ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Χρήση Τεχνικών Επιβλεπόμενης Μηχανικής Μάθησης για Ανίχνευση Ανωμαλιών Κίνησης Δικτύου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αριστείδης Μ. Τζιαπούρας

Επιβλέπων: Συμεών Παπαβασιλείου

Καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2024

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ
ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Χρήση Τεχνικών Επιβλεπόμενης Μηχανικής Μάθησης για Ανίχνευση Ανωμαλιών Κίνησης Δικτύου

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Αριστείδης Μ. Τζιαπούρας

Επιβλέπων: Συμεών Παπαβασιλείου

Καθηγητής Ε.Μ.Π

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 12η Ιουλίου 2024

…….	…….	…….
Συμεών Παπαβασιλείου	Βασίλειος Καρυώτης	Ελένη Στάη
Καθηγητής Ε.Μ.Π	Αν.Καθηγητής Ιόνιο Πανεπιστήμιο, Τμήμα Πληροφορικής	Καθηγήτρια Ε.Μ.Π

Αθήνα, Ιούλιος 2024

.....

Αριστείδης Μ. Τζιαπούρας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Η Μηχανική Μάθηση αποτελεί ένα από τους πιο επίκαιρους κλάδους στον οποίο η επιστημονική κοινότητα δίνει ιδιαίτερη έμφαση τα τελευταία χρόνια. Ο λόγος που γίνεται είναι κυρίως εξαιτίας της προοπτικής που παρουσιάζει η εφαρμογή τεχνικών Μηχανικής Μάθησης για επίλυση προβλημάτων που ανήκουν σε διαφορετικούς ερευνητικούς τομείς. Ταυτόχρονα, η ραγδαία εξέλιξη στην ταχύτητα μετάδοσης των δεδομένων αύξησε σημαντικά τον όγκο πληροφορίας και τον αριθμό των πακέτων σε ένα δίκτυο. Ως αποτέλεσμα, αυξάνεται ο κίνδυνος σχετικά με την ασφάλεια του δικτύου καθώς οι παραδοσιακές τεχνικές ανίχνευσης ανωμαλιών και επιθέσεων δεν είναι σε θέση να ελέγχουν αποδοτικά όλο αυτό τον όγκο των δεδομένων.

Η παρούσα διπλωματική εργασία αποτελεί μια μελέτη στην οποία συνδυάζονται τεχνολογίες Μηχανικής Μάθησης μαζί με κίνηση πακέτων σε δίκτυο με στόχο την ανάπτυξη μοντέλων για την προστασία του δικτύου, τα οποία θα ανιχνεύουν αποδοτικά ύποπτα πακέτα που αποτελούν δείγμα επίθεσης στο δίκτυο. Μέσα από μια σειρά πειραματικών διαδικασιών, στόχος είναι ο εντοπισμός του ιδανικότερου μοντέλου αλλά και η διερεύνηση και ανάπτυξη μοντέλων μέσω διαφορετικών προσεγγίσεων έτσι ώστε να υπάρξει μια πληθώρα επιλογών ανάλογα με τις ανάγκες του συστήματος. Τα πειράματα έχουν υλοποιηθεί και βασιστεί πάνω σε τεχνικές Επιβλεπόμενης Μηχανικής Μάθησης.

Λέξεις Κλειδιά

Δίκτυα Υπολογιστών, Πακέτα Πληροφοριών, Ανίχνευση Ανωμαλιών, Μηχανική Μάθηση , Επιβλεπόμενη Μάθηση, Μοντέλα Επιβλεπόμενης Μάθησης, Προεπεξεργασία Δεδομένων, Ταξινομητής Ψηφοφορίας, Υπερπαραμέτροι , Μείωση Διαστάσεων Χαρακτηριστικών

Abstract

Machine Learning is one of the most discussed topics to which the scientific community has given special emphasis in recent years. The main reason is because of the potential presented by the application of Machine Learning techniques to solve problems belonging to different research fields. At the same time, the rapid development in data transmission speed has greatly increased the volume of information and the number of packets in a network. As a result, network security risk increases as traditional anomaly and attack detection techniques are unable to efficiently control all this volume of data.

This thesis is a piece of research that combines Machine Learning technologies along with network packet traffic with the aim of developing models for network protection, which will efficiently detect suspicious packets that are can be considered as a network attack. Through a series of experimental procedures, models have been investigated and developed through different approaches so that there is a variety of model options depending on the needs of the system but also the identification of the most ideal model. The experiments have been implemented and based on Supervised Machine Learning techniques.

Keywords

Computer Networks, Data Packets, Anomaly Detection, Machine Learning , Supervised Learning, Supervised Learning Models, Data Preprocessing, Voting Classifier, Hyperparameters, Feature Dimension Reduction

Ευχαριστίες

Αρχικά θα ήθελα να αφιερώσω αυτή τη διπλωματική εργασία στην μνήμη του θείου μου Κώστα που έφυγε από τη ζωή ξαφνικά λίγες μέρες πριν της παρουσίαση της. Επίσης θέλω να πω ένα μεγάλο ευχαριστώ στους γονείς μου και όλους τους υπόλοιπους συγγενείς που με στήριξαν όλα αυτά τα χρόνια.

Ακόμη θα ήθελα να ευχαριστήσω τον Αναπληρωτή Καθ. Βασίλη Καρυώτη και τον Δρ. Γρηγόρη Κακκαβά όπως και τον επιβλέποντα της διπλωματικής εργασίας Καθ. Συμεών Παπαβασιλείου για την ανάθεση της, και την ευκαιρία που μου έδωσαν να ασχοληθώ με ένα τόσο ενδιαφέρον θέμα.

Τέλος θα ήθελα να ευχαριστήσω όλους τους φίλους, συμφοιτητές και καθηγητές που γνώρισα και συμπορεύτηκα αυτά τα πέντε χρόνια, οι οποίοι έκαναν αυτή την πανέμορφη εμπειρία μοναδική και αξέχαστη.

Πίνακας Περιεχομένων

ΕΙΣΑΓΩΓΗ

Τα τελευταία χρόνια με την εξέλιξη της τεχνολογίας, παρατηρείται εκθετική αύξηση στο σύνολο και την ταχύτητα της πληροφορίας η οποία αυτή ανταλλάζεται μέσω πακέτων δεδομένων μέσα σε ένα δίκτυο. Η ξαφνική αυτή αλλαγή έχει προκαλέσει σημαντικά προβλήματα σε θέματα ασφάλειας καθώς οι παραδοσιακές τεχνικές για την ανίχνευση ανωμαλίας σε ένα δίκτυο δεν είναι αρκετά αποδοτικές έτσι ώστε να καλύπτουν τις ανάγκες της σύγχρονης πραγματικότητας.

Ταυτόχρονα ο κλάδος της Μηχανικής Μάθησης αποτελεί μια από τις πιο επίκαιρες τεχνολογίες στην επιστημονική κοινότητα, στον οποίο δίνεται ιδιαίτερη έμφαση λόγω των δυνατοτήτων που παρέχει σε όλους του τομείς. Η ανάγκη για νέες τεχνολογίες στην κυβερνοασφάλεια, σε συνδυασμό με την διαφορετική προσέγγιση στα προβλήματα που φέρνει η Μηχανική Μάθηση, οδήγησε στην ανάπτυξη πληθώρας αλγορίθμων και μοντέλων με κοινό στόχο τον γρήγορο, φτηνό και αξιόπιστο έλεγχο της πληροφορίας που μετακινείται σε ένα δίκτυο. Παρόλο που έχουν γίνει πολλές διαφορετικές μελέτες με των συνδυασμό των 2 τεχνολογιών, η έρευνα βρίσκεται ακόμη σε αρχικά στάδια και το περιθώριο βελτίωσης είναι τεράστιο.

Με αφορμή την τελευταία πρόταση, στην παρούσα διπλωματική θα εξεταστούν διάφορα μοντέλα επιβλεπόμενης μηχανικής μάθησης. Λαμβάνοντας υπόψη όσο το δυνατό περισσότερους παραμέτρους και υπερπαραμέτρους, θα πρέπει τελικά να παραχθούν νέα μοντέλα τα οποία θα είναι σε θέση να ανιχνεύουν αποδοτικά οποιαδήποτε πιθανή ανωμαλία θα υπάρχει στα πακέτα δεδομένων της κίνησης ενός δικτύου

ΣΚΟΠΟΣ

Η διπλωματική εργασία καλύπτει το πλαίσιο του συνδυασμού της ανίχνευσης ανωμαλιών σε ένα δίκτυο μαζί με τις τεχνολογίες της Μηχανικής Μάθησης. Γνωρίζοντας ότι το συγκεκριμένο πλαίσιο βρίσκεται ακόμη στα αρχικά στάδια, υπάρχει μια πληθώρα μελετών-υλοποιήσεων που χρειάζονται βελτιστοποίηση και νέων κατευθύνσεων που είναι αναγκαίο να διερευνηθούν.

Ο γενικός στόχος είναι η ανάπτυξη μοντέλων για την αποδοτική ανίχνευση ανωμαλιών σε ένα δίκτυο, κάτι το οποίο γίνεται με την αξιοποίηση διαφορετικών τεχνικών μηχανικής μάθησης. Πιο ειδικά, μέσα από μια σειρά τεσσάρων πειραματικών διεργασιών, πρωταρχικός στόχος είναι η βελτιστοποίηση της απόδοσης ήδη υλοποιημένων μοντέλων προηγούμενων μελετών μέσω της επιπλέον παραμετροποίηση τους. Ένα δεύτερος αλλά εξίσου σημαντικών στόχος είναι η διερεύνηση για την ανάπτυξη νέων μοντέλων μέσω διαφορετικών προσεγγίσεων και έλεγχος της απόδοσης τους.

Το σύνολο δεδομένων σε όλες τις πειραματικές διαδικασίες είναι κοινό και είναι το UNSW-NB15 Dataset. Το συγκεκριμένο σύνολο δεδομένων αποτελείται από ομαλή κίνηση δικτύου μαζί με συνθετικές επιθέσεις και έχει υπάρξει ως το βασικό Dataset με το οποίο εκπαιδεύονται τα μοντέλα μηχανικής μάθησης σε πάρα πολλές μελέτες.

ΔΟΜΗ ΕΡΓΑΣΙΑΣ

Η Παρούσα διπλωματική εργασία αποτελείται από τα 4 κεφάλαια που περιγράφονται συνοπτικά στη συνέχεια

Κεφάλαιο 1: Θεωρητικό Μέρος

Στο κεφάλαιο 1 παρουσιάζονται χρήσιμοι ορισμοί έτσι ώστε να κατανοηθεί περισσότερο το πρόβλημα της ανίχνευσης ανωμαλιών σε ένα δίκτυο, το θεωρητικό υπόβαθρο και η λογική των μεθόδων που θα αξιοποιηθούν στην πειραματική διεργασία.

Στο 1.1 γίνεται αναφορά στο τι είναι ένα δίκτυο, ποιοι τύποι υπάρχουν και πως μεταδίδονται τα πακέτα μέσω ενός δικτύου. Στο 1.2 παρουσιάζονται δύο κατηγορίες θεμάτων σε ένα δίκτυο που απασχολούν την επιστημονική κοινότητα, η τομογραφία δικτύου και η ανίχνευση ανωμαλιών στο δίκτυο. Στο 1.3 γίνεται εκτενής αναφορά σε 2 εργαλεία που μπορούν να αξιοποιηθούν για την δημιουργία αξιόπιστων αλγορίθμων για προβλήματα που αφορούν ένα δίκτυο. Οι 2 τεχνικές είναι η δειγματοληψία και η μηχανική μάθηση. Τα μοντέλα της μηχανικής μάθησης χωρίζονται σε κατηγορίες ανάλογα με τον τρόπο εκπαίδευσης τους και υπάρχουν στο 1.4. Η επιβλεπόμενη μηχανική μάθηση ανήκει σε μια από αυτές τις κατηγορίες και στο 1.5 παρουσιάζονται κάποια από τα πιο γνωστά μοντέλα της. Επίσης στο 1.6 γίνεται αναφορά του τι είναι οι υπερπαραμέτροι, πως επηρεάζουν και ποιοι υπερπαραμέτροι μπορεί να υπάρχουν σε ένα μοντέλο επιβλεπόμενης μάθησης, το Random Forest. Στο 1.7 παρουσιάζονται διάφορες τεχνικές που μπορούν να αξιοποιηθούν στη προεπεξεργασία των δεδομένων, δηλαδή τεχνικές των οποίων η χρήση διαμορφώνει το σύνολο δεδομένων κατάλληλα έτσι ώστε να είναι έτοιμο για να εκπαιδεύσει ένα μοντέλο. Στο 1.8, εμφανίζονται οι μετρικές αξιολόγησης με τις οποίες ελέγχεται η απόδοση των μοντέλων που έχουν εκπαιδευτεί και τέλος, στο 1.9 υπάρχει ο τρόπος λειτουργίας ενός του autoencoder, ένα είδος μοντέλου που ανήκει στην κατηγορία της μη επιβλεπόμενης μηχανικής μάθησης και εξηγείτε του πως ένα τέτοιο μοντέλο μπορεί να αξιοποιηθεί για την μείωση των διαστάσεων ενός συνόλου χαρακτηριστικών κατά την διάρκεια της προεπεξεργασίας των δεδομένων.

Κεφάλαιο 2: Προηγούμενες Μελέτες

Στο κεφάλαιο 2 γίνεται αναφορά για τις προηγούμενες μελέτες στις οποίες έχει βασιστεί η πειραματικές διεργασίες της παρούσας διπλωματική εργασίας.

Στο 2.1 παρουσιάζεται εκτενώς το σύνολο δεδομένων UNSW-NB15, το οποίο θα αποτελέσει το σύνολο δεδομένων που θα τύχει επεξεργασίας στο κεφάλαιο 3. Στο 2.2 παρουσιάζεται μια από τις προηγούμενες μελέτες που έχουν γίνει πάνω στο UNSW-NB15, της οποίας τα αποτελέσματα θα αποτελέσουν την βάση της παρούσας πειραματικής διάταξης με στόχο την εύρεση ακόμη καλύτερων αποτελεσμάτων.

Κεφάλαιο 3: Πειραματική Διαδικασία

Το κεφάλαιο 3 ασχολείται με την πειραματική διαδικασία που έχει ακολουθηθεί. Στο 3.1 περιγράφεται η διαδικασία της προεπεξεργασίας των δεδομένων, δηλαδή όλες οι τεχνικές που έχουν εφαρμοστεί έτσι ώστε να τροποποιηθεί σύνολο δεδομένων και να είναι σε μορφή έτοιμο

ως είσοδο για μοντέλα που πρέπει να εκπαιδευτούν. Η υλοποίηση του 3.1 εφαρμόζεται σαν αρχικό βήμα σε όλα τα πειράματα στο 3.2 – 3.5.

Στο 3.2 γίνεται δοκιμή 5 διαφορετικών μοντέλων επιβλεπόμενης μηχανικής μάθησης (SVC, Logistic Regression , K-NN , Decision Trees , Random Forest) και σύγκριση των αποδόσεων τους έτσι ώστε να εντοπιστεί το ιδανικό μοντέλο. Στο 3.3 εφαρμόζεται ένας Voting Classifier, γίνεται σύγκριση διάφορων τεχνικών ψηφίσματος – "Voting" όπως και επίσης του πως τα βάρη – "Weights" επηρεάζουν την απόδοση. Στο 3.4 εντοπίζονται από ένα δεδομένο σύνολο τιμών οι τιμές των υπερπαραμέτρων ενός Random Forest μοντέλου και ελέγχεται το πως η εισαγωγή των υπερπαραμέτρων επηρεάζει την απόδοση ενός ταξινομητή. Επίσης εφαρμόζονται διάφορες τιμές κατωφλίου και συγκρίνεται η απόδοση. Στην πειραματική διαδικασία του 3.5 γίνεται επαναλαμβανόμενος έλεγχος για το αν η αφαίρεση ενός χαρακτηριστικού επηρεάζει την απόδοση του μοντέλου η όχι. Κάθε φορά από το σύνολο χαρακτηριστικών που απέμεινε αφαιρείτε ένα χαρακτηριστικό και υπολογίζονται οι μετρικές. Αν υπάρχει χαρακτηριστικό το οποίο δεν κάνει την απόδοση του μοντέλου χειρότερη, τότε αυτό αφαιρείται και η διαδικασία επαναλαμβάνεται. Τέλος, στο 3.6 υλοποιείται ένας autoencoder (Unsupervised Learning) με στόχο την εκμετάλλευση του πρώτου μέρους του autoencoder, δηλαδή του encoder έτσι ώστε να συμπιεστεί το αρχικό σύνολο δεδομένων σε ένα νέο σύνολο δεδομένων με λιγότερα χαρακτηριστικά – διαστάσεις.

Κεφάλαιο 4 : Συμπεράσματα

Στο κεφάλαιο 4 συνοψίζονται τα συμπεράσματα που προέκυψαν από την πειραματικές διεργασίες όπως και επίσης κάποιες προτάσεις για μελλοντικές μελέτες και βελτιώσεις. Στο 4.1 παρουσιάζονται τα συμπεράσματα των 5 πειραμάτων, όπου γίνεται αναφορά στο ποια μοντέλα υπερτερούν ως προς την απόδοση και τον χρόνο εκτέλεσης. Επίσης παρουσιάζονται ποια πειράματα οδήγησαν σε βελτιωμένους ταξινομητές με μεγαλύτερη απόδοση και με ποιες προσθήκες παραμέτρων και υπερπαραμέτρων ή με ποιες αφαιρέσεις χαρακτηριστικών έχει επιτευχθεί αυτό. Σε περιπτώσεις που η εισαγωγή κάποιας νέας ιδέας δεν έχει φέρει βελτιωμένο αποτέλεσμα (π.χ Voting Classifier), και πάλι παρουσιάζεται ποιος συνδυασμός παραμέτρων ήταν ο καλύτερος και πιθανούς λόγους για τους οποίους η λύση αυτή δεν αύξησε την απόδοση του μοντέλου. Ακόμη παρουσιάζεται το πως η χρήση ενός autoencoder για τον περιορισμό των διαστάσεων των χαρακτηριστικών επηρεάζει την απόδοση του μοντέλου συγκριτικά με τις υπόλοιπες μετρήσεις. Στο 4.2 αναφέρονται διάφορες προτάσεις οι οποίες μπορούν να υλοποιηθούν στο μέλλον έχοντας ως βάση και μέτρο σύγκρισης την παρούσα μελέτη έτσι ώστε να γίνει έλεγχος αν προκύψουν ακόμη καλύτερα αποτελέσματα

Κεφάλαιο 1: Θεωρητικό Μέρος

1.1 Δίκτυα Υπολογιστών – Computer Networks

1.1.1 Τι είναι ένα Δίκτυο – Χρήσιμοι Ορισμοί

Ένα Δίκτυο Υπολογιστών – Computer Network [1],[2] ορίζεται ως ένα σύστημα το οποίο συνδέει δύο ή περισσότερες συσκευές έτσι ώστε να είναι δυνατή η επικοινωνία τους, η ανταλλαγή πληροφορίας και από κοινού χρήση διάφορων πόρων (π.χ. εκτυπωτές). Κάθε **δίκτυο** αναπαρίσταται ως ένας γράφος με κόμβους και ακμές. Οι συσκευές του δικτύου αποτελούν τους **κόμβους** του γράφου και μπορεί να είναι υπολογιστές, Servers, Δρομολογητές – Routers , Switches κλπ. Οι **ακμές-κανάλια** εκφράζουν τον τρόπο επικοινωνίας των συσκευών μεταξύ τους η οποία γίνεται είτε ενσύρματα με καλώδια όπως οπτικές ίνες είτε ασύρματα (Wireless Networks). Η φυσική και λογική διάταξη των κόμβων σε ένα δίκτυο εκφράζει την **τοπολογία** του δικτύου, με τις πιο κοινές διατάξεις να είναι οι bus, star,ring, mesh και tree [3].

Η επικοινωνία μεταξύ δύο συσκευών δεν γίνεται αυθαίρετα αλλά πάντα ακολουθείται το **πρωτόκολλο επικοινωνίας.** Τα πρωτόκολλα είναι μια σειρά από προκαθορισμένους κανόνες και αλγόριθμους που πρέπει να ακολουθηθούν αυστηρά και δηλώνουν τον ακριβή τρόπο επικοινωνίας μεταξύ των συσκευών στο δίκτυο (π.χ. TCP, IP, UDP, ARP κλπ) [4]. Επιπλέον κάθε συσκευή στο δίκτυο έχει ένα μοναδικό αριθμητικό αναγνωριστικό, γνωστό ως **IP Address**[5]. Αυτό το αναγνωριστικό βοηθά στον εντοπισμό των συσκευής και κάνει δυνατή την μεταξύ τους επικοινωνία. Τέλος, υπάρχει το **Τείχος Προστασίας – Firewall,** δηλαδή μια συσκευή ασφαλείας που ελέγχει την εισερχόμενη και εξερχόμενη κίνηση με κύριο στόχο την προστασία του δικτύου, προσπαθώντας να αποτρέψει την μη εξουσιοδοτημένη πρόσβαση στο δίκτυο μαζί με ότι άλλες απειλές στην ασφάλεια μπορεί να υπάρξουν.

1.1.2 Τύποι Δικτύων

Τα δίκτυα υπολογιστών μπορούν να ταξινομηθούν με βάση διάφορα κριτήρια, όπως το μέσο μετάδοσης, το μέγεθος, την τοπολογία αλλά και την «γεωγραφική έκσταση» στην οποία ανήκουν. Οι 2 πιο κοινοί τύποι δικτύων είναι τα LAN – Local Area Network και τα WAN – Wide Area Network. Ενδεικτικά, κάποιοι διαφορετικοί τύποι δικτύων είναι τα MAN – Metropolitan Area Networks , τα WLAN – Wireless LAN και WWAN – Wireless WAN [6]

Local Area Network – LAN

Ένα LAN είναι ένα δίκτυο που καλύπτει τις ανάγκες επικοινωνίας των συσκευών σε μια μικρή γεωγραφική έκταση όπως για παράδειγμα ένα σχολείο, ένα γραφείο, ένα κτήριο ή ένα σπίτι

Wide Area Network - WAN

Αντίθετα με το LAN , το WAN είναι ένα δίκτυο που καλύπτει μια μεγάλη γεωγραφική έκταση όπως μια πόλη, μια χώρα ή και ακόμη όλο τον κόσμο. Τα WAN χρησιμοποιούνται για επικοινωνίες συσκευών που απέχουν μεγάλη απόσταση και επίσης είναι το μέσο με το οποίο τα πολλά μικρά διαφορετικά LAN μπορούν να ανταλλάξουν πληροφορίες μεταξύ τους.

1.1.3 Κίνηση στο Δίκτυο – Network Traffic

Με τον όρο **Network Traffic [7]** ορίζεται ως το σύνολο των δεδομένων που μετακινούνται από μια συσκευή σε μια άλλη ή άλλες συσκευές του δικτύου για μια δεδομένη χρονική στιγμή. Επειδή το σύνολο δεδομένων που πρέπει να αποσταλεί από ένα κόμβο-πομπό σε κάποιο κόμβο-δέκτη είναι αρκετά μεγάλο, για την αποδοτική μεταφορά της πληροφορίας, αντί για ένα μεγάλο ενιαίο πακέτο δημιουργούνται πολλά μικρότερα **Πακέτα Δεδομένων - Data Packets.** Στον αποστολέα η πληροφορία σπάει σε μικρότερα πακέτα τα οποία αποστέλλονται όλα στον ίδιο παραλήπτη. Η συσκευή του παραλήπτη είναι υπεύθυνη έτσι ώστε όταν φτάσουν όλα τα πακέτα, να συναρμολογήσει το μήνυμα ξανά, οργανώνοντας όλα τα πακέτα σε μια λογική σειρά έτσι ώστε να μην υπάρξει αλλοίωση στο τελικό αποτέλεσμα. Σημαντική είναι η αναφορά του ότι παρόλο που τα πακέτα έχουν ίδιο αποστολέα και παραλήπτη και στέλνονται ταυτόχρονα, υπάρχουν περιπτώσεις όπου στο δίκτυο μεταφέρονται από διαφορετικά κανάλια για σκοπούς ταχύτητας και έτσι ώστε να υπάρχει ομοιόμορφη κατανομή της κίνησης σε όλο το δίκτυο.

Στις περιπτώσεις που σε ένα δίκτυο υπάρχει υπερβολική κίνηση πακέτων, τότε αυτόματα σημαίνει πως τα πακέτα φτάνουν με καθυστέρηση στον παραλήπτη κάτι που μειώνει την ταχύτητα επικοινωνίας και την απόδοση του δικτύου. Τις πλείστες φορές το στο δίκτυο υπάρχει μόνο ομαλή κίνηση, δηλαδή κίνηση πακέτων και πληροφοριών στο δίκτυο χωρίς κακόβουλες προθέσεις. Από την άλλη, μέσα στο πλήθος των πακέτων ομαλής κίνησης, είναι πολύ πιθανό να υπάρξουν κακόβουλα πακέτα - ιοί, με στόχο την πρόσβαση εξωτερικών μη εξουσιοδοτημένων χρηστών στο δίκτυο για την απόσπαση σημαντικών πληροφοριών. Επομένως όσο αφορά την ασφάλεια του δικτύου, μια ασυνήθιστα ξαφνική ψηλή κυκλοφορία πακέτων προς ένα κόμβο ή προς όλο το δίκτυο ίσως να αποτελεί δείγμα μιας επίθεσης από εξωτερικούς παράγοντες και να χρειάζεται εξειδικευμένη έρευνα.

1.2 Τομογραφία Δικτύου και Ανίχνευση Ανωμαλιών Κίνησης Δικτύου

1.2.1 Τομογραφία Δικτύου - Network Tomography

Μπορούμε να φανταστούμε ένα οποιοδήποτε δίκτυο ως ένα γράφο με n κόμβους και m ακμές. Οι κόμβοι αποτελούν ένα υπολογιστή, ένα Router ή ακόμα και ένα ολόκληρο

υποδίκτυο ενώ οι ακμές δηλώνουν την άμεση σύνδεση μεταξύ δύο κόμβων μέσω ενός μονοπατιού (path). Επίσης κάθε μονοπάτι αποτελείται από ένα ή περισσότερα κανάλια μεταφοράς (links) τα οποία μπορεί να είναι είτε μιας κατεύθυνσης είτε αμφίδρομα. Τέλος, μέσα σε αυτό το δίκτυο όταν ο κόμβος αποστολέας στείλει μήνυμα σε ένα κόμβο δέκτη, το μήνυμα μεταφέρεται μέσα από μια σειρά πακέτων από bits μέσω τον καναλιών και ενδιάμεσα από διάφορους άλλους κόμβους.

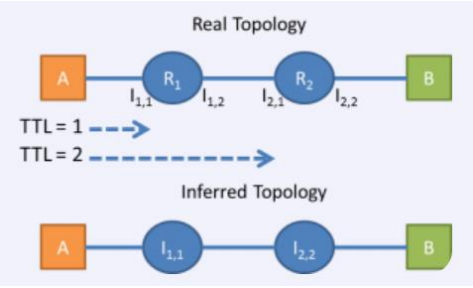
Σε μεγάλα δίκτυα το κόστος για τον ακριβή υπολογισμό διάφορων μετρήσεων σχετικά με την κίνηση στο δίκτυο είναι πολύ μεγάλο. Οπότε κρίνεται αναγκαία η εύρεση αποδοτικών μεθόδων έτσι ώστε να εκτιμώνται όσο το δυνατόν καλύτερα συμπεράσματα σχετικά με το εσωτερικό, την άγνωστη τοπολογία και την απόδοση ενός δικτύου παίρνοντας ελάχιστες μετρήσεις σε όσο το δυνατό λιγότερους κόμβους. Το πρόβλημα αυτό πήρε τον όρο Τομογραφία Δικτύου – Network Tomography [8].

USE CASES

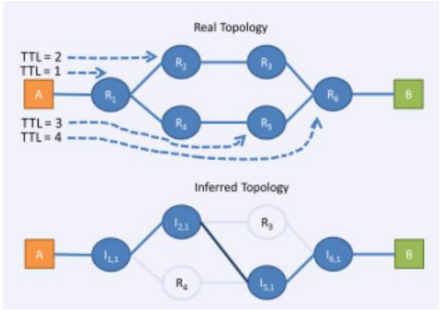
Κάποιες περιπτώσεις χρήσης (Use Cases) οι οποίες αναπαριστούν τις βασικές ιδέες της τομογραφίας δικτύου είναι οι εξής [9]:

1) Συμπέρασμα για την Λογική Τοπολογία του Δικτύου με χρήση Traceroute

Το traceroute [10] είναι ένα εργαλείο το οποίο δημιουργεί ένα χάρτη με το από που περνάνε και πως τα δεδομένα ταξιδεύουν από τον αποστολέα(source) στον προορισμό(destination). Ξεκινώντας από ένα κόμβο A και στέλνοντας Traceroute εντολές προς ένα B σχηματίζεται μια εικόνα σχετικά με την τοπολογία του δικτύου η οποία δεν ήταν γνωστή έως τώρα. Σε απλά δίκτυα είναι δυνατό ο εντοπισμός της ακριβής τοπολογίας όπως φαίνεται στο σχήμα 1.2.1.1 Αντίθετα σε πιο σύνθετα δίκτυα, η τοπολογία που εξάγετε μπορεί να μοιάζει αλλά δεν αναπαριστά ακριβώς το πραγματικό δίκτυο, όπως στην περίπτωση του σχήματος 1.2.1.2 .



Σχήμα 1.2.1.1: Traceroute σε απλό Δίκτυο

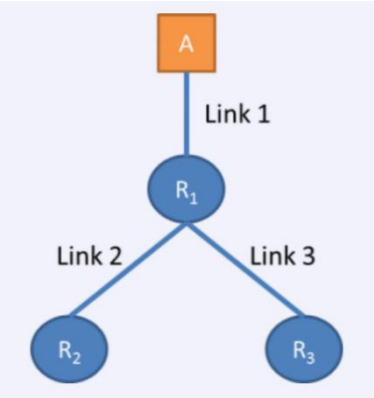


Σχήμα 1.2.1.2: Traceroute σε σύνθετο δίκτυο

2) Υπολογισμός Ρυθμού Απώλειας – Loss Rate

Στο συγκεκριμένο Use Case η τοπολογία του δικτύου θεωρείται γνωστή και εμφανίζεται η ανάγκη του υπολογισμού της πιθανότητας για απώλεια πακέτου σε κάθε δρομολογητή. Στην περίπτωση του σχήματος 1.2.1.3 για παράδειγμα, στέλνεται ένα πακέτο πολλαπλής

διανομής από το A προς τα R2 και R3. Ανάλογα με το σε ποιους από τους δρομολογητές R2,R3 έφτασε το πακέτο, είναι δυνατό η εξαγωγή συμπεράσματος ή η εκτίμηση του σε ποιο κανάλι μεταφοράς υπήρξε η απώλεια κάτι που παρουσιάζεται στον πίνακα 1.2.1.4.



Σχήμα 1.2.1.3: Πιθανή τοπολογία ενός Δικτύου

Πίνακας 1.2.1.4: Πιθανά σενάρια από αποστολή πακέτου

Άφιξη σε R2	Άφιξη σε R3	Συμπέρασμα
ΝΑΙ	ΝΑΙ	Καμία Απώλεια
ΝΑΙ	ΟΧΙ	Απώλεια σε Link 3
ΟΧΙ	ΝΑΙ	Απώλεια σε Link 2
ΟΧΙ	ΟΧΙ	Απώλεια σε Link 1 ή Απώλεια σε Link 2 και Link 3

Με επαναλαμβανόμενη αποστολή πακέτων από το A προς R2,R3 είναι δυνατή η εκτίμηση του ρυθμού απώλειας κάθε καναλιού. Το αποτέλεσμα θα αποτελεί εκτίμηση και όχι ακριβή μέτρηση καθώς στην περίπτωση 4, υπάρχουν 2 πιθανά σενάρια.

3) Κατανομή Καθυστέρησης σε κάθε Κανάλι – Delay Distributions

Ένα τρίτο Use Case είναι η ανάγκη υπολογισμού της καθυστέρησης που προκύπτει σε κάθε κανάλι(link). Έστω ότι το δίκτυο έχει την τοπολογία του σχήματος 1.2.1.3 και πως η ελάχιστη καθυστέρηση διάδοσης (Minimum Propagation Delay) σε κάθε κανάλι είναι γνωστή. Μια λύση να εντοπιστεί μια εκτίμηση για τη καθυστέρηση στα κανάλια 2 και 3 είναι να στέλνεται επαναλαμβανόμενα ένα multicast πακέτο από το A προς τα R2,R3 και να καταγράφεται ο χρόνος καθυστέρησης μέχρι να φτάσει στους 2 κόμβους. Το κανάλι 1 είναι κοινό και στις 2 περιπτώσεις και οποιαδήποτε χρονική διαφορά στην άφιξη οφείλεται στο χρόνο καθυστέρησης των καναλιών 2 και 3. Σε όσες περιπτώσεις ο χρόνος καθυστέρησης για να φτάσει το πακέτο στο R2 ήταν ο ελάχιστος δυνατός, αυτόματα σημαίνει πως και ο χρόνος καθυστέρησης για να φτάσει στο R1 ήταν ο ελάχιστος. Οπότε εφόσον είναι γνωστή η καθυστέρηση για το R3 αλλά και ο χρόνος καθυστέρησης για να φτάσει στο R1, με αφαίρεση των 2 μπορεί να γίνει εκτίμηση του χρόνου καθυστέρησης στο κανάλι 3. Η ίδια διαδικασία μπορεί να γίνει και αντίστροφα για το κανάλι 2 στις περιπτώσεις που υπάρχει η ελάχιστη δυνατή καθυστέρηση από το A στο R3.

