ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σύστημα Ανίχνευσης Εισβολής με Μηχανική Μάθηση και Νευρωνικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Περικλής Κατσαρός

Επιβλέπων: Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Αθήνα, Ιούνιος 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Σύστημα Ανίχνευσης Εισβολής με Μηχανική Μάθηση και Νευρωνικά Δίκτυα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Περικλής Κατσαρός

Επιβλέπων: Θεοδώρα Βαρβαρίγου
Καθηγήτρια Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 20η Ιουνίου 2023.

……...	…….	……..……
Θεοδώρα Βαρβαρίγου	Εμμανουήλ Βαρβαρίγος	Συμεών Παπαβασιλείου
Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2023

…….

Περικλής Κατσαρός

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π

Περίληψη

Με την παρούσα διπλωματική εργασία είχα την ευκαιρία να μελετήσω την ανάπτυξη ενός Συστήματος Ανίχνευσης Εισβολής ή IDS (Intrusion Detection System), υλοποιημένου με χρήση Μηχανικής Μάθησης (Machine Learning, ML).

Στο πλαίσιο αυτό αρχικά περιγράφεται η θεωρία της Μηχανικής Μάθησης, η προ-επεξεργασία των δεδομένων και οι αλγόριθμοι που χρησιμοποιούνται. Ύστερα, αναλύονται τα είδη των IDS με τις διαφορετικές φιλοσοφίες υλοποίησης τους, με έμφαση στις δυνατότητες αξιοποίησης της Μηχανικής Μάθησης. Κατόπιν αυτού, παρουσιάζονται τα πιο ευρέως διαδεδομένα σύνολα δεδομένων που χρησιμοποιούνται για την ανάπτυξη IDS μαζί με αντιπροσωπευτικές έρευνες πάνω σε αυτά.

Στην συνέχεια της εργασίας παρουσιάζεται ο σχεδιασμός του συστήματος που υλοποιήθηκε και αφορά την περιγραφή διαφορετικών μοντέλων για την μεταξύ τους σύγκριση, την προεπεξεργασία των δεδομένων και την διαδικασία παραμετροποίησης των συστημάτων. Ακολούθως παρουσιάζονται τα αποτελέσματα της έρευνας πάνω σε δεδομένα προερχόμενα απ’ τα ίδια σύνολα δεδομένων επί των οποίων έγινε η εκπαίδευση, καθώς και επί πραγματικών ροών δεδομένων, μαζί με σχολιασμό για τα συμπεράσματα που μπορούν να προκύψουν από αυτά. Τέλος, η εργασία ολοκληρώνεται με την σύνοψη ενώ στο παράρτημα περιλαμβάνονται επιλεγμένα τμήματα κώδικα και γίνεται μία περεταίρω ανάλυση για επιμέρους θέματα μηχανικής μάθησης που ξεφεύγουν απ’ το αντικείμενο της μελέτης αλλά βοηθούν στην καλύτερη κατανόηση της εφαρμογής τους στα IDS.

Λέξεις κλειδιά:

Σύστημα Ανίχνευσης Εισβολής, Μηχανική Μάθηση, Τεχνητή Νοημοσύνη, Σύνολα Δεδομένων, scikit-learn, PyTorch

Abstract

With this thesis I had the opportunity to study the development of an Intrusion Detection System or IDS (Intrusion Detection System), implemented using Machine Learning (ML).

In this context, the theory of Machine Learning, the pre-processing of the data and the algorithms used are initially described. Then, the types of IDS are analyzed with their different implementation philosophies, with an emphasis on the possibilities of exploiting Machine Learning. Following this, the most widespread datasets used for IDS development are presented along with representative research on them.

In the continuation of the work, the design of the implemented system is presented, concerning the description of the different models used for comparison, the pre-processing of the data and the system parameterization process. Next, the results of the research are presented on data collected from the same data sets on which the training took place, as well as on real data streams, along with a commentary on the conclusions that can be drawn from them. Finally, the paper concludes with the summary, while the appendix includes selected sections of code and a further analysis is made for individual machine learning topics that are beyond the scope of the study but help to better understand their application in IDS.

Keywords:

Intrusion Detection System, Machine Learning, Artificial Intelligence, Datasets, scikit-learn, PyTorch

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω την οικογένεια μου για την καθοδήγηση, ηθική συμπαράσταση και υλική υποστήριξη που μου παρείχαν κατά την διάρκεια των σπουδών μου αλλά και τους φίλους μου για τις αξέχαστες στιγμές που έκαναν αυτά τα χρόνια μοναδικά. Ταυτόχρονα θα ήθελα να ευχαριστήσω την κ. Βαρβαρίγου Θεοδώρα και τον κ. Ευθύμιο Χονδρογιάννη για την εμπιστοσύνη και υποστήριξη τους στην συγγραφή αυτής της διπλωματικής εργασίας.

Αθήνα, Ιούνιος 2023

Περικλής Κατσαρός

Περιεχόμενα

Ευρετήριο Σχημάτων

Ευρετήριο Πινάκων

1 Εισαγωγή

Η ασφάλεια των υπολογιστικών συστημάτων αποτέλεσε αντικείμενο μελέτης απ' την απαρχή των ψηφιακών υπολογιστών τις δεκαετίες του '50 και του '60. Αρχικά η προσοχή εστιάστηκε στην ασφαλή και εξουσιοδοτημένη φυσική επαφή με τις υπολογιστικές μονάδες και την πρόληψη μη εξουσιοδοτημένων αλλαγών ή διαγραφής αρχείων [1]. Αργότερα η συνεχώς όλο και μεγαλύτερη διασυνδεσιμότητα των υπολογιστών, με προεξέχοντα σημεία καμπής την εφεύρεση του διαδικτύου και αργότερα του παγκόσμιου ιστού, δημιούργησε νέες εστίες απειλών και κατέστησε την προστασία των δεδομένων μια πολύπλοκη διαδικασία. Σήμερα, η ανάπτυξη του τομέα των επικοινωνιών και της ηλεκτρονικής έχει οδηγήσει στην παραγωγή δεκάδων δισεκατομμυρίων διασυνδεδεμένων και φθηνών μικροϋπολογιστικών συστημάτων, αισθητήρων και συσκευών και την ενσωμάτωση τους σε οικιακές συσκευές, αυτοκίνητα, ρουχισμό, βιομηχανικό και ιατρικό εξοπλισμό, και υποδομές ευφυών πόλεων. Οι αλλαγές αυτές οδήγησαν μεταξύ άλλων σε αύξηση της παραγωγικότητας και της οικονομικής δραστηριότητας, την βελτίωση των παροχών υγείας και την εξοικονόμηση ενέργειας. Εν τούτοις, η τεράστια αύξηση των πληροφοριών και η μεταφορά της πολιτικής, οικονομικής, κοινωνικής ζωής στην σφαίρα του διαδικτύου πολλαπλασιάζει τους κινδύνους διαρροής δεδομένων και κυβερνοεπιθέσεων [2].

Τα εργαλεία προστασίας έναντι επιθέσεων περιλαμβάνουν Τείχη Προστασίας (Firewalls), Αντιικά προγράμματα (Antivirus), Συστήματα Ανίχνευσης Εισβολής (Intrusion Detection Systems, IDS) καθώς και τακτικές ενημερώσεις λογισμικού και λειτουργικού συστήματος. Τα Firewalls κάνουν χρήση προεπιλεγμένων κανόνων που αφορούν διευθύνσεις IP και πύλες (ports) για να πάρουν μια απόφαση αν θα επιτρέψουν ή όχι κάποια κίνηση πακέτων. Ως εκ τούτου δεν έχουν την ικανότητα να αναγνωρίζουν συγκεκριμένες επιθέσεις. Τα Antivirus έχουν καταχωρημένη μια βάση υπογραφών γνωστών επιθέσεων και σαρώνουν το σύστημα για να δουν αν υπάρχει κάποια ταύτιση με κάποια υπογραφή. Απ' την άλλη πλευρά τα IDS έχουν στόχο να ειδοποιήσουν τον διαχειριστή για την ενδεχόμενη κακόβουλη δικτυακή κίνηση και η διαδικασία λειτουργίας τους είναι πιο ευέλικτη. Παραδοσιακά χρησιμοποιούσαν είτε βάσεις υπογραφών με παρόμοιο τρόπο με τα Antivirus, ή απλές στατιστικές μεθόδους σε συνδυασμό με ειδικές γνώσεις κυβερνοασφάλειας για την επιλογή των καταλληλότερων παραμέτρων, είτε κάποιον συνδυασμό αυτών των δύο. Αμφότερες αυτές οι δύο τεχνικές καθιστούσαν το σύστημα λιγότερο ευπροσάρμοστο και εν τέλει αναποτελεσματικό σε νεότερες επιθέσεις. Τα τελευταία χρόνια γίνεται προσπάθεια εισαγωγής τεχνικών μηχανικής

μάθησης για την κατανόηση βαθύτερων, πιο πολύπλοκων σχέσεων μεταξύ των δεδομένων με στόχο την καλύτερη κλιμακωσιμότητα, προσαρμοστικότητα και αποφυγή ανθρώπινης παρέμβασης στον σχεδιασμό των μοντέλων. Απ' την άλλη πλευρά η Μηχανική Μάθηση, και ειδικότερα η Βαθιά Μάθηση, εν γένει απαιτεί περισσότερα δεδομένα και είναι λιγότερη ερμηνεύσιμη απ' τα απλούστερα μαθηματικά μοντέλα που προηγήθηκαν αυτής και συχνά παρουσιάζονταν με εύληπτους τρόπους όπως διαγράμματα και πίνακες.

Μέχρι σήμερα οι έρευνες στον τομέα της ανάπτυξης IDS με την χρήση Μηχανικής Μάθησης περιορίζονται στην αξιολόγηση των μοντέλων πάνω σε γνωστά και συνήθως μη αντιπροσωπευτικά σύνολα δεδομένων και την βελτίωση διαφόρων μετρικών με στόχο συνήθως την επιλογή του βέλτιστου μοντέλου ή παραμετροποίησης. Ωστόσο, η βιβλιογραφία για την εξέταση των μοντέλων πάνω σε πραγματικά δίκτυα κρίνεται ελλιπής. Εκτός αυτού, η απόδοση ενός μοντέλου πάνω σε ένα σύνολο δεδομένων σπανίως μεταφράζεται σε αντίστοιχη ικανότητα σε πραγματικές συνθήκες [3]. Στόχος της παρούσας διπλωματικής είναι να παρέχει μια συνολική εικόνα για την ανάπτυξη συστημάτων εισβολής με μηχανική μάθηση, και παρουσιάζει ένα εργαλείο που αναπτύχθηκε το οποίο θα ήταν χρήσιμο για ερευνητές και ειδικούς στον κλάδο της κυβερνοασφάλειας.

Ακολουθεί η διάρθρωση της εργασίας: Στην ενότητα 2 παρουσιάζονται οι θεωρητικές έννοιες που είναι απαραίτητες για την κατανόηση των IDS. Στην υποενότητα 2.1 γίνεται μια εισαγωγή στην θεωρία τη Μηχανικής Μάθησης με έμφαση σε πρακτικές εφαρμογές και λιγότερο την μαθηματική θεμελίωση. Στην υποενότητα 2.2 παρουσιάζονται αναλυτικά τα IDS, ποιες είναι οι αρχές λειτουργίας τους, ποια είναι τα διάφορα είδη τους, τι προβλήματα παρουσιάζουν, ποιες τεχνικές μηχανικής μάθησης δυνητικά ενσωματώνουν και ποιες οι προϋποθέσεις που θα πρέπει να καλύπτουν. Επειδή τα δεδομένα αποτελούν ίσως το πιο σημαντικό συστατικό ενός IDS, η υποενότητα 3.1 αφιερώνεται στην παρουσίαση μερικών απ' τα πιο γνωστά σύνολα δεδομένων, πως δημιουργήθηκαν και τι προβλήματα παρουσιάζουν, ενώ η υποενότητα 3.2 ολοκληρώνει την εξέταση της βιβλιογραφίας μαζί με προηγούμενες μελέτες ερευνητών πάνω σε αυτά και την ανάπτυξη IDS. Στην ενότητα 4 παρουσιάζεται αναλυτικά η μεθοδολογία υλοποίησης των μοντέλων που εξετάστηκαν. Προς αυτόν τον σκοπό η υποενότητα 4.2 αφιερώνεται αρχικά σε μια σύντομη παρουσίαση των τεσσάρων μεθόδων αυξανόμενης πολυπλοκότητα που χρησιμοποιήθηκαν: Δένδρο Απόφασης, Μηχανές Υποστήριξης Διανυσμάτων, Perceptron Πολλαπλών Επιπέδων και Αναγεννητικά Αντιπαραθετικά Δίκτυα. Στη συνέχεια παρουσιάζεται η διαδικασία προεπεξεργασίας των

δεδομένων καθώς επίσης και παραμετροποίησης των μοντέλων και τέλος δίνεται μια περιγραφή του online συστήματος. Στην υποενότητα 4.3 γίνεται η παρουσίαση όλων των βιβλιοθηκών, πακέτων και εν γένει εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος. Η υποενότητα 4.4 ολοκληρώνει την ενότητα με την διαδικασία εγκατάστασης των πακέτων. Στην ενότητα 5 παρουσιάζονται όλα τα αποτελέσματα των μετρήσεων που έγιναν, με τα αποτελέσματα να περιλαμβάνουν αποτελέσματα πάνω σε σύνολα ελέγχου γνωστών datasets καθώς επίσης ενός online συστήματος. Πιο συγκεκριμένα, στην υποενότητα 5.2 παρουσιάζονται τα αποτελέσματα των μοντέλων που εκπαιδεύτηκαν πάνω στο σύνολο δεδομένων CIC-IDS2017, στην υποενότητα 5.3 τα αποτελέσματα πάνω στο UNSW-NB15, στην υποενότητα 5.4 αναφέρονται τα αποτελέσματα της εκπαίδευσης πάνω στο CIC-IDS2017 και του ελέγχου πάνω στο UNSW-NB15, και στην υποενότητα 5.5 ολοκληρώνεται η παρουσίαση των αποτελεσμάτων με τα αποτελέσματα των μοντέλων εκπαιδευμένων πάνω σε ένα σύνολο δεδομένων που αποτελεί ένωση αυτών των δύο. Στην ενότητα 6 ολοκληρώνεται η εργασία με την παρουσίαση της σύνοψης της μελέτης, των αποτελεσμάτων και ενδεχόμενων επεκτάσεων.

2 Σχετική Βιβλιογραφία

2.1 Πληροφορίες για Μηχανική Μάθηση και χρήση της σε IDS

2.1.1 Γενικές Πληροφορίες

Για την αποσαφήνιση των όρων, τεχνικών υλοποίησης και ταξινομήσεων των IDS που θα αναλυθούν στην συνέχεια, κρίνεται σκόπιμη μια εισαγωγική παρουσίαση της θεωρίας της *Μηχανικής Μάθησης*.

Η Μηχανική Μάθηση (Machine Learning ή ML) αποτελεί έναν κλάδο της *Τεχνητής Νοημοσύνης* (Artificial Intelligence ή AI) που περιλαμβάνει όλες τις τεχνικές και αλγορίθμους που επιτρέπουν στις μηχανές να μάθουν αυτόματα και χωρίς ανθρώπινη παρέμβαση/ρητό προγραμματισμό, χρησιμοποιώντας μαθηματικά μοντέλα με στόχο την εξαγωγή χρήσιμης πληροφορίας από μεγάλα *σύνολα δεδομένων* (datasets) για την επιλογή κάποιας απόφασης [4] [5] [6]. Τα σύνολα δεδομένων μπορούν να θεωρηθούν χωρίς βλάβη της γενικότητας ως μια σειρά εγγραφών με τις στήλες να έχουν τιμές κάποιων *χαρακτηριστικών* (features) ή ιδιοτήτων τους [7] [8].

Ένας άλλος όρος με τον οποίο αναφέρεται και αλληλεπιδρά στενά η Μηχανική Μάθηση είναι η *Εξόρυξη Δεδομένων* (Data Mining). Με τον όρο Data Mining αναφερόμαστε στην εξαγωγή χρήσιμης και άγνωστης πληροφορίας από μεγάλες βάσεις δεδομένων χρησιμοποιώντας τεχνικές Μηχανικής Μάθησης [5] [9]. Η «χρήσιμη πληροφορία» μπορεί να είναι η αναγνώριση κάποιου μοτίβου που βοηθάει τον άνθρωπο να καταλάβει τα δεδομένα [6]. Τέτοιου είδους πληροφορία στη συνέχεια μπορεί να εισαχθεί σε ένα σύστημα ανίχνευσης εισβολής και να ληφθεί υπόψη για την τελική απόφαση. Το Data Mining είναι απαραίτητο σε ένα IDS επειδή η σχετική πληροφορία προέρχεται από πολλές διαφορετικές πηγές όπως δεδομένα τοπικού υπολογιστή, δικτυακά δεδομένα και μηνύματα συναγερμού και αυτό αυξάνει την πολυπλοκότητα του συστήματος. Επίσης ο όγκος των δεδομένων σχετικών με την κίνηση ενός δικτύου είναι τεράστιος, επομένως απαιτείται περαιτέρω επεξεργασία του [10].

Ένα IDS υλοποιημένο με χρήση τεχνικών Μηχανικής και Βαθιάς Μάθησης συνήθως περιλαμβάνει τα ακόλουθα τρία βήματα: *Προεπεξεργασία* (Preprocessing) Δεδομένων, *Εκπαίδευση* (Training) και *Έλεγχος* (Testing) του μοντέλου. Προαιρετικά ανάμεσα στην φάση της εκπαίδευσης και ελέγχου εμφωλεύεται η *Επαλήθευση* (Validation) με σκοπό την καλύτερη παραμετροποίηση και επιλογή μεταξύ μοντέλων. Ο πιο συχνός τρόπος επαλήθευσης

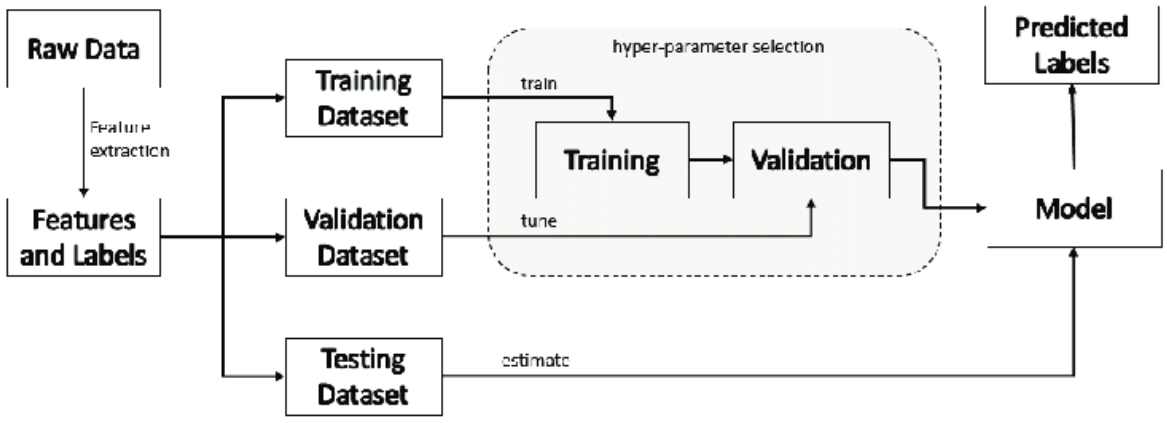
ονομάζεται διασταυρωμένη επικύρωση (αγγλ. cross validation ή n-fold cross validation) και χρησιμοποιεί μόνο το σύνολο δεδομένων εκπαίδευσης και το διαιρεί σε ν τμήματα, όπου τελικά το μοντέλο εκπαιδεύεται σύμφωνα με τα ν-1 και ελέγχεται σύμφωνα με το νιοστό τμήμα, με την διαδικασία να επαναλαμβάνεται για κάθε συνδυασμό τμημάτων για τον υπολογισμό κάποιου μέσου όρου [11] [12].

Η προεπεξεργασία των δεδομένων αφορά την μετατροπή τους σε μια μορφή κατάλληλη για να χρησιμοποιηθεί απ' τον αλγόριθμο. Σε αυτό το στάδιο υπεισέρχονται πολλές τεχνικές ανάλυσης δεδομένων και θα μπορούσε κανείς να ισχυριστεί ότι αποτελεί ένα απ' τα πιο κρίσιμα κομμάτια για την επιτυχή χρήση της μηχανικής μάθησης. Αυτό το στάδιο ενδεχομένως περιλαμβάνει *κωδικοποίηση* (encoding) και *κανονικοποίηση* (normalization) των δεδομένων. Ο πιο συχνός αλγόριθμος κωδικοποίησης είναι ίσως ο λεγόμενος «One-Hot-Encoding» ή OHE. Η συγκεκριμένη κωδικοποίηση αφορά την μετατροπή κατηγορικών[1] (categorical) χαρακτηριστικών σε ακολουθίες bits όπου μόνο ένα bit μπορεί να έχει την τιμή 1, με στόχο την αναπαράσταση της τιμής του χαρακτηριστικού με αριθμούς. Η κανονικοποίηση των χαρακτηριστικών λέγεται επίσης και *feature scaling* και έχει σκοπό τον περιορισμό των τιμών των χαρακτηριστικών σε ένα συγκεκριμένο εύρος τιμών. Οι δύο πιο διαδεδομένες τεχνικές κανονικοποίησης που χρησιμοποιούνται στην μηχανική μάθηση είναι το *Z-Score* και η *Min-Max* κανονικοποίηση. Πολλές φορές απαιτείται *καθαρισμός* (cleaning) με την έννοια της αφαίρεσης εγγραφών με απουσιάζοντα δεδομένα, *ακραίων τιμών* (outliers) ή διπλοεγγραφών. Άλλες φορές η αφαίρεση τους μπορεί να έχει μεγάλο κόστος στην απόδοση του μοντέλου, λ.χ. οι εγγραφές να περιέχουν χρήσιμη πληροφορία ανεξάρτητα απ' την απουσία τιμών σε κάποια χαρακτηριστικά που δεν μας συμφέρει να την πετάξουμε. Τότε, συνήθως χρησιμοποιούμε κάποια τεχνική συμπλήρωσης των απουσιαζόντων δεδομένων με τρόπο που δεν θα εισάγουμε έκτοπες ή ακραίες τιμές που θα οδηγούσε σε μεροληψία του μοντέλου, όπως π.χ. την χρησιμοποίηση κάποιου μέσου όρου του χαρακτηριστικού όπου απουσιάζει η τιμή. Μια άλλη διαδεδομένη τεχνική έχει να κάνει με την υποδειγματοληψία (undersampling) ή συχνότερα με *υπερδειγματοληψία* (oversampling) με σκοπό την αντιμετώπιση *ασύμμετρων* (unbalanced) συνόλων δεδομένων. Διαδεδομένη τεχνική υπερδειγματοληψίας αποτελεί ο αλγόριθμος SMOTE [13]. Τέλος, πολλές φορές είναι αναγκαία η *επιλογή*[2] (feature selection) ή *εξαγωγή* χαρακτηριστικών[3] (feature extraction) και γενικότερα η *μείωση της διαστατικότητας*

[1] Αντιπροσωπεύουν κατηγορίες και παρέχουν πληροφορίες μέσω ετικετών
[2] Επιλογή ενός υποσυνόλου των αρχικών χαρακτηριστικών.
[3] Εξαγωγή δευτερογενούς πληροφορίας απ' τον υπάρχοντα χώρο χαρακτηριστικών με στόχο την δημιουργία ενός νέου υποχώρου χαρακτηριστικών.

(dimensionality reduction) των δεδομένων για την αντιμετώπιση της λεγόμενης «κατάρας της διαστατικότητας»[4] [14] [15] [16]. Σε αυτό το βήμα είναι χρήσιμοι κλασικοί αλγόριθμοι όπως η μέθοδος *Ανάλυσης Κύριων Συνιστωσών* (Principal Component Analysis ή PCA [17].

Με την ολοκλήρωσης της φάσης προεπεξεργασίας, τα προεπεξεργασμένα δεδομένα ύστερα χωρίζονται τυχαίως σε μια δύο υποσύνολα, το υποσύνολο εκπαίδευσης και το υποσύνολο ελέγχου. Τυπικά, το σύνολο δεδομένων εκπαίδευσης καταλαμβάνει περίπου το 60% με 80% του αρχικού dataset και το υπόλοιπο αποτελεί το dataset ελέγχου. Στην συνέχεια ο αλγόριθμος μηχανικής μάθησης που επελέγη εκπαιδεύεται χρησιμοποιώντας το υποσύνολο εκπαίδευσης κατά την διάρκεια της φάσης εκπαίδευσης. Ο χρόνος εκπαίδευσης εξαρτάται απ' το μέγεθος των δεδομένων εκπαίδευσης και την πολυπλοκότητα του προτεινόμενου μοντέλου. Συνήθως τα μοντέλα βαθιάς μάθησης απαιτούν περισσότερο χρόνο λόγω της βαθιάς και πολύπλοκης δομής τους. Όταν το μοντέλο είναι πλέον εκπαιδευμένο, ελέγχεται χρησιμοποιώντας το σύνολο δεδομένων ελέγχου και αξιολογείται με βάση τις προβλέψεις που έκανε με διάφορες μετρικές αξιολόγησης και οι οποίες παρατίθενται αμέσως μετά. Εφόσον το μοντέλο φανεί ότι μπορεί να λειτουργήσει σωστά στο υποσύνολο ελέγχου, δοκιμάζεται σε πραγματικές συνθήκες εκτός «δοκιμαστικού σωλήνα». Η διαδικασία φαίνεται στην Εικόνα 1 που ακολουθεί.



Εικόνα 1: Διαδικασία Μηχανικής Μάθησης[5]

[4] Φαινόμενο κατά το οποίο ο χώρος χαρακτηριστικών γίνεται όλο και περισσότερο αραιός/άδειος όσο αυξάνονται οι άξονες των χαρακτηριστικών, με αποτέλεσμα ακόμη και τα κοντινότερα δείγματα να φαίνονται μακριά το ένα απ' το άλλο και κατά συνέπεια η απόδοση του μοντέλου να μειώνεται. Πρακτικά, αυτό απαιτεί εκθετική αύξηση του αναγκαίου αριθμού παραδειγμάτων όσο ο αριθμός χαρακτηριστικών αυξάνεται.

[5] https://www.researchgate.net/figure/General-workflow-diagram-of-machine-learning-algorithms_fig1_337173375

2.1.2 Ταξινόμηση μεθόδων Μηχανικής Μάθησης

2.1.2.1 Ταξινόμηση με βάση την χρήση ή όχι ετικετών

Οι τεχνικές της Μηχανική Μάθησης μπορούν να χωριστούν ανάλογα με την ύπαρξη και χρήση ή όχι ετικετών κατά την εκπαίδευση. *Ετικέτα* (label) είναι ένα χαρακτηριστικό στο dataset που προσδιορίζει την κατηγορία ή *κλάση* (class) ή επιθυμητό αποτέλεσμα της εκάστοτε εγγραφής, καθώς και το σύνολο των κατηγοριών που θα κληθεί το σύστημα να ταξινομήσει τις άγνωστες εγγραφές αργότερα όταν θα έχει ολοκληρώσει την εκπαίδευση του. Τα συστήματα στην πρώτη κατηγορία ονομάζονται συστήματα μηχανικής μάθησης *με επίβλεψη* (supervised) ενώ στην δεύτερη *χωρίς επίβλεψη* (unsupervised) [18] [19]. Σε γενικές γραμμές για μια πρώτη σύγκριση μπορούμε να πούμε ότι τα μοντέλα με επίβλεψη χρησιμοποιούνται πιο συχνά για την τελική απόφαση ενός συστήματος και επιπροσθέτως, τουλάχιστον στο πλαίσιο ανάπτυξης ενός IDS, είναι συνήθως πιο εύστοχα [20], αν και κάτι τέτοιο εξαρτάται και από τις τεχνικές που εφαρμόζονται και τα διαθέσιμα δεδομένα. Τα μοντέλα χωρίς επίβλεψη, παρόλο που εντοπίζονται πολύ συχνά στην πράξη αφού δεν απαιτούν ετικέτες, η συλλογή των οποίων είναι μια δύσκολη και κοστοβόρα υπόθεση αν όχι αδύνατη, και συχνά είναι ένα χρήσιμο βήμα πριν την εισαγωγή μοντέλων με επίβλεψη, έχουν το αρνητικό ότι υποθέτουν μια συγκεκριμένη κατανομή για τα δεδομένα και ο θόρυβος προκαλεί μείωση της απόδοσης τους [20] [21]. Τα τελευταία χρόνια χρησιμοποιείται επίσης και μια τεχνική *ημι-επιβλεπόμενης* (semi-supervised) μάθησης για προβλήματα όπου υπάρχουν μεν ετικέτες αλλά είναι λίγες, αποτελούμενη από δύο βήματα: Αρχικά ενός βήματος μάθησης χωρίς επίβλεψη ακολουθούμενο από ένα στάδιο επιβλεπόμενης μάθησης για την κατάλληλη παραμετροποίηση του μοντέλου. Αυτή η μεσοβέζικη λύση παράγει συστηματικά καλύτερα αποτελέσματα απ’ την unsupervised [22]. Τέλος υπάρχει και η περίπτωση της *ενισχυτικής μάθησης* (reinforcement learning). Σύμφωνα με την τελευταία ένας πράκτορας (agent) αλληλοεπιδρά με το περιβάλλον του και δέχεται σήματα επιβράβευσης ανάλογα με το κατά πόσο κοντά ενεργεί σύμφωνα με μια προκαθορισμένη λειτουργία. Ανάλογα με τα σήματα επιβράβευσης εκπαιδεύεται ώστε να μεγιστοποιήσει την ανταμοιβή μαθαίνοντας μέσα από μια διαδικασία trial and error [18]. Η τελευταία μέθοδος είναι προς το παρόν κυρίως αντικείμενο έρευνας με περιορισμένες εφαρμογές εκτός των παιχνιδιών αλλά είναι πολλά υποσχόμενη [23].

Τα supervised μοντέλα μπορούν να χωριστούν περεταίρω σε μοντέλα κατηγοριοποίησης ή *ταξινόμησης* (classification) και *παλινδρόμησης* (regression) ενώ τα unsupervised σε μοντέλα *συσταδοποίησης* (clustering) και *συσχέτισης* (correlation).

2.1.2.1.1 Επιβλεπόμενη μάθηση

Στόχος της ταξινόμησης είναι, δοθέντος μιας εισόδου, η τοποθέτηση της στην κατάλληλη κατηγορία. Η λειτουργία της βασίζεται σε δύο μέρη: Αφενός την εκπαίδευση του μοντέλου με βάση τα γνωστά δεδομένα που έχουμε, και αφετέρου την ταξινόμηση των αγνώστων δειγμάτων που θέλουμε να κατηγοριοποιήσουμε [24]. Τα γνωστά δεδομένα (ή δεδομένα εκπαίδευσης) συνοδεύονται από μια ετικέτα που αναγράφει την κατηγορία στην οποία ανήκουν. Παραδείγματος χάριν, στην δυαδική περίπτωση ταξινόμησης ενός IDS με χρήση μηχανικής μάθησης, η πρόβλεψη/ταξινόμηση θα είναι κανονική λειτουργία ή μη κανονική/επίθεση.

Η παλινδρόμηση απ' την άλλη πλευρά χρησιμοποιείται για την εξαγωγή μιας εκτίμησης για την σχέση μεταξύ μιας εξαρτημένης μεταβλητής, δηλαδή της ετικέτας (στην βιβλιογραφία αναφέρεται και ως απόκριση ή αποτέλεσμα), και ενός συνόλου ανεξάρτητων μεταβλητών που είναι τα χαρακτηριστικά που αναφέραμε και ονομάζονται επίσης predictors και explanatory variables. Η πιο διαδεδομένη μέθοδος παλινδρόμησης είναι η *γραμμική παλινδρόμηση* και σκοπό έχει την χάραξη της καταλληλότερης ευθείας ανάμεσα στα δείγματα με σκοπό την πρόβλεψη αποτελεσμάτων για διαφορετικές τιμές των predictors.

2.1.2.1.2 Μη Επιβλεπόμενη Μάθηση

Στην τεχνική της συσταδοποίησης τα διάφορα δείγματα εκπαίδευσης δεν έχουν ή δεν απαιτείται να έχουν ετικέτα. Ταξινομούνται σε συστάδες/ομάδες έτσι ώστε τα δείγματα εντός μιας συστάδας να είναι πιο κοντά το ένα με το άλλο παρά με μιας άλλης συστάδας, σύμφωνα με τον υπολογισμό κάποιας μετρικής που ορίζει και τις διάφορες παραλλαγές υλοποίησης των αλγορίθμων συσταδοποίησης. Για την τελική απόφαση τα δείγματα που βρίσκονται μακριά απ' όλες τις συστάδες θεωρούνται περιπτώσεις επίθεσης. Παραδείγματα τεχνικών συσταδοποίησης είναι οι αλγόριθμοι διχοτόμησης (partitioning algorithm), ιεραρχικοί (hierarchical-based), πυκνότητας (density-based) και πλέγματος (grid-based) [10].

Οι *κανόνες συσχέτισης* (association rules) ανήκουν σε μια μεγάλη ομάδα αλγορίθμων που σκοπό έχουν την παραγωγή κανόνων (*rule-based*) για την δημιουργία κάποιας πρόβλεψης [10]. Τα ruled-based συστήματα, σημαντικότερη υποκατηγορία των οποίων είναι τα *Έμπειρα Συστήματα* (Expert Systems) τα οποία χαρακτηρίζονται απ' το περιορισμένο πεδίο εφαρμογής τους και την ανάγκη ειδικών πεδίου για την ανάπτυξη και συντήρηση τους, μπορεί να είναι και supervised και unsupervised. Οι κανόνες συσχέτισης παρόλο που ανήκουν στα ruled-based

συστήματα που παραδοσιακά θεωρούνται διαφορετικός κλάδος της Τεχνητής Νοημοσύνης απ' την Μηχανική Μάθηση, στον βαθμό που οι κανόνες παράγονται μέσα από αυτοματοποιημένες διαδικασίες και αλγορίθμους μηχανικής μάθησης που βασίζονται στην αναγνώριση μοτίβων σε δεδομένα, παρά με την ενσωμάτωση γνώσης κάποιου ειδικού με την βοήθεια ειδικών προγραμμάτων-κελυφών ή ενός προγραμματιστή, μπορούν να θεωρηθούν ως μια ενδιάμεση κατάσταση. Τα ruled-based συστήματα εν γένει είναι απλούστερα και πιο ερμηνεύσιμα μοντέλα απ' τις υπόλοιπες τεχνικές που χρησιμοποιούνται στο πεδίο της Μηχανικής Μάθησης. Ένας κανόνας είναι μία απλή δήλωση τύπου IF-THEN που αποτελείται από μια κατάσταση (ζευγάρι χαρακτηριστικού-τιμής) ή ένωση καταστάσεων και μια πρόβλεψη. Ένα μοντέλο τυπικά περιλαμβάνει πολλούς τέτοιους κανόνες [25].

Στους κανόνες συσχέτισης κάθε ζευγάρι χαρακτηριστικού-τιμής θεωρείται ως ένα αντικείμενο και ο αλγόριθμος προσπαθεί να δημιουργήσει ένα σύνολο αντικειμένων από μια μεγάλη βάση δεδομένων όπου αυτά παρουσιάζονται συχνά. Σκοπός είναι η παραγωγή «καλών» κανόνων συσχέτισης για πολλαπλά χαρακτηριστικά. Ένα παράδειγμα χρήσης τους είναι η πιθανότητα αγοράς κάποιου προϊόντος στο supermarket δοθέντος ενός καλαθιού που ήδη περιλαμβάνει κάποια άλλα προϊόντα. Τα χαρακτηριστικότερα παραδείγματα κλιμακώσιμων αλγορίθμων προς αυτό τον σκοπό είναι οι *FP-Growth* και *Apriori* καθώς και οι παραλλαγές τους [10].

2.1.2.2 Ταξινόμηση με βάση την πολυπλοκότητα των μοντέλων

Οι τεχνικές της Μηχανικής Μάθησης μπορούν επίσης να χωριστούν σε δύο κατηγορίες, το *ρηχό* και το *βαθύ* μοντέλο, ανάλογα με την ικανότητα αναπαράστασης κάποιου κόσμου ως ενός συνόλου περίπλοκων εννοιών που ορίζονται απ' την σύνθεση απλούστερων εννοιών, και αφηρημένων αναπαραστάσεων υπολογισμένων από λιγότερο αφηρημένες αναπαραστάσεις. Στο ρηχό μοντέλο βασικά συγκαταλέγονται όλες οι κλασικές μέθοδοι που χρησιμοποιούνταν πριν την ευρεία υιοθέτηση τοπολογιών τεχνητών νευρωνικών δικτύων πολλαπλών «κρυφών» επιπέδων. Το βασικό γνώρισμα της πιο συχνής χρήσης μεθόδων βαθιάς μάθησης είναι ο υπολογισμός ιεραρχικών χαρακτηριστικών ή αναπαραστάσεων των παρατηρούμενων δεδομένων, όπου τα χαρακτηριστικά υψηλού επιπέδου (ή γινόμενα) εξάγονται από χαρακτηριστικά χαμηλότερων επιπέδων. Πρέπει να σημειωθεί ότι παρόλο της χρησιμότητας αυτού του διαχωρισμού, δεν υπάρχει μια απόλυτη συνθήκη για το βάθος των νευρωνικών δικτύων ώστε αυτά να χαρακτηριστούν ως ένα βαθύ μοντέλο [26] [27]. Οι τεχνικές βαθιάς μάθησης εκτός της παραδοσιακής επιβλεπόμενης εκπαίδευσης, μπορούν επίσης να

αξιοποιηθούν στην μάθηση μιας καλή αναπαράστασης των χαρακτηριστικών από ένα μεγάλο πλήθος unlabelled δεδομένων, οπότε το μοντέλο μπορεί να προ-εκπαιδευτεί με έναν εντελώς μη επιβλεπόμενο τρόπο. Στην συνέχεια τα λίγα υπάρχοντα labelled δεδομένα μπορούν να χρησιμοποιηθούν απλά και μόνο για να βελτιστοποιηθούν κάποιες παράμετροι για έναν συγκεκριμένο σκοπό επιβλεπόμενης ταξινόμησης. Με άλλα λόγια, γίνεται χρήση ημι-επιβλεπόμενης μάθησης.

Στην συνέχεια ακολουθεί μια ταξινόμηση των πιο συχνών τεχνικών μηχανικής μάθησης ως προς το ρηχό ή βαθύ μοντέλο μάθησης που εκπροσωπούν, που λειτουργεί και ως ευκαιρία για μια σύντομη αναφορά των πιο συχνών αλγορίθμων που χρησιμοποιούνται στα IDS με συγκεκριμένα παραδείγματα, καθώς και των διαφορών που παρουσιάζονται μεταξύ του ρηχού και του βαθιού μοντέλου μάθησης [28]. Η ανάλυση των μεθόδων ξεφεύγει απ’ το πλαίσιο της εργασίας αλλά για την καλύτερη κατανόηση παρατίθεται μια σύντομη περιγραφή στο Παράρτημα.

2.1.2.2.1 Ρηχό Μοντέλο Μάθησης

Στις πιο σημαντικές τεχνικές της επιβλεπόμενης μάθησης περιλαμβάνονται:

- *Ρηχά Τεχνητά Νευρωνικά Δίκτυα* (Artificial Neural Networks ή ANN)
- *Μηχανές Υποστήριξης Διανυσμάτων* (Support Vector Machines ή SVM)
- *Κ-Κοντινότεροι-Γείτονες* (K-Nearest-Neighbours/KNN)
- *Αφελή Μπαγεσιανά Δίκτυα* (Naive Bayesian Networks)
- *Λογιστική Παλινδρόμηση* (Logistic Regression)
- *Δένδρα Απόφασης* (Decision Trees)
- *Εξελικτικός Υπολογισμός* (Evolutionary Computation)

Στις τεχνικές της μη επιβλεπόμενης μάθησης συγκαταλέγονται:

- *Κ-μέσων* (K-Means)
- *Gaussian Mixture Model algorithm*
- *DBSCAN* και άλλοι αλγόριθμοι συσταδοποίησης
- *Κανόνες Συσχέτισης και Ασαφείς Κανόνες Συσχέτισης*

2.1.2.2.2 Βαθύ Μοντέλο Μάθησης

Στις μεθόδους επιβλεπόμενης μάθησης είναι:

- *Δίκτυα Βαθιάς Πίστης* (Deep Belief Networks ή DBN)

- *Perceptron Πολλαπλών Επιπέδων* (Multi-Layer Perceptron's ή MLP)

Στις μεθόδους Μη-επιβλεπόμενης μάθησης είναι:

- *Παραγωγικά Αντιπαραθετικά Δίκτυα* (Generative Adversarial Networks ή GAN)
- *Αυτοοργανώμενοι Χάρτες* (Self-organizing map ή SOM)
- *Περιοριστικές Μηχανές Boltzmann* (Restricted Boltzmann Machines ή RBM)
- *Αυτόματοι Κωδικοποιητές* (Autoencoders)

2.1.2.2.3 Διαφορές ρηχών και βαθιών μοντέλων μάθησης

Η βαθιά μάθηση έχει αποκτήσει μεγάλο ερευνητικό ενδιαφέρον εξαιτίας των παρακάτω πλεονεκτημάτων που παρουσιάζει:

- Τα βαθιά μοντέλα μπορούν να δεχθούν ως είσοδο ανεπεξέργαστα δεδομένα και να κάνουν *εξαγωγή χαρακτηριστικών* από μόνα τους, εν αντιθέσει με τα ρηχά που απαιτούν και προϋποθέτουν ανθρώπινη παρέμβαση, γνώση/ειδικούς πεδίου και χρόνο για ρητή εξαγωγή και επιλογή χαρακτηριστικών. Ταυτόχρονα, με την μικρότερη ανθρώπινη παρουσία, ελαχιστοποιείται και ο κίνδυνος ανθρώπινου λάθους.
- Τα βαθιά μοντέλα όντας πιο περίπλοκα, έχουν μεγαλύτερη ικανότητα *προσαρμογής* (fitting).

Ωστόσο παρουσιάζουν και ορισμένα μειονεκτήματα, μερικά απ' τα οποία έχουν αντιμετωπιστεί σε σημαντικά βαθμό τα τελευταία χρόνια:

- Ο *χρόνος* εκπαίδευσης και δοκιμής είναι πολύ μεγαλύτερος στο deep learning. Παραδοσιακά ένα εμπόδιο στην χρήση της βαθιάς μάθησης, η πρόοδος στον τομέα του υλικού και η ανάπτυξη παράλληλων μεθόδων και αρχιτεκτονικών, όπως η χρήση καρτών γραφικών για τον υπολογισμό πράξεων μεταξύ πινάκων, χρήση clusters υπολογιστών με TPU/FPGA και άλλα, θεωρείται ότι έχει οδηγήσει σε μια επιτάχυνση της τάξης των 100 φορών σε σχέση με τους συμβατικούς επεξεργαστές γενικού σκοπού (CPUs).
- Τα βαθιά μοντέλα είναι επιρρεπή σε *υπερπροσαρμογή* (overfitting) και απαιτούν περισσότερα δεδομένα. Υπερπροσαρμογή είναι η απομνημόνευση των δεδομένων εκπαίδευσης και η αδυναμία του μοντέλου να διαχωρίσει τον θόρυβο απ' τα ουσιώδη μοτίβα που θέλουμε να μάθει, οδηγώντας σε αδυναμία γενίκευσης και την συνύπαρξη μιας παραπλανητικά καλής επίδοσης στα δεδομένα εκπαίδευσης και μιας πολύ χαμηλής αποτελεσματικότητας σε πραγματικές συνθήκες.

