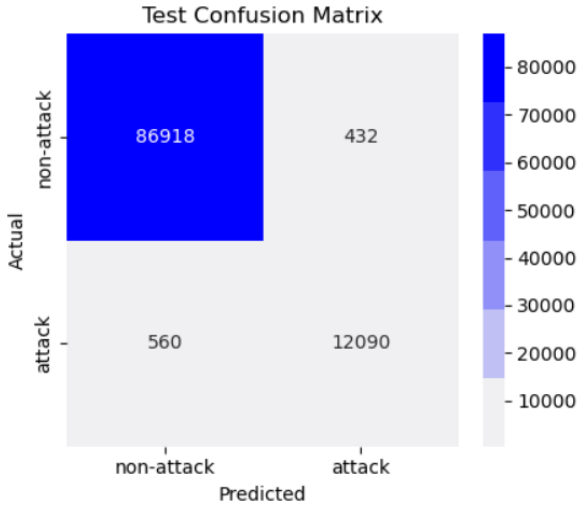


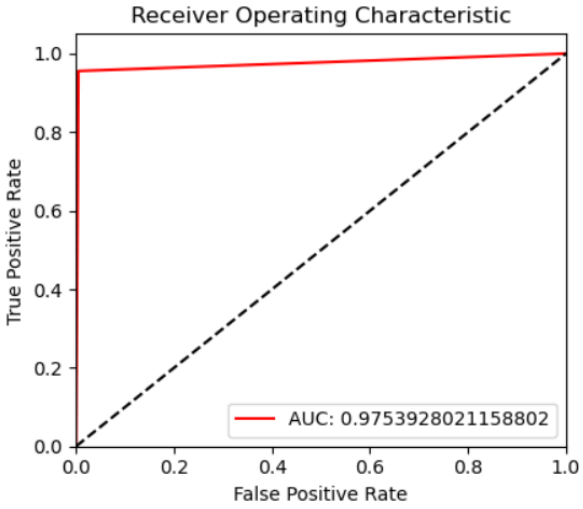
Ένας ακόμη λόγος, είναι και η μείωση της αραιότητας των δεδομένων, η οποία όταν υπάρχει σε μεγάλο βαθμό, δυσκολεύει το μοντέλο να κάνει προβλέψεις στις πιθανότητες.

Decision Tree Classifier (DTC)

Αμελητέα μείωση στις επιδόσεις, της ίδιας τάξης με το RFC.

Dataset	Model	AUC	F1-score
Train	DTC	0.9919328979815913	0.988340029937761
Test	DTC	0.9753928021158802	0.9605911330049262

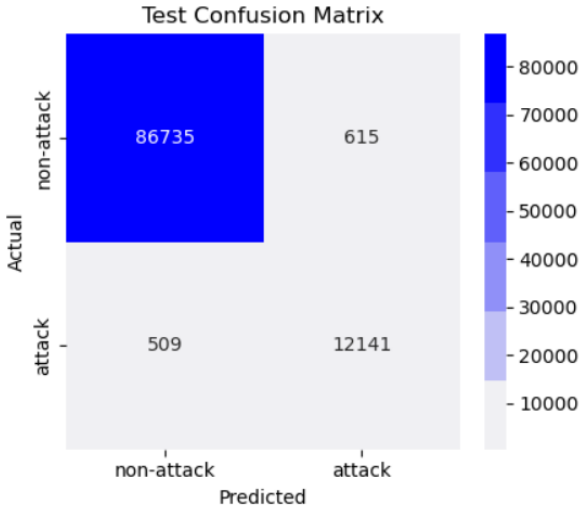


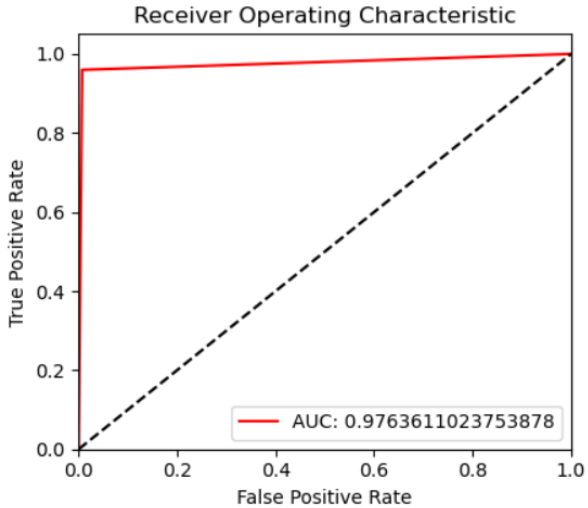


K Nearest Neighbors (KNN)

Ελαφρά αύξηση στο AUC, καμία σχεδόν αλλαγή στο F1-Score.

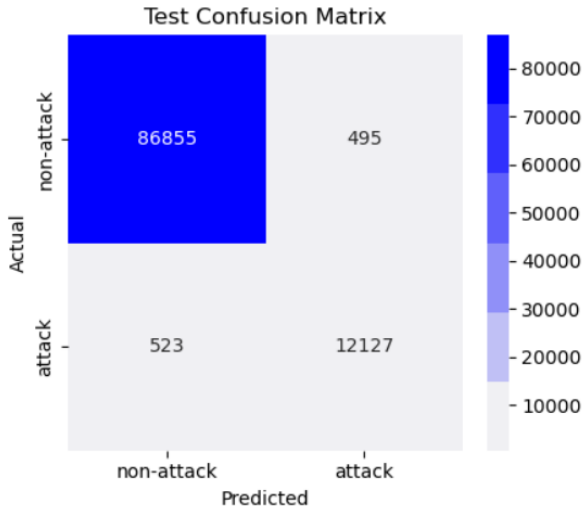
Dataset	Model	AUC	F1-score
Train	KNN	0.9812298737327877	0.9634836945909003
Test	KNN	0.9763611023753878	0.9557584822482877

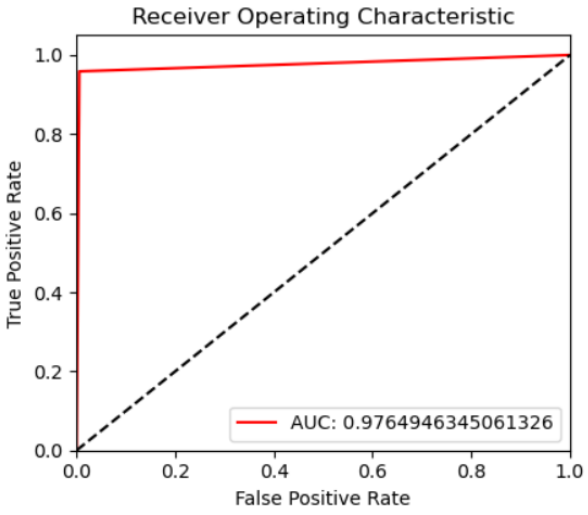




Neural Networks (NN)

Dataset	Model	AUC	F1-score
Train	NN	0.9783787906590613	0.9630153676846285
Test	NN	0.9764946345061326	0.9597182652738209

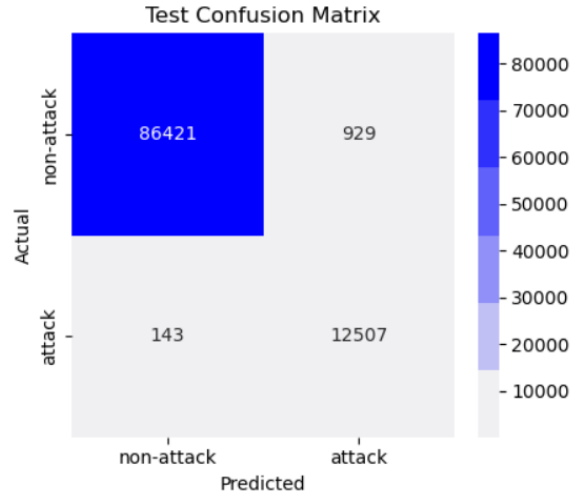


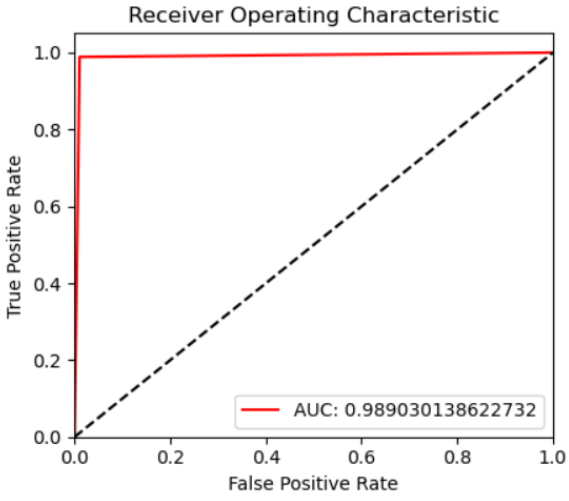


Στην προηγούμενη εκπαίδευση είχαν σημειωθεί οι καλές επιδόσεις του, αλλά με πολύ σημαντικό χρόνο εκτέλεσης. Εδώ τα αποτελέσματα από άποψη metrics, είναι ίδια αλλά όπως θα δειχθεί αργότερα η διαφορά στην χρονική πολυπλοκότητά είναι τεράστια.

Gradient Boosting (GR-B)

Dataset	Model	AUC	F1-score
Train	GR-B	0.9900582282898728	0.9608836412111978
Test	GR-B	0.989030138622732	0.9589051598558614



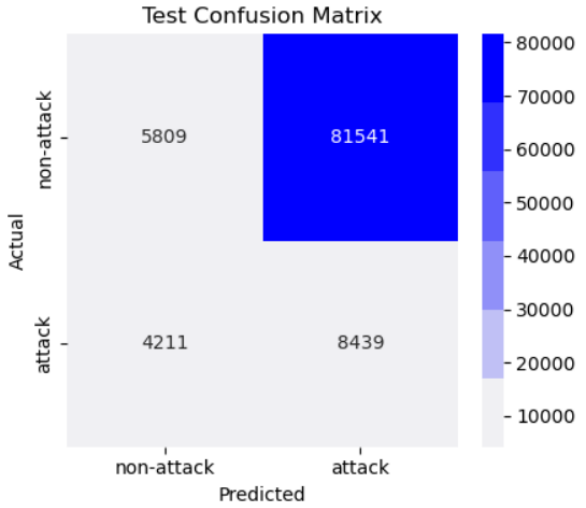


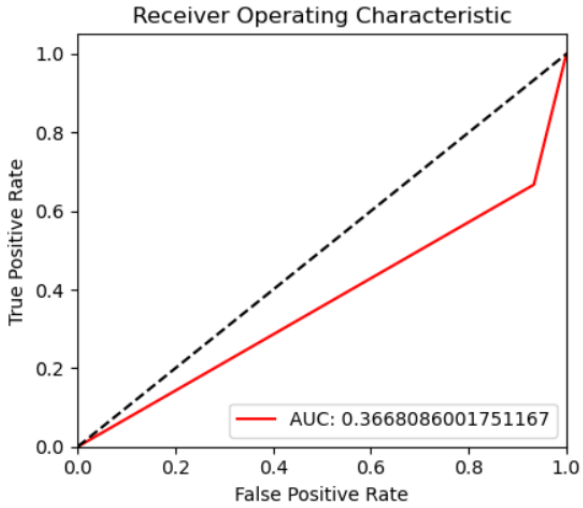
Πρακτικά μηδενική αλλαγή στην δυνατή προγνωστική ικανότητα του μοντέλου.

Isolation Forest (Iso_For)

Μπαίνοντας στο κομμάτι των unsupervised μοντέλων, πάλι παρατηρούμε ότι η απόδοση παραμένει στα ίδια χαμηλά επίπεδα. Η εξήγηση της μη αύξησης της απόδοσης, έχει να κάνει με την εξαιρετική ικανότητα των SOM, στη διατήρηση των τοπολογιών χαρακτηριστικών των features. Συνεπώς, για τους ίδιους με προηγουμένως λόγους, που το Iso_For, δεν μπορούσε να αποδώσει καλά, δεν τα καταφέρνει και εδώ.