Προηγούμενες Μελέτες

Η τομογραφία δικτύου είναι ένας τομέας στο οποίο έγιναν και γίνονται αυτή τη στιγμή πολλές μελέτες λόγω της πληθώρας προσεγγίσεων και των περιθωρίων βελτίωσης που υπάρχουν. Τα προβλήματα που αφορούν σε τομογραφία δικτύου μπορούν με μια καλή προσέγγιση να θεωρηθούν σαν ένα σύστημα με γραμμικές εξισώσεις: $\boldsymbol{Yt = A * Xt + \varepsilon}$, όπου ε το σφάλμα, Yt το διάνυσμα των γνωστών μετρήσεων, A ο πίνακας δρομολόγησης ο οποίος πολλές φορές είναι δυαδικός (1 για μονοπάτι, 0 για απουσία μονοπατιού) και Xt το διάνυσμα των άγνωστων παραμέτρων. Διάφοροι αλγόριθμοι για την βελτιστοποίηση του προβλήματος αναπτύσσονται συνεχώς όπως για παράδειγμα οι τεχνικές που προτείναν οι Rui Castro, Mark Coates, Gang Liang, Robert Nowak και Bin Yu για την αναγνώριση της άγνωστης τοπολογίας αλλά και την πιθανή κατανομή των αγνώστων παραμέτρων [11].

Η εξέλιξη της μηχανικής μάθησης έφερε ήδη νέα μοντέλα που αξιοποιούν τον τομέα αυτό για την βέλτιστη προσέγγιση του συγκεκριμένου προβλήματος. Μια ιδέα είναι η χρήση Deep Generative τεχνικών ,κάτι το οποίο έγινε πειραματικά από τους κ. GRIGORIOS KAKKAVAS,VASILEIOS KARYOTIS, NIKOLAOS FRYGANIOTIS, και SYMEON PAPAVASSILIOU [12]. Στο πείραμα εκπαιδεύτηκαν και πάρθηκαν μετρήσεις ως προς την εκτίμηση της κίνησης (Traffic Matrix Estimation) από διάφορα μοντέλα-παραλλαγές του VAE, αξιολογώντας τες με τις μετρικές RMSE,NMAE,SRE,TRE (Κεφάλαιο 1.8.2). Τα αποτελέσματα έδειξαν ότι τα συγκεκριμένα μοντέλα δίνουν καλές εκτιμήσεις και ανέδειξαν την προοπτική που έχει η χρήση της μηχανικής μάθησης στον χώρο αυτό.

1.2.2 Ανίχνευση Ανωμαλιών – Anomaly Detection

Ο Όρος ανίχνευση ανωμαλιών [13] αναφέρετε στην διαδικασία της αναγνώρισης παρατηρήσεων, γεγονότων ή σημείων δεδομένων τα οποία διαφέρουν από αυτά που χαρακτηρίζονται ως φυσιολογικά και αναμενόμενα , κάτι το οποίο τα καθιστά ως ανωμαλίες. Στόχος είναι η εύρεση αποδοτικής λύσης για τον εντοπισμό και επισήμανση της ύποπτης μη αναμενόμενης κίνησης η οποία είναι απαραίτητο να ερευνηθεί περαιτέρω. Τα δεδομένα που δεν συμβαδίζουν με την αναμενόμενη συμπεριφορά πολλές φορές σηματοδοτούν σοβαρά περιστατικά τα οποία δεν είχαν εντοπιστεί έως τώρα καθώς ο έλεγχος του από που και πως προήλθε αυτή η ανωμαλία είναι πιθανό να οδηγήσει στην ανακάλυψη κάποιου σφάλματος ή πιθανού λάθους στην λειτουργία ενός συστήματος ή απειλές όσο αφορά την ασφάλεια του δικτύου.

Ενδεικτικά, κάποιοι τομείς στους οποίους χρειάζεται η ανίχνευση των ανωμαλιών είναι η στην οικονομία για τον εντοπισμό ύποπτων συναλλαγών, στην υγεία για την ανίχνευση ασυνήθιστων περιστατικών σε ασθενείς, στην βιομηχανία για τον εντοπισμό ελαττωμάτων ή δυσλειτουργιών του εξοπλισμού αλλά και στον τομέα της κυβερνοασφάλειας, καθώς μια ασυνήθιστη συμπεριφορά σε ένα δίκτυο μπορεί να αποτελεί στοιχείο που θα φανερώσει μια επίθεση κυβερνοασφάλειας.

Έχουν αναπτυχθεί πάρα πολλές διαφορετικές τεχνικές και αλγόριθμοι ανάλογα με την περίσταση που χρειάζεται. Η πρώτη μέθοδος είναι με την **οπτικοποίηση** των δεδομένων, στην οποία αναλυτές κατασκευάζοντας διαγράμματα και γραφήματα οπτικοποιούν το σύνολο των δεδομένο προσπαθώντας να παρατηρήσουν μοτίβα και δεδομένα τα οποία δεν έχουν ομοιότητες με τα υπόλοιπα. Η δεύτερη μέθοδος είναι με **στατιστικούς ελέγχους,** δηλαδή μέσω της σύγκρισης των αποτελεσμάτων που καταγράφηκαν σε σχέση με την αναμενόμενη κατανομή ή μοτίβο**.** Σε περίπτωση που η γενικότερη εικόνα των δεδομένων μια χρονική στιγμή δεν συμβαδίζουν με την κατανομή που καταγραφόταν έως τώρα (μέση τιμή και διασπορά) τότε απαιτείται πιο εξειδικευμένος έλεγχος σχετικά με τον λόγο που γίνεται αυτό.

Τέλος, υπάρχει η ραγδαία αναπτυσσόμενη μέθοδος της **μηχανικής μάθησης,** στην οποία εκπαιδεύονται μοντέλα μηχανικής μάθησης με ήδη καταγεγραμμένα δεδομένα, έτσι ώστε να εντοπιστούν μοτίβα και αποκλίσεις για την αξιοποίηση τους στην πρόβλεψη μελλοντικής εισόδου και πιθανής ανωμαλίας σε κάποιο δίκτυο. Οι διάφοροι αλγόριθμοι μηχανικής μάθησης για την ανίχνευση ανωμαλιών χωρίζονται σε κατηγορίες ανάλογα με το αν γνωρίζουμε το τι είναι τα ήδη καταγεγραμμένα δεδομένα ή όχι. Αν είναι γνωστό ποια δεδομένα είναι από ομαλή κίνηση και ποια από ανωμαλίες στο δίκτυο , τότε μπορούν να αξιοποιηθούν μοντέλα **επιβλεπόμενης μάθησης – supervised learning**. Αντίθετα, υπάρχουν αλγόριθμοι **μη-επιβλεπόμενης μηχανικής – unsupervised learning** τα οποία επιτρέπουν την αποδοτική εκπαίδευση μοντέλων όταν δεν γνωρίζουμε το που «ανήκουν τα δεδομένα εισόδου» (ομαλή κίνηση ή όχι)

1.3 Τεχνικές Ελέγχου Ποιότητας Δεδομένων

1.3.1 Δειγματοληψία – Sampling

Δεν είναι κρυφό πως με την πάροδο του χρόνου η ποσότητα της κίνησης και η ανταλλαγή πληροφοριών και δεδομένων μέσα σε ένα δίκτυο αυξάνεται με εκθετικό ρυθμό. Παρόλο που ο στόχος της βελτίωσης των υποδομών κα ταχύτητας του δικτύου επιτυγχάνεται, από την άλλη εμφανίζονται νέα είδη προβλημάτων σε θέματα που αφορούν την ασφάλεια των δικτύων. Ένα από τα κύρια προβλήματα της εκθετικής αύξησης της κίνησης μάσα σε ένα δίκτυο είναι η αδυναμία του αξιόπιστου ελέγχου της ποιότητας των δεδομένων. Ο τεράστιος αυτός όγκος είναι αδύνατο να περάσει με ικανοποιητικό ρυθμό από ένα προς ένα έλεγχο, κάτι που κάνει αρκετές από τις παραδοσιακές τεχνικές μη αποδοτικές. Επομένως, λόγω της ταυτόχρονης ανάγκης για ταχύτητας αλλά και αξιόπιστου ελέγχου των δεδομένων που μετακινούνται σε ένα δίκτυο, εμφανίζεται ολοένα και περισσότερο η ανάγκη της ανάπτυξης αλγορίθμων βασισμένους στην διαδικασία της δειγματοληψίας – Sampling [14].

Ο όρος Sampling είναι κάτι το γενικό και με αυτό δηλώνεται η διαδικασία με την οποία αντί να ληφθεί υπόψη το σύνολο, μέσα από μια σειρά κριτηρίων, επιλέγεται ένα μικρό μέρος του συνόλου στο οποίο γίνεται ο έλεγχος που προοριζόταν για ολόκληρο το σύνολο. Το πιο

βασικό θέμα στην διαδικασία της δειγματοληψίας είναι η επιλογή των κατάλληλων κριτηρίων για το δειγματοληπτικό σύνολο. Αν η επιλογή των κριτηρίων είναι σωστή, τότε το μικρότερο «δείγμα» του συνόλου θα έχει χαρακτηριστικά ανάλογα και συγκρίσιμα με το ολόκληρο. Οπότε με πολύ λιγότερους ελέγχους που εφαρμόζονται στο μικρότερο σύνολο, είναι κατορθωτό ο σχηματισμός μιας ευρύτερης εικόνας στο ολικό σύνολο με ελάχιστες αποκλίσεις. Έτσι έγινε η υλοποίηση ενός αλγορίθμου που ελαττώνει σημαντικά το κόστος, το χρόνο και την πολυπλοκότητα της διαδικασίας ελέγχου.

Η δειγματοληψία εδώ και πολλά χρόνια χρησιμοποιείται ήδη σε πολλές περιπτώσεις στην καθημερινότητα. Ένα από τα πιο χαρακτηριστικά παραδείγματα είναι οι εκτιμήσεις που παρουσιάζουν οι αναλυτές το διάστημα πριν από μια εκλογική αναμέτρηση, γνωστά ως Exit Polls[15]. Καθώς είναι αδύνατο να ερωτηθούν όλοι που ψηφίζουν έτσι ώστε να υπάρχει ακριβείς αποτέλεσμα, γίνεται τις προηγούμενες μέρες μια κατάλληλη επιλογή του δείγματος από τον συνολικό πληθυσμό στους οποίους γίνονται ερωτήσεις σχετικά με το τι θα ψηφίσουν. Πρέπει να δοθεί αρκετή προσοχή στο να ερωτηθούν άνθρωποι διαφόρων ηλικιών, φύλου, κοινωνικής τάξης και περιοχή διαμονής σε αναλογία που θα αντικατοπτρίζει το γενικότερο σύνολο. Με αυτό τον τρόπο, μέσα από ερωτήσεις ενός μικρού δείγματος του συνόλου και κλιμακώνοντας τα αποτελέσματα μπορούν να εξαχθούν στατιστικά για τα πιθανά ποσοστά των υποψηφίων, λαμβάνοντας πάντα υπόψη την απόκλιση λόγω στατιστικού λάθους.

Για τους λόγους τους οποίους έγινε και η αναφορά πιο πάνω, κρίνεται αναγκαίο η εφαρμογή παρόμοιας λογικής για τον έλεγχο του πλήθους της κίνησης σε δίκτυα υπολογιστών. Ο όγκος της πληροφορίας που ανταλλάζεται είναι υπερβολικά μεγάλος, οπότε πρέπει μέσω στοχευμένων ελέγχων συγκεκριμένων πακέτων - δειγμάτων να υπάρξει όσο το δυνατό πιο αξιόπιστος έλεγχος για ολόκληρο το σύνολο των πακέτων. Προφανώς, αν ελέγχονται όλα τα πακέτα η απόδοση θα είναι μέγιστη αλλά ο χρόνος επεξεργασίας θα είναι πολύ κακός και αν ελέγχονται πολύ πιο λίγα ή όχι τα κατάλληλα πακέτα, ο χρόνος επεξεργασίας θα είναι ικανοποιητικός, όμως η απόδοση της υλοποίησης θα είναι πολύ χειρότερη από αυτό που πρέπει. Επομένως, το πιο σημαντικό πρόβλημα σε ένα αλγόριθμο δειγματοληπτικό στη περίπτωση της κίνησης σε ένα δίκτυο, είναι η επιλογή των κατάλληλων πακέτων, τέτοια ώστε και η απόδοση να είναι ικανοποιητική αλλά και ο χρόνος επεξεργασίας να είναι ο ελάχιστος δυνατός.

Οι τεχνικές για την δειγματοληψία μπορούν να κατηγοριοποιηθούν σε 2 μεγάλες υποκατηγορίες. Η πρώτη είναι οι μέθοδοι οι οποίες βασίζονται στην ποσότητα των δεδομένων – count based και η δεύτερη είναι οι μέθοδοι οι οποίες βασίζονται σε τεχνικές βασισμένες στον χρόνο – time based.

Τεχνικές Βασισμένες στην ποσότητα - Count Based Methods

Κάποιες βασικές τεχνικές βασισμένες στην ποσότητα είναι οι εξής [16]:

Τυχαία Επιλογή Δειγμάτων – Simple Random Sampling

Η τεχνική αυτή ίσως να είναι η πιο απλή στην εκτέλεση. Από τον σύνολο πακέτων που υπάρχει, επιλέγεται με τυχαίο τρόπο ένα δείγμα πακέτων ανάλογα με την επιθυμία του χρήστη (π.χ. το 10%). Δεν υπάρχει κάποιος περιορισμός για την επιλογή των δειγμάτων, παρά μόνο το ότι η επιλογή γίνεται τυχαία.

Συστηματική Δειγματοληψία – Systematic Sampling

Η λογική είναι εύκολα υλοποιήσιμη γι' αυτό η συστηματική δειγματοληψία εφαρμόζεται σε αρκετούς αλγορίθμους. Κάθε πακέτο του αρχικού συνόλου παίρνει έναν μοναδικό αύξοντα αριθμό και στην συνέχεια ανάλογα με το ποσοστό δειγματοληψίας που θέλει ο χρήστης, η επιλογή γίνεται μέσω προκαθορισμένων διαστημάτων. Αν σε ένα σύνολο 100 μελών χρειάζεται δειγματοληψία 10%, τότε επιλέγονται ως δείγματα τα πακέτα με αριθμό 1,11,21,31…,81,91.

Στρωματοποιημένη δειγματοληψία – Stratified Sampling

Η συγκεκριμένη τεχνική υλοποιείται αρχικά με τον διαχωρισμό των πακέτων του συνόλου σε N διαφορετικά υποσύνολα, με βάση κάποιο κοινό ή κάποια κοινά χαρακτηριστικά τα οποία έχουν. Στη συνέχεια συγκρίνοντας τις αναλογίες του πλήθους σε κάθε υποσύνολο και τον συνολικό αριθμό των δειγμάτων που απαιτούνται, για κάθε υποσύνολο υπολογίζεται ένας αριθμός n, το πλήθος δηλαδή των πακέτων που χρειάζεται να δειγματοληφθούν από το υποσύνολο αυτό για να διατηρηθεί η αναλογία. Σε κάθε υποσύνολο η επιλογή των n πακέτων μπορεί να επιτευχθεί με τη χρήση μιας δεύτερης τεχνικής, όπως για παράδειγμα με συστηματική δειγματοληψία. Το συγκεκριμένο είδος δειγματοληψίας είναι γνωστό και ως Random n-out N Sampling

Ομοιόμορφη Πιθανολογική Δειγματοληψία – Uniform Probabilistic Sampling

Η Δειγματοληψία με αυτή την μορφή είναι παρόμοια με την τεχνική Simple Random Sampling. Η κεντρική ιδέα είναι πως σε κάθε πακέτο από το αρχικό σύνολο δίνεται μια πιθανότητα p, η οποία συμβολίζει το πόσο πιθανό είναι ένα πακέτο να ενταχθεί στο σύνολο των δειγμάτων. Σε μεγάλη κλίμακα το ποσοστό του τελικού αριθμού των δειγμάτων θα τείνει προς την πιθανότητα p που έχει το κάθε πακέτο ξεχωριστά. Το συγκεκριμένο είδος δειγματοληψίας είναι γνωστό και ως Geometric Random Sampling

Τεχνικές Βασισμένες στον χρόνο - Time Based Methods

Οι τεχνικές βασισμένες στον χρόνο όπως έχει αποδειχθεί μέσα από πειραματικές διεργασίες (π.χ. [17])δεν είναι οι ιδανικές για τον τομέα της ανίχνευσης ανωμαλιών στα δίκτυα καθώς υπάρχει η τάση να χάνονται πολλά από τα μικρά χρονικά διαστήματα στα οποία υπάρχει έντονη ύποπτη μεγάλη κίνηση πακέτων όπου σε κανονικές συνθήκες πρέπει να ελεγχθεί η αξιοπιστία τους. Ενδεικτικά, μια Timed Based μέθοδος είναι η

συστηματική δειγματοληψία χρόνου – Systematic Time Sampling [18] , στην οποία τα πακέτα επιλέγονται μέσα από ένα συγκεκριμένο χρονικό διάστημα.

Μελέτη σε Count Based Methods

Στο Εθνικό Μετσόβιο Πολυτεχνείο έγινε μια μελέτη από τους Georgios Androulidakis, Vasilis Chatzigiannakis, Symeon Papavassiliou', Mary Grammatikou and Vasilis Maglaris [19] όπου χρησιμοποίησαν δειγματοληψία με Count Based μεθόδους πάνω σε τεχνικές ανίχνευσης ανωμαλιών με στόχο τον έλεγχο της απόδοσης.

Οι count based μέθοδοι που έχουν χρησιμοποιηθεί είναι η συστηματική δειγματοληψία – Systematic Sampling , όπου επιλέγεται κάθε k-οστό πακέτο για έλεγχο , το Random n-out-N sampling , όπου τα πακέτα χωρίζονται ανά N-άδες και επιλέγονται n τυχαία κάθε φορά (n<N) και τέλος το Uniform Probabilistic Sampling στην οποία κάθε πακέτο έχει μια πιθανότητα p (p<1) για την οποία θα υποστεί έλεγχο ή όχι.

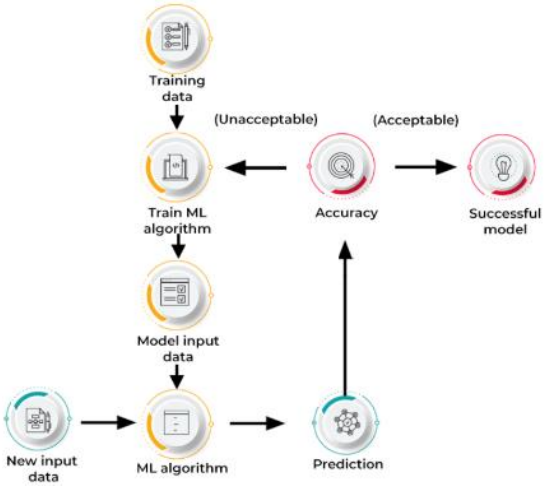
Για πειραματισμό χρησιμοποιήθηκαν 2 γνωστές τεχνικές ανίχνευσης ανωμαλιών που όμως αντί σε ολόκληρο το σύνολο των πακέτων, εφαρμόστηκαν πάνω στα πακέτα που έχουν δειγματοληφθεί προηγουμένως από τις πιο πάνω count-based τεχνικές. Η τεχνική ανίχνευσης ανωμαλιών είναι η μέθοδος Change-Point-Detection (CPD) [20] ενώ η δεύτερη είναι η Principal Component Analysis (PCA-based) Anomaly Detection [21]. Επίσης ως σύνολο δεδομένων έχουν συλλεχθεί πραγματικά δεδομένα από την σύνδεση μεταξύ NTUA – GRNET στα οποία υπάρχουν σε διαφορετικά διαστήματα 5 διαφορετικά SYN Attacks [22] με Attack Ratio που κυμαίνονται από 1% εως 20% σε σχέση με τη πραγματική κίνηση. Από τα πειράματα κατέληξαν στο ότι τόσο οι μετρικές που βασίζονται στα ίδια τα πακέτα (packet-based) όσο και οι μετρικές που βασίζονται στην κίνηση (flow-based) έχουν παρόμοιες επιδόσεις οι οποίες βασίζονται περισσότερο στο ρυθμό δειγματοληψίας- Sampling Rate και πολύ λιγότερο με ποια μεθοδολογία έχει γίνει η δειγματοληψία. Ακόμη, σε μικρούς ρυθμούς δειγματοληψίας παρατηρείται ότι η συστηματική δειγματοληψία παρουσιάζει χειρότερα αποτελέσματα συγκριτικά με τις υπόλοιπες τεχνικές. Τέλος, η συστηματική δειγματοληψία παραμένει η πιο ευρεία διαδεδομένη τεχνική λόγω της απλότητας υλοποίησης της.

1.3.2 Μηχανική Μάθηση – Machine Learning

Η ανάπτυξη των υπολογιστικών πόρων μαζί με η δυνατότητα να εκτελούνται εκατομμύρια πράξεις σε ελάχιστο χρονικό χρόνο και σε συνδυασμό με την προοπτική που έχει ο τομέας αυτός, οδήγησε όλη την επιστημονική κοινότητα τα τελευταία χρόνια να δώσει τεράστια έμφαση στο κεφάλαιο της Τεχνητής Νοημοσύνης – Artificial Intelligence[23] και της Μηχανικής Μάθησης [24]. Ο κύριος κοινός στόχος είναι η ανάπτυξη αποδοτικών

αλγορίθμων οι οποίοι θα αναπαριστούν όσο το δυνατό πιο καλά τη διαδικασία λήψης αποφάσεων του ανθρώπινου μυαλού.

Η μηχανική μάθηση βασίζεται στην αξιοποίηση προηγούμενης γνώστης, για την πρόβλεψη μελλοντικών σεναρίων. Οποιοδήποτε πρόβλημα λύνεται με την συγκεκριμένη λογική, είναι πάρα πολύ πιθανό να έχουν ήδη μελέτες και να αναπτυχθήκαν μοντέλα μηχανικής μάθησης πάνω σε αυτό. Μετά από τη συλλογή παλιών περιστατικών δημιουργείται ένα σύνολο δεδομένων, το οποίο «εκπαιδεύει» μέσω ενός αλγορίθμου ένα μοντέλο, έτσι ώστε να προβλέπει σωστά παρόμοια περιστατικά στο μέλλον. Όπως φαίνεται και από το σχήμα 1.3.2 για την εκπαίδευση ενός μοντέλου απαιτούνται 2 σύνολα δεδομένων. Το πρώτο είναι το σύνολο δεδομένων εκπαίδευσης (Train Data) και το δεύτερο είναι το σύνολο δεδομένων ελέγχου (Test Data). Στη συνέχεια και εφόσον γίνει η προεπεξεργασία των δεδομένων, η επιλογή του ταξινομητή όπως και διαφόρων παραμέτρων ξεκινάει η διαδικασία εκπαίδευσης με την αξιοποίηση του Train Dataset. Ακολούθως, γίνεται έλεγχος της απόδοσης μέσω του Test Dataset. Σε αυτό το στάδιο το εκπαιδευόμενο μοντέλο κάνει προβλέψεις σε ένα νέο σύνολο δεδομένων το οποίο δεν γνωρίζει(Test Dataset) με στόχο την αξιολόγηση της απόδοσης του. Αν η απόδοση είναι ικανοποιητική, τότε ο αλγόριθμος ήταν πετυχημένος και το μοντέλο έτοιμο για χρήση για μελλοντικές προβλέψεις. Αν όμως η απόδοση δεν έφτασε τα επιθυμητά επίπεδα, τότε η διαδικασία επαναλαμβάνεται με διαφορετικές παραμέτρους εως ότου προκύψει μοντέλο με αποδεκτή απόδοση.



Σχήμα 1.3.2: Διαδικασία Εκπαίδευσης στην Μηχανική Μάθηση[25]

Η μηχανική μάθηση έχει εφαρμογές σε πολλά πεδία της καθημερινής μας ζωής [26]. Έχουν αναπτυχθεί μοντέλα που εξυπηρετούν τους τομείς της υγείας, της οικονομίας, των μεταφορών (π.χ. Google Maps) κλπ. Τέλος, μπορεί να αποτελέσει ένα αρκετά χρήσιμο εργαλείο στον τομέα της ασφάλειας καθώς συμβάλει στην προσπάθεια εντοπισμού αλλά

και ταυτόχρονα στην προσπάθεια να αποτραπεί οποιαδήποτε κακόβουλη κίνηση στην ομαλή κυκλοφορία ενός δικτύου.

1.4 Κατηγορίες Μοντέλων Μηχανικής Μάθησης

Υπάρχουν 4 μεγάλες κατηγορίες στις οποίες μπορούν να συνοψιστούν όλες οι μέθοδοι της μηχανικής μάθησης βασισμένες στο βαθμό που ο ανθρώπινος παράγοντας επηρεάζει και επιβλέπει την διαδικασία της εκμάθησης – Learning Process. Οι κατηγορίες αυτές είναι: **Supervised Learning, Unsupervised Learning, Semi-Supervised Learning και Reinforcement Learning.**

1.4.1 Supervised Learning

Στην **επιβλεπόμενη μάθηση – Supervised Learning** [27], τα δεδομένα έχουν κατηγοριοποιηθεί και για κάθε είσοδο γνωρίζουμε την επιθυμητή έξοδο. Για κάθε εγγραφή στο σύνολο δεδομένων υπάρχει μια ετικέτα (Label) που φανερώνει σε ποια κλάση ανήκει η κάθε εγγραφή. Στόχος είναι με την αξιοποίηση της γνώσης του που ανήκει κάθε στοιχείο, να γίνει μια όσο το πιο αποδοτική ανάπτυξη ενός μοντέλου όπου με τις δεδομένες γνωστές πληροφορίες να κατηγοριοποιούνται σωστά τα νέα άγνωστα δεδομένα της εισόδου. Κάποιες από τις πιο γνωστές τεχνικές είναι το classification και το regression.

Classification

Στα προβλήματα Ταξινόμησης – Classification Problems [28] στόχος είναι το σύνολο δεδομένων να διαχωριστεί σωστά στις υποκατηγορίες τις οποίες ανήκουν. Οι πιθανές κατηγορίες είναι γνωστές και προκαθορισμένες και βασίζονται στο πλήθος των διαφορετικών πιθανόν τιμών που παίρνει η ετικέτα – Label. Με την αξιοποίηση του συνόλου δεδομένων εκπαιδεύονται μοντέλα επιβλεπόμενης μηχανικής μάθησης, τα οποία στη συνέχεια πρέπει να κατατάσσουν όσο πιο αποδοτικά είναι δυνατό τις νέες εγγραφές στις κλάσεις που ανήκουν. Το πρόβλημα μπορεί να χρειάζεται είτε την δημιουργία ενός δυαδικού ταξινομητή (Binary Classification) σε περίπτωση που υπάρχουν μόνο 2 πιθανές κλάσεις ή ένα μοντέλο που να είναι σε θέση να διαχωρίζει περισσότερες κλάσεις (Multiclass Classification).

Στην περίπτωση του δυαδικού ταξινομητή υπάρχουν σύνολα δεδομένων τα οποία δεν είναι ισορροπημένα αλλά η μια κλάση υπερτερεί ποσοτικά της δεύτερης (Imbalanced Classification). Αυτό συμβαίνει σε αρκετές περιπτώσεις όπως για παράδειγμα στον τομέα της υγείας, όπου ένας ασθενής πάσχει από μια σπάνια ασθένεια αλλά και σε ένα δίκτυο, όπου η ανώμαλη κίνηση είναι μηδαμινή συγκριτικά με την ομαλή φυσιολογική κίνηση. Σε μη ισορροπημένα σύνολα δεδομένων, η μετρική Accuracy κρίνεται ακατάλληλη και είναι προτιμότερο η χρήση των μετρικών F2-Score και AUC Score. Ο λόγος είναι ότι το σημαντικότερο είναι να εντοπιστούν ορθά όσο το δυνατό περισσότερες εγγραφές που ανήκουν στην κλάση των ανωμαλιών.

Regression

Στα προβλήματα Παλινδρόμησης – Regression Problems [29] είναι μια ακόμη υποκατηγορία του κλάδου της επιβλεπόμενης μηχανικής μάθησης και η κύρια διαφορά που υπάρχει συγκριτικά με το classification είναι πως μια ετικέτα μπορεί να πάρει οποιαδήποτε συνεχόμενη τιμή. Στόχος είναι με βάση το σύνολο δεδομένων εισόδου, στα οποία είναι γνωστό η συνεχή τιμή, είναι να αξιοποιηθούν ώστε να σχηματιστεί μια όσο το δυνατό ιδανικότερη γραμμή ή καμπύλη, όπου η χρήση της θα προβλέπει αξιόπιστα μελλοντικές εισόδους. Οι μετρικές στο κεφάλαιο 1.8.2 αξιολογούν το σφάλμα και είναι το μέσο με το οποίο αξιολογείται η απόδοση του εκπαιδευόμενου μοντέλου. Τέλος πολλές φορές υπάρχει έντονο το φαινόμενο της υπερπροσαρμογής - Overfitting [30] του μοντέλου στα δεδομένα εισόδου , το οποίο αρκετές φορές εξασθενεί με την ομαλοποίηση των δεδομένων – Regularization[31] χρησιμοποιώντας για παράδειγμα τους συντελεστές ομαλοποίησης L1,L2.

1.4.2 Unsupervised Learning

Μια δεύτερη μεγάλη κατηγορία είναι η **μη επιβλεπόμενη μάθηση – Unsupervised Learning** [32] στην οποία ανήκουν οι περιπτώσεις όπου δεν υπάρχει οποιαδήποτε πληροφορία για το σε ποια κατηγορία ανήκει το κάθε στοιχείο από το σύνολο δεδομένων. Στόχος είναι μέσα από την εκπαίδευση ενός μοντέλου, να εντοπιστούν διάφορα κοινά μοτίβα μεταξύ των δεδομένων εισόδου έτσι ώστε να ομαδοποιηθούν με κατάλληλο τρόπο σε κατηγορίες ανάλογα με τα κοινά χαρακτηριστικά τους.

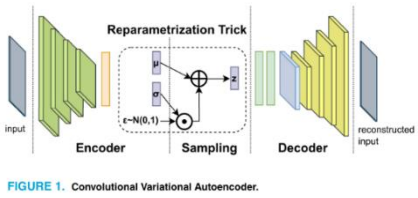


FIGURE 1. Convolutional Variational Autoencoder.

Σχήμα 1.4.2: Convolutional Variational Autoencoder

Ένα από τα πιο συνηθισμένα είδη μοντέλων που χρησιμοποιούνται στην μη επιβλεπόμενη μάθηση είναι τα Generative Models όπως για παράδειγμα οι Variational Auto-Encoders (VAE). Γενικά οι VAE αποτελούνται από 1 κωδικοποιητή και 1 αποκωδικοποιητή. Ο πρώτος παίρνει τα δεδομένα εισόδου και εκπαιδεύεται έτσι ώστε να κωδικοποιεί αυτά τα δεδομένα σε μια κατανομή (συνήθως Gaussian) μέσα σε ένα κρυμμένο χώρο μικρότερων διαστάσεων (latent space). Στη συνέχεια δημιουργείται ένα διάνυσμα z μέσω της διαδικασίας της δειγματοληψίας της κατανομής το οποίο προωθείται στον αποκωδικοποιητή. Από την άλλη, το καθήκον του αποκωδικοποιητή είναι να εκπαιδευτεί έτσι ώστε να μπορεί μέσω του διανύσματος z να αυξήσει τις διαστάσεις και να πάρει ως έξοδο δεδομένο x' όσο πιο κοντά στο αρχικό x.