- Ο αριθμός των παραμέτρων και *υπερπαραμέτρων* (hyperparameters), δηλαδή παραμέτρων/μεταβλητών που επιλέγονται χειροκίνητα πριν το testing και δεν υπολογίζονται αυτόματα απ' το μοντέλο, είναι ομοίως πολύ μεγαλύτερος.
- Τα βαθιά μοντέλα είναι μαύρα κουτιά και δεν παρέχουν ίδιου βαθμού *ερμηνευτικότητα* σε σύγκριση με κάποια ρηχά μοντέλα (π.χ. δένδρα απόφασης)

2.1.2.2.4 Υβριδικά Μοντέλα και Συνολικά Μοντέλα

Τα τελευταία χρόνια γίνεται όλο και μεγαλύτερη χρήση *υβριδικών* μοντέλων. Με τον όρο υβριδικά μοντέλα αναφερόμαστε κατά κύριο λόγο σε δύο πράγματα. Το πρώτο αφορά τον γενικότερο συνδυασμό ρηχών και βαθιών μοντέλων μάθησης όπου υπάρχουν διάφορες αρχιτεκτονικές υλοποίησης και συνήθως αφορά την σειριακή παράταση των μοντέλων όπου σε κάθε στάδιο η έξοδος του πρώτου μοντέλου παίρνει τον ρόλο της εισόδου του δεύτερου και ούτως καθεξής [29]. Παραδείγματος χάριν ένα υβριδικό μοντέλο μπορεί να χρησιμοποιεί ένα ρηχό μοντέλο μάθησης για την εξαγωγή χαρακτηριστικών και την επακόλουθη χρήση ενός μοντέλου βαθιάς μάθησης για την τελική καταγραφή της σχέσης μεταξύ των χαρακτηριστικών. Ένας άλλος τρόπος εργασίας είναι η αντίστροφη διαδικασία με το νευρωνικό δίκτυο να παίζει τον ρόλο του εξαγωγέα χαρακτηριστικών και παραδοσιακές τεχνικές μηχανικής μάθησης όπως μηχανές υποστήριξης διανυσμάτων τον ρόλο του τελικού ταξινομητή που θα εκμεταλλεύεται την μείωση της διαστατικότητας των δεδομένων. Ένας τρίτος τρόπος σκέψης είναι ο συνδυασμός χρήσης ρηχών μοντέλων για αναγνώριση μοτίβων σε πραγματικό χρόνο και η χρήση βαθιών μοντέλων μάθησης για offline ανάλυση πολλών δεδομένων. Πολλά μοντέλα ημι-επιβλεπόμενης μάθησης είναι υβριδικά μοντέλα.

Ο δεύτερος τρόπος με τον οποίο παρουσιάζεται στην βιβλιογραφία ο όρος υβριδικό μοντέλο αφορά πιο συγκεκριμένα τον τομέα ανάπτυξης IDS και τον συνδυασμό διαφορετικών μεθόδων αναγνώρισης απειλής όπως anomaly-based και signature/rule-based, όχι αναγκαστικά υλοποιημένες με μηχανική μάθηση, που αναλύονται στην επόμενη παράγραφο [30].

Μερικά άλλα παραδείγματα στη βιβλιογραφία αποτελούν συνδυασμοί πολύ διαφορετικών τεχνικών (λ.χ. νευρωνικά δίκτυα και ασαφή λογική), τεχνικές συσταδοποίησης ως ένα πρώτο βήμα προεπεξεργασίας δεδομένων πριν τον τελικό ταξινομητή, ή ακόμα και έναν βελτιστοποιητή παραμέτρων για το δεύτερο υποσύστημα. Τα υβριδικά μοντέλα συναποτελούμενα από βαθιά μοντέλα έχουν το αρνητικό ότι δεν μπορούμε να επέμβουμε στην εσωτερική αναπαράσταση των δεδομένων στα κρυφά στρώματα του νευρωνικού δικτύου. Η έρευνα προσανατολίζεται στην βέλτιστη επιλογή μοντέλων (συχνότερα ταξινομητών) και τον

τρόπο που αυτά θα συνδυαστούν. Κάποιες νεότερες μελέτες υλοποίησης υβριδικών μοντέλων για την ανάπτυξη IDS υιοθετούν νευρωνικά δίκτυα μιας κλάσης (One-Class Neural Networks, εν συντομία OC-NNs). Τα μοντέλα μιας κλάσης θεωρούνται μοντέλα χωρίς επίβλεψη επειδή βασίζονται στην υπόθεση ότι η μεγάλη πλειοψηφία των δειγμάτων είναι «κανονική» και εκπαιδεύονται πάνω στα κανονικά δείγματα χωρίς καμία πληροφορία για τις ακραίες ή ασυνήθιστες τιμές, επομένως χωρίς ετικέτες. Σχετίζονται στενά με τις μεθόδους αναγνώρισης ανωμαλίας που περιγράφονται σε επόμενη παράγραφο στο πλαίσιο των IDS [31].

Εκτός των υβριδικών μοντέλων, υπάρχει και το *Συνολικό* Μοντέλο (Ensemble Method). Το Συνολικό μοντέλο μπορεί να χωριστεί σε τρεις κατηγορίες: *Bagging*, *Boosting* και *Stacking* [32].

Το bagging αναφέρεται σε μια διαδικασία παράλληλης αρχιτεκτονικής και ανεξάρτητης μάθησης μεταξύ ίδιου τύπου αδύναμων ταξινομητών και στην συνέχεια μιας ψηφοφορίας για την τελική απόφαση. Το boosting απ’ την άλλη πλευρά είναι μια ακολουθιακή τεχνική οργάνωσης ίδιου τύπου ταξινομητών όπου το κάθε μοντέλο εξαρτάται απ’ το προηγούμενο του. Τέλος, το stacking αφορά ετερογενή ταξινομητές που εκπαιδεύονται παράλληλα και τα αποτελέσματα των οποίων εισέρχονται ως είσοδο σε έναν τελικό ταξινομητή.

2.1.2.3 Ταξινόμηση με βάση τον τρόπο λειτουργίας των μοντέλων

2.1.2.3.1 Διευκρινιστικά μοντέλα

Στην επιστήμη της Στατιστικής ως *διευκρινιστικά μοντέλα* (discriminative models) ορίζονται όλα τα μοντέλα τα οποία έχουν στόχο να διαχωρίσουν παραδείγματα σε κατηγορίες μοντελοποιώντας την σχέση ανάμεσα στις εξαρτημένες και ανεξάρτητες μεταβλητές, αρχικά υπολογίζοντας απευθείας μέσα απ’ τα δεδομένα τις εκ των υστέρων πιθανότητες (posterior probabilities) $p(C_k, x)$, δηλαδή τις δεσμευμένες πιθανότητες δοθέντος ενός δείγματος με χαρακτηριστικά x να ανήκει στην κλάση C_k, και ύστερα επιλέγοντας την κλάση με την μεγαλύτερη εκ των υστέρων πιθανότητα [33]. Στην Μηχανική Μάθηση τα μοντέλα που χρησιμοποιούνται στην πράξη μοντελοποιούν ή προσεγγίζουν με κάποιον τρόπο την posterior, χωρίς απαραίτητα να την υπολογίζουν αναλυτικά. Έτσι, οι μέθοδοι που χρησιμοποιούν τα διευκρινιστικά μοντέλα ποικίλουν από την δημιουργία κανόνων (Δένδρα Απόφασης), προσαρμογή καμπύλης (Λογιστική Παλινδρόμηση), εύρεση υπερεπιπέδου (Μηχανές Υποστήριξης Διανυσμάτων), βέλτιστα βάρη (Perceptron Πολλαπλών Επιπέδων) και άλλα.

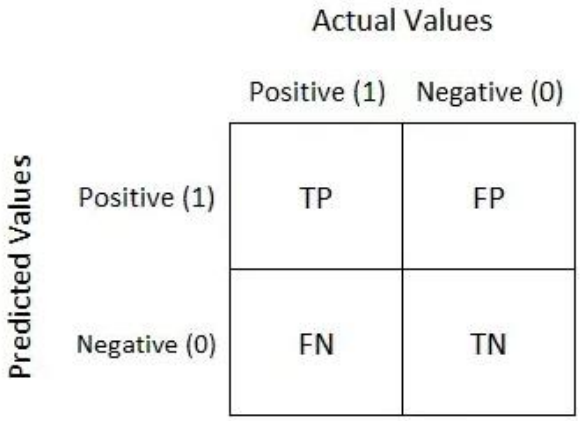
2.1.2.3.2 Αναγεννητικά μοντέλα

Αντίθετα στα *αναγεννητικά μοντέλα* (generative models) ο βασικός μηχανισμός λειτουργίας τους είναι πιο σύνθετος και αφορά τον υπολογισμό των από κοινού κατανομών πιθανότητας P(X, Y) ή P(Y, X), όπου X είναι τα χαρακτηριστικά και Y οι κλάσεις, ή απλούστερα της P(X) όταν δεν υπάρχουν κλάσεις. Με τον τρόπο αυτόν τα αναγεννητικά μοντέλα μαθαίνουν την κατανομή των χαρακτηριστικών που εμφανίζονται για κάθε κλάση. Με αυτή την πληροφορία γνωστή, τα αναγεννητικά μοντέλα μπορούν να επεκταθούν με δύο τρόπους για να επιτελέσουν διαφορετικούς σκοπούς [33]:

- Να υπολογίσουν την εκ των υστέρων πιθανότητα $P(Y_k \mid X)$ ώστε το παράδειγμα με χαρακτηριστικά $X = (x_1,\ldots, x_n)$ να ανήκει στην κλάση Y_k και ύστερα να εισάγουν κάποια μέθοδο επιλογής όπως π.χ. συνηθέστερα να επιλέξουν την κλάση με την μεγαλύτερη πιθανότητα, δηλαδή να λειτουργήσουν ως ταξινομητές με παρόμοιο τρόπο με τα Διευκρινιστικά μοντέλα. Ο υπολογισμός γίνεται μέσω του κανόνα του Bayes που συνδέει την εκ των υστέρων (δεσμευμένη) πιθανότητα με την από κοινού πιθανότητα που υπολόγισαν τα μοντέλα, είτε άμεσα ως P(X, Y), είτε έμμεσα υπολογίζοντας το γινόμενο P(X | Y) * P(X). Η οριακή πιθανότητα στον παρανομαστή μπορεί να αγνοηθεί αφού τα χαρακτηριστικά είναι γνωστά σε κάθε παράδειγμα οπότε λειτουργούν ως ένας παράγοντας κανονικοποίησης και σε κάθε περίπτωση είναι ίδια για τον υπολογισμό της εκ των υστέρων πιθανότητας για κάθε κλάση. Εναλλακτικά, η οριακή πιθανότητα μπορεί να υπολογιστεί ως το άθροισμα των από κοινού κατανομών για όλες τις κλάσεις και να δώσει ως απάντηση πόσο πιθανή είναι η εμφάνιση ενός παραδείγματος, λειτουργώντας στην πράξη ως ανιχνευτής ανωμαλιών. Η διευκρινιστική λειτουργία των αναγεννητικών μοντέλων έχει το αρνητικό ότι απαιτεί περισσότερα δεδομένα απ' ότι τα διευκρινιστικά που ειδικεύονται στην ταξινόμηση. Ένα παράδειγμα χρήσης ταξινόμησης αναγεννητικού μοντέλου είναι ο απλός ταξινομητής Bayes. Δεν υπολογίζει την πραγματική από κοινού πιθανότητα αλλά κάνει μια απλουστετική υπόθεση ανεξαρτησίας των χαρακτηριστικών. Παρόλο αυτά λειτουργεί αρκετά καλά στην πράξη.
- Μέσω δειγματοληψίας των κατανομών των χαρακτηριστικών που έμαθαν για κάθε κλάση, να δημιουργήσουν νέα συνθετικά δεδομένα, δικαιολογώντας την ονομασία τους. Παράδειγμα τέτοιας χρήσης είναι τα μοντέλα τύπου GAN σε εφαρμογές εικόνας.

2.1.3 Μετρικές Αξιολόγησης

Υπάρχουν πολλές μετρικές με διάφορα ονόματα για την αξιολόγηση των συστημάτων ταξινόμησης. Κεντρικό ρόλο στην διαδικασία αξιολόγησης παίζει ο λεγόμενος *πίνακας σύγχυσης* (confusion matrix), ή αλλιώς πίνακας συνάφειας ή σφάλματος [34] [35] [36], που χρησιμοποιείται συνήθως για την περίπτωση των supervised μεθόδων (σε περιπτώσεις unsupervised μεθόδων αποκαλείται συνήθως πίνακας ταιριάσματος ή matching matrix), και για την περίπτωση της δυαδικής ταξινόμησης έχει την παρακάτω μορφή (Εικόνα 2):



Εικόνα 2: Πίνακας Σύγχυσης[6]

Οι μεταβλητές που παρουσιάζονται στις τιμές του είναι το πλήθος των

- Αληθώς Θετικών (True Positive, TP) ή αλλιώς το πλήθος των δειγμάτων ελέγχου που ο ταξινομητής προβλέπει σωστά ως θετικά.
- Αληθώς Ψευδών (True Negative, TN) ή αλλιώς το πλήθος που ο ταξινομητής προβλέπει σωστά ως αρνητικά
- Ψευδώς Θετικών (False Positive, FP) ή αλλιώς το πλήθος που προβλέπει λανθασμένα ως θετικά
- Ψευδώς Αρνητικών (False Negative, FN) ή αλλιώς το πλήθος που προβλέπει λανθασμένα ως αρνητικά

Στην περίπτωση της δυαδικής ταξινόμησης (νορμάλ ή επίθεση) οι πιο σημαντικές μετρικές που παράγονται και χρησιμοποιούνται απ' τον πίνακα σύγχυσης είναι οι παρακάτω:

[6] https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Ακρίβεια (Accuracy) ή Ποσοστό Σωστών (Proportion Correct) είναι το ποσοστό των σωστών προβλέψεων του αλγορίθμου για κάθε κατηγορία και ορίζεται απ' τον λόγο

$$\frac{TP + TN}{TP + TN + FP + FN}$$

Η συγκεκριμένη μετρική έχει ειδική σημασία γιατί χαρακτηρίζει την συνολική απόδοση του μοντέλου ανεξαρτήτως της κάθε κατηγορίας/κλάσης. Παρόλο αυτά, το μειονέκτημα της είναι ότι αν τα δεδομένα δεν είναι ισορροπημένα, δεν δείχνει την πραγματική εικόνα της ικανότητας του μοντέλου να μαθαίνει και καθίσταται αναγκαία η εισαγωγή άλλων πιο αναλυτικών μετρικών. Έτσι, πολλές φορές στη πράξη η χρησιμότητα της είναι περιορισμένη.

Προγνωστική Αξία (Predictive Value) είναι η σχετική ακρίβεια του μοντέλου για κάθε κατηγορία ξεχωριστά.

Στην περίπτωση της κλάσης των θετικών σε μια δυαδική ταξινόμηση, η *Θετική Προγνωστική Αξία* (Positive Predictive Value) ή Ακρίβεια (*Precision*), ορίζεται απ' τον λόγο των σωστών θετικών προβλέψεων προς το σύνολο όλων των θετικών προβλέψεων, συμπεριλαμβανομένων αυτών που προβλέφθηκαν λανθασμένα θετικά, δηλαδή

$$\frac{TP}{TP + FP}$$

Αντίστοιχα η περίπτωση των αρνητικών προβλέψεων ονομάζεται *Αρνητική Προγνωστική Αξία* (Negative Predictive Value) και ορίζεται απ' τον λόγο

$$\frac{TN}{TN + FN}$$

Ευαισθησία (Sensitivity) ή *Ανάκληση* (Recall) ή *Δείκτης Αληθώς Θετικών* (True Positive Rate) ή *Ανίχνευση Πιθανότητας* (Probability Detection) ή *Δείκτης Ανίχνευσης* (Detection Rate) είναι το σχετικό με κάθε κατηγορία ποσοστό θετικών προβλέψεων προς το σύνολο των δειγμάτων που ανήκουν στην θετική κατηγορία και θα έπρεπε να είχαν προβλεφθεί ως θετικά, δηλαδή.

$$\frac{TP}{TP + FN}$$

Εξειδίκευση (Specificity) ή *TN Rate* ονομάζεται η μετρική της Ανάκλησης για την αρνητική κλάση και ορίζεται απ' τον λόγο

$$\frac{TN}{TN + FP}$$

Δείκτης Ψευδώς Θετικών ή *FAR* (False Alarm Rate) ή *FP Rate* ή *Fall-out* ορίζεται ως το συμπληρωματικό της εξειδίκευσης, δηλαδή $FAR = 1 - Specificity$ ή σε μορφή κλάσματος ως

$$\frac{FP}{TN + FP}$$

Είναι πολύ σημαντική μετρική και συνήθως τίθεται κάποιο όριο (ιδανικά <1%) στην επιλογή του τελικού ταξινομητή.

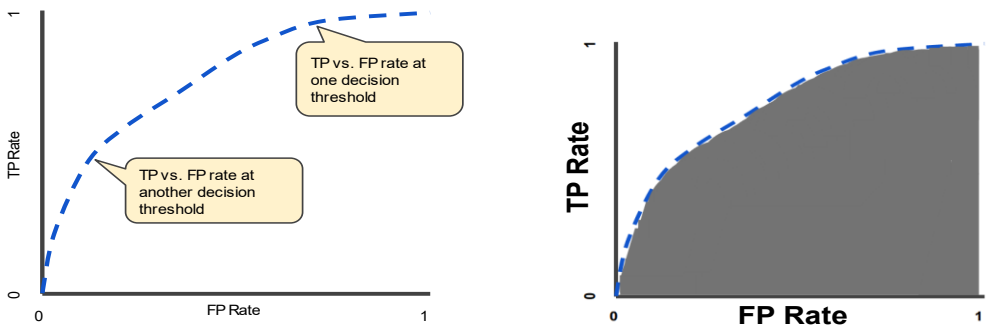
F-Score ή *F-Measure* παραδοσιακά ορίζεται ως ο αρμονικός μέσος όρος των μετρικών Recall και Precision δηλαδή:

$$2 \cdot \frac{Recall \cdot Precision}{Recall + Precision}$$

Στην περίπτωση των μεθόδων χωρίς επίβλεψη οι μετρικές μπορούν να χωριστούν σε [37]:

- Εσωτερικές, που χρησιμοποιούνται στα δεδομένα που συσταδοποιήθηκαν (clustered) και δεν χρησιμοποιούν ετικέτες, με παραδείγματα όπως λ.χ. απόσταση μεταξύ συστάδων, απόσταση μελών εντός συστάδας, δείκτης Dunn κτλ.
- Εξωτερικές, που χρησιμοποιούνται για δεδομένα όπου υπάρχουν ετικέτες, παρόλο που η μέθοδος δεν τις χρησιμοποιεί.

Εκτός αυτών, υπάρχουν και άλλες, γραφικές, μέθοδοι όπως οι καμπύλες Receiver Operating Characteristics Curve (*ROC*), Reliability Curve ή Bathtub Curve ή Failure Rate Curve και η μετρική Area Under the Curve (*AUC*) , παραδείγματα των οποίων φαίνονται στην Εικόνα 3.



Εικόνα 3: Καμπύλες ROC και AUC[7]

[7] https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

2.2 Πληροφορίες για IDS

2.2.1 Γενικές Πληροφορίες

Οι γενικές αρχές σχεδιασμού ενός IDS με χρήση μηχανικής μάθησης είναι απλές [10]:

1. *Συλλογή* πληροφοριών του δικτύου υπό εξέταση μέσω κάποιου λογισμικού καταγραφής κίνησης.
2. *Επιλογή* ή/και *εξαγωγή* ή μηχανική χαρακτηριστικών
3. *Ανάλυση* των δεδομένων με βάση κάποιον αλγόριθμο ML και απόφαση για το αν πρόκειται για επίθεση ή όχι.
4. *Ειδοποίηση* του διαχειριστή σχετικά με την επίθεση και προαιρετική ενεργητική προστασία του δικτύου.

Τα IDS μπορούν αφενός να χωριστούν με βάση την *πηγή της πληροφορίας* που χρησιμοποιούν και αφετέρου με βάση την *μεθοδολογία ανίχνευσης* [38].

Ως προς τον δεύτερο τρόπο ταξινόμησης των IDS, δηλαδή με βάση την μεθοδολογία ανίχνευσης, τα IDS μπορούν να χωριστούν σε συστήματα *ανίχνευσης ανωμαλίας* (anomaly detection) ή ακραίων τιμών (outlier detection) ή καινούργιων επιθέσεων (novelty detection) καθώς και *κακής χρήσης* (misuse detection) ή ανίχνευσης υπογραφών (signature detection) ή γνώσεως (knowledge-based) και *υβριδικά* [38].

Τα συστήματα ανίχνευσης ανωμαλίας ή συστήματα συμπεριφοράς (behavior-based) λειτουργούν ορίζοντας μία κανονική λειτουργία και την θεώρηση οποιαδήποτε μεγάλης απόκλιση απ' αυτή ως ανωμαλία και ενδεχόμενη απόπειρα επίθεσης [3]. Με βάση τον τύπο τους οι ανωμαλίες, και κατ' επέκταση τα συστήματα, μπορούν να διαχωριστούν σε *ανωμαλία σημείου*, *συμφραζόμενη ανωμαλία* και *συλλογική ανωμαλία* [39] [40]. Το πλεονέκτημα της συγκεκριμένης μεθόδου είναι ότι μπορεί θεωρητικά να χρησιμοποιηθεί και για την αναγνώριση αγνώστων και νέων επιθέσεων (novel ή zero-day attacks) και τα βασικά της μειονεκτήματα είναι τα πολλά false positives και η δυσκολία καθορισμού του συνόρου ανάμεσα στην κανονική λειτουργία και την αφύσικη/ασυνήθιστη [39].

Σε αντίθεση με τα συστήματα ανίχνευσης ανωμαλίας, τα συστήματα ανίχνευσης κακής χρήσης προϋποθέτουν την δημιουργία μιας βάσης δεδομένων με γνωστές υπογραφές [41], δηλαδή μοναδικές αναγνωστικές ακολουθίες bytes προγραμμάτων ή αρχείων που προκύπτουν από κρυπτογραφικές συναρτήσεις κατακερματισμού (hash functions), και βάσει αυτών με

pattern matching όπως κανονικές εκφράσεις (regular expressions) κρίνεται αν υπάρχει ή όχι επίθεση, και στην συνέχεια με μια δήλωση τύπου IF-THEN ειδοποιούν την διαχειριστή του συστήματος για την παρουσία κακόβουλης κίνησης δεδομένων, φιλοσοφία που υιοθετούν τα προγράμματα Antivirus. Τα πλεονεκτήματα της είναι η υψηλή ακρίβεια εντοπισμού επιθέσεων που περιλαμβάνονται εντός της βάσης δεδομένων και το χαμηλό false positive rate. Όμως το μεγάλο της μειονέκτημα είναι ότι δεν μπορεί να χρησιμοποιηθεί για νέες επιθέσεις που δεν έχουν καταχωρηθεί προηγουμένως στην βάση δεδομένων με αποτέλεσμα να καθίσταται απαραίτητη η συνεχής ενημέρωση της [41].

Η τρίτη κατηγορία των υβριδικών συστημάτων ανίχνευσης εισβολής προσπαθεί να συγκεράσει τις δύο προηγούμενες εκμεταλλευόμενη τα μοναδικά θετικά στοιχεία της κάθε μιας και περιορίζοντας τα αρνητικά τους. Ο συνδυασμός και των δύο μεθόδων παράγει καλύτερα αποτελέσματα.

Για λόγους πληρότητας σημειώνεται ότι για την ανίχνευση ανωμαλίας μπορούν να χρησιμοποιηθούν εκτός της Μηχανικής Μάθησης και το *Στατιστικό Μοντέλο* καθώς και οι *Χρονοσειρές* (Time Series) ενώ για την ανίχνευση κακής χρήσης η *Ανίχνευση μοτίβων* (Pattern Matching), τα *Έμπειρα Συστήματα*, και η *Μηχανή πεπερασμένων καταστάσεων* (Finite State Machines) [41]. Ένα παράδειγμα χρήσης στατιστικού μοντέλου είναι τα *Κρυφά Μαρκοβιανά Μοντέλα* (Hidden Markov Models ή HMM). Μία Αλυσίδα ή Διαδικασία Μαρκόβ ορίζεται ως Καταστάσεις που ενώνονται με ακμές που έχουν κάποια πιθανότητα. Το Κρυφό Μαρκοβιανό Μοντέλο είναι αλυσίδα Μαρκόβ, δηλαδή γράφος όπου κάθε επόμενη κατάσταση εξαρτάται μόνο απ’ την τωρινή, όπου οι εσωτερικές καταστάσεις του συστήματος είναι κρυφές. Οι καταστάσεις είναι οι αόρατες συνθήκες που μοντελοποιούνται ενώ επιπρόσθετα υπάρχουν άλλες ορατές μεταβλητές/έξοδοι που ενώνονται με ακμές πιθανότητας με τις κρυφές καταστάσεις. Τέτοιες καταστάσεις θα μπορούσε να είναι η φύση μιας δικτυακής σύνδεσης και οι έξοδοι τιμές χαρακτηριστικών λ.χ. στο πλαίσιο των IDS. Κάθε κρυφή κατάσταση αντιστοιχίζεται με μια κατανομή πιθανοτήτων εξόδων και το μοντέλο αφενός εκπαιδεύεται ώστε να μάθει τις κατανομές εξόδων που αντιστοιχίζονται με τις κρυφές καταστάσεις καθώς και τις πιθανότητες των ακμών, βάσει των παραδειγμάτων που του δόθηκαν, αφετέρου την εύρεση της πιο πιθανής ακολουθίας καταστάσεων δοθέντος μιας ακολουθίας εξόδων [42]. Οι Μηχανές Πεπερασμένων Καταστάσεων μοντελοποιούν την κίνηση ενός δικτύου ως ένα σύνολο καταστάσεων, που μπορεί να αντιπροσωπεύουν την κανονική λειτουργία, είδη επιθέσεων κτλ., και τις μεταξύ τους μεταβιβάσεις, οι οποίες υπαγορεύονται από κάποιες συνθήκες, οι οποίες μπορεί να πηγάζουν από κάποιο ruled-based σύστημα ή κάποιο μοντέλο

μηχανικής μάθησης εκπαιδευμένο πάνω σε κάποιο σύνολο δεδομένων με κάποιον αλγόριθμο επιβλεπόμενης ή μη επιβλεπόμενης μάθησης, ή ακόμα με κάποιον συνδυασμό ruled-based και machine learning-based συστημάτων.

2.2.2 Ταξινόμηση IDS

Ως προς τον *πρώτο τρόπο* κατηγοριοποίησης τους, σε ένα πρώτο επίπεδο μπορούν να διαχωριστούν σε host και network-based ανάλογα με το αν εξετάζουν πληροφορίες που πηγάζουν/αφορούν κάποιο δίκτυο ή κάποιον υπολογιστή (host) [41].

2.2.2.1 Host-based IDS

Ένα *host-based IDS* αναλύει γεγονότα όπως αναγνωριστικά διεργασιών (PID) και κλήσεις πυρήνα (kernel calls) του λειτουργικού συστήματος [39] [43]. Οι εισβολές είναι στην μορφή ανώμαλων υποακολουθιών των ιχνών και μεταφράζονται σε κακόβουλα λογισμικά, παράνομη συμπεριφορά και υπονόμευση πολιτικών ασφαλείας [3] [39] [43]. Τα host-based IDS πρέπει να μπορούν να διαχειριστούν την ακολουθιακή φύση των δεδομένων, είτε μοντελοποιώντας τα, είτε υπολογίζοντας μια μετρική ομοιότητας μεταξύ τους. Οι τεχνικές που αναλύουν τις ανωμαλίες χρησιμοποιώντας ίχνη από κλήσεις συστήματος μπορούν να χωριστούν σε τεχνικές μικρών ακολουθιών (short sequence-based) και τεχνικές συχνότητας (frequency-based) [39]. Στις τεχνικές μικρών ακολουθιών εξετάζονται ακολουθίες κλήσεων συστήματος από μια συγκεκριμένη διεργασία για την αναγνώριση ανωμαλιών που ξεφεύγουν απ' την κανονική λειτουργία αυτής της διεργασίας. Αντίθετα στις τεχνικές συχνότητας εξετάζεται η συχνότητα των παραγόμενων από μια διεργασία ή χρήστη κλήσεων συστήματος, και κατά πόσο αυτή η συχνότητα αποκλίνει απ' την κανονική συχνότητα αυτών των κλήσεων. Στην πρώτη κατηγορία περιλαμβάνονται μοντέλα HMMs και N-gram ενώ στην δεύτερη ruled-based συστήματα καθώς και Gaussian Mixture Models (GMMs) και Support Vector Data Description[8] (SVDD) [44].

Τα host-based IDS στην πράξη επεξεργάζονται πληροφορίες που πηγάζουν από *logs*, είτε του λειτουργικού συστήματος είτε εφαρμογών. Τα logs με την σειρά τους μπορούν να επεξεργαστούν με τρεις διαφορετικούς τρόπους χρησιμοποιώντας μηχανική μάθηση. Αυτοί είναι: *Υβριδικά συστήματα* κανόνων και μηχανικής μάθησης (Hybrid Rule and Machine

[8] Μοιάζει με το SVM αλλά αντίθετα με την εύρεση ενός υπερεπιπέδου, προσπαθεί μια βρει την μικρότερη δυνατή σφαίρα που περιλαμβάνει όλα τα παραδείγματα εκπαίδευσης.

Learning-based Systems), *Μηχανική Χαρακτηριστικών* (Feature Engineering) και *Ανάλυση Κειμένου* (Text Processing) [41].

2.2.2.1.1 Πληροφορίες για τις log-based τεχνικές

Η ανίχνευση εισβολής μέσα από αρχεία καταγραφής (log files) είναι χρήσιμη για SQL injections[9], U2R[10], και R2L[11] επιθέσεις επειδή τα logs εμπεριέχουν αναλυτικές πληροφορίες περιεχομένου. Τα αρχεία καταγραφής περιλαμβάνουν πληροφορίες σχετικές με χρήστες και χρονοσφραγίδες που μπορούν να χρησιμοποιηθούν για τον εντοπισμό του δράστη και του χρόνου επίθεσης. Επίσης επειδή καταγράφουν την συνολική διαδικασία εισβολής είναι εύκολα ερμηνεύσιμα. Το αρνητικό τους είναι κυρίως ότι απαιτείται γνώση ψηφιακής ασφάλειας για την κατανόηση και χρήση τους και δευτερευόντως ότι τα logs διαφορετικών εφαρμογών έχουν διαφορετικά formats [41].

Η πρώτη μέθοδος των υβριδικών μοντέλων χρησιμοποιεί την έξοδο rule-based συστημάτων όπως το Snort[12] ως είσοδο σε συστήματα μηχανικής μάθησης για την «τακτοποίηση» των πολλών εσφαλμένων συναγερμών (false alarms) που μαστίζουν τέτοια συστήματα. Η δεύτερη μέθοδος της μηχανικής ή εξαγωγής χαρακτηριστικών προσπαθεί να εξάγει χαρακτηριστικά μέσα απ' τα logs χρησιμοποιώντας γνώση πεδίου, και ύστερα τα εισάγει σε αλγορίθμους μηχανικής μάθησης. Μια διαδεδομένη μέθοδος εξαγωγής χαρακτηριστικών από logs κάνει χρήση του κυλιόμενου παραθύρου[13] (sliding window). Η τεχνική του sliding window αξιοποιεί την συμφραζόμενη πληροφορία που περιέχεται στα logs και ενδείκνυται για εφαρμογές streaming. Η τρίτη μέθοδος αντιμετωπίζει τα logs ως απλά αρχεία κειμένου, κάνει χρήση n-grams[14] και προσφέρει μεγαλύτερη ερμηνευτικότητα σε σχέση με μεθόδους εξαγωγής χαρακτηριστικών από logs [41].

---

[9] SQL Injections: Τύπος επίθεσης όπου ο δράστης εισάγει κακόβουλο κώδικα SQL σε φόρμες ή παραμέτρους URL με στόχο την είσοδο ή τον έλεγχο κάποιας βάσης δεδομένων.

[10] User to Root: Σύνολο από προγράμματα εκμετάλλευσης κενών ασφαλείας ή ευπάθειας (exploits) που σκοπό έχουν την είσοδο με δικαιώματα διαχειριστή (root) σε έναν υπολογιστή από έναν τοπικό χρήστη περιορισμένων δικαιωμάτων.

[11] Remote to User: Σύνολο από exploits με στόχο την είσοδο σε έναν τοπικό υπολογιστή μέσω της χρησιμοποίησης κάποιου δικτύου με τον οποίο συνδέεται ο υπολογιστής με τον επιτιθέμενο.

[12] https://www.snort.org/. Τα ruled-based συστήματα που χρησιμοποιούνται σε IDS όπως το Snort παράγουν μια σειρά κανόνων που μπορεί να αφορούν την κανονική λειτουργία ή/και κάποια επίθεση.

[13] Τεχνική που αναλύει μια ακολουθία δεδομένων, αρχικά χωρίζοντας τα σε ομάδες ίδιου μήκους (ή παράθυρα), και ύστερα μετακινώντας το παράθυρο κατά ένα δεδομένο την φορά.

[14] Ένα n-gram είναι μια συνεχή ακολουθία από n αντικείμενα (συνήθως λέξεις) από ένα δοσμένο κείμενο. Μοντέλα που κάνουν χρήση n-gram μπορούν να αναλύσουν τη συχνότητα και την ταυτόχρονη εμφάνιση λέξεων για να εξαγωγή μοτίβων και την παραγωγή προβλέψεων.

2.2.2.2 Network-based IDS

Τα *network-based IDS* ή *NIDS* απ' την άλλη πλευρά παρουσιάζουν μεγαλύτερη ποικιλία και αναλύουν γεγονότα σχετικά με το δίκτυο, όπως όγκος κίνησης δεδομένων, διευθύνσεις IP, πόρτες υπηρεσιών, χρήση πρωτοκόλλων κτλ. Στα anomaly-based NIDS οι εισβολές χαρακτηρίζονται ως ανώμαλες ή ασυνήθιστες μέσα από συνεχή παρακολούθηση και μοντελοποίηση της κανονικής συμπεριφοράς στα δίκτυα. Τα δεδομένα είναι υψηλής διαστατικότητας και συνήθως είναι ένα μείγμα κατηγορικών-συνεχών χαρακτηριστικών. Ως αποτέλεσμα, οι τεχνικές ανίχνευσης ανωμαλίας πρέπει να είναι υπολογιστικά αποδοτικές για να αντιμετωπίσουν αυτές τις μεγάλες εισόδους. Επιπρόσθετα, τα δεδομένα μεταδίδονται ζωντανά και επομένως απαιτούν on-line ανάλυση [39]. Υποστήριξη των NIDS με host-based ανάλυση δεδομένων δημιουργεί πιο στιβαρά IDS.

Τα NIDS με βάση τα δεδομένα μπορούν να πάρουν την μορφή *packet-based, flow-based* ή *session-based*. Οι επιλογές σχετικά με την πρώτη κατηγορία περιλαμβάνουν διάβασμα πακέτων (packet parsing) και ανάλυση ωφέλιμου πακέτου (payload analysis). Για την δεύτερη έχουμε μηχανική χαρακτηριστικών (feature engineering), «βαθιά» μάθηση (deep learning) και ομαδοποίηση κίνησης (traffic grouping). Τέλος για την τρίτη κατηγορία των δεδομένων σχετικά με συνόδους (session) υπάρχει η στατιστική ανάλυση χαρακτηριστικών και χαρακτηριστικά ακολουθίας. Όλες αυτές οι μέθοδοι μπορούν να υλοποιηθούν χρησιμοποιώντας μηχανική μάθηση [41].

2.2.2.2.1 Πληροφορίες για την δομή και την ανάλυση πακέτων

Τα πακέτα περιλαμβάνουν δυαδικά δεδομένα και αποτελούνται απ' την επικεφαλίδα (header), που περιέχει βασικά χαρακτηριστικά του αποστολέα, παραλήπτη και της σύνδεσης (IP, ports κτλ.), καθώς και του ωφέλιμου πακέτου (payload). Περιλαμβάνουν περιεχόμενα επικοινωνίας και ως εκ τούτου είναι χρήσιμα για U2L και R2L επιθέσεις. Επίσης λόγω της ύπαρξης των διευθύνσεων IP και των χρονοσφραγίδων είναι χρήσιμα για ακριβή εντοπισμό επιθέσεων και της ανίχνευσης σε πραγματικό χρόνο. Ένα αρνητικό τους είναι ότι μεμονωμένα πακέτα δεν αποκαλύπτουν πληροφορίες απ' τα συμφραζόμενα, κάτι που είναι χρήσιμο π.χ. για DDoS[15] επιθέσεις [41].

[15] Distributed Denial of Service: Είδος επίθεσης όπου ένα δίκτυο μολυσμένων υπολογιστών (botnet) προσπαθεί να πλημμυρίσει με δεδομένα έναν ή περισσότερους servers με στόχο να γίνουν μη διαθέσιμοι σε πραγματικούς χρήστες.

Ως προς τις τεχνικές ανάλυσης πακέτων μπορούν να ειπωθούν τα εξής: Βασίζονται στην πληροφορία που εμπεριέχεται στις επικεφαλίδες των πακέτων και χρησιμοποιούνται προγράμματα όπως το Wireshark[16] και το Bro[17] για το διάβασμα τους και την χρήση των τιμών των πιο σημαντικών πεδίων ως διανύσματα χαρακτηριστικών, δίνοντας ιδιαίτερη έμφαση στις κατηγορίες πρωτοκόλλων. Στην συνέχεια αυτά τα διανύσματα χαρακτηριστικών δίνονται ως είσοδος σε ρηχά μοντέλα μηχανικής μάθησης. Τέτοιες τεχνικές συχνά υιοθετούν μια προσέγγιση δύο σταδίων. Αρχικά τα δεδομένα συσταδοποιούνται χωρίς επίβλεψη με κάποιον αλγόριθμο συσταδοποίησης όπως K-means ή fuzzy C-means, και στην συνέχεια χρησιμοποιείται κάποιος αλγόριθμος επιβλεπόμενης μάθησης όπως SVM [41].

Απ' την άλλη πλευρά η ανάλυση ωφέλιμου φορτίου επικεντρώνεται στα δεδομένα των εφαρμογών. Επειδή δεν χρησιμοποιούν τις πληροφορίες των επικεφαλίδων, μπορούν να χρησιμοποιηθούν για πολλαπλά πρωτόκολλα. Η υλοποίηση αυτών των μεθόδων μπορεί να γίνει απευθείας με εφαρμογή βαθιών μοντέλων; κατά συνέπεια δεν απαιτείται προεπεξεργασία για εξαγωγή χαρακτηριστικών άρα έχουμε μικρότερο κόστος σε ανθρώπινη εργασία και παραβίαση ιδιωτικότητας [41].

2.2.2.2.2 Πληροφορίες για τις flow-based τεχνικές

Μία ροή (flow) ορίζεται ως ένα σύνολο πακέτων εντός ενός χρονικού διαστήματος/περιόδου μεταξύ δύο άκρων (endpoints) ενός δικτύου που μοιράζονται κάποια συγκεκριμένα χαρακτηριστικά που περιγράφουν την κίνηση, όπως διεύθυνση IP αποστολέα, διεύθυνση IP παραλήπτη, αριθμός πυλών (port numbers) αποστολέα και παραλήπτη, πρωτόκολλο επικοινωνίας, το συνολικό μέγεθος των δεδομένων που μεταδόθηκαν και άλλα [45] [41]. Οι σύγχρονοι δρομολογητές (routers) και μεταγωγείς (switches) περιλαμβάνουν λογισμικό, παραδείγματος χάριν το NetFlow[18] της Cisco, που πραγματοποιεί δειγματοληψία των πακέτων που προωθούνται και παράγουν υψηλού επιπέδου πληροφορία (metadata) σχετικά με τα χαρακτηριστικά των ροών για την αντιμετώπιση προβλημάτων, την βελτιστοποίηση της απόδοσης του δικτύου και την αναγνώριση και πρόληψη απειλών ασφαλείας. Διαφορετικές εκδόσεις του NetFlow και αντίστοιχων προγραμμάτων μπορεί να χρησιμοποιούν ελαφρώς διαφορετικές παραμέτρους ορισμού μιας ροής καθώς και διαφορετικά παραγόμενα στατιστικά

[16] https://www.wireshark.org/. Λογισμικό ανοικτού κώδικα για την ανάλυση δικτύου και ανάπτυξη πρωτοκόλλων.
[17] https://zeek.org/. Πρώην Bro, το Zeek είναι ένα framework που παρέχει μια γλώσσα υψηλού επιπέδου για την ανάλυση δικτυακής κίνησης και την ανάπτυξη εφαρμογών κυβερνοασφάλειας.
[18] https://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html

χαρακτηριστικά κίνησης σύμφωνα με τις ρυθμίσεις της συσκευής [45]. Είναι ο πιο συχνός τύπος δεδομένων που χρησιμοποιούνται απ' τα IDS και παρέχουν μια γενικότερη εικόνα της δικτυακής κίνησης οπότε είναι χρήσιμη για την αναγνώριση DoS και Probe[19] επιθέσεων. Επίσης η απαραίτητη προεπεξεργασία είναι απλούστερη αφού δεν απαιτεί το διάβασμα πακέτων ή την αναδιάρθρωση του session (session restructuring). Ωστόσο επειδή αγνοεί το περιεχόμενο των πακέτων (payload) παρουσιάζει το μειονέκτημα ότι δεν βοηθάει για την ανίχνευση U2R και R2L επιθέσεων, όπως επίσης ότι τα πακέτα χρειάζεται να αποθηκευτούν σε προσωρινή/κρυφή μνήμη (cache) για την ανάλυση χαρακτηριστικών ροής, οπότε και καθιστά την ανίχνευση σε πραγματικό χρόνο πιο δύσκολη. Μπορεί να υλοποιηθεί με χρήση είτε ρηχών είτε βαθιών μοντέλων μάθησης; στην πρώτη περίπτωση είναι επιπλέον απαραίτητο ένα στάδιο εξαγωγής χαρακτηριστικών. Η μεγάλη ετερογένεια των ροών μπορεί να προκαλέσει χαμηλή απόδοση. Η συγκέντρωση της κίνησης (traffic grouping) είναι ο συνήθης τρόπος αντιμετώπισης αυτού του προβλήματος [41].

Η πρώτη ιστορικά και απλούστερη μέθοδος βασίζεται στην χρήση της *μηχανικής χαρακτηριστικών* και αποτελεί ουσιαστικά το πρώτο βήμα πριν την εισαγωγή ρηχών μοντέλων μάθησης. Τα διανύσματα χαρακτηριστικών θα πρέπει είναι ερμηνεύσιμα με τα πιο συνηθισμένα χαρακτηριστικά που χρησιμοποιούνται να είναι μεταξύ άλλων το μέσο μήκος πακέτου, η διασπορά του μήκους πακέτων, ο λόγος πακέτων TCP προς UDP, η αναλογία σημαιών TCP και άλλα. Ο συνδυασμός πολλών μεμονωμένων αδύναμων ταξινομητών, δηλαδή μια μέθοδο ensemble, μπορεί να μετριάσει το πρόβλημα του υψηλού false alarm rate που παρουσιάζεται στα σημερινά IDS που βασίζονται σε εξαγωγή χαρακτηριστικών [41].

Ο δεύτερος τρόπος λειτουργεί με την αξιοποίηση της *βαθιάς μάθησης* και έχει το πλεονέκτημα ότι τα βαθιά δίκτυα απ' την φύση τους δεν απαιτούν γνώση πεδίου (domain) που συνήθως αποτελεί τον μεγαλύτερο περιοριστικό παράγοντα στην απόδοση του μοντέλου. Ως εκ τούτου τείνουν να γίνουν η mainstream επιλογή ανάπτυξης IDS [41].

Την τρίτη και τελευταία μέθοδο αποτελεί η *ομαδοποίηση κίνησης* παρέχει μια λύση στην ετερογένεια των ροών και κατ' επέκταση την εισαγωγή θορύβου και την κακή ανιχνευτική επίδραση που αυτό συνεπάγεται για το μοντέλο. Τέτοιους είδους τεχνικές μπορούν να χωριστούν σε μεθόδους ομαδοποίησης κίνησης που βασίζονται σε πρωτόκολλα και δεδομένα.

[19] Probe ή Scanning επιθέσεις περιλαμβάνουν την σάρωση ενός συστήματος για κενά ασφαλείας ή ανοιχτές πύλες μέσω αυτοματοποιημένων μεθόδων, με σκοπό τον εντοπισμό πιθανών στόχων για περεταίρω επιθέσεις.

Στην πρώτη κατηγορία η βασική ιδέα είναι ότι τα χαρακτηριστικά κίνησης διαφορετικών πρωτοκόλλων παρουσιάζουν σημαντικές διαφορές και ως εκ τούτου κρίνεται σκόπιμη μια αρχική ταξινόμηση με βάση το πρωτόκολλο για την βελτίωση της απόδοσης του μοντέλου [41].