Dataset	Model	AUC	F1-score
Train	Iso_For	0.3678315901469494	0.16593988475315372
Test	Iso_For	0.3668086001751167	0.16445483776673486

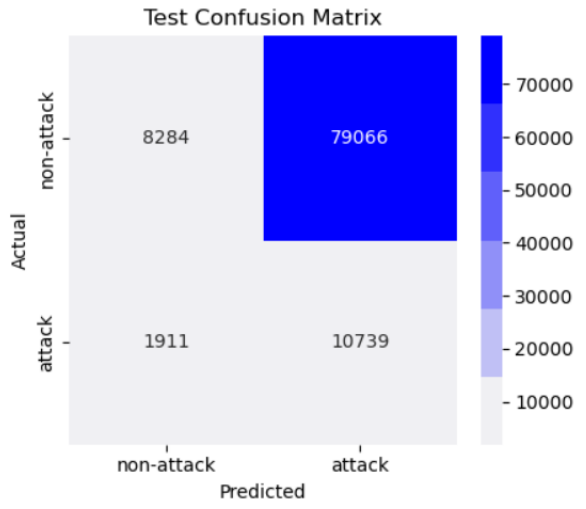


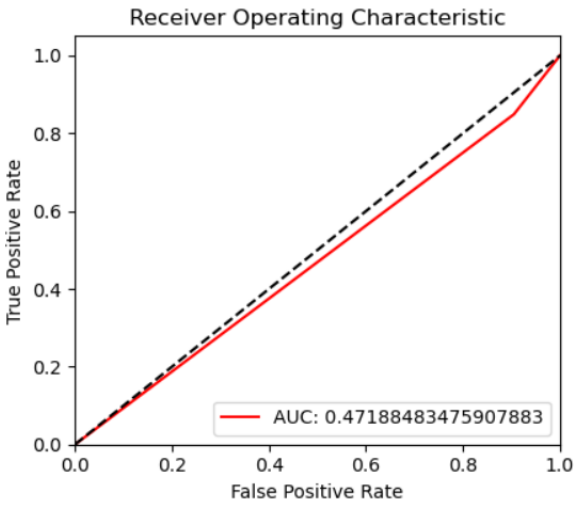


Local Outlier Factor (LOF)

Καμία αλλαγή στις επιδόσεις.

Dataset	Model	AUC	F1-score
Train	LOF	0.475982974837283	0.2135972313588651
Test	LOF	0.47188483475907883	0.2096334976331072

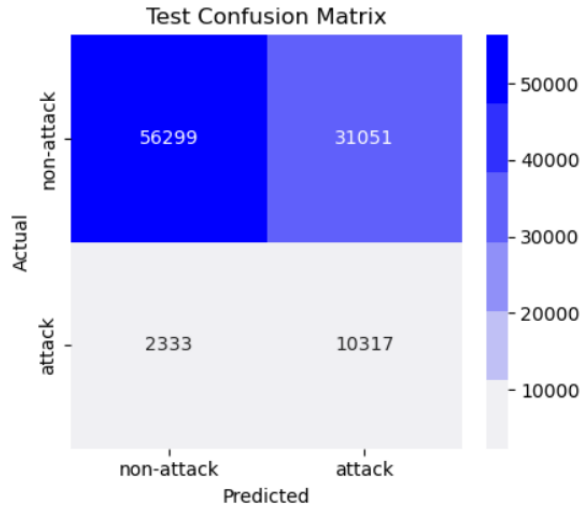


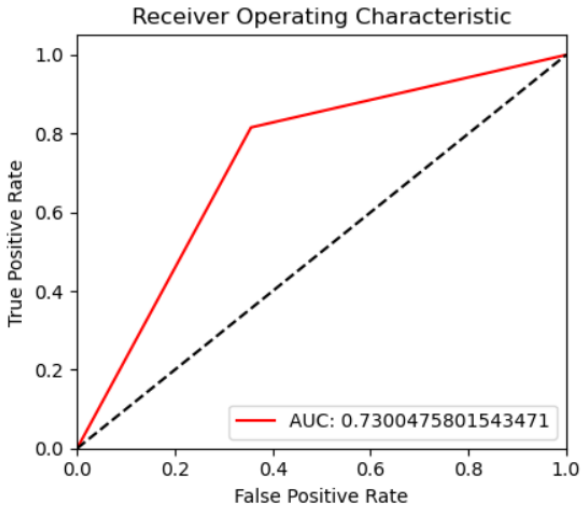


K- Means (KMEANS)

Ελαφρά βελτίωση.

Dataset	Model	AUC	F1-score
Train	KMEANS	0.732204688789957	0.385445142223864
Test	KMEANS	0.7300475801543471	0.38198378318338333





4.4.6 Αποτελέσματα και παρατηρήσεις

Σε κάθε προσομοίωση μοντέλου, προσθέσαμε ένα set εντολών για την χρονομέτρηση της διαδικασίας εκπαίδευσης των μοντέλων. Το ζευγάρι *start_time = time.time(),* πριν την εκπαίδευση και *end_time = time.time(),* μετά, με μία απλή αφαίρεση μας έδινε τον χρόνο που χρειαζόταν κάθε μοντέλο για την εκπαίδευση του. Όλα τα αποτελέσματα, από τις προσομοιώσεις και συγκεκριμένα οι μετρικές AUC, F1-Score και οι χρόνοι εκπαίδευσης, αποθηκεύτηκαν σε ένα dataframe, το οποίο παρουσιάζεται παρακάτω.

	Names	No SOM				With SOM			
		Type	AUC	F1	Time	Type	AUC	F1	Time
1	LR	supervised	0.979656	0.955985	4.49	supervised	0.981185	0.957057	2.07
2	SVM	supervised	0.953750	0.908935	1.41	supervised	0.953901	0.906245	1.07
3	RFC	supervised	0.989160	0.963400	7.16	supervised	0.982731	0.967234	6.67
4	N-B	supervised	0.557767	0.207252	2.23	supervised	0.930061	0.702969	1.38
5	DTC	supervised	0.976376	0.962036	1.57	supervised	0.975393	0.960591	1.15
6	KNN	supervised	0.975432	0.955774	319.09	supervised	0.976361	0.955758	179.53
7	NN	supervised	0.976978	0.956970	370.99	supervised	0.976495	0.959718	145.44
8	GR-B	supervised	0.989020	0.959267	46.95	supervised	0.989030	0.958905	19.30
9	ISO_FOR	unsupervised	0.374788	0.167981	105.36	unsupervised	0.366809	0.164455	20.41
10	LOF	unsupervised	0.454579	0.202236	279.19	unsupervised	0.471885	0.209633	104.80
11	KMEANS	unsupervised	0.729979	0.381899	1.43	unsupervised	0.730047	0.381983	0.71

Τα μοντέλα που ξεχώρισαν φαίνεται να είναι τα τρία μοντέλα του RFC, DTC και GR-B, τα οποία από πίσω κρύβουν την ίδια εγγενή λειτουργία, με το RFC να φαίνεται το πιο «καλό» και από τα τρία, λόγω απόδοσης σε συνδυασμό με χρονική πολυπλοκότητα. Είναι η πρώτη φορά που αποτυπώνουμε τις χρονικές πολυπλοκότητες των μοντέλων και δίνουν μία άλλη διάσταση στην

εικόνα τους. Οι χρόνοι που βλέπουμε είναι σε sec και έχει ενδιαφέρον πως τα μοντέλα εκπαιδευτήκαν όλα σε ένα μεγάλο subset του dataset, αλλά όχι σε ολόκληρο, με τους χρόνους εκπαίδευσης, στις περιπτώσεις του KNN και NN, να ξεφεύγουν ακόμα και με μειωμένο πλήθος δεδομένων.

Ωστόσο, κοιτώντας στα δεξιά, βλέπουμε τα αποτελέσματα της εκπαίδευσης των μοντέλων με την χρήση του dataset, που έχει επεξεργαστεί με SOM. Είναι φανερό, πως σε όλα τα μοντέλα υπάρχει επιτάχυνση, και στα πιο χρονοβόρα, είναι ακόμα πιο έντονη, με τις επιδόσεις των μοντέλων, όχι μόνο να μένουν ίδιες αλλά όπως στην περίπτωση του N-B, να βελτιώνονται κιόλας. Με ένα γρήγορο Μ.Ο., βγαίνει μαθηματικά πως με την χρήση SOM, η εκπαίδευση ολοκληρώνεται περίπου στο 50% του χρόνου χωρίς την χρήση SOM.

4.5 KDDCUP'99

4.5.1 Προ-επεξεργασία του dataset

Το dataset είναι διαθέσιμο στην ιστοσελίδα του KDDCUP. Το αρχείο που χρησιμοποιήσαμε εμείς είναι το corrected, με τα labels διορθωμένα και το *kddcup.names*, με τα ονόματα των χαρακτηριστικών του dataset.

Το διάβασμα του αρχείου και το πέρασμα του σε dataframe, έγινε με μία απλή εντολή read_csv(), όπως και το διάβασμα του αρχείου με τα ονόματα των features και με μία περαιτέρω επεξεργασία δημιουργήθηκε το dataframe με το dataset του kddcup.

Ενδιαφέρον, σε αυτό το dataset, είναι η ύπαρξη αρκετών ειδών label, όσον αφορά το είδος των πακέτων.

label	
smurf.	2807886
neptune.	1072017
normal.	972780
satan.	15892
ipsweep.	12481
portsweep.	10413
nmap.	2316
back.	2203
warezclient.	1020
teardrop.	979
pod.	264
guess_passwd.	53
buffer_overflow.	30
land.	21
warezmaster.	20
imap.	12
rootkit.	10
loadmodule.	9
ftp_write.	8
multihop.	7
phf.	4
perl.	3
spy.	2

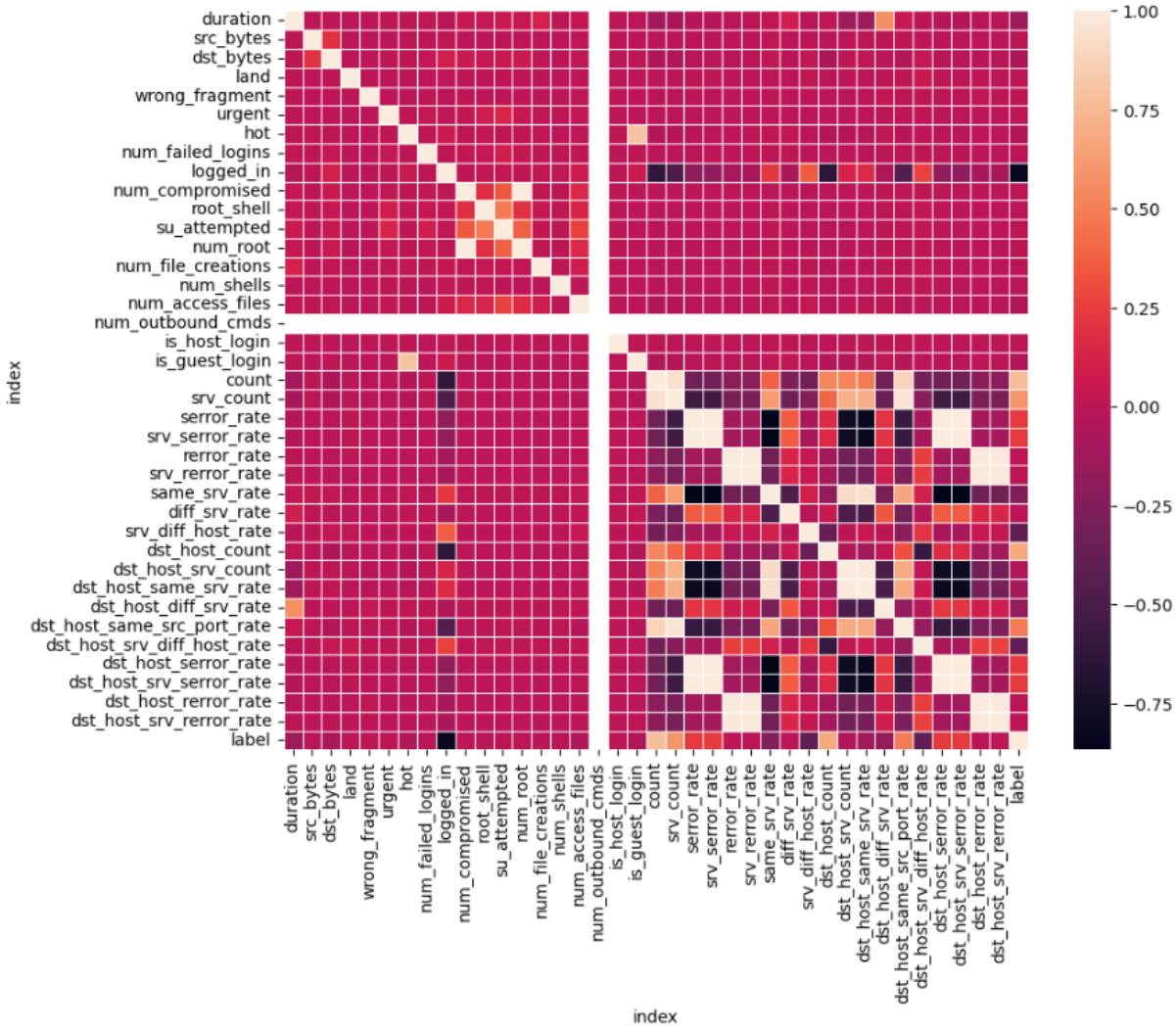
Από τον ιστότοπο του kddcup, τα label των 'back.', 'land.', 'neptune.', 'pod.', 'smurf.' και 'teardrop.', έχουν να κάνουν με DoS επιθέσεις και τα 'normal.', αποτελούν τα κανονικά πακέτα κίνησης δικτύου. Με την χρήση των παρακάτω εντολών λοιπόν, αδειάζουμε το dataset, από τα πακέτα με τα υπόλοιπα labels.

```
mask = kddcup_dataset['label'].isin(['back.', 'land.', 'neptune.', 'pod.', 'smurf.', 'teardrop.', 'normal.'])
kddcup = kddcup_dataset[mask]
```

Ουσιαστικά, φτιάχνουμε μία μάσκα με τις τιμές labels που μας αφορούν(1η εντολή) και περνάμε το dataset από αυτήν(2η εντολή).