Υπάρχουν πολλές έρευνες και υλοποιήσεις των VAE για μη επιβλεπόμενη μάθηση και πολλές φορές γίνεται και συνδυασμός με άλλα εργαλεία. Μια από αυτές έγινε από τους Run-Qing Chen, Guang-Hui Shi, Wan-Lei Zhao και Chang-Hui Liang [33], οι οποίοι συνέδεσαν έναν VAE για την ανίχνευση ανωμαλιών μαζί με ένα LSTM (Long Short-Term

Memory) Prediction Block έτσι ώστε όχι μόνο να εντοπίζουν πιθανές ανωμαλίες μέσα σε ένα συγκεκριμένο χρονικό διάστημα, αλλά ταυτόχρονα να έχουν μια εκτίμηση για την αμέσως επόμενη είσοδο στο σύστημα. Για τον έλεγχο πήραν το σύνολο δεδομένων Yahoo και KPI και μέσα από διάφορους πειραματισμούς κατέληξαν στο συμπέρασμα ότι ο συνδυασμός των VAE και LSTM δίνει καλύτερα αποτελέσματα από ότι αν δούλευαν ξεχωριστά, τόσο στην ανίχνευση ανωμαλιών (VAE), όσο και στην μελλοντική πρόβλεψη (LSTM)

1.4.3 Semi-Supervised Learning

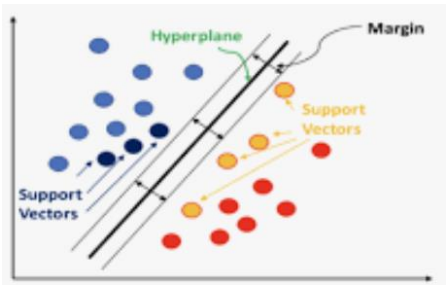
Στην **ημί-επιβλεπόμενη μάθηση – Semi-Supervised Learning** [34] το σύνολο δεδομένων αποτελείται από ένα συνδυασμό των προηγούμενων 2 περιπτώσεων, δηλαδή παρόλο που ένα μεγάλο μέρος του συνόλου δεδομένου δεν έχει κατηγοριοποιηθεί, ένα μικρό δείγμα από σύνολο έχει τοποθετηθεί στην σωστή κατηγορία. Στόχος είναι η εύρεση αλγορίθμων έτσι ώστε να γίνει συνδυασμός των πιο πάνω μεθόδων και να επιτευχθεί όσο καλύτερη επίδοση. Μια τέτοια υλοποίηση έγινε από τους Lukas Ruff Robert A. Vandermeule, Alexander Binder, Nico Görnitz, Emmanuel Müller, Klaus-Robert Müller,Marius Kloft [35] όπου ανέπτυξαν τον αλγόριθμο Deep SAD, μια μέθοδος που αποτελεί γενικοποίηση του μη-επιβλεπόμενου DEEP SVDD. Έχοντας ως δεδομένα τα 3 Dataset CIFAR-10,MNIST και Fashion MNIST στα οποία υπάρχουν 10 διαφορετικές κατηγορίες, έφτιαξαν ένα σύνολο δεδομένων όπου θεώρησαν την 1 από τις 10 κατηγορίες ως ανωμαλία και τις υπόλοιπες φυσιολογικές. Επίσης έχουν θεώρησαν ελάχιστα δεδομένα ως labeled και τα πλείστα τα τοποθέτησαν unlabeled. Από τις μετρήσεις που πήραν έδειξαν ότι το AUC Score τις πλείστες φορές είναι ψηλότερο στην δική τους περίπτωση συγκριτικά με τα υπόλοιπα Supervised και Unsupervised μοντέλα. Αυτό αποδεικνύει πως σε ένα σύνολο δεδομένων που τα labeled δεδομένα είναι λίγα η επιβλεπόμενη μάθηση δεν είναι η ιδανική και επίσης ότι η αξιοποίηση των λίγων labeled δεδομένων επηρεάζει θετικά ένα μοντέλο και αυξάνει την επίδοση του σχετικά με τα unsupervised μοντέλα.

Μια δεύτερη προσέγγιση είναι η αξιοποίηση των Generative Adversarial Networks (GAN). Η λογική είναι πως αν μπορούμε να διαχωρίσουμε ένα σύνολο δεδομένων σε φυσιολογικά και ανωμαλίες, τότε μπορούμε να εκπαιδεύσουμε ένα GAN μόνο με τα ομαλά δεδομένα. Ως αποτέλεσμα, μετά στην φάση ελέγχου, το GAN θα είναι ικανό να αναπαράγει τα φυσιολογικά δεδομένα, όμως επειδή δεν έχει εκπαιδευτεί σε αυτά, δεν θα είναι σε θέση να παράγει αντίγραφα των ανωμαλιών οπότε με αυτό τον τρόπο είναι δυνατή η ανίχνευση τους. Προσέγγιση της συγκεκριμένης ιδέας έγινε από τους Samet Akcay, Amir Atapour-Abarghouei και Toby P. Breckon [36] όπου με τη βοήθεια των MNIST και CIFAR-10 dataset ανέπτυξαν ένα GAN μοντέλο όπου στόχο έχει την ανίχνευση ανωμαλιών και με κάποιες παραμετροποιήσεις συγκριτικά με τα προυπάρχοντα μοντέλα σε αυτό τον τομέα κατάφεραν να έχουν ψηλότερο AUC Score (κεφάλαιο 1.8.1).

1.5 Μοντέλα Επιβλεπόμενης Μηχανικής Μάθησης – Supervised Learning Models

1.5.1 Support Vector Classifier

Ένας τρόπος για προβλεφθεί σε ποια κλάση ταξινομείται ένα άγνωστο σημείο είναι με την χρήση μηχανών διανυσμάτων υποστήριξης (Support Vector Machines) [37]. Στην περίπτωση που υπάρχει ένα σύνολο δεδομένων για εκπαίδευση με 2 πιθανές κλάσεις (δυαδικό) και με γνωστή την κλάση στην οποία ανήκουν, τότε μπορούμε με επιβλεπόμενη μηχανική μάθηση και ένα ταξινομητή διανυσμάτων υποστήριξης να προβλέψουμε την κλάση μελλοντικών σημείων.

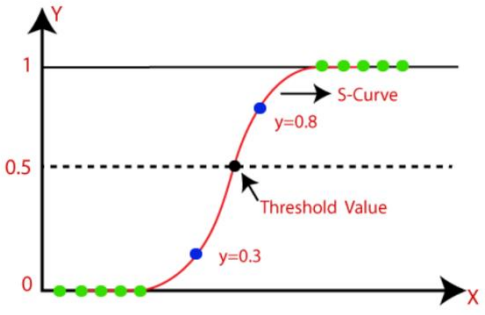


Σχήμα 1.5.1: Support Vector Classifier

Κατά την διάρκεια της εκπαίδευσης, ο αλγόριθμος προσπαθεί να εντοπίσει το βέλτιστο διαχωριστικό υπερεπίπεδο το οποίο θα διαχωρίσει το χώρο σε 2 μικρότερα επίπεδα, 1 για κάθε κλάση. Επειδή όμως μπορεί να υπάρχουν πολλά πιθανά υπερεπίπεδα διαχωρισμού, ο αλγόριθμος ταυτόχρονα προσπαθεί να μεγιστοποιήσει το περιθώριο, δηλαδή την απόσταση μεταξύ των κοντινότερων σημείων από κάθε κλάση και του διαχωριστικού υπερεπιπέδου. Τα σημεία αυτά που βρίσκονται πιο κοντά στο υπερεπίπεδο ονομάζονται διανύσματα υποστήριξης (Supporting Vectors)

1.5.2 Logistic Regression Classifier

Στην κατηγορία της επιβλεπόμενης μηχανικής μάθησης υπάρχουν επίσης οι ταξινομητές Logistic Regression. Στις συγκεκριμένες περιπτώσεις γίνεται χρήση της σιγμοειδής συνάρτησης [38] και στόχος είναι η εκπαίδευση ενός μοντέλου για τη δυαδική ταξινόμηση (True/False ή 1/0) ενός συνόλου δεδομένου. Ως είσοδο το εκπαιδευμένο μοντέλο λαμβάνει τα ανεξάρτητα χαρακτηριστικά του συνόλου δεδομένου, τα οποία «προσαρμόζονται» στο σχήμα “S” της σιγμοειδής συνάρτησης έτσι ώστε να υπολογιστεί η πιθανότητα του να ανήκει στην κατηγορία 0 ή 1 (τιμή 1 = 100% δηλώνει πως ανήκει σίγουρα στην 1 ενώ η τιμή 0 = 0% δηλώνει πως ανήκει σίγουρα στην κατηγορία 0). Στη συνέχεια με βάση το κατώφλι-threshold με εύρος τιμών 0 εώς 1 που έχει ανατεθεί , το μοντέλο προβλέπει το που ανήκει τελικά η εγγραφή αυτή. [39]



Η ιδέα του Logistic Regression παρουσιάζει παρόμοια λογική με την φιλοσοφία του Linear Regression με την κύρια διαφορά τους να είναι στο ότι το Linear Regression χρησιμοποιείται την επίλυση προβλημάτων παλινδρόμησης(Regression), ενώ αντίθετα το Logistic Regression για την επίλυση προβλημάτων ταξινόμησης(Classification) [40]

Σχήμα 1.5.2: Logistic Regression Classifier

1.5.3 k-Nearest Neighbors Classifier

Ένας ακόμη αλγόριθμος που χρησιμοποιείται για την ταξινόμηση στην επιβλεπόμενη μηχανική μάθηση είναι ο αλγόριθμος των k-κοντινότερων γειτόνων (k-Nearest Neighbors) [41]. Η λογική του συγκεκριμένου μοντέλου είναι ότι αρχικά κάθε εγγραφή στο σύνολο δεδομένων εκπαίδευσης αποθηκεύεται ένα σημείο στον n-διάστατο χώρο βάση των χαρακτηριστικών. Στη συνέχεια όταν χρειάζεται να γίνει πρόβλεψη της κλάσης ενός νέου σημείου, ο αλγόριθμος υπολογίζει τις αποστάσεις του συγκριτικά με τα γνωστά σημεία έτσι ώστε να εντοπιστούν μέσω μιας ευριστικής (Manhattan ή Ευκλείδεια) τα k κοντινότερα σημεία του συνόλου εκπαίδευσης. Το νέο αυτό σημείο τελικά ταξινομείται ανάλογα με το που ανήκει η πλειοψηφία των k κοντινότερων σημείων σε αυτό. Η ίδια διαδικασία μπορεί να υλοποιηθεί και στο Regression, με τη διαφορά ότι η τιμή του νέου σημείου θα είναι ο μέσος όρος των k κοντινότερων του.

Ο συγκεκριμένος ταξινομητής συνήθως εφαρμόζεται σε μικρά και ισορροπημένα σύνολα δεδομένων. Αν το σύνολο δεδομένων είναι πολύ μεγάλο, τότε θα απαιτείται μεγάλο υπολογιστικό κόστος καθώς για κάθε νέο σημείο πρέπει να υπολογίζονται όλες οι αποστάσεις του σε σχέση με τα γνωστά σημεία. Επίσης αν το μεγαλύτερο ποσοστό των σημείων ανήκουν σε μια κλάση, τότε ο πολύ πιθανόν να υπάρχει μια «προκατάληψη» προς τη συγκεκριμένη κλάση στην διαδικασία της πρόβλεψης.

1.5.4 Decision Tree Classifier

Τα Δέντρα Αποφάσεων - Decision Trees [42] είναι μοντέλα που χρησιμοποιούνται στην επιβλεπόμενη Μηχανική Μάθηση τόσο σε περιπτώσεις ταξινόμησης (Classification) όσο και για σκοπούς παλινδρόμησης (Regression). Η κεντρική ιδέα είναι η δημιουργία ενός δέντρου το οποίο θα διαχωρίζεται με βάση τα χαρακτηριστικά στα δεδομένα εκπαίδευσης. Ο διαχωρισμός γίνεται με τέτοιο τρόπο έτσι ώστε στο τέλος σε κάθε κόμβο φύλλο να αναπαρίσταται η καλύτερη δυνατή πρόβλεψη σχετικά με την κατηγορία για την περίπτωση του Classification ή η πιο πιθανή αριθμητική τιμή στην περίπτωση του Regression.

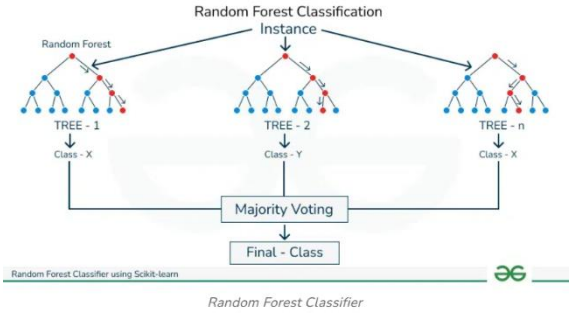
Για την επιλογή της ρίζας γίνεται προσπάθεια για την εύρεση του χαρακτηριστικού το οποίο διαχωρίζει όσο πιο αποδοτικά γίνεται τα δεδομένα στις διάφορες κατηγορίες, δηλαδή κάθε φορά η επιλογή του χαρακτηριστικού γίνεται με στόχο το να υπάρξει το πιο

μεγάλο δυνατό κέρδος πληροφορίας. Υπάρχουν διάφοροι αλγόριθμοι για τον υπολογισμό του κέρδους της πληροφορίας, με τους πιο γνωστούς να είναι με βάση την εντροπία ή με τον υπολογισμό του Gini Index [43] στην περίπτωση του Classification και με βάση του Mean Square Error στην περίπτωση του Regression. Ο συγκεκριμένος αλγόριθμος για την επιλογή του βέλτιστου χαρακτηριστικού που θα χωρίσει το δέντρο σε «μικρότερα δέντρα» γίνεται επαναληπτικά μέχρι να καταλήξει σε κόμβους – φύλλα οι οποίοι θα παρέχουν την πρόβλεψη για την πιθανή κατηγορία που ανήκει το δεδομένο.

Ένα από τα βασικά προβλήματα των δέντρων αποφάσεων είναι ότι είναι πολύ εύκολο να υπάρξει υπερπροσαρμογή – Overfitting καθώς είναι με τέτοιο τρόπο δομημένα έτσι ώστε να μπορούν να ταξινομήσουν όλα τα δεδομένα εκπαίδευσης στην σωστή κατηγορία, κάτι το οποίο θα οδηγήσει σε μεγάλα δέντρα απόφασης που είναι δύσκολο να γενικευτούν στα δεδομένα ελέγχου. Για την αποφυγή της υπερπροσαρμογής γίνεται προσπάθεια για συστηματικής απλοποίησης του δέντρου μέσω μεθόδων όπως της ιδέας του κλαδέματος [44] (pruning) αλλά και με την εισαγωγή ενός νέου ταξινομητή που κατασκευάζεται από πολλά δέντρα αποφάσεως μαζί, γνωστό ως Τυχαίο Δάσος– Random Forest.

1.5.5 Random Forest Classifier

Κάτι που είχε παρατηρηθεί είναι πως ακόμη και για μικρές διαφορές στο σύνολο δεδομένων, οι αλγόριθμοι εκμάθησης δέντρων αποφάσεων μπορούν να κατασκευάσουν δέντρα με ελαφρώς διαφορετική δομή αλλά με μεγάλη διαφορά σχετικά στις προβλέψεις τους. Εξαιτίας της παρατήρησης αυτή έχει αναπτυχθεί ο αλγόριθμος Random Forest [45] με κεντρική ιδέα τη δημιουργία πολλών διαφορετικών δέντρων αποφάσεων χρησιμοποιώντας ξεχωριστά τμήματα του συνόλου δεδομένων. Αφού έχει δημιουργηθεί ένα «δάσος» από δέντρα αποφάσεων, στο τέλος η πρόβλεψη γίνεται με βάση των συνδυασμό όλων αυτών των διαφορετικών δέντρων και της επιλογή της επικρατέστερης πρόβλεψης.



Random Forest Classifier

Σχήμα 1.5.5: Random Forest Classifier

Το σύνολο δεδομένων μπορεί να διαχωριστεί σε τμήματα ανάλογα με την επιλογή συγκεκριμένης μεθόδου όπως για παράδειγμα του Bagging και Feature Bagging. Στο Bagging γίνεται διαχωρισμός του συνόλου δεδομένων σε k διαφορετικά υποσύνολα, ενώ αντίθετα στο Feature Bagging γίνεται επιλογή k διαφορετικών υποσυνόλων των χαρακτηριστικών έτσι ώστε να κατασκευαστούν k νέα δέντρα αποφάσεων με σύνολο δεδομένων μικρότερων διαστάσεων. Και στις 2 περιπτώσεις η πρόβλεψη της εξόδου για μια νέα είσοδο θα είναι η πρόβλεψη που θα δώσει η πλειοψηφία των k δέντρων

αποφάσεων. Με αυτό τον τρόπο περιορίζεται το πρόβλημα της υπερπροσαρμογής το οποίο παρουσιάζεται όταν υπάρχει ένα μεγάλο ενιαίο δέντρο απόφασης.

1.5.6 Voting Classifier

Όλα τα πιο πάνω αποτελούν μοντέλα επιβλεπόμενης μηχανικής μάθησης τα οποία εκπαιδεύονται έτσι ώστε να κατηγοριοποιούν μια εγγραφή ανάλογα με τα χαρακτηριστικά της στην κλάση την οποία πιστεύουν ότι ανήκει. Μια διαφορετική ιδέα είναι αντί να υπάρχει ένα μοντέλο που να κάνει αυτό το διαχωρισμό στο που ανήκει μια εγγραφή, να υπάρχει μια ομάδα ταξινομητών για την οποία ο κάθε ταξινομητής θα ψηφίζει το τι πιστεύει. Στο τέλος ένας νέος ταξινομητής (Voting Classifier) [46], λαμβάνοντας υπόψη τις αποφάσεις και τις πιθανότητες της απόφασης κάθε ταξινομητή, είναι υπεύθυνος για την τελική κατηγοριοποίηση της εγγραφής στην κλάση την οποία πιστεύει ότι ανήκει. Για τον Voting Classifier υπάρχουν μερικές παραμέτροι οι οποίοι καθορίζουν τον τρόπο λειτουργίας του ταξινομητή. Η πρώτη παράμετρος είναι μια λίστα από 'estimators'-εκτιμητές, στην οποία θα καθορίζονται ποιοι και πόσοι ταξινομητές πρέπει να εκπαιδευτούν και θα ψηφίζουν όταν μια νέα εγγραφή χρειάζεται κατηγοριοποίηση. Επιπλέον υπάρχει η παράμετρος 'voting', η οποία δηλώνει τον τρόπο με τον οποίο δίνεται η τελική ψήφος από τον Voting Classifier, αφού πρώτα έχουν ψηφίσει τα υπόλοιπα μοντέλα. Υπάρχουν 2 επιλογές, το Soft Voting και το Hard Voting [47]. Στο Hard Voting ο Voting Classifier προβλέπει ως έξοδο την κλάση η οποία έχει λάβει τις περισσότερες ψήφους από τα μοντέλα που έχουν καθοριστεί και εκπαιδευτεί προηγουμένως. Αντίθετα, στο Soft Voting ο Voting Classifier συνδυάζει τις πιθανότητες του να ανήκει μια εγγραφή σε κάθε συγκεκριμένη κλάση από όλα τα υπόλοιπα μοντέλα και στο τέλος επιλέγει την κλάση με την μεγαλύτερη συνολική πιθανότητα. Ακόμη με τη χρήση της παραμέτρου 'weights' μπορεί να δοθεί διαφορετική βαρύτητα σε κάθε ταξινομητή και ανάλογα με τις τιμές, επηρεάζεται η επίδραση του κάθε ταξινομητή στην τελική απόφαση του Voting Classifier. Για μεγάλη η τιμή του weight ο συγκεκριμένος ταξινομητής έχει μεγάλη η επιρροή στο Voting Classifier, ενώ για μικρές τιμές ο ταξινομητής ναι μεν επηρεάζει την απόφαση με την ψήφο του, αλλά όχι στο σημείο που το κάνουν τα υπόλοιπα μοντέλα.

1.6 Υπερπαραμέτροι στην Μηχανική Μάθηση

1.6.1 Τύποι μεταβλητών και κατηγορίες Υπερπαραμέτρων

Στην μηχανική μάθηση υπάρχουν 2 διαφορετικοί τύποι μεταβλητών που ελέγχουν την συμπεριφορά της εκμάθησης του αλγορίθμου. Το πρώτο είδος είναι οι παράμετροι, που μπορούν να χαρακτηριστούν ως μεταβλητές που μαθαίνουν μέσα από το σύνολο δεδομένων εκπαίδευσης, έτσι ώστε να παριστάνουν τις σχέσεις μεταξύ των δεδομένων, να διαμορφώσουν τα χαρακτηριστικά και να επιτρέψουν στο μοντέλο τεχνητής νοημοσύνης να προσαρμοστεί σε περίπλοκα μοτίβα.

Στην δεύτερη κατηγορία βρίσκονται οι υπερπαράμετροι, δηλαδή μια ομάδα μεταβλητών οι οποίες είναι προκαθορισμένες από πριν ξεκινήσει η εκπαίδευση του συνόλου δεδομένων, κάτι το οποίο αποτελεί την κύρια διαφορά μεταξύ των 2 κατηγοριών μεταβλητών. Ουσιαστικά αποτελούν τα μέσα με τα οποία μπορεί ο χρήστης να επηρεάσει την συμπεριφορά και τον τρόπο που θα εκπαιδευτεί ένα συγκεκριμένο μοντέλο πριν καν ξεκινήσει η εκπαίδευση, δίνοντας του κατευθυντήριες γραμμές και περιορισμούς.

Οι υπερπαραμέτροι μπορούν να διαχωριστούν σε 3 μεγάλες υποκατηγορίες ανάλογα με τον σκοπό τους. Υπάρχουν υπερπαραμέτροι που αφορούν την **αρχιτεκτονική** του μοντέλου, κάποιες άλλες για την **βελτιστοποίηση** και τέλος, υπερπαραμέτροι υπεύθυνοι για την **ομαλοποίηση** του μοντέλου [48].

1. Υπερπαραμέτροι Αρχιτεκτονικής:

Οι συγκεκριμένοι υπερπαραμέτροι ελέγχουν την αρχιτεκτονική ενός μοντέλου επιτρέποντας τον έλεγχο της πολυπλοκότητας και το πως αναπαρίστανται τα δεδομένα. Κάποια παραδείγματα είναι ο αριθμός των στρωμάτων σε ένα νευρωνικό δίκτυο, ο αριθμός των νευρώνων σε κάθε στρώμα ή στην περίπτωση του Random Forest, ο αριθμός των δέντρων αποφάσεων που θα δημιουργηθούν.

2. Υπερπαραμέτροι Βελτιστοποίησης:

Κατά τη διάρκεια της εκπαίδευσης το μοντέλο προσπαθεί με βάση τα υπάρχοντα δεδομένα να παράξει το βέλτιστο αποτέλεσμα. Σε αυτή την κατηγορία ανήκουν όσες υπερπαραμέτροι ασχολούνται με το να επηρεάσουν το πως τα βάρη ενός μοντέλου ανανεώνονται σε αυτό το διάστημα δίνοντας δικαίωμα για έλεγχο στο ρυθμό της ταχύτητας και στην σταθερότητα της βελτιστοποίησης. Κάποια από τα παραδείγματα των συγκεκριμένων υπερπαραμέτρων είναι ο ρυθμός μάθησης σε ένα νευρωνικό δίκτυο, το batch size και ο αριθμός επαναλήψεων.

3. Υπερπαραμέτροι Ομαλοποίησης:

Σε αυτή την κατηγορία ανήκουν όσες υπερπαραμέτροι έχουν ως στόχο την εισαγωγή κάποιων περιορισμών στις παραμέτρους κατά την διάρκεια της εκπαίδευσης έτσι ώστε να αποτραπεί μια πιθανή υπερπροσμαρμογή στο σύνολο των δεδομένων εκπαίδευσης.

Κάποια τέτοια παραδείγματα είναι ο έλεγχος της δύναμης των συντελεστών L1,L2 αλλά και ο ρυθμός εγκατάλειψης – Dropout Rate. Σε ένα νευρωνικό δίκτυο το Dropout Rate ελέγχει το ποσοστό των τυχαίων νευρώνων (διαφορετικών κάθε φορά) που απενεργοποιούνται έτσι ώστε να περιοριστεί η υπερπροσαρμογή και να μην μαζευτεί όλη η πληροφορία σε ένα νευρώνα, με το ρίσκο να υπάρξει τελικά υποπροσαρμογή (Underfitting).

1.6.2 Υπερπαραμέτροι στο Random Forest

Όπως σε όλα τα μοντέλα της μηχανικής μάθησης, έτσι και στον Random Forest υπάρχουν συγκεκριμένοι υπερπαράμετροι, οι οποίοι ορίζονται από πριν έτσι ώστε να καθοριστεί ο τρόπος εκπαίδευσης του μοντέλου. Οι πιο διαδεδομένοι υπερπαραμέτροι για τον ταξινομητή Random Forest είναι οι εξής [49]:

1.n_estimators:

Καθορίζει τον αριθμό των διαφορετικών δέντρων αποφάσεων που θα δημιουργηθούν κατά την διάρκεια της εκπαίδευσης. Ο προκαθορισμένος αριθμός δέντρων είναι **100**.

2.max_depth:

Καθορίζει το μέγιστο ύψος το οποίο μπορεί να φτάσει κάθε δέντρο. Η επιλογή της τιμής αυτής είναι σημαντική καθώς αν είναι πολύ μικρή, το ύψος δεν θα είναι αρκετό για να διαχωριστεί η πληροφορία και τα αποτελέσματα των μετρικών θα είναι χαμηλά. Επίσης αν η τιμή είναι πολύ μεγάλη, τότε θα εμφανιστεί το πρόβλημα της υπερπροσαρμογής το οποίο πρέπει να αποφευχθεί. Η προκαθορισμένη τιμή είναι **None**, δηλαδή το δέντρο θα μεγαλώσει μέχρι τέλους ή μέχρι να ικανοποιείται η συνθήκη στο **min_samples_split**

3.min_samples_split:

Καθορίζει τον ελάχιστο αριθμό δειγμάτων που απαιτούνται σε ένα κόμβο έτσι ώστε ο κόμβος να διαχωριστεί και να δημιουργηθούν 2 νέοι κόμβοι-παιδιά. Αν η τιμή των δειγμάτων στον κόμβο δεν φτάσει αυτή την προκαθορισμένη τιμή, τότε ο κόμβος αυτός θα παραμείνει κόμβος – φύλλο και θα σταματήσει η επέκταση του δέντρου προς αυτή την κατεύθυνση. Μικρή τιμή στο min_samples_split μπορεί να οδηγήσει σε υπερπροσαρμογή, ενώ αντίθετα μια μεγάλη τιμή μπορεί να εμφανίσει προβλήματα υποπροσαρμογής. Η προκαθορισμένη τιμή είναι **2.**

4.min_samples_leaf:

Παρόμοια με προηγουμένως, η συγκεκριμένη υπερπαράμετρος καθορίζει τον ελάχιστο αριθμό δειγμάτων που απαιτούνται σε ένα κόμβο – φύλλο. Αν ένας διαχωρισμός οδηγεί σε κόμβο-φύλλο με αριθμό δειγμάτων μικρότερο από τον προκαθορισμένο, τότε ο διαχωρισμός αυτός απορρίπτεται. Μεγάλη τιμή στο Min_samples_leaf μπορεί να οδηγήσει σε υποπροσαρμογή. Ο προκαθορισμένος αριθμός είναι **1.**

5.max_features:

Στον Random Forest, όταν βρισκόμαστε σε ένα κόμβο και χρειάζεται να διαχωριστεί το δέντρο περισσότερο, τότε μέσα από ένα πλήθος χαρακτηριστικών επιλέγεται αυτό που ο διαχωρισμός του θα δώσει το μεγαλύτερο κέρδος πληροφορίας. Η υπερπαράμετρος max_features καθορίζει τον αριθμό των δειγμάτων που ελέγχονται στην συγκεκριμένη περίπτωση και οι τιμές που μπορεί να πάρει είναι ίσο με τη ρίζα (sqrt), τον λογάριθμο (log2) και τον ακριβή αριθμό των δειγμάτων (None). Όσο πιο μεγάλη τιμή έχει το max_features, τόσο πιο όμοια θα είναι τα δέντρα αποφάσεων μεταξύ τους στο Random

Forest, ενώ αντίθετα για μικρές τιμές τα δέντρα θα διαφέρουν, κάτι που είναι πιθανό να μειώσει προβλήματα υπερπροσαρμογής. Η προκαθορισμένη τιμή είναι **sqrt,** δηλαδή το max_features ισούται με την ρίζα του πλήθους του συνόλου των χαρακτηριστικών.

6.max_leaf_nodes:

Καθορίζει τον μέγιστο αριθμό των κόμβων – φύλλων που μπορεί να έχει κάθε δέντρο απόφασης στο Random Forest. Όσο πιο μεγάλη τιμή έχει, τόσο πιο περιορισμένο θα είναι το βάθος σε κάθε δέντρο απόφασης. Η προκαθοριμένη τιμή είναι **None**, δηλαδή να μην υπάρχει οποιοσδήποτε περιορισμός όσο αφορά το πλήθος των κόμβων – φύλλων.

7.max_samples:

Καθορίζει τον μέγιστο αριθμό δειγμάτων από το σύνολο δεδομένων εκπαίδευσης για το κάθε δέντρο απόφασης. Οι πιθανές τιμές που μπορεί να πάρει είναι ένας ακέραιος που θα συμβολίζει τον ακριβή αριθμό δειγμάτων για την εκπαίδευση κάθε δέντρου, ένας δεκαδικός στο εύρος 0.0 – 1.0 το οποίο δηλώνει πως κάθε δέντρο θα παίρνει τόσο τις εκατό από τυχαία δείγματα για την εκπαίδευση (π.χ. 0.5 = 50% των δειγμάτων) και το “auto” που δηλώνει πως η τιμή του max_samples θα είναι ίσος με το πλήθος δειγμάτων στα δεδομένα εκπαίδευσης. Η προκαθορισμένη τιμή είναι το **auto.**

8.class_weight:

Πολλές φορές στο σύνολο εκπαίδευσης τα δείγματα δεν διαχωρίζονται ομοιόμορφα στις κλάσεις, αλλά μπορεί να υπάρχει μια κλάση στην οποία βρίσκεται η πλειοψηφία των δειγμάτων. Επομένως, το μοντέλο κατά την διάρκεια της εκπαίδευσης και λόγο της συνεχής εισόδου δειγμάτων από αυτή τη κλάση, είναι πολύ πιθανό να «εκπαιδευτεί» να κατατάσσει τα δείγματα ευκολότερα προς την συγκεκριμένη κλάση. Για την αποφυγή του προβλήματος μπορεί να αξιοποιηθεί η υπερπαράμετρος class_weight η οποία καθορίζει την βαρύτητα που θα έχει κάθε δείγμα ανάλογα με την κλάση στην οποία διαχωρίζεται. Επομένως για τις κλάσεις με ελάχιστα δείγματα, το κάθε δείγμα είναι σημαντικό και είναι αναγκαίο να δοθεί μια μεγάλη τιμή στο class_weight έτσι ώστε να ληφθεί υπόψη περισσότερο συγκριτικά με άλλα δείγματα από το μοντέλο. Η προκαθορισμένη τιμή είναι **None,** δηλαδή όλα τα δείγματα έχουν την ίδια βαρύτητα και επηρεάζουν ισάξια το αποτέλεσμα κατά τη διάρκεια της εκπαίδευσης. Μια άλλη τιμή που παίρνει το class_weight είναι το balanced, το οποίο δίνει βαρύτητα στα δείγματα αντιστρόφως ανάλογη με τη συχνότητα εμφάνισης της κλάσης στο σύνολο εκπαίδευσης στην οποία ανήκουν.

1.7 Τεχνικές για Προεπεξεργασία Δεδομένων Στην Μηχανική Μάθηση

1.7.1 Label Encoding – One Hot Encoding – Ordinal Encoding

Πολλές φορές κάποια από τα χαρακτηριστικά στο σύνολο δεδομένων δεν έχουν αριθμητικές τιμές αλλά ανήκουν σε κατηγορίες καταστάσεων. Επομένως για τη εκπαίδευση των μοντέλων με την χρήση μηχανικής μάθησης απαιτείται η ανάγκη της μετατροπής αυτών των κατηγοριών σε αριθμητικές ή δυαδικές τιμές. Οι πιο συνηθισμένες τεχνικές για την κωδικοποίηση αυτή είναι με Label Encoder, One-Hot Encoding αλλά και Ordinal Encoding.

1.7.1.1 Label Encoding

Στο Label Encoding για κάθε κατηγορηματικό χαρακτηριστικό υπολογίζεται ο αριθμός των διαφορετικών τιμών k που μπορεί να πάρει στο σύνολο δεδομένων. Στη συνέχεια για κάθε μια από τις k διαφορετικές τιμές ανατίθεται ένα νούμερο από το 0 μέχρι k-1, και γίνεται η αντικατάσταση του χαρακτηριστικού αυτού με τις νέες αριθμητικές τιμές. Μια σημαντική λεπτομέρεια είναι πως ένα κατηγορηματικό χαρακτηριστικό μετατρέπεται σε μια στήλη – νέο χαρακτηριστικό με αριθμητικές τιμές χωρίς την αύξηση του αριθμού των στηλών - χαρακτηριστικών στο σύνολο δεδομένων [50].

Παράδειγμα Label Encoder

Έστω ότι έχω το χαρακτηριστικό Χρώμα με πιθανές τιμές Red,Green,Blue και άρα k=3

Τότε θα δημιουργηθεί μια νέα στήλη – χαρακτηριστικό με τιμή 0 για RED , 1 για GREEN και 2 για BLUE

Χρώμα	Label Encoding ➔	**Χρώμα**
RED		0
GREEN		1
BLUE		2
RED		0

1.7.1.2 One Hot Encoding

Στο One-Hot Encoding για κάθε κατηγορηματικό χαρακτηριστικό υπολογίζεται ο αριθμός των διαφορετικών τιμών k ακριβώς όπως και στην περίπτωση του Label Encoding και με βάση αυτό δημιουργούνται k νέες στήλες – χαρακτηριστικά που παίρνουν δυαδικές τιμές. Με αυτή την τεχνική γίνεται η μετατροπή του κατηγορηματικού χαρακτηριστικού όμως αυξάνεται σημαντικά ο αριθμός των χαρακτηριστικών στο σύνολο δεδομένων[51].