2.2.2.2.3 Πληροφορίες για τις session-based τεχνικές

Η σύνοδος (session) είναι το χρονικό διάστημα μεταξύ μιας αλληλεπίδρασης δύο τερματικών εφαρμογών (πρακτικά ο χρόνος που ένας χρήστης επισκέπτεται μια ιστοσελίδα ή ένα προκαθορισμένο όριο), και συνήθως τα δεδομένα μιας συνόδου χαρακτηρίζονται απ' τα ίδια χαρακτηριστικά με αυτά της ροής, δηλαδή το 5-tuple με IP πελάτη, πύλη πελάτη, IP εξυπηρετητή, πύλη εξυπηρετητή και πρωτόκολλο επικοινωνίας, αλλά σε αντίθεση με τις ροές περιλαμβάνουν πληροφορίες σχετικά με το ωφέλιμο πακέτο, την αλληλουχία των πακέτων και το χρονικό διάστημα της σύνδεσης. Η συλλογή των δεδομένων συνόδου γίνεται συνήθως με Firewalls και IDS. Τα δεδομένα συνόδου είναι χρήσιμα για την ανίχνευση επιθέσεων μεταξύ συγκεκριμένων IP όπως επιθέσεις τύπου tunnel και trojans. Περιλαμβάνει επικοινωνιακά δεδομένα μεταξύ θύματος και θύτη, επομένως βοηθά στον εντοπισμό της επίθεσης. Η διάρκεια ποικίλλει και ενδεχομένως να χρειάζεται cache, επομένως ίσως εισάγει σημαντική καθυστέρηση. Υπάρχουν δύο τρόποι εργασίας με τα δεδομένα συνόδου; τα χαρακτηριστικά στατιστικής ανάλυσης (statistical-based features) και τα χαρακτηριστικά ακολουθιών (sequence-based features) [41].

Η μέθοδος της στατιστικής ανάλυσης χαρακτηριστικών βασίζεται σε πληροφορίες και χαρακτηριστικά στατιστικής φύσης όπως επικεφαλίδες πακέτων, αριθμός πακέτων, αναλογία πακέτων που έρχονται από διαφορετικές πηγές κτλ. Τα χαρακτηριστικά είναι υψηλού επιπέδου επομένως παρέχουν ερμηνευτικότητα και «κουμπώνουν» με κάποια ρηχά μοντέλα όπως δέντρα απόφασης και κανόνες συσχέτισης. Το μεγάλο μειονέκτημα ωστόσο τέτοιων μεθόδων είναι ότι αγνοούν την πληροφορία που πηγάζει από την ακολουθία των πακέτων. Ως αποτέλεσμα είναι προβληματικές στην ανίχνευση εισβολών που σχετίζονται με περιεχόμενο δικτύων. Η υλοποίηση μπορεί να γίνει με επίβλεψη και χωρίς επίβλεψη. Στην πρώτη περίπτωση τα βασικά χαρακτηριστικά των συνόδων χρησιμοποιούνται για την ταξινόμηση κανονικών και ασυνήθιστων μεθόδων. Τέτοια μοντέλα συχνά αντιμετωπίζουν προβλήματα χαμηλής ακρίβειας και υψηλού κόστους χρόνου εκτέλεσης [41].

Η μέθοδος των ακολουθιακών χαρακτηριστικών εκμεταλλεύεται την ακολουθιακή φύση των πακέτων μιας συνόδου και χρησιμοποιεί συνήθως χαρακτηριστικά όπως την

ακολουθία του μήκους των πακέτων και το διάστημα μεταξύ των πακέτων. Κατά κύριο λόγο τέτοια μοντέλα περιορίζονται στην χρήση RNN (Recurrent Neural Networks) επειδή οι περισσότεροι παραδοσιακοί αλγόριθμοι δεν μπορούν να χρησιμοποιηθούν για ακολουθιακά δεδομένα. Απαιτούν ένα βήμα προεπεξεργασίας κειμένου και κωδικοποίησης όπως bag of words (BoW) [41].

2.2.3 Προκλήσεις σε IDS

Το πεδίο ανάπτυξης IDS με μηχανική μάθηση έχει δει μικρότερη επιτυχία σε περιβάλλοντα εργασίας σε σχέση με άλλους τομείς όπου γίνεται χρήση της μηχανικής μάθησης. Οι παράγοντες, μεταξύ άλλων, που κάνουν την διαφορά είναι δυνατόν να ταξινομηθούν ως εξής [46] [47]:

2.2.3.1 Απ’ την πλευρά των μεθόδων

Η μηχανική μάθηση δουλεύει εγγενώς καλύτερα σε προβλήματα ταξινόμησης όπου τα δεδομένα εκπαίδευσης περιέχουν μεγάλο αριθμό παραδειγμάτων από όλες τις κλάσεις. Αντίθετα ο στόχος που καλείται να επιτύχει ένα IDS με μηχανική μάθηση είναι η αναγνώριση επιθέσεων που δεν έχει ξανασυναντήσει και εκ των πραγμάτων δεν έχει εκπαιδευτεί να αναγνωρίζει. Με άλλα λόγια είναι ένα πρόβλημα ανίχνευσης ανωμαλίας, μια υποκατηγορία προβλημάτων δυαδικής ταξινόμησης που χαρακτηρίζεται από μεγάλη ανισορροπία κλάσεων. Υπό αυτήν την έννοια, η επιτυχής αναγνώριση ανωμαλιών που δεν έχει ξανασυναντήσει το σύστημα ανάγεται στο πρόβλημα της τέλειας αναγνώρισης κανονικής λειτουργίας, έτσι ώστε όταν έρθει αντιμέτωπο με ένα παράδειγμα που δεν εμπίπτει σε αυτήν, να ταξινομηθεί ως επίθεση. Το πρόβλημα είναι τώρα εμφανές; Η κανονική λειτουργία δεν είναι ένα σύνολο συμπεριφορών που κάποιος μπορεί να είναι βέβαιος ότι έχει καλύψει όλες τις δυνατές υποπεριπτώσεις. Αντίθετα, σε ένα τυπικό πρόβλημα ταξινόμησης με διάφορες κλάσεις και μια πλούσια συλλογή παραδειγμάτων, το μοντέλο αρκεί να μπορεί να ξεχωρίσει τις κλάσεις, όπως παραδείγματος χάριν, κάνει ένα σύστημα συστάσεων (recommendation system). Έχοντας πει αυτά, ένα IDS με μηχανική μάθηση είναι καλύτερα προετοιμασμένο, όντας εκπαιδευμένο με παραδείγματα κανονικής συμπεριφοράς και γνωστών επιθέσεων, να αναγνωρίζει παραλλαγές των επιθέσεων που εκπαιδεύτηκε να αναγνωρίζει, παρά εντελώς καινούργιες επιθέσεις.

Ένα δεύτερο πρόβλημα που συνδέεται με την ανάπτυξη IDS είναι το υψηλό κόστος που αφορά τις λάθος ταξινομήσεις, αφενός με τα ψευδώς θετικά που σπαταλούν πολύτιμο χρόνο ειδικών, αφετέρου με τα ψευδώς αρνητικά που ενέχουν σοβαρούς κινδύνους για την

επιχείρηση. Ξανά, το κόστος λάθους ταξινόμησης δεν είναι το ίδιο μεγάλο σε άλλους κλάδους αξιοποίησης της μηχανικής μάθησης, όπως συστήματα συστάσεων και οπτικής αναγνώρισης χαρακτήρων.

Το τρίτο πρόβλημα αφορά την δυσκολία ερμηνείας των αποτελεσμάτων των συστημάτων ανίχνευσης ανωμαλιών, που είναι χρήσιμες για την γεφύρωση του χάσματος ανάμεσα σε μία επίθεση και μία απλώς ατυπική κατάσταση κανονικής λειτουργίας, και την μετάφραση τους σε ενέργειες των διαχειριστών δικτύων. Επίσης οι κανόνες ασφαλείας που ορίζουν τι είναι επίθεση και τι κανονική λειτουργία ποικίλουν και είναι απαραίτητο να ανανεώνονται συχνά για να προσαρμόζεται το σύστημα σε διαφορετικά δικτυακά περιβάλλοντα. Ταυτόχρονα, οι πολιτικές ασφαλείας είναι συχνά μη ακριβείς, κάτι που δυσχεραίνει την δουλειά των IDS που βασίζονται σε ανίχνευση ανωμαλίας.

Τέταρτο πρόβλημα είναι η ποικιλομορφία της δικτυακής κίνησης που καθιστά τα χαρακτηριστικά των συνδέσεων σε βραχυχρόνια βάση να είναι περιορισμένης βοήθειας, ενώ σε μεγαλύτερες βάσεις δεδομένων που αποκτούν μια στατιστική δομή, απλούστερα μοντέλα ενδεχομένως είναι εξίσου αποτελεσματικά.

Το τελευταίο πρόβλημα, τέλος, αφορά τις δυσκολίες με την επαλήθευση και ερμήνευση των αποτελεσμάτων του μοντέλου εξαιτίας της απουσίας δημόσιων συνόλων δεδομένων. Αν και τα τελευταία χρόνια έχουν κάνει την εμφάνιση τους όλο και περισσότερα datasets, πολλές έρευνες εξακολουθούν να χρησιμοποιούν ξεπερασμένα datasets που δεν αντιπροσωπεύουν τις σημερινές επιθέσεις. Σε αυτό ευθύνεται και η ίδια η φύση της κυβερνοασφάλειας που προδιαθέτει σε μια «κούρσα εξοπλισμών» ανάμεσα στους μηχανικούς ασφαλείας και τους εισβολείς όπου ο καθένας προσπαθεί να ξεγελάσει ή να προβλέψει τον άλλον.

2.2.3.2 Απ’ την πλευρά των δεδομένων

2.2.3.2.1 Απουσία δεδομένων από πραγματικά δίκτυα

Τα δεδομένα κίνησης που καταγράφονται σε σύνολα δεδομένων πολύ γρήγορα καθίστανται απαρχαιωμένα και δεν ανταποκρίνονται στις επιθέσεις που συμβαίνουν στον πραγματικό κόσμο. Επειδή η συλλογή δεδομένων από πραγματικά δίκτυα είναι κοστοβόρα διαδικασία, οι ερευνητές συχνά καταφεύγουν σε περιβάλλοντα προσομοίωσης με παραγωγή συνθετικών δεδομένων, με αρνητική όμως επίδραση στην προγνωστική αξία των συνόλων δεδομένων.

Ταυτόχρονα όμως δεν υπάρχει καμία διαβεβαίωση για το πόσο καλά θα ανταποκριθεί ένα anomaly-based IDS σε πραγματικά δίκτυα που καταναλωτές ή επιχειρήσεις χρησιμοποιούν. Θεωρείται ότι η απόκτηση ενός συνόλου δεδομένων που να αντανακλά σημερινά πραγματικά δίκτυα είναι εξαιρετικά δύσκολη υπόθεση λόγω της σπανιότητας και της εμπιστευτικότητας των διεισδύσεων δικτύων. Την ίδια στιγμή δεν θα ήταν μια καλή ιδέα η εκπαίδευση του IDS υπό πραγματικές συνθήκες εξαιτίας των συνεπειών στην ασφάλεια του δικτύου των εισβολών που δεν ανιχνεύτηκαν. Παρόλο αυτά, η γενική ιδέα για την ανάπτυξη ενός IDS είναι ότι οι ερευνητές μπορούν να μεταφέρουν ό,τι έχει μάθει ένα IDS μέσα σε ένα περιβάλλον προσομοίωσης, σε ένα εν-χρήσει δίκτυο, υπό την προϋπόθεση ότι το περιβάλλον προσομοίωσης μιμείται τα μοτίβα κίνησης που εμφανίζονται στο πραγματικό δίκτυο.

2.2.3.2.2 Προβλήματα με χαρακτηριστικά

Ένα πρόβλημα που τονίζεται στην βιβλιογραφία είναι ότι τα σύνολα δεδομένων είναι δυνατόν να εμπεριέχουν ακραίες τιμές. Για την αντιμετώπιση τους έχουν εφαρμοστεί μέθοδοι κανονικοποίησης χαρακτηριστικών με στόχο την ελάττωση της βαρύτητας του θορύβου. Η επιλογή χαρακτηριστικών με βάση την πυκνότητα είναι μια άλλη μέθοδος που έχει χρησιμοποιηθεί για τον εντοπισμό των πιο σημαντικών χαρακτηριστικών μέσω της εύρεσης των επικαλύψεων και μη επικαλύψεων των κατανομών πιθανότητας των χαρακτηριστικών.

Ένα άλλο πρόβλημα είναι τα πλεονάζοντα δεδομένα όπου κάποια χαρακτηριστικά σε ένα σύνολο δεδομένων μπορεί να μην συνεισφέρουν σημαντικά στην προληπτική δύναμη ενός μοντέλου, οπότε και μπορούν να αφαιρεθούν, δοθέντος μιας μετρικής της σημαντικότητας των χαρακτηριστικών.

Η απουσία ισχυρής συσχέτισης μεταξύ χαρακτηριστικών στα δεδομένα επίσης τονίζεται ως ένας παράγοντας που μπορεί να καταστήσει την δημιουργία ενός μοντέλου δυσκολότερη. Η συσχέτιση μπορεί να αυξηθεί μέσω της αύξησης της διαστατικότητας των δεδομένων είτε μέσω της ένωσης δεδομένων (data fusion), είτε μέσω της εισαγωγής νέων χαρακτηριστικών.

Άμεσα σχετική με τα προηγούμενα είναι η ιδέα της στιβαρότητας ενός μοντέλου. Ένα μοντέλο θεωρείται στιβαρό όταν η ακρίβεια ενός μοντέλου δεν επηρεάζεται από αλλαγές στα δεδομένα εισόδου όπως από μετακινήσεις κατανομών ή ακραίες τιμές. Για την ανίχνευση εισβολής, αλλαγές σε ροές δικτυακών δεδομένων μπορούν επίσης να έρθουν από έναν κακοπροαίρετο «αντίπαλο» που θα προσπαθήσει να περιπλέξει τα ωφέλιμα φορτία πακέτων

επίθεσης με στόχο να μιμηθούν τα αντίστοιχα πακέτα κανονικής λειτουργίας. Μπορούμε να διαχωρίσουμε τους αντιπάλους σε προγράμματα γεννήτορες δεδομένων και ειδικούς κυβερνοασφάλειας ικανούς να καμουφλάρουν ωφέλιμα φορτία δικτύων με σκοπό να εμφανιστούν ως καλοήθη και να εξαπατήσουν προγράμματα IDS. Για τον μετριασμό της απώλειας στην συνολική προγνωστική αξία των μοντέλων λόγω θορύβου ή αντιπάλων, διαφορετικές στιβαρές μέθοδοι έχουν αναπτυχθεί.

2.2.3.2.3 Προβλήματα με ετικέτες

Ένα σημαντικό πρόβλημα που εμφανίζεται στην πράξη είναι η απουσία ετικετών, ειδικά όταν η δικτυακή κίνηση είναι αμφιλεγόμενη ή χωρίς ετικέτα. Προς την επίλυση αυτού του προβλήματος έχουν αναπτυχθεί παραδείγματα μεθόδων όπως χρήση συνδυασμού μάθησης χωρίς επίβλεψη για την ετικοποίηση των δεδομένων ακολουθούμενο από έναν ταξινομητή, transfer learning όπου γίνεται μεταφορά γνώσης από άλλες ετικοποιημένες πηγές, και αντιπαραθετικής παραγωγής συνθετικών παραδειγμάτων.

Για την αντιμετώπιση της περίπτωσης της ανισορροπίας των κλάσεων όπως έχει αναφερθεί προηγουμένως υπάρχουν μέθοδοι υπερδειγματοληψίας και υποδειγματοληψίας και αυτές αποτελούν το πρώτο τείχος άμυνας. Άλλοι τρόποι αντιμετώπισης αποτελούν μέθοδοι βέλτιστης εξαγωγής χαρακτηριστικών, σιαμαία νευρωνικά δίκτυα, ένωση χαρακτηριστικών καθώς και γενετικός προγραμματισμός.

Τέλος δεν πρέπει να υποτιμηθεί η εξηγησιμότητα του μοντέλου. Διάφορες ερμηνεύσιμες μέθοδοι μπορούν να χρησιμοποιούν όπως συστήματα κανόνων, στατιστική ανάλυση αποτελεσμάτων νευρωνικών δικτύων και μέθοδοι μη επιβλεπόμενης μάθησης όπως μεταβλητοί αυτοκωδικοποιητές.

2.2.3.2.4 Προβλήματα με instances

Τα προβλήματα που παρουσιάζονται σε παραδείγματα είναι η αναγκαία αντιμετώπιση δυναμικών δεδομένων, όπως επίσης και ο πολύ μεγάλος ή πολύ μικρός όγκος των δεδομένων. Για την πρώτη περίπτωση μπορούν να χρησιμοποιηθούν μοντέλα που ειδικεύονται σε δεδομένα ροής (streaming data), όπως π.χ. μέθοδοι συσταδοποίησης πυκνότητας και διαχωρισμού, αξιοποίηση παράλληλων υπολογιστικών μονάδων όπως GPU, μαζί με προσαρμοστικά μοντέλα που λειτουργούν με επαυξητική ενημέρωση, καθώς επίσης και reinforcement learning. Για την δεύτερη περίπτωση των Big Data μπορεί να χρησιμοποιηθεί επαυξητική ενημέρωση μαζί με υπολογιστικά νέφη (cloud computing) όπως Amazon Web

Services και Apache Spark. Τέλος για την τελευταία κατηγορία των Small Data μπορεί να γίνει χρήση μεθόδων Meta learning όπως τεχνικές αυτόματης μάθησης, ensemble τεχνικές με Error-correcting output code (ECOC) και Adaboost, few-shot learning αλλά και Transfer learning.

Ας σημειωθεί ότι με τον όρο *επαυξητική ενημέρωση* (incremental update) λογίζεται η διαδικασία κατά την οποία ένα μοντέλο δεν εκπαιδεύεται με όλα τα διαθέσιμα δεδομένα ταυτόχρονα, αλλά ενημερώνεται σταδιακά όταν νέα δεδομένα γίνονται διαθέσιμα.

2.2.4 Προδιαγραφές και σύγκριση Αλγορίθμων Μηχανικής Μάθησης για IDS

Εξαιτίας των πολλών μεθόδων μηχανικής μάθησης και αλλά και των παραμέτρων που παρουσιάζονται κατά την διάρκεια της προσαρμογής του μοντέλου ώστε να ανταποκρίνεται καλύτερα στην κατανομή των δεδομένων, είναι κρίσιμη μια μελέτη της υπολογιστικής πολυπλοκότητας των αλγορίθμων. Προς αυτό τον σκοπό οι Buczak et al. [48] έκαναν μια συστηματική αναζήτηση της σχετικής βιβλιογραφίας και τα αποτελέσματα συνοψίζονται στον Πίνακα 1. Ας σημειωθεί ότι όπως οι ίδιοι οι συγγραφείς παραδέχονται, κάποιες απ’ τις χρονικές πολυπλοκότητες είναι προσεγγιστικές και επιδέχονται κριτικής αφού εν γένει εξαρτώνται απ’ την εμπειρία του χρήστη και την συγκεκριμένη υλοποίηση. Στον πίνακα υποτίθεται ότι τα δεδομένα αποτελούνται από n δείγματα όπου το καθένα δείγμα περιγράφεται από m χαρακτηριστικά και ο αριθμός δειγμάτων είναι πολύ μεγαλύτερος απ’ το πλήθος των χαρακτηριστικών.

Οι online υλοποιήσεις όπως είναι εύλογο παρουσιάζουν ορισμένες προϋποθέσεις που τις διαχωρίζουν απ’ τις offline, όπως τις εισαγωγές και εξαγωγές ροών δεδομένων, τον διαχωρισμό των ροών δεδομένων εισαγωγής, το buffering, την εμφάνιση των αποτελεσμάτων με την κατάλληλη χρονική πληροφορία καθώς και την συλλογή και συγκέντρωση των αποτελεσμάτων εν παραλλήλω.

Γενικά οι εφαρμογές πραγματικού χρόνου πρέπει να ικανοποιούν τρεις προδιαγραφές; χρονική πολυπλοκότητα, επαυξητική ενημέρωση και ικανότητα γενίκευσης.

2.2.4.1 Χρονική πολυπλοκότητα

Οι αλγόριθμοι πολυπλοκότητας O(n) και O(nlogn) θεωρούνται προσεγγιστικά γραμμικοί και μπορούν να χρησιμοποιηθούν σε εφαρμογές πραγματικού χρόνου (online). Οι τετραγωνικοί αλγόριθμοι πολυπλοκότητας $O(n^2)$ κρίνονται αποδεκτοί για τις περισσότερες εφαρμογές. Οι κυβικοί αλγόριθμοι είναι πολύ πιο αργοί και χρησιμοποιούνται μόνο για offline εφαρμογές.

2.2.4.2 Επαυξητική ενημέρωση

Η τεχνική αυτή βελτιώνει την αποδοτικότητα του μοντέλου και του προσδίδει κλιμακωσιμότητα, κάνοντας εφικτή την υλοποίηση online συστημάτων που βελτιώνονται γρήγορα με λίγα δεδομένα κάθε φορά. Κάτι τέτοιο είναι πολύ σημαντικό σε εφαρμογές IDS, καθώς το μοντέλο μπορεί να ενημερώνεται ακόμη και επί καθημερινής βάσεως με νέα signatures οπότε τίθεται θέμα χρόνου. Τα μοντέλα που βασίζονται σε μεθόδους συσταδοποίησης, στατιστικές μεθόδους και συνολική μάθηση είναι προσιτά σε κάτι τέτοιο. Τα τεχνητά νευρωνικά δίκτυα, SVMs, και οι εξελικτικοί αλγόριθμοι ωστόσο είναι δυνατόν να παρουσιάσουν επιπλοκές. Για την περίπτωση των νευρωνικών δικτύων η επαυξητική ενημέρωση μπορεί να υλοποιηθεί με την υιοθέτηση online learning τεχνικών όπου το νευρωνικό δίκτυο ενημερώνεται κάθε φορά που κάποιο νέο δείγμα γίνεται διαθέσιμο. Ο άλλος τρόπος είναι η ενημέρωση με μικρά σύνολα (batches) δεδομένων κάθε φορά.

Ένα απ' τα πιο σημαντικά προβλήματα που μπορούν να προκύψουν είναι το λεγόμενο Catastrophic Forgetting. Το πρόβλημα αυτό εμφανίζεται όταν το μοντέλο ξεχνάει ό,τι έχει προηγουμένως μάθει στην προσπάθεια του να συμπεριλάβει νέα δεδομένα. Το μοντέλο ιδανικά θα πρέπει να μπορεί να βρει μια ισορροπία ανάμεσα στα παλιά και νέα δεδομένα, να παρουσιάζει δηλαδή μια σταθερότητα απ' την εμπειρία που απέκτησε ενώ ταυτόχρονα διατηρεί την ικανότητα προσαρμογής ακόμα και σε μικρό αριθμό νέων δεδομένων που διαφοροποιούνται χωρίς ωστόσο να παρουσιάζει υπερπροσαρμογή.

2.2.4.3 Ικανότητα γενίκευσης

Το τελευταίο πρόβλημα είναι πολύ σημαντικό: Είναι αναγκαίο τα μοντέλα να μπορούν να γενικευτούν «καλά», με την έννοια ότι νέα δεδομένα δεν θα πρέπει να προκαλούν μεγάλη απόκλιση στο νέο εκπαιδευμένο μοντέλο και θα πρέπει να μπορούν να ανταποκρίνονται «επιτυχώς» σε άγνωστα δεδομένα. Οι περισσότεροι σύγχρονοι αλγόριθμοι πληρούν αυτό το κριτήριο.

2.2.4.4 Παράγοντες απόδοσης ενός IDS

Η απόδοση των IDS χαρακτηρίζεται αφενός απ' τα δεδομένα, ιδανικά περιλαμβάνοντας και δικτυακά και του λειτουργικού συστήματος (λ.χ. kernel calls), αφετέρου απ' την επιλογή του/των αλγορίθμου/ων και της συνολικής αρχιτεκτονικής του συστήματος (λ.χ. βάση υπογραφών). Η κατανοησιμότητα του μοντέλου είναι επίσης σημαντικής σημασίας καθώς

βοηθά τον διαχειριστή να αναγνωρίζει εύκολα τα χαρακτηριστικά του μοντέλου και να βελτιώνει το σύστημα.

Πίνακας 1: Σύγκριση αλγορίθμων μηχανικής μάθησης για online IDS

Αλγόριθμος	Τυπική Χρονική Πολυπλοκότητα	Ικανότητα Streaming	Παράμετροι
Τεχνητά Νευρωνικά Δίκτυα	O(emnk)	Μικρή	e: αριθμός εποχών k: αριθμός νευρώνων
Κανόνες Συσχέτισης	>> $O(n^3)$	Μικρή	
Μπαγεσιανά Δίκτυα	O(mn)	Υψηλή	
Συσταδοποίηση k-means	O(kmni)	Υψηλή	i: αριθμός επαναλήψεων μέχρι την εύρεση του κατωφλίου k: αριθμός συστάδων
Ιεραρχική Συσταδοποίηση	$O(n^3)$	Χαμηλή	
Συσταδοποίηση με DBSCAN	O(nlogn)	Υψηλή	
Δένδρα Απόφασης	$O(mn^2)$	Μέτρια	
Γενετικοί Αλγόριθμοι	O(gkmn)	Μέτρια	g: αριθμός γενεών k: μέγεθος πληθυσμού
Αφελής Μπάγιες	O(mn)	Υψηλή	
Κοντινότερος Γείτονας k-NN	O(nlogk)	Υψηλή	k: αριθμός γειτόνων
HMM	$O(nc^2)$	Μέτρια	c: αριθμός καταστάσεων (κατηγοριών)
Τυχαίο Δάσος	O(Mmnlogn)	Μέτρια	M: πλήθος δένδρων απόφασης
Εξόρυξη Ακολουθιών	>>$O(n^3)$	Χαμηλή	
Μηχανές Υποστήριξης Διανυσμάτων	$O(n^2)$	Μέτρια	

3 Datasets και υλοποιήσεις

Ξεκινώντας απ' το πρώτο βήμα σχεδιασμού ενός IDS, που αποτελεί η συλλογή και χρήση συνόλων δεδομένων, μπορούμε καταρχάς να αναφέρουμε μερικά απ' τα διαθέσιμα δημόσια datasets που εμφανίζονται περισσότερο στην βιβλιογραφία.

3.1 Datasets

3.1.1 KDDCup1999

Το KDDCup1999 [49] αποτέλεσε ιστορικά το πρώτο dataset ευρείας χρήσης. Είναι μια έκδοση του DARPA Intrusion Detection Evaluation Program του 1998 που συλλέχθηκε απ' το Lincoln Labs του MIT. Ακόμη και σήμερα, περισσότερο από δύο δεκαετίες αργότερα, χρησιμοποιείται ευρύτατα για την εκπαίδευση και επαλήθευση μοντέλων. Το Lincoln Labs σύλλεξε ανεπεξέργαστα δεδομένα εννέα εβδομάδων μέσω tcp dump από ένα δίκτυο LAN που προσομοιάζει ένα LAN της αεροπορίας. Οι επιθέσεις ταξινομούνται σε τέσσερις κατηγορίες: DoS, R2L, U2R και Port Scanning. Αν και το dataset θεωρείται σχετικά μεγάλο στον βαθμό που περιλαμβάνει 41 χαρακτηριστικά[20] και περισσότερες από 4.8 εκατομμύρια εγγραφές δεδομένων, παρουσιάζει το πρόβλημα διπλοεγγραφών μεταξύ των συνόλων δεδομένων εκπαίδευσης και ελέγχου. Σημαντικά χαρακτηριστικά όπως διευθύνσεις IP απουσιάζουν απ' τα δεδομένα, και παρόλο που έχει καταγραφεί ένας ικανοποιητικός αριθμός επιθέσεων, τα δεδομένα έχουν συλλεχθεί σε ένα συνθετικό δίκτυο (ή δίκτυο προσομοίωσης). Εν κατακλείδι τα δεδομένα είναι παρωχημένα επειδή συλλέχθηκαν περισσότερο από δύο δεκαετίες πριν και επειδή τα δεδομένα είναι συνθετικά, κάτι που το καθιστούν ανεπαρκές για την ανάπτυξη ενός σύγχρονου IDS.

Εκτός αυτών παρουσιάζει και τα εξής προβλήματα:

- Όλα τα πακέτα δεδομένων με επιθέσεις σε αυτά έχουν TLL (Time-To-Live) 126 ή 253 ενώ τα πακέτα της κίνησης κυρίως έχουν 127 ή 254. Όμως, οι τιμές TTL 126 ή 253 δεν υπάρχουν στα δεδομένα εκπαίδευσης των επιθέσεων.
- Η κατανομή πιθανότητας του συνόλου test είναι διαφορετική από την κατανομή πιθανότητας στα δεδομένα εκπαίδευσης λόγω της προσθήκης καταγραφών νέων επιθέσεων στο σύνολο δοκιμών. Αυτό οδηγεί τις μεθόδους ταξινόμησης να έχουν

[20] http://kdd.ics.uci.edu/databases/kddcup99/task.html

προκατάληψη σε κάποιες καταγραφές και να μην υπάρχει ισορροπία μεταξύ των επιθέσεων και της κανονικής λειτουργίας.

- Το σύνολο δεδομένων δεν είναι μια διεξοδική αναπαράσταση των τελευταίων επιθέσεων μικρού αποτυπώματος που έχουν παρατηρηθεί.

3.1.2 NSL-KDD 2009

Το NSL-KDD [49] αποτελεί την βελτιωμένη εκδοχή του KDD και κατασκευάστηκε με σκοπό να έχει ορισμένα πλεονεκτήματα σε σχέση με το αρχικό dataset:

- Αφαίρεσε τις διπλές εγγραφές ώστε οι ταξινομητές να μην προτιμάνε αναίτια την κατηγοριοποίηση με τις περισσότερες εγγραφές.
- Έγινε επιλογή εγγραφών από διαφορετικά μέρη του αρχικού KDD ώστε οι ταξινομητές να παρουσιάζουν πιο συνεπή αποτελέσματα.
- Απάλειψε το πρόβλημα ανισορροπίας μεταξύ του δεδομένων εκπαίδευσης και δοκιμών και μείωσε το False Alarm Rate.

Παρόλο αυτά το βασικό πρόβλημα που είχε το αρχικό dataset παραμένει και το NSL-KDD δεν παρέχει μια χαρακτηριστική απεικόνιση των νέων τύπου επιθέσεων μικρού αποτυπώματος που έχουν προκύψει τα τελευταία χρόνια.

Περιλαμβάνει ετικέτες κανονικής λειτουργίας και τεσσάρων οικογενειών επιθέσεων, έχει 41 χαρακτηριστικά εκ των οποίων 3 είναι κατηγορικά, 4 δυαδικά και τα υπόλοιπα συνεχή. Ακόμα το training dataset περιλαμβάνει ετικέτες για 22 ειδών επιθέσεων ενώ το testing dataset περιλαμβάνει 21 ειδών επιθέσεων που βρίσκονται στο training dataset και 16 καινούργιες επιθέσεις.

3.1.3 UNSW-NB15

Για την κατασκευή του UNSW-NB15 [50] χρησιμοποιήθηκαν 3 διακομιστές, ενός εκ των οποίων δεχόταν την κίνηση με τις επιθέσεις, και δύο routers, στο ένα εκ των οποίων είχε εγκατασταθεί το tcpdump για την παραγωγή των pcap αρχείων. Στην συνέχεια τα pcap αρχεία/ανεπεξέργαστα πακέτα πέρασαν στο λογισμικό Argus με σκοπό την παραγωγή αρχείων σε μορφή ροών/flows. Επίσης έγινε χρήση του Bro-IDS tool για την ανάλυση κίνησης, το οποίο δημιουργεί 3 log files απ' τα pcap files, σχετικά με πληροφορίες σύνδεσης, αιτήσεις και απαντήσεις HTTP και καταγραφές FTP. Τέλος τα κοινά δεδομένα απ' τα δύο προγράμματα συμπτύχθηκαν σε μια βάση δεδομένων. Η βάση δεδομένων περιλαμβάνει 35[21] μικτά features

[21] https://www.researchgate.net/figure/Features-of-UNSW-NB15-dataset_tbl1_324601933

σε επίπεδο πακέτων και ροής και χωρίζονται σε τρεις κατηγορίες: Βασικά, Περιεχομένου και Χρόνου. Επίσης υπάρχουν 12 επιπρόσθετα features που δεν περιλαμβάνονται στην βάση δεδομένων. Οι ετικέτες χωρίζονται σε δύο features, ένα για την κατηγοριοποίηση της επίθεσης και ένα για την ύπαρξη ή όχι επίθεσης. Υπάρχουν συνολικά τέσσερα csv αρχεία με τα δεδομένα κίνησης καθώς και ένα αρχείο με επεξήγηση των χαρακτηριστικών. Χρησιμοποιήθηκε για τις ανάγκες της διπλωματικής.

3.1.4 UGR'16

Το UGR'16 [51] συλλέχθηκε από πολλαπλούς NetFlow v9 collectis στο δίκτυο ενός Ισπανικού ISP από ερευνητές του πανεπιστημίου της Γρανάδας στην Ισπανία. Τα δεδομένα έχουν χωριστεί ένα calibration και ένα training set, όπου η μακρόχρονη εξέλιξη και περιοδικότητα στα δεδομένα είναι ένα μεγάλο πλεονέκτημα σε σύγκριση με προηγούμενα σύνολα δεδομένων. Παρόλο αυτά ένα σημαντικό πρόβλημα είναι ότι το μεγαλύτερο μέρος της κίνησης έχει ως ετικέτα «background» που υποδηλώνει είτε ανωμαλία είτε benign. Επίσης περιλαμβάνει μίξη συνθετικών επιθέσεων μαζί με πραγματικές επιθέσεις, που δεν μπορεί να συγκριθεί με δεδομένα όπου δεν υπάρχει καμία προσομοίωση. Το dataset έγινε annotated σύμφωνα με τα logs απ' το honeypot σύστημα της υποδομής και περιλαμβάνει 12 χαρακτηριστικά[22].

3.1.5 CIDD-001

Το CIDDS-001 [52] συλλέχθηκε το 2017 από τέσσερις ερευνητές, δύο διδακτορικούς φοιτητές, και δύο καθηγητές, που συνεργάζονται με το πανεπιστήμιο εφαρμοσμένων επιστημών στο Coburg της Γερμανίας. Τα δεδομένα ήταν μέρος του project WISENT, χρηματοδοτούμενου απ' το Βαυαρικό Υπουργείο Οικονομικών. Το dataset δημιουργήθηκε με στόχο να χρησιμοποιηθεί ως evaluation dataset σε συστήματα ανίχνευσης εισβολής βασιζόμενα σε αναγνώριση ανωμαλιών. Περιλαμβάνει ετικέτες και είναι οργανωμένο σε μορφή ροών ενώ βασίστηκε στην προσομοίωση ενός επιχειρηματικού περιβάλλοντος πάνω στο OpenStack. Η συνολική αρχιτεκτονική αποτελείται από τρεις επιτιθέμενους και έναν εξωτερικό εξυπηρετητή που έχει ένα firewall που τον διαχωρίζει από έναν άλλο εξυπηρετητή όπου υπάρχουν τρία επίπεδα: Ανάπτυξης λογισμικού, Office και management. Υπάρχουν τέσσερις servers στο περιβάλλον OpenStack που περιέχουν τα τρία subnet επίπεδα. Τα είδη των επιθέσεων που περιλαμβάνονται στο δίκτυο είναι DoS, Brute Force και Port Scanning[23].

[22] https://www.researchgate.net/figure/Data-features_tbl1_341408384
[23] https://www.researchgate.net/figure/Features-of-the-CIDDS-001-dataset_tbl1_323759289

Υπάρχουν τριών ειδών ετικετών σχετικών με τις επιθέσεις. Η πρώτη ετικέτα κατηγοριοποιεί την κίνηση ως normal, attacker, victim, suspicious και unknown. Η δεύτερη είναι το είδος της επίθεσης και η τρίτη ένα αναγνωριστικό ID της επίθεσης. Το συγκεκριμένο dataset χρησιμοποιείται κυρίως για σκοπούς benchmarking. Περιλαμβάνει μικρό αριθμό τύπου επιθέσεων.

3.1.6 CICIDS'17

Αυτό το σύνολο δεδομένων [53] χρησιμοποιήθηκε στην παρούσα εργασία. Δημιουργήθηκε απ' το Καναδικό Ινστιτούτο Κυβερνοασφάλειας σε συνεργασία με το Πανεπιστήμιο του New Brunswick. Το δίκτυο των επιτιθέμενων αποτελείται από δύο IPs, ενός υπολογιστή που τρέχει Kali Linux και ενός που τρέχει Windows. Το δίκτυο των θυμάτων αποτελείται από δύο Web servers 16, έξι Ubuntu servers, πέντε Windows servers και έναν MAC server. Περιλαμβάνει δεδομένα σε επίπεδο pcap αρχείων και σε επίπεδο ροών κατασκευασμένων απ' το CICFlowMeter, παρεχόμενα σε CSV αρχεία. Αποτελείται από δύο zip αρχεία: GeneratedLabelledFlows και MachineLearningCVE. Το πρώτο περιλαμβάνει 85[24] χαρακτηριστικά συν ένα για την ετικέτα κατηγοριοποίησης των επιθέσεων. Το δεύτερο περιλαμβάνει 78, δηλαδή 6 λιγότερα απ' το άλλο. Αυτά τα έξι σχετίζονται με την αναγνώριση των ροών (διευθύνσεις IP, αναγνωριστικά ροών κτλ.) και δεν πρέπει να χρησιμοποιούνται για την εκπαίδευση μοντέλων, επομένως χρησιμοποιήθηκε το δεύτερο αρχείο. Το dataset επελέγη επειδή αποτελεί ρεαλιστικό παράδειγμα κίνησης δικτύου, περιλαμβάνει ετικέτες, ανωνυμότητα, ετερογένεια, διαθεσιμότητα πρωτοκόλλων, ποικιλομορφία επιθέσεων καθώς και την πλήρη παραμετροποίηση του δικτύου υπό εξέταση. Η καταγραφή των δεδομένων έγινε σε πέντε εργάσιμες ημέρες. Η Δευτέρα ήταν η μέρα της κανονικής λειτουργίας. Οι επιθέσεις (ετικέτες) περιλαμβάνουν DoS, DDoS, Heartbleed, Web Attacks, FTP Brute Force, SSH Brute Force, Infiltration, Botnet και Port Scans.

[24] https://www.researchgate.net/figure/Features-in-cic-ids-2017-dataset_tbl1_343850781

3.2 Σχετικές Μελέτες

Στο άρθρο τους οι Javaid et al. [54] κάνουν χρήση του NSL-KDD dataset και προτείνουν μια βαθιά μέθοδο ημιεπιβλεπόμενης ανίχνευσης ανωμαλίας αποτελούμενη από δύο επίπεδα. Στο πρώτο επίπεδο μια «καλή» αναπαράσταση των χαρακτηριστικών μαθαίνεται από μια μεγάλη συλλογή δεδομένων χωρίς αναγνωριστική ετικέτα μέσα από μια διαδικασία μη επιβλεπόμενης μάθησης. Στο δεύτερο επίπεδο η νέα αναπαράσταση των χαρακτηριστικών εφαρμόζεται σε δεδομένα με αναγνωριστική ετικέτα και γίνεται η εκπαίδευση της ταξινόμησης. Τα δεδομένα με και χωρίς αναγνωριστική ετικέτα μπορούν να προέρχονται από διαφορετικές κατανομές αλλά πρέπει να συσχετίζονται. Για το πρώτο βήμα της μη επιβλεπόμενης μάθησης οι επιλογές περιλαμβάνουν μεταξύ άλλων τεχνικές όπως Αραιός Αυτοκωδικοποιητής, Περιορισμένη Μηχανή Boltzmann, K-Μέσων Συσταδοποίηση και γκαουσιανοί συνδυασμοί (Gaussian Mixtures). Οι συγγραφείς επέλεξαν τον Αραιό Αυτοκωδικοποιητή για το μοντέλο τους λόγω της απλότητας και της απόδοσης του. Ο αυτοκωδικοποιητής που υλοποιήθηκε αποτελείται από τρία πλήρως συνδεδεμένα στρώματα, ένα στρώμα εισαγωγής, ένα κρυφό στρώμα και ένα στρώμα εξόδου. Μία σύντομή περιγραφή της λειτουργίας του αυτοκωδικοποιητή παρέχεται στο Παράρτημα.

Πιο αναλυτικά το μοντέλο λειτουργεί ως εξής: Αρχικά γίνεται μία στοιχειώδης προεπεξεργασία των δεδομένων όπως One-Hot κωδικοποίηση, Min-Max κανονικοποίηση, και στην συνέχεια τα δεδομένα περνάνε στον αυτοκωδικοποιητή προκειμένου να εκπαιδευτεί και να μάθει μία καλή συμπιεσμένη αναπαράσταση των δεδομένων, στην προσπάθεια του δικτύου στην έξοδο του να ανακατασκευάσει πιστά την είσοδο που δέχθηκε. Στο επόμενο βήμα οι τιμές των βαρών (weights) και πολώσεων (biases) που έμαθε το κομμάτι του κωδικοποιητή του αυτοκωδικοποιητή, εφαρμόζονται σε έναν μικρό αριθμό νέων ετικετοποιημένων δεδομένων για τον σχεδιασμό ενός δεύτερου συστήματος πολυωνυμική λογιστικής παλινδρόμησης (ή παλινδρόμηση softmax) που εκπαιδεύεται ως ταξινομητής χρησιμοποιώντας την κρυφή αναπαράσταση του έμαθε ο αυτοκωδικοποιητής. Η ελαχιστοποίηση της συνάρτησης κόστους του αυτοκωδικοποιητή γίνεται με χρήση του αλγόριθμου οπισθοδιάδοσης. Για το κρυφό και εξωτερικό επίπεδο ως συνάρτηση ενεργοποίησης χρησιμοποιείται η σιγμοειδής (sigmoid).

Οι συγγραφείς εργάστηκαν με δύο τρόπους για την συλλογή των αποτελεσμάτων: n-cross validation πάνω στο training dataset, και με διαχωρισμό των δεδομένων σε διαφορετικά training και test datasets. Η τελευταία αποφέρει πιο ρεαλιστικά αποτελέσματα. Επιπλέον εξετάστηκε η ταξινόμηση δύο κατηγοριών (normal/attack), 5 κατηγοριών (normal/4 attack

categories), και τέλος 23 κατηγοριών (normal/22 attacks) με χρήση των τεσσάρων πιο κοινών μετρικών (Accuracy, Precision, Recall και F-score). Τέλος, συγκρίνουν τα αποτελέσματα του δικού τους ολοκληρωμένου μοντέλου 2 σταδίων με ένα απλούστερο χωρίς χρήση της μη επιβλεπόμενης μάθησης στο πρώτο στάδιο, δηλαδή με απλό soft-max regression.

Τα αποτελέσματα επί δεδομένων του training dataset έδειξαν ότι το ολοκληρωμένο μοντέλο των συγγραφέων παρήγαγε συγκρίσιμη επίδοση με τα καλύτερα αποτελέσματα της βιβλιογραφίας που εξέτασαν οι συγγραφείς. Μερικά τέτοια μοντέλα επιγραμματικά είναι Δένδρα Απόφασης τύπου J48, Αφελής Μπάγιες, Τυχαίο Δάσος, PCA μαζί με SVM. Αλλά και για δεδομένα του testing dataset που δεν είχε συναντήσει το μοντέλο κατά την εκπαίδευση του επίσης παράγει συνολικά καλύτερα αποτελέσματα συγκρινόμενο με το απλό soft-max μοντέλο παλινδρόμησης.