Εν συνεχεία, με την χρήση της εντολής:
kddcup['label'] = kddcup['label'].apply(lambda x: 1 if x in ['back.', 'land.', 'neptune.', 'pod.', 'smurf.', 'teardrop.'] else 0)
αντικαθιστούμε τους διάφορους τύπους επίθεσης απλά σαν επίθεση, καθιστώντας τα label , σε μία Boolean τιμή, επίθεση ή όχι. Κάνουμε και έναν έλεγχο για το αν υπάρχουν τιμές NaN, με την εντολή *isnull()*, και εξακριβώνουμε πως το dataset είναι καθαρό.

Όπως και στο προηγούμενο dataset, με την χρήση της εντολής .corr() και του seaborn, οπτικοποιούμε την συσχέτιση μεταξύ των διαφόρων attributes του dataset, με σκοπό την αφαίρεση υψηλά συσχετιζόμενων features.
Η λίστα με τις εντολές είναι πανομοιότυπη με μοναδική αλλαγή το όνομα του dataset.



Κατευθείαν παρατηρούμε τον λευκό σταυρό, που μας λέει πως το στοιχείο ‘*num_outbound_cmds’*, παραμένει σταθερό και άρα δεν έχει συνεισφορά στην εκπαίδευση.
Παρακάτω εμφανίζονται και τα ζευγάρια με τα features με τα μεγαλύτερα correlations. Με μία απλή συνεπαγωγή μεταξύ των ζευγαριών, τα features που καταλήγουν να έχουν μεγάλη συσχέτιση είναι τα *['num_outbound_cmds', 'rerror_rate', 'srv_rerror_rate', 'dst_host_srv_rerror_rate', 'serror_rate', 'dst_host_srv_serror_rate', 'dst_host_serror_rate']*, τα οποία αφαιρούνται με μία απλή εντολή *drop()*.

```
Highly correlated feature pairs:
('num_root', 'num_compromised')
('srv_count', 'count')
('srv_serror_rate', 'serror_rate')
('srv_rerror_rate', 'rerror_rate')
('same_srv_rate', 'serror_rate')
('same_srv_rate', 'srv_serror_rate')
('dst_host_srv_count', 'same_srv_rate')
('dst_host_same_srv_rate', 'serror_rate')
('dst_host_same_srv_rate', 'srv_serror_rate')
('dst_host_same_srv_rate', 'same_srv_rate')
('dst_host_same_srv_rate', 'dst_host_srv_count')
('dst_host_same_src_port_rate', 'count')
('dst_host_same_src_port_rate', 'srv_count')
('dst_host_serror_rate', 'serror_rate')
('dst_host_serror_rate', 'srv_serror_rate')
('dst_host_serror_rate', 'same_srv_rate')
('dst_host_serror_rate', 'dst_host_same_srv_rate')
('dst_host_srv_serror_rate', 'serror_rate')
('dst_host_srv_serror_rate', 'srv_serror_rate')
('dst_host_srv_serror_rate', 'same_srv_rate')
('dst_host_srv_serror_rate', 'dst_host_same_srv_rate')
('dst_host_srv_serror_rate', 'dst_host_serror_rate')
('dst_host_rerror_rate', 'rerror_rate')
('dst_host_rerror_rate', 'srv_rerror_rate')
('dst_host_srv_rerror_rate', 'rerror_rate')
('dst_host_srv_rerror_rate', 'srv_rerror_rate')
('dst_host_srv_rerror_rate', 'dst_host_rerror_rate')
('label', 'logged_in')
```

Έτσι, καταλήγουμε στο dataset, με 34 features, εκ των οποίων τα 3 είναι τύπου object, τα οποία θα ονομάσουμε και πάλι **X** dataset, συν 1 για τις ετικέτες, το οποίο θα είναι το **y** dataset. Το αρχικό dataset είχε 41 features συν 1 για ετικέτες.

4.5.2 Τελικά configurations και εκπαίδευση μοντέλων

Προκειμένου, το dataset, να μπορέσει να χρησιμοποιηθεί στην εκπαίδευση των μοντέλων, μένει τα τρία features τύπου object, ‘protocol_type’, ‘service’ και ‘flag’, να μετατραπούν και αυτά σε αριθμητικές τιμές. Ομοίως, με την χρήση του OneHotEncoding, το dataset που δημιουργείται έχει 109 features, με μόνο αριθμητικούς τύπους και είναι έτοιμο να χρησιμοποιηθεί για εκπαίδευση.