Παράδειγμα One-Hot Encoder

Έστω ότι έχω το χαρακτηριστικό Χρώμα με πιθανές τιμές Red,Green,Blue και άρα k=3

Τότε θα δημιουργηθούν 3 νέες στήλες – χαρακτηριστικά, μια για RED , μια για GREEN και μια για BLUE

Χρώμα	One-Hot Encoding ➔	RED	GREEN	BLUE
RED		1	0	0
GREEN		0	1	0
BLUE		0	0	1
RED		1	0	0

1.7.1.3 Ordinal Encoding

Η τεχνική Ordinal Encoding [52] έχει αρκετές ομοιότητες με το Label Encoding. Αφού πρώτα εντοπίζονται οι k διαφορετικές τιμές που είναι πιθανό να πάρει ένα κατηγορηματικό χαρακτηριστικό, στη συνέχεια για κάθε μια από τις k διαφορετικές τιμές ανατίθεται ένα νούμερο από το 0 μέχρι k-1και σε κάθε εγγραφή γίνεται η αντικατάσταση του χαρακτηριστικού αυτού με τις νέες αριθμητικές τιμές. Επομένως ένα κατηγορηματικό χαρακτηριστικό μετατρέπεται σε μια νέα στήλη που αποτελείται από αριθμητικές τιμές. Η κύρια διαφορά σε σχέση με το Label Encoding είναι πως διατηρείται κάποιου είδους πληροφορίας, καθώς η οποιαδήποτε πληροφορία υπήρχε στην κατηγορηματική λίστα διατηρείται και μετά την μετατροπή της σε αριθμητικές τιμές.

Παράδειγμα Ordinal Encoder

Size	Ordinal Encoding ➔	Size
“Small”		0
“Medium”		1
“Large”		2
“Extra Large”		3

Στο παράδειγμα το αρχικό χαρακτηριστικό Size είχε 4 πιθανές κατηγορηματικές τιμές ανάλογα με το μέγεθος της φανέλας. Με τη χρήση του Ordinal Encoding όσο πιο μεγάλο ήταν το μέγεθος της φανέλας, τόσο πιο μεγάλη η αριθμητική τιμή που παίρνει μετά την κωδικοποίηση. Έτσι η νέα σειρά μας δίνει μια επιπλέον πληροφορία και η είσοδος είναι κατάλληλα επεξεργασμένη για εκπαίδευση.

1.7.2 Scaling

Οι αριθμητικές τιμές στα χαρακτηριστικά ενός συνόλου δεδομένων, μπορεί να κυμαίνονται σέ ένα πολύ μεγάλο εύρος, πιθανόν και ως το άπειρο. Με τον όρο Scaling [53] εννοείται η διαδικασία η οποία ακολουθείται έτσι ώστε να μετατραπούν τα αριθμητικά χαρακτηριστικά ενός συνόλου δεδομένου από τυχαία, σε ένα προκαθορισμένο εύρος τιμών. Καθώς όλα τα αριθμητικά χαρακτηριστικά έχουν παρόμοιες τιμές, εξασφαλίζεται πως όλα συμβάλουν ισόνομα στην εκπαίδευση κάτι το οποίο αυξάνει την τελική απόδοση του μοντέλου. Έτσι μειώνεται όσο το δυνατό περισσότερο η προκατάληψη – bias προς ως

προς ένα χαρακτηριστικό το οποίο αν είχε πολύ μεγάλες τιμές, είναι πιθανό η τελική πρόβλεψη του μοντέλου να βασιζόταν στη τιμή συγκεκριμένου πεδίου.

Οι δύο πιο διαδεδομένες Scaling τεχνικές είναι οι εξής:

1.Normalization

Με την τεχνική Normalization[54], οι τιμές ενός χαρακτηριστικού μετατρέπονται σε ένα εύρος από 0 ως 1. Η μικρότερη πιθανή τιμή που παίρνει το χαρακτηριστικό θα έχει ως νέα τιμή την τιμή 0, ενώ η μεγαλύτερη πιθανή τιμή που παίρνει θα έχει ως νέα τιμή την 1. Όλες οι υπόλοιπες τιμές μετατρέπονται στο εύρος 0-1 με βάση τον τύπο:

$$\text{Normalized Value } z_i = \frac{\text{Old Value } x_i - \text{Min(x)}}{\text{Max(x)} - \text{Min(x)}}$$

Η κατανομή της Normalized μορφής θα είναι ακριβώς η ίδια με την κατανομή του αρχικού συνόλου.

2.Standardization

Η τεχνική Standardization[55] έχει ως στόχο την μετατροπή του συνόλου δεδομένων από την μορφή που έχει, σε ένα νέο σύνολο με μέση τιμή μ=0 και διασπορά σ=1. Η μετατροπή αυτή γίνεται από τον εξής τύπο:

$$Standardized\ Value\ z_i = \frac{Old\ Value\ x_i - \mu}{\sigma}$$

Όπου μ η μέση τιμή στο αρχικό σύνολο δεδομένων και σ η διασπορά του.

1.8 Μετρικές Αξιολόγησης Πρόβλεψης

1.8.1 Μετρικές Αξιολόγησης Πρόβλεψης σε Προβλήματα Ταξινόμησης – Classification Problems

Στα προβλήματα ταξινόμησης – Classification η αξιολόγηση της απόδοσης που παρουσιάζει το μοντέλο που έχει εκπαιδευτεί δίνεται με βάση των πιο κάτω μετρικών αξιολόγησης. Από ένα σύνολο δεδομένων ελέγχου μπορεί να υπολογιστεί ο αριθμός των προβλέψεων που κατέληξαν ως **True Positive(TP), True Negative(TN),False Positive(FP)** και **False Negative(FN)**. Ανάλογα με τις ανάγκες του συστήματος γίνεται η επιλογή της μετρικής ή των μετρικών αξιολόγησης που θέλει ο χρήστης να έχουν ψηλά ποσοστά έτσι ώστε να ελέγξει την απόδοση του συστήματος του.

Έστω ότι έχω 2 κλάσεις εξόδου, την κλάση 0-Negative και την κλάση 1-Positive:

- TP-True Positive είναι όσα παραδείγματα ταξινομήθηκαν σωστά στην κλάση 1-Positive

- FN-False Negative είναι όσα παραδείγματα ταξινομήθηκαν λάθος στην κλάση 0-Negative, ενώ ανήκουν στην κλάση 1-Positive
- FP-False Positive είναι όσα παραδείγματα ταξινομήθηκαν λάθος στην κλάση 1-Positive, ενώ ανήκουν στην κλάση 0-Negative
- TN-True Negative είναι όσα παραδείγματα ταξινομήθηκαν σωστά στην κλάση 0-Negative

Κάποιες από τις μετρικές είναι [56],[57]:

i. $\textbf{Ορθότητα} - \textbf{Accuracy} = \frac{\text{Total Correct Predictions}}{\text{Total Predictions}} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}}$

ii. $\textbf{Ακρίβεια} - \textbf{Precision} = \frac{\text{TP}}{\text{TP+FP}}$

iii. $\textbf{Ανάκληση} - \textbf{Recall} = \frac{\text{TP}}{\text{TP+FN}}$

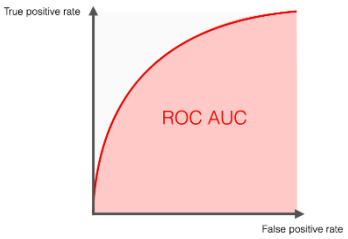
iv. $\textbf{F1 Score } (\boldsymbol{\beta = 1}) = \frac{(1+\beta^2)*\left(\frac{\text{TP}}{\text{TP+FP}}\right)*\left(\frac{\text{TP}}{\text{TP+FN}}\right)}{\beta^2*\left(\frac{\text{TP}}{\text{TP+FP}}\right)+\left(\frac{\text{TP}}{\text{TP+FN}}\right)} = \frac{2*\text{Precision}*\text{Recall}}{\text{Precision+Recall}}$

v. $\textbf{F2 Score } (\boldsymbol{\beta = 2}) = \frac{(1+\beta^2)*\left(\frac{\text{TP}}{\text{TP+FP}}\right)*\left(\frac{\text{TP}}{\text{TP+FN}}\right)}{\beta^2*\left(\frac{\text{TP}}{\text{TP+FP}}\right)+\left(\frac{\text{TP}}{\text{TP+FN}}\right)} = \frac{5*\text{Precision}*\text{Recall}}{4*\text{Precision+Recall}}$

vi. **AUC Score**

$$\text{True Positive Rate: TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

$$\text{False Positive Rate: FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}$$



Σχήμα 1.8.1.1: Καμπύλη AUC Score [58]

1.8.2 Μετρικές Αξιολόγησης Πρόβλεψης σε Προβλήματα Παλινδρόμησης – Regression Problems

Σε προβλήματα παλινδρόμησης – Regression η αξιολόγηση της απόδοσης γίνεται με μετρικές που αλλάζουν ελάχιστα ως προς τον τρόπο μέτρησης, αλλά όλες εστιάζουν και έχουν ως κύριο στόχο την ελαχιστοποίηση της «απόστασης» της πρόβλεψης μιας εισόδου συγκριτικά με το πραγματικό αποτέλεσμα του.

Κάποιες από τις μετρικές που μπορούν να υπολογιστούν είναι [59],[60],[61]:

i. $\boldsymbol{Mean\ Square\ Error - MSE}$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2$$

ii. $\boldsymbol{Root\ Mean\ Square\ Error - RMSE}$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}$$

iii. $\boldsymbol{Mean\ Absolute\ Error - MAE}$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - \hat{x}_i|$$

iv. $\boldsymbol{Normalized\ Mean\ Absolute\ Error - NMAE}$

$$NMAE = \frac{\sum_{i=1}^{N}|x_i - \hat{x}_i|}{\sum_{i=1}^{N}|x_i|}$$

v. $\boldsymbol{Spatial\ Relative\ Error - SRE}$

$$SRE(i) = \frac{\sqrt{\sum_{t=1}^{T}(x_t - \hat{x}_t)^2}}{\sqrt{\sum_{t=1}^{T}(x_t)^2}}$$

vi. $\boldsymbol{Temporal\ Relative\ Error - TRE}$

$$TRE(t) = \frac{\sqrt{\sum_{i=1}^{p}(x_t(i) - \hat{x}_t(i))^2}}{\sqrt{\sum_{i=1}^{p}(x_t(i))^2}}$$

Όπου

$\boldsymbol{\hat{x}_i} = Η\ πρόβλεψη\ για\ την\ τιμή\ της\ εισόδου\ xi$

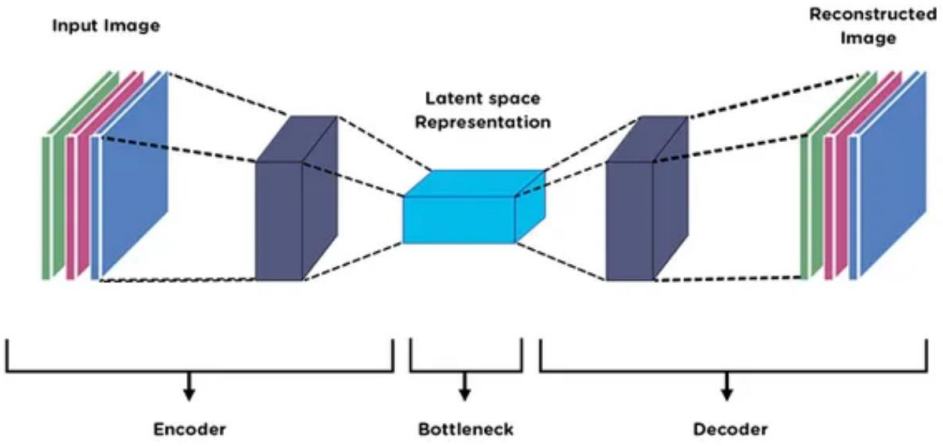
$\boldsymbol{x_i} = Η\ πραγματική\ τιμή\ της\ εισοδου\ xi$

1.9 Autoencoders και Μείωση Διαστάσεων - Dimensionality Reduction

1.9.1 Autoencoders

Οι Autoencoders [62] (σχήμα 1.9.1) αποτελούν ένα είδος αρχιτεκτονικής νευρωνικού δικτύου. Αναπτύχθηκαν με τέτοιο τρόπο έτσι ώστε από έναν αρχικό όγκο δεδομένων να επιτυγχάνεται η συμπίεση και η συλλογή μόνο της σημαντικής πληροφορίας. Η διαδικασία που ακολουθείται είναι πως ο όγκος των δεδομένων εισόδου πρέπει να συμπιέζεται αποδοτικά μέσω ενός Κωδικοποιητή - Encoder και στη συνέχεια με τη χρήση ενός Αποκωδικοποιητή – Decoder να αναπαραχθεί ως έξοδος, μέσω της συμπιεσμένης μορφής, η αρχική μορφή των δεδομένων εισόδου. Αν το μοντέλο λειτουργεί ιδανικά, τότε η συμπιεσμένη μορφή των δεδομένων από την έξοδο του Encoder παρουσιάζει σε ένα πολύ μικρότερο όγκο δεδομένων, όλη την πληροφορία που είχε και το αρχικό σύνολο.

Χρησιμοποιώντας μη επιβλεπόμενη μηχανική μάθηση – Unsupervised Learning οι autoencoders εκπαιδεύονται με στόχο να ανακαλυφθούν κρυφές ή τυχαίοι συνδυασμοί μεταβλητών που παρόλο που στο αρχικό σύνολο δεδομένων δεν μπορούν να παρατηρηθούν, εντούτοις καθορίζουν σε ένα τεράστιο βαθμό τον τρόπο με τον οποίο τα δεδομένα κατανέμονται. Αυτές οι κρυφές μεταβλητές συλλέγονται και παρουσιάζονται στο μεσαίο στρώμα του autoencoder, που είναι γνωστό ως “Latent Space. την διάρκεια της εκπαίδευσης ο autoencoder μαθαίνει ποιες από τις κρυφές μεταβλητές μπορούν να χρησιμοποιηθούν έτσι ώστε να κατασκευαστεί ξανά, μετά από τη συμπίεση, το αρχικό σύνολο δεδομένων. Το latent space που είναι γνωστό και ως “bottleneck” αναπαριστά, με τη χρήση πολύ λιγότερων μεταβλητών - νευρώνων σε σχέση με τα υπόλοιπα στρώματα, μόνο την πιο σημαντική και αναγκαία πληροφορία που περιέχεται στο αρχικό σύνολο δεδομένων.



Σχήμα 1.9.1 : Δομή ενός Autoencoder : Το αρχικό σύνολο δεδομένων (image) κωδικοποιείται έτσι ώστε να συμπιεστεί όσο καλύτερα στο latent space και στη συνέχεια αποκωδικοποιείται για να ξαναφτιαχτεί η είσοδος (Reconstructed Image) [63]

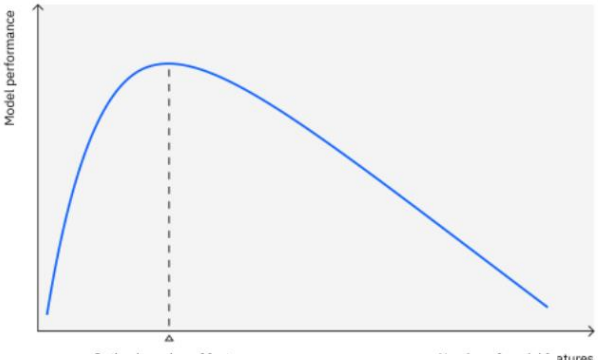
Κάποιες από τις περιπτώσεις χρήσης των Autoencoders είναι η **Μείωση Διαστάσεων – Dimensionality Reduction ,** η Συμπίεση Εικόνας – Image Compression , το Image Denoising,η Δημιουργία Εικόνας – Image Generation και η Εξαγωγή Χαρακτηριστικών – Feature Extraction. [64]

1.9.2 Μείωση Διαστάσεων - Dimensionality Reduction

Με τον όρο Dimensionality Reduction [65] αναφερόμαστε στην μεθοδολογία με την οποία ένα δεδομένο σύνολο δεδομένων αναπαρίσταται με τη χρήση μικρότερου αριθμού χαρακτηριστικών (δηλαδή διαστάσεων), χωρίς όμως αυτό να επηρεάζει την ποιότητα των δεδομένων, καθώς σχεδόν όλη η σημαντική πληροφορία θα περιέχεται και στο μικρότερο σύνολο. Παρόλο που υπάρχει μια πληθώρα υλοποιήσεων ως προς την τεχνική που ακολουθείται για τον περιορισμό των διαστάσεων, όλες έχουν ως στόχο την προεπεξεργασία των δεδομένων κατάλληλα έτσι ώστε να περιορίσουν των όγκο των δεδομένων που θα εισέρχεται σε ένα μοντέλο προς εκπαίδευση.

Στην μηχανική μάθηση, η λίστα με τα χαρακτηριστικά αποτελούν τις παραμέτρους που θα καθορίσουν τον τρόπο εκπαίδευσης, την πρόβλεψη και της έξοδο ενός μοντέλου. Τα σύνολα δεδομένων πολλών διαστάσεων δεν είναι τα ιδανικά και εμφανίζουν προβλήματα σε αλγόριθμους μηχανικής μάθησης. Ένα βασικό θέμα είναι πως εξαιτίας των πολλών διαφορετικών παραμέτρων, ο χρόνος εκπαίδευσης είναι σημαντικά μεγαλύτερος συγκριτικά με ένα σύνολο δεδομένων μικρότερων διαστάσεων όπως και του ότι ο όγκος των δεδομένων προκαλεί ζητήματα ως προς τον τρόπο αποθήκευσης τους. Επίσης σε όλο αυτό το σύνολο είναι απίθανο να μην υπάρχει περιττή πληροφορία, οπότε σε μοντέλα μεγάλων διαστάσεων οι ταξινομητές δεν εκπαιδεύονται βέλτιστα και συνήθως παρουσιάζεται μειωμένη απόδοση

Η τελευταία πρόταση είναι κάτι που απασχολεί την επιστημονική κοινότητα. Λόγω της περιττής πληροφορίας και της αυξημένης πολυπλοκότητας, σε περίπτωση που ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον ιδανικό, η απόδοση του μοντέλου υστερεί συγκριτικά με τα πιο απλά μοντέλα. Όμως, στην περίπτωση που ο αριθμός των χαρακτηριστικών μειωθεί υπερβολικά τότε υπάρχει ο κίνδυνος της απώλειας σημαντικής πληροφορίας κάτι που ρίχνει κατακόρυφα την απόδοση του ταξινομητή. Οπότε το ζήτημα είναι να εντοπιστεί ο βέλτιστος αριθμός χαρακτηριστικών έτσι ώστε να μην χαθεί η αναγκαία πληροφορία αλλά ούτε και να υπάρχει πλεονασμός, με στόχο



Σχήμα 1.9.2. Απόδοση Μοντέλου ανάλογα με τον Αριθμό Χαρακτηριστικών – Διαστάσεων [65]

την καλύτερη δυνατή απόδοση. (Σχήμα 1.9.2)

1.9.3 Dimensionality Reduction με Autoencoders

Όπως έχει αναφερθεί προηγουμένως, υπάρχουν πολλές διαφορετικές προσεγγίσεις ως προς την μεθοδολογία για τον περιορισμό των διαστάσεων ενός συνόλου δεδομένων. Μια από αυτές είναι μέσω της χρήσης μη επιβλεπόμενης μηχανικής μάθησης και ειδικότερα των autoencoders, όπως για παράδειγμα η μελέτη των Wang, Yasi & Yao, Hongxun & Zhao, Sicheng [66].

Ο κύριος στόχος στην μείωση των διαστάσεων είναι από ένα αρχικό μεγάλο σύνολο δεδομένων, να περιοριστεί ο αριθμός των χαρακτηριστικών με την ελάχιστη δυνατή απώλεια πληροφορίας. Ένας autoencoder δέχεται ένα σύνολο δεδομένων και προσπαθεί μέσω επαναληπτικών διασχίσεων στο νευρωνικό δίκτυο που είχε κατασκευαστεί , να εντοπιστούν τα ιδανικά βάρη σε κάθε νευρώνα έτσι ώστε να συμπιεστούν τα δεδομένα όσο καλύτερα και στη συνέχεια να ξανακατασκευαστεί το αρχικό σύνολο. Αυτό το συμπιεσμένο σύνολο στο μεσαίο στρώμα του autoencoder εκφράζει το αρχικό σύνολο που δέχθηκε ως είσοδο αλλά σε μικρότερη διάσταση. Ο συνδυασμός των 2 παρουσιάζει μεγάλη προοπτική και είναι κάτι που θα μας απασχολήσει στο πειραματικό μέρος

Έστω πως υπάρχει ένα σύνολο δεδομένων το οποίο αποτελείται από μια μεγάλη λίστα χαρακτηριστικών στην οποία προφανώς υπάρχει αρκετή περιττή πληροφορία. Αν κατά την προεπεξεργασία των δεδομένων δεν υλοποιηθεί κάποιος αλγόριθμος περιορισμού διαστάσεων, τότε κατά της διάρκεια της εκπαίδευσης ενός μοντέλου ο χρόνος εκπαίδευσης θα είναι χαρακτηριστικά μεγαλύτερος από τον ιδανικό και ίσως ο ταξινομητής να καταλήξει στο να έχει τελικά μικρότερη απόδοση από την αναμενόμενη. Αν όμως πριν την εκπαίδευση του μοντέλου κατά την προεπεξεργασία των δεδομένων γίνει εισαγωγή ενός autoencoder, τότε θα προκύψουν διαφορετικά αποτελέσματα. Το αρχικό σύνολο δεδομένων θα αποτελεί την είσοδο του autoencoder, τον οποίο autoencoder θα εκπαιδεύσουμε για ένα επιθυμητό αριθμό εποχών. Όπως είναι γνωστό ο autoencoder αποτελείται από έναν encoder που καταλήγει στο latent space και είναι συνδεδεμένος με έναν decoder που αναπαράγει το αρχικό σύνολο. Όταν εκπαιδευτεί ο autoencoder, μπορεί να αξιοποιηθεί μόνο το πρώτο μέρος του, δηλαδή ο encoder, έτσι ώστε το αρχικό σύνολο δεδομένων να περάσει μέσα από τον κωδικοποιητή και να συμπιεστούν τα δεδομένα. Με αυτό τον τρόπο πετύχαμε μείωση των διαστάσεων από το αρχικό σύνολο, σε ένα νέο σύνολο χαρακτηριστικών με διαστάσεις όσο αριθμό νευρώνων υπάρχει στο latent space. (Κεφάλαιο 3.6, Πείραμα 5)

Κεφάλαιο 2: Προηγούμενες Μελέτες

Η ραγδαία ανάπτυξη του τομέα της μηχανικής μάθησης έχει ήδη οδηγήσει σε μια πληθώρα μελετών στο κομμάτι της αξιοποίησης ενός τέτοιου εργαλείου για τον εντοπισμό πιθανής ανώμαλης κίνησης σε ένα δίκτυο. Κάθε μελέτη βασίζεται σε διαφορετική φιλοσοφία ως προς τον τρόπο εκπαίδευσης (Supervised Learning, Unsupervised Learning κλπ) και σε ένα διαφορετικό αρχικό σύνολο δεδομένων . Όπως είναι γνωστό, για την εκπαίδευση ενός μοντέλου χρειάζεται ένα σύνολο δεδομένων με επαρκή αριθμό εγγραφών το οποίο είτε είναι δεδομένα πραγματικής κίνησης από κάποιο δίκτυο, είτε έχει κατασκευαστεί συνθετικά έτσι ώστε να υπάρχουν εγγραφές τόσο από ομαλή κίνηση όσο και από ανωμαλίες. Ένα τέτοιο σύνολο δεδομένων είναι και το UNSW-15 Dataset [67],[68],[69],[70],[71],[72](2.1), που αποτελεί ένα από τα πιο πρόσφατα σύνολα δεδομένων που έχουν δημιουργηθεί και είναι πάνω στο οποίο θα βασιστεί η πειραματική διαδικασία. Ενδεικτικά κάποια άλλα σύνολα δεδομένων που παρουσιάζουν παρόμοια χαρακτηριστικά με το UNSW-15 Dataset είναι τα KDD CUP 99[73],NSL-KDD[74] και CICIDS 2017 [75]

2.1 UNSW-NB15 Dataset

Το συγκεκριμένο Dataset έχει δημιουργηθεί από μια ομάδα ερευνητών στο εργαστήριο Cyber Range Lab του πανεπιστημίου UNSW Canberra στην Αυστραλία το 2015. Με την χρήση του εργαλείου IXIA PerfectStorm καταγράφηκαν 100gb κίνησης, μέσα από ένα συνδυασμό πραγματικής ομαλής κίνησης σε ένα δίκτυο μαζί με συνθετικές συμπεριφορές πιθανής επίθεσης. Κύριος στόχος είναι ήταν η δημιουργία ενός αξιόπιστου συνόλου δεδομένων έτσι ώστε να μπορούν να κατασκευαστούν μελλοντικά καλύτερα συστήματα ανίχνευσης εισβολής σε δίκτυο - Network Intrusion Detection System (NIDS).

Το Dataset έχει δεδομένα κίνησης πακέτων από 9 είδη διαφορετικών επιθέσεων μαζί με Ομαλή Κίνηση, η οποία αποτελεί την πλειοψηφία των εγγραφών στο σύνολο δεδομένων. Υπάρχουν συνολικά 49 χαρακτηριστικά – features εκ των οποίων το 1 είναι η ετικέτα-Label με πιθανή τιμή 1 για τις εγγραφές οι οποίες αποτελούν επιθέσεις και 0 για τις περιπτώσεις που υπάρχει ομαλή κίνηση.

2.1.1. Σύνολο Δεδομένων στο UNSW-NB15 Dataset

Το σύνολο δεδομένων αποτελείται συνολικά από 2.540.044 εγγραφές κατανεμημένα σε 4 csv files. Επίσης υπάρχει ξεχωριστό training set και test set με 175,341 και 82,332 εγγραφές αντίστοιχα.

Πίνακας 2.1.1: Αριθμός Εγγραφών σε κάθε Dataset μαζί με την κατηγορία στην οποία ανήκουν

Τύπος Επίθεσης	Αριθμός Εγγραφών PARTIAL DATASET ¼ (A)	Αριθμός Εγγραφών PARTIAL DATASET ¼ (B)	Αριθμός Εγγραφών PARTIAL DATASET ¼ (Γ)	Αριθμός Εγγραφών PARTIAL DATASET ¼ (Δ)	Αριθμός Εγγραφών FULL DATASET
Normal	677786	647252	542676	351150	2218764
Fuzzers	5051	4668	9137	5390	24246
Analysis	526	608	873	670	2677
Backdoors	534	370	759	666	2329
DoS	1167	4637	5642	4907	16353
Exploits	5409	11103	16574	11439	44525
Generic	7522	27883	118198	61878	215481
Reconnaissance	1759	3116	5582	3530	13987
Shellcode	223	324	593	371	1511
Worms	24	40	67	43	174
Ποσοστό Ομαλής Κίνησης/Ανωμαλίας	96.82% - 3.18%	92.46% - 7.54%	77.53% - 22.47%	79.80% - 20.20%	87.35% - 12.65%
Σύνολο	700001	700001	700001	440044	2540047

2.1.2 Λίστα των χαρακτηριστικών – Features στο UNSW-NB15 Dataset:

Το συγκεκριμένο Dataset αποτελείται από 49 διαφορετικά είδη χαρακτηριστικών τα οποία είναι τα εξής:

Πίνακας 2.1.2: Λίστα Χαρακτηριστικών στο UNSW-NB15 Dataset

No	Name	Type	Description
1	srcip	nominal	Source IP address
2	sport	integer	Source port number
3	dstip	nominal	Destination IP address
4	dsport	integer	Destination port number
5	proto	nominal	Transaction protocol
6	state	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Record total duration
8	sbytes	Integer	Source to destination transaction bytes
9	dbytes	Integer	Destination to source transaction bytes
10	sttl	Integer	Source to destination time to live value
11	dttl	Integer	Destination to source time to live value
12	sloss	Integer	Source packets retransmitted or dropped
13	dloss	Integer	Destination packets retransmitted or dropped

14	service	nominal	http, ftp, smtp, ssh, dns, ftp-data ,irc and (-) if not much used service
15	Sload	Float	Source bits per second
16	Dload	Float	Destination bits per second
17	Spkts	integer	Source to destination packet count
18	Dpkts	integer	Destination to source packet count
19	swin	integer	Source TCP window advertisement value
20	dwin	integer	Destination TCP window advertisement value
21	stcpb	integer	Source TCP base sequence number
22	dtcpb	integer	Destination TCP base sequence number
23	smeansz	integer	Mean of the ?ow packet size transmitted by the src
24	dmeansz	integer	Mean of the ?ow packet size transmitted by the dst
25	trans_depth	integer	Represents the pipelined depth into the connection of http request/response transaction
26	res_bdy_len	integer	Actual uncompressed content size of the data transferred from the server's http service.
27	Sjit	Float	Source jitter (mSec)
28	Djit	Float	Destination jitter (mSec)
29	Stime	Timestamp	record start time
30	Ltime	Timestamp	record last time
31	Sintpkt	Float	Source interpacket arrival time (mSec)
32	Dintpkt	Float	Destination interpacket arrival time (mSec)
33	tcprtt	Float	TCP connection setup round-trip time, the sum of 'synack' and 'ackdat'.
34	synack	Float	TCP connection setup time, the time between the SYN and the SYN_ACK packets.
35	ackdat	Float	TCP connection setup time, the time between the SYN_ACK and the ACK packets.
36	is_sm_ips_ports	Binary	If source (1) and destination (3)IP addresses equal and port numbers (2)(4) equal then, this variable takes value 1 else 0
37	ct_state_ttl	Integer	No. for each state (6) according to specific range of values for source/destination time to live (10) (11).
38	ct_flw_http_mthd	Integer	No. of flows that has methods such as Get and Post in http service.
39	is_ftp_login	Binary	If the ftp session is accessed by user and password then 1 else 0.
40	ct_ftp_cmd	integer	No of flows that has a command in ftp session.
41	ct_srv_src	integer	No. of connections that contain the same service (14) and source address (1) in 100 connections according to the last time (26).

42	ct_srv_dst	integer	No. of connections that contain the same service (14) and destination address (3) in 100 connections according to the last time (26).
43	ct_dst_ltm	integer	No. of connections of the same destination address (3) in 100 connections according to the last time (26).
44	ct_src_ ltm	integer	No. of connections of the same source address (1) in 100 connections according to the last time (26).
45	ct_src_dport_ltm	integer	No of connections of the same source address (1) and the destination port (4) in 100 connections according to the last time (26).
46	ct_dst_sport_ltm	integer	No of connections of the same destination address (3) and the source port (2) in 100 connections according to the last time (26).
47	ct_dst_src_ltm	integer	No of connections of the same source (1) and the destination (3) address in in 100 connections according to the last time (26).
48	attack_cat	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

2.1.3 Πιθανά Είδη Επιθέσεων στο UNSW-NB15 Dataset:

Εκτός από την ομαλή κίνηση, στο σύνολο δεδομένων υπάρχουν και εγγραφές οι οποίες χαρακτηρίζονται ως επιθέσεις. Όλες οι ανωμαλίες στο δίκτυο μπορούν να κατηγοριοποιηθούν σε 9 κατηγορίες, ανάλογα με το είδος της επίθεσης. Τα πιθανά είδη επιθέσεων που υπάρχουν στο σύνολο δεδομένων είναι τα πιο κάτω:

2.1.3.1 Fuzzers:

Το fuzzer [76] είναι μια τεχνική επίθεσης στον κλάδο της κυβερνο-ασφάλειας όπου ο επιτιθέμενος προσπαθεί χρησιμοποιώντας τα ποιο παράξενα/σπάνια δεδομένα ως είσοδο σε ένα σύστημα ή δίκτυο με στόχο να ανακαλύψει «κενά» και αδυναμίες του συστήματος. Κύριος στόχος είναι να προκαλέσει τη διαρροή δεδομένων, την αποκάλυψη πληροφοριών ή την εκτέλεση κάποιου κακόβουλου κώδικα.