Οι Vinayakumar et al. [43] εξετάζουν την χρήση Convolution Neural Networks (CNN) και παραλλαγών τους στον τομέα της ανίχνευσης εισβολών και NIDS. Συγκεκριμένα εξετάζουν τις παραλλαγές CNN-RNN, CNN-LSTM, CNN-GRU καθώς και το απλό CNN με 1, 2 και 3 στρώματα (layers). Πιο συγκεκριμένα οι ερευνητές τοποθέτησαν ένα στρώμα εισαγωγής, ένα συνελικτικό στρώμα που δέχεται ένα διάνυσμα 41x1 χαρακτηριστικών του στρώματος εισαγωγής, ακολουθούμενο από ένα max-pooling στρώμα και έπειτα ένα από τα RNN, LSTM, GRU στρώμα ή ένα απλό feed forward νευρωνικό δίκτυο που καταλήγουν σε ένα τελικό στρώμα αποτελεσμάτων. Έγινε έλεγχος για τον κατάλληλο αριθμό filters των συνελικτικών νευρωνικών δικτύων. Επίσης έγινε σύγκριση και με Multi-Layer-Perceptron. Το dataset που χρησιμοποιούν είναι το KDD99. Η ταξινόμηση έγινε υπό δύο μορφές: Δυαδική ταξινόμηση σε «normal» και «επίθεση» και ταξινόμηση όπου η κάθε επίθεση ταξινομείται σε τέσσερις υποκατηγορίες «Dos», «Probe», «u2r», «r2l». Τέλος έγινε μελέτη και για την αξιολόγηση απόδοσης με χρήση μειωμένων συνόλων χαρακτηριστικών πάνω σε CNN-LSTM.

Η επαλήθευση έγινε με το test dataset του KDD99. Για την ταξινόμηση με ετικέτα τύπου επίθεσης το CNN-LSTM είχε καλή απόδοση συγκρινόμενο με τα υπόλοιπα δίκτυα και παρήγαγε αποδεκτή ανίχνευση σε όλες τις κατηγορίες εκτός της επίθεσης «u2l». Για την δυαδική ταξινόμηση το απλό CNN με 1 layer είχε την καλύτερη απόδοση ακολουθούμενο απ' τα υβρίδια CNN-LSTM, CNN-GRU, CNN-RNN με 2 layers. Στην αξιολόγηση των minimal feature sets το CNN-LSTM είχε την καλύτερη απόδοση ενώ και το «απλό» Multi-Layer-Perceptron είχε καλή απόδοση.

Τα συμπεράσματα που προέκυψαν είναι ότι τα CNN έδειξαν ότι υπερτερούν των μέχρι τώρα αποτελεσμάτων της βιβλιογραφίας. Οι παραλλαγές CNN-RNN, κτλ. με πρώτο layer ένα CNN ακολουθούμενο από RNN έδειξαν ότι δεν βελτιώνουν την απόδοση του απλού CNN στις περισσότερες περιπτώσεις. Υπάρχει ανάγκη για καλύτερα datasets και ως προς αυτό οι ερευνητές προσανατολίζονται στο UNSW-NB15. Επίσης στο μέλλον μένει να γίνει on-line ανάλυση, ενδεχομένως συνδυάζοντας περισσότερα δεδομένα από άλλα logs, firewalls, syslogs, routers κτλ. Τέλος τα πιο περίπλοκα μοντέλα απαιτούν μεγαλύτερη υπολογιστική δύναμη και ως προς αυτό η κατεύθυνση είναι στην χρήση ανεπτυγμένων παράλληλων αρχιτεκτονικών.

Οι Dong et al. [55] χρησιμοποιούν το γνωστό KDD99 dataset και την τεχνική SMOTE (Synthetic Minority Oversampling Technique) για να αντιμετωπίσουν την ανισορροπία των δεδομένων. Αξιολογήθηκαν τα ακόλουθα μοντέλα: Δένδρο Απόφασης, Αφελής Μπάγιες, Μηχανή Υποστήριξη Διανυσμάτων και SVM-RBNS. Η μετρική που υιοθετήθηκε ήταν το Precision.

Το αποτέλεσμα ήταν ότι το υβριδικό μοντέλο SVM-RBNS είχε συστηματικά την καλύτερη απόδοση για κάθε κατηγορία (Normal, DOS, UToR, RToL). Αν και το αρχικό paper εμφανίζει δύο φορές το DOS και δεν εμφανίζει καθόλου τα αποτελέσματα για το Probing Attack, υποθέτουμε ότι ισχύουν τα ίδια και για αυτό. Επίσης οι ερευνητές δείχνουν ότι η απόδοση μεγαλώνει όσο μεγαλώνει το training dataset και ακόμη φανερώνουν ότι το SMOTE προκαλεί μεγάλη βελτίωση για την κατηγορία UToR η οποία υποεκπροσωπείται σημαντικά στο αρχικό dataset.

Οι Belavagi et al. [56] συγκρίνουν μεθόδους επιβλεπόμενης μάθησης για να καταλήξουν ποια είναι η καλύτερη για intrusion detection. Οι μέθοδοι είναι οι Λογιστική Παλινδρόμηση, Γκαουσιανός Αφελής Μπάγιες, Μηχανή Υποστήριξης Διανυσμάτων και Τυχαίο Δάσος. Το dataset που επέλεξαν ήταν το NSL-KDD. Έκαναν χρήση όλων των χαρακτηριστικών του και η ταξινόμηση ήταν δυαδική (normal ή επίθεση).

Η αξιολόγηση γίνεται βάσει των μετρικών Precision, Recall, F1-Score και Accuracy. Αρχικά γίνεται μια προεπεξεργασία των δεδομένων του NSL-KDD και χωρίζονται σε training και testing υποσύνολα. Τα δεδομένα εκπαίδευσης εισάγονται στα μοντέλα για την εκπαίδευση και τα εκπαιδευμένα μοντέλα ύστερα δέχονται το testing υποσύνολο για την αξιολόγηση τους.

Οι σχεδιαστές κατασκεύασαν την Reliability Curve, στην οποίο ο ιδανικό ταξινομητής έχει την μορφή της διαγώνιου ευθείας, και αποδείχθηκε ότι το Τυχαίο Δάσος υπερτερεί των υπόλοιπων μεθόδων. Επίσης σχεδίασαν την Receiver Operating Characteristics (ROC) Curve, με άξονες False Positive Rate και True Positive Rate και ξανά φάνηκε ότι το Τυχαίο Δάσος υπερτερεί. Τέλος σε όλες τις 4 μετρικές που αξιολογήθηκαν, το Τυχαίο Δάσος υπερτερεί σημαντικά των άλλων μεθόδων και καταγράφει σε όλες 99%.

Οι Brandao et al. [57] χρησιμοποίησαν το γνωστό σύνολο δεδομένων KDD CUP1999. Με χρήση της Βελτιστοποιημένης Επιλογής Χαρακτηριστικών, υλοποιημένης απ' το στατιστικό πακέτο RapidMiner[25], οι συγγραφείς κατέληξαν σε 23 σημαντικά features. Η βελτιστοποιημένη επιλογή χαρακτηριστικών γενικά μπορεί να υλοποιηθεί με δύο τρόπους, forward selection και backward elimination και αφορά την επιλογή των πιο χρήσιμων χαρακτηριστικών με σκοπό την μείωση της διαστατικότητας και πολυπλοκότητας χωρίς αυτό να οδηγήσει σε μείωση της προβλεπτικής ικανότητας. Επίσης με χρήση βιβλιοθηκών της R έγινε Ανάλυση Παραγόντων και η τελική μειωμένη λίστα χαρακτηριστικών βρέθηκε ότι είναι: service, protocol type, flag, count, logged in, DST host count. Η ανάλυση παραγόντων εστιάζεται στην αναγνώριση παραγόντων που δεν είναι ορατοί (latent ή κρυφοί) και εξηγούν την συσχέτιση ενός ή περισσοτέρων χαρακτηριστικών οπότε και η αρχική λίστα χαρακτηριστικών αντικαθίσταται από μια μικρότερη λίστα παραγόντων με έναν παρόμοιο τρόπο με το PCA. Έγινε προεπεξεργασία για τα κατηγορικά χαρακτηριστικά με One Hot Encoding. Όσον αφορά τα προβλεπτικά μοντέλα έγινε χρήση και σύγκριση Δένδρων Απόφασης, Κ-Κοντινότεροι-Γείτονες και Νευρωνικών Δικτύων. Τα αποτελέσματα έδειξαν ότι τα προβλεπτικά μοντέλα δεν παρουσιάζουν μεγάλη απόκλιση το ένα απ' το άλλο, με τα Δένδρα Απόφασης να είναι λίγο καλύτερα. Το συμπέρασμα είναι ότι με προσεκτική επιλογή των χαρακτηριστικών, το προβλεπτικό μοντέλο επιλογής απ' τα μοντέλα ML που εξετάστηκαν παίζει δευτερεύουσα σημασία στην τελική επίδοση του συστήματος.

Οι Maithem et al. [58] χρησιμοποίησαν ένα βαθύ νευρωνικό δίκτυο με ReLU ως την συνάρτηση ενεργοποίησης για τα κρυφά επίπεδα και softmax για το τελευταίο επίπεδο αποτελεσμάτων, Adam ως τον βελτιστοποιητή και διασταυρούμενη εντροπία ως την συνάρτηση απωλειών (ή κόστους). Το σχήμα του δικτύου αποτελείται από ένα στρώμα εισαγωγής, δύο κρυφά στρώματα 50 και 30 νευρώνων αντίστοιχα, ένα κρυφό στρώμα 2

[25] https://rapidminer.com/. Λογισμικό για εφαρμογές προεπεξεργασίας και εξόρυξης δεδομένων και μηχανικής μάθησης.

νευρώνων που παριστάνουν τις δύο καταστάσεις normal κίνησης και επίθεσης, και ένα τελευταίο στρώμα εξόδου όπου αναλύεται περαιτέρω το είδος της επίθεσης με συνολικά 5 νευρώνες, ένα για την κανονική λειτουργία και 4 για το είδος της επίθεσης. Ως προς την προεπεξεργασία των δεδομένων χρησιμοποιήθηκε One Hot Encoding για τα κατηγορικά δεδομένα και Z-Score για την κανονικοποίηση. Το dataset που χρησιμοποιήθηκε ήταν το KDD CUP 1999. Τα αποτελέσματα και της δυαδικής ταξινόμησης (νορμάλ ή επίθεση) αλλά και την multi-class ταξινόμησης ανάλογα με το είδος της επίθεσης βρέθηκαν αρκετά μεγαλύτερα του 99% σε όλες τις σημαντικές μετρικές (accuracy, recall, fscore κτλ.).

Οι He et al. [59] χρησιμοποίησαν τρία datasets (UNSW-NB15, VPN2016, συνδυασμός CIC2012 και CIC2017) και υλοποίησαν ένα συνελικτικό νευρωνικό δίκτυο σε τρεις εκδοχές ένωσης των δεδομένων που αποτελούνται από pcap και «business feature data». Τα πρώτα αποτελούνται από δεκαεξαδικούς αριθμούς που αντιστοιχίζονται στα μεγέθη σε bytes των επικεφαλίδων και των δεδομένων των πακέτων. Τα δεύτερα αποτελούνται από χαρακτηριστικά ροής, χαρακτηριστικά βάσης, χαρακτηριστικά περιεχομένου, χαρακτηριστικά χρόνου και τέλος παραγόμενα χαρακτηριστικά.

Οι τρεις εκδοχές ένωσης είναι:

- Πρώιμη ένωση: Σειριακή σύνδεση των δεδομένων απ’ την αρχή και είσοδο τους ως ένα μονοδιάστατο διάνυσμα στο CNN και ύστερα η έξοδος σε έναν classifier.
- Ένωση χαρακτηριστικών: Σε αυτή την περίπτωση περνάμε τα δεδομένα (pcap και business feature data) σε δύο ξεχωριστά CNN, η έξοδος των οποίων ενώνεται και εισάγεται σε έναν τελικό classifier.
- Ένωση αποφάσεων: Εδώ όπως και με την δεύτερη περίπτωση τα δεδομένα γίνονται αρχικά είσοδος σε δύο CNN, μετά περνάνε από δύο ξεχωριστούς classifiers, η έξοδος των οποίων περνάει ως είσοδος σε μια συνάρτηση προσδιορισμού βαρών για τους δύο τύπους δεδομένων. Τέλος η έξοδος των νευρωνικών δικτύων περνάει από δύο παράλληλες softmax συναρτήσεις που παράγουν μια κατανομή πιθανοτήτων για normal, abnormal κατηγοριοποιήσεις και σε συνδυασμό με τα παραγόμενα βάρη συνθέτουν δύο εισόδους που καταλήγουν σε έναν τελικό classifier.

Και οι τρεις μέθοδοι παρουσίασαν από μικρή έως σημαντική βελτίωση σε σχέση με τα απλά μοντέλα χρησιμοποίησης μόνο των pcap ή μόνο των business feature data. Τα UNSW-NB15 και CIC datasets παρουσίασαν την μεγαλύτερη βελτίωση στην δεύτερη μέθοδο ενώ το VPN2016 στην τρίτη και πολυπλοκότερη μέθοδο. Η εισαγόμενη καθυστέρηση θεωρείται

μικρή. Οι συγγραφείς επίσης σύγκριναν τα αποτελέσματα τους με έρευνες άλλων συγγραφέων που χρησιμοποίησαν άλλα μοντέλα και συμπέραναν ότι το δικό τους είναι πιο συνεπές. Επίσης συμπέραναν ότι η χρήση μονοδιάστατων CNN είναι προτιμότερη απ' την πιο παραδοσιακή χρήση 2D CNN όσο αναφορά δικτυακά δεδομένα και συστήματα ανίχνευσης εισβολής.

Οι Y. Xiao et al. [60] επέλεξαν ένα συνελικτικό νευρωνικό δίκτυο με στόχο την υλοποίηση ενός IDS, βασιζόμενοι στο KDD dataset και το σύγκριναν με παραδοσιακές μεθόδους μηχανικής μάθησης. Αρχικά μετέτρεψαν τα κατηγορικά χαρακτηριστικά σε αριθμούς με OHE και συνέχισαν με κανονικοποίηση των χαρακτηριστικών. Οι ετικέτες αντικαταστάθηκαν με ακέραιους. Ύστερα χρησιμοποίησαν PCA και Autoencoder για την απαλοιφή μη-χρήσιμων χαρακτηριστικών και συμπίεση των δεδομένων. Ακολούθως άλλαξαν την μορφή του 1D διανύσματος χαρακτηριστικών σε 2D πίνακα για χρήση ως εισόδου κατάλληλης σε συνελικτικό δίκτυο. Τελικά χώρισαν τα δεδομένα σε train και test, αναζήτησαν τις καλύτερες υπερπαραμέτρους για το νευρωνικό δίκτυο και αξιολόγησαν την απόδοση πάνω στο test. Έδειξαν ότι η μείωση της διαστατικότητας με PCA ή αυτοκωδικοποιητή δεν παρουσιάζει μεγάλες διαφορές στην απόδοση του μοντέλου. Επίσης έγινε σύγκριση με μοντέλα λογιστικής παλινδρόμησης, δένδρων απόφασης, τυχαίου δάσους, μηχανής υποστήριξης διανυσμάτων, AdaBoost και Αφελή Μπάγιες και κατέληξαν στην υπεροχή του μοντέλου τους. Ως τελική σύγκριση οι ερευνητές αξιολόγησαν το συνελικτικό δίκτυο συναρτήσει ενός DNN (Deep Neural Network) και ενός RNN (Recurrent Neural Network) και κατέληξαν ότι το CNN υπερτερεί και ως προς τον χρόνο εκπαίδευσης και ως προς το detection rate σε μικρότερο βαθμό.

4 Μεθοδολογία Υλοποίησης Συστήματος

4.1 Γενικά

Στο πλαίσιο της διπλωματική εργασίας υλοποιήθηκαν τέσσερα διαφορετικά μοντέλα ανίχνευσης εισβολής αυξανόμενης πολυπλοκότητας με χρήση Δένδρων Απόφασης, Μηχανής Υποστήριξης Διανυσμάτων (με και χωρίς χρήση πυρήνα), απλού feedforward νευρωνικού δικτύου με ένα κρυφό επίπεδο και ενός πιο εξελιγμένου νευρωνικού δικτύου τύπου GAN με σκοπό την μεταξύ τους σύγκριση.

Η γλώσσα προγραμματισμού που επελέγη ήταν η Python[26]. Η Python είναι μια γλώσσα γενικού σκοπού υψηλού επιπέδου η οποία σχεδιάστηκε με στόχο τον ευανάγνωστο κώδικα με αποτέλεσμα να είναι ιδιαιτέρως δημοφιλής στον τομέα της Μηχανικής Μάθησης και Εξόρυξης Δεδομένων. Απολαμβάνει μιας μεγάλης και ενεργής κοινότητας και ως εκ τούτου συνοδεύεται από μια πλούσια συλλογή βιβλιοθηκών στο αντικείμενο της μηχανικής μάθησης όπως scikit-learn, TensorFlow, Keras και PyTorch. Η δημοτικότητα της επίσης συνέβαλε στην ανάπτυξη πολλών διαδικτυακών και μη βοηθημάτων, καθιστώντας ακόμη πιο εύκολη την διαδικασία εκμάθησης της.

Τα τρία πρώτα μοντέλα της εργασίας έγιναν με χρήση του πακέτου scikit-learn ενώ το τελευταίο με χρήση του πακέτου PyTorch. Τα δεδομένα που χρησιμοποιήθηκαν για την εκπαίδευση των μοντέλων είναι τα σύνολα δεδομένων CICIDS-2017 και UNSW-NB15 καθώς επίσης μία ένωση αυτών. Η εισαγωγή των δεδομένων έγινε με χρήση της βιβλιοθήκης Pandas σε dataframes. Για την υλοποίηση του συστήματος έγινε χρήση της πλατφόρμας διαδραστικού υπολογισμού JupyterLab και της διανομής Miniconda.

[26] https://www.python.org/

4.2 Προσέγγιση που ακολουθήθηκε

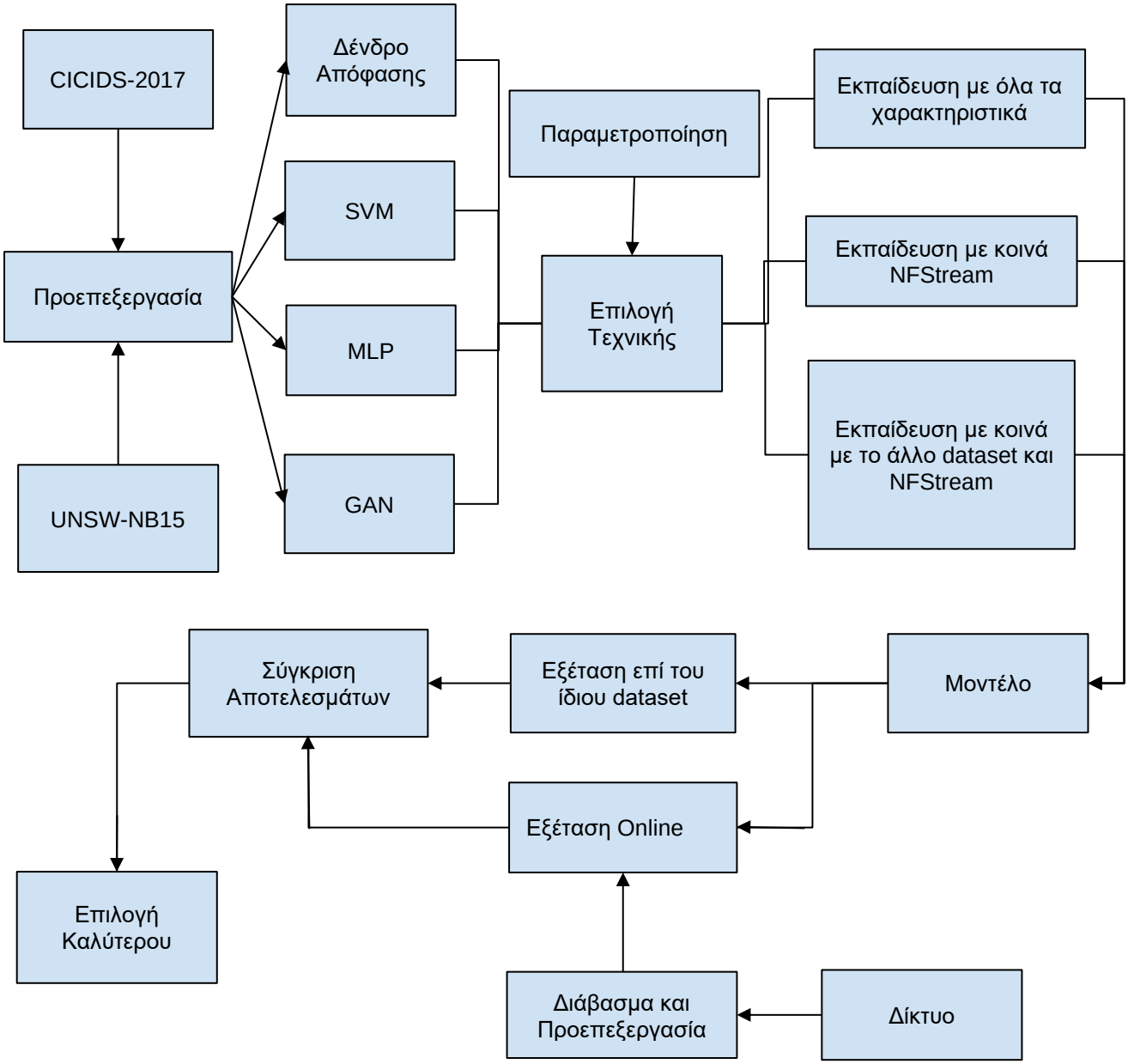
4.2.1 Σύντομη περιγραφή

Το πρώτο βήμα στην ανάπτυξη ενός Online IDS με μηχανική μάθηση είναι η δημιουργία ενός offline μοντέλου με βάση κάποιο dataset. Αφού γίνει η επιλογή του dataset, πρέπει να γίνει μια προεπεξεργασία των δεδομένων και μια επιλογή του αλγόριθμου που θα χρησιμοποιηθεί για τον διαχωρισμό των δεδομένων σε κλάσεις. Ύστερα πρέπει να εξεταστεί και αποφασιστεί η βέλτιστη παραμετροποίηση του μοντέλου. Το εκπαιδευμένο μοντέλο μετά μπορεί να εξεταστεί αν αναγνωρίζει επιτυχώς τις κλάσεις σε δεδομένα του ίδιου dataset που εκπαιδεύτηκε αλλά δεν έχει ξαναδεί. Αν όλα φαίνεται να δουλεύουν καλά εκεί, τότε το μοντέλο αποθηκεύεται και περνάμε στο επόμενο βήμα που είναι η δημιουργία ενός προγράμματος που διαβάζει τη ροή δεδομένων ενός δικτύου υπό εξέταση. Το πρόγραμμα αυτό θα πρέπει να μπορεί να προεπεξεργάζεται τα δεδομένα σε πραγματικό χρόνο και να τα μετασχηματίζει σε μια μορφή που να είναι συμβατή με τα δεδομένα που εκπαιδεύτηκε το μοντέλο σύμφωνα με το dataset. Μετά την επεξεργασία τα δικτυακά δεδομένα «περνάνε» ως είσοδος στο αποθηκευμένο μοντέλο και περιμένουμε την απάντηση. Τα σύνολα δεδομένων CICIDS-2017 και UNSW-NB15 επιλέχθηκαν εκτός του ότι αποτελούν μερικά τα πιο σύγχρονα και ευρέως χρησιμοποιημένα datasets, επειδή παρουσιάζουν πολλά χαρακτηριστικά που είναι εύκολο να εξαχθούν απ’ τα δεδομένα ροής μέσω του NFStream που περιγράφεται στην υποενότητα με τα εργαλεία και βιβλιοθήκες που χρησιμοποιήθηκαν.

Η εκπαίδευση των μοντέλων έγινε πάνω:

- Τα σύνολα δεδομένων με όλα τα χαρακτηριστικά.
- Τα σύνολα δεδομένων με κατάλληλη επιλογή και εξαγωγή χαρακτηριστικών που να είναι εύκολα προσβάσιμα σε ροές κίνησης σε πραγματικό χρόνο μέσω του NFStream.
- Στο σύνολο δεδομένων ένωσης

Η αξιολόγηση έγινε σύμφωνα αφενός με την αποτελεσματικότητα πάνω στα ίδια τα σύνολα δεδομένων επί των οποίων πραγματοποιήθηκε η εκπαίδευση, αφετέρου το online μοντέλο σε πραγματικές συνθήκες. Η διαδικασία περιγράφεται εποπτικά στην Εικόνα 4 που ακολουθεί.



Εικόνα 4: Επισκόπηση προσέγγισης που ακολουθήθηκε

4.2.2 Περιγραφή αρχιτεκτονικής μοντέλων

Παρακάτω παρέχεται μια σύντομη περιγραφή του τρόπου λειτουργίας των τεχνικών που χρησιμοποιήθηκαν. Για περεταίρω πληροφορίες ο αναγνώστης παραπέμπεται στο Παράρτημα.

4.2.2.1 Δένδρο Απόφασης

Ένα δένδρο απόφασης αποτελείται από μια απλή δεντρική δομή δεδομένων όπου κάθε κόμβος αναπαριστά μια απόφαση βασισμένη σε κάποιο χαρακτηριστικό, και κάθε ακμή ή κλάδος αναπαριστά τα πιθανά αποτελέσματα ή τιμές αυτής της απόφασης. Σε κάθε βήμα της διαδικασίας εκπαίδευσης ο αλγόριθμος χωρίζει τα δεδομένα σύμφωνα με το χαρακτηριστικό που παρέχει το μεγαλύτερο κέρδος πληροφορίας (information gain). Τα δένδρα απόφασης μπορούν να χρησιμοποιηθούν και για σκοπούς ταξινόμησης και παλινδρόμησης, ενώ συγκαταλέγονται στις πιο ερμηνεύσιμες τεχνικές μηχανικής μάθησης.

4.2.2.2 Μηχανή Υποστήριξης Διανυσμάτων

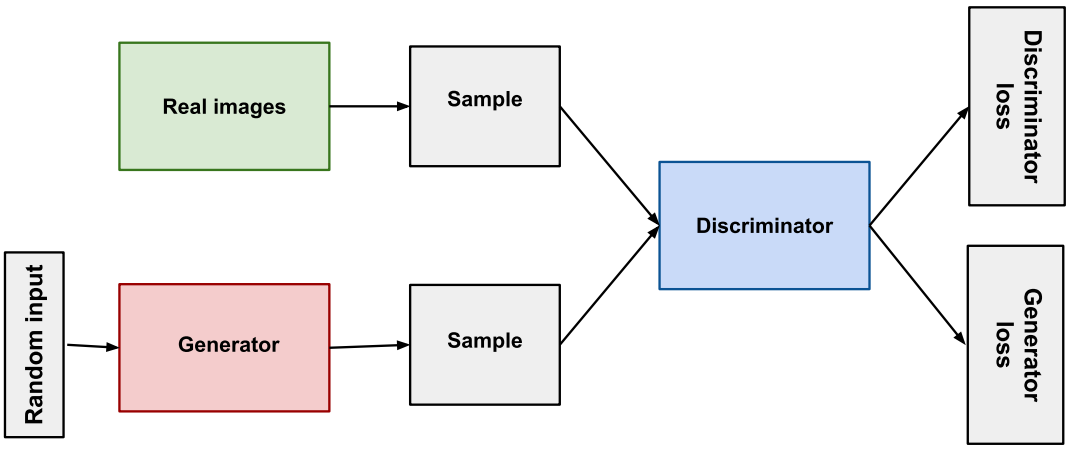
Οι Μηχανές Υποστήριξης Διανυσμάτων λειτουργούν με την εύρεση ενός υπερεπιπέδου σε έναν διανυσματικό χώρο πολλών διαστάσεων το οποίο διαχωρίζει όσο περισσότερο γίνεται τα δεδομένα. Τα SVMs μπορούν να χειριστούν μη γραμμικά όρια απόφασης με την χρήση συναρτήσεων πυρήνα και την απεικόνιση των δεδομένων σε έναν χώρο περισσότερων διαστάσεων όπου ο διαχωρισμός είναι πιο εύκολος. Λειτουργούν καλά σε προβλήματα πολλών χαρακτηριστικών και θεωρούνταν απ’ τις πιο αξιόπιστες τεχνικές πριν την «έκρηξη» της βαθιάς μάθησης την δεκαετία του 2010.

4.2.2.3 Perceptron πολλαπλών επιπέδων

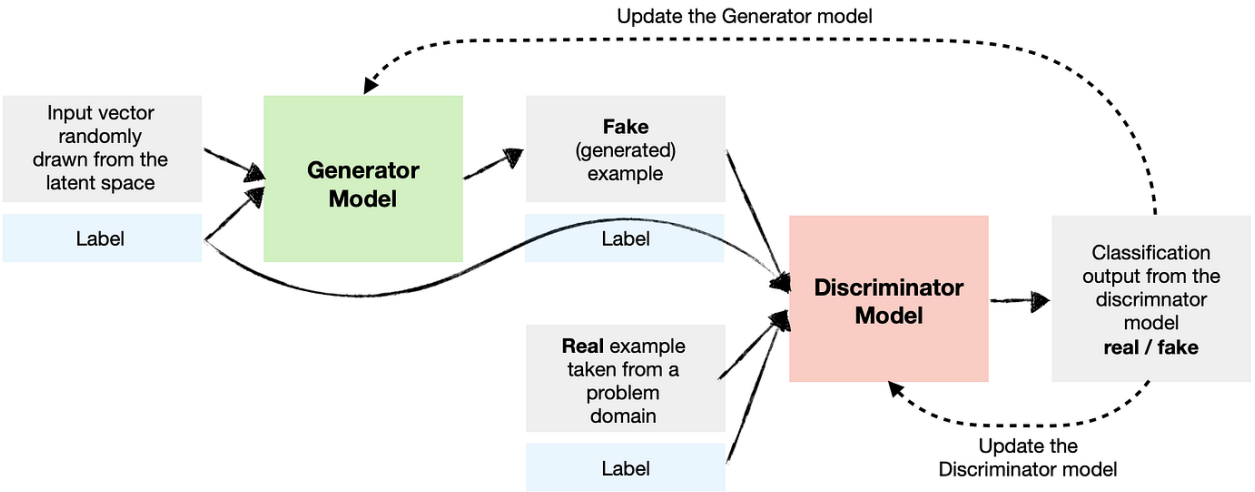
Το Perceptron πολλαπλών επιπέδων, ή στα αγγλικά Multilayer perceptron (MLP), είναι ένας τύπος εμπροσθοτροφοδοτούμενου (feed-forward) νευρωνικού δικτύου. Αν και θεωρητικά μπορεί να αποτελείται από ένα αυθαίρετο αριθμό κρυφών επιπέδων, στην παρούσα εργασία χρησιμοποιήθηκε ως αντιπροσωπευτικό των ρηχών νευρωνικών δικτύων και αποτελείται από ένα κρυφό επίπεδο. Κάθε επίπεδο γενικά αποτελείται από έναν αριθμό κόμβων ή νευρώνων με την ικανότητα υπολογισμού ενός βαθμισμένου αθροίσματος των εισόδων τους, την εφαρμογή μιας συνάρτησης ενεργοποίησης στο αποτέλεσμα και την μεταφορά των εξόδων τους στο επόμενο επίπεδο. Μπορούν να χρησιμοποιηθούν για σκοπού ταξινόμησης και στην περίπτωση αυτή οι κόμβοι στο τελευταίο επίπεδο αντιστοιχούν στις διάφορες κλάσεις.

4.2.2.4 GAN

Ένα GAN (Generative Adversarial Networks ή στα ελληνικά Αναγεννητικά Ανταγωνιστικά Δίκτυα) αποτελείται από ένα ζεύγος νευρωνικών δικτύων, τον Γεννήτορα και τον Διευκρινιστή (στα αγγλικά Generator και Discriminator αντίστοιχα). Ο Διευκρινιστής εκπαιδεύεται δεχόμενος ως είσοδο ζεύγη πραγματικών δεδομένων απ' τα δεδομένα εκπαίδευσης και συνθετικών δεδομένων παραγόμενων απ' τον Γεννήτορα και παράγει ως έξοδο έναν αριθμό ανάμεσα σε ένα εύρος τιμών που θα υποδηλώνει αν πιστεύει ότι τα δεδομένα του Γεννήτορα βρίσκονται «κοντά» στα δεδομένα εκπαίδευσης, άρα είναι πραγματικά, ή θα τα αναγνωρίζει για αυτό που είναι, δηλαδή «ψεύτικα». Κατ' ουσίαν δηλαδή εκπαιδεύεται ως ταξινομητής. Ο Γεννήτορας από την άλλη πλευρά εκπαιδεύεται έμμεσα μέσω των εξόδων του διευκρινιστή, δεχόμενος ως είσοδο ένα διάνυσμα τυχαίων αριθμών από μια γνωστή κατανομή, και προσπαθώντας να παράξει εξόδους που να ξεγελούν τον διευκρινιστή ότι τα παραγόμενα δεδομένα του είναι πραγματικά. Πρακτικά, τα δύο νευρωνικά δίκτυα αντιπαλεύονται το ένα με το άλλο γιατί εκπαιδεύονται προσπαθώντας να ελαχιστοποιήσουν απώλειες που προκύπτουν από αντίθετες συναρτήσεις απωλειών σε ένα παιχνίδι μηδενικού κέρδους όπου όσο το ένα τα πηγαίνει καλύτερα, αυτό γίνεται αναγκαστικά εις βάρος του άλλου. Η σύγκλιση των δικτύων είναι ιδιαίτερα ασταθής και μετά από κάποιο σημείο εκπαίδευσης η απόδοση πέφτει οπότε πρέπει να γίνει κατάλληλη επιλογή υπερπαραμέτρων. Στο πλαίσιο της εργασίας έχει υλοποιηθεί ένα binary conditional GAN (cGAN) που συμπεριλαμβάνει την πληροφορία των ετικετών, benign ή επίθεσης, στον υπολογισμό της συνάρτησης κόστους. Η κλασική αρχιτεκτονική ενός GAN δικτύου όπως προτάθηκε απ' τους Goodfellow et al. [61] απεικονίζεται στην Εικόνα 5. Στην εκδοχή που υλοποιήθηκε στην εργασία ο Διευκρινιστής εκπαιδεύεται εκτός απ' το να ξεχωρίζει πραγματικά από ψεύτικα παραδείγματα, να ξεχωρίζει επιπλέον παραδείγματα κανονικής λειτουργίας και ενδεχόμενης επίθεσης. Επίσης ο Γεννήτορας είναι σε θέση να παράγει συνθετικά δεδομένα κανονικής λειτουργίας ή επίθεσης κατά βούληση. Η αρχιτεκτονική ενός cGAN φαίνεται στην Εικόνα 6.



Εικόνα 5: Αρχιτεκτονική GAN[27]



Εικόνα 6: Αρχιτεκτονική cGAN[28]

[27] https://developers.google.com/machine-learning/gan/gan_structure
[28] https://towardsdatascience.com/cgan-conditional-generative-adversarial-network-how-to-gain-control-over-gan-outputs-b30620bd0cc8

4.2.3 Προεπεξεργασία Δεδομένων

4.2.3.1 Δημιουργία μοντέλων με το CICIDS 2017

Το CICIDS2017 όπως αναφέραμε στην ενότητα των συνόλων δεδομένων αποτελείται από δύο φακέλους εκ των οποίων κατάλληλος για χρήση σε εφαρμογές μηχανικής μάθησης είναι ο φάκελος με τον ομώνυμο τίτλο (MachineLearningCVE). Σε αυτόν περιλαμβάνονται οκτώ αρχεία csv. Έγινε συνένωση σε ένα ενιαίο αρχείο με όλες τις καταγραφές.

Για την ανάπτυξη των μοντέλων χρησιμοποιήθηκαν τρία διανύσματα χαρακτηριστικών όπως αναφέρθηκε στην νωρίτερα. Το πρώτο διάνυσμα περιλαμβάνει όλα τα χαρακτηριστικά που dataset. Για το δεύτερο έγινε επιλογή χαρακτηριστικών προκειμένου να μπορεί να γίνει αργότερα η αντιστοίχιση με χαρακτηριστικά που προκύπτουν απ' το online μοντέλο που βασίζεται στο NFStream. Αυτό το διάνυσμα χαρακτηριστικών είναι το ακόλουθο;

Πίνακας 2: Διάνυσμα χαρακτηριστικών για το CIC-IDS2017

```
selected_features = ['Destination Port', 'Total Fwd Packets', 'Total
Backward Packets', 'Total Length of Fwd Packets', 'Total Length of Bwd
Packets', 'Fwd Packet Length Max', 'Fwd Packet Length Min', 'Fwd Packet
Length Mean', 'Fwd Packet Length Std', 'Bwd Packet Length Max', 'Bwd
Packet Length Min', 'Bwd Packet Length Mean', 'Bwd Packet Length Std',
'Flow IAT Mean', 'Flow IAT Std', 'Flow IAT Max', 'Flow IAT Min', 'Fwd
IAT Mean', 'Fwd IAT Std', 'Fwd IAT Max', 'Fwd IAT Min', 'Bwd IAT Mean',
'Bwd IAT Std', 'Bwd IAT Max', 'Bwd IAT Min', 'Min Packet Length', 'Max
Packet Length', 'Packet Length Mean', 'Packet Length Std', 'FIN Flag
Count', 'SYN Flag Count', 'RST Flag Count', 'PSH Flag Count', 'ACK Flag
Count', 'URG Flag Count', 'CWE Flag Count', 'ECE Flag Count', 'Fwd PSH
Flags', 'Fwd URG Flags', 'Bwd PSH Flags', 'Bwd URG Flags', 'Flow
Duration', 'Flow Bytes/s', 'Flow Packets/s', 'Fwd Packets/s', 'Bwd
Packets/s', 'Packet Length Variance']
```

Για την εκπαίδευση και επαλήθευση των μοντέλων που βασίζονται στο CICIDS-2017 έγινε εισαγωγή του ενιαίου csv αρχείου σε DataFrame, κατόπιν δειγματοληψία με την συνάρτηση sample του pandas και δημιουργήθηκαν δύο αρχεία: cicids_sample.csv και cicids_test_datasetv1.csv με παραμέτρους frac=0.05 και frac=0.1 αντίστοιχα. Για την αναπαραγωγησιμότητα της δειγματοληψίας επιλέχθηκε αυθαίρετα η τιμή παραμέτρου random_state=1 και random_state=100 αντίστοιχα. Στην συνέχεια έγινε μια δευτερογενής επεξεργασία και δημιουργήθηκαν δύο νέα αρχεία cic_train_sample_binary.csv και

cic_test_sample_binary.csv όπου όλες οι κατηγορίες επιθέσεων συμπτύχθηκαν σε μία. Επίσης σε αυτό το στάδιο πραγματοποιήθηκε η αντικατάσταση των τιμών απείρου np.inf και -np.inf με np.nan. Τέλος, ας σημειωθεί ότι έγινε μια εξέταση για την απόδοση των μοντέλων σε σύνολα δεδομένων με και χωρίς υπερδειγματοληψία για την εξισορρόπηση των κλάσεων. Οι τρόποι δειγματοληψίας που εξετάστηκαν ήταν random oversampling και SMOTE μέσω της βιβλιοθήκης Imbalanced-learn. Τα αποτελέσματα έδειξαν ότι υπήρχε μια μικρή μείωση της απόδοσης με υπερδειγματοληψία, η δε διαφορά απόδοσης μεταξύ random oversampling και SMOTE ήταν αμελητέα. Ως εκ τούτου, επιλέχθηκε η εργασία χωρίς δειγματοληψία.

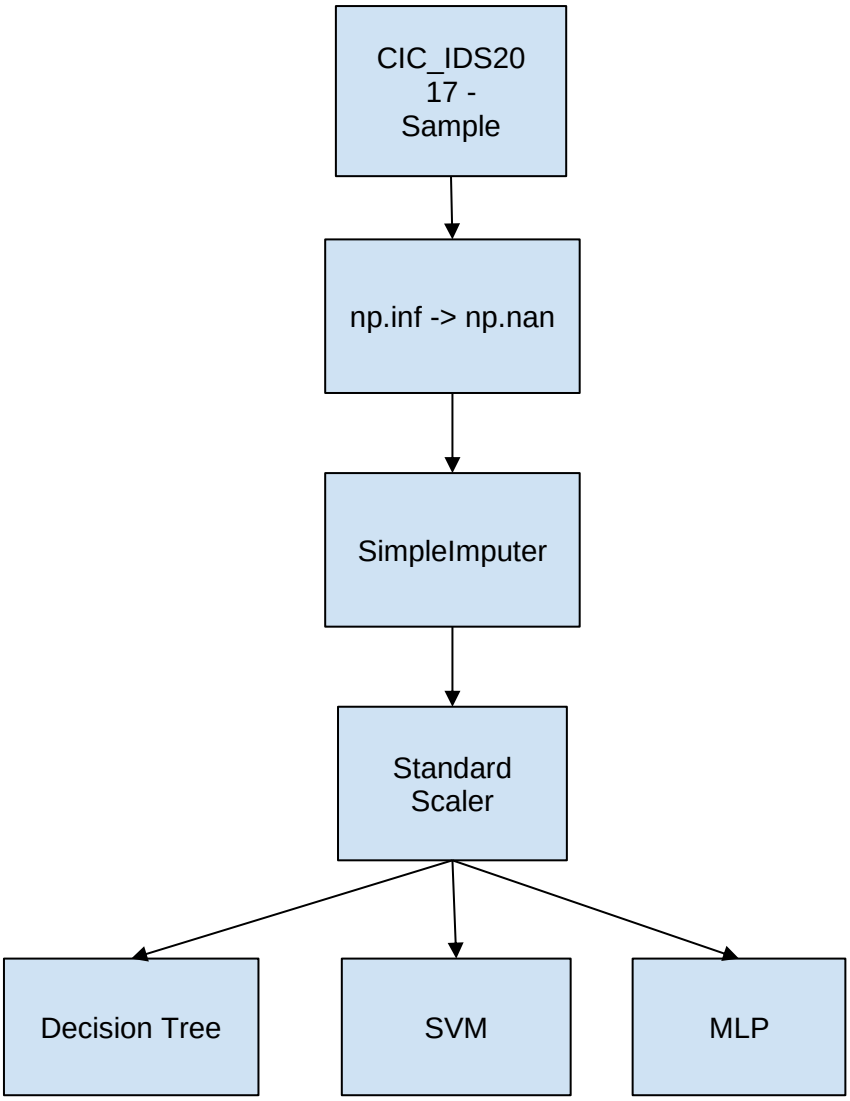
Η διαδικασία προεπεξεργασίας περιλαμβάνει:

1. Δημιουργία αντικειμένου της κλάσης SimpleImputer και αντικατάσταση των κενών τιμών με την μέση τιμή της αντίστοιχης στήλης.
2. Δημιουργία αντικειμένου της κλάσης StandardScaler και εκπαίδευση και μετασχηματισμός των δεδομένων εκπαίδευσης. Επίσης αποθήκευση του εκπαιδευμένου scaler για μελλοντική χρήση στο online μοντέλο.

Ύστερα έγινε εισαγωγή των επεξεργασμένων δεδομένων στον ταξινομητή επιλογής. Ή όλη διαδικασία για την περίπτωση των ρηχών μοντέλων υλοποιήθηκε με την χρησιμοποίηση της συνάρτησης make_pipeline του scikit-learn για την δημιουργία τριών pipelines για το Δένδρο Απόφασης, SVM, και MLP όπως φαίνεται αμέσως μετά αλλά και σχηματικά στην Εικόνα 7:

Πίνακας 3: Pipelines

```
pipe1 = make_pipeline(
    SimpleImputer(), StandardScaler(), DecisionTreeClassifier()
)
pipe2 = make_pipeline(
    SimpleImputer(), StandardScaler(), LinearSVC()
)
pipe3 = make_pipeline(
    SimpleImputer(), StandardScaler(), MLPClassifier()
)
```



Εικόνα 7: Pipelines ρηχών μοντέλων

Για την δημιουργία του μοντέλου βασισμένο σε GAN χρησιμοποιήθηκε το framework PyTorch.

Για το διάβασμα των δεδομένων και την μετατροπή τους στην κατάλληλη μορφή ακολουθήθηκε η εξής διαδικασία:

- Δημιουργία μιας κλάσης MyDataset που κληρονομεί απ' την κλάση Dataset του PyTorch και δέχεται ως είσοδο το csv εκπαίδευσης και μιας λίστας χαρακτηριστικών και υλοποιεί τις μεθόδους __len()__ και __getitem__() της κλάσης Dataset.
- Η κλάση MyDataset εσωτερικά στην μέθοδο __init__() κάνει χρήση μιας άλλης κλάσης PreProcessor που πραγματοποιεί την προεπεξεργασία των δεδομένων.
- Η κλάση PreProcessor υλοποιεί αντίστοιχη προεπεξεργασία με τα ρηχά μοντέλα.