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4856150 entries, 0 to 4856149
Columns: 109 entries, duration to flag_SH
dtypes: float64(109)
memory usage: 3.9 GB
```

Το split του dataset, γίνεται ξανά σε 70% training data και 30% test data.

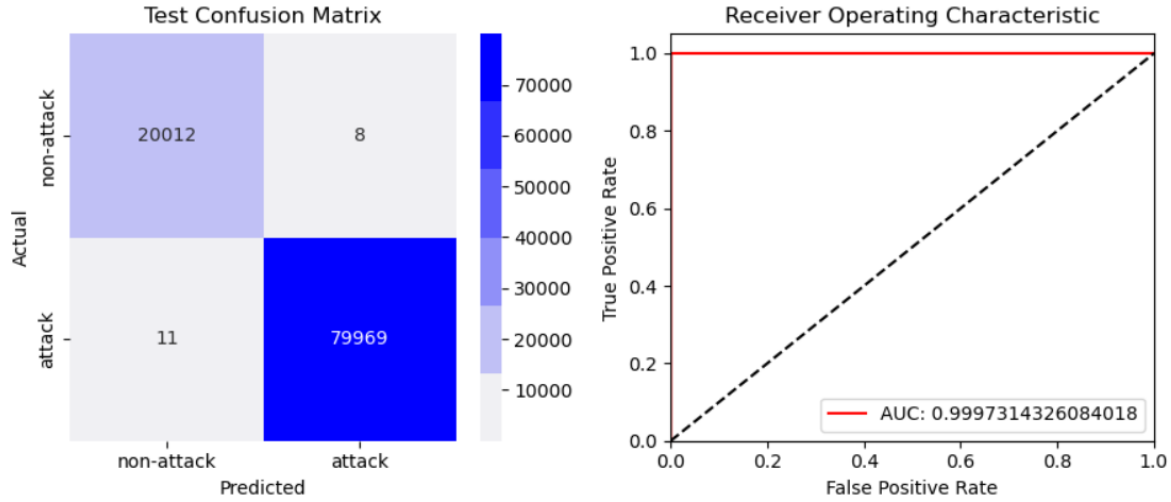
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)

4.5.3 Προσομοιώσεις Μοντέλων και Επιδόσεις

Logistic Regression (LR)

Ακολουθώντας την ίδια σειρά, ξεκινάμε με LR. Οι παράμετροι ορίζονται από την συνάρτηση μας, param_tuning(). Τα αποτελέσματα της εκπαίδευσης είναι:

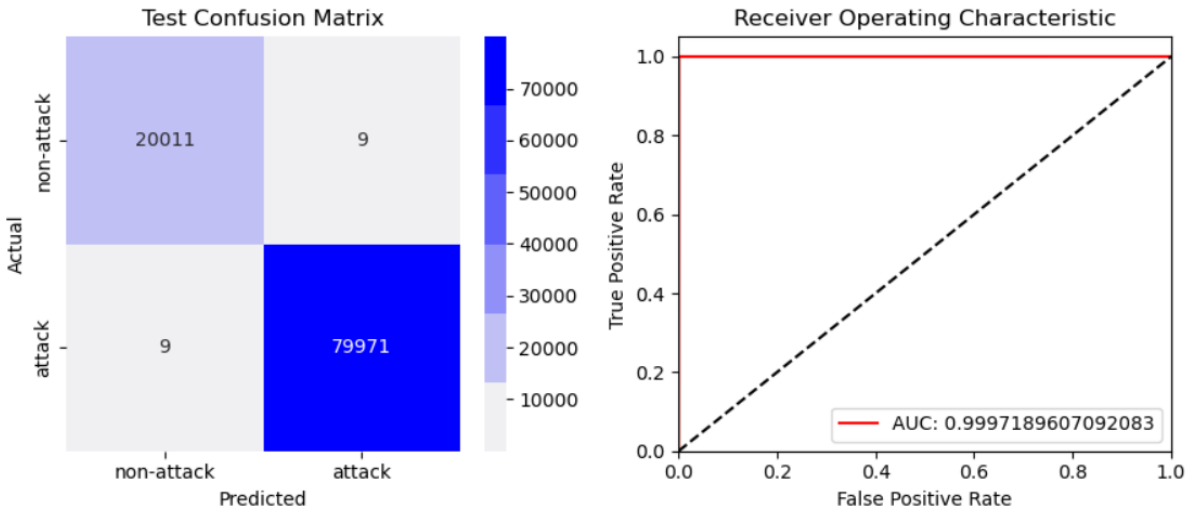
Dataset	Model	AUC	F1-score
Train	LR	0.9997129174616206	0.9998624140087556
Test	LR	0.9997314326084018	0.9998812180773583



Αυτό που παρατηρούμε άμεσα είναι οι άριστες επιδόσεις, κάτι που θα παρατηρηθεί καθ΄όλα τα αποτελέσματα πάνω σε αυτό το dataset. Ο λόγος είναι πως το dataset, ενώ είναι unbalanced, τα πακέτα επίθεσης DoS, είναι αυτά που υπερτερούν έναντι των κανονικών πακέτων. Τα μοντέλα που προσομοιώνουμε λοιπόν, έχουν περισσότερα δεδομένα για τις επιθέσεις, πράγμα που τα κάνει να γίνουν καλύτερα στην γενίκευση(υψηλά test score), καλύτερα στην μάθηση (υψηλά train score) και καλύτερα στην ταξινόμηση (υψηλά AUC score).

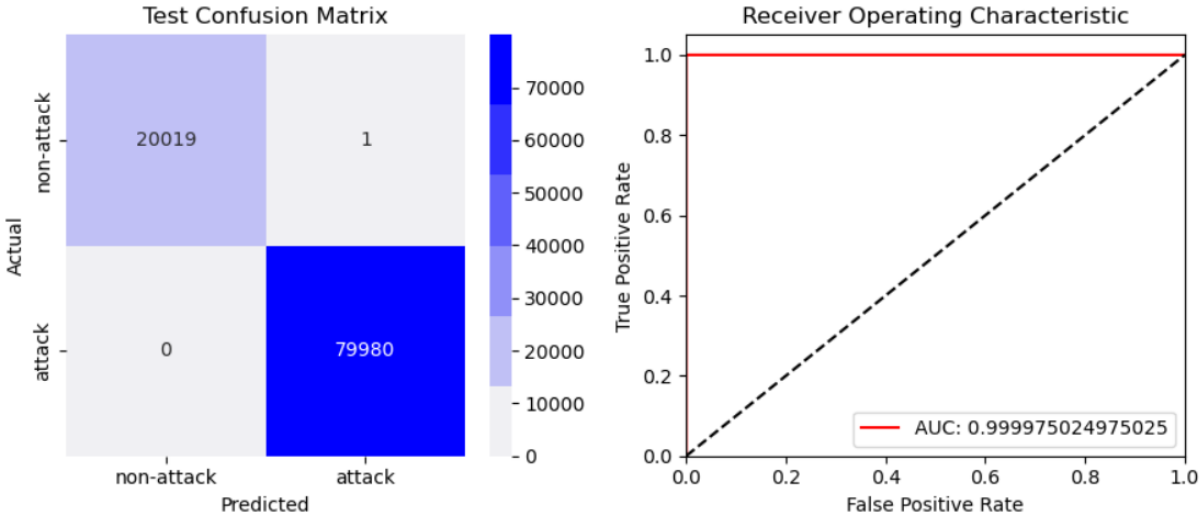
Support Vector Machine (SVM)

Dataset	Model	AUC	F1-score
Train	SVM	0.9995820301572882	0.9998436610822407
Test	SVM	0.9997189607092083	0.999887471867967



Radom Forest Classifier (RFC)

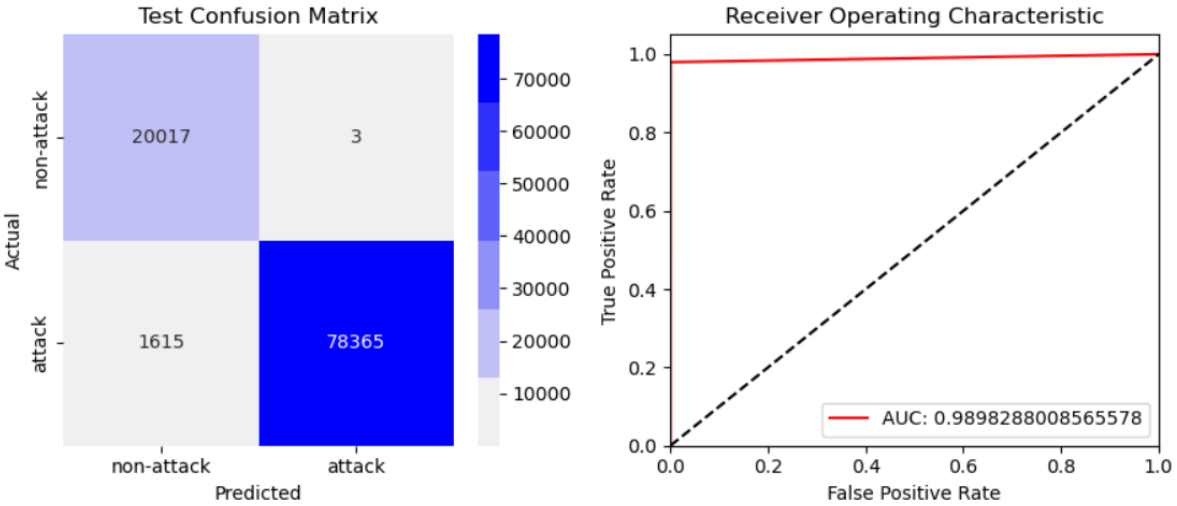
Dataset	Model	AUC	F1-score
Train	RFC	1.0	1.0
Test	RFC	0.999975024975025	0.9999937484761912



Αξίζει να σημειωθεί το 1.0 F1-Score, στα train data, το οποίο δηλώνει πόσο, πρακτικά, τέλεια, το RFC, μπορεί να διαβάσει το dataset και να εκπαιδευτεί πάνω σε αυτό.

Naïve Bayes (N-B)

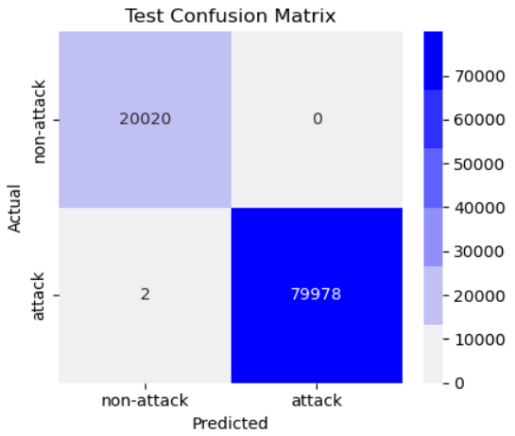
Dataset	Model	AUC	F1-score
Train	N-B	0.9902823770191106	0.9903404212333955
Test	N-B	0.9898288008565578	0.9897819991411322

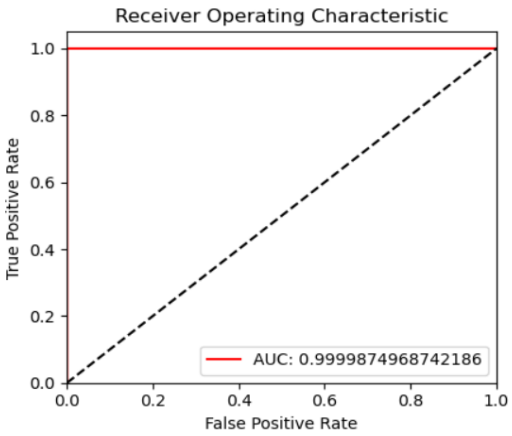


Στο προηγούμενο dataset, ο N-B, δεν θα ήταν ένα μοντέλο που θα προτιμούταν. Ωστόσο εδώ, βλέπουμε πως τα αποτελέσματα του, είναι καλύτερα από την καλύτερη επίδοση του καλύτερου μοντέλου στο προηγούμενο dataset.

Decision Tree Classifier (DTC)

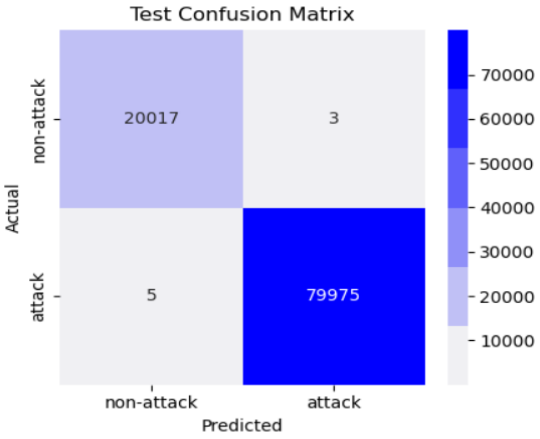
Dataset	Model	AUC	F1-score
Train	DTC	1.0	1.0
Test	DTC	0.9999874968742186	0.9999874967178884

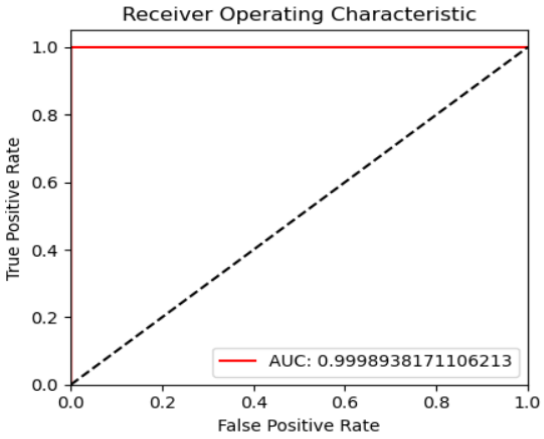




K Nearest Neighbors (KNN)

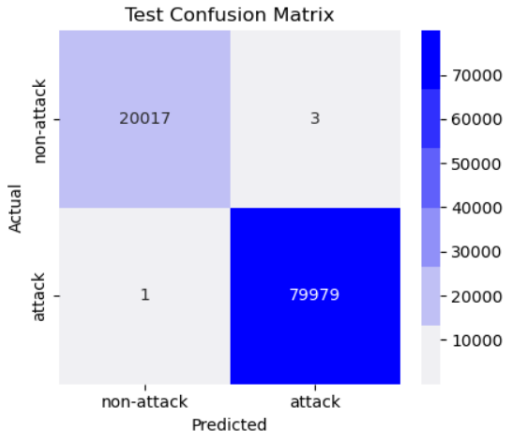
Dataset	Model	AUC	F1-score
Train	KNN	0.9998876537925826	0.9999437158777509
Test	KNN	0.9998938171106213	0.9999499868715538

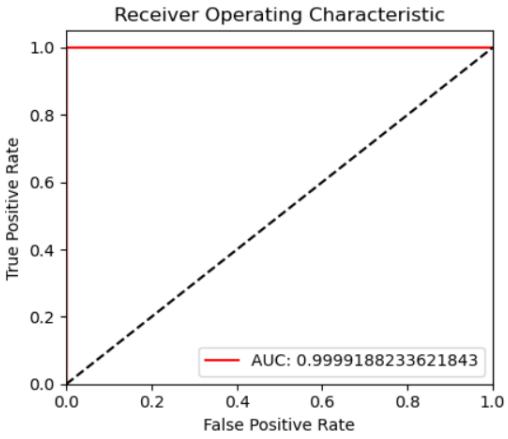




Neural Networks (NN)

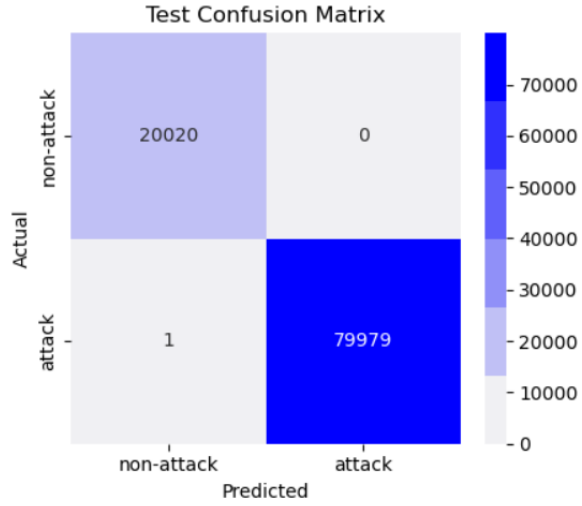
Dataset	Model	AUC	F1-score
Train	NN	0.9998440979995336	0.999956225650518
Test	NN	0.9999188233621843	0.9999749940610895

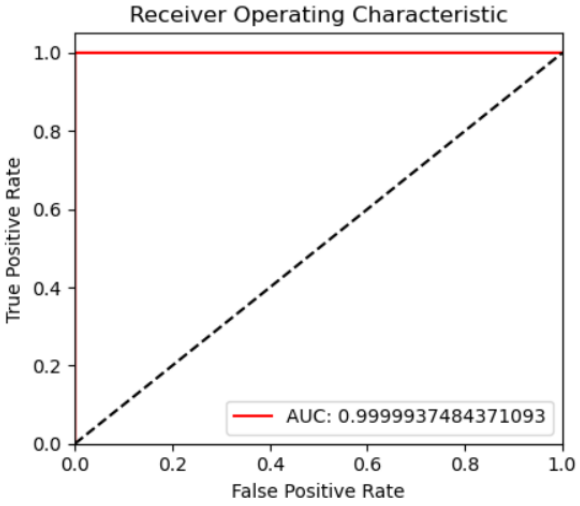




Gradient Boosting (GR-B)

Dataset	Model	AUC	F1-score
Train	GR-B	0.9999937463259665	0.9999937462868578
Test	GR-B	0.9999937484371093	0.9999937483980269



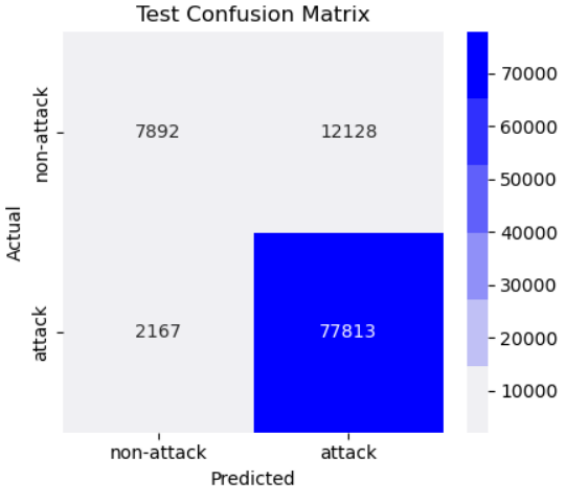


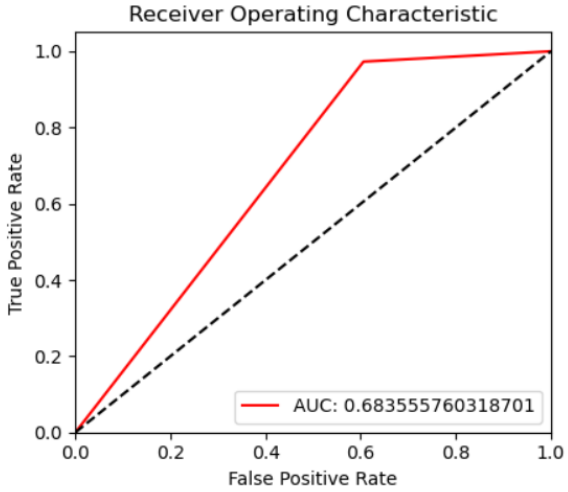
Isolation Forest (Iso_For)

Μπαίνοντας και πάλι στα unsupervised μοντέλα, βλέπουμε πως λόγω του καλού dataset, οι επιδόσεις είναι ανεβασμένες. Το dataset του KDDCUP, έχει λιγότερα είδη επίθεσης από το UNSW dataset, και τα «κατηγοριοποιεί» σε τέσσερις βασικές κατηγορίες, εκ των οποίων η μία είναι DoS attack. Αυτή η απομόνωση των επιθέσεων στο KDDCUP, είναι πιο κοντά στον τρόπο λειτουργίας του Iso_For, το οποίο απομονώνει τις ανωμαλίες μέσα στο dataset, προκειμένου να τις διακρίνει. Αυτό, συνεπώς, εξηγεί πώς το ίδιο μοντέλο έχει σημαντικά καλύτερες επιδόσεις, από το dataset του UNSW. Ωστόσο, και πάλι, τα αποτελέσματα του δεν μπορούν να συγκριθούν

με τις άριστες επιδόσεις των supervised μοντέλων παραπάνω.

Dataset	Model	AUC	F1-score
Train	Iso_For	0.683592297048567	0.9160532617841404
Test	Iso_For	0.683555760318701	0.9158726702408767

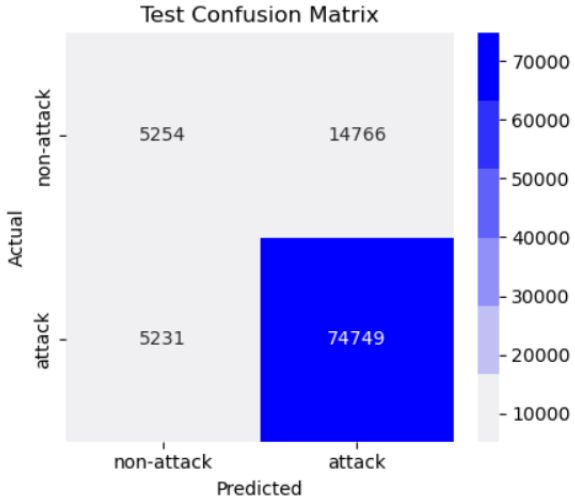


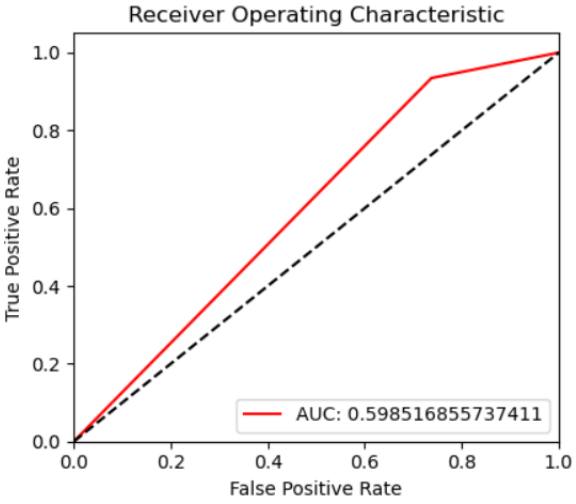


Local Outlier Factor (LOF)

Πάλι, στο ίδιο μοτίβο με το Iso_For, το μοντέλο του LOF, έχει αρκετά καλό F1-Score με σχετικά κακό AUC.

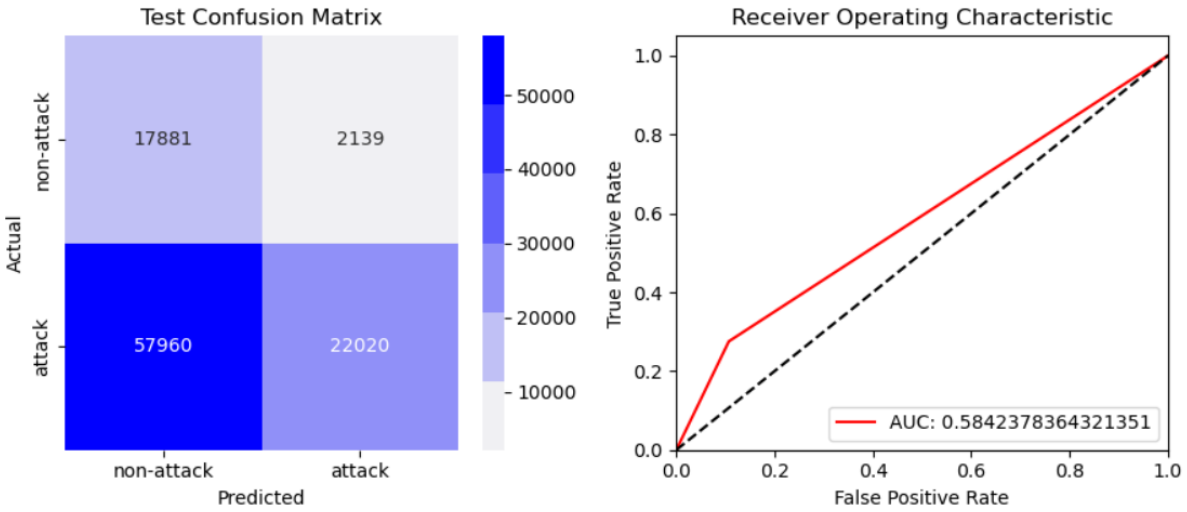
Dataset	Model	AUC	F1-score
Train	LOF	0.5784065866411388	0.882913034605405
Test	LOF	0.598516855737411	0.8820201185875689





K- Means (KMEANS)

Dataset	Model	AUC	F1-score
Train	KMEANS	0.5843322854656284	0.423736297490833
Test	KMEANS	0.5842378364321351	0.4228963212629274



Στο dataset του KDDCUP, τα είδη επιθέσεων μπορούν να θεωρηθούν σαν outliers, κάτι το οποίο ενώ βοηθά στην καλύτερη επίδοση μοντέλων όπως το Iso_For kai to LOF, δυσχεραίνει ιδιαίτερα το μοντέλο του KMEANS, του οποίου οι «ομαδοποιήσεις» επηρεάζονται αρνητικά από την ύπαρξη outliers. Ακόμα, ο KMEANS, υποθέτει την ύπαρξη σφαιρικών clusters (ομάδων) και σχετικά παρόμοιων μεγεθών, με κανένα από τα δύο να έχει ισχύ στο dataset αυτό. Συνεπώς, τα αποτελέσματα του είναι δικαιολογημένα σε χαμηλά επίπεδα.

4.5.4 Εφαρμογή dimensionality reduction με την χρήση SOM

Για μία ακόμα φορά θα εφαρμόσουμε dimensionality reduction με την χρήση SOM. Οι εντολές και ο κώδικας παραμένουν ίδιες και πάλι ο χρόνος δημιουργίας και εκπαίδευσης του SOM, είναι πολύ μικρός.