2.1.3.2 Analysis:

Η επίθεση ανάλυσης (Analysis Attack [77]) είναι μια μέθοδος στην οποία ο επιτιθέμενος παρατηρεί ένα δίκτυο έτσι ώστε να επιτύχει την ανάλυση της κυκλοφορίας του και να αποσπάσει σημαντικές πληροφορίες όπως την ανίχνευση σημαντικών κόμβων, την δομή

της δρομολόγησης και πιθανά μοτίβα στην συμπεριφορά. Επομένως, στόχος είναι η ανακάλυψη αδυναμιών μέσα από μελέτη και η προσπάθεια εκμετάλλευσης τους

2.1.3.3 Backdoors:

Η επίθεση από την πίσω πόρτα (Backdoors Attack [78]) αποτελεί μια μορφή κυβερνοεπίθεσης, στην οποία ο επιτιθέμενος εκμεταλλεύεται ευπάθειες σε ένα σύστημα έτσι ώστε να δημιουργήσει μια «κρυφή πόρτα» που του επιτρέπει να παρακάμψει τους διάφορους μηχανισμούς πιστοποίησης και να αποκτήσει παράνομα έλεγχο σε κομμάτια ή ακόμα και σε ολόκληρο το σύστημα. Οι πιο αποτελεσματικές Backdoor επιθέσεις είναι καλά κρυμμένες και δύσκολο να ανιχνευθούν από τους διαχειριστές και τα εργαλεία ασφαλείας που διαθέτουν κάτι το οποίο δίνει την ευκαιρία στον επιτιθέμενο να έχει πρόσβαση και να παρατηρεί για μεγάλο χρονικό διάστημα χωρίς να εντοπιστεί. Κύριος στόχος είναι η κλοπή δεδομένων του συστήματος, η κατανόηση του πως λειτουργεί το σύστημα και η πιθανή κακόβουλη τροποποίηση του

2.1.3.4 DoS:

Η επίθεση DOS (Denial of Service Attack [79]) είναι ένα είδος στην οποία ο επιτιθέμενος έχει στόχο αναστείλει τελείως ή να μειώσει τη διαθεσιμότητα ενός συστήματος, υπηρεσίας ή δικτύου έτσι ώστε να μην είναι προσβάσιμο, αργό ή μη λειτουργικό για τους νόμιμους χρήστες. Ο επιτιθέμενος αποστέλλει μεγάλο όγκο κακόβουλων δεδομένων ή επικοινωνιακών αιτημάτων προς τον στόχο – κόμβο κάτι το οποίο οδηγεί σε υπερφόρτιση του συστήματος που έχει ως αποτέλεσμα την δυσλειτουργία και την αργή εξυπηρέτηση στα αιτήματα των υπόλοιπων χρηστών. Πολλές φορές οι «στόχοι» είναι μεγάλες εταιρίες στις οποίες παρόλο που το ρίσκο για διαρροή δεδομένων είναι χαμηλό, δημιουργούν διακοπές που χρειάζονται χρόνο, προκαλούν μεγάλες οικονομικές ζημιές και κλονίζουν την αξιοπιστία της επιχείρησης στους απλούς χρήστες.

2.1.3.5 Exploits:

Η επίθεση εκμετάλλευσης (Exploit Attack [80]) είναι μια μορφή επίθεσης στην οποία ο επιτιθέμενος εκμεταλλεύεται μια αδυναμία του συστήματος μέσω εξειδικευμένων εργαλείων ή κακόβουλου κώδικα έτσι ώστε να αποκτήσει πρόσβαση σε κομμάτια του συστήματος που δεν θα έπρεπε κανονικά. Κάποιες αδυναμίες σε ένα σύστημα μπορεί να αναφέρονται σε αδυναμίες του λογισμικού ή ίσως κάποια με ενημερωμένα συστήματα. Στόχος του επιτιθέμενου είναι μέσω αυτού να προβεί σε δραστηριότητες όπως η παραβίαση ασφάλειας, η πρόσβαση σε προσωπικά δεδομένα ή εγκατάσταση κακόβουλου λογισμικού (π.χ. Backdoor attack) .

2.1.3.6 Generic:

Η Γενικευμένη επίθεση (Generic Attack [81]) αναφέρεται στην ευρύτερη κατηγορία των επιθέσεων όπου δεν υπάρχει συγκεκριμένη αδυναμία του συστήματος ή συγκεκριμένος στόχος-κόμβος, αλλά μπορεί να λάβει πολλές διαφορετικές μορφές ανάλογα με το περιβάλλον και τις συνθήκες. Μέσω διάφορων μεθόδων και τεχνικών όπως για παράδειγμα Phishing, Malware, Dos κλπ, ο επιτιθέμενος προσπαθεί να εκμεταλλευτεί αδυναμίες του συστήματος με κύριο στόχο να προκαλέσει ζημιά ή να κερδίσει πρόσβαση σε ευαίσθητα δεδομένα χωρίς να απαιτείται ειδική γνώση και εξειδίκευση από τον ίδιο.

2.1.3.7 Reconnaissance:

Στην επίθεση αναγνώρισης (Reconnaissance Attack [82]) ο επιτιθέμενος έχει ως στόχο να εξερευνήσει, δηλαδή να συγκεντρώσει όσο το δυνατό περισσότερα δεδομένα σχετικά με ένα σύστημα, δίκτυο ή οργανισμό έτσι ώστε να εντοπίσει κενά και ευπάθειες για προετοιμασία μελλοντικών επιθέσεων. Η διαδικασία περιλαμβάνει τη σάρωση του δικτύου για ανοικτές θύρες, τη συλλογή πληροφοριών για τα πρωτόκολλα λειτουργίας του συστήματος, πληροφορίες για την υποδομή και τους χρήστες αλλά και τον εντοπισμό πιθανών ευπαθειών του συστήματος. Ουσιαστικά η επίθεση αναγνώρισης αποτελεί ένα εργαλείο προετοιμασίας για την πλήρη κατανόηση του στόχου και την καλύτερη οργάνωση της πραγματικής επίθεσης.

2.1.3.8 Shellcode:

Η επίθεση με χρήση Shellcode [83] αναφέρεται σε μια τεχνική κυβερνοεπίθεσης στην οποία ο επιτιθέμενος με τη βοήθεια ενός συνήθως μικρού εκτελέσιμου προγράμματος αποκτά πρόσβαση στο σύστημα, προσπαθώντας να εκμεταλλευτεί αδυναμίες και να εκτελέσει κακόβουλες εντολές. Τα συγκεκριμένα κομμάτια κώδικα συνήθως ξεκινούν ένα Command Shell (CLI , GUI) και έχει ως στόχο την εκτέλεση εντολών που θα επιτρέψουν στον επιτιθέμενο να αναλάβει τον πλήρη έλεγχο του συστήματος, την εγκατάσταση πίσω πόρτας για μελλοντική πρόσβαση και την κλοπή δεδομένων

2.1.3.9 Worms:

Η επίθεση με Worms [84] επιτυγχάνεται όταν ένα κακόβουλο λογισμικό, γνωστό ως Worm, έχει εισβάλει σε ένα υπολογιστή ενός δικτύου μέσα από αδυναμίες του συστήματος, από Spam Email ή ακόμη και από ένα απλό USB. Τα Worms όπως και ένας απλός ιός μπορεί να προκαλέσουν απώλεια δεδομένων και διακοπή λειτουργίας διάφορων συστημάτων αλλά με την διαφορά συγκριτικά με απλούς ιούς και το οποίο καθιστά την συγκεκριμένη επίθεση επικίνδυνη, είναι ότι από την στιγμή που τα Worms έχουν εισβάλει στο δίκτυο, έχουν την δυνατότητα να αναπαραχθούν και να διαδίδονται αυτόνομα μέσα στους υπολογιστές του δικτύου χωρίς την ανάγκη ανθρώπινης παρέμβασης έτσι ώστε να «μολύνουν» ακόμη περισσότερα μηχανήματα του δικτύου.

2.2 Προηγούμενη Έρευνα στο UNSW-NB15 Dataset με Μοντέλα Επιβλεπόμενης Μηχανικής Μάθησης

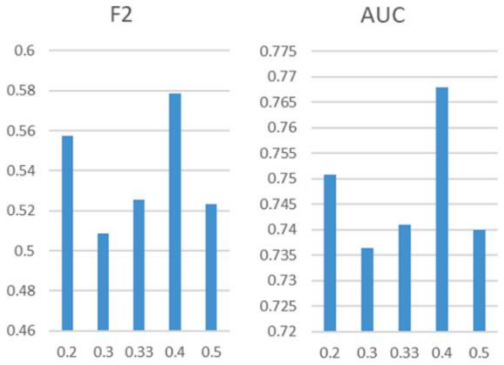
Από την στιγμή της δημιουργίας του UNSW-NB15 Dataset και της παραχώρησης από τους δημιουργούς του για ελεύθερο πειραματισμό από όλους, υπήρξαν πολλές μελέτες με κύριο στόχο της δημιουργία ενός όσο το δυνατό πιο αποδοτικού μοντέλου μηχανικής μάθησης για την ανίχνευση ανωμαλιών.

Μια από αυτές έγινε από τους Igor Fosi, Drago Zagar , Kresimir Grgi και Visnja Krizanovi [85], οι οποίοι ανέπτυξαν μοντέλα επιβλεπόμενης μάθησης και είχαν ως κύρια διαφορά ότι επιλέγηκαν πολύ λιγότερα χαρακτηριστικά συγκριτικά με όλες τις παρόμοιες έρευνες. Από τα 49 χαρακτηριστικά του αρχικού συνόλου δεδομένων κράτησαν μόνο τα 8, όσα χαρακτηριστικά δηλαδή συμβαδίζουν με το πρωτόκολλο NetFlow[86] της Cisco, ένα πρωτόκολλο με συγκεκριμένη δομή για την συλλογή της κίνησης της πληροφορίας σε ένα δίκτυο. Επειδή το νέο σύνολο δεδομένων είναι πολύ μικρότερο και διαφορετικό από το αρχικό, η μελέτη τους βασίστηκε στο να πάρουν μια σειρά από μετρήσεις έτσι ώστε να εντοπίσουν παραμέτρους που βελτιστοποιούν το αποτέλεσμα. Μετά από την προεπεξεργασία των δεδομένων εκπαίδευσαν μοντέλα τα οποία αξιολογούσαν με βάση το F2 Score και το AUC Score. Στην συγκεκριμένη περίπτωση καθώς αναφερόμαστε σε ανωμαλίες, ο στόχος ιδανικά είναι να μην υπάρχει πρόβλεψη που να κατατάσσεται ως False Negative, δηλαδή ο ταξινομητής να προβλέψει λάθος μια εγγραφή που αποτελεί επίθεση ως ομαλή κίνηση. Όσο πιο ψηλή η τιμή των μετρικών F2 και AUC Score, τόσο πιο λίγες False Negative προβλέψεις υπάρχουν. Η κεντρική ιδέα τους ήταν να εντοπίσουν ποιο μοντέλο επιβλεπόμενης μηχανικής μάθησης εμφανίζει την πιο ψηλή απόδοση στις μετρικές, ποιο είναι το ιδανικό Train/Test Ratio για το νέο σύνολο δεδομένων που έχουν φτιάξει και ποια από τις τεχνικές Label Encoding – One Hot Encoding είναι η προτιμότερη σε αυτή την περίπτωση για τα κατηγορηματικά χαρακτηριστικά.

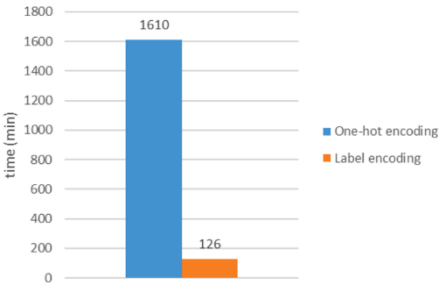
Στην πειραματική του διεργασία πήραν μετρήσεις από 7 μοντέλα επιβλεπόμενης μηχανικής μάθησης τα οποία εκπαιδεύτηκαν μια φορά με τη χρήση του Label Encoder και μια φορά με One Hot Encoder για σκοπούς σύγκρισης. Όλο αυτό έγινε 5 φορές για διαφορετικά Train/Test Ratio. Όταν υπάρχει ένα σύνολο δεδομένων, το Train/Test Ratio δηλώνει το ποσοστό των εγγραφών του συνόλου δεδομένων το οποίο θα χρησιμοποιηθεί για την εκπαίδευση του μοντέλου και το ποσοστό των εγγραφών το οποίο θα αποτελέσει το κριτήριο για την μέτρηση της απόδοσης του ταξινομητή που έχει εκπαιδευτεί. Οι αναλογίες που ελέγχθηκαν είναι το 0.8-0.2 , 0.7-0.3 , 0.666-0.333, 0.6-0.4, 0.5-0.5.

Τα αποτελέσματα των μετρήσεων έδειξαν ότι για Train/Test Ratio 60/40 (0.6-0.4) η μέση τιμή της απόδοσης είναι η καλύτερη τόσο για το F2 Score όσο και για το AUC Score (Σχήμα 2.2.1) . Επίσης όπως είχαν παρατηρήσει, η χρήση του One Hot Encoder για την κατηγοριοποίηση των χαρακτηριστικών απαιτεί 12-13 φορές περισσότερο χρόνο στην διαδικασία εκπαίδευσης του μοντέλου σε σύγκριση με το Label Encoder (Σχήμα 2.2.2).

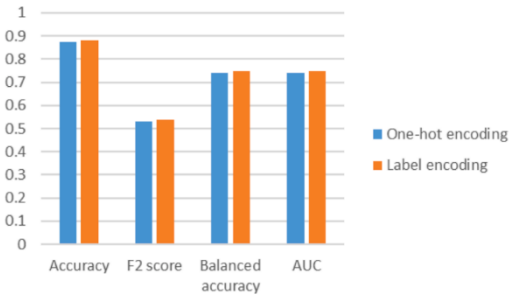
Παρόλο που ο χρόνος του πρώτου είναι σημαντικά μεγαλύτερος, η απόδοση των δύο τεχνικών είναι πανομοιότυπη και ίσως και λίγο καλύτερη σε κάποιες μετρικές με τη χρήση του Label Encoder (Σχήμα 2.2.3). Επομένως κατέληξαν στο συμπέρασμα ότι το Label Encoding είναι προτιμότερο από το One Hot Encoding, όχι λόγω απόδοσης αλλά λόγω του χρόνου εκπαίδευσης των μοντέλων. Τα υποψήφια μοντέλα επιβλεπόμενης μηχανικής μάθησης ήταν 7, ο **SGD** (Stochastic Gradient Descent) [87] , **SVC** (1.5.1), **KNN** (1.5.3), **GNB** (Gaussian Naïve Bayes) [88], **Decision Tree** (1.5.4) , **Random Forest** (1.5.5) και ο **AB** (AdaBoost)[89] ταξινομητής. Η πειραματική διαδικασία έδειξε πως από όλα αυτά τα μοντέλα, το Random Forest ήταν ο ταξινομητής που είχε την καλύτερη απόδοση σε μετρικές Accuracy , F2 Score και AUC Score και άρα το θεώρησαν ως την καλύτερη επιλογή για μελλοντικές έρευνες.



Σχήμα 2.2.1: Απόδοση F2 Score και AUC Score στην Έρευνα 2.2 για διαφορετικές τιμές στο Train/Test Ratio



Σχήμα 2.2.2: Σύγκριση συνολικού χρόνου εκπαίδευσης όλης της πειραματικής διαδικασίας ανάλογα με τη χρήση Label Encoding και One Hot Encoding



Σχήμα 2.2.3: Μέση απόδοση μετρικών στην Έρευνα 2.2 με τη χρήση Label Encoding και One Hot Encoding

Κεφάλαιο 3: Πειραματική Διαδικασία

3.1 Προεπεξεργασία Δεδομένων

Κύριος στόχος στο στάδιο προεπεξεργασίας δεδομένων είναι να ακολουθηθεί μια σειρά από τα κατάλληλα βήματα, έτσι ώστε από ένα αρχικό ακατέργαστο σύνολο δεδομένων, τελικά να καταλήξουμε σε ένα ουσιαστικά καινούριο σύνολο δεδομένων που θα αποτελέσει τη βάση στην οποία θα εκπαιδευτούν όλα τα μοντέλα στη συνέχεια της πειραματικής διαδικασίας. Το αρχικό σύνολο δεδομένων μπορεί να περιέχει περιττή πληροφορία, προβληματικές εγγραφές, κενά πεδία και γενικότερα μια ανομοιομορφία μεταξύ των διαφόρων τιμών που αν δεν τύχουν επεξεργασίας, η εισαγωγή ενός μοντέλου στην πορεία δεν θα δώσει τα επιθυμητά αποτελέσματα κατά την διάρκεια της εκπαίδευσης. Όπως έχει αναφερθεί προηγουμένως, για το UNSW-NB15 Dataset υπάρχουν 4 αρχεία csv. Το πρώτο βήμα είναι η ανάγνωση των τεσσάρων αυτών αρχείων ξεχωριστά έτσι ώστε να δημιουργηθούν 4 μικρότερα υποσύνολα δεδομένων (Partial Datasets A,B,C,D) και ταυτόχρονα, η συνένωση όλων των εγγραφών από όλα τα αρχεία για να σχηματιστεί ένα μεγάλο ολοκληρωμένο σύνολο δεδομένων (Full Dataset). Εφόσον πλέον υπάρχουν τα 5 σύνολα δεδομένων, τώρα ακολουθεί η επεξεργασία τους έτσι ώστε να φτάσουν σε μια μορφή που θα επιτρέψει την αποδοτική εκπαίδευση των μοντέλων. Από τα 49 αρχικά χαρακτηριστικά-Features έχουν επιλεγεί και διατηρηθεί τα 8 τα οποία είχαν επιλεγεί στην μελέτη 2.2 όπως και επίσης το 48.attack_cat που παρουσιάζει το είδος της επίθεσης και είναι χρήσιμο στις μετρήσεις για την εξαγωγή στατιστικών στοιχείων(πίνακας 3.1.1).

Πίνακας 3.1.1: Χαρακτηριστικά που απέμειναν από το αρχικό σύνολο δεδομένων

No	Name	Type	Description
2	Sport	integer	Source port number
4	Dsport	integer	Destination port number
5	Proto	nominal	Transaction protocol
6	State	nominal	Indicates to the state and its dependent protocol, e.g. ACC, CLO, CON, ECO, ECR, FIN, INT, MAS, PAR, REQ, RST, TST, TXD, URH, URN, and (-) (if not used state)
7	dur	Float	Record total duration
8	sbytes	Integer	Source to destination transaction bytes
17	Spkts	integer	Source to destination packet count
48	attack_cat	nominal	The name of each attack category. In this data set , nine categories e.g. Fuzzers, Analysis, Backdoors, DoS Exploits, Generic, Reconnaissance, Shellcode and Worms
49	Label	binary	0 for normal and 1 for attack records

Στη συνέχεια έγινε έλεγχος για τιμές που λείπουν ή τιμές προβληματικές που δεν θα επιτρέψουν την εκπαίδευση του μοντέλου και θα προκαλέσουν σφάλμα. Από τον έλεγχο διαπιστώθηκε πως για το χαρακτηριστικό 2.sport (source port) και 4.dsport (destination port) υπήρχαν εγγραφές με προβληματικές τιμές. Συγκεκριμένα έχουν εντοπιστεί 308 προβληματικές τιμές στα Partial Datasets οι οποίες υπάρχουν και στο Full Dataset (πίνακας 3.1.2).:

Πίνακας 3.1.2: Λίστα Προβληματικών εγγραφών στα σύνολα δεδομένων

PARTIAL Dataset A/B/C/D	Χαρακτηριστικό sport/dsport	Τιμή	Αριθμός Εγγραφών
A	sport	0x000c	4
A	sport	0x000b	2
A	sport	-	2
A	dsport	0xc0a8	53
A	dsport	0x20205321	1
A	dsport	-	5
B	dsport	0xcc09	61
C	dsport	0xcc09	105
D	dsport	0xcc09	75

Υπάρχουν κάποιες τιμές που με τροποποίηση μπορούν να μετατραπούν σε φυσιολογικές όπως για παράδειγμα το destination port 0xcc09 στο δεκαεξαδικό σύστημα που μπορεί να μετατραπεί σε 52233 στο δεκαδικό, το οποίο και αποτελεί μια πραγματική πόρτα. Όμως στην έρευνα 2.2 αυτές τις 308 εγγραφές έχουν διαγραφεί, εξηγώντας ότι είναι πολύ λίγες οι περιπτώσεις και πως αποτελούν μόνο εγγραφές από ομαλή κίνηση. Επομένως, για να υπάρξει ακριβώς το ίδιο σύνολο δεδομένων και σε αυτή την περίπτωση, αυτά τα 308 δείγματα έχουν αφαιρεθεί και από τα Partial Datasets αλλά και από το Full Dataset. Έπειτα, στη νέα λίστα των χαρακτηριστικών υπάρχουν 2 τα οποία δεν έχουν αριθμητικές τιμές αλλά χωρίζονται σε κατηγορίες καταστάσεων. Αυτά τα 2 χαρακτηριστικά είναι το 5.proto και 6.state και τα οποία αποτελούνται από 16 (π.χ. tcp,udp) και 134 (π.χ. FIN,CON,INT) διαφορετικές κατηγορίες αντίστοιχα. Για την διαδικασία της εκπαίδευσης μοντέλων με την χρήση μηχανικής μάθησης απαιτείται η ανάγκη της μετατροπής αυτών των κατηγοριών σε αριθμητικές ή δυαδικές τιμές. Στη πειραματική διαδικασία έγινε η επιλογή του Ordinal Encoder, μια τεχνική μετατροπής με παρόμοιας φιλοσοφίας με το Label Encoder, η οποία όμως υποστηρίζεται από το scklearn.pipeline[90] και έτσι επιλέγηκε λόγω ευκολίας ως προς την υλοποίηση. Εκτός όμως από τα κατηγορηματικά χαρακτηριστικά, υπάρχουν και τα αριθμητικά χαρακτηριστικά που αποτελούν τις άλλες 5 στήλες του συνόλου χαρακτηριστικών. Για την επεξεργασία αυτών θα χρησιμοποιηθεί ο Standard Scaler, έτσι ώστε να περιοριστεί το εύρος τιμών τους γύρω από το 0 κάτι το οποίο αυξάνει την απόδοση του μοντέλου, καθώς όλα τα χαρακτηριστικά θα έχουν την

ίδια βαρύτητα. Ο Standard Scaler λαμβάνει υπόψη τις διαφορετικές τιμές που μπορεί να πάρει κάθε χαρακτηριστικό και τις μετατρέπει σε ένα νέο σύνολο τιμών με μέση τιμή μ=0 και διασπορά σ=1. Ακολουθώντας όλα αυτά τα βήματα τα 5 νέα σύνολα δεδομένων είναι πλέον στην ιδανική μορφή για να ξεκινήσει η εκπαίδευση των μοντέλων. Τελευταία εργασία είναι η προεπιλογή του Train/Test Ratio το οποίο σύμφωνα με την έρευνα 2.2 η ιδανική τιμή είναι το 60/40.

3.2 Πείραμα 1: Εκπαίδευση και Σύγκριση Μοντέλων Επιβλεπόμενης Μηχανικής Μάθησης.

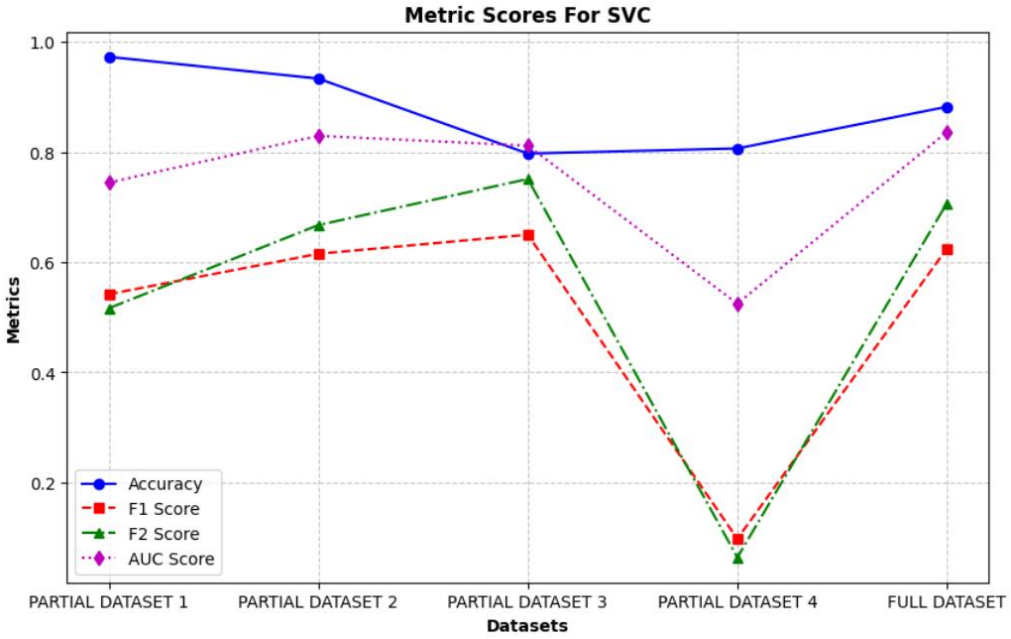
Στη πρώτη πειραματική έρευνα κύριος στόχος είναι να εντοπιστεί το καλύτερο μοντέλο επιβλεπόμενης μηχανικής μάθησης, το οποίο λαμβάνοντας ως είσοδο τα διαφορετικά σύνολα δεδομένων, θα παρουσιάζει όσο το δυνατό ψηλότερα ποσοστά στα αποτελέσματα των μετρικών. Οι τιμές των μετρικών είναι το καταλληλότερο μέτρο σύγκρισης έτσι ώστε να εντοπιστεί ποιος ταξινομητής είναι ο ιδανικός για την ανίχνευση ανωμαλιών στη περίπτωση του συνόλου δεδομένων UNSW-15. Για να μπορούν να ληφθούν μετρήσεις πρέπει τα σύνολα δεδομένων να τροποποιηθούν με τέτοιο τρόπο ώστε να είναι συμβατά με τις απαιτήσεις των μοντέλων. Επομένως, το πρώτο βήμα για την πειραματική διαδικασία και γενικότερα σε όλες τις μετέπειτα πειραματικές διαδικασίες είναι να προηγηθεί η προεπεξεργασία των δεδομένων πριν την εκπαίδευση των μοντέλων. Τα βήματα για την προεπεξεργασία δεδομένων περιγράφονται στο 3.1 και συνοπτικά είναι τα εξής. Αρχικά απαιτείται η ανάγνωση των 4 υποσυνόλων δεδομένων (Partial Datasets A,B,C,D) και η ένωση τους σε ένα μεγάλο συνολικό σύνολο δεδομένων (Full Dataset). Στη συνέχεια στα 5 σύνολα δεδομένων γίνεται ο διαχωρισμός των χαρακτηριστικών έτσι ώστε να απομείνουν αυτά που είναι όντως χρήσιμα για τις παρακάτω μετρήσεις. Από 49 που ήταν στη αρχή τελικά σε κάθε σύνολο δεδομένων παραμένουν 9, αυτά που βρίσκονται στον πίνακα 2. Έπειτα διαγράφονται οι ανεπιθύμητες εγγραφές, οι οποίες κατατάσσονται ως ανεπιθύμητες λόγω προβληματικών τιμών στα χαρακτηριστικά Source και Destination Port, κάτι που θα εμπόδιζε την ομαλή εκπαίδευση των μοντέλων. Ακολούθως είναι αναγκαία η μετατροπή των χαρακτηριστικών Source και Destination Port από συμβολοσειρές σε ακέραιους και τα χαρακτηριστικά Spkts και sbytes από συμβολοσειρές σε δεκαδικούς αριθμούς. Αυτό έγινε για την ευκολία υλοποίησης του επόμενου βήματος το οποίο είναι η μετατροπή των αριθμητικών χαρακτηριστικών ‘sport’,’dsport’,’sbytes’,’Spkts’,’dur’ μέσω ενός Standard Scaler έτσι ώστε να περιοριστεί το εύρος τιμών τους από 0 ως άπειρο σε τιμές γύρω από το 0. Ταυτόχρονα τα κατηγορηματικά χαρακτηριστικά ‘proto’, ’state’ θα τύχουν επεξεργασίας από έναν Ordinal Encoder. Τα 2 αυτά τελευταία τοποθετούνται σε ένα προεπεξεργαστή - preprocessor, που αποτελεί το τελευταίο βήμα της προεπεξεργασίας των δεδομένων εισόδου.

Μετά την προεπεξεργασία των δεδομένων και εφόσον τα σύνολα δεδομένων είναι σε κατάλληλη μορφή για εκπαίδευση, απαιτείται πειραματική έρευνα έτσι ώστε να εντοπιστεί το ιδανικό Supervised Μοντέλο που δίνει τα υψηλότερα ποσοστά στις μετρικές. Στο πείραμα έχουν υλοποιηθεί τα επιβλεπόμενα μοντέλα SVC,Logistic Regression,k-Nearest Neighbours , Decision Tree και Random Forest όπου έχουν εκπαιδευτεί στα 5 διαφορετικά σύνολα δεδομένων (4 μικρότερα και 1 ολικό) με κύριο στόχο την καταγραφή των μετρικών Accuracy,F1 Score,F2 Score,AUC Score πρωτίστως και Precision,Recall ως δευτερεύων. Τα αποτελέσματα των πιο πάνω πειραματισμών παρουσιάζονται πιο κάτω.

3.2.1 Εκπαίδευση Μοντέλου SVC

Πίνακας 3.2.1.1: Αποτελέσματα μετρήσεων διαφορετικών συνόλων δεδομένων με SVC

	PARTIAL DATASET 1	PARTIAL DATASET 2	PARTIAL DATASET 3	PARTIAL DATASET 4	FULL DATASET
Accuracy	0.9729	0.9334	0.7973	0.8066	0.8822
Precision	0.5906	0.5446	0.5310	0.8389	0.5233
Recall	0.5007	0.7072	0.8379	0.052	0.7751
F1 Score	0.5419	0.6153	0.6500	0.0978	0.6248
F2 Score	0.5164	0.6674	0.7511	0.064	0.7071
AUC Score	0.7446	0.8295	0.8117	0.5247	0.8364



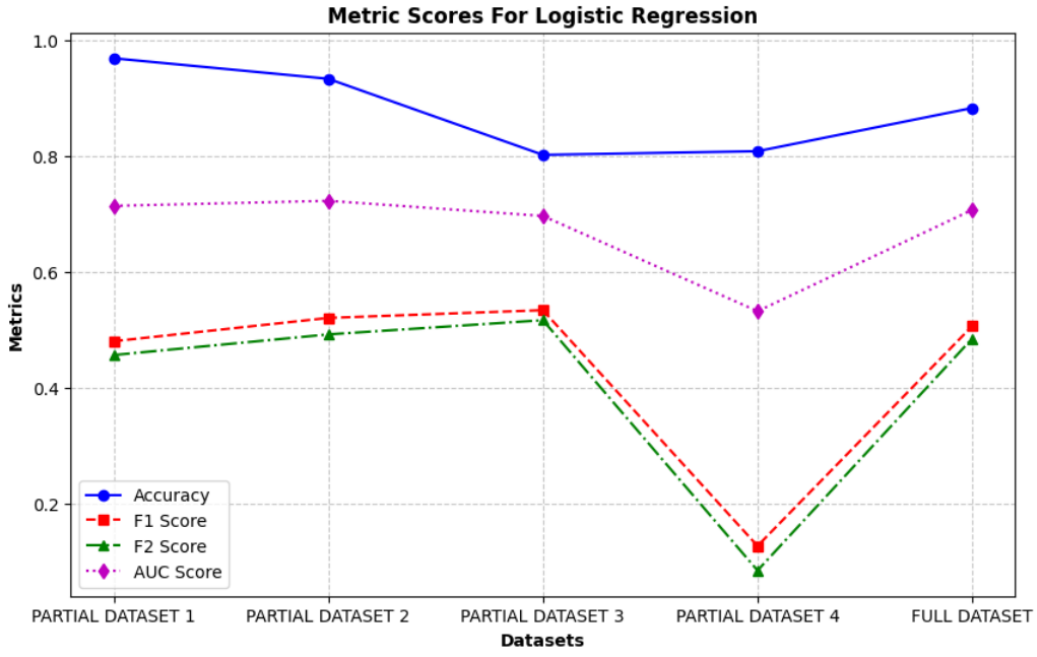
Σχήμα 3.2.1.2: Γραφική Παράσταση Απόδοσης Μετρικών σε διαφορετικά σύνολα Δεδομένων ενός SVC Μοντέλου

Με τη χρήση του SVC μοντέλου επιβλεπόμενης μηχανικής μάθησης παρατηρείται μια αστάθεια ως προς την απόδοση των μετρικών, ανάλογα με το σύνολο δεδομένων το οποίο χρησιμοποιείται για την εκπαίδευση του. Χαρακτηριστικό παράδειγμα είναι το Partial Dataset 4 για το οποίο το F1 Score ,F2 Score και AUC Score παρουσιάζουν πτώση στην απόδοση πάνω από 20% σε σχέση με τα υπόλοιπα σύνολα δεδομένων. Επίσης γενικά εμφανίζει χειρότερα αποτελέσματα από μοντέλα που παρουσιάζονται στη συνέχεια, παρόλο που ο χρόνος εκπαίδευσης στα SVC ήταν σημαντικά μεγαλύτερος συγκριτικά με τους υπόλοιπους ταξινομητές.