- Το αντικείμενο της κλάσης MyDataset γίνεται εισαγωγή ως παράμετρος στην κλάση DataLoader του PyTorch. Η κλάση DataLoader επιστρέφει ένα αντικείμενο επί του οποίου θα γίνει η διάσχιση των δεδομένων.

Για την εξέταση των GAN-based μοντέλων χρησιμοποιήθηκε το ίδιο σύνολο ελέγχου με τα ρηχά μοντέλα (cic_test_sample_binary.csv), αλλά με μια επιπλέον δειγματοληψία με frac=0.1 για έναν πιο γρήγορο έλεγχο των αποτελεσμάτων.

4.2.3.2 Δημιουργία μοντέλων με το UNSW-NB15

Η ίδια διαδικασία συνένωσης των αρχείων csv επαναλήφθηκε με τα τέσσερα csv αρχεία του UNSW-NB15. Στην περίπτωση του UNSW-NB15 η προεπεξεργασία των δεδομένων είναι ελαφρώς μεγαλύτερη. Αρχικά το διάνυσμα των χαρακτηριστικών επιλογής είναι το ακόλουθο:

Πίνακας 4: Διάνυσμα χαρακτηριστικών για το UNSW-NB15

```
selected_features_nb15 = ['Dsport', 'dur', 'Spkts', 'Dpkts', 'sbytes',
'dbytes', 'smeansz', 'dmeansz', 'flow_bytes/s', 'flow_packets/s',
'fwd_packets/s', 'bwd_packets/s']
```

Όπως φαίνεται, τα χαρακτηριστικά που είναι εύκολο να προκύψουν απ' το NFStream και να αντιστοιχηθούν σε χαρακτηριστικά του UNSW-NB15 είναι σαφώς λιγότερα. Πρέπει να τονιστεί ότι το χαρακτηριστικό 'dur' που περιγράφει τον χρόνο της ροής, και χρησιμοποιείται για τον καθορισμό αρκετών χαρακτηριστικών, θεωρήθηκε απουσία άλλων στοιχείων ότι βρίσκεται στην κλίμακα των milliseconds.

Επιπρόσθετα, τα τέσσερα τελευταία χαρακτηριστικά που αποτελούν ρυθμούς μετάδοσης πακέτων και bytes είναι σύνθετα και υπολογίστηκαν από απλούστερα χαρακτηριστικά του UNSW-NB15 ως εξής:

Πίνακας 5: Εξαγωγή χαρακτηριστικών απ' το UNSW-NB15

```
nb_selected_df['flow_bytes/s'] = (full_nb15_dataset['sbytes'] +
full_nb15_dataset['dbytes']) / (full_nb15_dataset['dur'] / 10**3)

nb_selected_df['flow_packets/s'] = (full_nb15_dataset['Spkts'] +
full_nb15_dataset['Dpkts']) / (full_nb15_dataset['dur']  / 10**3)

nb_selected_df['fwd_packets/s'] = full_nb15_dataset['Spkts'] /
```

```
(full_nb15_dataset['dur'] / 10**3)

nb_selected_df['bwd_packets/s'] = full_nb15_dataset['Dpkts'] /
(full_nb15_dataset['dur'] / 10**3)
```

Επίσης αποκλείστηκαν κάποιες σειρές όπου η στήλη του χαρακτηριστικού dsport δεν έχει αριθμητική τιμή, και έγινε αντικατάσταση της στήλης ετικέτας από 0 και 1 σε BENIGN και ATTACK αντίστοιχα.
Στην περίπτωση με όλα (σχεδόν) τα χαρακτηριστικά έγιναν οι εξής αλλαγές:

- Κατάργηση των στηλών srcip, dstip και attack_cat
- Κατάργηση των σειρών όπου η τιμή των dsport, sport δεν είχε αριθμητικές τιμές
- Κατάργηση της στήλης ct_ftp_cmd επειδή περιλαμβάνει μία τιμή με string το λευκό κενό που μάλιστα εμφανίζεται περισσότερο από όλα τα υπόλοιπα και που δεν είναι σαφές τι προσδιορίζει

Στο τέλος αυτής της διαδικασίας δημιουργήθηκαν δύο αρχεία nb_sample_almost_all_feat.csv και nb_sample_selected.csv.

Σε επόμενη φάση, με βάση αυτά τα δύο αρχεία έγινε αφενός αντικατάσταση των τιμών απείρου με τιμές nan, αφετέρου για την περίπτωση του αρχείου με τα περισσότερα χαρακτηριστικά έγινε binary encoding για τα χαρακτηριστικά proto, state και service με την συνάρτηση get_dummies() του pandas.
Τα αποτελέσματα αποθηκεύτηκαν σε δύο ομάδες δεδομένων train και test
nb_all_feat_train_dataset.csv και nb_all_feat_test_dataset.csv
nb_12_feat_train_dataset.csv και nb_12_feat_test_dataset.csv
Για την δημιουργία αυτών των αρχείων έγινε χρήση της συνάρτησης train_test_split του scikit-learn με τιμή παραμέτρου test_size=0.2. και random_state=42.

4.2.3.3 Δημιουργία μοντέλων με την ένωση των datasets

Προκειμένου η σύνθεση των δύο διαφορετικών datasets να γίνει εφικτή, κατασκευάζεται ένα διάνυσμα κοινών χαρακτηριστικών στα δύο σύνολα δεδομένων επί των οποίων θα γίνει η επεξεργασία.

Το διάνυσμα αυτό απ' την πλευρά του CICIDS2017 είναι:

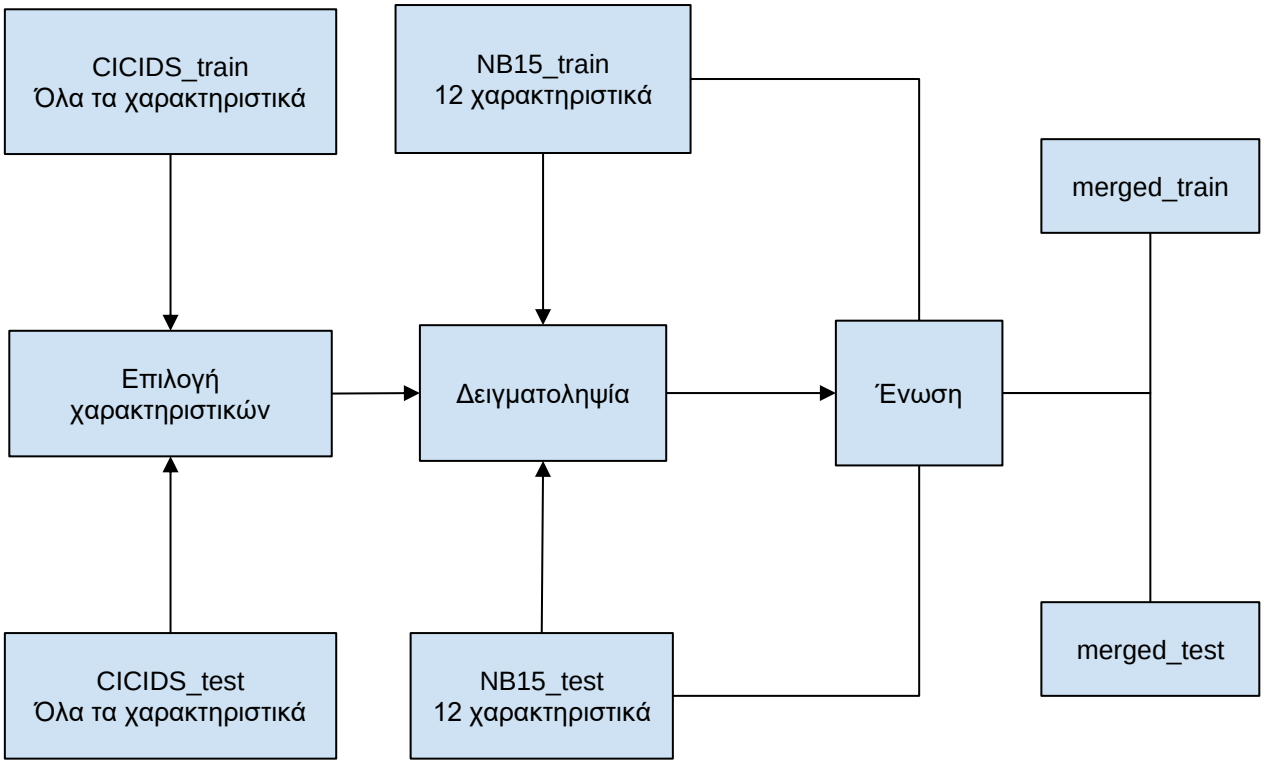
Πίνακας 6: Διάνυσμα 12 χαρακτηριστικών για το CIC-IDS2017

```
selected_features_cic = ['Destination Port', 'Flow Duration', 'Total Fwd
Packets', 'Total Backward Packets', 'Total Length of Fwd Packets',
'Total Length of Bwd Packets', 'Fwd Packet Length Mean', 'Bwd Packet
Length Mean', 'Flow Bytes/s', 'Flow Packets/s', 'Fwd Packets/s', 'Bwd
Packets/s']
```

Ενώ απ' την πλευρά του UNSW-NB15 έχει περιγραφεί στην προηγούμενη υποενότητα.

Υπάρχουν ακόμη πιθανότατα επί πλέον δύο κοινά χαρακτηριστικά, τα 'Fwd IAT Total', 'Bwd IAT Total' απ' το CIC-IDS2017 και τα 'Sintpkt', 'Dintpkt' απ' το UNSW NB15 IDS, ωστόσο εν απουσία περισσότερων στοιχείων για τα χαρακτηριστικά του UNSW-NB15, δεν γίνεται χρήση τους.

Σχηματικά στην Εικόνα 8 φαίνεται η διαδικασία που ακολουθήθηκε για την σύνθεση των δύο συνόλων δεδομένων.



Εικόνα 8: Δημιουργία ενιαίου συνόλου δεδομένων

Ακολουθεί αναλυτικά η διαδικασία προεπεξεργασίας που ακολουθήθηκε:

1. Διάβασμα των αρχείων που χρησιμοποιήθηκαν για την εκπαίδευση με το CICIDS-2017 και UNSW-NB15. Συγκεκριμένα του αρχείου για την εκπαίδευση με το UNSW-NB15 με τα 12 χαρακτηριστικά, και του CICIDS με όλα τα χαρακτηριστικά. Στα σύνολα αυτά έχει ήδη γίνει η αφαίρεση πιθανών τιμών απείρου και αντικατάσταση τους με NaN.
2. Επιλογή χαρακτηριστικών του CICIDS με τα αντίστοιχα του UNSW-NB15.
3. Δειγματοληψία έτσι ώστε τα δύο υποσύνολα εκπαίδευσης προερχόμενα απ' τα CICIDS και UNSW-NB15 να έχουν περίπου το ίδιο πλήθος εγγραφών. Το ίδιο και για τα δύο υποσύνολα ελέγχου.
4. Ενσωμάτωση των δύο υποσυνόλων εκπαίδευσης και ελέγχου σε ένα σύνολο εκπαίδευσης και ελέγχου αντίστοιχα.

Κατόπιν ακολουθήθηκε η ίδια διαδικασία με το προηγούμενο σύνολο δεδομένων, δηλαδή: Είσοδος αυτών των διανυσμάτων στα pipelines εκπαίδευσης με τα εξής στάδια: SimpleImputer(), StandardScaler() και ο τελικός ταξινομητής, δηλαδή

DecisionTreeClassifier(), LinearSVC() ή MLPClassifier() για την περίπτωση του Δέντρου Απόφασης, γραμμικού πυρήνα SVM ή του απλού νευρωνικού δικτύου αντίστοιχα.

Για το μοντέλο που βασίστηκε στο GAN ακολουθήθηκε η ίδια προεπεξεργασία με τα ρηχά μοντέλα και υλοποιήθηκε με τον ίδιο τρόπο όπως στα μοντέλα που βασίστηκαν στα μεμονωμένα datasets.

4.2.4 Παραμετροποίηση Συστημάτων

Η βέλτιστη παραμετροποίηση των μοντέλων μηχανικής μάθησης είναι γενικά μια χρονοβόρα διαδικασία. Στην παρούσα εργασία αποφεύχθηκε μια εξαντλητική εξερεύνηση όλων των υπερπαραμέτρων και αντί αυτού η προσοχή εστιάστηκε σε μια μικρή ομάδα μερικών απ' των πιο σημαντικών υπερπαραμέτρων. Τα ρηχά μοντέλα απ' την φύση τους παρουσιάζουν λιγότερες υπερπαραμέτρους σε σχέση με τα βαθιά μοντέλα όπως το GAN που υλοποιήθηκε. Επίσης η εξέταση των ρηχών μοντέλων μπορεί να γίνει με την βοήθεια έτοιμων συναρτήσεων του πακέτου scikit-learn.

Γενικά οι πιο συχνοί τρόποι tuning των μοντέλων είναι[29]:

- Grid Search (εξαντλητική εξερεύνηση)
- Random Grid Search
- Μπαγεσιανή Βελτιστοποίηση (Bayesian Optimization)
- Εξελικτικοί Αλγόριθμοι

Στα ρηχά μοντέλα όπου το εύρος των πιθανών τιμών υπερπαραμέτρων εξέτασης ήταν πολύ μικρό και ο χρόνος εκπαίδευσης των μοντέλων επέτρεπε κάτι τέτοιο, επιλέχθηκε η εξαντλητική εξερεύνηση, διαφορετικά έγινε χρήση Τυχαίας Αναζήτησης σε συνδυασμό με δειγματοληψία.

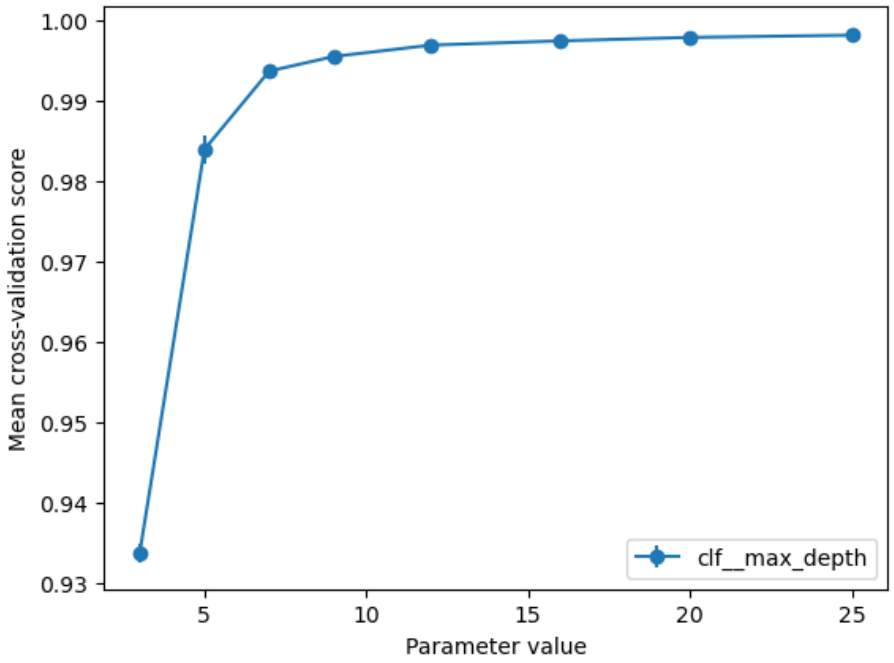
Στα GAN μοντέλα ο τρόπος βελτιστοποίησης είναι εν γένει εμπειρικός. Σε αυτό το πλαίσιο εξετάστηκαν διάφορες υπερπαράμετροι, αρχιτεκτονικές δικτύων και τρόπων εκπαίδευσης μέχρι την κατάληξη ενός τελικού που αποδίδει τα καλύτερα αποτελέσματα και είναι πιο σταθερό.

[29] https://analyticsindiamag.com/top-8-approaches-for-tuning-hyperparameters-of-machine-learning-models/

4.2.4.1 Παραμετροποίηση μοντέλων με το CICIDS-2017

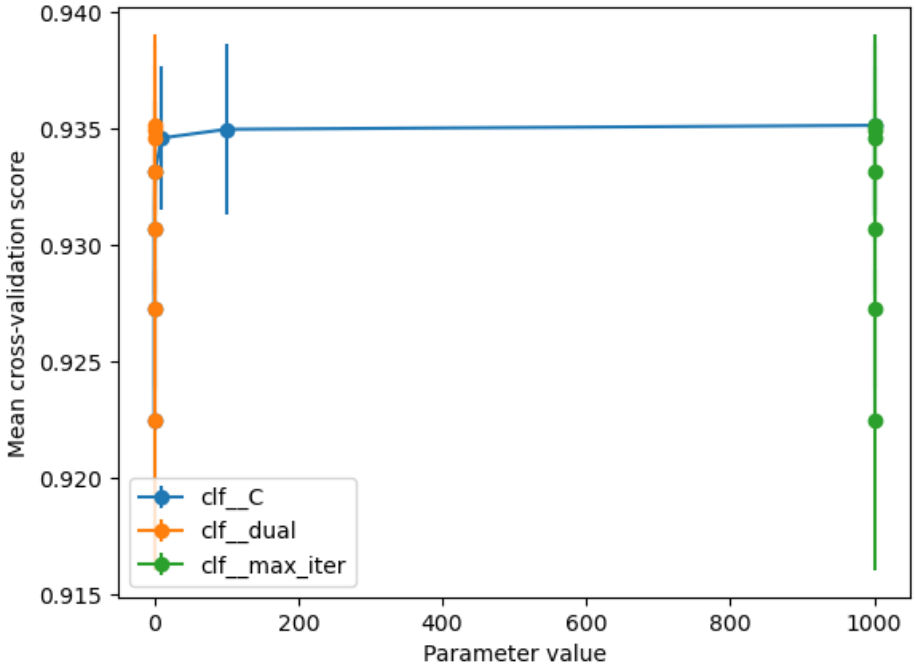
Για την περίπτωση του Δέντρου Απόφασης έγινε μια απλή εξερεύνηση της υπερπαραμέτρου για το μέγιστο βάθος του δέντρου με την συνάρτηση GridSearchCV() του scikit-learn που υλοποιεί grid search και 5-fold cross validation. Εξετάστηκαν και κάποιες άλλες παράμετροι αλλά δεν έδειξαν κάποια διαφοροποίηση απ' την περίπτωση των προεπιλεγμένων τιμών οπότε αφέθηκαν ως έχει.

Το μοντέλο του δέντρου απόφασης για το σύνολο δεδομένων CICIDS-2017 για την εκπαίδευση με όλα τα δεδομένα δέχθηκε ως παράμετρο το max_depth=8 για την αποφυγή overfitting. Οι τιμές του μέγιστου βάθους που εξετάστηκαν ήταν οι 3, 5, 7, 9, 12, 16, 20 και 25. Το αποτέλεσμα του grid search έδειξε ότι πάνω από βάθος δέντρου 7, η βελτίωση είναι αμελητέα (Εικόνα 9). Για την περίπτωση με επιλογή χαρακτηριστικών τα αποτελέσματα ήταν ταυτόσημα.



Εικόνα 9: Αναζήτηση πλέγματος για το βάθος του δένδρου για το CIC-IDS2017

Το μοντέλο του LinearSVC δέχθηκε ως παραμέτρους τα max_iter=1000, dual=False για την αποφυγή μη σύγκλισης. Γενικότερα η παράμετρος dual=False προτιμάται όταν ο αριθμός των παραδειγμάτων είναι μεγαλύτερος απ' τον αριθμό των χαρακτηριστικών. Επίσης εξετάστηκε η υπερπαράμετρος C (Εικόνα 10) αλλά τα αποτελέσματα δεν έδειξαν διαφοροποίηση οπότε αφέθηκε στην προεπιλεγμένη τιμή.



Εικόνα 10: Αναζήτηση πλέγματος της υπερπαραμέτρου C του γραμμικού SVM με το CIC-IDS2017

Για την εξέταση των υπερπαραμέτρων του μοντέλου του SVM με πολυωνυμικό πυρήνα επιλέχθηκε η μέθοδος RandomizedSearchCV() του scikit-learn. Οι υπερπαράμετροι που εξετάστηκαν ήταν οι C, gamma και kernel. Οι τιμές του gamma που εξετάστηκαν ήταν οι 0.001, 0.01, 0.1 και 1, του C οι 0.1, 1, 10 και 100 ενώ του kernel οι 'linear', 'poly', 'rbf' και 'sigmoid'. Επειδή ο χρόνος εκπαίδευσης του SVM με πολυωνυμικό πυρήνα αυξάνεται γρήγορα (πολυωνυμικά), έγινε μια επιπλέον δειγματοληψία με frac=1/10 του συνόλου εκπαίδευσης. Τα αποτελέσματα για την περίπτωση εκπαίδευσης με όλα τα χαρακτηριστικά ήταν C=0.1, gamma=1, kernel='poly' , ενώ για την περίπτωση της επιλογής των 47 χαρακτηριστικών ήταν C=1, gamma=1, kernel='poly'.

Τέλος το μοντέλο του MLPClassifier πήρε ως παραμέτρους το max_iter=1000 για την αποφυγή warning μη σύγκλισης. Έγινε χρήση της GridSearchCV() για τον αριθμό των νευρώνων στο κρυφό επίπεδο αλλά επειδή η διαφορά απόδοσης ήταν μικρή (Εικόνα 11), αφέθηκε η προεπιλεγμένη τιμή των 100 νευρώνων για λόγους απλότητας και ευκολότερης σύγκρισης μεταξύ των μοντέλων.

Εικόνα 11: Αναζήτηση πλέγματος για το μέγεθος του κρυφού επιπέδου του MLP με το CIC-IDS2017

Για το μοντέλο που βασίζεται στο GAN και λεπτομέρειες σχετικά με τον ορισμό των δικτύων Generator και Discriminator ο αναγνώστης παραπέμπεται στο παράρτημα στο τμήμα κώδικα:

Οι υπερπαράμετροι που επιλέχθηκαν είναι:

Πίνακας 7: Υπερπαράμετροι GAN με το CIC-IDS2017

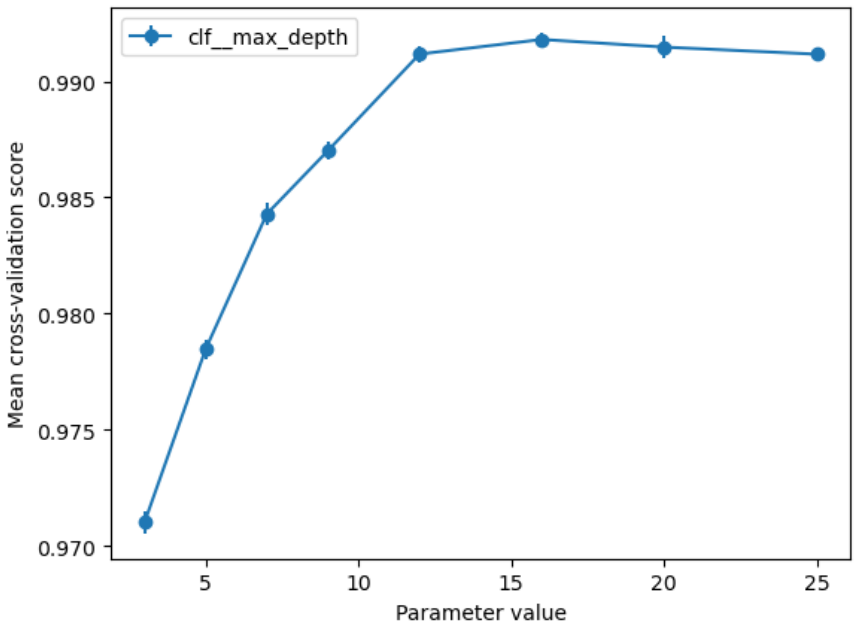
Υπερπαράμετρος	Τιμή
Αριθμός κρυφών επιπέδων Διευκρινιστή	4
Αριθμός νευρώνων Διευκρινιστή	data_dim->40->32->16->8->1
data_dim	78 ή 47
latent_dim	8
Learning Rate	0.0002
betas	(0.5, 0.999)
batch_size	64
Συνάρτηση Κόστους	Binary Cross-Entropy (BCE)
Βελτιστοποιητής	Adam

Έγινε μελέτη για την επιλογή των καταλληλότερου αριθμού εποχών (epochs) εκπαίδευσης. Εξετάστηκαν epochs στο διάστημα 0-100. Συγκεκριμένα σε κάθε επανάληψη του βρόχου εκπαίδευσης αποθηκεύτηκαν οι παράμετροι των νευρωνικών δικτύων σε ξεχωριστά αρχεία σε έναν φάκελο, με τα βασικότερα χαρακτηριστικά τους να περιλαμβάνονται στο όνομα του αρχείου για ευκολότερη εύρεση.

4.2.4.2 Παραμετροποίηση μοντέλων με το UNSW-NB15

Για το Δένδρο Απόφασης έγινε επανάληψη της ίδιας διαδικασίας με τα μοντέλα που βασίζονται στο CICIDS-2017, δηλαδή εξετάστηκε το βέλτιστο βάθος του δέντρου με την μέθοδο του Grid Search για τις ίδιες τιμές όπως και στο CICIDS-2017.

Τα αποτελέσματα της αναζήτησης πλέγματος ήταν ταυτόσημα για τις δύο περιπτώσεις, δηλαδή την εκπαίδευση με και χωρίς επιλογή χαρακτηριστικών. Το βέλτιστο βάθος βρέθηκε ότι ήταν το 16 αλλά με σχεδόν μηδαμινή διαφορά απ' το 12 (Εικόνα 12), οπότε επιλέχθηκε το 12.



Εικόνα 12: Αναζήτηση πλέγματος για το βάθος του δένδρου με το UNSW-NB15

Για την περίπτωση του SVM με πολυωνυμικό πυρήνα, ακολουθήθηκε η ίδια διαδικασία με το αντίστοιχο μοντέλο που βασίζεται στο σύνολο δεδομένων CICIDS-2017, δηλαδή έγινε χρήση της μεθόδου RandomizedSearchCV() και εξέταση των υπερπαραμέτρων C, gamma και kernel. Τα αποτελέσματα για την περίπτωση της επιλογής χαρακτηριστικών ήταν C=100, gamma=1, kernel='rbf', ενώ για την περίπτωση με όλα τα χαρακτηριστικά ήταν kernel='rbf', gamma=0.001 και C=1. Για την μείωση του χρόνου υπολογισμού των βέλτιστων παραμέτρων, χρησιμοποιήθηκε και εδώ δειγματοληψία του συνόλου εκπαίδευσης μέσω της συνάρτησης sample(), αυτή την φορά με παράμετρο frac=1/20 λόγω του μεγαλύτερο αριθμού εγγραφών.

Για το MLP όπως και στην προηγούμενη περίπτωση εξετάστηκε μόνο ο αριθμός των νευρώνων στο κρυφό επίπεδο και όπως και προηγουμένως αφέθηκε στην προεπιλεγμένη τιμή των 100 νευρώνων επειδή η διαφορά στην επίδοση ήταν πολύ μικρή (Εικόνα 13).

Εικόνα 13: Αναζήτηση πλέγματος για το μέγεθος του κρυφού επιπέδου με το UNSW-NB15

Για την περίπτωση του GAN, εξαιτίας του διαφορετικού αριθμού εισόδων, η αρχιτεκτονική των δικτύων του Διευκρινιστή και Γεννήτορα προσαρμόζεται ανάλογα και παρατίθεται στο παράρτημα. Η μόνη διαφορετική παράμετρος μεταξύ των δύο εκδοχών με και

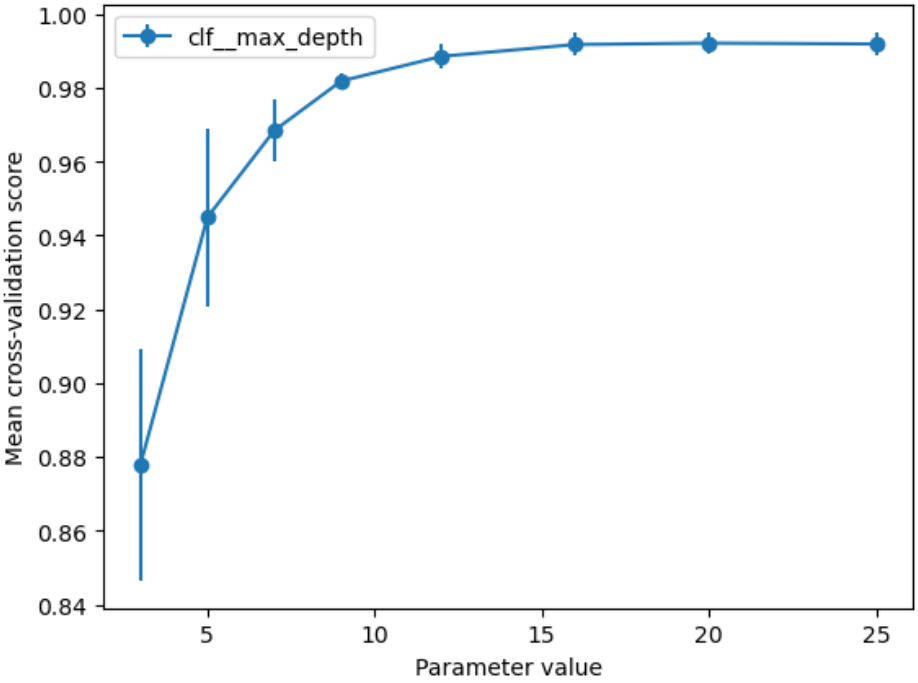
χωρίς επιλογή χαρακτηριστικών είναι το μέγεθος της μεταβλητής data_dim που στην περίπτωση χωρίς επιλογή χαρακτηριστικών είναι 202 (ίση με τον αριθμό των στηλών μετά την επεξεργασία των κατηγορικών χαρακτηριστικών) ενώ στην περίπτωση της επιλογής των 12 χαρακτηριστικών είναι προφανώς 12.

Πίνακας 8: Υπερπαράμετροι GAN με το UNSW-NB15

Υπερπαράμετρος	Τιμή
Αριθμός κρυφών επιπέδων Διευκρινιστή	2
Αριθμός νευρώνων Διευκρινιστή	data_dim->8->4->1
data_dim	202 ή 12
latent_dim	4
Learning Rate	0.0002
betas	(0.5, 0.999)
batch_size	64
Συνάρτηση Κόστους (Loss Function)	Binary Cross-Entropy (BCE)
Βελτιστοποιητής	Adam

4.2.4.3 Παραμετροποίηση μοντέλων με το Ενιαίο dataset

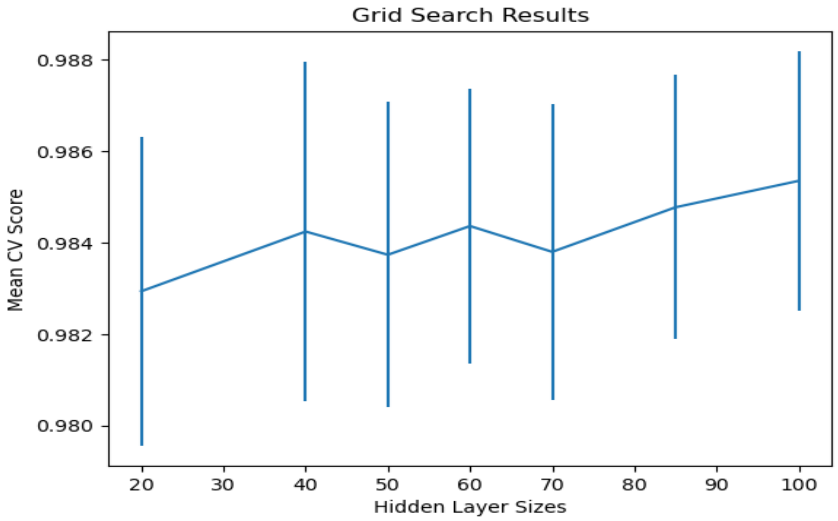
Για το Δένδρο Απόφασης επιλέχθηκε το βάθος 15 (Εικόνα 14).



Εικόνα 14: Αναζήτηση πλέγματος για το βάθους του δέντρου με το ενιαίο σύνολο

Για το SVM πολυωνυμικού πυρήνα με χρήση της RandomizedSearchCV() έγινε η ακόλουθη επιλογή υπερπαραμέτρων: C=10, gamma=1, kernel='poly'.

Για το MLP αφέθηκε στην προεπιλεγμένη τιμή νευρώνων (100) επειδή εκεί παρουσίασε την καλύτερη επίδοση.



Εικόνα 15: Αναζήτηση πλέγματος για το μέγεθος του κρυφού επιπέδου με το ενιαίο σύνολο

Για το GAN χρησιμοποιήθηκαν οι ίδιες παράμετροι με του GAN που εκπαιδεύτηκε με το NB15.

4.2.5 On-line Σύστημα

Για την ανάπτυξη του On-line συστήματος χρησιμοποιήθηκε η βιβλιοθήκη NFStream. Με το NFStream η ανάπτυξη και ενσωμάτωση μοντέλων μηχανικής μάθησης που εφαρμόζονται αυτοματοποιημένα πάνω σε ροές δικτυακής κίνησης είναι μια απλή υπόθεση, και επαφίεται απλώς στην μεταβίβαση μιας κλάσης που κληρονομεί απ' την κλάση NFPlugin και δέχεται ως παράμετρο το μοντέλο της επιλογή μας. Συγκεκριμένα δημιουργήθηκε αντικείμενο της βασικής κλάσης του NFStream, το NFStreamer, ως εξής:

Πίνακας 9: Αντικείμενο NFStreamer

```
ml_streamer = NFStreamer(source="Realtek 8822CE Wireless LAN 802.11ac
PCI-E NIC", udps=ModelPrediction(my_model=model, my_scaler=sscaler),
active_timeout=5, idle_timeout=5, statistical_analysis=True)
```

Όπου οι παράμετροι model και sscaler της κλάσης ModelPrediction, που κληρονομεί απ' την κλάση NFPlugin, είναι το αποθηκευμένο μοντέλο και scaler αντίστοιχα, και source η κάρτα δικτύου του υπολογιστή. Ας σημειωθεί ότι για λόγους που έχουν να κάνουν με το JupyterLab, η κλάση ModelPrediction πρέπει να αποθηκευτεί ως ξεχωριστό .py αρχείο και να γίνει import, αντί για την απευθείας υλοποίηση μέσα στο ίδιο notebook.

Η κλάση ModelPrediction έχει ως πρωταρχικό στόχο αφενός το διάβασμα της κίνησης, αφετέρου την προεπεξεργασία και την αντιστοίχιση των χαρακτηριστικών της με τα αντίστοιχα χαρακτηριστικά των δεδομένων εκπαίδευσης. Η προεπεξεργασία περιλαμβάνει αλλαγές χρονικής μονάδας μέτρησης, δημιουργία σύνθετων χαρακτηριστικών από απλούστερα χαρακτηριστικά του NFStream, κανονικοποίηση και άλλα. Η διαδικασία είναι εντελώς αντίστοιχη με αυτήν κατά την εκπαίδευση του μοντέλου.

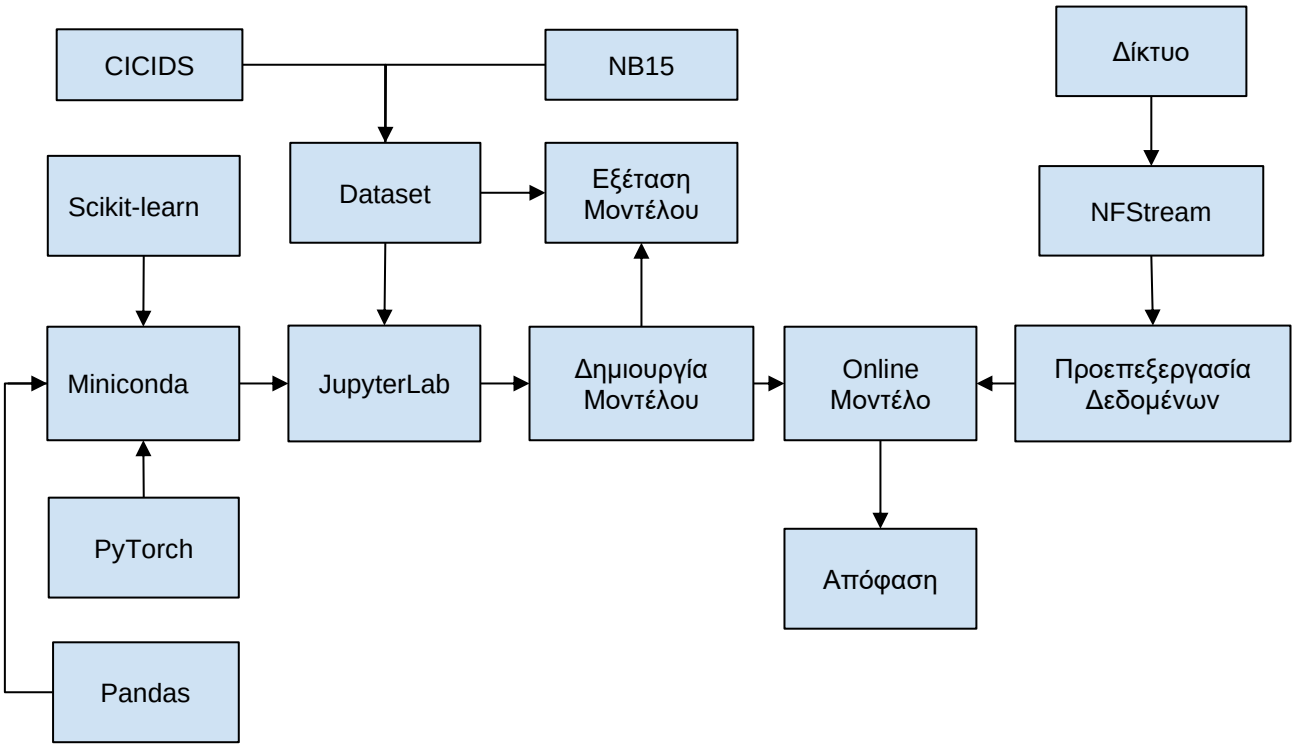
Στην κλάση ModelPrediction υλοποιούμε ορισμένες μεθόδους που κληρονομεί η κλάση απ' την κλάση NFPlugin όπως on_init και on_expire και ορίζουμε μια δική μας μέθοδο preprocess όπου θα πραγματοποιηθεί η απαραίτητη προεπεξεργασία. Περισσότερες λεπτομέρειες περιγράφονται στο Παράρτημα.

Για την καλύτερη σύγκριση μεταξύ των μοντέλων υλοποιήθηκε και ένα μικρό τμήμα κώδικα που στέλνει http requests σε μια λίστα από urls γνωστών ιστοσελίδων αποθηκευμένων σε ένα αρχείο, που λειτουργούν ως benchmark για την αναγνώριση κανονικής συμπεριφοράς, τουλάχιστον σε επίπεδο πρωτοκόλλου http. Η κλάση ModelPrediction εξετάζει αποκλειστικά ροές κίνησης που προέρχονται μόνο από αυτά τα domain names.

4.3 Εργαλεία/Τεχνολογίες

4.3.1 Επισκόπηση

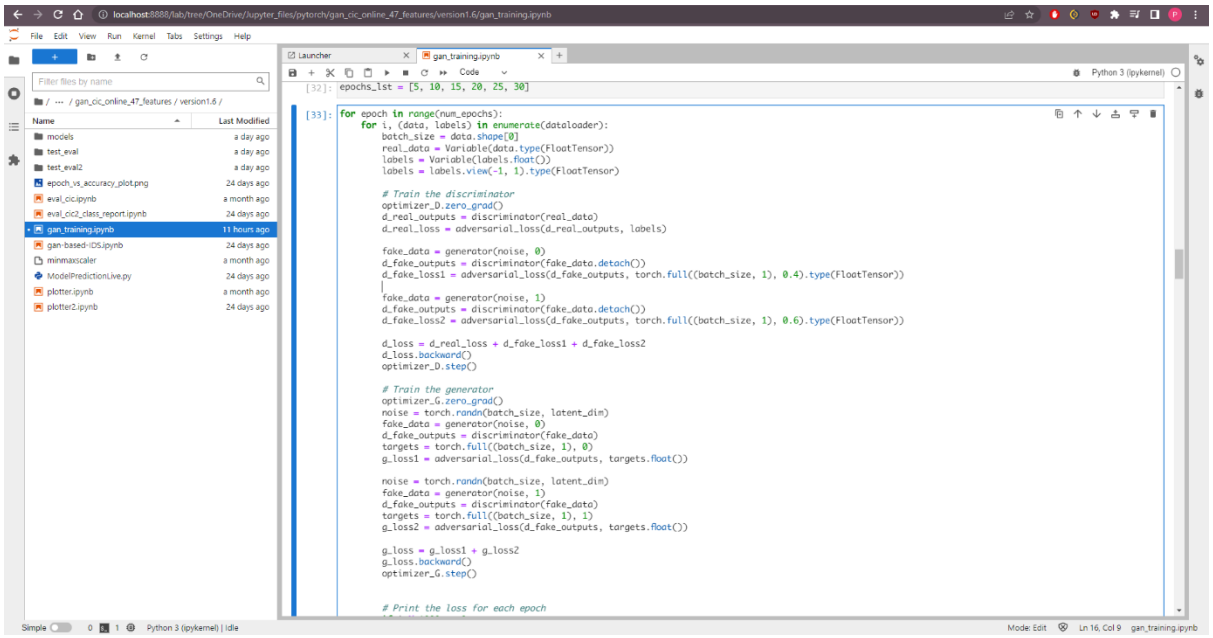
Ο τρόπος που ενσωματώνονται όλα τα εργαλεία ανάπτυξης για την παραγωγή του τελικού online μοντέλου περιγράφεται στην Εικόνα 16 . Στην συνέχεια περιγράφονται τα βασικότερα πακέτα και βιβλιοθήκες λογισμικού.



Εικόνα 16: Εργαλειοθήκη

4.3.2 JupyterLab

Το JupyterLab[30] είναι ανοικτού κώδικα λογισμικό που παρέχει ένα φιλικό προς τον χρήστη γραφικό περιβάλλον (Εικόνα 17) για το γράψιμο και την εκτέλεση κώδικα σε διάφορες γλώσσες προγραμματισμού όπως Python, R και Julia. Αποτελεί μετεξέλιξη του κλασσικού Jupyter Notebook και περιλαμβάνει διευκολύνσεις όπως άνοιγμα csv files σε κελιά όπως στο excel μέσα στους browsers. Αμφότερα αποτελούν λογισμικό διαδραστικού υπολογισμού (interactive computation) που χρησιμοποιείται ευρέως στην κοινότητα του Machine Learning και της Python. Μέσω αυτού, ο χρήστης μπορεί να δημιουργήσει και να επεξεργαστεί notebooks, τα οποία είναι αρχεία που περιέχουν κώδικα, κείμενο και πολυμεσικά αρχεία, και τα οποία μπορούν να διαμοιραστούν σε άλλους χρήστες με στόχο την πιο εύκολη συνεργασία μεταξύ τους. Τα κελιά που περιέχουν κώδικα δίνουν την δυνατότητα σε κάποιον να τρέξει τον κώδικα με το πάτημα ενός κουμπιού και το αποτέλεσμα να εμφανιστεί αμέσως από κάτω. Προαιρετικά ο χρήστης μπορεί να δημιουργήσει και κελιά με Markdown ώστε το συνολικό έγγραφο να είναι πιο ευανάγνωστο.