```
logged_in ===> Importance 199.4536747768545
root_shell ===> Importance 127.11553546516022
srv_diff_host_rate ===> Importance 121.69476897322974
dst_host_srv_diff_host_rate ===> Importance 115.55270215403418
flag_SF ===> Importance 99.432426892943
is_guest_login ===> Importance 87.89770153839372
protocol_type_tcp ===> Importance 87.72229597420622
hot ===> Importance 71.40786328537708
service_http ===> Importance 69.64647686546724
same_srv_rate ===> Importance 51.25598418859514
num_file_creations ===> Importance 42.284913295370814
dst_bytes ===> Importance 37.35365159559454
dst_host_same_srv_rate ===> Importance 35.811773884284186
num_access_files ===> Importance 23.852692812887447
dst_host_srv_count ===> Importance 14.208316975876386
su_attempted ===> Importance 14.171597153160807
service_smtp ===> Importance 10.882582892772612
```

Καλώντας στην οθόνη τα feature με την μεγαλύτερη βαρύτητα, και εν συνεχεία αφαιρώντας αυτά με την μικρότερη, καταλήγουμε στο τελικό dataset.

Το dataset χωρίς την χρήση των SOM, είχε 106 features. Μετά το SOM, έχει μόλις 28. Έχει ενδιαφέρον, εφόσον τα μοντέλα απέδωσαν άριστα, εάν προφανώς πέρα από την βελτίωση χρονική πολυπλοκότητα, διατήρησαν και τις επιδόσεις τους στο ίδιο επίπεδο. Η βελτίωση στις επιδόσεις των metrics, δεν έχει ιδιαίτερη σημασία καθώς τα