3.2.2 Εκπαίδευση Μοντέλου Logistic Regression

Πίνακας 3.2.2.1: Αποτελέσματα μετρήσεων διαφορετικών συνόλων δεδομένων με Logistic Regression

	PARTIAL DATASET 1	PARTIAL DATASET 2	PARTIAL DATASET 3	PARTIAL DATASET 4	FULL DATASET
Accuracy	0.9695	0.934	0.8025	0.8088	0.8835
Precision	0.5263	0.5764	0.5655	0.805	0.5468
Recall	0.4417	0.4747	0.5057	0.0686	0.4715
F1 Score	0.4803	0.5206	0.5340	0.1264	0.5064
F2 Score	0.4564	0.4921	0.5167	0.084	0.4849
AUC Score	0.7143	0.7231	0.6969	0.5322	0.7074



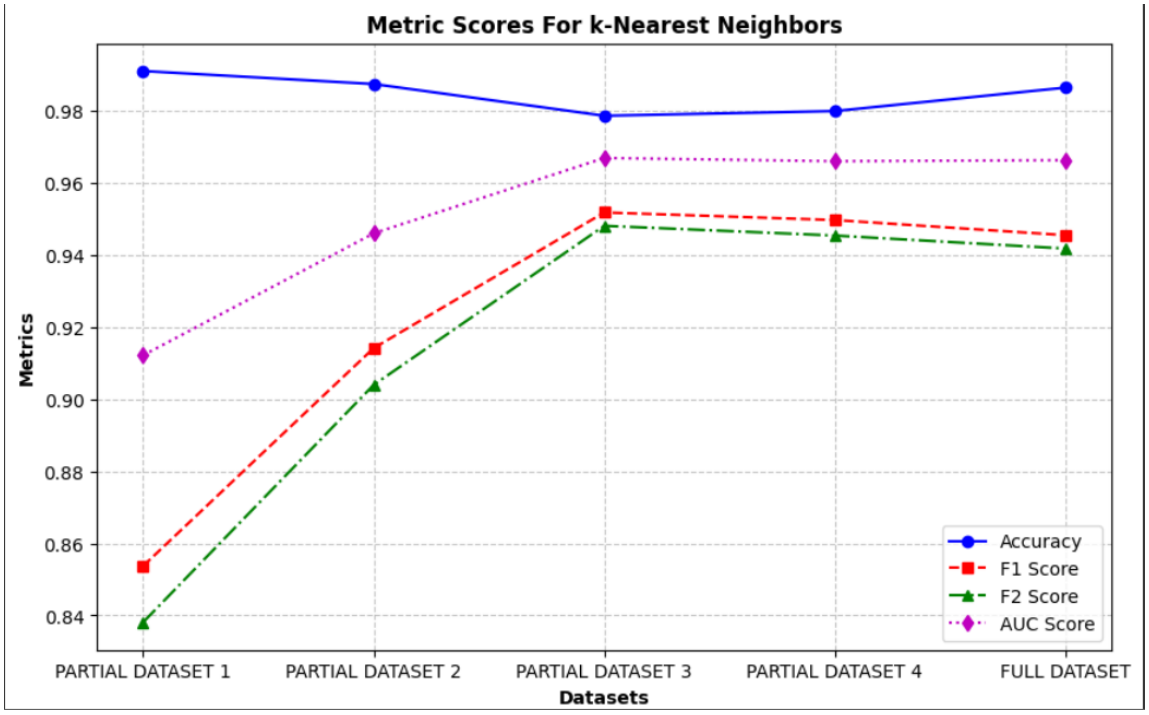
Σχήμα 3.2.2.2: Γραφική Παράσταση Απόδοσης Μετρικών σε διαφορετικά σύνολα Δεδομένων ενός Logistic Regression Μοντέλου

Στο μοντέλο Logistic Regression εμφανίζονται τα χειρότερα αποτελέσματα ως προς την απόδοση. Επίσης όπως και στην περίπτωση του SVC, στο Partial Dataset 4 υπάρχει μια σημαντική μείωση της απόδοσης στις μετρικές F1 Score ,F2 Score και AUC Score και οπότε φαίνεται πως το σύνολο δεδομένων μπορεί να επηρεάσει κατά πολύ τον τρόπο εκπαίδευσης των Logistic Regression Μοντέλων. Από την άλλη όμως, η εκπαίδευση του συγκεκριμένου μοντέλου γίνεται πολύ πιο γρήγορα σε σχέση με το SVC (20 φορές ταχύτερα) κάτι που το καθιστά προτιμότερη επιλογή από το SVC στο πείραμα 3.3

3.2.3 Εκπαίδευση Μοντέλου K-Nearest Neighbors

Πίνακας 3.2.3.1: Αποτελέσματα μετρήσεων διαφορετικών συνόλων δεδομένων με k-Nearest Neighbours

	PARTIAL DATASET 1	PARTIAL DATASET 2	PARTIAL DATASET 3	PARTIAL DATASET 4	FULL DATASET
Accuracy	0.9909	0.9873	0.9785	0.9798	0.9863
Precision	0.8811	0.9315	0.9580	0.9569	0.9517
Recall	0.8279	0.8973	0.9456	0.9425	0.9393
F1 Score	0.8537	0.9141	0.9517	0.9496	0.9454
F2 Score	0.8381	0.9039	0.9480	0.9453	0.9417
AUC Score	0.9121	0.9459	0.9668	0.9659	0.9662



Σχήμα 3.2.3.2: Γραφική Παράσταση Απόδοσης Μετρικών σε διαφορετικά σύνολα Δεδομένων ενός K-NN Μοντέλου

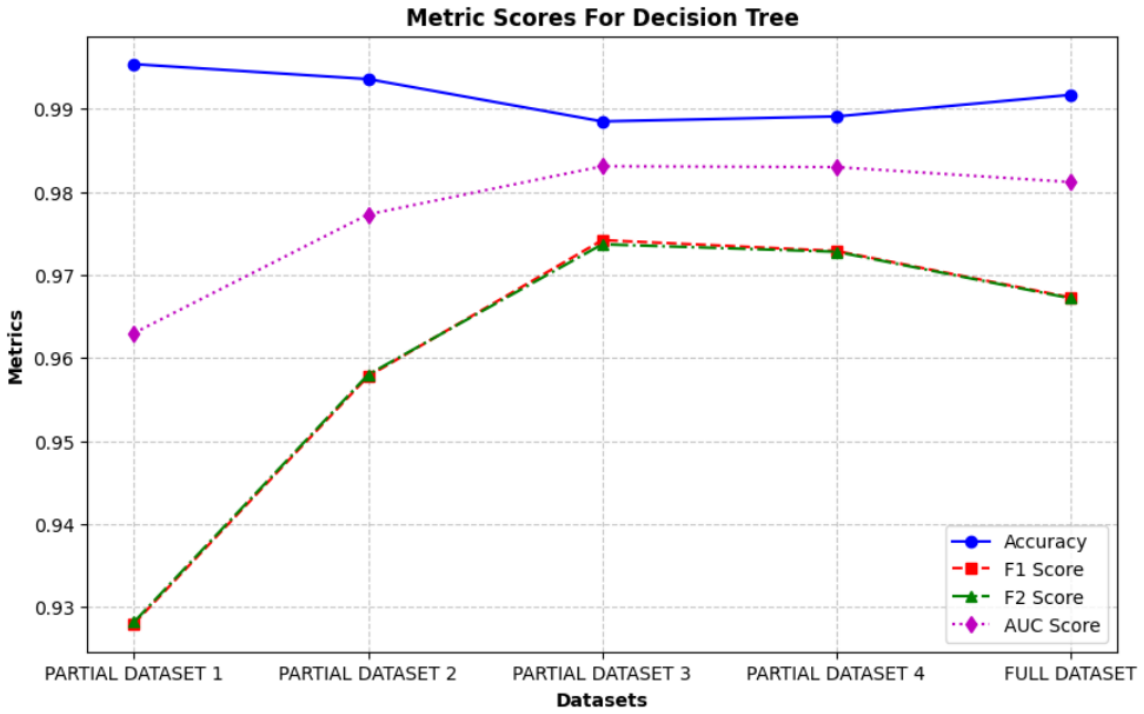
Το μοντέλο K-Nearest Neighbors όπως αποδεικνύεται από όπως πίνακες και όπως γραφικές παρουσιάζει σημαντικά ψηλότερα ποσοστά ως όπως την απόδοση σε σχέση με

όπως προηγούμενους ταξινομητές. Ο χρόνος εκπαίδευσης και πρόβλεψης του ήταν λίγο μεγαλύτερος από το Logistic Regression αλλά ταυτόχρονα πολύ γρηγορότερος από το χρόνο που χρειάστηκε στο SVC. Επιπλέον από την γραφική παράσταση εξάγεται το συμπέρασμα ότι όσο πιο ψηλό ποσοστό ανωμαλιών υπάρχει στο σύνολο δεδομένων, τόσο πιο καλή είναι η απόδοση του KNN μοντέλου. Τα Partial Datasets A και B (<10% ποσοστό επιθέσεων) έχουν πιο χαμηλές τιμές σε F1 Score, F2 Score και AUC Score συγκριτικά με τα Partial Datasets C και D (>20% επιθέσεις) αλλά και συγκριτικά με το Full Dataset (>10% επιθέσεις)

3.2.4 Εκπαίδευση Μοντέλου Decision Tree

Πίνακας 3.2.4.1: Αποτελέσματα μετρήσεων διαφορετικών συνόλων δεδομένων με Decision Tree

	PARTIAL DATASET 1	PARTIAL DATASET 2	PARTIAL DATASET 3	PARTIAL DATASET 4	FULL DATASET
Accuracy	0.9954	0.9936	0.9885	0.9891	0.9917
Precision	0.9276	0.9574	0.9752	0.9731	0.9675
Recall	0.9284	0.9582	0.9733	0.9728	0.9671
F1 Score	0.928	0.9578	0.9742	0.9729	0.9673
F2 Score	0.9282	0.958	0.9737	0.9728	0.9672
AUC Score	0.963	0.9773	0.9831	0.983	0.9812



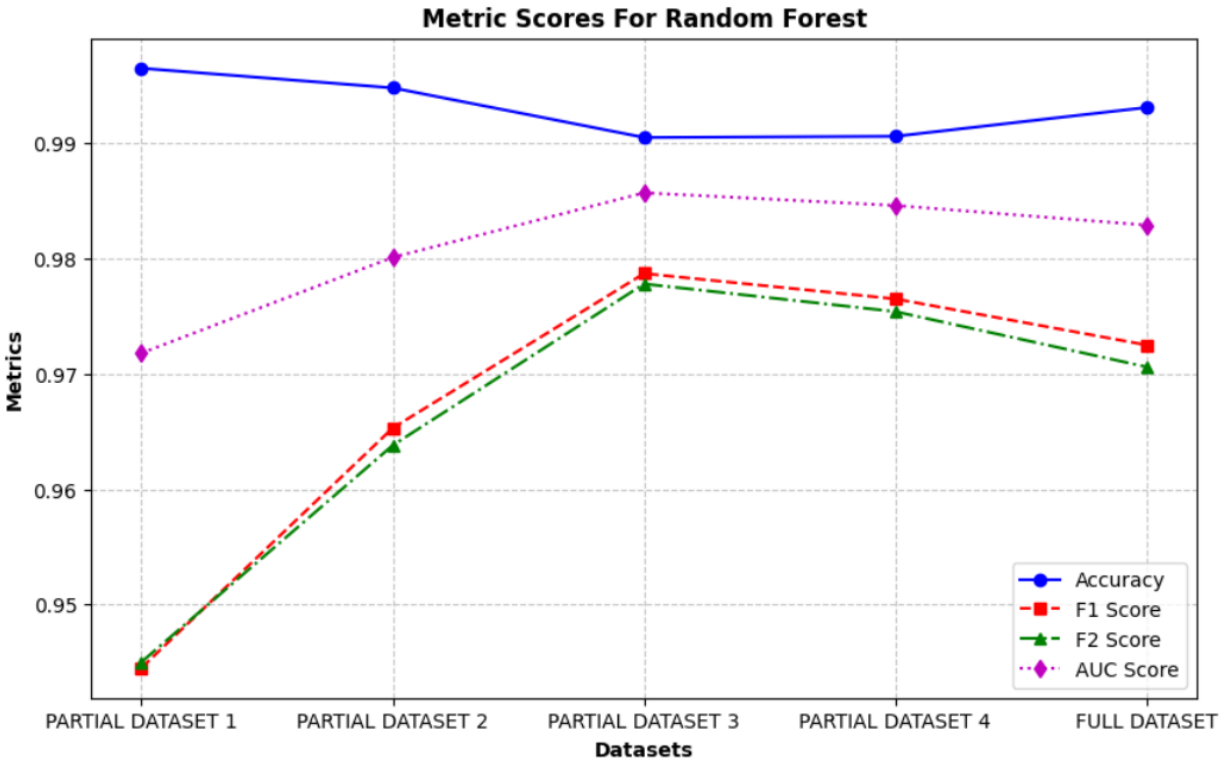
Σχήμα 3.2.4.2: Γραφική Παράσταση Απόδοσης Μετρικών σε διαφορετικά σύνολα Δεδομένων ενός Decision Tree Μοντέλου

Σχετικά με τον χρόνο εκπαίδευσης και πρόβλεψης, τα μοντέλα Decision Tree ήταν με διαφορά τα πιο γρήγορα σε όλα τα υποσύνολα δεδομένων. Όπως, όχι μόνο ήταν γρήγορα, αλλά όπως μετρικές F1 Score, F2 Score και AUC Score έχουν τα καλύτερα αποτελέσματα συγκριτικά με όπως 3 ταξινομητές που έχουν σχολιαστεί πιο πάνω. Μόνο το Random Forest που θα περιγραφεί στη συνέχεια έχει λίγο καλύτερα αποτελέσματα, κάτι που είναι λογικό καθώς το μοντέλο Random Forest ουσιαστικά είναι πολλά Decision Tree που εκπαιδεύονται μαζί. Όπως και στην περίπτωση του KNN φαίνεται ότι για σύνολα δεδομένων με ψηλά ποσοστά ανωμαλιών, η απόδοση του μοντέλου είναι μεγαλύτερη.

3.2.5 Εκπαίδευση Μοντέλου Random Forest

Πίνακας 3.2.5.1: Αποτελέσματα μετρήσεων διαφορετικών συνόλων δεδομένων με Random Forest

	PARTIAL DATASET 1	**PARTIAL DATASET 2**	**PARTIAL DATASET 3**	**PARTIAL DATASET 4**	**FULL DATASET**
Accuracy	0.9965	0.9948	0.9905	0.9906	0.9931
Precision	0.9437	0.9678	0.9803	0.9784	0.9758
Recall	0.9454	0.9628	0.9771	0.9746	0.9693
F1 Score	0.9445	0.9653	0.9787	0.9765	0.9725
F2 Score	0.945	0.9638	0.9778	0.9754	0.9706
AUC Score	0.9718	0.9801	0.9857	0.9846	0.9829



Σχήμα 3.2.5.2: Γραφική Παράσταση Απόδοσης Μετρικών σε διαφορετικά σύνολα Δεδομένων ενός Random Forest Μοντέλου

Το Random Forest αποτελεί το τελευταίο μοντέλο επιβλεπόμενης μηχανικής μάθησης στο πείραμα, και είναι το μοντέλο που παρουσιάζει την καλύτερη απόδοση σε σχέση με όλους τους υπόλοιπους ταξινομητές. Ο χρόνος εκπαίδευσης και πρόβλεψης είναι μεν πιο αργός από το Decision Tree όμως κυμαίνεται στα ίδια επίπεδα με τους KNN και Logistic Regression, αλλά και πολύ πιο γρήγορος από το SVC μοντέλο. Και πάλι, για σύνολα δεδομένων με ψηλά ποσοστά ανωμαλιών, η απόδοση του μοντέλου είναι μεγαλύτερη. Το Random Forest λόγω του ότι έχει την καλύτερη απόδοση, θα αποτελέσει το μοντέλο με το οποίο θα εκτελεστούν οι πειραματικές διαδικασίες 3.4 για την εύρεση υπερπαραμέτρων και 3.5 για την μείωση των διαστάσεων των χαρακτηριστικών.

3.2.6 Σύγκριση Αποτελεσμάτων Πειράματος

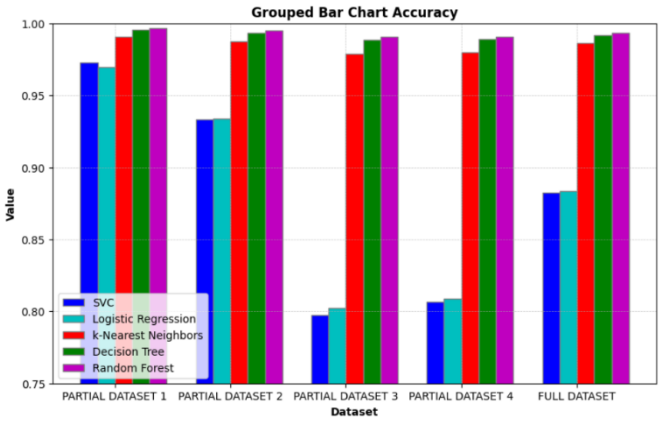
Πιο κάτω παρουσιάζονται όλα τα αποτελέσματα των μετρήσεων συγκεντρωτικά:

Πίνακας 3.2.6.1: Μέσος Όρος τιμών στα 4 μικρότερα (Partial) Datasest

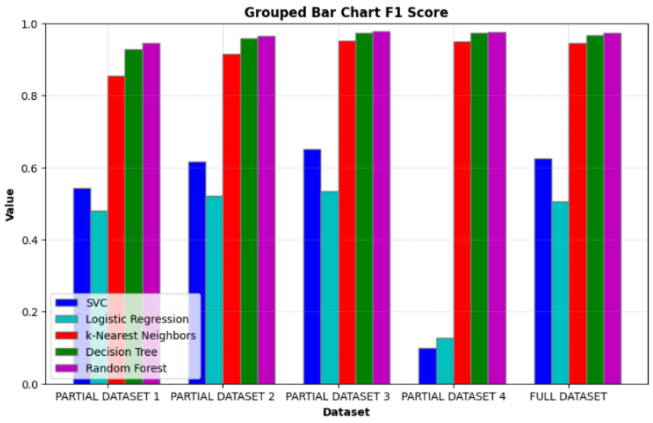
	SVC	LOGISTIC REGRESSION	KNN	DECISION TREE	RANDOM FOREST
Accuracy	0.8776	0.8787	0.9841	0.9917	0.9931
Precision	0.6263	0.6183	0.9319	0.9583	0.9676
Recall	0.5244	0.3727	0.9033	0.9582	0.965
F1 Score	0.4763	0.4153	0.9173	0.9582	0.9663
F2 Score	0.4997	0.3873	0.9088	0.9582	0.9655
AUC Score	0.7276	0.6666	0.9477	0.9766	0.9805

Πίνακας 3.2.6.2: Αποτελέσματα στο ολοκληρωμένο (Full) Datasest

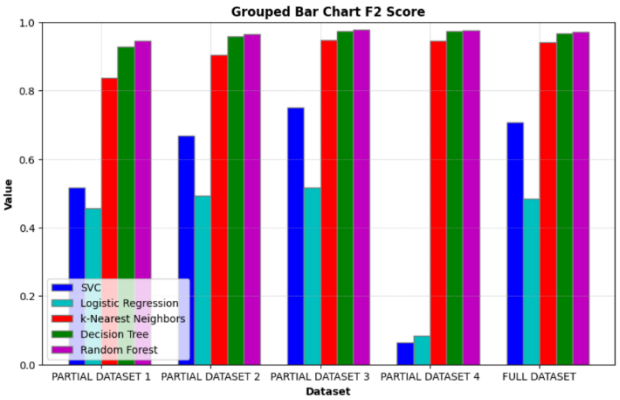
	SVC	LOGISTIC REGRESSION	KNN	DECISION TREE	RANDOM FOREST
Accuracy	0.8822	0.8835	0.9863	0.9917	0.9931
Precision	0.5233	0.5468	0.9517	0.9675	0.9758
Recall	0.7751	0.4715	0.9393	0.9671	0.9693
F1 Score	0.6248	0.5064	0.9454	0.9673	0.9725
F2 Score	0.7071	0.4849	0.9417	0.9672	0.9706
AUC Score	0.8364	0.7074	0.9662	0.9812	0.9829



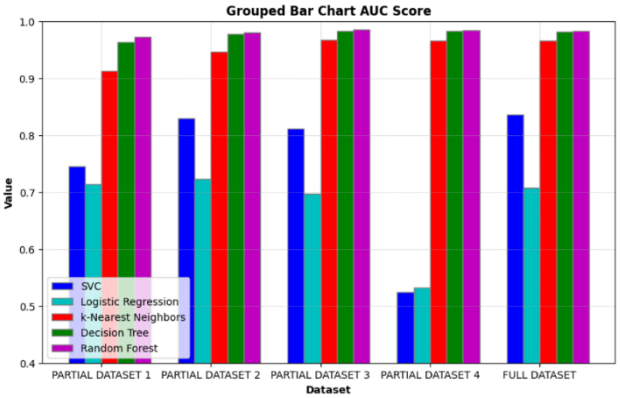
Σχήμα 3.2.6.3: Απόδοση Μετρικής Accuracy σε όλες τις πειραματικές διαδικασίες



Σχήμα 3.2.6.7: Απόδοση Μετρικής F1 Score σε όλες τις πειραματικές διαδικασίες



Σχήμα 3.2.6.8: Απόδοση Μετρικής F2 Score σε όλες τις πειραματικές διαδικασίες



Σχήμα 3.2.6.9: Απόδοση Μετρικής AUC Score σε όλες τις πειραματικές διαδικασίες

Με βάση όλες τις πιο πάνω γραφικές παραστάσεις είναι εύκολα αντιληπτό πως τα Μοντέλα K-NN , Decision Tree και Random Forest έχουν σταθερά ψηλές αποδόσεις σε όλες τις μετρικές, με το Random Forest να υπερτερεί ελάχιστα σε όλες τις περιπτώσεις των 2 υπόλοιπων ταξινομητών.

Από την άλλη, τα Μοντέλα SVC και Logistic Regression παρουσιάζουν αστάθεια ως προς την απόδοση των μετρικών, κυρίως στο Partial Dataset 4 όπου τα αποτελέσματα των μετρικών είναι πολύ χειρότερα συγκριτικά με τις μετρήσεις στα υπόλοιπα Datasets. Αυτά τα 2 μοντέλα σε καμία μετρική δεν πλησίασαν τις αποδόσεις των τριών ταξινομητών που αναφέρθηκαν προηγουμένως.

3.3 Πείραμα 2: Χρήση Voting Classifier και σύγκριση αποτελεσμάτων.

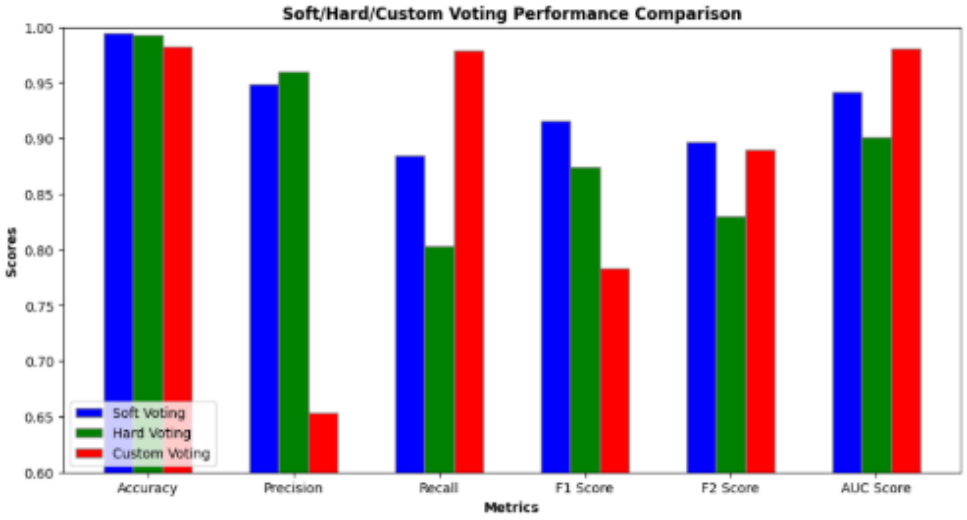
Αυτό το πείραμα και έχοντας ήδη μελετήσει τα διάφορα μοντέλα επιβλεπόμενης μηχανικής μάθησης στο 3.2 έχει στόχο την εισαγωγή ενός ταξινομητή ψηφοφορίας (Voting Classifier) και την σύγκριση των αποτελεσμάτων που θα προκύψουν σε σχέση με τις μετρήσεις στο πείραμα 3.2. Αφού πρώτα έχει γίνει η προεπεξεργασία δεδομένων με ακριβώς τον ίδιο τρόπο που περιγράφεται στο 3.1, στη συνέχεια επιλέχθηκαν τα Logistic Regression, KNN , Decision Tree και Random Forest μοντέλα έτσι ώστε να αποτελέσουν τους ταξινομητές οι οποίοι θα «ψηφίζουν» την πιθανή έξοδο του Voting Classifier. Το μόνο μοντέλο που απορρίφθηκε είναι το SVC καθώς χρειάζεται 5 με 6 φορές περισσότερο χρόνο εκπαίδευσης συγκριτικά με τα υπόλοιπα. Ταυτόχρονα το μοντέλο SVC έχει την δεύτερη χειρότερη απόδοση σε σχέση με τα 4 άλλα μονέλα ξεπερνώντας μόνο ελάχιστα το Logistic Regression και παρουσιάζοντας πολύ χειρότερα αποτελέσματα από τα KNN,Decision Tree και Random Forest μοντέλα. Όλες οι μετρήσεις στο πείραμα έγιναν με τη χρήση του Partial Dataset A.

3.3.1 Σύγκριση απόδοσης με Soft, Hard και Custom Voting απουσία βαρών

Ως μια πρώτη πειραματική διεργασία έχουν υπολογιστεί οι αποδόσεις του Voting Classifier που έχει σχηματιστεί χρησιμοποιώντας Soft Voting, Hard Voting και Custom Voting απουσία βαρών. Για υπενθύμιση, στην υλοποίηση του Soft Voting υπολογίζεται ο μέσος όρος της πιθανότητας συνολικά από όλα τα μοντέλα του να ανήκει μια εγγραφή σε κάθε κατηγορία και ως έξοδο παρουσιάζεται η κατηγορία με την πιο υψηλή πιθανότητα. Αντίθετα στο Hard Voting η λογική είναι πως ο κάθε ταξινομητής ψηφίζει στο ποια κατηγορία ανήκει η εγγραφή και ως έξοδο ο Voting Classifier δείχνει το ποια κατηγορία έχει τις περισσότερες ψήφους. Επιπλέον έχει υλοποιηθεί ένα Custom Voting, το οποίο κατατάσσει την εγγραφή στην κλάση 1 ως ανωμαλία αν οποιαδήποτε εγγραφή έχει τουλάχιστον 1 ψήφο από τα 4 επιβλεπόμενα μοντέλα προς αυτή τη κατηγορία. Δηλαδή αν τουλάχιστον 1 ταξινομητής προβλέπει ότι η εγγραφή είναι πιθανή ανωμαλία, τότε ο Voting Classifier προβλέπει και αυτός το ίδιο. Στον πίνακα 3.3.1.1 παρουσιάζεται η απόδοση του Voting Classifier για τις 3 διαφορετικές επιλογές Voting τα οποία απεικονίζονται και μέσα από την πιο κάτω γραφική παράσταση (Σχήμα 3.3.1.2).

Πίνακας 3.3.1: Απόδοση Voting Classifier με Soft, Hard και Custom Voting

	SOFT VOTING	**HARD VOTING**	**CUSTOM VOTING**
Accuracy	0.9948	0.9926	0.9827
Precision	0.9488	0.9595	0.6531
Recall	0.8846	0.8029	0.9786
F1 Score	0.9156	0.8742	0.7834
F2 Score	0.8967	0.8300	0.8899
AUC Score	0.9415	0.9009	0.9807

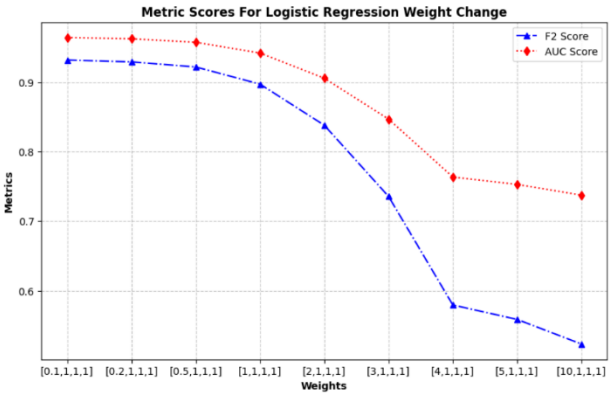


Σχήμα 3.3.1.2: Απόδοση Voting Classifier με Soft, Hard και Custom τρόπο ψηφοφορίας

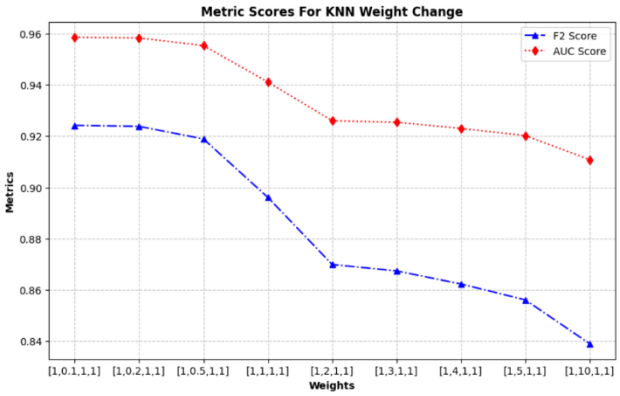
Από τα αποτελέσματα των μετρικών μπορούμε να καταλήξουμε στις εξής παρατηρήσεις. Αρχικά σχεδόν σε όλες τις μετρικές βλέπουμε ότι με Soft Voting υπάρχουν ψηλότερα ποσοστά στην απόδοση συγκριτικά με Hard Voting. Μόνο στην μετρική Precision το Hard Voting ξεπέρασε το Soft Voting σε απόδοση αλλά και αυτό με ελάχιστη διαφορά. Επίσης η υλοποίηση του Custom Voting φαίνεται να παρουσιάζει λίγο καλύτερα αποτελέσματα στις μετρικές Recall και AUC Score, αλλά ταυτόχρονα δίνει πολύ χειρότερα αποτελέσματα στις κατηγορίες Precision και F1 Score συγκριτικά με τις άλλες δύο τεχνικές ψήφισης. Ο λόγος που γίνεται αυτό είναι επειδή με βάση την υλοποίηση, το Custom Voting ψηφίζει ως ανωμαλία οποιαδήποτε εγγραφή έχει έστω και μια ψήφο από τους ταξινομητές προς αυτή την κλάση. Οπότε είναι πολύ πιθανό να υπάρχουν πολλά False Positive αποτελέσματα, τα οποία ρίχνουν την απόδοση κυρίως του Precision, που δίνει πολλή βαρύτητα σε αυτό το είδος αποτελέσματος. Η μόνη μετρική που ξεπέρασε τις τιμές των μοντέλων στο 3.2 είναι το AUC Score με τη χρήση του Custom Voting, το οποίο όμως υστερούσε στην απόδοση στις υπόλοιπες κατηγορίες.

3.3.2 Σύγκριση Απόδοσης Voting Classifier με την εισαγωγή βαρών

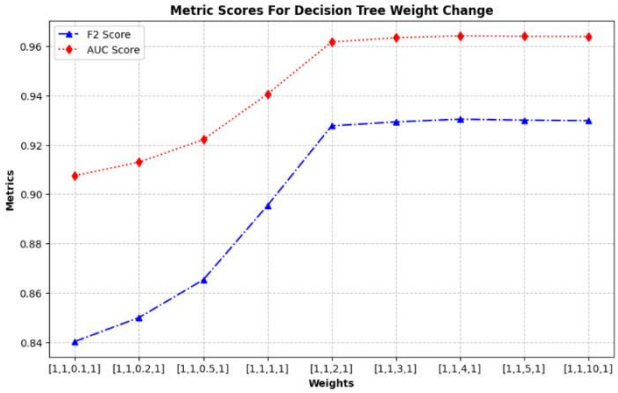
Στη συνέχεια γίνεται έλεγχος για το πως οι τιμές των βαρών που επιλέγονται για τον κάθε ταξινομητή επηρεάζουν την απόδοση των μετρικών. Για κάθε ένα από τα 4 μοντέλα Logistic Regression, KNN, Decision Tree και Random Forest έχουν ληφθεί μετρήσεις μεταβάλλοντας το εύρος των βαρών τους από 0.1 εως 10, διατηρώντας τα υπόλοιπα βάρη σταθερά και ίσα με 1. Ως τεχνική ψήφισης έχει τεθεί το Soft Voting επειδή όπως έχουμε παρατηρήσει προηγουμένως παρουσιάζει ψηλότερες αποδόσεις συγκριτικά με το Hard Voting αλλά και ποιο σταθερές τιμές ανάμεσα στις μετρικές συγκριτικά με το Custom Voting. Τα αποτελέσματα των μετρικών F2 Score και AUC Score παρουσιάζονται στις πιο κάτω γραφικές παραστάσεις (Σχήμα 3.3.2.1 – 3.3.2.4) όπως και συνολικά για όλες τις μετρικές στους πίνακες 3.3.2.5 – 3.3.2.8



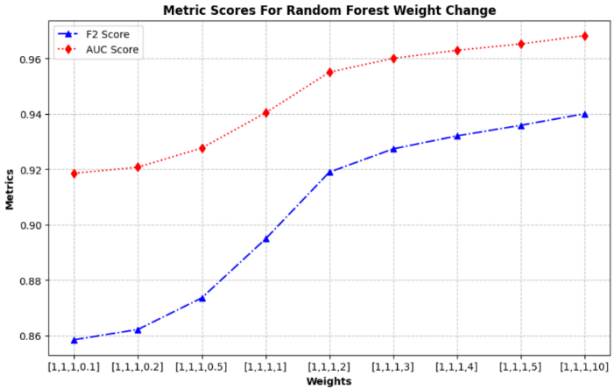
Σχήμα 3.3.2.1: Απόδοση Μετρικών με Αλλαγή Βάρους στον Ταξινομητή Logistic Regression



Σχήμα 3.3.2.2: Απόδοση Μετρικών με Αλλαγή Βάρους στον Ταξινομητή K-NN



Σχήμα 3.3.2.3: Απόδοση Μετρικών με Αλλαγή Βάρους στον Ταξινομητή Decision Tree



Σχήμα 3.3.2.4: Απόδοση Μετρικών με Αλλαγή Βάρους στον Ταξινομητή Random Forest

Παρατηρούμε ότι όσο αυξάνεται η τιμή του βάρους στα μοντέλα, η μετρικές F2 Score και AUC Score στο Voting Classifier συγκλίνουν προς την τιμή που θα είχε το μοντέλο αν εκπαιδευόταν μόνο του. Αυτό ήταν και το αναμενόμενο, καθώς ο συγκεκριμένος ταξινομητής έχει πολύ μεγαλύτερη βαρύτητα από τους υπόλοιπους και ουσιαστικά η επιρροή των υπόλοιπων μοντέλων στην τελική απόφαση είναι αμελητέα. Όσο αυξάνεται η τιμή του βάρους, η απόδοση των μετρικών στο πείραμα με το Logistic Regression και το KNN μειώνεται ενώ η απόδοση στο πείραμα με το Decision Tree και το Random Forest αυξάνεται. Σε όλες τις περιπτώσεις για τη μεταβολή του βάρους ενός από τα μοντέλα υπάρχει τουλάχιστον ένας συνδυασμός στο οποίο προκύπτουν υψηλότερα ποσοστά σχετικά με την περίπτωση απουσίας βαρών, δηλαδή το σενάριο τα βάρη να είναι [1,1,1,1]. Στο Logistic Regression και στο KNN αν η βαρύτητα τους μειωθεί σε τιμή λιγότερο από 1, η απόδοση αυξάνεται, ενώ στα άλλα 2 μοντέλα αν η βαρύτητα τους αυξηθεί σε τιμή μεγαλύτερη από 1, η απόδοση μειώνεται. Επίσης σημαντική είναι η αναφορά στο ότι κανένας συνδυασμός βαρών δεν ξεπέρασε τις μετρικές του Random Forest στο 3.1 στην

περίπτωση εκπαιδεύτηκε για το Partial Dataset A,δηλαδή το σύνολο δεδομένων το οποίο χρησιμοποιείται και εδώ.