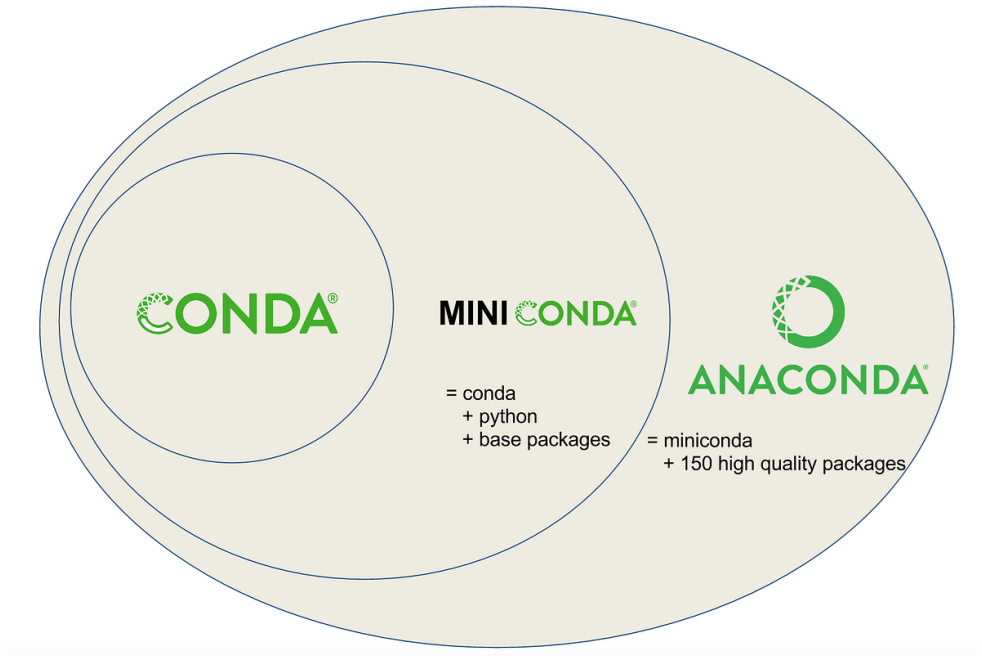


Εικόνα 17: Γραφικό περιβάλλον JupyterLab

[30] https://jupyter.org/

4.3.3 Miniconda

Το Miniconda[31] είναι μια ελαφριά εκδοχή του διαχειριστή πακέτων Conda για την Python μαζί με μερικά βασικά πακέτα. Το Conda επιτρέπει την εύκολη και γρήγορη εγκατάσταση, ενημέρωση και εν γένει διαχείριση πακέτων λογισμικού και εξαρτήσεων (dependencies). Η διαφορά του Miniconda απ' την ολοκληρωμένη σουίτα Anaconda είναι ότι εναπόκειται στον χρήστη να κατεβάσει και εγκαταστήσει τα πακέτα που χρειάζεται, απελευθερώνοντας έτσι χώρο στον σκληρό δίσκο από περιττά πακέτα (Εικόνα 18). Το Conda επίσης επιτρέπει την δημιουργία απομονωμένων περιβαλλόντων εργασίας με συγκεκριμένες εκδόσεις πακέτων και εξαρτήσεων, καθιστώντας έτσι πιο εύκολη την αναπαραγωγησιμότητα (reproducibility) κώδικα και προβλημάτων που μπορεί να παρουσιαστούν από ενδεχόμενη σύγκρουση λογισμικού (dependency conflict). Στο πλαίσιο της εργασίας δημιουργήθηκε ένα ξεχωριστό περιβάλλον εργασίας με όνομα «thesis» για την αποθήκευση των απαραίτητων πακέτων.



Εικόνα 18: Διάγραμμα Venn της διανομής Anaconda[32]

[31] https://www.anaconda.com/

[32] https://towardsdatascience.com/managing-project-specific-environments-with-conda-b8b50aa8be0e

4.3.4 NFStream

Για την ανάπτυξη του online συστήματος είναι αρχικά απαραίτητος ένας τρόπος καταγραφής και επεξεργασίας των δικτυακών δεδομένων. Για τον σκοπό αυτό επιλέχθηκε το λογισμικό NFStream [62].

Το NFStream είναι ένα εργαλείο ανάλυσης δικτυακών δεδομένων. Είναι σχεδιασμένο απ' τον δημιουργό του να είναι ευέλικτο, να προσφέρει στατιστική ανάλυση σε πραγματικό χρόνο, και να παρέχει αξιόπιστα δεδομένα για μια σύγχρονη χρήση δικτύου. Ταυτόχρονα, φιλοδοξεί να καλύψει το κενό ανάμεσα στην ανάλυση δικτύου και τις μεθόδους μηχανικής μάθησης.

4.3.4.1 Λειτουργία

Το NFStream είναι σχεδιασμένο να πραγματοποιεί ανάλυση δικτυακής κίνησης υπό μορφή ροών με υψηλή παραγωγικότητα και να τρέχει σε υπολογιστικούς πόρους ευρείας κατανάλωσης. Οι ροές (flows) κατασκευάζονται απ' την συνένωση πακέτων που μοιράζονται ένα κοινό κλειδί. Το κλειδί είναι ένα 7-tuple που αποτελείται απ' τα ακόλουθα:

- IP Πηγής
- Πύλη (Port) Πηγής
- IP Προορισμού
- Πύλη Προορισμού
- Πρωτόκολλο
- VLAN
- Αναγνωριστικά tunnel

Τα χαρακτηριστικά ροής (flow features) και το κλειδί ροής είναι καταχωρημένα στις εισαγωγές ροών (flow entries). Τα χαρακτηριστικά ροής πηγάζουν απ' τις επικεφαλίδες των IP, TCP, και UDP πακέτων και υπολογίζονται με στατιστικές μεθόδους. Εκτός αυτών, το NFStream έχει την ικανότητα να αναγνωρίζει τους τύπους των εφαρμογών που παράγουν τις ροές. Το NFStream έχει σχεδιαστεί για να λειτουργεί και live και offline και απαιτεί την χρήση της βιβλιοθήκης libpcap για την καταγραφή των πακέτων. Το NFStream παρέχει την δυνατότητα tunnel decoding και την αναγνώριση πακέτων από διαφορετικές ροές. Το NFStream είναι σχεδιασμένο να μπορεί να χρησιμοποιηθεί ως βιβλιοθήκη από άλλες εφαρμογές και να μπορεί να επεκταθεί. Είναι υλοποιημένο έτσι ώστε να μπορεί να συνεργαστεί εύκολα με τις βασικότερες βιβλιοθήκες Machine Learning που έχουν αναπτυχθεί την τελευταία δεκαετία σε γλώσσα Python (π.χ. TensorFlow, PyTorch, scikit-learn κτλ.). Τέλος, το NFStream μπορεί να

αξιοποιηθεί για την μεταφορά και χρησιμοποίηση μοντέλων μηχανικής μάθησης πάνω σε μια κοινή υποδομή όπου καθίσταται δυνατή η αναπαραγωγή αποτελεσμάτων (π.χ. εργασία πάνω σε συλλογή ίδιων τύπου χαρακτηριστικών), χωρίς να απαιτείται η μεταφορά συνόλων δεδομένων που πολλές φορές προσκρούει πάνω στην προστασία της ιδιωτικότητας.

4.3.4.2 Περιγραφή Δομής

Το NFStream αποτελείται απ' το αντικείμενο NFStreamer που επεξεργάζεται τις NFlows-αντικείμενα Python που αντιστοιχίζονται σε ροές, και μια σειρά από παράλληλα «Meters» που δημιουργούν τις NFlows και τις εισάγουν στο NFStreamer, έπειτα από επεξεργασία NFPackets. Κάθε Meter αποτελείται από ένα επίπεδο Packet Observation που δέχεται ως είσοδο ανεπεξέργαστα πακέτα και δίνει ως έξοδο ένα NFPacket αντικείμενο Python, και ένα επίπεδο Flow Metering που αναλαμβάνει την αντιστοίχιση των NFPackets σε NFlows.
Εντός του επιπέδου Packet Observation περιλαμβάνονται οι εξής λειτουργίες:

1. Καταγραφή πακέτου (Packet Capture)
2. Αποκοπή πακέτου (Packet truncation)
3. Χρονοσφράγιση πακέτου (Packet timestamping)
4. Φιλτράρισμα πακέτου (Packet filtering)
5. Επεξεργασία πακέτου (Packet processing)
6. Αποστολή πακέτου (Packet dispatching)

Το επίπεδο Flow Metering αποτελείται απ' τα εξής στοιχεία:

1. NFCache
2. Διαχείριση τερματισμού
3. NFPlugins

Η κλάση NFStreamer είναι η βασική κλάση του NFStream. Αποτελεί το επίπεδο εξαγωγής και είναι υπεύθυνη για την ενορχήστρωση των παράλληλων Meter διαδικασιών και τον ορισμό της δομής που θα έχουν οι εξαγώγιμες ροές. Δέχεται πολλούς παραμέτρους για τον έλεγχο των δύο διαδικασιών/επιπέδων. Η κλάση παρέχει πολλούς τρόπους εξαγωγής των ροών, με τους συνηθέστερους να είναι σε μορφή αρχείων CSV και Pandas Dataframes. Τέλος, δίνεται η δυνατότητα ανωνυμοποίησης των χαρακτηριστικών μέσω του αλγόριθμου Blake2[33].

[33] https://www.blake2.net/

4.3.4.3 Σύγκριση με άλλες τεχνολογίες

Το NFStream ξεχωρίζει απ' τις υπόλοιπες σημερινές τεχνολογίες για την ταυτόχρονη χρήση παράλληλης δυνατότητας tunnel decoding, αναγνώρισης εφαρμογής (application awareness) και επεξεργασίας ιχνών σε online και offline περιβάλλοντα.

Οι δημιουργοί έκαναν μια σύγκριση του NFStream με διάφορες παραμέτρους όπως αριθμός χαρακτηριστικών, χρήση διαφόρων λειτουργιών, χρήση διαφορετικού αριθμού πυρήνων κτλ. για την αξιολόγηση του λογισμικού σε πραγματικές συνθήκες μέσα από ένα pcap αρχείο που συλλέχθηκε σε ένα πανεπιστημιακό δίκτυο. Τα αποτελέσματα έδειξαν ότι υπάρχει σαφής βελτίωση ως προς τον χρόνο με χρήση PyPy αντί CPython για τον ρόλο του ερμηνευτή/compiler. Έγινε επίσης μια απόπειρα σύγκρισης με άλλα δημοφιλή λογισμικά όπως CICFlowMeter και ndpiReader που έδειξαν το NFStream να υπερτερεί σε χρόνο αλλά οι ερευνητές τονίζουν ότι αυτά τα προγράμματα δεν έχουν αναπτυχθεί για να τρέχουν σε πολλαπλούς πυρήνες επεξεργασίας.

Το NFStream απολαμβάνει συνεχώς αυξανόμενης απήχησης και έχει χρησιμοποιηθεί μεταξύ άλλων σε εφαρμογές video streaming και μηχανικής μάθησης σε 5G δίκτυα, IDS (Intrusion Detection Systems) για την συλλογή συνόλων δεδομένων (datasets), ετικετοποίηση ροών και εξαγωγή χαρακτηριστικών πακέτων από αρχεία .pcap, ανάλυση κίνησης δικτύου, και ανάλυση του δικτύου του TOR.

Συμπερασματικά είναι ένα χρήσιμο εργαλείο για την ανάλυση δικτύων, την χρήση του για την σύγκριση διαφορετικών προσεγγίσεων μηχανικής μάθησης και την δημιουργία συνόλων δεδομένων.

4.3.5 Πληροφορίες για πακέτα Μηχανικής Μάθησης και επεξεργασίας δεδομένων

Στον κλάδο της μηχανικής μάθησης, ένας απ' τους κύριους λόγους που κυριαρχεί τα τελευταία χρόνια η χρήση της Python είναι οι πολύ καλές βιβλιοθήκες της που την καθιστούν ιδιαίτερα εύχρηστη και αποτελεσματική. Εδώ θα δοθούν μερικές εισαγωγικές πληροφορίες για τις δυνατότητες χρήσης των πακέτων που χρησιμοποιήθηκαν στην παρούσα εργασία.

4.3.5.1 Scikit-Learn

Το scikit-learn[34] (πρώην scikits.learn και επίσης γνωστό ως sklearn) είναι μια δωρεάν βιβλιοθήκη μηχανικής εκμάθησης λογισμικού για τη γλώσσα προγραμματισμού Python. Διαθέτει διάφορους αλγόριθμους ταξινόμησης, παλινδρόμησης και συσταδοποίησης, συμπεριλαμβανομένων μηχανών υποστήριξης διανυσμάτων, τυχαίων δασών, ενίσχυσης κλίσης (gradient boosting), k-means και DBSCAN, και έχει σχεδιαστεί για να λειτουργεί με τις αριθμητικές και επιστημονικές βιβλιοθήκες NumPy και SciPy.

4.3.5.2 PyTorch

Το PyTorch[35] είναι ένα framework μηχανικής εκμάθησης που βασίζεται στη βιβλιοθήκη Torch, και χρησιμοποιείται για εφαρμογές όπως η όραση υπολογιστή και η επεξεργασία φυσικής γλώσσας. Αναπτύχθηκε αρχικά από τη Meta AI (πρώην Facebook) και τώρα αποτελεί μέρος του Ιδρύματος Linux. Είναι δωρεάν λογισμικό ανοιχτού κώδικα που κυκλοφορεί με την τροποποιημένη άδεια BSD. Παρόλο που η χρήση της κατά κύριο λόγο περιορίζεται στην Python, όπου και αποτελεί το κύριο επίκεντρο της ανάπτυξης, η PyTorch διαθέτει επίσης μια διεπαφή (interface) C++.

Ορισμένα κομμάτια λογισμικού βαθιάς εκμάθησης είναι χτισμένα πάνω από το PyTorch, συμπεριλαμβανομένων των Tesla Autopilot, Uber's Pyro, Hugging Face's Transformers, PyTorch Lightning, και Catalyst.

Το PyTorch παρέχει δύο δυνατότητες υψηλού επιπέδου:

- Υπολογισμός τανυστή (όπως το NumPy) με ισχυρή επιτάχυνση μέσω μονάδων επεξεργασίας γραφικών (GPU)
- Βαθιά νευρωνικά δίκτυα χτισμένα σε ένα σύστημα αυτόματης διαφοροποίησης που βασίζεται σε ταινία. (Deep neural networks built on a tape-based automatic differentiation system)

4.3.5.3 Pandas

Η βιβλιοθήκη Pandas[36] είναι μια από τις πιο δημοφιλείς βιβλιοθήκες για επεξεργασία και ανάλυση δεδομένων στην Python. Το όνομα Pandas προέρχεται από τον οικονομετρικό όρο "panel data" που αναφέρεται σε σύνολα δεδομένων που περιέχουν τιμές παρατηρήσεων

[34] https://scikit-learn.org
[35] https://pytorch.org/
[36] https://pandas.pydata.org/

χαρακτηριστικών σε πολλαπλά χρονικά διαστήματα. Η βιβλιοθήκη περιλαμβάνει αντικειμενοστραφείς δομές δεδομένων, όπως DataFrame, Series και Panel, που επιτρέπουν την εύκολη εισαγωγή, επεξεργασία, καθαρισμό και ανάλυση δεδομένων. Επιπλέον, η βιβλιοθήκη επιτρέπει την ανάγνωση δεδομένων από διάφορες πηγές, όπως αρχεία CSV, Excel, SQL, JSON κ.λπ. Στο πλαίσιο της μηχανικής μάθησης και εξόρυξης δεδομένων, η βιβλιοθήκη χρησιμοποιείται στο αρχικό στάδιο της αποδοτικής προεπεξεργασίας και καθαρισμού μεγάλων συνόλων δεδομένων. Τέλος, η βιβλιοθήκη περιλαμβάνει πολλά εργαλεία οπτικοποίησης δεδομένων για την καλύτερη κατανόηση των δεδομένων.

4.3.5.4 Matplotlib

Για την δημιουργία κάποιων γραφικών που περιλαμβάνει η εργασία χρησιμοποιήθηκε η βιβλιοθήκη Matplotlib.

4.4 Προαπαιτούμενα

4.4.1 Εγκατάσταση βιβλιοθηκών

Αρχικά πρέπει να κατέβει ο installer της Miniconda και να γίνει εγκατάσταση του package manager Conda καθώς και της python. Ο installer βρίσκεται στην διεύθυνση https://docs.conda.io/en/latest/miniconda.html. Ύστερα αφού γίνει η εγκατάσταση μπορούμε να χρησιμοποιήσουμε την εντολή conda στο τερματικό για την εγκατάσταση οποιουδήποτε άλλου πακέτου χρειαζόμαστε. Προαιρετικά μπορούμε να δημιουργήσουμε νέο environment για την απομόνωση των πακέτων καθώς και της έκδοσης python που έρχεται μαζί με το Miniconda.

Με την εντολή

```
conda install jupyter
```

γίνεται η εγκατάσταση της σουίτας Jupyter που περιλαμβάνει τα Jupyter Notebook, Lab.

Με τις εντολές

```
conda install -c conda-forge scikit-learn
conda install pytorch torchvision torchaudio pytorch-cuda=11.7 -c
pytorch -c nvidia
```

γίνεται η εγκατάσταση των πακέτων μηχανικής μάθησης sklearn και PyTorch αντίστοιχα. Εκτός αυτών χρειαζόμαστε το πακέτο pandas για την διαχείριση των δεδομένων μέσω

dataframes. Αυτό γίνεται (αν δεν έχει ήδη γίνει η εγκατάσταση λόγω dependencies προηγούμενων πακέτων) μέσω της εντολής

```
conda install pandas
```

Τυπικά ο package manager θα κατεβάσει αυτόματα όλα τα πακέτα που προαπαιτούνται για την χρήση του pandas όπως NumPy κτλ. Τέλος, χρειαζόμαστε τα πακέτα imbalanced-learn και NFStream. Το πρώτο χρησιμοποιείται για over και undersampling τεχνικές και μπορεί να «κατεβαστεί» μέσω Conda, ενώ το δεύτερο αφορά το διάβασμα των πακέτων είτε live είτε με χρήση pcap αρχείων και η εγκατάσταση του μπορεί να γίνει μέσω pip.

```
conda install -c conda-forge imbalanced-learn
pip install nfstream
```

4.4.2 Λήψη datasets

Το CICIDS17 μπορεί να γίνει download απ’ την διεύθυνση

https://www.unb.ca/cic/datasets/ids-2017.html

Ενώ το UNSW-NB15 απ’ την διεύθυνση

https://research.unsw.edu.au/projects/unsw-nb15-dataset

5 Αποτελέσματα και Συζήτηση

5.1 Γενικά

Η αξιολόγηση των μοντέλων δεν είναι αμφιμονοσήμαντη και εξαρτάται καίρια απ' τα δεδομένα επί των οποίων θα γίνει ο έλεγχος. Είναι προφανές ότι σε δεδομένα που ακολουθούν την πιθανοτική κατανομή των δεδομένων εκπαίδευσης, παραδείγματος χάριν δεδομένα δοκιμής προερχόμενα απ' τα ίδια dataset επί των οποίων έγινε η εκπαίδευση, ακόμη και αν το μοντέλο δεν τα έχει δει προηγουμένως, μπορούμε να αναμένουμε υψηλά ποσοστά ευστοχίας ακόμα και στα απλούστερα μοντέλα. Αντίθετα σε πραγματικά δίκτυα όπου είναι πολύ πιο δύσκολο να υπάρξει καλή ταύτιση των κατανομών των δεδομένων εκπαίδευσης και δοκιμής, τα αποτελέσματα είναι κατά βάση χειρότερα. Έχοντας πει αυτά, έγινε καταρχάς ένας πρώτος έλεγχος επί των συνόλων δεδομένων δοκιμής που δημιουργήσαμε απ' το CIC-IDS2017, και UNSW-NB15 καθώς και το ενιαίο σύνολο δεδομένων των CIC-IDS2017 και UNSW-NB15, για να βεβαιωθούμε ότι τα μοντέλα έχουν οριστεί καλά. Στην συνέχεια γίνεται και ένας έλεγχος των μοντέλων πάνω σε πραγματικές ροές δεδομένων μετά από http requests σε διεθνείς και ελληνικές ιστοσελίδες για την αξιολόγηση επί του πεδίου. Εκεί μας ενδιαφέρει αν τα μοντέλα αναγνωρίζουν επιτυχώς τις συνδέσεις ως BENIGN, δηλαδή κανονικής λειτουργίας.
Τα αποτελέσματα των ρηχών μοντέλων που εξετάστηκαν προέκυψαν απ' τις συναρτήσεις metrics.classification_report και metrics.accuracy_score του scikit-learn, που εμφανίζουν τις μετρικές accuracy, precision, recall και f1-score.

Επαναλαμβάνουμε εδώ τι σημαίνουν οι μετρικές. Η μετρική accuracy δείχνει πόσες σωστές ταξινομήσεις και των δύο κατηγοριών πέτυχε το μοντέλο. Επειδή τα σύνολα δεδομένων παρουσιάζουν μεγάλη ανισορροπία κλάσεων, δηλαδή ασχολούμαστε με αναγνώριση ανωμαλίας παρά με ένα απλό πρόβλημα ταξινόμησης που συνήθως οι κλάσεις έχουν περίπου τον ίδιο αριθμό παραδειγμάτων, η μετρική accuracy από μόνη της δεν είναι αρκετή για την αξιολόγηση του μοντέλου· ένα μοντέλο θα μπορούσε κάλλιστα να μάθει να δίνει ως έξοδο την κλάση με τα περισσότερα παραδείγματα και να έχει υψηλό accuracy, χωρίς πραγματικά να έχει μάθει να διαχωρίζει τις κλάσεις. Ως εκ τούτου, είναι απαραίτητες και άλλες μετρικές. Η μετρική Precision δείχνει πόσο σωστό είναι το μοντέλο σε μια πρόβλεψη του σε κάποια κλάση, δηλαδή αν για παράδειγμα κάνει 100 προβλέψεις για εγγραφές ότι ανήκουν στην κλάση Α, η μετρική Precision δείχνει πόσο σωστό ήταν στις προβλέψεις τους. Για να είναι εντελώς σωστό θα πρέπει και οι 100 εγγραφές να ανήκουν πράγματι στην κλάση Α. Η

μετρική Recall συμπληρώνει την μετρική Precision και δείχνει πόσες εγγραφές που ανήκουν στην κλάση Α το μοντέλο αναγνώρισε επιτυχώς ότι ανήκουν στην κλάση Α. Αν για παράδειγμα σε ένα σύνολο δεδομένων έχουμε 100 εγγραφές που ανήκουν στην κλάση Α και το μοντέλο αναγνώρισε από αυτές τις 100 εγγραφές ότι μόνο οι 90 ανήκουν στην κλάση Α, τότε η μετρική Recall θα είναι 90% ή 0.9 σε δεκαδικές μονάδες. Τέλος, η μετρική F1-Score ορίζεται ως ο αρμονικός μέσος όρος των Precision και Recall, και με αυτόν τον τρόπο συνδυάζει τις πληροφορίες που παρέχουν αυτές οι δύο μετρικές και δίνει μια πιο ολοκληρωμένη εικόνα. Στους πίνακες που ακολουθούν η στήλη «support» δείχνει τον αριθμό των εγγραφών που υπάρχουν σε κάθε κατηγορία.

5.2 Αποτελέσματα για το σύνολο δεδομένων CIC-IDS2017

Η προεπεξεργασία των συνόλων δεδομένων, ο διαχωρισμός τους σε σύνολα εκπαίδευσης και ελέγχου και η επιλογή των χαρακτηριστικών περιγράφεται στην υποενότητα 4.2.3.1 της προεπεξεργασίας στην ενότητα 4 με τίτλο «Δημιουργία μοντέλων με το CICIDS 2017», ενώ οι υπερπαράμετροι των μοντέλων που εξετάστηκαν παρουσιάζονται στην υποενότητα 4.2.4.1 της παραμετροποίησης, με τίτλο «Παραμετροποίηση μοντέλων με το CICIDS-2017». Το σύνολο των χαρακτηριστικών που περιλαμβάνονται στο CIC-IDS2017 παρέχεται σε σχετικό υπερσύνδεσμο στην υποενότητα 3.1.6 όπου περιγράφεται το σύνολο δεδομένων αναλυτικά.

5.2.1 Αποτελέσματα για σύνολο δεδομένων επαλήθευσης

5.2.1.1 Δένδρο Απόφασης

Το μοντέλο που βασίζεται σε Δένδρο Απόφασης εκπαιδεύτηκε με μέγιστο βάθος 8 και παρουσίασε πρακτικά την ίδια συμπεριφορά με την εκπαίδευση πάνω σε όλα τα χαρακτηριστικά και στο διάνυσμα 47 χαρακτηριστικών που επιλέχθηκε. Η απόδοση ήταν πολύ υψηλή, με τιμή ακριβείας άνω του 99%, η υψηλότερη όλων των μοντέλων που εξετάστηκαν.

5.2.1.1.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9950648946918473

Πίνακας 10: Αποτελέσματα Δένδρου Απόφασης χωρίς επιλογή χαρακτηριστικών για το CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.99	0.98	0.99	55763

BENIGN	1.00	1.00	1.00	227311
accuracy			1.00	283074
macro avg	0.99	0.99	0.99	283074
weighted avg	1.00	1.00	1.00	283074

5.2.1.1.2 Επιλογή χαρακτηριστικών

Accuracy: 0.9934469432021309

Πίνακας 11: Αποτελέσματα Δένδρου Απόφασης με επιλογή χαρακτηριστικών για το CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.99	0.98	0.98	55763
BENIGN	0.99	1.00	1.00	227311
accuracy			0.99	283074
macro avg	0.99	0.99	0.99	283074
weighted avg	0.99	0.99	0.99	283074

5.2.1.2 SVM-Linear

Το μοντέλο που βασίστηκε στο γραμμικό SVM έδειξε ότι δεν μπορούσε να διαχωρίσει τα δεδομένα ικανοποιητικά και παρουσίασε χαμηλή αποτελεσματικότητα στην αναγνώριση των επιθέσεων με ή χωρίς επιλογή χαρακτηριστικών.

5.2.1.2.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9319047316249461

Πίνακας 12: Αποτελέσματα γραμμικού SVM χωρίς επιλογή χαρακτηριστικά στο CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.85	0.79	0.82	55763
BENIGN	0.95	0.97	0.96	227311
accuracy			0.93	283074

macro avg	0.90	0.88	0.89	283074
weighted avg	0.93	0.93	0.93	283074

5.2.1.2.2 Επιλογή χαρακτηριστικών

Accuracy: 0.9079463320545158

Πίνακας 13: Αποτελέσματα γραμμικού SVM με επιλογή χαρακτηριστικών για το CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.79	0.73	0.76	55763
BENIGN	0.94	0.95	0.94	227311
accuracy			0.91	283074
macro avg	0.86	0.84	0.85	283074
weighted avg	0.91	0.91	0.91	283074

5.2.1.3 SVM-Kernel

Το μοντέλο SVM που έκανε χρήση πολυωνυμικού πυρήνα παρουσίασε μία ικανοποιητική απόδοση χωρίς σημαντική διαφοροποίηση μεταξύ των εκπαιδεύσεων με ή χωρίς επιλογή χαρακτηριστικών. Παρουσίασε μια σαφής βελτίωση με το SVM γραμμικού διαχωρισμού. Υπολείπεται ελάχιστα σε σχέση με το Δένδρο Απόφασης στην αναγνώριση των επιθέσεων. Υπενθυμίζουμε ότι οι ακριβείς τιμές υπερπαραμέτρων που επιλέχθηκαν αναφέρονται στην υποενότητα της προεπεξεργασίας στην ενότητα 4.

5.2.1.3.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9890735284766528

Πίνακας 14: Αποτελέσματα SVM πολυωνυμικού πυρήνα χωρίς επιλογή χαρακτηριστικών για το CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.98	0.96	0.97	55763
BENIGN	0.99	1.00	0.99	227311

accuracy			0.99	283074
macro avg	0.98	0.98	0.98	283074
weighted avg	0.99	0.99	0.99	283074

5.2.1.3.2 Επιλογή Χαρακτηριστικών

Accuracy: 0.9863427937571094

Πίνακας 15: Αποτελέσματα SVM πολυωνυμικού πυρήνα με επιλογή χαρακτηριστικών για το CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.97	0.96	0.96	55763
BENIGN	0.99	0.99	0.99	227311
accuracy			0.99	283074
Macro avg	0.98	0.97	0.98	283074
Weighted avg	0.99	0.99	0.99	283074

5.2.1.4 MLP

Το MLP παρουσίασε και αυτό πολύ υψηλή απόδοση στην ταξινόμηση των παραδειγμάτων και μια ανεπαίσθητη μείωση της στην περίπτωση της επιλογής χαρακτηριστικών στην αναγνώριση των επιθέσεων. Στην περίπτωση της επιλογής χαρακτηριστικών η συνολική ακρίβεια (accuracy) είναι πρακτικά ίδια με του SVM πολυωνυμικού πυρήνα.

5.2.1.4.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9922847029398674

Πίνακας 16: Αποτελέσματα MLP χωρίς επιλογή χαρακτηριστικών για το CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.97	0.99	0.98	55763
BENIGN	1.00	0.99	1.00	227311

accuracy			0.99	283074
macro avg	0.98	0.99	0.99	283074
weighted avg	0.99	0.99	0.99	283074

5.2.1.4.2 Επιλογή χαρακτηριστικών

Accuracy: 0.986706656210037

Πίνακας 17: Αποτελέσματα MLP με επιλογή χαρακτηριστικών για το CIC-IDS2017

	precision	recall	f1-score	support
ATTACK	0.96	0.98	0.97	55763
BENIGN	0.99	0.99	0.99	227311
accuracy			0.99	283074
macro avg	0.97	0.98	0.98	283074
weighted avg	0.99	0.99	0.99	283074

5.2.1.5 GAN

Το GAN εκπαιδεύτηκε συνολικά για 100 epochs και αποθηκεύτηκε το κάθε μοντέλο που αντιστοιχεί σε κάθε epoch για την μεταξύ τους εξέταση, σύγκριση και επιλογή. Οι μετρικές που υπολογίστηκαν ήταν οι ίδιες με τις μετρικές των ρηχών μοντέλων, δηλαδή Accuracy, Precision, Recall και F1-Score. Δεδομένου ότι μοντέλο που να υπερτερεί σε όλες τις μετρικές δεν υπήρχε, αποφασίστηκε η τελική επιλογή να γίνει με βάση την μετρική Accuracy, που αφορά συνολικά την επίδοση του μοντέλου πάνω στο σύνολο δεδομένων ελέγχου. Η συνολική απόδοση ήταν υψηλή, της τάξεως του ~98%, συγκρίσιμη αλλά ελαφρώς μικρότερη των υπόλοιπων μοντέλων (πλην SVM γραμμικού διαχωρισμού). Η χαμηλότερη μετρική ήταν η Precision στο 92%. Η συμπεριφορά του μοντέλου δεν παρουσίασε διαφορά απόδοσης μεταξύ της εκπαίδευσης με όλα τα χαρακτηριστικά και της επιλογής χαρακτηριστικών. Ο ακριβής αριθμός εποχών που το μοντέλο παρουσίασε την καλύτερη απόδοση δεν ενδιαφέρει αφού η διαδικασία είναι στοχαστική και ποικίλει σε κάθε επαναληπτικό βρόχο εκπαίδευσης. Η τιμή κατωφλίου (threshold) που επιλέχθηκε για την επιλογή της κλάσης απ’ το μοντέλο ήταν το 0.5,

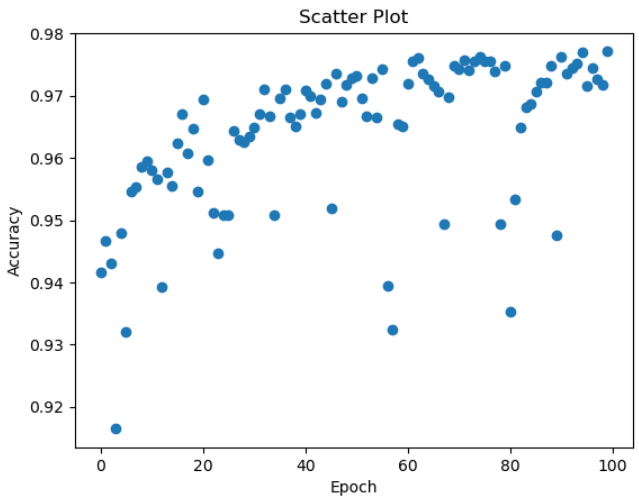
δηλαδή τιμές μεγαλύτερες του 0.5 θεωρούνται ως BENIGN ενώ έξοδοι μικρότεροι του 0.5 κατατάσσονται ως ATTACK. Ακολούθως παρατίθενται τα αποτελέσματα του καλύτερου μοντέλου στις δύο περιπτώσεις που εξετάζονται, δηλαδή με όλα τα χαρακτηριστικά και με την επιλογή χαρακτηριστικών που θα χρησιμοποιηθούν στο online μοντέλο, και αμέσως μετά ακολουθεί ένα scatterplot για τις τιμές accuracy των 100 μοντέλων που αποθηκεύτηκαν και εξετάστηκαν. Όπως φαίνεται στις Εικόνες 19 και 20, το GAN παρουσιάζει σχετική ομαλότητα στις τιμές accuracy χωρίς μεγάλες αποκλίσεις μεταξύ των εποχών εκπαίδευσης και παρουσιάζει σύγκλιση μετά από έναν ορισμένο αριθμό epochs εκπαίδευση. Ο κώδικας που εξετάζει το σύνολο δεδομένων ελέγχου και υπολογίζει τις μετρικές αξιολόγησης παρατίθεται στο παράρτημα.

5.2.1.5.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9770728088458686

Πίνακας 18: Αποτελέσματα GAN χωρίς επιλογή χαρακτηριστικών για το CIC-IDS2017

	Precision	Recall	F1-Score
BENIGN	0.993433	0.977922	0.985616
ATTACK	0.91526	0.973604	0.943531



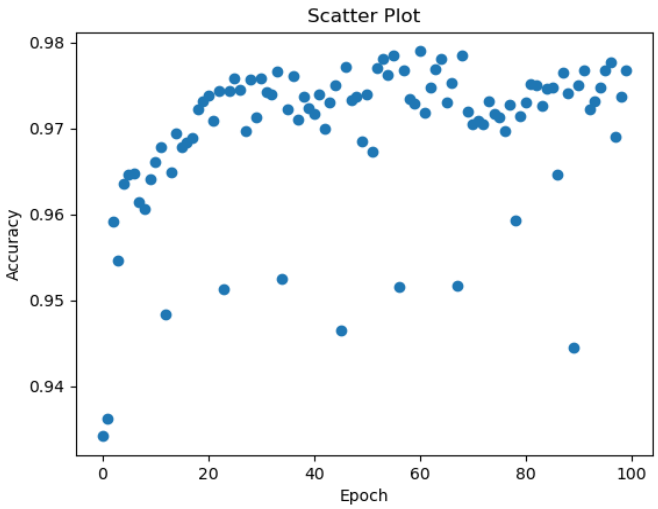
Εικόνα 19: Ακρίβεια ανά αριθμό εποχών χωρίς επιλογή χαρακτηριστικών με το CIC-IDS2017

5.2.1.5.2 Επιλογή Χαρακτηριστικών

Accuracy: 0.9789804641961353

Πίνακας 19: Αποτελέσματα GAN με επιλογή χαρακτηριστικών για το CIC-IDS2017

	Precision	Recall	F1-Score
BENIGN	0.993463	0.980341	0.986858
ATTACK	0.923024	0.973365	0.947526



Εικόνα 20: Ακρίβεια ανά αριθμό εποχών με επιλογή χαρακτηριστικών με το CIC-IDS2017

5.2.2 Αποτελέσματα για το online σύστημα

Όπως έχουμε αναφέρει και προηγουμένως, για την αξιολόγηση του online συστήματος εξετάστηκε η απόκριση των μοντέλων σε ένα αρχείο με urls που προγραμματιστικά μέσω python έγιναν http requests. Τα urls αποτελούν μερικές απ' τις πιο συχνά επισκεπτόμενες ελληνικές και ξένες ιστοσελίδες οι οποίες θεωρούνται BENIGN. Μετρήθηκε ο αριθμός των λάθος ταξινομήσεων, δηλαδή επιθέσεων, στις αποκρίσεις απ' τα συγκεκριμένα http requests. Όπως φαίνεται στον πίνακα που ακολουθεί με εξαίρεση το μοντέλο SVM γραμμικού διαχωρισμού που, όπως άλλωστε ήταν αναμενόμενο απ' την χαμηλή απόδοση του στο σύνολο δεδομένων ελέγχου, αδυνατεί να διαχωρίσει σωστά τα δεδομένα, και το GAN που παρουσιάζει κάποιες λάθος αποκρίσεις, εν γένει τα μοντέλα φαίνεται να δουλεύουν σωστά.

Πίνακας 20: Αποτελέσματα online συστήματος εκπαιδευμένο με το CIC-IDS2017

	BENIGN	ATTACK	Support
Δένδρο Απόφασης	93	0	93
SVM-Linear	51	50	101
SVM-Kernel	84	0	84
MLP	81	0	81
GAN	74	10	84

5.3 Αποτελέσματα για το UNSW-NB15

5.3.1 Αποτελέσματα για σύνολο δεδομένων επαλήθευσης

Υπενθυμίζεται όπως και προηγουμένως ότι η προεπεξεργασία των συνόλων δεδομένων, ο διαχωρισμός τους σε σύνολα εκπαίδευσης και ελέγχου και η επιλογή χαρακτηριστικών περιγράφεται στην υποενότητα 4.2.3.2 της προεπεξεργασίας στην ενότητα 4 με τίτλο «Δημιουργία μοντέλων με το UNSW-NB15», ενώ οι υπερπαράμετροι των μοντέλων που εξετάστηκαν παρουσιάζονται στην υποενότητα 4.2.4.2 της παραμετροποίησης, με τίτλο «Παραμετροποίηση μοντέλων με το UNSW-NB15». Το σύνολο των χαρακτηριστικών υπάρχει στην υποενότητα 3.1 όπου παρουσιάζεται το σχετικό dataset.

5.3.1.1 Δένδρο Απόφασης

Το Δένδρο Απόφασης παρουσίασε, όπως και στην εκπαίδευση με το σύνολο δεδομένων CICIDS-2017, πολύ υψηλή ευστοχία που υπερβαίνει το 99% και για τα δύο διανύσματα χαρακτηριστικών. Η απόδοση είναι γενικότερα συγκρίσιμη με το αντίστοιχο μοντέλο που εκπαιδεύτηκε με το CICIDS-2017, αν και ελαφρώς χειρότερη στην αναγνώριση των παραδειγμάτων επίθεσης.

5.3.1.1.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9931300564949509

Πίνακας 21: Αποτελέσματα Δένδρου Απόφασης χωρίς επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support

ATTACK	0.98	0.97	0.97	12859
BENIGN	1.00	1.00	1.00	88743
accuracy			0.99	101602
macro avg	0.99	0.98	0.98	101602
weighted avg	0.99	0.99	0.99	101602

5.3.1.1.2 Επιλογή χαρακτηριστικών

Accuracy: 0.9910926950256884

Πίνακας 22: Αποτελέσματα Δένδρου Απόφασης με επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.98	0.95	0.96	12859
BENIGN	0.99	1.00	0.99	88743
accuracy			0.99	101602
macro avg	0.99	0.97	0.98	101602
weighted avg	0.99	0.99	0.99	101602

5.3.1.2 SVM-Linear

Το SVM γραμμικού διαχωρισμού παρουσίασε αποδεκτή απόδοση στην περίπτωση όλων των χαρακτηριστικών αλλά πολύ κακή απόδοση στην ανίχνευση επιθέσεων στην περίπτωση της επιλογής χαρακτηριστικών.

5.3.1.2.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9894293419420878

Πίνακας 23: Αποτελέσματα γραμμικού SVM χωρίς επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.94	0.98	0.96	12859

BENIGN	1.00	0.99	0.99	88743
accuracy			0.99	101602
macro avg	0.97	0.98	0.98	101602
weighted avg	0.99	0.99	0.99	101602

5.3.1.2.2 Επιλογή χαρακτηριστικών

Accuracy: 0.904509753745005

Πίνακας 24: Αποτελέσματα γραμμικού SVM με επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.89	0.28	0.43	12859
BENIGN	0.91	0.99	0.95	88743
accuracy			0.90	101602
macro avg	0.90	0.64	0.69	101602
weighted avg	0.90	0.90	0.88	101602

5.3.1.3 SVM-Kernel

Το SVM με συνάρτηση πυρήνα όπως ήταν αναμενόμενο παρουσίασε πολύ μεγάλη βελτίωση σε σχέση με του γραμμικού διαχωρισμού. Η διαφορά μεταξύ των μοντέλων των δύο διανυσμάτων χαρακτηριστικών ήταν συνολικά μικρή.

5.3.1.3.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9878447274659947

Πίνακας 25: Αποτελέσματα SVM πολυωνυμικού πυρήνα χωρίς επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.91	1.00	0.95	12859
BENIGN	1.00	0.99	0.99	88743

accuracy			0.99	101602
macro avg	0.96	0.99	0.97	101602
weighted avg	0.99	0.99	0.99	101602

5.3.1.3.2 Επιλογή Χαρακτηριστικών

Accuracy: 0.9849018720103935

Πίνακας 26: Αποτελέσματα SVM πολυωνυμικού πυρήνα με επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.93	0.93	0.94	12859
BENIGN	0.99	0.99	0.99	88743
accuracy			0.98	101602
macro avg	0.96	0.97	0.97	101602
weighted avg	0.99	0.98	0.99	101602

5.3.1.4 MLP

5.3.1.4.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9906990019881499

Πίνακας 27: Αποτελέσματα MLP χωρίς επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.95	0.98	0.96	12859
BENIGN	1.00	0.99	0.99	88743
accuracy			0.99	101602
macro avg	0.97	0.99	0.98	101602
weighted avg	0.99	0.99	0.99	101602

5.3.1.4.2 Επιλογή χαρακτηριστικών

Accuracy: 0.98717544930217

Πίνακας 28: Αποτελέσματα MLP με επιλογή χαρακτηριστικών για το UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.94	0.96	0.95	12859
BENIGN	0.99	0.99	0.99	88743
accuracy			0.99	101602
macro avg	0.97	0.98	0.97	101602
weighted avg	0.99	0.99	0.99	101602

5.3.1.5 GAN

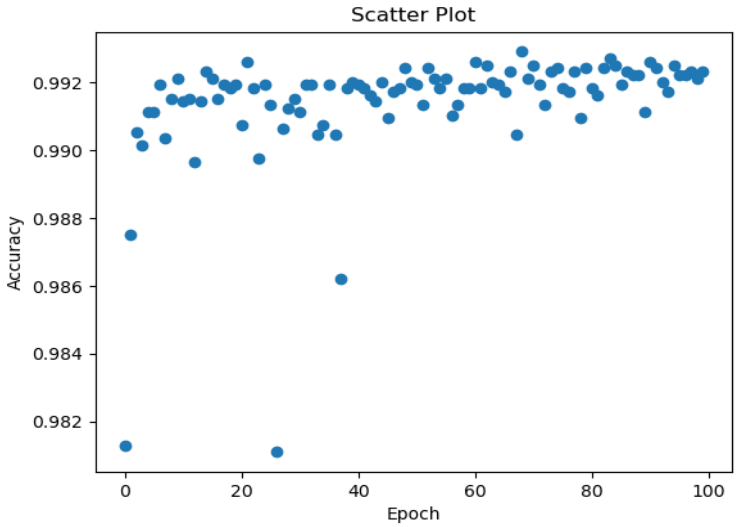
Ακολουθήθηκε η ίδια διαδικασία όπως και με το CIC-IDS2017. Σε αυτό το dataset το GAN έδειξε πολύ καλή απόδοση στην περίπτωση της χρήσης όλων των χαρακτηριστικών, με ελαφρώς καλύτερα αποτελέσματα απ' του CIC-IDS2017, αλλά ταυτόχρονα παρουσίασε μεγάλη απόκλιση στην ικανότητα του να ανιχνεύσει τις επιθέσεις όταν έγινε χρήση του μειωμένου διανύσματος χαρακτηριστικών, με το Recall να πέφτει απ' το 97% στο 77%. Κάτι τέτοιο μπορεί να ερμηνευθεί απ' την πολύ μεγαλύτερη μείωση του αριθμού των χαρακτηριστικών που εξετάστηκαν σε σχέση με το CIC-IDS2017. Ωστόσο, επειδή τα ρηχά μοντέλα καταφέρνουν πολύ καλύτερες επιδόσεις στην εκπαίδευση με την επιλογή χαρακτηριστικών, περεταίρω έρευνα σχετικά με την υπερπαραμετροποίηση του GAN, όπως για παράδειγμα μέσω μιας πιο πολύπλοκης αρχιτεκτονικής των δικτύων του Γεννήτορα και Διευκρινιστή με περισσότερα επίπεδα, πιθανότατα θα οδηγούσε σε βελτίωση της απόδοσης. Στην Εικόνα 21 φαίνεται ότι το GAN ήταν ιδιαίτερα σταθερό και πρακτικά δεν παρουσίασε μεταβολές στις συνολικές τιμές accuracy όταν εκπαιδεύτηκε με όλα τα χαρακτηριστικά, ιδιαίτερα σε σύγκριση με την περίπτωση της επιλογής χαρακτηριστικών (Εικόνα 22).