```
X_som.shape
```

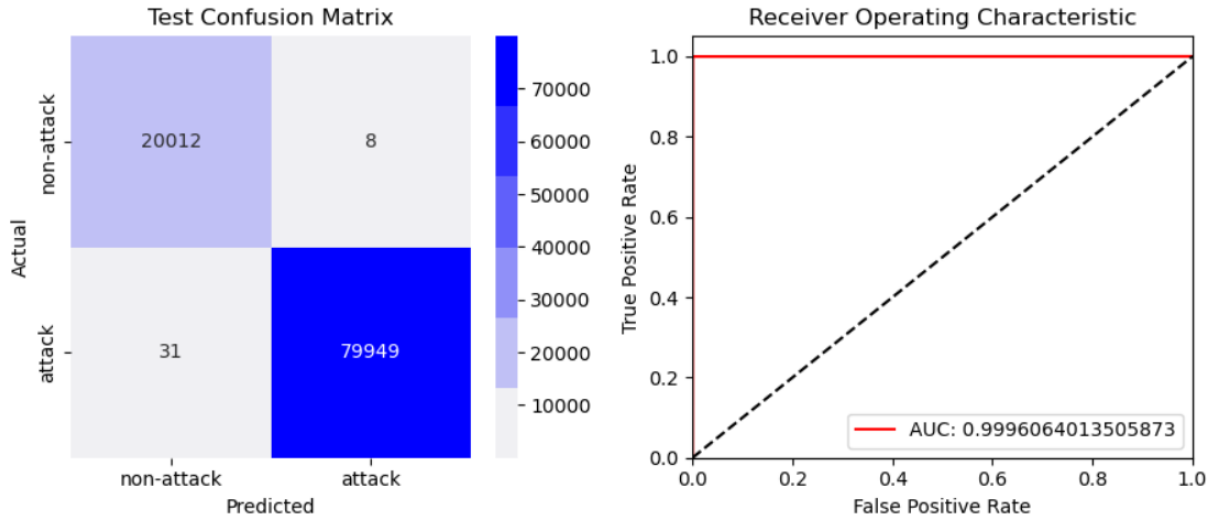
```
(4856150, 28)
```

περισσότερα είναι μακριά από την ιδανική μονάδα απόδοσης κατά μερικά χιλιοστά. Συνεπώς, η διατήρηση των επιδόσεων μας είναι αρκετή και μένει να διατηρηθεί σταθερή καθώς μειώνουμε την χρονική και υπολογιστική πολυπλοκότητα.

4.5.5 Προσομοιώσεις και Επιδόσεις με SOM

Logistic Regression (LR)

Dataset	Model	AUC	F1-score
Train	LR	0.9993819125882164	0.999643475922115
Test	LR	0.9996064013505873	0.9997561539856318

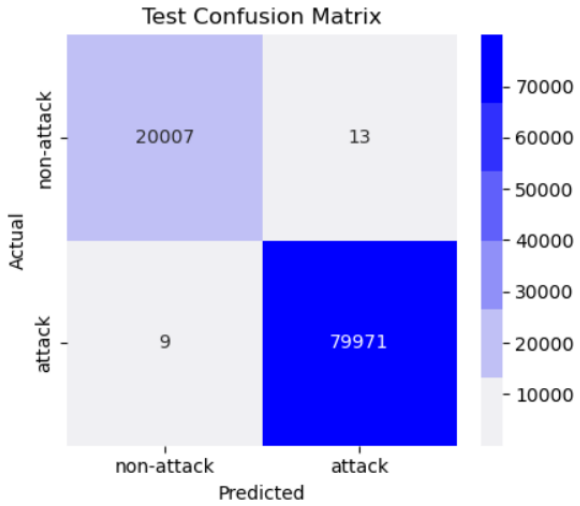


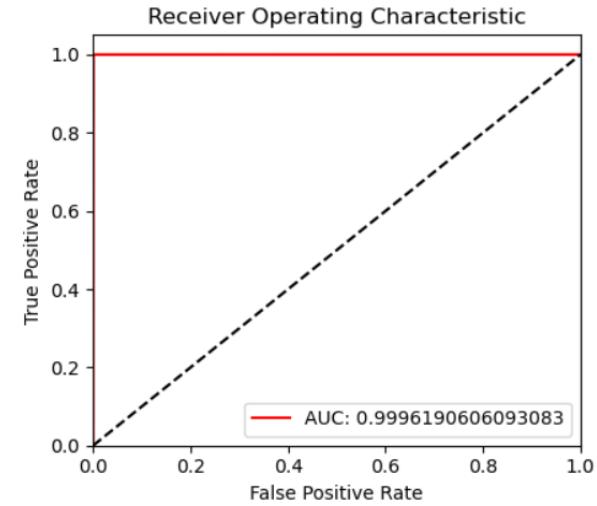
Ο χρόνος εκπαίδευσης θα παρουσιαστεί στο τέλος στον συγκεντρωτικό πίνακα, ωστόσο παρατηρούμε πως η πτώση της επίδοσης είναι στο τέταρτο δεκαδικό ψηφίο. Με απλά μαθηματικά, υπήρξε διαφορά στο AUC και στο F1-Score κατά 0,0001. Ας αναφερθεί όμως, ότι ο χρόνος εκπαίδευσης, είναι σχεδόν ο μισός.

Support Vector Machine (SVM)

Ίδιες καταστάσεις, με αλλαγές εδώ στο πέμπτο δεκαδικό.

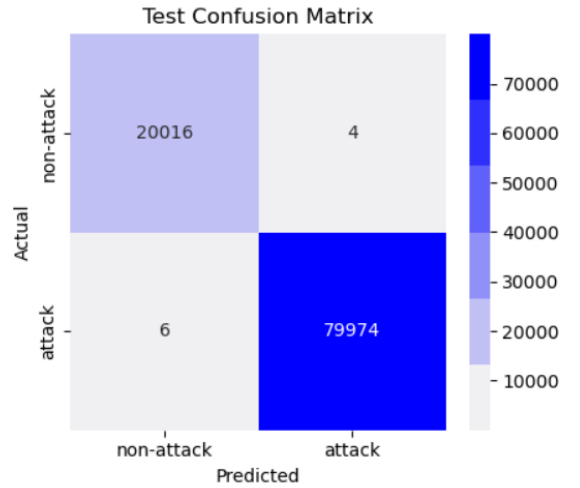
Dataset	Model	AUC	F1-score
Train	SVM	0.9994446692537369	0.9997623544420958
Test	SVM	0.9996190606093083	0.9998624690555376

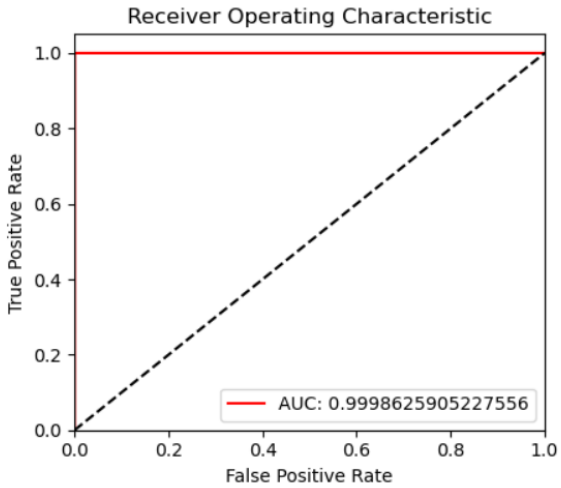




Radom Forest Classifier (RFC)

Dataset	Model	AUC	F1-score
Train	RFC	0.9999937463259665	0.9999937462868578
Test	RFC	0.9998625905227556	0.9999374835894422



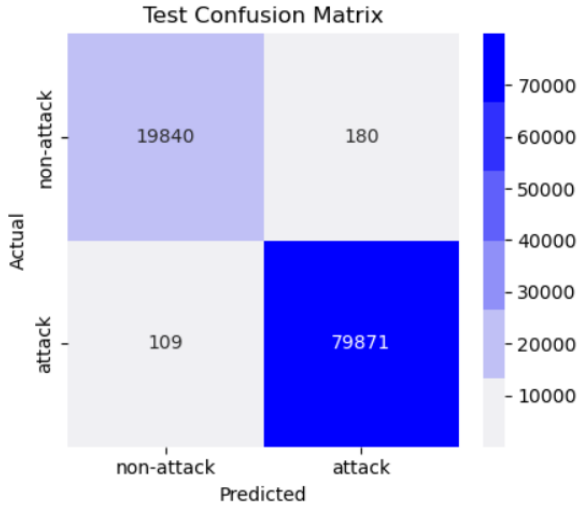


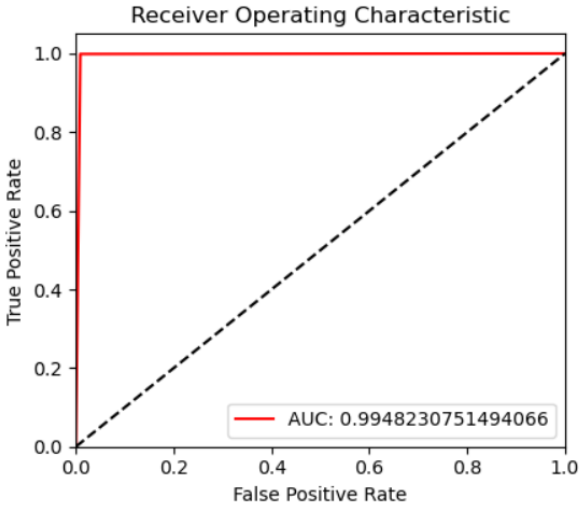
Στο μέχρι στιγμής καλύτερο μοντέλο και πάλι οι αλλαγές γίνονται ορατές στο έκτο δεκαδικό ψηφίο των metrics.

Naïve Bayes (N-B)

Παρά το ότι οι ίδιες επιδόσεις, λόγω της σημαντικής επιτάχυνσης, μας αρκούν, το μοντέλο του N-B, παρουσιάζει και καλύτερες επιδόσεις μετά την χρήση των SOM, με τα metrics του να βελτιώνονται στο δεύτερο δεκαδικό.

Dataset	Model	AUC	F1-score
Train	N-B	0.9946730710452913	0.9981686011988473
Test	N-B	0.9948230751494066	0.9981940998931457

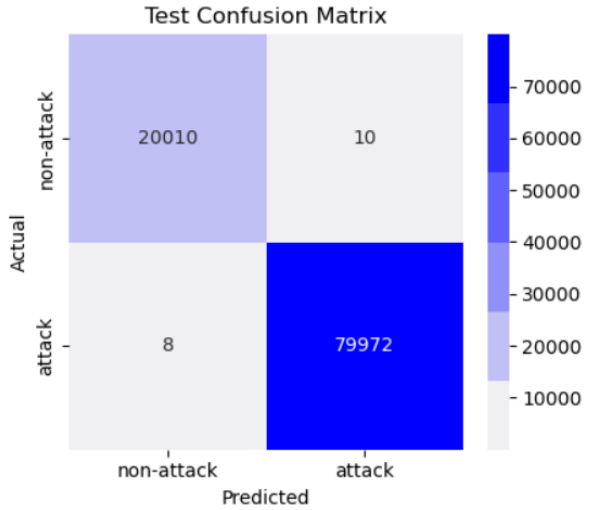


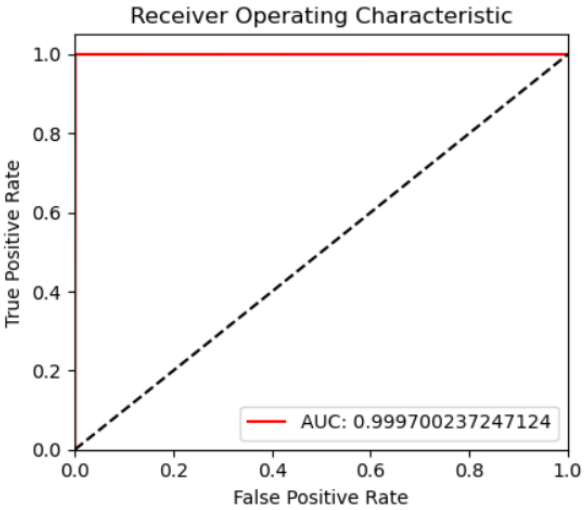


Decision Tree Classifier (DTC)

Και πάλι, αλλαγές στο τέταρτο δεκαδικό.

Dataset	Model	AUC	F1-score
Train	DTC	0.9999500439161272	0.9999687310431951
Test	DTC	0.999700237247124	0.9998874732749028

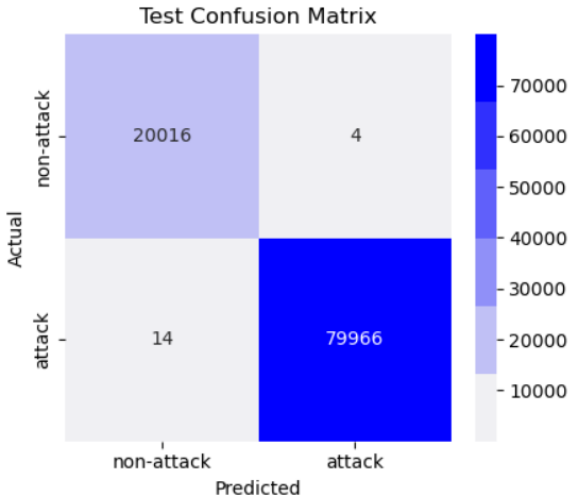


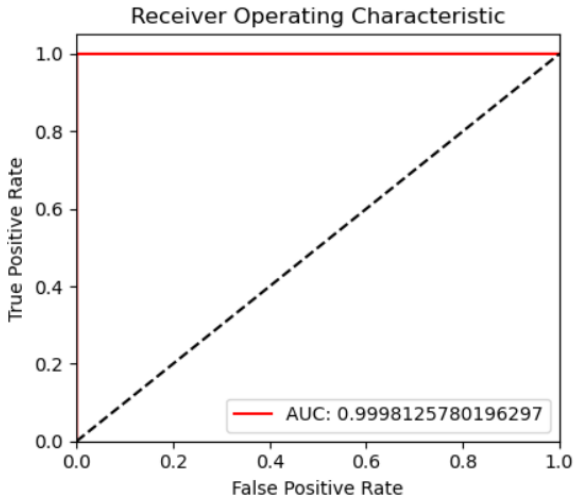


K Nearest Neighbors (KNN)

Μείωση στα AUC και F1-Score κατά 0,00001.

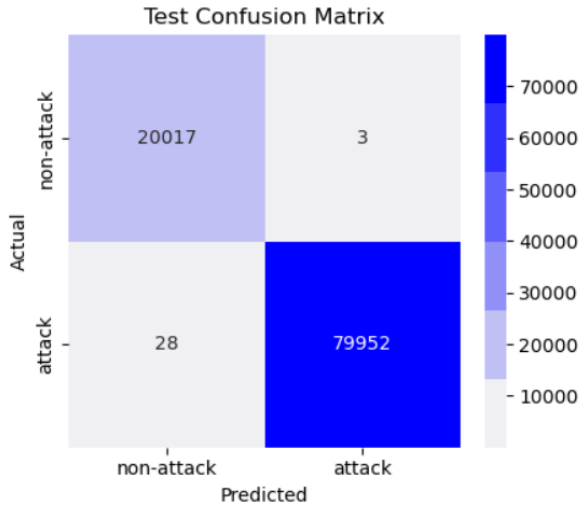
Dataset	Model	AUC	F1-score
Train	KNN	0.9998189366866094	0.9998936815575027
Test	KNN	0.9998125780196297	0.9998874648327603

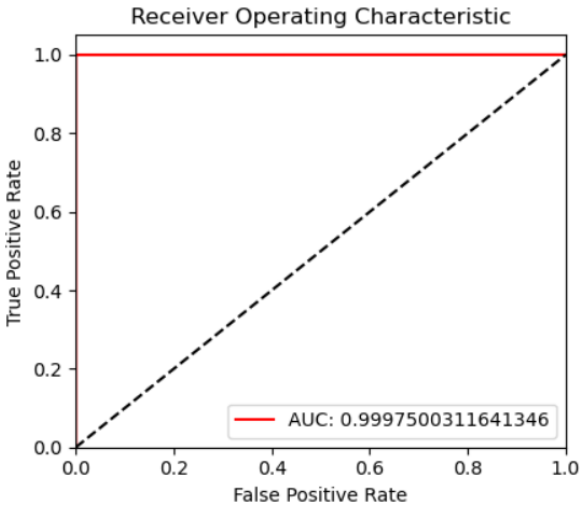




Neural Networks (NN)

Dataset	Model	AUC	F1-score
Train	NN	0.9997811214088277	0.9997810734905017
Test	NN	0.9997500311641346	0.9998061712570733

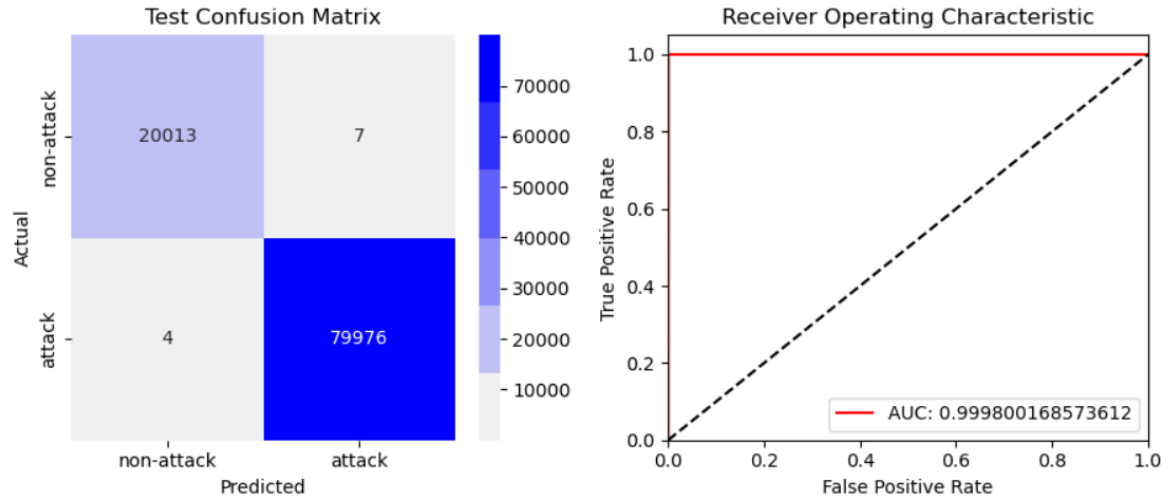




Μείωση στα metrics στο τέταρτο δεκαδικό, με την διαδικασία εκπαίδευσης να ολοκληρώνεται σχεδόν στο ένα τρίτο της αρχικής.

Gradient Boosting (GR-B)

Dataset	Model	AUC	F1-score
Train	GR-B	0.9999812389778995	0.999981238625917
Test	GR-B	0.999800168573612	0.9999312340978852

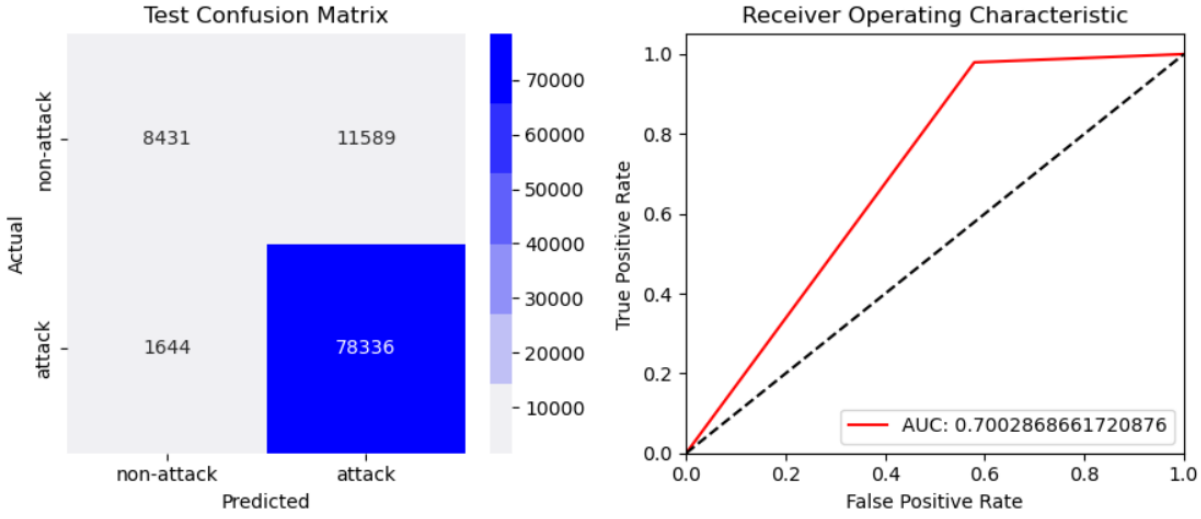


Αλλαγή στο πέμπτο δεκαδικό.

Isolation Forest (Iso_For)

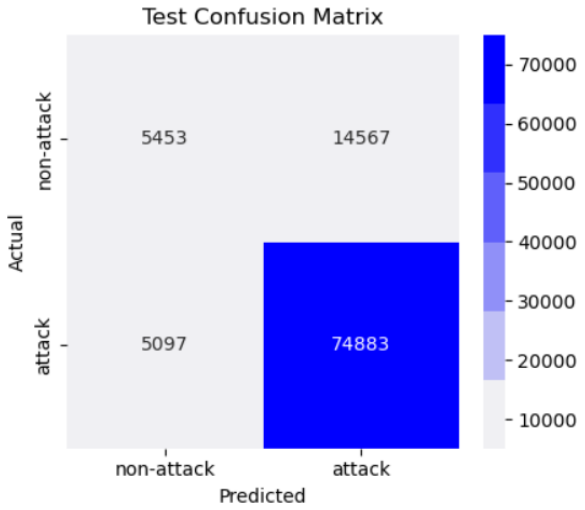
Στα unsupervised, το dimensionality reduction, πέρα από την επιτάχυνση στο Iso_For έφερε και ελαφριά αύξηση στα επιδόσεις. Στο dataset του KDDCUP, με τις επιθέσεις να μπορούν να θεωρηθούν σαν καλά outliers, το dimensionality reduction, όχι μόνο μπορεί να τονίσει τα outliers μέσα σε ένα χώρο μικρότερων διαστάσεων, αλλά να βοηθήσει και με τον χωρικό διαχωρισμό των ανωμαλιών καθώς προβάλλονται σε μικρότερες διαστάσεις. Unsupervised μοντέλα, σαν τα Iso_For και LOF, επηρεάζονται θετικά από την χρήση των SOM, και στις επιδόσεις, πέρα από την χρονική πολυπλοκότητα.

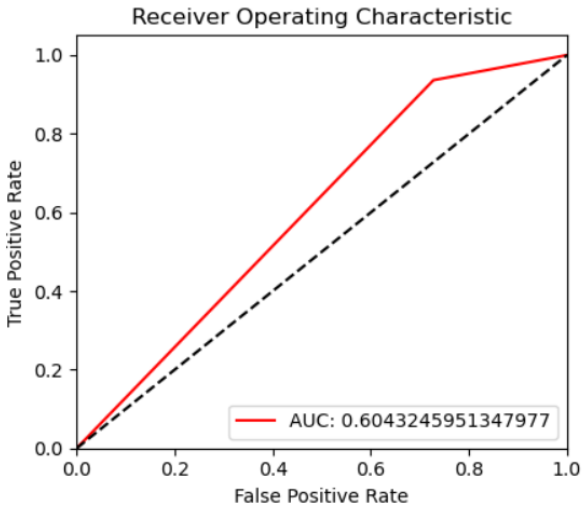
Dataset	Model	AUC	F1-score
Train	Iso_For	0.6991898279347213	0.9219372414726426
Test	Iso_For	0.7002868661720876	0.9221152997263176



Local Outlier Factor (LOF)

Dataset	Model	AUC	F1-score
Train	LOF	0.5836547836272427	0.8852120802396466
Test	LOF	0.6043245951347977	0.8839402703181254

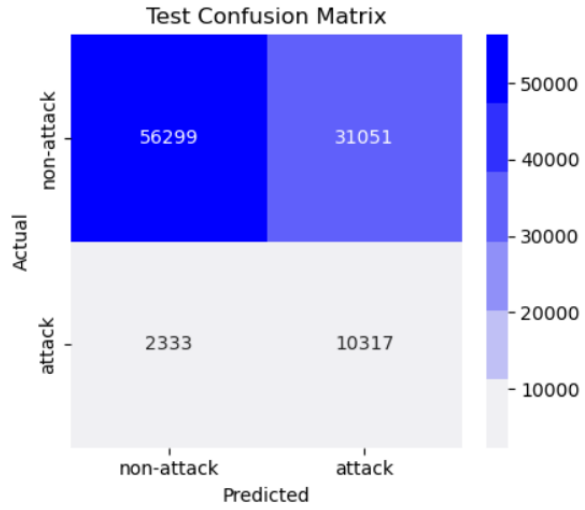


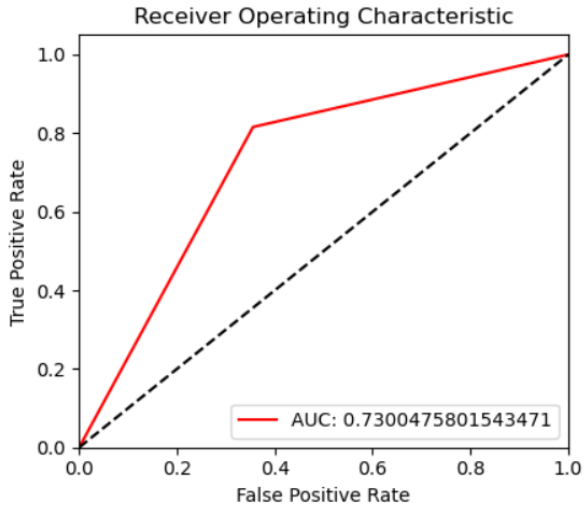


K- Means (KMEANS)

Ελαφρά βελτίωση.

Dataset	Model	AUC	F1-score
Train	KMEANS	0.732204688789957	0.385445142223864
Test	KMEANS	0.7300475801543471	0.38198378318338333





4.5.6 Αποτελέσματα και παρατηρήσεις

Όπως και πριν, με το ζευγάρι *start_time = time.time(),* πριν την εκπαίδευση και *end_time = time.time(),* μετά, συγκεντρώσαμε τους χρόνους εκπαίδευσης για κάθε μοντέλο. Το συγκεντρωτικό dataframe, με τα αποτελέσματα για τα metrics και τους χρόνους εκπαίδευσης για κάθε μοντέλο, παρουσιάζεται παρακάτω.

	Names	No SOM				With SOM			
		Type	AUC	F1	Time	Type	AUC	F1	Time
1	LR	supervised	0.999731	0.999881	4.33	supervised	0.999606	0.999756	2.30
2	SVM	supervised	0.999719	0.999887	1.53	supervised	0.999619	0.999862	0.87
3	RFC	supervised	0.999975	0.999994	3.97	supervised	0.999863	0.999937	3.75
4	N-B	supervised	0.989829	0.989782	1.37	supervised	0.994823	0.998194	0.79
5	DTC	supervised	0.999987	0.999987	1.11	supervised	0.999700	0.999887	0.74
6	KNN	supervised	0.999894	0.999950	117.24	supervised	0.999813	0.999887	61.76
7	NN	supervised	0.999919	0.999975	167.73	supervised	0.999750	0.999806	64.44
8	GR-B	supervised	0.999994	0.999994	19.78	supervised	0.999800	0.999931	9.31
9	ISO_FOR	unsupervised	0.683556	0.915873	67.38	unsupervised	0.700287	0.922115	13.51
10	LOF	unsupervised	0.598517	0.882020	167.19	unsupervised	0.604325	0.883940	86.37
11	KMEANS	unsupervised	0.584238	0.422896	0.99	unsupervised	0.578837	0.422062	0.70

Οι επιδόσεις εδώ πέρα, έχουν αμελητέες διαφορές μεταξύ τους, καθιστώντας όλα τα μοντέλα εξαιρετικά, όσον αφορά τις επιδόσεις τους. Ας δούμε όμως, και πάλι την επιτάχυνση που προσέφερε η χρήση των SOM. Οι χρόνοι είναι σε sec και αναλογικά με το πρώτο dataset, φαίνεται και πάλι η μεγάλη χρονική πολυπλοκότητα των KNN, NN και LOF.

Αριστερά, είναι τα metrics AUC, F1-Score και οι χρόνοι εκπαίδευσης χωρίς την χρήση των SOM και δεξιά με την χρήση SOM, για την επεξεργασία του dataset. Είναι φανερό, για μία ακόμα φορά, πως σε όλα τα μοντέλα υπάρχει σημαντική επιτάχυνση και οι επιδόσεις των μοντέλων, παραμένουν ίδιες σε πολύ ψηλά επίπεδα.

4.6 Συμπεράσματα

Βάζοντας τα πλεονεκτήματα που προκύπτουν από την χρήση SOM για dimensionality reduction σε ένα (Intrusion Detection System - IDS), έχουμε:

1. **Ταχύτερους χρόνους εκπαίδευσης**: Η χρήση των SOM για τη μείωση των διαστάσεων, μπορεί να οδηγήσει σε σημαντικά ταχύτερους χρόνους εκπαίδευσης, κάτι ιδιαίτερα κρίσιμο σε ένα IDS, καθώς επιτρέπει στο σύστημα να επεξεργάζεται και αναλύει τα δεδομένα κυκλοφορίας δικτύου πιο αποτελεσματικά, μειώνοντας την καθυστέρηση στον εντοπισμό και την αντίδραση σε παρεμβάσεις.