Πίνακας 3.3.2.5: Απόδοση Voting Classifier με αλλαγή στις τιμές του βάρους στον ταξινομητή Logistic Regression

Metrics	Logistic Regression with Weight Change = [w,1,1,1]								
	w=0.1	w=0.2	w=0.5	w=1	w=2	w=3	w=4	w=5	w=10
Accuracy	0.9958	0.9958	0.9956	0.9948	0.9929	0.9894	0.9841	0.9834	0.9806
Precision	0.9388	0.9399	0.9443	0.9483	0.9604	0.9645	0.9546	0.9514	0.8468
Recall	0.9299	0.9263	0.9163	0.885	0.8120	0.6943	0.5276	0.5064	0.4777
F1 Score	0.9343	0.9331	0.9301	0.9156	0.8800	0.8074	0.6796	0.6610	0.6108
F2 Score	0.9317	0.9290	0.9217	0.8970	0.8379	0.7355	0.5794	0.5587	0.5233
AUC Score	0.9640	0.9622	0.9572	0.9417	0.9055	0.8467	0.7634	0.7528	0.7374

Πίνακας 3.3.2.6: Απόδοση Voting Classifier με αλλαγή στις τιμές του βάρους στον ταξινομητή K-Nearest Neighbors

Metrics	K-Nearest Neighbors with Weight Change = [1,w,1,1]								
	w=0.1	w=0.2	w=0.5	w=1	w=2	w=3	w=4	w=5	w=10
Accuracy	0.9957	0.9957	0.9956	0.9948	0.9936	0.9933	0.9929	0.9924	0.9915
Precision	0.9462	0.9467	0.9467	0.9491	0.9412	0.9304	0.9232	0.9133	0.9020
Recall	0.9188	0.9183	0.9122	0.8838	0.8537	0.8529	0.8483	0.8429	0.8246
F1 Score	0.9323	0.9323	0.9291	0.9153	0.8953	0.8900	0.8842	0.8767	0.8616
F2 Score	0.9242	0.9238	0.9189	0.8961	0.8699	0.8674	0.8623	0.8561	0.8390
AUC Score	0.9586	0.9583	0.9533	0.9411	0.9260	0.9254	0.923	0.9202	0.9108

Πίνακας 3.3.2.7: Απόδοση Voting Classifier με αλλαγή στις τιμές του βάρους στον ταξινομητή Decision Tree

Metrics	Decision Tree with Weight Change = [1,1,w,1]								
	w=0.1	w=0.2	w=0.5	w=1	w=2	w=3	w=4	w=5	w=10
Accuracy	0.9928	0.9932	0.9938	0.9948	0.9956	0.9955	0.9955	0.9955	0.9955
Precision	0.9525	0.9543	0.9555	0.9495	0.9364	0.9299	0.9286	0.9289	0.9285
Recall	0.8163	0.8274	0.8454	0.8828	0.9256	0.9291	0.9308	0.9303	0.9301
F1 Score	0.8791	0.8863	0.8971	0.915	0.9309	0.9295	0.9297	0.9296	0.9293
F2 Score	0.8403	0.8500	0.8653	0.8954	0.9277	0.9293	0.9304	0.9300	0.9298
AUC Score	0.9075	0.9130	0.9221	0.9406	0.9617	0.9634	0.9642	0.9640	0.9639

Πίνακας 3.3.2.8: Απόδοση Voting Classifier με αλλαγή στις τιμές του βάρους στον ταξινομητή Random Forest

Metrics	Random Forest with Weight Change = [1,1,1,w]								
	w=0.1	w=0.2	w=0.5	w=1	w=2	w=3	w=4	w=5	w=10
Accuracy	0.9934	0.9935	0.9939	0.9947	0.9956	0.9960	0.9961	0.9963	0.9964
Precision	0.9478	0.9481	0.9475	0.9483	0.9490	0.9508	0.9503	0.9509	0.9475
Recall	0.8388	0.8431	0.8569	0.8825	0.9118	0.9218	0.9277	0.9323	0.9383
F1 Score	0.8900	0.8925	0.9000	0.9142	0.9300	0.9361	0.9389	0.9415	0.9429
F2 Score	0.8585	0.8622	0.8736	0.8949	0.9190	0.9274	0.9321	0.9359	0.9401
AUC Score	0.9186	0.9208	0.9277	0.9404	0.9551	0.9601	0.9630	0.9653	0.9683

3.4 Πείραμα 3: Χρήση υπερπαραμέτρων σε Random Forest ταξινομητή για προσπάθεια βελτίωσης των μετρικών του μοντέλου.

Από την πειραματική διαδικασία του 3.2 φάνηκε πως ο ταξινομητής Random Forest έχει με διαφορά την καλύτερη απόδοση συγκριτικά με τα υπόλοιπα μοντέλα, ειδικά στις μετρικές F2 Score και AUC Score, οι οποίες ιδανικά θέλουμε να έχουν όσο το δυνατό ψηλότερες τιμές. Μέχρι τώρα η εκπαίδευση των μοντέλων έγινε χωρίς την χρήση υπερπαραμέτρων (1.6), δηλαδή την εισαγωγή κάποιων προκαθορισμένων μεταβλητών πριν ξεκινήσει η εκπαίδευση του συνόλου δεδομένων έτσι ώστε ένας χρήστης να επηρεάσει την συμπεριφορά και τον τρόπο που θα εκπαιδευτεί ένα συγκεκριμένο μοντέλο. Οι πειραματικές διαδικασίες 3.4.1 – 3.4.3 που ακολουθούν έγιναν με τη χρήση του Partial Dataset A.

3.4.1 Εύρεση Βέλτιστων τιμών σε Υπερπαραμέτρους ενός Random Forest Μοντέλου.

Κύριος στόχος της πειραματικής διαδικασίας 3.4 είναι να εντοπιστούν ποιες από τις υπερπαραμέτρους του Random Forest(1.6.2) βελτιώνουν τις τιμές στις μετρικές και ποιες θα είναι αυτές οι νέες αποδόσεις των μοντέλων. Οι υπερπαραμέτροι οι οποίοι ελέγχονται στο πείραμα είναι οι n_estimators(1.6.2.1), max_depth(1.6.2.2), min_samples_split(1.6.2.3), min_samples_leaf(1.6.2.4) και class_weight (1.6.2.8). Για κάθε μια από τις 5 υπερπαμέτρους έχουν προκαθοριστεί από 2 έως 6 πιθανές τιμές με στόχο στην συνέχεια μέσα από μια σειρά μετρήσεων να εντοπιστεί η ιδανική τιμή σε κάθε περίπτωση. Συνολικά στην πειραματική διαδικασία υπάρχουν 648 πιθανοί συνδυασμοί των τιμών που μπορούν να λάβουν οι 5 διαφορετικοί υπερπαράμετροι. Ο τρόπος με τον οποίο τελικά θα εντοπιστούν οι ιδανικές τιμές είναι με την εκτέλεση 648 διαφορετικών Random Forest μοντέλων, 1 για κάθε συνδυασμό τιμών, και στο τέλος να ληφθεί υπόψη ο μέσος όρος στις μετρικές για κάθε τιμή κάθε υπερπαμαμέτρου. Για παράδειγμα για το n_estimators έχουν προεπιλεχθεί 6 διαφορετικές τιμές οπότε για κάθε μια από τις 6 τιμές

θα βρεθεί ο μέσος όρος από τα 648/6 = 108 πειράματα που έχουν εκτελεστεί στην οποία υπήρχε αυτή η τιμή.

Η πρώτη υπερπαράμετρος που μας απασχόλησε είναι το n_estimators, δηλαδή ο αριθμός των δέντρων αποφάσεων που θα δημιουργηθούν κατά της διάρκεια της εκπαίδευσης. Συνολικά μπορεί να πάρει 6 διαφορετικές τιμές, τις [1,5,10,30,50,100] .Η δεύτερη είναι το max_depth, δηλαδή το μέγιστο ύψος το οποίο μπορεί να φτάσει κάθε δέντρο. Οι τιμές που ελέγχονται είναι 6, οι [10,15,20,25,30,None]. Η τρίτη είναι το min_samples_split,που δηλώνει τον αριθμό των δειγμάτων που απαιτούνται σε ένα κόμβο έτσι ώστε ο κόμβος να διαχωριστεί και να δημιουργηθούν 2 νέοι κόμβοι-παιδιά . Μπορεί να πάρει 3 πιθανές τιμές, τις [2,5,10]. Επιπλέον υπάρχει το min_samples_leaf όπου καθορίζει τον αριθμό των δειγμάτων που απαιτούνται σε ένα κόμβο – φύλλο. Οι τιμές που παίρνει είναι 3, οι [1,2,4]. Η τελευταία υπερπαράμετρος είναι το class_weight, όπου λειτουργία του είναι ο καθορισμός της βαρύτητας κάθε δείγματος ανάλογα με την κλάση στην οποία ανήκει. Οι τιμές που μπορεί να πάρει είναι 2. Η πρώτη είναι η κλασσική περίπτωση όπου κάθε δείγμα έχει την ίδια βαρύτητα ανεξάρτητα με το σε πια κλάση ανήκει (Default). Όσο αφορά την δεύτερη τιμή , έχει υπολογιστεί πόσα δείγματα ανήκουν στο σύνολο με την κλάση 1 σε σχέση με το συνολικό αριθμό των δειγμάτων, και έχει δοθεί αντιστρόφως ανάλογη βαρύτητα στην κλάση 1 (Custom). Δηλαδή αν υπάρχουν 100 δείγματα κλάσης 0 και 1 δείγμα κλάσης 1, τότε η βαρύτητα των δειγμάτων της κλάσης 1 είναι 100 φορές μεγαλύτερη σε σχέση με τα δείγματα της κλάσης 0. Στους πίνακες 3.4.1.1 – 3.4.1.3 παρουσιάζονται αναλυτικά οι μέσοι όροι των πιθανών τιμών της κάθε υπερπαραμέτρου.

Πίνακας 3.4.1.1: Απόδοση διαφορετικών τιμών της υπερπαραμέτρου n_estimators

Metrics	n_estimators = ? (ΜΟ. 108 Πειραμάτων για κάθε τιμή)					
	1	5	10	30	50	100
Accuracy	0.9915	0.9940	0.9944	0.9946	0.9946	0.9947
Precision	0.8445	0.8811	0.8882	0.8893	0.8897	0.8900
Recall	0.9233	0.9462	0.9499	0.9539	0.9555	0.9558
F1 Score	0.8776	0.9102	0.9162	0.9186	0.9196	0.9199
F2 Score	0.9030	0.9309	0.9356	0.9390	0.9403	0.9406
AUC Score	0.9585	0.9709	0.9729	0.9749	0.9757	0.9759

Πίνακας 3.4.1.2: Απόδοση διαφορετικών τιμών της υπερπαραμέτρου max_depth

Metrics	max_depth = ? (ΜΟ. 108 Πειραμάτων για κάθε τιμή)					
	10	15	20	25	30	None
Accuracy	0.9904	0.9940	0.9946	0.9949	0.9949	0.9950
Precision	0.8252	0.8776	0.8889	0.8946	0.8963	0.9003
Recall	0.9170	0.9502	0.9565	0.9554	0.9539	0.9517
F1 Score	0.8624	0.9101	0.9199	0.9227	0.9230	0.9241
F2 Score	0.8924	0.9331	0.9410	0.9417	0.9410	0.9401
AUC Score	0.9549	0.9728	0.9762	0.9758	0.9751	0.9740

Πίνακας 3.4.1.3: Απόδοση διαφορετικών τιμών των υπερπαραμέτρων min_samples_split, min_samples_leaf και class_weights

Metrics	Min_samples_split = ? (ΜΟ. 216 Πειραμάτων)			Min_samples_leaf = ? (ΜΟ. 216 Πειραμάτων)			Class_weights=? (ΜΟ. 324 Πειραμάτων)	
	2	5	10	1	2	4	Default	Custom
Accuracy	0.9940	0.9941	0.9938	0.9942	0.9941	0.9936	0.9949	0.9930
Precision	0.8827	0.8830	0.8758	0.8890	0.8818	0.8707	0.9266	0.8344
Recall	0.9458	0.9479	0.9486	0.9454	0.9477	0.9492	0.9138	0.9810
F1 Score	0.9108	0.9122	0.9081	0.9141	0.9114	0.9056	0.9200	0.9007
F2 Score	0.9308	0.9327	0.9312	0.9319	0.9322	0.9305	0.9162	0.9469
AUC Score	0.9707	0.9718	0.9719	0.9706	0.9716	0.9722	0.9557	0.9872

Από όλους τους μέσους όρους έγινε η επιλογή των τιμών οι οποίες έχουν τις ψηλότερες τιμές στις μετρικές που μας ενδιαφέρουν, δηλαδή το F2 Score και AUC Score. Υπάρχουν υπερπαμαμέτροι που η επιλογή είναι προφανής καθώς υπάρχει μια συγκεκριμένη τιμή που τόσο στο F2 Score όσο και στο AUC Score εμφανίζεται η πιο ψηλή απόδοση. Τέτοιες περιπτώσεις είναι η τιμή n_estimators = 100, η τιμή min_samples_split = 5 και η τιμή class_weights = custom , δηλαδή η εισαγωγή βαρύτητας αντιστρόφως ανάλογη της συχνότητας εμφάνισης της κάθε κλάσης, για οποία παρατηρείται τεράστια διαφορά στην απόδοση συγκριτικά με την άλλη περίπτωση. Στις υπόλοιπες 2 υπερπαραμέτρους, για max_depth = 25 και min_samples_leaf = 2 εμφανίζονται σε γενικές γραμμές οι ψηλότερες αποδόσεις, οπότε προτιμήθηκαν αυτές οι τιμές.

3.4.2 Σύγκριση Απόδοσης Random Forest Μοντέλου με Default και με νέες Υπερπαραμέτρους

Εφόσον έγινε ο εντοπισμός των βέλτιστων τιμών για τις 5 υπερπαραμέτρους, σε αυτό το στάδιο εκπαιδεύτηκε ένας νέος Random Forest ταξινομητής με στόχο τον έλεγχο του αν όντως η εισαγωγή των υπερπαραμέτρων βοήθησε στην αύξηση της απόδοσης του αρχικού μοντέλου. Τα αποτελέσματα εμφανίζονται στον πίνακα 3.4.2.1 μαζί με την απόδοση του Random Forest στο 3.1 με τις Default υπερπαραμέτρους για σκοπούς σύγκρισης.

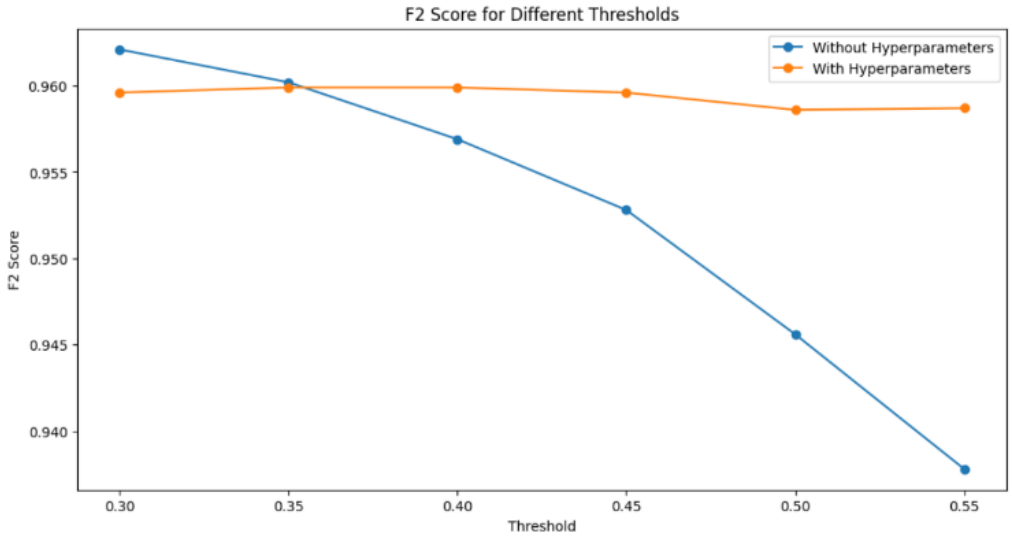
Πίνακας 3.4.2.1: Σύγκριση Απόδοσης Random Forest με Default και με νέες υπερπαραμέτρους.

	Random Forest με Default Υπερπαραμέτρους	**Random Forest Με Υπερπαραμέτρους**	**Διαφορά**
Accuracy	0.9965	0.9946	-0.0019
Precision	0.9437	0.8631	-0.0806
Recall	0.9454	0.9875	+0.0421
F1 Score	0.9445	0.9211	-0.0234
F2 Score	0.945	0.9598	+0.0148
AUC Score	0.9718	0.9912	+0.0198

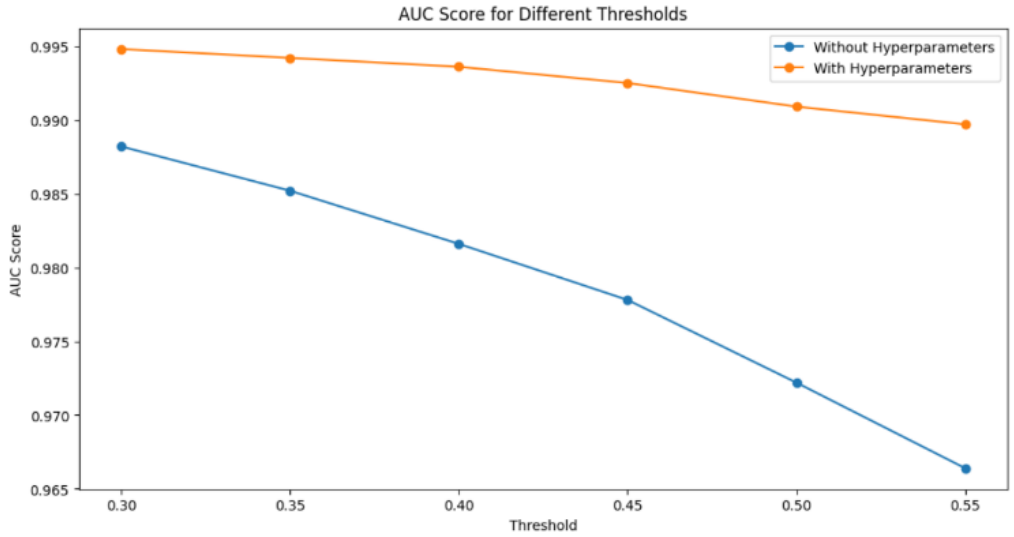
Παρατηρούμε πως η εισαγωγή των παραμέτρων άλλαξε κατά πολύ τις αποδόσεις των μετρικών. Υπάρχουν μετρικές που ο συνδυασμός των τιμών στις υπερπαραμέτρους μείωσε αισθητά τα ποσοστά τους, με χαρακτηριστικότερο παράδειγμα το precision, το οποίο παρουσίασε πτώση κατά 8%. Παρόλα αυτά στόχος ήταν η αύξηση των αποδόσεων των F2 Score και AUC Score, κάτι το οποίο επιτεύχθηκε, καθώς υπήρξε άνοδος στην απόδοση τους σχεδόν 1.5% και 2% αντίστοιχα, αύξηση που έγινε σε τιμές που ήταν ήδη αρκετά ψηλές.

3.4.3 Έλεγχος Συμπεριφοράς Μοντέλων για διαφορετικές τιμές κατωφλίου

Σε ένα δυαδικό ταξινομητή όπως στην περίπτωση μας, το μοντέλο προβλέπει την πιθανότητα μια εγγραφή να ανήκει στην κλάση 1, δηλαδή να αποτελεί ανωμαλία στο δίκτυο και αν αυτή η πιθανότητα είναι πάνω από την προκαθορισμένη τιμή του κατωφλίου 0.5 (50%) τότε κατατάσσει την συγκεκριμένη κίνηση ως επίθεση. Ένα τελευταίο πείραμα είναι η εφαρμογή διαφορετικών κατωφλίων-threshold και ο έλεγχος του πως η τιμή κατωφλίου επηρεάζει την απόδοση των μετρικών. Ως μοντέλα θα χρησιμοποιηθούν τα 2 που υπάρχουν στον πίνακα 3.4.2.1, δηλαδή ένα Random Forest που θα εκπαιδευτεί με και ένα Random Forest χωρίς υπερπαραμέτρους. Ως σύνολο δεδομένων θα χρησιμοποιηθεί το Partial Dataset A και το κατώφλι θα κυμαίνεται στο εύρος 0.35-0.55. Οι πιο κάτω γραφικές παραστάσεις (Σχήμα 3.4.3.1 και 3.4.3.2) παρουσιάζουν τα αποτελέσματα για το F2 Score και το AUC Score, τα οποία υπάρχουν αναλυτικά και στους πίνακες 3.4.3.3 - 3.4.3.4



Σχήμα 3.4.3.1: Σύγκριση Απόδοσης F2 Score σε Random Forest Μοντέλο με Default και με νέες υπερπαραμέτρους για διαφορετικές τιμές κατωφλίου.



Σχήμα 3.4.3.2: Σύγκριση Απόδοσης AUC Score σε Random Forest Μοντέλο με Default και με νέες υπερπαραμέτρους για διαφορετικές τιμές κατωφλίου

Αυτό που παρατηρούμε από τις γραφικές είναι πως όσο αφορά το AUC Score, η εισαγωγή υπερπαραμέτρων βελτιώνει την απόδοση σε όλες τις περιπτώσεις ανεξάρτητα με την τιμή κατωφλίου. Στο F2 Score παρατηρείται πως με υπερπαραμέτρους η απόδοση των μοντέλων είναι σταθερά λίγο πιο κάτω από 0.96 σε όλες τις περιπτώσεις, ενώ αντίθετα με τις προκαθορισμένες υπερπαραμέτρους η απόδοση επηρεάζεται σε μεγάλο βαθμό από την τιμή του κατωφλίου. Για κατώφλι <0.35, το μοντέλο με Default υπερπαραμέτρους δίνει καλύτερα αποτελέσματα στο F2 Score συγκριτικά με το μοντέλο με υπερπαραμέτρους, ενώ για κατώφλι > 0.35 συμβαίνει το αντίθετο. Σημαντική είναι η

αναφορά στο ότι οι ιδανικές τιμές των υπερπαραμέτρων υπολογίστηκαν για κατώφλι 0.5, οπότε αν επαναληφθεί η ίδια διαδικασία για π.χ. κατώφλι = 0.3, τότε είναι πολύ πιθανό να υπάρχουν διαφορετικές ιδανικές τιμές.

Πίνακας 3.4.3.3: Απόδοση μοντέλου Random Forest με Default υπερπαραμέτρους για διαφορετικές τιμές κατωφλίου

Metrics	Random Forest με Default Υπερπαραμέτρους / threshold = ?					
	0.30	0.35	0.40	0.45	0.50	0.55
Accuracy	0.9957	0.9961	0.9963	0.9965	0.9965	0.9964
Precision	0.8964	0.9100	0.9228	0.9329	0.9429	0.9514
Recall	0.9801	0.9736	0.9659	0.9579	0.9463	0.9344
F1 Score	0.9364	0.9407	0.9438	0.9453	0.9446	0.9428
F2 Score	0.9621	0.9602	0.9569	0.9528	0.9456	0.9378
AUC Score	0.9882	0.9852	0.9816	0.9778	0.9722	0.9664

Πίνακας 3.4.3.4: Απόδοση μοντέλου Random Forest με υπερπαραμέτρους για διαφορετικές τιμές κατωφλίου

Metrics	Random Forest με Υπερπαραμέτρους / threshold = ?					
	0.30	0.35	0.40	0.45	0.50	0.55
Accuracy	0.9937	0.9939	0.9940	0.9942	0.9944	0.9947
Precision	0.8371	0.8422	0.8466	0.8527	0.8595	0.8686
Recall	0.9960	0.9946	0.9932	0.9906	0.9871	0.9842
F1 Score	0.9097	0.9121	0.9141	0.9165	0.9189	0.9228
F2 Score	0.9596	0.9599	0.9599	0.9596	0.9586	0.9587
AUC Score	0.9948	0.9942	0.9936	0.9925	0.9909	0.9897

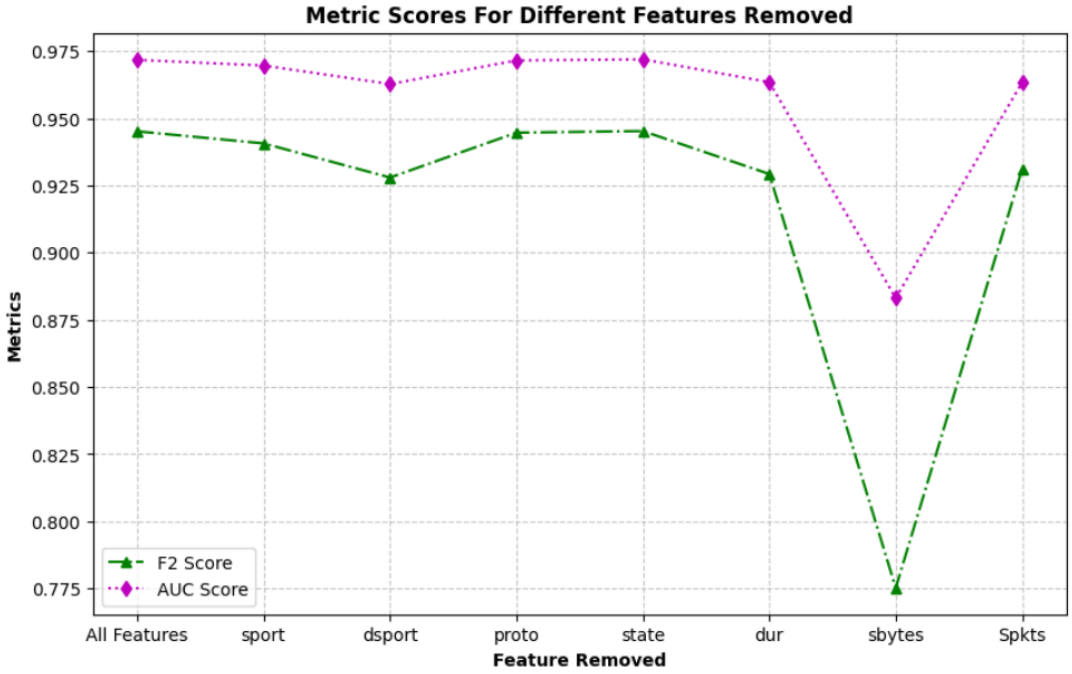
3.5 Πείραμα 4: Προσπάθεια μείωσης του αριθμού των χαρακτηριστικών στο Random Forest

Μια άλλη πειραματική διαδικασία είναι η προσπάθεια μείωσης των χαρακτηριστικών εισόδου στους ταξινομητές έτσι ώστε να απλοποιηθεί η διαδικασία ως προς το πλήθος των υπολογισμών αλλά και τον χρόνο εκτέλεσης. Στόχος είναι να διαγραφούν όσα χαρακτηριστικά δεν εξυπηρετούν ουσιαστικά το μοντέλο καθώς επηρεάζουν ελάχιστα, καθόλου, ή και ακόμη έχουν αρνητικό αντίκτυπο στα αποτελέσματα των μετρήσεων. Ως προεπιλεγμένο μοντέλο έχει επιλεγεί το Random Forest και το σύνολο δεδομένων αυτού του πειράματος είναι το Partial Dataset A.

3.5.1 Πρώτος Έλεγχός για μείωση των Διαστάσεων των Χαρακτηριστικών

Στο πείραμα έγινε επαναληπτικά η εξής διαδικασία. Αρχικά έγινε η προεπεξεργασία του συνόλου δεδομένου με τον ίδιο τρόπο όπως ακριβώς και στα προηγούμενα πειράματα.

Οπότε μετά από αυτό, υπάρχει το υποσύνολο δεδομένων A το οποίο είναι κατάλληλα επεξεργασμένο έτσι ώστε να ξεκινήσει το κομμάτι των μετρήσεων. Πρώτα από όλα έχουν καταγραφεί οι τιμές των μετρικών λαμβάνοντας υπόψη όλα τα χαρακτηριστικά για σκοπούς σύγκρισης. Ακολούθως έγιναν 7 διαφορετικές μετρήσεις όπου σε κάθε μια είχε αφαιρεθεί ένα μοναδικό χαρακτηριστικό από το σύνολο δεδομένων. Η πιο κάτω γραφική παράσταση (Σχήμα 3.5.1.1) παρουσιάζει τα αποτελέσματα των μετρήσεων, τα οποία υπάρχουν και στον Πίνακα 3.5.1.2

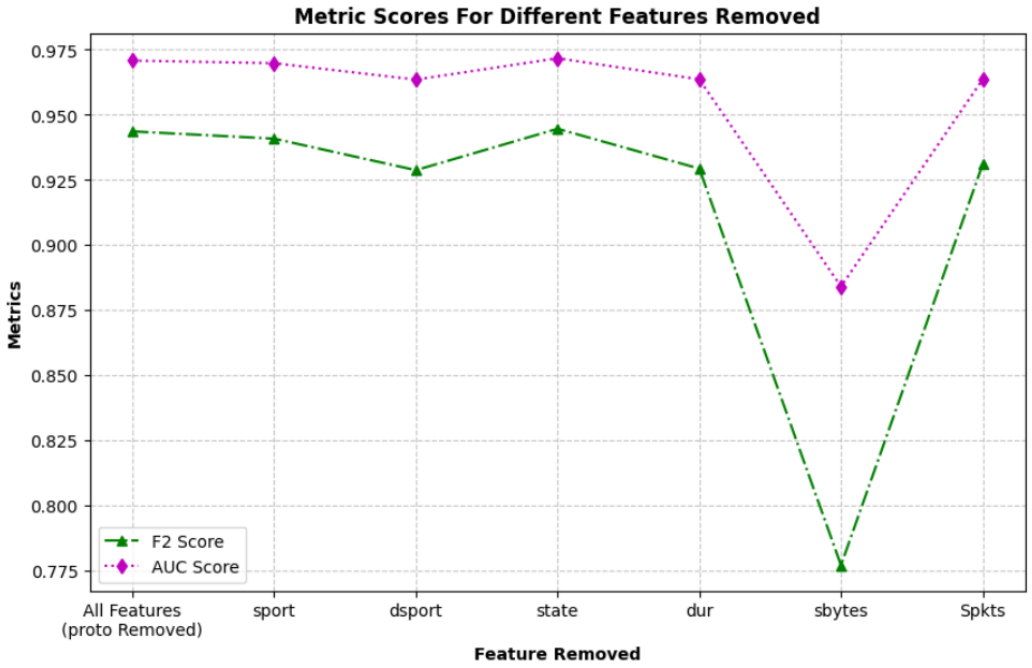


Σχήμα 3..5.1.1: Απόδοση Μετρικών με την αφαίρεση ενός πρώτου χαρακτηριστικού

Αυτό που παρατηρούμε είναι πως υπάρχουν 2 χαρακτηριστικά, το proto και το state, τα οποία η αφαίρεση τους οδηγεί σε πανομοιότυπα αποτελέσματα συγκριτικά με το αρχικό σύνολο χαρακτηριστικών. Εφόσον και 2 αυτά χαρακτηριστικά οδηγούν στις ίδιες μετρήσεις, έγινε αυθαίρετα η επιλογή της διαγραφής του χαρακτηριστικού proto από το σύνολο δεδομένων.