5.3.1.5.1 Όλα τα χαρακτηριστικά

Accuracy: 0.9929133858267717

Πίνακας 29: Αποτελέσματα GAN χωρίς επιλογή χαρακτηριστικών για το UNSW-NB15

	Precision	Recall	F1-Score
BENIGN	0.995493	0.996392	0.995942
ATTACK	0.975078	0.96904	0.97205



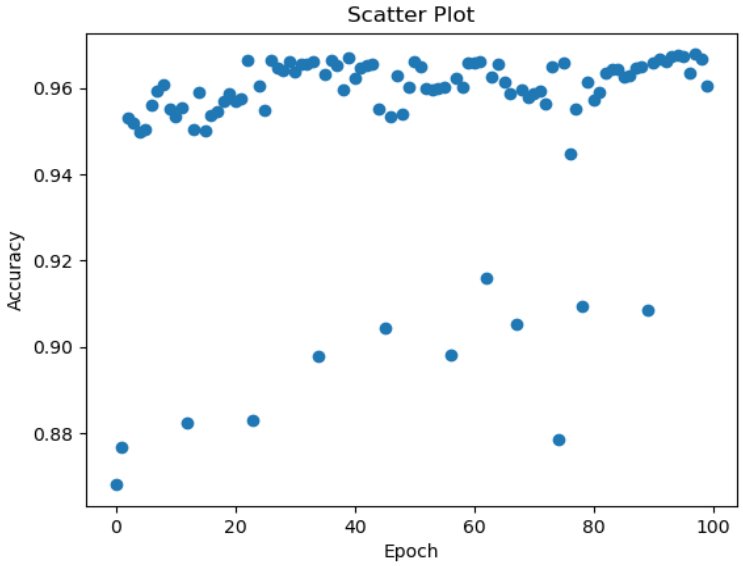
Εικόνα 21: Ακρίβεια ανά αριθμό εποχών χωρίς επιλογή χαρακτηριστικών με το UNSW-NB15

5.3.1.5.2 Επιλογή χαρακτηριστικών

Accuracy: 0.9678149606299212

Πίνακας 30: Αποτελέσματα GAN με επιλογή χαρακτηριστικών για το UNSW-NB15

	Precision	Recall	F1-Score
BENIGN	0.967346	0.996726	0.981816
ATTACK	0.971954	0.771297	0.860077



Εικόνα 22: Ακρίβεια ανά αριθμό εποχών με επιλογή χαρακτηριστικών με το UNSW-NB15

5.3.2 Αποτελέσματα για το online σύστημα

Στην εκτέλεση του online IDS όλα τα μοντέλα κατέταξαν σωστά όλες ή σχεδόν όλες τις συνδέσεις ως κανονικής λειτουργίας με την εξαίρεση του Δέντρου Απόφασης που τις κατέταξε όλες στην κατηγορία των επιθέσεων. Χρειάζεται περαιτέρω διερεύνηση γιατί εμφανίζεται αυτή η συμπεριφορά.

Πίνακας 31: Αποτελέσματα online συστήματος εκπαιδευμένο με το UNSW-NB15

	BENIGN	ATTACK	Support
Δένδρο Απόφασης	0	99	99
SVM-Linear	87	0	87
SVM-Kernel	75	0	75
MLP	88	0	88
GAN	75	2	77

5.4 Εκπαίδευση με CIC-IDS2017, έλεγχος με UNSW-NB15

Σκοπός αυτής της ενότητας είναι ο έλεγχος του συνόλου δεδομένων UNSW-NB15 με χρήση μοντέλων εκπαιδευμένων με δεδομένα απ' το CIC-IDS2017, δηλαδή θέλουμε να δούμε αν υπάρχει data shift στα δεδομένα των δύο συνόλων δεδομένων, και αν μπορούμε να αναγνωρίσουμε τις ροές δεδομένων που έχουν καταχωρηθεί ως επιθέσεις στο UNSW-NB15, με την γνώση που περιέχεται στα δεδομένα του CIC-IDS2017. Αυτό θα ήταν πολύ χρήσιμο αν μπορούσε να γίνει γιατί ο έλεγχος επιθέσεων με ένα IDS είναι μεγάλης διαγνωστικής αξίας για το αν το IDS δουλεύει στην πράξη, αλλά ταυτόχρονα είναι γενικά μια δύσκολη υπόθεση που προϋποθέτει ετικετοποίηση (labelling) και μεγάλο ανθρώπινο κόστος σε εργατοώρες. Αν και υπάρχουν πολλά διαθέσιμα pcaps αρχεία με καταγραφές επιθέσεων, χωρίς την ετικετοποίηση των ροών δεν μπορούν να χρησιμοποιηθούν για μια ποιοτική και ποσοτική εξέταση του IDS και κατά συνέπεια την χρήση μετρικών. Όπως φαίνεται απ' τους πίνακες που ακολουθούν, δεν είναι δυνατή η αναγνώριση, κυρίως αλλά όχι αποκλειστικά, των επιθέσεων. Αυτό το εύρημα λειτουργεί ως κίνητρο για την επόμενη υποενότητα, όπου εξετάζεται η δημιουργία ενός νέου συνόλου δεδομένων απ' την ένωση αυτών των δύο. Η ένωση των δύο συνόλων μπορεί να οδηγήσει θεωρητικά, στον βαθμό που τα δεδομένα δεν περιέχουν αντιφατικές πληροφορίες, στην δημιουργία ενός μοντέλου που αναγνωρίζει παραδείγματα και απ' τα δύο σύνολα δεδομένων.

5.4.1 Δένδρο Απόφασης

Accuracy: 0.8830042715694573

Πίνακας 32: Αποτελέσματα Δένδρου Απόφασης, εκπαίδευση CIC-IDS2017, έλεγχος UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.53	0.77	0.63	12859
BENIGN	0.96	0.90	0.93	88743
accuracy			0.88	101602
macro avg	0.75	0.84	0.78	101602
weighted avg	0.91	0.88	0.89	101602

5.4.2 SVM-kernel

Accuracy: 0.8643235369382493

Πίνακας 33: Αποτελέσματα SVM πολυωνυμικού πυρήνα, εκπαίδευση CIC-IDS2017, έλεγχος UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.12	0.01	0.02	12859
BENIGN	0.87	0.99	0.93	88743
accuracy			0.86	101602
macro avg	0.50	0.50	0.47	101602
weighted avg	0.78	0.86	0.81	101602

5.4.3 MLP

Accuracy: 0.7353004862109014

Πίνακας 34: Αποτελέσματα MLP, εκπαίδευση CIC-IDS2017, έλεγχος UNSW-NB15

	precision	recall	f1-score	support
ATTACK	0.02	0.02	0.02	12859
BENIGN	0.86	0.84	0.85	88743
accuracy			0.74	101602
macro avg	0.44	0.43	0.43	101602
weighted avg	0.75	0.74	0.74	101602

5.5 Αποτελέσματα για το ενιαίο σύνολο δεδομένων

Σκοπός της ένωσης των συνόλων δεδομένων είναι η εξέταση κατά πόσο ένα μοντέλο μπορεί να συγκεράσει τις πληροφορίες των δύο συνόλων και να μπορεί να ανταποκριθεί με επιτυχία και στα δύο σύνολα δεδομένων. Η διαδικασία με την προεπεξεργασία των συνόλων δεδομένων και οι παράμετροι των μοντέλων που επιλέχθηκαν αναλύονται στην ενότητα 4. Τα αποτελέσματα έδειξαν ότι τα μοντέλα με το Δένδρο Απόφασης καθώς και το MLP μπορούν εν γένει να διαχωρίσουν επιτυχώς τις κλάσεις αλλά η απόδοση τους στην κατηγορία των επιθέσεων είναι ποικίλει από ελάχιστα χαμηλότερη (Δένδρο Απόφασης) έως αρκετά χαμηλότερη (MLP), σε σχέση με τις δύο προηγούμενες εκπαιδεύσεις με τα μεμονωμένα σύνολα δεδομένων εκπαίδευσης και την επαλήθευση με τα αντίστοιχα σύνολα ελέγχου. Και τα δύο SVM δεν κατάφεραν να διαχωρίσουν επιτυχώς τα δεδομένα, με το SVM πολυωνυμικού πυρήνα να παρουσιάζει πάντως υψηλότερη επίδοση σε σχέση με το SVM γραμμικού διαχωρισμού. Το GAN παρουσίασε επίσης δυσκολίες στην αναγνώριση επιθέσεων. Είχε το δεύτερο υψηλότερο Precision αλλά επίσης το δεύτερο χαμηλότερο, μετά το SVM γραμμικού πυρήνα, Recall στο 57%. Επίσης στην εικόνα με την ακρίβεια ανά αριθμό εποχών εκπαίδευσης παρατηρείται ότι ήταν πολύ λιγότερο σταθερό στην συμπεριφορά του σε σχέση με τα υπόλοιπα μοντέλα GAN που εξετάστηκαν προηγουμένως (Εικόνα 23). Διαφορετική υπερπαραμετροποίηση του GAN σε σχέση με τα μοντέλα που εκπαιδεύονται με τα δύο datasets μεμονωμένα πιθανώς να παράξει καλύτερα αποτελέσματα. Το Online IDS φαίνεται πως κληρονομεί την συμπεριφορά του απ' το σύνολο δεδομένων NB15, αφού όπως και στην περίπτωση εκπαίδευσης μόνο με το NB15, το Δένδρο Απόφασης κατατάσσει όλες τις συνδέσεις ως επιθέσεις ενώ τα υπόλοιπα μοντέλα τις κατατάσσουν σωστά ως BENIGN.

5.5.1 Αποτελέσματα επί του evaluation test dataset

Τα αποτελέσματα των μοντέλων με Δένδρα Απόφασης, Μηχανή Υποστήριξης Διανυσμάτων, Multi Layer Perceptron (MLP) και GAN παρατίθενται παρακάτω.

5.5.1.1 Δένδρο Απόφασης

Accuracy: 0.9926172603702256

Πίνακας 35: Αποτελέσματα Δένδρου Απόφασης για το ενιαίο σύνολο

	precision	recall	f1-score	support

ATTACK	0.98	0.97	0.98	32550
BENIGN	0.99	1.00	1.00	169543
accuracy			0.99	202093
macro avg	0.99	0.98	0.99	202093
weight avg	0.99	0.99	0.99	202093

5.5.1.2 SVM-Linear

Accuracy: 0.8951769729777874

Πίνακας 36: Αποτελέσματα γραμμικού SVM για το ενιαίο σύνολο

	precision	recall	f1-score	support
ATTACK	0.94	0.37	0.53	32550
BENIGN	0.89	1.00	0.94	169543
accuracy			0.90	202093
macro avg	0.92	0.68	0.74	202093
weight avg	0.90	0.90	0.88	202093

5.5.1.3 SVM-Kernel

Accuracy: 0.9319174835348082

Πίνακας 37: Αποτελέσματα SVM πολυωνυμικού πυρήνα για το ενιαίο σύνολο

	precision	recall	f1-score	support
ATTACK	0.95	0.61	0.74	32550
BENIGN	0.93	0.99	0.96	169543
accuracy			0.93	202093
macro avg	0.94	0.80	0.85	202093
weighted avg	0.93	0.93	0.93	202093

5.5.1.4 MLP

Accuracy: 0.9747838866264542

Πίνακας 38: Αποτελέσματα MLP για το ενιαίο σύνολο

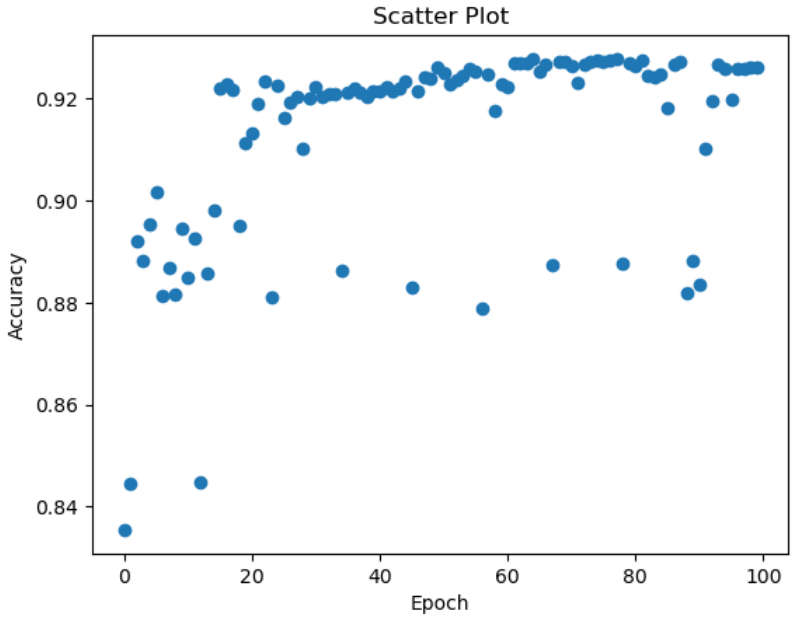
	precision	recall	f1-score	support
ATTACK	0.94	0.90	0.92	32550
BENIGN	0.98	0.99	0.99	169543
accuracy			0.97	202093
macro avg	0.96	0.94	0.95	202093
weight avg	0.97	0.97	0.97	202093

5.5.1.5 GAN

Accuracy: 0.9278539264684051

Πίνακας 39:Αποτελέσματα GAN για το ενιαίο σύνολο

	Precision	Recall	F1-Score
ATTACK	0.962148	0.57611	0.72069
BENIGN	0.924181	0.995633	0.958577



Εικόνα 23: Ακρίβεια ανά εποχή εκπαίδευσης για το ενιαίο σύνολο

5.5.2 Αποτελέσματα για το online σύστημα

Πίνακας 40: Αποτελέσματα online συστήματος εκπαιδευμένο με το ενιαίο σύνολο

	BENIGN	ATTACK	support
Δένδρο Απόφασης	0	122	122
SVM-Linear	132	0	132
SVM-Kernel	86	0	86
MLP	132	0	132
GAN	87	0	87

6 Σύνοψη

Σε αυτή την εργασία εξετάστηκε η υλοποίηση ενός Συστήματος Ανίχνευσης Εισβολής (Intrusion Detection System, IDS) με χρήση Μηχανικής Μάθησης (Machine Learning, ML). Η μηχανική μάθηση έχει την προοπτική να βοηθήσει στην ανάπτυξη πιο ευπροσάρμοστων IDS με καλύτερες δυνατότητες εφαρμογής τους σε μεγάλα δίκτυα με πολλών νέου τύπου επιθέσεων να εμφανίζονται συχνά. Η μηχανική μάθηση επίσης παρουσιάζει το πλεονέκτημα της μικρότερης αναγκαίας παρουσίας του ανθρώπου κατά την διαδικασία ενημέρωσης του συστήματος, συγκριτικά με παλαιότερες μεθόδους που χρησιμοποιούνταν σε IDS όπως βάσεων γνώσης. Σε αυτήν την εργασία υλοποιήθηκαν και εξετάστηκαν τέσσερα συστήματα ανίχνευσης ανωμαλίας (anomaly detection) βασισμένα πάνω σε Δένδρο Απόφασης, Μηχανές Υποστήριξης Διανυσμάτων (Support Vector Machines, SVM) με και χωρίς χρήση συναρτήσεων πυρήνα, Perceptron πολλαπλών επιπέδων (Multi Layer Perceptron, MLP) και ένα νευρωνικό δίκτυο αρχιτεκτονικής Αναγεννητικών Αντιπαραθετικών Δικτύων (Generative Adversarial Networks, GAN). Τα σύνολα δεδομένων (datasets) που χρησιμοποιήθηκαν ήταν τα CIC-IDS2017 και UNSW-NB15, καθώς και μια ένωση αυτών. Αυτά τα datasets είναι σχετικά νέα και αντιμετωπίζουν προβλήματα άλλων παλαιότερων datasets που δεν ανταποκρίνονται στις σημερινές συνθήκες και που χρησιμοποιούνται ακόμη και σήμερα. Πριν την εισαγωγή των δεδομένων στις διάφορες τεχνικές, εφαρμόστηκε επιλογή χαρακτηριστικών και προεπεξεργασία μέσω καθαρισμού και κανονικοποίησης των δεδομένων. Έγινε εξέταση μιας σειράς υπερπαραμέτρων και αρχιτεκτονικών για την βέλτιστη απόδοση των μοντέλων. Η αξιολόγηση των μοντέλων βασίστηκε πάνω στις μετρικές Accuracy, Precision, Recall και F1-Score και ήταν διττή: Τα μοντέλα εξετάστηκαν αφενός πάνω σε σύνολα ελέγχου προερχόμενα απ’ το ίδιο σύνολο δεδομένων εκ των οποίων προήλθαν τα σύνολα εκπαίδευσης τους, αφετέρου εξετάστηκαν πάνω στην αναγνώριση συνδέσεων κανονικής λειτουργίας με ένα online IDS που διάβαζε την κίνηση του δικτύου υπό μορφή ροών (flows) μέσω του λογισμικού NFStream και σε πραγματικό χρόνο αποφάσιζε για το αν πρόκειται για συνδέσεις κανονικής λειτουργίας ή απόπειρα επίθεσης. Το τελευταίο επίσης απουσιάζει απ’ την βιβλιογραφία, με τις περισσότερες έρευνες να περιορίζονται στην εξέταση επί των ίδιων συνόλων δεδομένων χωρίς εξέταση σε πραγματικές συνθήκες. Τα αποτελέσματα ήταν ενθαρρυντικά: Όλα τα μοντέλα που εκπαιδεύτηκαν με τα μεμονωμένα σύνολα δεδομένων εν γένει λειτούργησαν επιτυχώς με συνολικά ποσοστά επιτυχίας άνω του 97% στα σύνολα ελέγχου, με την εξαίρεση του γραμμικού SVM, ενώ και το online IDS αναγνώρισε σωστά τις συνδέσεις ως κανονικής λειτουργίας, εκτός του Δέντρου Απόφασης εκπαιδευμένου με το UNSW-NB15 και του

ενωμένου dataset που αναγνώρισε όλες τις συνδέσεις ως επιθέσεις. Επίσης εξετάστηκε η εκπαίδευση ενός μοντέλου με το CIC-IDS2017, και ο έλεγχος με το UNSW-NB15. Τα αποτελέσματα έδειξαν ότι υπάρχει σημαντικό data shift στα δύο σύνολα δεδομένων και τα μοντέλα αποτυγχάνουν να αναγνωρίσουν τις επιθέσεις. Κατά συνέπεια υλοποιήθηκε τελικά και ένα σύστημα που εκπαιδεύτηκε απ' την ένωση των δύο datasets, το οποίο και παρουσίασε ελαφρώς χειρότερη απόδοση. Η χρησιμοποίηση άλλων συνόλων δεδομένων για τον έλεγχο των μοντέλων πάνω σε επιθέσεις, ή και ενδεχομένως ένας μηχανισμός που αντιμετωπίζει το data shift, καθώς επίσης ο συνδυασμός network και host-based προσεγγίσεων, όπως επίσης και η ενσωμάτωση συστημάτων κανόνων για την αναγνώριση γνωστών επιθέσεων κρίνεται ως ένα καλό αντικείμενο επόμενης μελέτης.

7 Βιβλιογραφική Παραπομπή

[1] M. Bishop, «Risk Analysis,» σε *Computer Security: Art and Science*, Addison-Wesley Professional, 2003, p. 101.

[2] C. Easttom, «Introduction to Computer Security,» σε *Computer Security Fundamentals*, Pearson It Certification, 2019, pp. 2-3.

[3] Z. Ahmad, A. Shahid Khan, C. Wai Shiang, J. Abdullah και F. Ahmad, «Network intrusion detection system: A systematic study of machine learning and deep learning approaches,» *Transactions on Emerging Telecommunications Technologies*, τόμ. 32, αρ. 1, 2020.

[4] T. M. Mitchell, «Well-Posed Learning Problems,» σε *Machine Learning*, McGraw-Hill Science/Engineering/Math, 1997, p. 2.

[5] E. Alpaydin, «What is Machine Learning,» σε *Introduction to Machine Learning*, MIT Press, 2020, pp. 1-4.

[6] T. Hastie, R. Tibshirani και J. Friedman, «Preface to the First Edition,» σε *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*, Springer Science & Business Media, 2009.

[7] S. Raschka και V. Mirjalili, «Regression for predicting continuous outcomes,» σε *Python Machine Learning*, Packt Publishing Ltd, 2017, pp. 4-5.

[8] G. James, D. Witten, T. Hastie και R. Tibshirani, «What Is Statistical Learning?,» σε *An Introduction to Statistical Learning: with Applications in R*, Springer Science & Business Media, 2013, pp. 15-16.

[9] D. T. Larose και C. D. Larose, «Preface,» σε *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley & Sons, 2014.

[10] D. K. Denatius και A. John, «Survey on data mining techniques to enchance intrusion detection,» σε *2012 International Conference on Computer Communication and Informatics, Jan. 2012*.

[11] F. Chollet, «Training, validation, and test sets,» σε *Deep Learning with Python*, Manning Publications, 2017, pp. 97-100.

[12] S. Raschka και V. Mirjalili, «K-fold cross-validation,» σε *Python Machine Learning*, Packt Publishing Ltd, 2017, pp. 197-200.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall και W. P. Kegelmeyer, «SMOTE: Synthetic Minority Over-sampling Technique,» *Journal of Artificial Intelligence Research*, τόμ. 16, pp. 321-357, 2002.

[14] A. Géron, «End-to-End Machine Learning Project,» σε *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019, pp. 37-86.

[15] S. Raschka και V. Mirjalili, «Building Good Training Datasets – Data Preprocessing,» σε *Python Machine Learning*, Packt Publishing, 2019, pp. 109-144.

[16] J. Brownlee, «Prepare Your Data For Machine Learning,» σε *Machine learning mastery with Python: Understand your data, create accurate models, and work projects end-to-end*, 2019, pp. 47-56.

[17] S. Raschka και V. Mirijalili, «Compressing Data via Dimensionality Reduction,» σε *Python Machine Learning*, Packt Publishing Ltd, 2019, pp. 145-190.

[18] S. Raschka και V. Mirjalili, «The three different types of machine learning,» σε *Python Machine Learning*, Packt Publishing Ltd, 2019, pp. 2-7.

[19] A. Géron, «Types of Machine Learning Systems,» σε *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019, pp. 8-15.

[20] H. Liu και B. Lang, «Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,» *Applied Sciences,* τόμ. 9, αρ. 20, p. 5, 2019.

[21] F. Chollet, «Unsupervised learning,» σε *Deep Learning with Python*, Manning Publications, 2017, p. 94.

[22] A. L. Buczak και E. Guven, «A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,» *IEEE Communications Surveys & Tutorials,* τόμ. 18, αρ. 2, p. 1115, 2016.

[23] F. Chollet, «Reinforcement learning,» σε *Deep Learning with Python*, Manning Publications, 2017, p. 95.

[24] S. Raschka και V. Mirjalili, «Making predictions about the future with supervised learning,» σε *Python Machine Learning*, Packt Publishing Ltd, 2019, pp. 3-5.

[25] M. Negnevitsky, «Rules as a knowledge representation technique,» σε *Artificial Intelligence: A Guide to Intelligent Systems*, Pearson, 2005, pp. 26-27.

[26] F. Chollet, «The "deep" in deep learning,» σε *Deep Learning with Python*, Manning Publications, 2017, pp. 8-9.

[27] I. Goodfellow, Y. Bengio και A. Courville, «Introduction,» σε *Deep learning*, MIT press, 2016, pp. 5-8.

[28] H. Liu και B. Lang, «Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,» *Applied Sciences,* τόμ. 9, αρ. 20, pp. 5-11, 2019.

[29] C. Tsai, Y. Hsu, C. Lin και W. Lin, «Intrusion detection by machine learning: A review,» *Elsevier Ltd,* τόμ. 36, αρ. 10, 2009.

[30] H. Liu και B. Lang, «Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,» *Applied Sciences,* τόμ. 9, αρ. 20, p. 19, 2019.

[31] R. Chalapathy, M. Aditya και C. Sanjay, «Anomaly Detection using One-Class Neural Networks,» 2018.

[32] J. Rocca, «"Ensemble methods: bagging, boosting and stacking," Towards Data Science,» 23 Apr 2019. [Ηλεκτρονικό]. Available: https://towardsdatascience.com/ensemble-methods-bagging-boosting-and-stacking-c9214a10a205.

[33] C. M. Bishop, «Decision Theory,» σε *Pattern Recognition and Machine Learning*, Springer Verlag, 2006, pp. 42-43.

[34] S. Raschka και V. Mirjalili, «Looking at different performance evaluation metrics,» σε *Python Machine Learning*, Packt Publishing Ltd, 2019, pp. 211-222.

[35] G. James, D. Witten, T. Hastie και R. Tibshirani, «Linear Discriminant Analysis for p >1,» σε *An Introduction to Statistical Learning:*, Springer Science & Business Media, 2013, pp. 145-146.

[36] A. Géron, «Performance Measures,» σε *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly Media, 2019, pp. 90-101.

[37] A. L. Buczak και E. Guven, «A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,» *IEEE Communications Surveys & Tutorials*, τόμ. 18, αρ. 2, p. 1156, 2016.

[38] H. Liu και B. Lang, «Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,» *Applied Sciences*, τόμ. 9, αρ. 20, pp. 2-5, 2019.

[39] T. V. Nguyen, N. T. Tran και L. T. Sach, «An anomaly-based network intrusion detection system using Deep learning,» σε *2017 International Conference on System Science and Engineering (ICSSE)*, 2017.

[40] R. Chalapathy και S. Chawla, «Deep Learning for Anomaly Detection: A Survey,» 2019.

[41] H. Liu και B. Lang, «Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey,» *Applied Sciences*, τόμ. 9, αρ. 20, 2019.

[42] C. M. Bishop, «Sequential Data,» σε *Pattern Recognition and Machine Learning*, Springer Verlag, 2006, pp. 605-615.

[43] R. Vinayakumar, K. P. Soman και P. Poornachandran, «Applying convolutional neural network for network intrusion detection,» σε *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017.

[44] S. Forrest, S. Hofmeyr και A. Somayaji, «The Evolution of System-Call Monitoring,» σε *2008 Annual Computer Security Applications Conference (ACSAC)*, 2008.

[45] A. L. Buczak και E. Guven, «A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,» *IEEE Communications Surveys & Tutorials*, τόμ. 18, αρ. 2, pp. 1156-1157, 2016.

[46] R. Sommer και V. Paxson, «Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,» σε *2010 IEEE Symposium on Security and Privacy*, 2010.

[47] D. Chou και M. Jiang, «A Survey on Data-driven Network Intrusion Detection,» *ACM Computing Surveys*, τόμ. 54, αρ. 9, pp. 1-36, 2021.

[48] A. L. Buczak και E. Guven, «A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection,» *IEEE Communications Surveys & Tutorials*, τόμ. 18, αρ. 2, pp. 1170-1174, 2016.

[49] M. Tavallaee, E. Bagheri, W. Lu και A. A. Ghorbani, «A Detailed Analysis of the KDD CUP 99 Data Set,» σε *2009 IEEE Symposium on Computational Intelligence in Security and Defense Applications (CISDA 2009)*, 2009.

[50] N. Moustafa και J. Slay, «UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set),» σε *2015 Military Communications and*, 2015.

[51] G. Maciá-Fernández, J. Camacho, R. Magán-Carrión, P. García-Teodoro και R. Therón, «UGR'16: A New Dataset for the Evaluation of Cyclostationarity-Based Network IDSs,» *Computers & Security*, τόμ. 73, pp. 411-424, 2018.

[52] M. Ring, S. Wunderlich, D. Grüdl, D. Landes και A. Hotho, «Flow-based benchmark data sets for intrusion detection,» 2017.

[53] I. Sharafaldin, A. H. Lashkari και A. A. Ghorbani, «Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization,» σε *4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, 2018.

[54] A. Javaid, Q. Niyaz, W. Sun και M. Alam, «A Deep Learning Approach for Network Intrusion Detection System,» σε *Proceedings of the 9th EAI International Conference on*, 2016.

[55] B. Dong και X. Wang, «Comparison deep learning method to traditional methods using for network intrusion detection,» σε *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, 2016.

[56] M. C. Belavagi και B. Muniyal, «Performance Evaluation of Supervised Machine Learning Algorithms for Intrusion Detection,» *Procedia Computer Science*, τόμ. 89, 2016.

[57] A. Brandao και P. Georgieva, «Log Files Analysis For Network Intrusion Detection,» σε *2020 IEEE 10th International Conference on Intelligent Systems (IS)*, 2020.

[58] M. Maithem και G. A. Al-sultany, «Network intrusion detection system using deep neural networks,» *Journal of Physics: Conference Series*, τόμ. 1804, αρ. 1, 2021.

[59] M. He, X. Wang, L. Jin, B. Dai, K. Kacuila και X. Xue, «Malicious Network Behavior Detection Using Fusion of Packet Captures Files and Business Feature Data,» *Sensors*, τόμ. 21, αρ. 17, 2021.

[60] Y. Xiao, C. Xing, T. Zhang και Z. Zhao, «An Intrusion Detection Model Based on Feature Reduction and Convolutional Neural Networks,» *IEEE Access*, τόμ. 7, p. 42210–42219, 2019.

[61] I. J. Goodfellow, J. P. Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville και Y. Bengio, «Generative Adversarial Nets,» 2014. [Ηλεκτρονικό]. Available: https://arxiv.org/abs/1406.2661.

[62] Z. Aouini και A. Pekar, «NFStream,» *Computer Networks*, τόμ. 204, 2022.

8 Παράρτημα

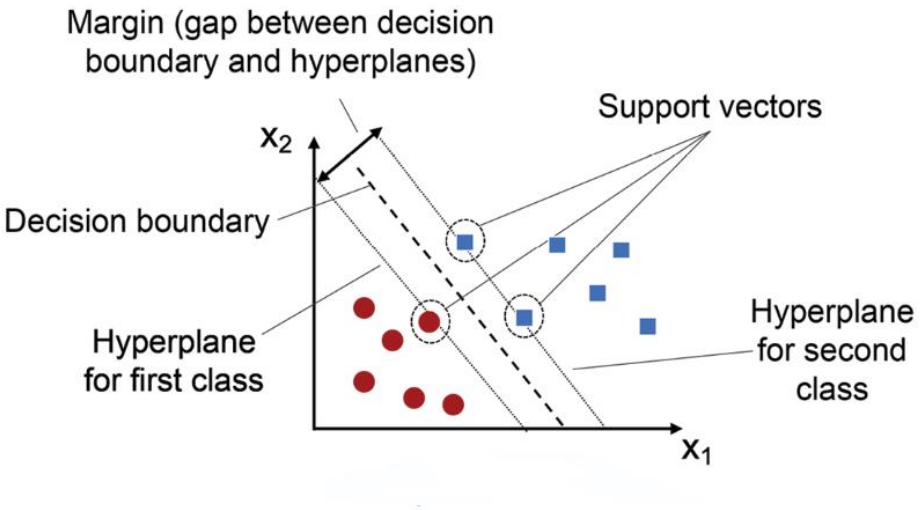
8.1 Επεξήγηση Τεχνικών Μηχανικής Μάθησης

Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks, ANN): Ένας τεράστιος παράλληλος επεξεργαστής κατανεμημένης αρχιτεκτονικής αποτελούμενος από πολλές απλές μονάδες επεξεργασίας (νευρώνες) με την δυνατότητα αποθήκευσης εμπειρικής γνώσης. Η αρχική πηγή έμπνευσης τους ήταν η ικανότητα μη γραμμικής επεξεργασίας πληροφοριών του ανθρώπινου εγκεφάλου. Αποτελούνται από ένα στρώμα εισαγωγής, ένα ή περισσότερα κρυφά στρώματα και ένα εξωτερικό στρώμα. Κάθε στρώμα αποτελείται από έναν αριθμό νευρώνων. Στην βιβλιογραφία αναφέρονται και ως Perceptron πολλαπλών επιπέδων (Multilayer Perceptron, MLP) ή feedforward ANN, επειδή η τοπολογία δεν παρουσιάζει βρόχους και τα δεδομένα πάνε μόνο προς μια κατεύθυνση σε αντιδιαστολή με άλλες νεότερες τοπολογίες, και είναι ιστορικά το πρώτο και απλούστερο είδος νευρωνικού δικτύου που αναπτύχθηκε. Κάθε σύνδεση νευρώνων ή κόμβων αντιστοιχίζεται με ένα συναπτικό βάρος. Εκπαιδεύεται απ' τον αλγόριθμο της *οπισθοδιάδοσης*[37] (back propagation) που σκοπό έχει την επιλογή των καταλληλότερων βαρών έτσι ώστε η παραγόμενη έξοδος να είναι όσο πιο κοντά στην αναμενόμενη βάσει των δεδομένων με ετικέτες και του υπολογισμού μιας συνάρτησης κόστους που ποσοτικοποιεί αυτή την διαφορά. Πιο συγκεκριμένα, η έξοδος του δικτύου, που είναι συνάρτηση των παραμέτρων του δικτύου, εισάγεται ως είσοδος σε μια συνάρτηση κόστους, που συγκρίνει την έξοδο με την ιδανική συμπεριφορά που θέλουμε να έχει. Αυτή η συνάρτηση μπορεί να πάρει πολλές μορφές, με την πιο απλή να είναι η λεγόμενη mean-squared error (MSE) όπου αθροίζονται οι τετραγωνισμένες διαφορές των αποστάσεων μεταξύ των εξόδων και μιας τιμής που ορίζει ο προγραμματιστής ως σωστής (ground truth table) και στην συνέχεια διαιρούνται με το πλήθος των παραδειγμάτων. Ύστερα, το μοντέλο εκπαιδεύεται με τέτοιο τρόπο ώστε οι τιμές της συνάρτησης κόστους να ελαχιστοποιηθούν. Προς αυτόν τον σκοπό χρησιμοποιείται η θεωρία της μαθηματικής ανάλυσης για την ελαχιστοποίηση μιας συνάρτησης και αρχικά υπολογίζονται οι μερικές παράγωγοι ως προς τις παραμέτρους του δικτύου και το άθροισμα τους, δηλαδή η κλίση της συνάρτησης (gradient) πολλών ανεξάρτητων μεταβλητών, εισάγεται στον αλγόριθμο βελτιστοποίησης (gradient descent),

[37] Επαναληπτικός αλγόριθμος υπολογισμού των gradients για την ελαχιστοποίηση μιας συνάρτησης κόστους κατά την εκπαίδευση των νευρωνικών δικτύων. Στην συνέχεια ενεργεί ο αλγόριθμος βελτιστοποίησης των παραμέτρων (gradient descent). Μία στοχαστική αποδοτική παραλλαγή του τελευταίου είναι μεταξύ άλλων ο Adam. Δεν εξασφαλίζει ότι θα βρει το ολικό ελάχιστο αλλά μπορεί να συγκλίνει σε κάποια θέση τοπικού ελαχίστου.

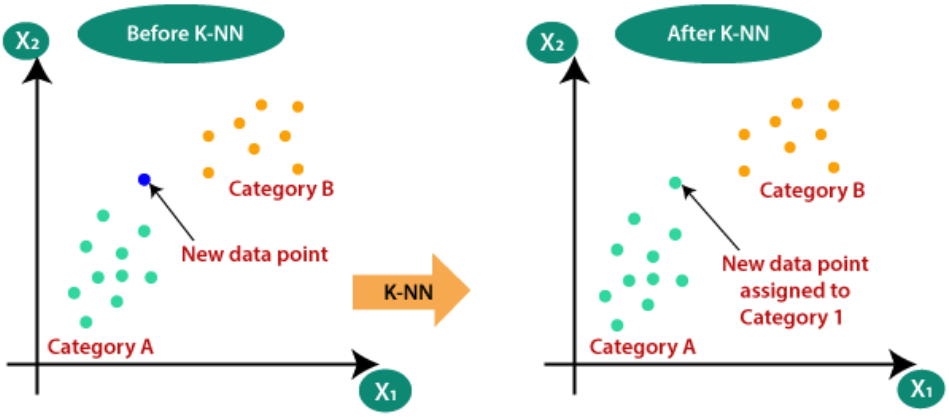
όπου πολλαπλασιασμένο με έναν πολύ μικρό παράγοντα που λέγεται ρυθμός μάθησης (learning rate), αφαιρείται απ' την τρέχουσα τιμή της παραμέτρου, και η διαδικασία επαναλαμβάνεται μέχρι το σφάλμα να μηδενιστεί ή να πέσει κάτω από ένα όριο. Εκτός της βέλτιστης επιλογής βαρών, τα νευρωνικά δίκτυα επίσης κάνουν χρήση πολώσεων (bias), δηλαδή προσθαφαίρεσης ενός αριθμού, και συναρτήσεων ενεργοποίησης (activation functions), με στόχο την μεγαλύτερη ικανότητα προσαρμογής και αντιμετώπιση φαινομένων όπως ασυμμετρίας και μη γραμμικής σχέσης μεταξύ των χαρακτηριστικών και της εξόδου. Οι νευρώνες σε γειτονικά επίπεδα είναι πλήρως συνδεδεμένοι (fully connected) ενώ στους νευρώνες εντός ενός επιπέδου δεν υπάρχουν συνδέσεις μεταξύ τους. Η τοπολογία εν γένει είναι ένα DAG (Directed Acyclic Graph). Αρχικά αποτελούνταν από ένα κρυφό επίπεδο αλλά η πρόοδος της τεχνολογίας έχει επιτρέψει σήμερα την χρησιμοποίηση δικτύων με πολλά κρυφά επίπεδα, δημιουργώντας έτσι το πεδίο του Βαθιάς Μάθησης και την δυνατότητα εκμάθησης πολύπλοκων ιεραρχικών χαρακτηριστικών.

Μηχανές Υποστήριξης Διανυσμάτων (Support Vector Machines, SVM): Στόχος της μεθόδου είναι η εύρεση ενός επιπέδου (εν γένει υπερεπιπέδου) που χωρίζει καλύτερα τα δεδομένα. Επιλέγει το υπερεπίπεδο με την μεγαλύτερη απόσταση απ' τα κοντινότερα εκατέρωθεν δείγματα ή αλλιώς διανύσματα υποστήριξης. Για μη γραμμικά δεδομένα που δεν μπορούν να χωριστούν απευθείας έτσι, χρησιμοποιούνται συναρτήσεις πυρήνα όπως π.χ. πολυωνυμική ή η RBF, με σκοπό την αναγωγή των δειγμάτων σε υψηλότερο διαστατικό χώρο (dimensional space) όπου είναι δυνατός ο γραμμικός διαχωρισμός και η εύρεση υπερεπιπέδου. Επειδή στην πράξη η ύπαρξη ακραίων τιμών συχνά δημιουργεί προβλήματα στην επιλογή ενός «καλού» υπερεπιπέδου, δηλαδή επηρεάζει το πάχος ή εύρος της απόστασης μεταξύ του υπερεπιπέδου και των διανυσμάτων υποστήριξης (περιθώριο/margin), τα SVM υποστηρίζουν την επιλογή μιας μεταβλητής/παραμέτρου (penalty factor), που καθορίζει πόσα δείγματα ακραίων τιμών αποδεχόμαστε ως λάθος ταξινομημένα με στόχο να μεγαλώσουμε το margin και την συνολική αποτελεσματικότητα του αλγορίθμου για την μέση περίπτωση. Είναι πολύ χρήσιμα για περιπτώσεις με πολλά χαρακτηριστικά και λίγα δείγματα.



Εικόνα 24: Μηχανές Υποστήριξης Διανυσμάτων

Κ-Κοντινότεροι-Γείτονες (K-Nearest-Neighbours/KNN): Για ένα καινούργιο διάνυσμα εισαγωγής ή δεδομένο, ο αλγόριθμος υπολογίζει αποστάσεις από προηγούμενα δείγματα εκπαίδευσης και καταχωρίζει το νέο δεδομένο στην κλάση της πλειοψηφίας των Κ κοντινότερων γειτόνων του (τα Κ δείγματα με την μικρότερη απόσταση). Επειδή ο αλγόριθμος απαιτεί την αποθήκευση στην μνήμη των δεδομένων εκπαίδευσης, και ο υπολογισμός περιορίζεται στο στάδιο της εισαγωγής και κατηγοριοποίησης του νέου δεδομένου, λέμε ότι ανήκει στην κατηγορία των instance ή memory based ή lazy μεθόδων μηχανικής μάθησης. Όσο μεγαλύτερο είναι το Κ, τόσο μεγαλύτερη πολυπλοκότητα εισάγεται με κίνδυνο υπερπροσαρμογής. Αντίθετα όσο μικρότερο είναι το Κ τόσο περισσότερο οδεύει προς την τετριμμένη περίπτωση.



Εικόνα 25: Κ-Κοντινότεροι-Γείτονες

Αφελής Bayes (Naive Bayes):

Ο Αφελής Μπάγιες χρησιμοποιεί τον τύπο του Μπάγιες της δεσμευμένης πιθανότητας δηλαδή

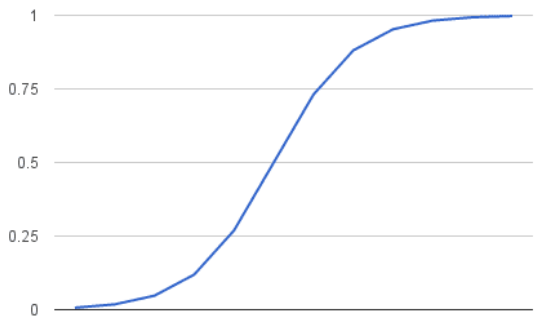
$$P(M|N) = \frac{P(N|M)*P(M)}{P(N)}$$

όπου Μ συμβολίζει κάποιο ενδεχόμενο(π.χ. στο πλαίσιο της μηχανικής μάθησης το δείγμα να ανήκει σε κάποιο σύνολο ή να έχει κάποια ετικέτα), Ν κάποια προϋπόθεση(π.χ. το σύνολο των χαρακτηριστικών που προσδιορίζουν το δείγμα). Η πιθανότητα στον αριθμητή του δεύτερου μέλους ισούται με την από κοινού πιθανότητα, δηλαδή P(M, N) = P(N, M) = P(N|M) * P(M). Ως κοινή πιθανότητα δύο ενδεχομένων ορίζεται η πιθανότητα να συμβούν ταυτόχρονα. Η πιθανότητα στον παρανομαστή είναι ανεξάρτητη του Μ και μπορεί να αγνοηθεί λέγοντας ότι η δεσμευμένη πιθανότητα P(M|N) είναι ανάλογη της πιθανότητας του αριθμητή αφού δεν εξαρτάται απ' το ενδεχόμενο Μ που εξετάζεται. Ο νόμος του Μπάγιες στο πλαίσιο της ταξινόμησης μπορεί να ερμηνευτεί με τον εξής απλό τρόπο: Αν φανταστούμε στον ρόλο του υπολογιστή τον ανθρώπινο εγκέφαλο και στον ρόλο ενός παραδείγματος ένα δένδρο που βλέπει ο άνθρωπος, μπορούμε να πούμε ότι αυτός αντιλαμβάνεται αυτό που βλέπει ως δένδρο, ως το αποτέλεσμα τριών παραγόντων. Ο πρώτος παράγοντας αντιστοιχίζεται στον όρο P(N|M) και δηλώνει την πιθανότητα αν κάτι είναι δένδρο, αυτό να έχει τα χαρακτηριστικά που βλέπει (κλαδιά, χρώμα, ύψος κτλ.). Ουσιαστικά πρόκειται για την ανθρώπινη εμπειρία. Ο δεύτερος όρος είναι το P(M) και δηλώνει ότι όσο περισσότερα είναι τα δένδρα στον κόσμο, τόσο περισσότερο πιθανό είναι ο άνθρωπος να βλέπει ένα δένδρο. Ο τρίτος όρος είναι ο P(N) και δηλώνει πόσο πιθανά είναι τα χαρακτηριστικά που βλέπει και είναι αντιστρόφως ανάλογο της δεσμευμένης πιθανότητας. Πρόκειται ουσιαστικά για το κατά πόσο αυτά τα χαρακτηριστικά που βλέπει ο άνθρωπος (κλαδιά, χρώμα, ύψος κτλ.) είναι δηλωτικά του δένδρου ή μπορεί να εμφανίζονται και σε άλλα πράγματα, καθιστώντας την απόφαση για την κατηγοριοποίηση του αντικειμένου πιο δύσκολη. Ως εκ τούτου, όσο λιγότερο πιθανό είναι να συναντήσει ο άνθρωπος αυτά τα χαρακτηριστικά, τόσο περισσότερο πιθανό είναι όταν τα συναντήσει αυτά να αφορούν δένδρα. Οι δύο πιθανότητες P(N|M) και P(M) μπορούν να υπολογιστούν απευθείας απ' τα δεδομένα. Εναλλακτικά υπολογίζεται η από κοινού πιθανότητα με την υπόθεση της ανεξαρτησίας των χαρακτηριστικών οπότε καταλήγει σε ένα γινόμενο δεσμευμένων πιθανοτήτων P(Y) * (P(X1|Y) * P(X2|Y) ….P(Xn|Y)). Βασική υπόθεση της τελευταίας απλοποίησης και χρήσης του θεωρήματος Μπάγιες είναι η ανεξαρτησία των

χαρακτηριστικών δοθέντος μιας κλάσης. Η απόδοση ποικίλλει ανάλογα με τον βαθμό που αυτή η υπόθεση ισχύει για τα πραγματικά δεδομένα.