2. **Διατήρηση της απόδοσης**: Το γεγονός ότι τα αποτελέσματα στις μετρικές μετά τη χρήση των SOM παρέμειναν ίδια, υποδηλώνει ότι διατηρούν αποτελεσματικά σημαντικές πληροφορίες ενώ μειώνουν τον χώρο των χαρακτηριστικών. Αυτό εξασφαλίζει ότι το IDS σύστημά σας συνεχίζει να λειτουργεί καλά όσον αφορά την ακρίβεια της ανίχνευσης και τα ποσοστά ψευδών θετικών.

3. **Βελτιωμένη Οπτικοποίηση**: Τα SOM μπορούν να παρέχουν μια πολύ καλή οπτική παράσταση των δεδομένων, δίνοντας την δυνατότητα απόκτησης μίας πολύ καλής διαίσθηση της δομής κυκλοφορίας του δικτύου. Αυτή η οπτικοποίηση μπορεί να βοηθήσει σε βαθύτερη κατανόηση των δεδομένων και σε εντοπισμό μοτίβων και ανωμαλιών.
4. **Μείωση του Overfitting**: Με τη μείωση των διαστάσεων των δεδομένων, τα SOM μπορούν να βοηθήσουν στη μείωση του overfitting, όπου το μοντέλο μαθαίνει θόρυβο στα δεδομένα αντί για πραγματικά μοτίβα. Αυτό οδηγεί σε ένα πιο ανθεκτικό σύστημα IDS που γενικεύει καλύτερα σε νέες και μη γνωστές απειλές.
5. **Βελτιωμένη Real Time Ανίχνευση**: Οι ταχύτεροι χρόνοι εκπαίδευσης και η μείωση της διαστασιμότητας, επιτρέπουν την ανίχνευση πραγματικού χρόνου, κρίσιμη για έγκαιρη αντίδραση σε ασφάλειας.
6. **Λιγότεροι Υπολογιστικοί Πόροι**: Με λιγότερα χαρακτηριστικά προς επεξεργασία, το IDS σύστημα απαιτεί λιγότερους υπολογιστικούς πόρους, και κατ' επέκταση οικονομικά αποδοτικότερο.

Τα αποτελέσματα των προσομοιώσεων μας έδειξαν πως τα SOMs, έχουν την ικανότητα μείωσης των διαστάσεων, διατηρώντας, παράλληλα, τα ουσιώδη χαρακτηριστικά των δεδομένων, επιτρέποντας τον ταχύτερο εντοπισμό σε πραγματικό χρόνο ή χρόνο επιθέσεων, όπως DDoS, με τη μείωση του χώρου των χαρακτηριστικών και τη δυνατότητα γρηγορότερης ανάλυσης των δεδομένων κυκλοφορίας στο δίκτυο. Ακόμα, από την φύση τους, είναι προσαρμόσιμα σε μεταβαλλόμενα μοτίβα επιθέσεων, καθιστώντας τα κατάλληλα για σενάρια, όπου οι μέθοδοι επίθεσης μπορεί να εξελιχθούν με τον χρόνο.

Τελικώς, τα SOMs προσφέρουν αποτελεσματικότητα, ταχύτητα και ευελιξία στον τομέα της ανίχνευσης επιθέσεων, αλλά πρέπει να ενσωματωθούν σε μια συνολική στρατηγική αμυντικής ασφάλειας, που συνδυάζει πολλές τεχνικές για μία αξιόπιστη προστασία.

Επίλογος

Στόχος της παρούσας διπλωματικής εργασίας ήταν η μελέτη των εφαρμογών της Μηχανικής Μάθησης για την δημιουργία ενός πιο ανθεκτικού και προσαρμόσιμου Συστήματος Εντοπισμού Εισβολών (IDS) προσανατολισμένου στον τομέα του edge. Η αναζήτηση μίας αποτελεσματικής και αποδοτικής λύσης μας οδήγησε να εξερευνήσουμε τον κόσμο των Χαρτών Αυτο-οργάνωσης (Self-Organizing Maps - SOM) για τη μείωση της διαστασιμότητας, και αυτό αναμφισβήτητα αποτέλεσε και το βασικό σημείο, όσον αφορά την αντιμετώπιση επιθέσεων τύπου DDoS.

Επικεντρωθήκαμε στην αξιοποίηση της δύναμης των Χαρτών Αυτο-οργάνωσης, μιας τεχνικής μάθησης χωρίς επίβλεψη, βασισμένης σε νευρωνικά δίκτυα. Χρησιμοποιήσαμε τα SOMs για να μετασχηματίσουμε τα υψηλής διαστασιμότητας δεδομένα κυκλοφορίας δικτύου σε έναν πιο διαχειρίσιμο και αντιπροσωπευτικό χώρο χαρακτηριστικών. Το αποτέλεσμα ήταν η μείωση της πολυπλοκότητας των δεδομένων ενώ διατηρούσε τα βασικά του χαρακτηριστικά, επιτρέποντας έτσι πιο αποδοτική επεξεργασία σε ένα edge περιβάλλον.

Κατά τη διάρκεια των πειραμάτων, παρατηρήσαμε σημαντικές βελτιώσεις στις επιδόσεις των μοντέλων εντοπισμού εισβολών, σε δύο γνωστά σετ δεδομένων για επιθέσεις τύπου DDoS. Η μείωση της διαστασιμότητας μέσω SOM ενίσχυσε όχι μόνο την ακρίβεια στον εντοπισμό πιθανών επιθέσεων DDoS, αλλά μείωσε δραστικά τον χρόνο επεξεργασίας των δεδομένων και εκπαίδευσης των μοντέλων Μηχανικής Μάθησης, επιτρέποντας στο σύστημα να επιτελεί αποτελεσματικά και γρήγορα τον εντοπισμό ενδεχόμενων εισβολών, ενισχύοντας όχι μόνο τον τομέα της ασφάλειας αλλά και του φόρτου στους υπολογιστικούς πόρους στον edge, καθιστώντας το IDS πιο βιώσιμο και οικονομικά αποδοτικό.

Κατά τον Yann LeCun, "Η Μηχανική Μάθηση δεν έχει να κάνει μόνο με την ανάπτυξη των μοντέλων, αλλά και με την κατανόηση των δεδομένων και την βελτιστοποίηση των διαδικασιών επεξεργασίας τους." Σε αυτήν την εποχή όπου οι συσκευές edge, διαδραματίζουν κυρίαρχο ρόλο, η ανάπτυξη ανθεκτικών και έξυπνων συστημάτων εντοπισμού εισβολών με την χρήση Μηχανικής Μάθησης, θα χρειαστεί όποια βοήθεια μπορεί να βρει για να επιτύχει.

Βιβλιογραφία

[1] Islam, M. S., Pourmajidi, W., Zhang, L., & Steinbacher, J. (2020). *Anomaly Detection in a Large-scale Cloud Platform*

[2] Data Education. (2023, August 10). *A Brief History of Cloud Computing*. [Ηλεκτρονικό]. Available: https://www.dataversity.net/brief-history-cloud-computing/

[3] AI Consulting Group. *Carbon Benefits of Cloud Computing*. [Ηλεκτρονικό]. Available: https://www.aiconsultinggroup.com.au/carbon-benefits-of-cloud-computing/

[4] TechTarget. *Edge Computing*. [Ηλεκτρονικό]. Available: https://www.techtarget.com/searchdatacenter/definition/edge-computing/

[5] Xu, S., Qian, Y., & Hu, R. Q. (2020). *Data-Driven Edge Intelligence for Robust Network Anomaly Detection*. IEEE.

[6] Choi, J., Choi, C., Ko, B., Choi, D., & Kim, P. (2013). *Detecting Web-based DDoS Attack using MapReduce operations in Cloud Computing Environment*. Vol 3.

[7] Deshmukh, R. V., & Devadkar, K. K. (2015). *Understanding DDoS Attack & its Effect in Cloud Environment*. *Procedia Computer Science*, 49, 202-210.

[8] Wong, F., & Tan, C. X. (2014). *A Survey of Trends in Massive DDoS Attacks and Cloud-Based Mitigations*. *International Journal of Network Security & Its Applications (IJNSA)*, 6(3), Article 57.

[9] Osanaiye, O., Choo, K.-K. R., & Dlodlo, M. (2016). Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. *Journal of Network and Computer Applications*, 67, 147-165.

[10] Beitollahi, H., & Deconinck, G. (2012). Analyzing well-known countermeasures against distributed denial of service attacks. *Computer Communications*, 35(11), 1312-1332.

[11] Bhuyan, M. H., Kashyap, H. J., Bhattacharyya, D. K., & Kalita, J. K. (2014). Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions. *The Computer Journal*, 57(4)

[12] Darwish, M., Ouda, A., & Capretz, L. F. (2015, November 27). Cloud-based DDoS Attacks and Defenses.

[13] Rui, X., Wen-Li, M., & Wen-Ling, Z. (2009). Defending against UDP Flooding by Negative Selection Algorithm Based on Eigenvalue Sets. In Proceedings of IEEE Fifth International Conference on Information Assurance and Security (IAS'09), China, Vol. 2, pp. 342–345.

[14] Choi, J., Choi, C., Ko, B., & Kim, P. (2014, March 11). A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. *Soft Computing*, 18, 1697–1703.

[15] Karnwal, T., Sivakumar, T., & Aghila, G. (2012). A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack. In Proceedings of IEEE Students' Conference on Electrical, Electronics, and Computer Science (SCEECS), Bhopal, India.

[16] Gruschka, N., & Iacono, L. L. (2009). Vulnerable Cloud: SOAP Message Security Validation Revisited. In Proceedings of IEEE International Conference on Web Services (ICWS 2009), Los Angeles, USA.

[17] Bhuyan, M. H., Kashyap, H. J., Bhattacharyya, D. K., & Kalita, J. K. (2014, April). Detecting Distributed Denial of Service Attacks: Methods, Tools and Future Directions. *The Computer Journal*, 57(4). Oxford University Press.

[18] Iyengar, N. Ch. Sriman Narayana, Ganapathy, Gopinath, Kumar, P. C. Mogan, & Abraham, Ajith. (2014). A Multilevel Thrust Filtration Defending Mechanism Against DDoS Attacks in Cloud Computing Environment. *Journal Name*, 5(4), 236-248

[19] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys*, 41(3), Άρθρο No.: 15, 1–58

[20] Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. (2009). A detailed analysis of the KDD CUP 99 dataset. In Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA 2009), Ottawa, Canada, 1–6.

[21] Vissers, T., Somasundaram, T. S., Pieters, L., Govindarajan, K., & Hellinckx, P. (2014). DDoS defense system for web services in a cloud environment. *Future Generation Computer Systems*, 37, 37–45.

[22] Shamsolmoali, P., & Zareapoor, M. (2014). Statistical-based filtering system against DDoS attacks in cloud computing. In Proceedings of the International Conference on Advances in Computing, Communications, and Informatics (ICACCI), New Delhi, India, 1234–1239.

[23] Joshi, B., & Joshi, B. K. (2012). Securing cloud computing environment against DDoS attacks. In Proceedings of IEEE International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 1–5.

[24] Huang, V. S., Huang, R., & Chiang, M. (2013). A DDoS mitigation system with multi-stage detection and text-based Turing testing in cloud computing. In Proceedings of the 27th IEEE International Conference on Advanced Information Networking and Applications Workshops (WAINA), Barcelona, Spain, 655–662.

[25] Chonka, A., & Abawajy, J. (2012). Detecting and mitigating HX-DoS attacks against cloud web services. In Proceedings of the 15th IEEE International Conference on Network-Based Information Systems (NBiS), Melbourne, Australia, 429–434.

[26] Chonka, A., Xiang, Y., Zhou, W., & Bonti, A. (2011). Cloud security defence to protect cloud computing against HTTP-DoS and XML-DoS attacks. *Journal of Network and Computer Applications*, 34(4), 1097-1107.

[27] Lonea, A. M., Popescu, D. E., Prostean, Q., & Tianfield, H. (2013). Soft computing applications: Evaluation of experiments on detecting distributed denial-of-service (DDoS) attacks in Eucalyptus private cloud. 367–379.

[28] Gupta, S., & Kumar, P. (2013). VM profile-based optimized network attack pattern detection scheme for DDoS attacks in cloud. In Proceedings of the International Symposium on Security in Computing and Communications (SSCC 2013), Mysore, India, 255–261.

[29] Palmieri, F., Fiore, U., & Castiglione, A. (2013). A distributed approach to network anomaly detection based on independent component analysis. *Special Issue Paper*.

[30] Krishnan, D., & Chatterjee, M. (2012). An adaptive distributed intrusion detection system for cloud computing framework. In Proceedings of the International Conference on Recent Trends in Computer Networks and Distributed Systems Security (SNDS), Trivandrum, India, 466–473.

[31] Modi, C. N., Patel, D. R., Patel, A., & Muttukrishnan, R. (2012). Bayesian Classifier and Snort-based network intrusion detection system in cloud computing. In Proceedings of the 3rd International IEEE Conference on Computing, Communication & Networking Technologies (ICCCNT), Coimbatore, India, 1–7.

[32] Yang, L., Zhang, T., Song, J., Wang, J., & Chen, P. (2012). Defense of DDoS attack for cloud computing. In Proceedings of the IEEE International Conference on Computer Science and Automation Engineering (CSAE), vol. 2, Zhangjiajie, China, 626–629.

[33] Jeyanthi, N., Barde, U., Sravani, M., Tiwari, V., & Iyengar, N. Ch. S. N. (2013b). Detection of Distributed Denial of Service Attacks in Cloud Computing by Identifying Spoofed IP. *Journal Name*, 11(3), 262-279.

[34] Osanaiye, O. (2015). IP spoofing detection for preventing DDoS attack in cloud computing. In Proceedings of the 18th IEEE International Conference on Intelligence in Next Generation Networks (ICIN), Paris, France, 139–141.

[35] AltexSoft. Semi-Supervised Learning. AltexSoft Blog. [Ηλεκτρονικό]. Available: https://www.altexsoft.com/blog/semi-supervised-learning/

[36] Synopsys. AI Solutions. [Ηλεκτρονικό]. Available: https://www.synopsys.com/ai

[37] IBM. IBM Topics. [Ηλεκτρονικό]. Available: https://www.ibm.com/topics

[38] scikit-learn developers. scikit-learn. [Ηλεκτρονικό]. Available: https://scikit-learn.org/stable/

[39] JavaTpoint. Machine Learning. JavaTpoint. [Ηλεκτρονικό]. Available: https://www.javatpoint.com/machine-learning

[40] Hagemann, T., & Katsarou, K. (2020). A Systematic Review on Anomaly Detection for Cloud Computing Environments. In *AICCC '20: Proceedings of the 2020 3rd Artificial Intelligence and Cloud Computing Conference*, December 2020, Pages 83–96.

[41] Liu, F. T., Ting, K. M., & Zhou, Z. H. (2008). Isolation Forest. Available: http://www.lamda.nju.edu.cn/publication/icdm08b.pdf

[42] Miljković, D. (2017). Brief Review of Self-Organizing Maps. Available: http://161.53.22.65/datoteka/877545.Brief_Review_of_Self-Organizing_Maps.pdf

[43] Moustafa, N., Keshk, M., Choo, K.-K. R., Lynar, T., Camtepe, S., & Whitty, M. (2021). DAD: A Distributed Anomaly Detection system using ensemble one-class statistical learning in edge networks. *Future Generation Computer Systems*, 118, 240-251.

[44] Python Software Foundation. Python. [Ηλεκτρονικό]. Available: https://www.python.org/

[45] Project Jupyter. Jupyter. [Ηλεκτρονικό]. Available: https://jupyter.org/

[46] UNSW Sydney. UNSW-NB15 Dataset. UNSW Research. [Ηλεκτρονικό]. Available: https://research.unsw.edu.au/projects/unsw-nb15-dataset

[47] UCI Machine Learning Repository. KDD Cup 1999 Data. UCI Machine Learning Repository. [Ηλεκτρονικό]. Available: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html