Αφελή Μπαγεσιανά Δίκτυα (Naive Bayesian Networks): Είναι ένας ακυκλικός κατευθυνόμενος γράφος με κόμβους που μπορούν να πάρουν μία διακριτή ή συνεχής τιμή από ένα σύνολο τιμών με κάποια πιθανότητα. Το ζητούμενο είναι ο υπολογισμός των πιθανοτήτων των άγνωστων μεταβλητών/καταστάσεων σύμφωνα με το Θεώρημα Δεσμευμένης Πιθανότητας του Bayes. Κάθε κόμβος εξαρτάται μόνο απ' τους γονείς του.

Λογιστική Παλινδρόμηση (Logistic Regression): Στην Λογιστική Παλινδρόμηση, αντίθετα με την γραμμική παλινδρόμηση, η καμπύλη εδώ είναι η λογιστική καμπύλη, ένα είδος σιγμοειδούς καμπύλης με μοιάζει σχηματικά με το λατινικό γράμμα S. Η καμπύλη δείχνει την πιθανότητα κάποιου γεγονότος συναρτήσει ενός γραμμικού συνδυασμού ενός ή περισσοτέρων ανεξάρτητων μεταβλητών. Στην μηχανική μάθηση η απλή εκδοχή της χρησιμοποιείται κυρίως στην περίπτωση της δυαδικής ταξινόμησης αλλά υπάρχει και επέκταση της για την περίπτωση πολλών κλάσεων, όπου εμφανίζεται με πολλά ονόματα, και μπορεί να χρησιμοποιηθεί στο τελευταίο στρώμα των νευρωνικών δικτύων για την τελική απόφαση. Ως παράδειγμα, στην περίπτωση ενός χαρακτηριστικού και ενός δυαδικού αποτελέσματος, ο οριζόντιος άξονας θα δείχνει τις τιμές του χαρακτηριστικού (ισοδύναμα τον πολλαπλασιαστικό παράγοντα του χαρακτηριστικού στην γενικευμένη περίπτωση του γραμμικού συνδυασμού πολλαπλών χαρακτηριστικών) και ο οριζόντιος την πιθανότητα του γεγονότος, που στο πλαίσιο μιας ταξινόμησης θα μπορούσε να είναι η πιθανότητα ενός παραδείγματος να ανήκει σε κάποια κλάση, και αν αφαιρεθεί αυτή η τιμή απ' την μονάδα υπολογίζεται η πιθανότητα να ανήκει στην συμπληρωματική κλάση. Η μαθηματική έκφραση της συνάρτησης της καμπύλης είναι L(x) = e^x / (1 + e^x) όπου x στην πράξη είναι τα χαρακτηριστικά του παραδείγματος και L(x) είναι η πιθανότητα συμμετοχής στην πρώτη κλάση. Σε έναν αλγόριθμο ταξινόμησης τα χαρακτηριστικά δεν θα δίνονται όπως έχει αλλά οι τιμές τους θα υποστούν κάποια γραμμική επεξεργασία με κάποιους παραμέτρους που μαθαίνει το μοντέλο εκπαίδευσης μέσα από ένα σύνολο δεδομένων, μιας συνάρτησης κόστους και ενός βελτιστοποιητή. Όπως φαίνεται απ' την εικόνα, οι τιμές της L(x) είναι μεταξύ 0 και 1, όπως αρμόζει σε πιθανότητες.



Εικόνα 26: Λογιστική καμπύλη

Δένδρα Απόφασης (Decision Trees): Τα Δέντρα Απόφασης ταξινομούν ένα δείγμα σύμφωνα με μια ιεραρχική ακολουθία αποφάσεων δενδρικής μορφής στους κόμβους απόφασης, δηλαδή κόμβους που έχουν παιδιά, σύμφωνα με κάποιο κριτήριο που αφορά κάποια απ' τα χαρακτηριστικά του. Η αναζήτηση ξεκινάει απ' τον κόμβο ρίζα, που χρησιμοποιεί το σύνολο των δεδομένων και το χαρακτηριστικό με την μεγαλύτερη διακριτική ικανότητα, συνεχίζει ανάλογα σε χαμηλότερα επίπεδα με άλλα χαρακτηριστικά χαμηλότερης διακριτικής ικανότητας και καταλήγει σε κόμβους-φύλλα που δεν έχουν παιδιά και αναπαριστούν κάποια ετικέτα/κατηγορία. Γνωστό παράδειγμα προγράμματος που κατασκευάζει Δέντρα Απόφασης είναι το CART. Τα Δέντρα Απόφασης που δεν ταξινομούν το δείγμα σε διακριτές/συμβολικές ετικέτες αλλά σε συνεχείς αριθμητικές τιμές λέγονται Δέντρα Παλινδρόμησης, κατά αναλογία με την Γραμμική Παλινδρόμηση όπου βρίσκουμε μια κατάλληλη ευθεία για να προβλέψουμε νέες τιμές. Η μαθηματική θεωρία πάνω στην οποία βασίζονται τα δένδρα απόφασης για την επιλογή του καταλληλότερου χαρακτηριστικού σε κάθε κόμβο πηγάζει απ' το πεδίο της Θεωρίας Πληροφορίας. Οι δύο πιο συνηθισμένοι τρόποι είναι το Κέρδος Πληροφορίας (Information Gain) και Αγνότητα Gini (Gini Impurity). Πολλά δένδρα που εκπαιδεύονται ανεξάρτητα και τα αποτελέσματα τους συνδυάζονται συνθέτουν το *Τυχαίο Δάσος* (Random Forest) που αποτελεί χαρακτηριστικό παράδειγμα συνεργατικής μεθόδου. Για σκοπούς ταξινόμησης η έξοδος από ένα Τυχαίο Δάσος είναι η κλάση που έχει επιλεγεί απ' τα περισσότερα δένδρα απόφασης. Τα Δένδρα Απόφασης είναι πολύ εύκολα ερμηνεύσιμα ενώ το Τυχαίο Δάσος σε γενικές γραμμές υπερτερεί σε επίπεδο προσαρμοστικότητας από ένα μεμονωμένο Δένδρο Απόφασης.

Εξελικτικός Υπολογισμός (Evolutionary Computation): Είναι μία ευρεία οικογένεια αλγορίθμων και παραδειγμάτων ανάπτυξης αλγορίθμων που περιλαμβάνει προσεγγίσεις όπως Γενετικούς Αλγόριθμους (Genetic Algorithms ή GA), Γενετικό Προγραμματισμό (Genetic Programming ή GP), Εξελικτικές Στρατηγικές, Βελτιστοποίηση Σμήνους Σωματιδίων (Particle Swarm Optimization ή PSO), Βελτιστοποίηση Αποικίας Μυρμηγκιών (Ant Colony Optimization) και Τεχνητά Ανοσοποιητικά Συστήματα (Artificial Immune Systems). Όπως φανερώνουν τα ονόματα τους είναι εμπνευσμένοι απ' την βιολογία. Οι δύο πρώτοι συγκεκριμένα είναι εμπνευσμένοι απ' την γενετική, την θεωρία της φυσικής επιλογής και την επιβίωση του ισχυρότερου. Οι GA χρησιμοποιούν ένα μεγάλο πλήθος, αρχικά τυχαίων, υποψηφίων λύσεων και πολλές επαναλήψεις για να απορρίψουν τις χαμηλής επίδοσης λύσεις και να τις αντικαταστήσουν από συνδυασμούς λύσεων υψηλότερης επίδοσης, μέσω διαδικασιών που αντανακλούν τις φυσικές διεργασίες της μίξης οργανισμών που δημιουργούν καλύτερους απογόνους και της μετάλλαξης που προκαλεί τυχαίες μεταβολές στον γενετικό κώδικα ενός οργανισμού. Είναι περισσότερο μία φιλοσοφία ανάπτυξης ευριστικών αλγορίθμων παρά ένας αλγόριθμος με συγκεκριμένα βήματα και οι λεπτομέρειες ανάπτυξης ποικίλουν από πρόβλημα σε πρόβλημα. Εξαιτίας της γενικότητας της φιλοσοφίας, γενετικοί αλγόριθμοι βρίσκουν εφαρμογή σε πολλά διαφορετικά πεδία όπως βελτιστοποίηση, επιλογή χαρακτηριστικών και υπερπαραμέτρων, ακόμα και σχεδιασμού ενός μοντέλου. Οι GP επεκτείνουν την ιδέα των γενετικών αλγορίθμων για να συμπεριλάβουν προγράμματα και μαθηματικές εκφράσεις ως λύσεις, εκτός από π.χ. αριθμητικές τιμές. Χρησιμοποιούνται για σκοπούς συμβολικής παλινδρόμησης, δηλαδή την εύρεση κλειστού τύπου μαθηματικής έκφρασης που να εξηγεί τα δεδομένα ενός συνόλου, συστημάτων ελέγχου και την βέλτιστη αρχιτεκτονική νευρωνικών δικτύων. Οι PSO και ACO είναι αλγόριθμοι βελτιστοποίησης και προσομοιώνουν την κοινωνική συμπεριφορά των πτηνών και μυρμηγκιών αντίστοιχα, με τα πρώτα να συγκεντρώνονται σε συγκεκριμένες περιοχές του χώρου, και τα δεύτερα να βρίσκουν την βέλτιστη διαδρομή για την εύρεση τροφής.

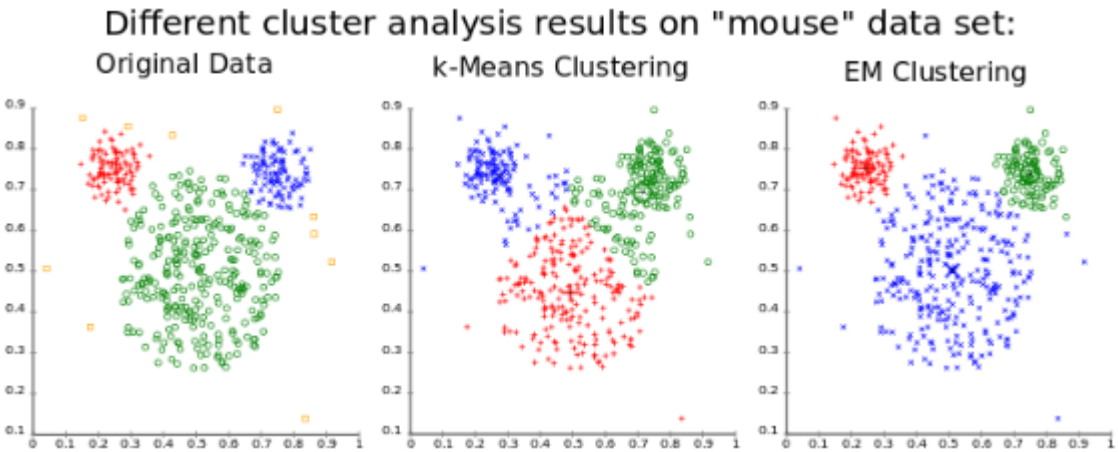
K-μέσων (K-Means): Centroid-based αλγόριθμος συσταδοποίησης και περίπτωση αλγορίθμου διχοτόμησης. Είναι ο πιο συχνός αλγόριθμος συσταδοποίησης. Χρησιμοποιεί την απόσταση για την μετρική ομοιότητας. Όσο πιο μικρή είναι η απόσταση μεταξύ δύο δειγμάτων, τόσο πιο πιθανό να καταλήξουν στην ίδια συστάδα. Λόγω του ότι είναι απλός και σχετικά γρήγορος, χρησιμοποιείται συχνά ως ένα βήμα προεπεξεργασίας σε άλλους αλγορίθμους για την εύρεση μιας «καλής» αρχικής παραμετροποίησης.

Gaussian Mixture Model algorithm: Το συγκεκριμένο μοντέλο υιοθετεί την ιδέα ότι τα δεδομένα προέρχονται από έναν συνδυασμό γκαουσιανών κατανομών και προσπαθεί να βρει τις κατάλληλες παραμέτρους τους. Κάθε διαφορετική γκαουσιανή κατανομή υποτίθεται ότι μοντελοποιεί μία συστάδα.

DBSCAN: Ο πιο διαδεδομένος αλγόριθμος των τεχνικών συσταδοποίησης που χρησιμοποιούν την έννοια της πυκνότητας για τον διαχωρισμό των ομάδων. Οι περιοχές με υψηλή πυκνότητα δεδομένων θεωρείται ως ομάδα, ενώ αντικείμενα σε περιοχές χαμηλής πυκνότητας αντιμετωπίζονται ως θόρυβος ή σύνορο ομάδας.

Για πληρότητα αναφέρονται και κάποιοι άλλοι αλγόριθμοι συσταδοποίησης όπως BIRCH, Affinity Propagation, Mean-Shift, Optics, Agglomerative Hierarchy, Divisive Hierarchical, Mini-Batch K-Means, Spectral Clustering, K-Medoids, Fuzzy C-Means.

Υπενθυμίζεται εδώ ότι όπως αναφέρθηκε και στην εισαγωγή, οι τεχνικές συσταδοποίησης κάνουν κάποιες παραδοχές για τα δεδομένα με αποτέλεσμα κάποια συγκεκριμένη μέθοδος συσταδοποίησης να δουλεύει καλά σε κάποια σύνολα δεδομένων ενώ αποτυγχάνει σε κάποια άλλα. Αυτό εξηγεί εν μέρει και το μεγάλο πλήθος των διαφορετικών παραλλαγών συσταδοποίησης. Παραδείγματος χάριν, όπως φαίνεται στην Εικόνα 27 για το γνωστό σύνολο δεδομένων «mouse», εκεί όπου ο K-Means αποτυγχάνει, ο EM πετυχαίνει.



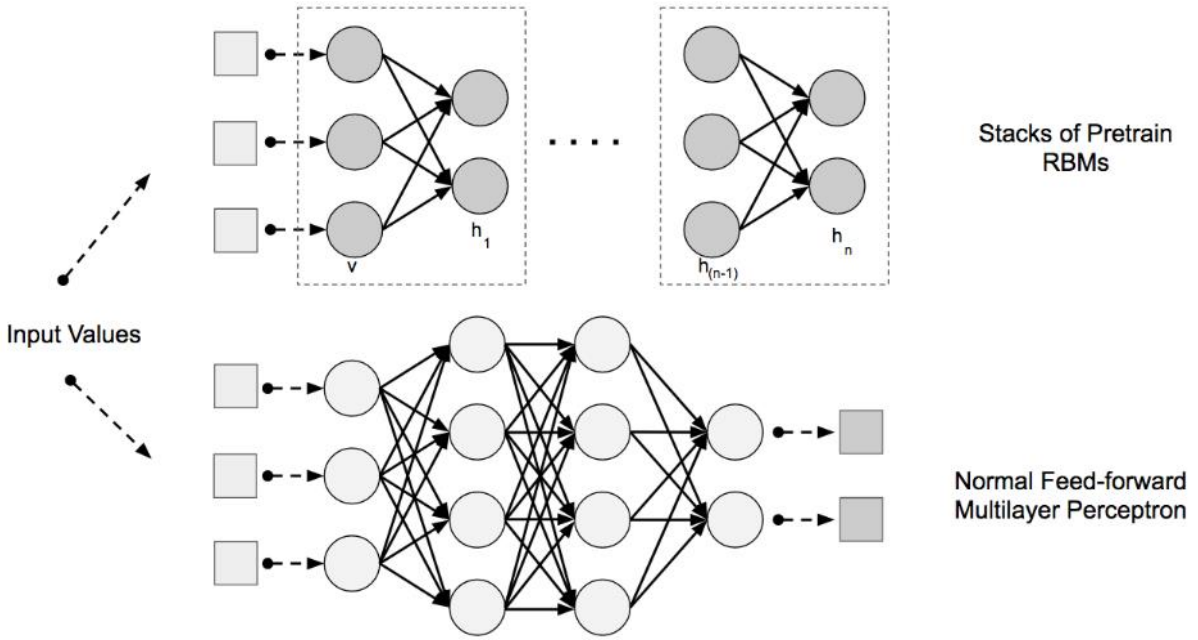
Εικόνα 27: Τεχνικές Συσταδοποίησης

Κανόνες Συσχέτισης και Ασαφείς Κανόνες Συσχέτισης: Οι κανόνες συσχέτισης έχουν περιγραφεί προηγουμένως. Εν συντομία ένας κανόνας συσχέτισης περιγράφει μία σχέση μεταξύ δύο ή περισσοτέρων χαρακτηριστικών. Οι κανόνες είναι if/then statements που ισχύουν

με κάποια πιθανότητα. Χρησιμοποιούν δύο στατιστικά κριτήρια: Υποστήριξη και Εμπιστοσύνη. Για αριθμητικά και κατηγορικά δεδομένα μπορεί να χρησιμοποιηθεί και η ασαφής επέκταση της που βασίζεται στους κανόνες της *Ασαφούς Λογικής* (Fuzzy Logic). Βασίζεται την ιδέα ότι, όπως και στη φύση, φαινόμενα μπορεί να ισχύουν σε κάποιο βαθμό, και όχι απόλυτα ναι ή όχι. Η ιδέα επεκτείνεται στην συμμετοχή κάποιου δείγματος σε μια ομάδα ή κατηγορία.

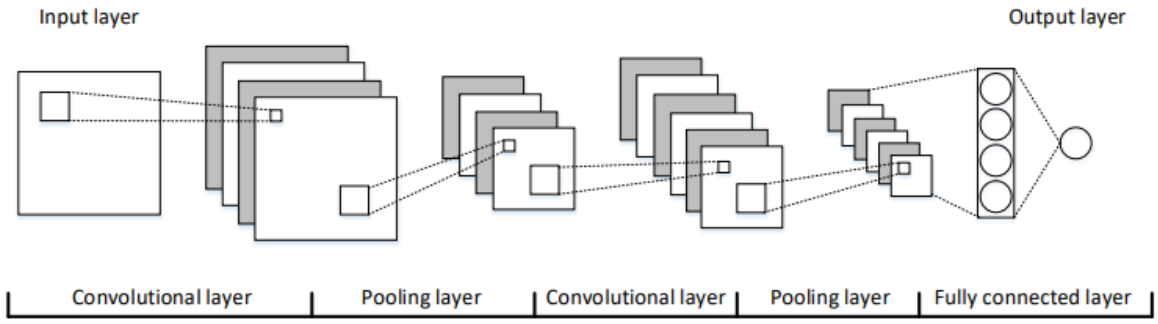
Δίκτυα Βαθιάς Πίστης (Deep Belief Networks ή DBN): Αποτελούνται από Περιορισμένες Μηχανές Boltzmann (RBM) σε σειρά και στο τέλος ένα επίπεδο ταξινόμησης softmax. Οι RBMs αποτελούν μέθοδο μη επιβλεπόμενης μάθησης και περιγράφονται αναλυτικά σε επόμενη υποπαράγραφο. Απαιτεί δύο στάδια εκπαίδευσης. Μιας «άπλειστης» (greedy) μεθόδου μη επιβλεπόμενης προ-εκπαίδευσης όπου γίνεται ο καθορισμός των βαρών μεταξύ κάθε RBM επιπέδων, ακολουθούμενης από μιας επιβλεπόμενης για την βελτιστοποίηση των βαρών με στόχο την ελαχιστοποίηση της συνάρτησης απωλειών (ή κόστους) κάποιου συγκεκριμένου σκοπού μέσω Gradient Descent και του αλγορίθμου οπισθοδιάδοσης.

Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks ή DNN): Αποτελείται από πολλά κρυμμένα επίπεδα. Εκπαιδεύεται συνήθως αρχικά μη επιβλεπόμενα και στη συνέχεια επιβλεπόμενα.



Εικόνα 28: RBMs και MLPs

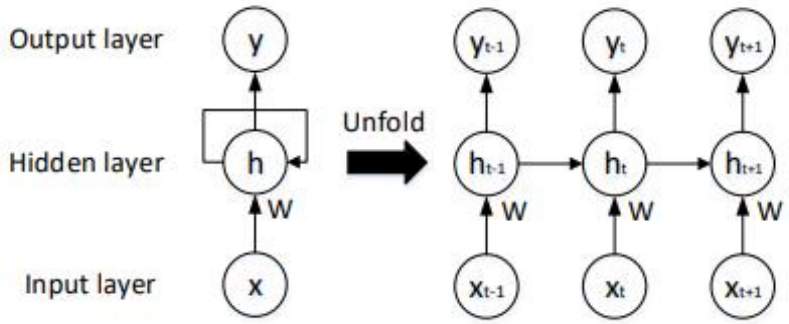
Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks ή CNN): Μοντέλο προσομοίωσης του ανθρώπινου οπτικού συστήματος. Δέχεται ως είσοδο πίνακες. Εναλλαγή συνελικτικών και συγκεντρωτικών (pooling) στρωμάτων ή επιπέδων (layers). Τα συνελικτικά στρώματα κάνουν εξαγωγή χαρακτηριστικών με τον τύπο της συνέλιξης και τα συγκεντρωτικά έχουν ως στόχο την γενίκευση των χαρακτηριστικών. Τα συνελικτικά νευρωνικά δίκτυα απαιτούν ως είσοδο δισδιάστατα δεδομένα, όποτε τα δεδομένα εισαγωγής πρέπει με κάποιον τρόπο να μετατραπούν σε πίνακες.



Εικόνα 29: Αρχιτεκτονική Συνελικτικών Δικτύων

Αναδρομικά Νευρωνικά Δίκτυα (Recurrent Neural Networks ή RNN): Χρησιμοποιείται για ακολουθιακά δεδομένα και στην επεξεργασία φυσικής γλώσσας. Η ιδέα είναι στην δημιουργία κάποιου είδους μνήμης για κατανόηση συμφραζομένων. Κάνει χρήση βρόχων όπου η έξοδος των κρυφών επιπέδων μετά την συνάρτηση ενεργοποίησης, για κάποιο παράδειγμα X_t, πολλαπλασιασμένη με ένα νέο βάρος, αθροίζεται μαζί με την είσοδο της συνάρτησης ενεργοποίησης για το επόμενο παράδειγμα X_{t+1} και ούτω καθεξής. Ισοδύναμα, με ξεδίπλωμα του βρόχου έχουμε μία αρχιτεκτονική ίδιων «απλών» νευρωνικών δικτύων που μοιράζονται τις ίδιες παραμέτρους και το πλήθος των οποίων ταυτίζεται με το πλήθος των ακολουθιακών δεδομένων (Εικόνα 30). Διάφορες παραλλαγές έχουν εφευρεθεί για να λύσουν το πρόβλημα της εξαφανιζόμενης (ή εκρηγνυόμενης) κλίσης[38] (vanishing ή exploding gradients): Bi-RNN, Δίκτυα Μακράς Βραχύχρονης Μνήμης (Long Short-Term Memory ή LSTM), GRU (Gated Recurrent Unit).

[38] Προβλήματα που εμφανίζονται όταν οι τιμές των κλίσεων μπορεί να είναι πολύ μεγάλες ή πολύ μικρές και κατά συνέπεια οι ενημερωμένες τιμές των παραμέτρων πολύ μικρές ή πολύ μεγάλες αντίστοιχα, δυσκολεύοντας έτσι την εύρεση των βέλτιστων τιμών τους που ελαχιστοποιούν την συνάρτηση κόστους.



Εικόνα 30: Δομή RNNs

Παραγωγικά Αντιπαραθετικά Δίκτυα (Generative Adversarial Networks ή GAN): Αποτελείται από δύο δίκτυα: Έναν Γεννήτορα και έναν Διευκρινιστή. Ο Γεννήτορας προσπαθεί να παράξει συνθετικά δεδομένα που μοιάζουν με τα πραγματικά. Ο Διευκρινιστής προσπαθεί να τα διαχωρίσει. Το ένα δίκτυο αντιπαλεύεται το άλλο ώστε να βελτιωθεί. Είναι χρήσιμα για την παραγωγή νέων δεδομένων όπου τα πραγματικά δεν επαρκούν.

Αυτοοργανώμενοι Χάρτες (Self-organizing map ή SOM): Οι SOM είναι ένα είδος νευρωνικού δικτύου που χρησιμοποιεί ανταγωνιστική και μη επιβλεπόμενη μάθηση με στόχο την αντιστοίχιση και συσταδοποίηση δεδομένων μεγάλης διαστατικότητας σε χώρο μικρότερης διαστατικότητας, συνηθέστερα δύο διαστάσεων για την απεικόνιση του. Το δίκτυο βρίσκει τον νευρώνα και το αντίστοιχο βάρος του που βρίσκεται πιο κοντά σε κάποιο δείγμα εκπαίδευσης και τον μετακινεί προς σε αυτό.

Περιοριστικές Μηχανές Boltzmann (Restricted Boltzmann Machines ή RBM): Είναι ένα τεχνητό νευρωνικό δίκτυο δύο επιπέδων, το ορατό και το κρυμμένο, όπου κάθε κόμβος του πρώτου συνδέεται με κάθε κόμβο του δεύτερου. Οι κόμβοι στο ορατό επίπεδο αναπαριστούν τα δεδομένα εισόδου, ενώ οι κόμβοι στα κρυμμένα επίπεδα αναπαριστούν τα χαρακτηριστικά των δεδομένων που έμαθε το δίκτυο. Αντίθετα απ' τις «απλές» Μηχανές Boltzmann, οι κόμβοι εντός του ίδιου επιπέδου δεν επικοινωνούν μεταξύ τους. Το αποτέλεσμα του δικτύου είναι η παραγωγή μια πιθανοτικής κατανομής των εισόδων του. Πιο συγκεκριμένα το κρυφό επίπεδο δίνει την έξοδο ή το αποτέλεσμα μέσω ενός ζυγισμένου αθροίσματος των εισόδων και της συνάρτησης ενεργοποίησης. Το μοντέλο μπορεί να επεκταθεί σε μορφή στοίβας όπου κάθε κρυμμένο επίπεδο μιας RBM ταυτίζεται με το ορατό επίπεδο της επόμενης, όπως αναφέρθηκε άλλωστε και για τα Δίκτυα Βαθιάς Πίστης προηγουμένως. Τα RBMs παίρνουν το όνομα τους απ' την κατανομή Boltzmann που εμφανίζεται στον κλάδο της στατιστικής μηχανικής και

περιγράφει την πιθανότητα ενός συστήματος να καταλάβει κάποια κατάσταση δοθέντος της θερμοκρασίας του, και καθορίζεται απ' την ενέργεια του συστήματος σε αυτήν την κατάσταση. Στο πλαίσιο των RBMs κάθε νευρώνας καταλαμβάνει μια τιμή ενέργειας και η πιθανότητα μιας δοσμένης συνδεσμολογίας των ορατών και κρυφών νευρώνων είναι ανάλογη της κατανομής Boltzmann της ενέργειας αυτής της συνδεσμολογίας. Τα RBM μαθαίνουν να προσαρμόζουν τα βάρη μεταξύ των επιπέδων ώστε να ελαχιστοποιήσουν την ενέργεια του συστήματος, ή ισοδύναμα να μεγιστοποιήσουν την πιθανότητα των παραγόμενων δεδομένων. Τον ρόλο του εκπαιδευτή παίζει ο αλγόριθμος της Αντίθετης Διακλάδωσης (Contrastive Divergence). Η διαδικασία συνοψίζεται ως εξής: είσοδος δεδομένων, υπολογισμός των τιμών της συνάρτησης ενεργοποίησης, ανακατασκευή της εισόδου απ' τις κρυμμένες μονάδες, υπολογισμός της αφαίρεσης των τιμών εισόδου και ανακατασκευασμένων εισόδων και προσαρμογή των βαρών μέσω ενός αλγορίθμου gradient descent. Οι Περιοριστικές Μηχανές Boltzmann ανήκουν στην κλάση των αναγεννητικών νευρωνικών δικτύων επειδή μόλις εκπαιδευτούν και μάθουν να ανακατασκευάζουν τα δεδομένα εισόδου μέσω της μοντελοποίησης της κατανομής πιθανότητας των εισόδων, μπορούν ύστερα να χρησιμοποιηθούν για να την παραγωγή νέων δειγμάτων απ' την κατανομή που έμαθαν κατά την διάρκεια της εκπαίδευσης τους.

Αυτόματοι Κωδικοποιητές (Autoencoders): Αποτελείται από δύο συμμετρικά νευρωνικά δίκτυα, τον Κωδικοποιητή και τον Αποκωδικοποιητή. Η έξοδος του Κωδικοποιητή αποτελεί μια εξαγόμενη αναπαράσταση γνώσης (feature learning) της εισόδου ή αλλιώς μια συμπίεση της εισόδου παρόμοια με την τεχνική PCA (για την περίπτωση γραμμικών συναρτήσεων ενεργοποίησης ταυτίζεται με το PCA). Αυτό το διάνυσμα λιγότερων διαστάσεων ονομάζεται bottleneck ή, δανειζόμενοι απ' την ορολογία της στατιστικής, κρυφό διάνυσμα (latent vector) ή κρυφός κώδικας (latent code), οι δε στήλες του μπορούν να θεωρηθούν ως κάποια νέα κρυφά χαρακτηριστικά ή μεταβλητές (latent features ή latent variables αντίστοιχα), τα οποία εν γένει δεν ανταποκρίνονται σε μετρήσιμες ποσότητες και είναι λιγότερο ερμηνεύσιμα αλλά αποτελούν τα ουσιαστικότερα χαρακτηριστικά των αρχικών δεδομένων. Ο Αποκωδικοποιητής δέχεται ως είσοδο τα εξαγόμενα χαρακτηριστικά και προσπαθεί να ανακατασκευάσει τα αρχικά δεδομένα ελαχιστοποιώντας το σφάλμα ανάμεσα στην είσοδο και την έξοδο κατά την φάση εκπαίδευσης του (χωρίς επίβλεψη). Ο πιο συχνός τρόπος που χρησιμοποιούνται οι απλοί Αυτοκωδικοποιητές αποτελεί η χρήση του δικτύου του Κωδικοποιητή ως ένα βήμα προεπεξεργασίας για την μείωση της διαστατικότητας και εξαγωγής χαρακτηριστικών, ή εν γένει ως μιας αρχικής κατάστασης προτού τα δεδομένα περάσουν σε άλλα μοντέλα. Άλλα

παραδείγματα παραλλαγών των Αυτοκωδικοποιητών είναι οι Στοιβαγμένοι Αυτόματοι Κωδικοποιητές, Αραιοί Αυτόματοι Κωδικοποιητές, Αποθορυβοποιητές Αυτόματοι Κωδικοποιητές.

Μια πολύ σημαντική παραλλαγή και επέκταση του απλού αυτοκωδικοποιητή είναι ο λεγόμενος *αυτοκωδικοποιητής διακυμάνσεων* ή μεταβλητός αυτοκωδικοποιητής (variational autoencoder). Σε αυτή την αρχιτεκτονική η έξοδος του κωδικοποιητή δεν είναι ένα fixed κρυφό διάνυσμα που προήλθε απ' το εκάστοτε data point. Αντίθετα, ο κωδικοποιητής παράγει δύο τιμές, για την ακρίβεια δύο διανύσματα, κάθε τιμή των οποίων αντιστοιχίζεται με μια κρυφή μεταβλητή για κάθε data point. Αυτά τα διανύσματα μπορούν να ερμηνευτούν πιθανοτικά ως η μέση τιμή και η τυπική απόκλιση μιας κατανομής για κάθε κρυφή μεταβλητή κάθε παραδείγματος εισόδου (η ερμηνεία αυτή δεν είναι αυθαίρετη αλλά πηγάζει απ' τον ορισμό της συνάρτησης κόστους). Το κρυφό διάνυσμα που θα λάβει ως είσοδο ο αποκωδικοποιητής δημιουργείται μέσω δειγματοληψίας της κατανομής που ορίζεται απ' τα δύο διανύσματα μέσης τιμής και τυπικής απόκλισης που παρήγαγε ο κωδικοποιητής. Το συνολικό δίκτυο εκπαιδεύεται από άκρη σε άκρη με τον αλγόριθμο της οπισθοδιάδοσης, του λεγόμενου parameterization trick[39] και μιας συνάρτησης απωλειών όπου εκτός του σφάλματος ανακατασκευής, συμπεριλαμβάνει έναν δεύτερο όρο που αφορά την ομοιότητα[40] της posterior[41] κατανομής που δημιούργησε ο κωδικοποιητής, και της τυποποιημένης κανονικής κατανομής, που «πριμοδοτούμε» κατά την εκπαίδευση του δικτύου. Δηλαδή το μοντέλο εκπαιδεύεται έτσι ώστε όχι μόνο η έξοδος να ταυτίζεται με την είσοδο αλλά επιπλέον οι τιμές στο bottleneck να ακολουθούν κατά το δυνατόν[42] την κανονική κατανομή. Αυτό γίνεται για τους παρακάτω λόγους: Η κανονική κατανομή είναι «βολική» κατά την διαδικασία δειγματοληψίας της, έχει κλειστό τύπο (closed form) και επειδή εισάγει κανονικοποίηση στον κρυφό χώρο. Το τελευταίο αφορά την ιδιότητα κατά την οποία μέσω δειγματοληψίας απ' την prior[43] κανονική κατανομή (αναφέρεται ως sampling from the latent space), μπορούμε να

[39] Διαφορετικός μαθηματικός τρόπος έκφρασης του δειγματοληπτημένου κρυφού διανύσματος ως $Z = \mu + \sigma * E$ όπου $E \sim N(0, 1)$ αντί για $Z \sim N(\mu, \sigma)$ όπου μ και σ οι παράμετροι που υπολογίστηκαν απ' τον κωδικοποιητή, με σκοπό να μπορεί να προχωρήσει προς τα πίσω ο αλγόριθμος της οπισθοδιάδοσης.

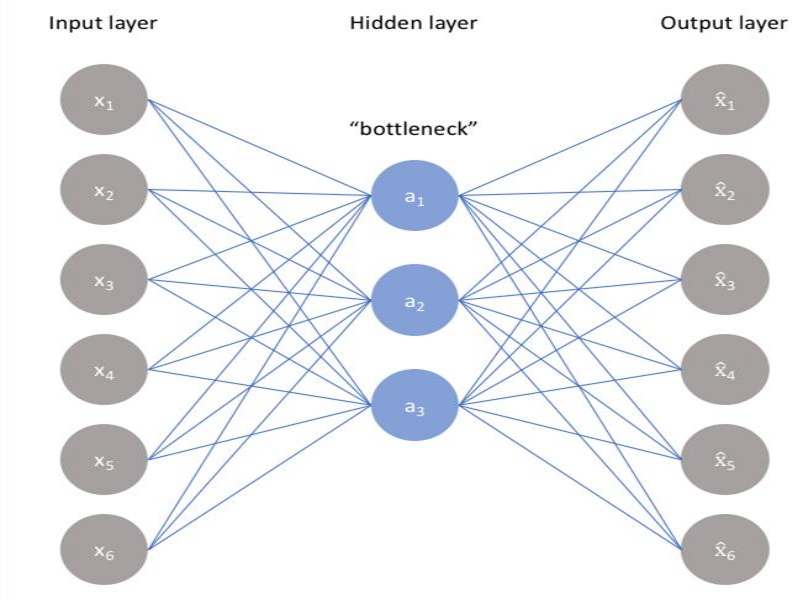
[40] Υλοποιείται συνήθως με τον αλγόριθμο ελαχιστοποίησης Kullback-Leibler divergence και στην πράξη υπολογίζεται μέσω μεγιστοποίησης της συνάρτησης ELBO.

[41] Όρος της Στατιστικής: Η κατανομή ή πιθανότητα που προέρχεται από την επεξεργασία μιας προηγούμενης υποθετικής κατανομής ύστερα από την παρουσία δεδομένων. Ονομάζεται επίσης και δεσμευμένη πιθανότητα.

[42] Υπάρχει ένα εγγενές trade-off κατά την εκπαίδευση με βάση αυτά τα δύο κριτήρια της συνάρτησης κόστους που δεν επιτρέπει την εκμάθηση ακριβώς τυποποιημένων κατανομών για τις κρυφές μεταβλητές.

[43] Η εκ των προτέρων πιθανότητα ή κατανομή που κάνουμε ως υπόθεση εργασίας χωρίς προηγούμενες ενδείξεις ή κάποια πληροφορία από παραδείγματα. Συνήθως προτιμώνται κατανομές που κάνουν τις ελάχιστες δυνατές παραδοχές για τα δεδομένα. Στα VAE είναι συνήθως η τυποποιημένη κανονική κατανομή. Οι δύο πιθανότητες συνδέονται με το Θεώρημα του Bayes.

χρησιμοποιήσουμε το δίκτυο του αποκωδικοποιητή για την συνεπή παραγωγή νέων ρεαλιστικών παραδειγμάτων. Κάτι τέτοιο δεν ισχύει για την περίπτωση του απλού αυτοκωδικοποιητή, όπου ο κρυφός χώρος είναι μη κανονικοποιημένος, υπάρχουν ασυνέχειες και περιοχές του χώρου όπου η είσοδος στον αποκωδικοποιητή τυχαίων παραδειγμάτων που ανήκουν εκεί, θα δώσει ως έξοδο «σκουπίδια» που δεν ανταποκρίνονται στα δεδομένα εισόδου. Επομένως οι Variational αυτοκωδικοποιητές εισάγουν αυτή την δυνατότητα παραγωγής νέων ρεαλιστικών δειγμάτων μέσω δειγματοληψίας της τυποποιημένης κανονικής κατανομής και αποτελούν χαρακτηριστικό παράδειγμα αναγεννητικού μοντέλου μαζί με τα GAN.



Εικόνα 31: Αυτόματοι Κωδικοποιητές

8.2 Τμήματα Κώδικα

8.2.1 Ορισμός Δικτύου GAN για CIC-IDS2017, 47 χαρακτηριστικά

```
class Generator(nn.Module):
    def __init__(self):
        super(Generator, self).__init__()
        self.model = nn.Sequential(
            nn.Linear(latent_dim + 1, 16),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(16, 24),
            nn.BatchNorm1d(24, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(24, 32),
            nn.BatchNorm1d(32, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(32, 40),
            nn.BatchNorm1d(40, 0.8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(40, data_dim),
            nn.Sigmoid()
        )

    def forward(self, z, label):
        label = torch.full((batch_size, 1), label)
        z = torch.cat((noise, label), 1)
        data = self.model(z)
        return data
```

```
class Discriminator(nn.Module):
    def __init__(self):
        super(Discriminator, self).__init__()
        self.model = nn.Sequential(
            nn.Linear(data_dim, 40),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(40, 32),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(32, 16),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(16, 8),
            nn.LeakyReLU(0.2, inplace=True),
            nn.Linear(8, 1),
```

```
            nn.Sigmoid()
        )

    def forward(self, data):
        validity = self.model(data)
        return validity
```

8.2.2 Top level Online ρηχών μοντέλων

```
    def on_init(self, packet, flow):
        flow.udps.model_prediction = 0
    def on_expire(self, flow):
        proc_flow = self.preprocess(flow)
        flow.udps.model_prediction = self.my_model.predict(proc_flow)
```

Για το «τρέξιμο» της εφαρμογής αρκεί ο παρακάτω κώδικας:

```
flows_predictions = []
n_normal = 0
n_attacks = 0
for flow in ml_streamer:
    if flow.requested_server_name in url_list:
        prediction = flow.udps.model_prediction
        flows_predictions.append(prediction)
        if prediction == 'BENIGN':
            n_normal += 1
        elif prediction == 'ATTACK':
            n_attacks += 1
```

8.2.3 Βρόχος εκπαίδευσης GAN

```
for epoch in range(num_epochs):
    for i, (data, labels) in enumerate(dataloader):
        batch_size = data.shape[0]
        real_data = Variable(data.type(FloatTensor))
        labels = Variable(labels.float())
        labels = labels.view(-1, 1).type(FloatTensor)

        # Train the discriminator
```

```
        optimizer_D.zero_grad()
        d_real_outputs = discriminator(real_data)
        d_real_loss = adversarial_loss(d_real_outputs, labels)

        fake_data = generator(noise, 0)
        d_fake_outputs = discriminator(fake_data.detach())
        d_fake_loss1 = adversarial_loss(d_fake_outputs, torch.full((batch_size, 1),
0.4).type(FloatTensor))

        fake_data = generator(noise, 1)
        d_fake_outputs = discriminator(fake_data.detach())
        d_fake_loss2 = adversarial_loss(d_fake_outputs, torch.full((batch_size, 1),
0.6).type(FloatTensor))

        d_loss = d_real_loss + d_fake_loss1 + d_fake_loss2
        d_loss.backward()
        optimizer_D.step()

        # Train the generator
        optimizer_G.zero_grad()
        noise = torch.randn(batch_size, latent_dim)
        fake_data = generator(noise, 0)
        d_fake_outputs = discriminator(fake_data)
        targets = torch.full((batch_size, 1), 0)
        g_loss1 = adversarial_loss(d_fake_outputs, targets.float())

        noise = torch.randn(batch_size, latent_dim)
        fake_data = generator(noise, 1)
        d_fake_outputs = discriminator(fake_data)
        targets = torch.full((batch_size, 1), 1)
        g_loss2 = adversarial_loss(d_fake_outputs, targets.float())

        g_loss = g_loss1 + g_loss2
        g_loss.backward()
        optimizer_G.step()
```

8.2.4 Έλεγχος GAN

```
for PATH_discr in discr_models_lst:
    discriminator.load_state_dict(torch.load(PATH_discr))
    discriminator.eval()
    true_positive = 0
```

```
    true_negative = 0
    false_positive = 0
    false_negative = 0
    for i in range(len(cicdataset_features)):
        row_label = cicdataset_labels[i]
        row_features = cicdataset_features[i]
        row_features = torch.tensor(row_features).float()
        model_output = discriminator(row_features)
        if model_output < 0.5:
            attack = True
        else:
            attack = False
        if (row_label == 'BENIGN' and attack == False):
            true_negative += 1
        if (row_label == 'ATTACK' and attack == True):
            true_positive += 1
        if (row_label == 'BENIGN' and attack == True):
            false_positive += 1
        if (row_label == 'ATTACK' and attack == False):
            false_negative += 1

    benign_recall = true_negative / (true_negative + false_positive)
    attack_recall = true_positive / (true_positive + false_negative)

    benign_precision = true_negative / (true_negative + false_negative)
    attack_precision = true_positive / (true_positive + false_positive)

    benign_f1 = 2 * (benign_recall * benign_precision) / (benign_recall + benign_precision)
    attack_f1 = 2 * (attack_recall * attack_precision) / (attack_recall + attack_precision)

    accuracy = (true_positive + true_negative) / (true_positive + true_negative +
false_positive + false_negative)
```

8.2.5 Top Level – NFStream, GAN

```
    def on_init(self, packet, flow):
        flow.udps.model_prediction = 0
        self.my_classifier = Discriminator()
        self.my_classifier.load_state_dict(torch.load(self.my_model))
        self.my_classifier.eval()
    def on_expire(self, flow):
```

```
        if flow.requested_server_name in url_list:
            proc_flow = self.preprocess(flow)
            proc_flow = proc_flow[0]
            proc_flow = torch.tensor(proc_flow).float()
            model_output = self.my_classifier(proc_flow)
            if model_output < 0.5:
                model_output = 'ATTACK'
            else:
                model_output = 'BENIGN'
            flow.udps.model_prediction = model_output
```