3.5.2 Δεύτερος Έλεγχός για μείωση των Διαστάσεων των Χαρακτηριστικών

Στη συνέχεια έγινε με τον ίδιο ακριβώς τρόπο η διαδικασία που είχε ακολουθηθεί προηγουμένως. Δηλαδή, στο επεξεργασμένο υποσύνολο δεδομένων A έγιναν 6 διαφορετικές μετρήσεις, όπου κάθε φορά διαγραφόταν ένα από τα εναπομείναντα 6 χαρακτηριστικά. Τα αποτελέσματα που προκύπτουν εμφανίζονται στην πιο κάτω γραφική παράσταση (Σχήμα 3.5.2.1), αλλά και στον Πίνακα 3.5.2.2

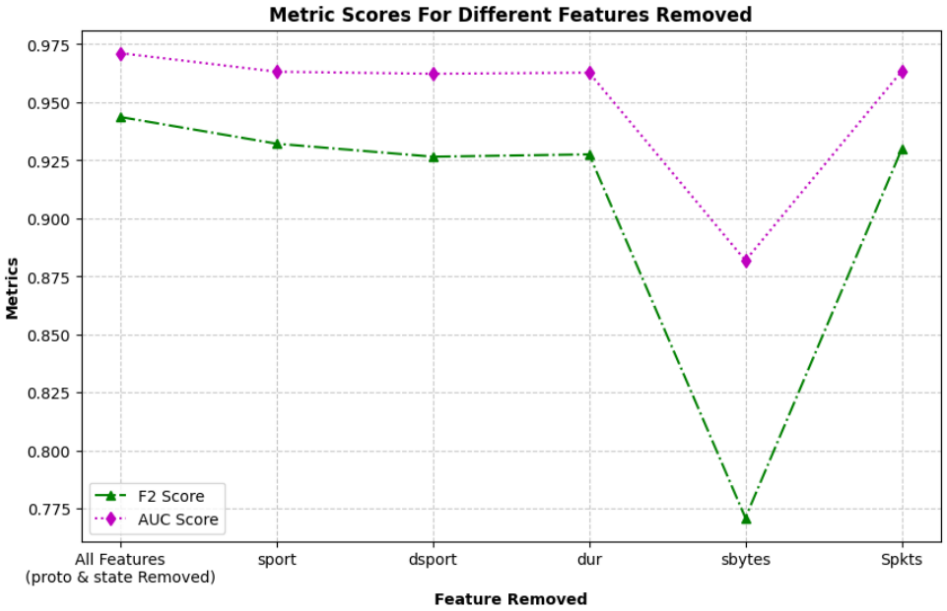


Σχήμα 3.5.2.1: Απόδοση Μετρικών με την αφαίρεση ενός δεύτερου χαρακτηριστικού

Όπως και προηγουμένως, έχει παρατηρηθεί πως η διαγραφή του χαρακτηριστικού "state" δεν επηρεάζει καθόλου τα αποτελέσματα όχι μόνο του F2 Score και toy AUC Score, αλλά και γενικά όλων των μετρικών. Οπότε εκτός από το χαρακτηριστικό "proto", μπορεί να θεωρηθεί ότι και η αφαίρεση του χαρακτηριστικού "state" θα απλοποιήσει την πολυπλοκότητα και τον χρόνο εκτέλεσης χωρίς να έχει τις ίδιες συνέπειες όσο αφορά την απόδοση του μοντέλου.

3.5.3 Τρίτος Έλεγχός για μείωση των Διαστάσεων των Χαρακτηριστικών

Επιπλέον έγινε ακόμη μια προσπάθεια για την εύρεση ενός τρίτου χαρακτηριστικού για να υπάρξει ακόμη περισσότερος περιορισμός των διαστάσεων του συνόλου δεδομένων. Όπως ακριβώς έγιναν τα βήματα πριν έτσι και τώρα, αφού πρώτα έχουν ληφθεί μετρήσεις χωρίς τα χαρακτηριστικά "proto" και "state" και στη συνέχεια καταγράφηκαν οι τιμές που προκύπτουν από την αφαίρεση των 5 χαρακτηριστικών που απέμειναν. Στον Πίνακα 3.5.3.2 υπάρχουν όλα όσα έχουν καταγραφεί και με βάση αυτά δημιουργείται η εξής γραφική (Σχήμα 3.5.3.1)



Σχήμα 3.5.3.1: Απόδοση Μετρικών με την αφαίρεση ενός τρίτου χαρακτηριστικού

Με βάση τα πιο πάνω βλέπουμε ότι η αφαίρεση ενός από τα χαρακτηριστικά που απέμειναν οδηγεί κάποιες φορές σε μικρή ενώ άλλες μεγάλη πτώση στα αποτελέσματα των μετρικών το οποίο δεν είναι επιθυμητό. Επομένως, συνοψίζοντας τα παραπάνω, καταλήγουμε στο συμπέρασμα ότι από το αρχικό σύνολο χαρακτηριστικών μπορούμε να αφαιρέσουμε το “proto” και το “state” χωρίς αυτό να επηρεάσει καθόλου την απόδοση του Random Forest. Ο λόγος που πιθανό να οφείλεται σε αυτό είναι επειδή τόσο το “proto” όσο και το “state” είναι χαρακτηριστικά κατηγοριοποίησης με πολλές διαφορετικές επιλογές, 16 και 134 αντίστοιχα, τα οποία με τη χρήση του label encoder φαίνεται να μην επηρεάζουν το μοντέλο. Με την διαγραφή οποιουδήποτε άλλου χαρακτηριστικού οι μετρικές παρουσιάζουν χειρότερα αποτελέσματα.

Πίνακας 3.5.1.2: Αφαίρεση ενός χαρακτηριστικού από το αρχικό σύνολο δεδομένων.

Metrics	**Feature Removed**							
	All Features	sport	dsport	proto	state	dur	sbytes	Spkts
Accuracy	0.9965	0.9962	0.9954	0.9964	0.9965	0.9955	0.9859	0.9959
Precision	0.9447	0.9383	0.9280	0.9429	0.9434	0.9292	0.7835	0.9411
Recall	0.9454	0.9413	0.9280	0.9451	0.9458	0.9294	0.7832	0.9288
F1 Score	0.9451	0.9398	0.9280	0.9440	0.9446	0.9293	0.7783	0.9349
F2 Score	0.9452	0.9407	0.9280	0.9447	0.9453	0.9293	0.7752	0.9312
AUC Score	0.9718	0.9697	0.9628	0.9716	0.9720	0.9635	0.8831	0.9634

Πίνακας 3.5.2.2: Αφαίρεση ενός δεύτερου χαρακτηριστικού από το αρχικό σύνολο δεδομένων.

Metrics	Feature Removed (no “proto”)						
	All Features	sport	dsport	state	dur	sbytes	Spkts
Accuracy	0.9964	0.9961	0.954	0.9964	0.9955	0.9860	0.9958
Precision	0.9446	0.9374	0.9261	0.9415	0.9280	0.7843	0.9383
Recall	0.9434	0.9417	0.9294	0.9454	0.9296	0.7751	0.9293
F1 Score	0.9440	0.9395	0.9278	0.9434	0.9288	0.7797	0.9337
F2 Score	0.9436	0.9408	0.9287	0.9446	0.9293	0.7769	0.9310
AUC Score	0.9708	0.9698	0.9635	0.9717	0.9636	0.8840	0.9636

Πίνακας 3.5.3.2: Αφαίρεση ενός τρίτου χαρακτηριστικού από το αρχικό σύνολο δεδομένων.

Metrics	Feature Removed (no “proto” & “state”)					
	All Features	sport	dsport	dur	sbytes	Spkts
Accuracy	0.9963	0.9961	0.9953	0.9953	0.9853	0.9956
Precision	0.9408	0.9490	0.9250	0.9263	0.7696	0.9339
Recall	0.9443	0.9279	0.9269	0.9278	0.7713	0.9287
F1 Score	0.9425	0.9384	0.9260	0.9271	0.770	0.9313
F2 Score	0.9436	0.9321	0.9265	0.9275	0.771	0.9297
AUC Score	0.9711	0.9631	0.9622	0.9627	0.8818	0.9633

3.6 Πείραμα 5: Dimensionality Reduction με Autoencoders.

Σε όλες τις μέχρι τώρα πειραματικές διαδικασίες, γινόταν αποκλειστικά χρήση τεχνικών επιβλεπόμενης μηχανικής μάθησης πάνω στο σύνολο των 8 χαρακτηριστικών (συμπεριλαμβάνεται και η ετικέτα) που είχαν προεπιλεχθεί.

3.6.1 Προεπεξεργασία Δεδομένων, Δομή Autoencoder και Πειραματική Διαδικασία

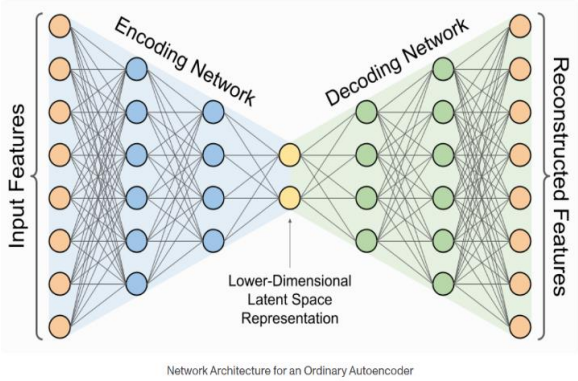
Αντιθέτως, σε αυτή την πειραματική διαδικασία θα εξεταστεί το πως η χρήση ενός Autoencoder, που ανήκει στην κατηγορία της μη επιβλεπόμενης μηχανικής μάθησης, μπορεί να αποτελέσει ένα αξιόπιστο εργαλείο για τον περιορισμό των διαστάσεων των χαρακτηριστικών ενός συνόλου δεδομένων. Το σύνολο δεδομένων θα αποτελείται από τα 49 χαρακτηριστικά του UNSW-NB15 Dataset. Στόχος είναι με τη χρήση ενός autoencoder, να περιοριστεί ο αριθμός των διαστάσεων σε κάτω από 10, τα οποία θα αποτελέσουν της είσοδο σε ένα Random Forest ταξινομητή και να ληφθούν μετρήσεις του πως η συγκεκριμένη τεχνική μείωσης διαστάσεων επηρεάζει την απόδοση του μοντέλου.

Προεπεξεργασία Δεδομένων

Η πειραματική διαδικασία που ακολουθείται είναι η εξής. Αρχικά, γίνεται το διάβασμα του υποσυνόλου δεδομένου A (Partial Dataset A) που αποτελείται από 700000 εγγραφές και 49 χαρακτηριστικά. Στη συνέχεια αφαιρούνται οι 67 προβληματικές εγγραφές των χαρακτηριστικών 2.'sport' και 4.'dsport' που παρουσιάζονται αναλυτικά στον πίνακα 3.1.2. Επίσης καθώς οι AutoEncoders κατατάσσονται στη μη επιβλεπόμενη μηχανική μάθηση, τα χαρακτηριστικά 48.'attack_cat' και 49.'Label' αφαιρούνται, εφόσον παρέχουν πληροφορία για την κλάση στην οποία ανήκει η κάθε εγγραφή. Από τα 47 χαρακτηριστικά που απέμειναν, υπάρχουν 5 που είναι κατηγορηματικά χαρακτηριστικά τα οποία είναι τα 1.'srcip', 3.'dstip', 5.'proto', 6.'state', 14.'service'. Για την μετατροπή τους σε αριθμητικές τιμές χρησιμοποιήθηκε Label Encoder και ακολούθως έγινε χρήση ενός Standard Scaler έτσι ώστε όλα τα χαρακτηριστικά να έχουν παρόμοια βαρύτητα και να παίρνουν τιμές που να κυμαίνονται γύρω από το 0. (Μέση τιμή μ=0 , διασπορά σ=1)

Δομή Autoencoder και τρόπος εκπαίδευσης του

Όταν η προεπεξεργασία των δεδομένων έχει ολοκληρωθεί, μέσω ενός autoencoder θα γίνει προσπάθεια περιορισμού των διαστάσεων από 47 σε κάτω από 10. Ο autoencoder του πειράματος έχει τη δομή του σχήματος 1 και πιο ειδικά έχει κατασκευαστεί με βάση τα χαρακτηριστικά που περιγράφονται πιο κάτω. Ως είσοδο στον autoencoders δίνονται τα Input Features, που είναι τα 47 χαρακτηριστικά του συνόλου δεδομένων. Αμέσως μετά είναι συνδεδεμένος ο Encoder, που αποτελείται από 2 κρυφά στρώματα με 32 και 16 νευρώνες αντίστοιχα. Το μεσαίο στρώμα αποτελεί κομμάτι τόσο του Encoder (τελευταίο βήμα) αλλά και του Decoder(πρώτο βήμα),αποτελεί το bottleneck του μοντέλου, είναι ο χώρος στον οποίο υπάρχει ο μικρότερος αριθμός νευρώνων και άρα ο ελάχιστος αριθμός χαρακτηριστικών. Στο πείραμα το συγκεκριμένο στρώμα θα κυμαίνεται από 4 ως 10 νευρώνες και η πειραματική διαδικασία θα εκτελεστεί 7 διαφορετικές φορές, όπου για κάθε φορά η τιμή του συγκεκριμένου στρώματος θα είναι διαφορετική στο εύρος 4-10. Στο σημείο αυτό τα χαρακτηριστικά της εισόδου έχουν κωδικοποιηθεί και περιοριστεί στο μικρότερο αριθμό διαστάσεων που επιτρέπει το μοντέλο. Μετά από αυτό ξεκινά η διαδικασία της αποκωδικοποίησης, όπου μέσω ενός Decoder τα χαρακτηριστικά αναμένεται να ξαναφτάσουν στον αρχικό αριθμό διαστάσεων με όσο το δυνατό λιγότερη χαμένη πληροφορία. Ο Decoder είναι φτιαγμένος με τα ίδια στρώματα όπως στον Encoder, δηλαδή από 2 κρυφά στρώματα με 16 και 32 νευρώνες, που καταλήγουν στο τελικό στρώμα με τα Reconstructed Features των 47 νευρώνων. Όλα τα στρώματα έχουν ως συνάρτηση ενεργοποίησης (Activation Function) την ReLu [92].



Σχήμα 3.6.1:Δομή AutoEncoder[91]

Για την εκπαίδευση του autoencoder χρησιμοποιήθηκαν 3 διαφορετικά Train/Test Split Ratio με τιμές 80/20,75/25,70/30. Ο optimizer είναι ένας αλγόριθμος ο οποίος ανανεώνει τα βάρη στο μοντέλο ανάλογα με την τιμή της συνάρτησης λάθους (Loss Function). Ως optimizer έχει επιλεγεί ο Adam (**Ad**aptive **M**oment **E**stimation) . Η συνάρτηση λάθους υπολογίζει το σφάλμα ανακατασκευής του συνόλου μετά την διαδικασία αποκωδικοποίησης (Decoding) συγκριτικά με το αρχικό σύνολο. Ως Loss Function έχει επιλεγεί το Mean Square Error – MSE (κεφάλαιο 1.8.2). Για την εκπαίδευση χρησιμοποιήθηκαν επίσης οι παράμετροι epochs = 50, batch_size = 256 , Shuffling = True. Οι epochs εκφράζουν το πλήθος των πλήρων διελεύσεων του συνόλου δεδομένων εκπαίδευσης. Το 50 δηλώνει πως το μοντέλο θα εκπαιδευτεί ελέγχοντας ολόκληρο το σύνολο δεδομένων 50 φορές. Το batch_size καθορίζει για κάθε πόσους ελέγχους θα εφαρμόζεται η ανανέωση των βαρών στο μοντέλο. Για batch_size = 256 σημαίνει πως το μοντέλο θα επεξεργάζεται 256 δείγματα και στη συνέχεια θα υπάρχει αλλαγή βαρών, λαμβάνοντας υπόψη τα Loss Function των δειγμάτων. Το shuffle είναι μια παράμετρος η οποία καθορίζει το αν η σειρά των δεδομένων εισόδου θα αλλάξει μετά από κάθε εποχή. Με τη δήλωση του Shuffle = True τα δεδομένα ανακατεύονται, κάτι που πιθανώς θα εμποδίσει το μοντέλο από το να δημιουργήσει πλαστά μοτίβα με βάση την σειρά που φτάνουν τα δείγματα.

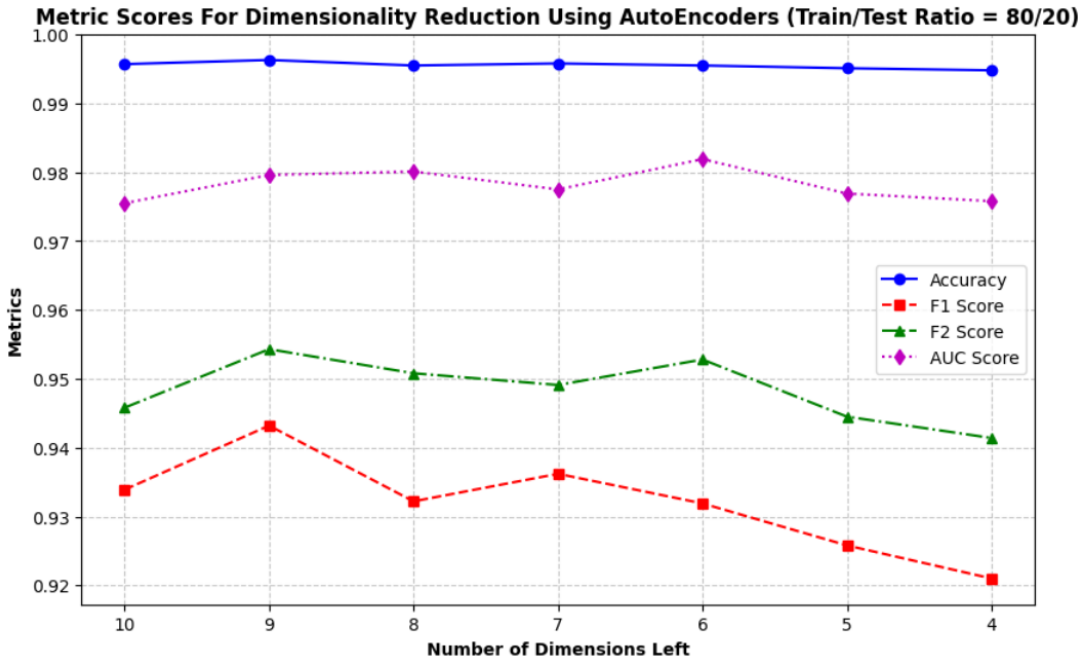
Χρήση Encoder για μείωση αριθμού διαστάσεων και Αξιολόγηση απόδοσης σε Random Forest Ταξινομητή.

Για κάθε ένα από τα 3 Train/Test Ratio εκπαιδεύονταν 7 νέοι autoencoders (με βάση τις πιο πάνω ιδιότητες) οπότε τελικά δημιουργήθηκαν 21 νέα μοντέλα. Ο αρχικός στόχος ήταν και είναι ο περιορισμός των διαστάσεων από 47 σε κάτω από 10 κάτι που μπορεί να επιτευχθεί με την αξιοποίηση του Encoder, δηλαδή του πρώτου μέρους των μοντέλων που έχουν παραχθεί. Τα 47 χαρακτηριστικά του αρχικού συνόλου δεδομένων θα περάσουν μέσα από τον Encoder έτσι ώστε να μειωθεί η διάσταση των χαρακτηριστικών όσο είναι ο αριθμός των νευρώνων στο μεσαίο στρώμα. Με αυτό τον τρόπο ο αριθμός των χαρακτηριστικών μειώθηκε σε κάτω από 10 και αποτελούν πλέον το νέο σύνολο δεδομένων που θα λειτουργήσει ως η είσοδος σε ένα Random Forest ταξινομητή που έχει δημιουργηθεί όπως στο πείραμα 3.2.5.

Στους πίνακες 3.6.2,3.6.3,3.6.4 παρουσιάζεται η απόδοση της συγκεκριμένης τεχνικής για τη μείωση των διαστάσεων για διαφορετικές τιμές του Train/Test Split Ratio αλλά και με βάση των αριθμό των διαστάσεων που τελικά απέμειναν μετά την χρήση του Autoencoder

Πίνακας 3.6.2: Αποτελέσματα Μετρικών σε Random Forest με χρήση autoencoder για μείωση διαστάσεων (Train/Test Ratio 80/20)

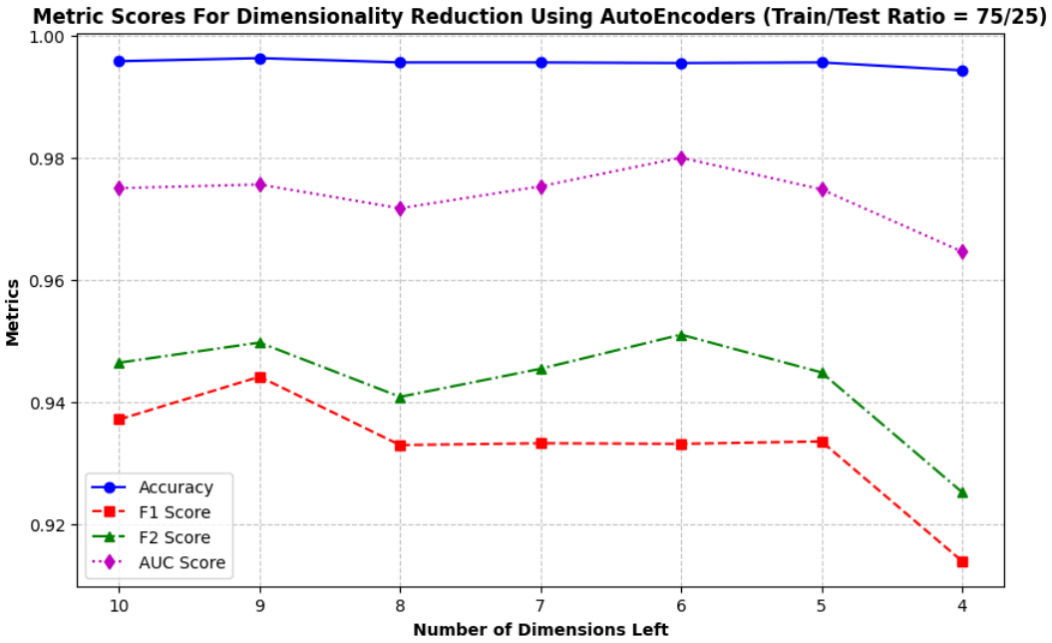
Metrics	Train/Test Ratio = 80/20, Μείωση Διαστάσεων σε d=? με AutoEncoder						
	d=10	d=9	d=8	d=7	d=6	d=5	d=4
Accuracy	0.9957	0.9963	0.9955	0.9958	0.9955	0.9951	0.9948
Precision	0.9148	0.9254	0.9028	0.9155	0.8991	0.8962	0.8888
Recall	0.9539	0.9618	0.9636	0.9579	0.9673	0.9575	0.9556
F1 Score	0.9339	0.9432	0.9322	0.9362	0.9319	0.9258	0.9210
F2 Score	0.9458	0.9543	0.9508	0.9491	0.9528	0.9445	0.9414
AUC Score	0.9755	0.9796	0.9801	0.9775	0.9819	0.9769	0.9758



Σχήμα 3.6.5: Απόδοση Μετρικών σε Random Forest με χρήση autoencoder για μείωση διαστάσεων (Train/Test Ratio 80/20)

Πίνακας 3.6.3: Αποτελέσματα Μετρικών σε Random Forest με χρήση autoencoder για μείωση διαστάσεων (Train/Test Ratio 75/25)

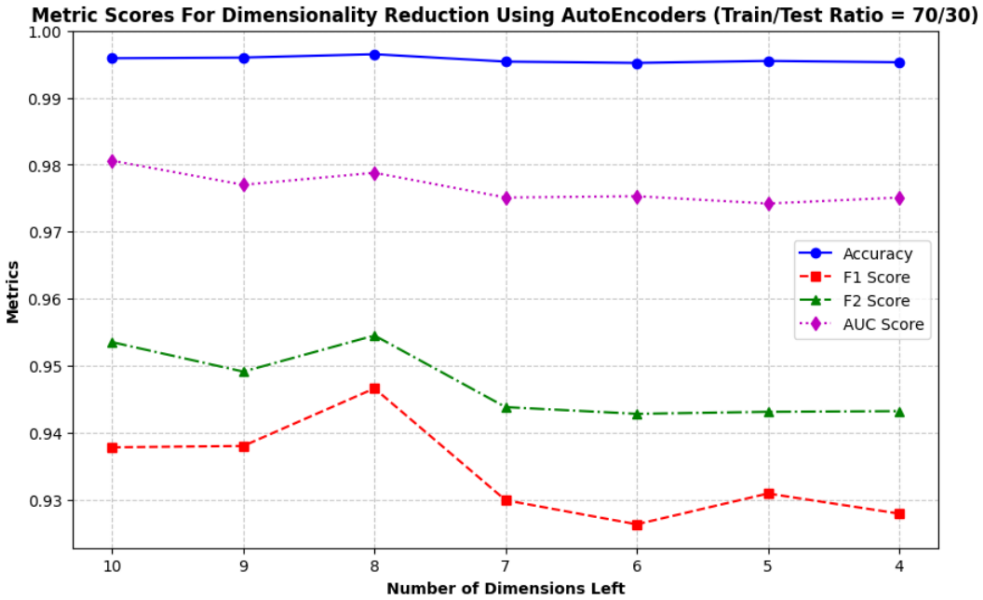
Metrics	Train/Test Ratio = 75/25, Μείωση Διαστάσεων σε d=? με AutoEncoder						
	d=10	d=9	d=8	d=7	d=6	d=5	d=4
Accuracy	0.9959	0.9964	0.9957	0.9957	0.9956	0.9957	0.9944
Precision	0.9222	0.9349	0.9200	0.9138	0.9047	0.9153	0.8957
Recall	0.9528	0.9537	0.9463	0.9538	0.9635	0.9526	0.9329
F1 Score	0.9372	0.9442	0.9330	0.9333	0.9332	0.9336	0.9140
F2 Score	0.9465	0.9498	0.9409	0.9455	0.9511	0.9449	0.9253
AUC Score	0.9751	0.9757	0.9718	0.9754	0.9801	0.9749	0.9647



Σχήμα 3.6.6: Απόδοση Μετρικών σε Random Forest με χρήση autoencoder για μείωση διαστάσεων (Train/Test Ratio 75/25)

Πίνακας 3.6.4: Αποτελέσματα Μετρικών σε Random Forest με χρήση autoencoder για μείωση διαστάσεων (Train/Test Ratio 70/30)

Metrics	**Train/Test Ratio = 70/30, Μείωση Διαστάσεων σε d=? με AutoEncoder**						
	d=10	d=9	d=8	d=7	d=6	d=5	d=4
Accuracy	0.9959	0.9960	0.9965	0.9954	0.9952	0.9955	0.9953
Precision	0.9127	0.9200	0.9338	0.9076	0.9001	0.9112	0.9035
Recall	0.9643	0.9567	0.9598	0.9533	0.9541	0.9514	0.9537
F1 Score	0.9378	0.9380	0.9465	0.9299	0.9263	0.9309	0.9279
F2 Score	0.9535	0.9491	0.9545	0.9438	0.9428	0.9431	0.9432
AUC Score	0.9806	0.9770	0.9788	0.9751	0.9753	0.9742	0.9751



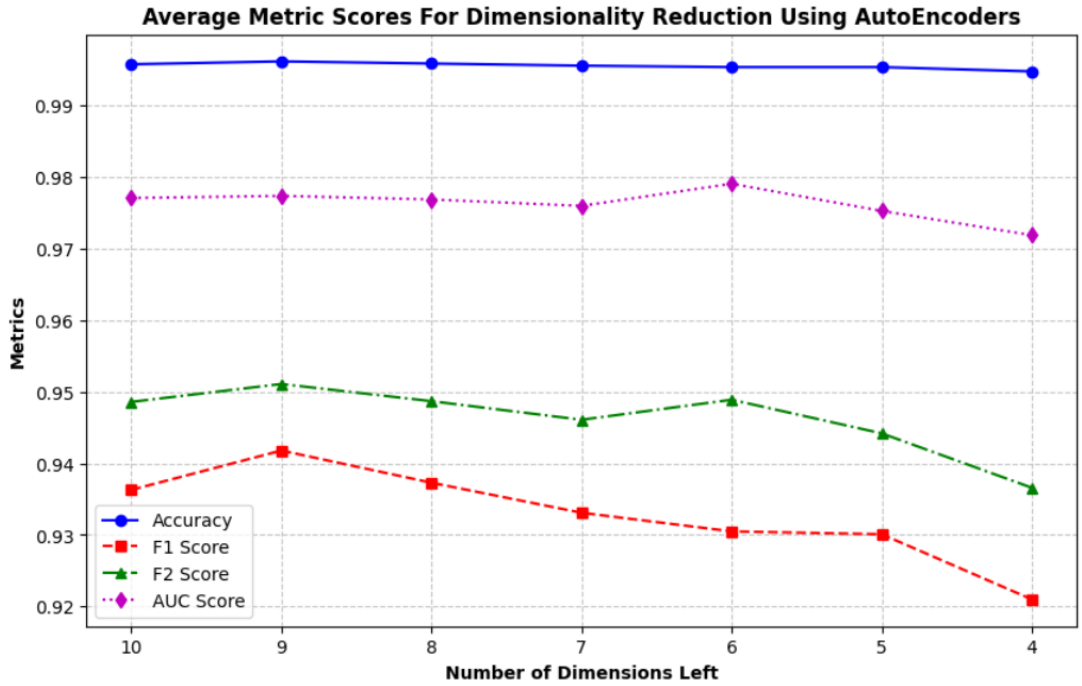
Σχήμα 3.6.7: Απόδοση Μετρικών σε Random Forest με χρήση autoencoder για μείωση διαστάσεων (Train/Test Ratio 70/30)

3.6.2 Σύγκριση Απόδοσης για τις Διαφορετικές πιθανές Τιμές Χαρακτηριστικών που Απέμειναν

Στον πίνακα 3.6.2.1 αλλά και στην γραφική παράσταση (Σχήμα 3.6.2.2) παρουσιάζεται ο μέσος όρος των αποτελεσμάτων των μετρήσεων της χρήσης του autoencoder για τις 3 διαφορετικές περιπτώσεις που εξετάστηκαν.

Πίνακας 3.6.2.1: Μέσος Όρος 3 πειραμάτων των Μετρικών σε Random Forest με χρήση διαφορετικών autoencoder (3.6.1.1,3.6.1.2,3.6.1.3) για μείωση διαστάσεων

Metrics	**Μέσος Όρος 3 Μετρήσεων Μείωση Διαστάσεων σε d=? με AutoEncoder**						
	d=10	d=9	d=8	d=7	d=6	d=5	d=4
Accuracy	0.9958	0.9962	0.9959	0.9956	0.9954	0.9954	0.9948
Precision	0.9166	0.9267	0.9189	0.9123	0.9013	0.9076	0.8960
Recall	0.9570	0.9574	0.9565	0.9550	0.9616	0.9538	0.9474
F1 Score	0.9363	0.9418	0.9373	0.9331	0.9305	0.9301	0.9210
F2 Score	0.9486	0.9511	0.9487	0.9461	0.9489	0.9442	0.9366
AUC Score	0.9771	0.9774	0.9769	0.9760	0.9791	0.9753	0.9719



Σχήμα 3.6.2.2: Μέσος Όρος Αποδόσεων των Μετρικών των 3 πειραμάτων σε Random Forest με χρήση διαφορετικών autoencoder (3.6.1.1,3.6.1.2,3.6.1.3) για μείωση διαστάσεων

Λαμβάνοντας υπόψη τις 3 διαφορετικές γραφικές που προέκυψαν από τις μετρήσεις αλλά και τη γραφική 3.4 στην οποία παρουσιάζεται ο μέσος όρος των πειραματικών διεργασιών, παρατηρούμε ότι δεν υπάρχει κάποιο σταθερό μοτίβο ως προς τον αριθμό των χαρακτηριστικών που απέμειναν συγκριτικά με την απόδοση των μετρικών. Σαν γενική εικόνα ισχύει η αναμενόμενη εκτίμηση του ότι όσο μειώνεται ο αριθμός των διαστάσεων, τόσο περισσότερη πληροφορία χάνεται και άρα να πέφτει η απόδοση στις μετρικές, κάτι που επαληθεύεται ως ένα σημείο. Χαρακτηριστικότερο παράδειγμα είναι οι «4 διαστάσεις» που είναι η ελάχιστη δυνατή τιμή στο πείραμα και εμφανίζει τα χειρότερα αποτελέσματα. Από την άλλη για τις υπόλοιπες τιμές όσο μειώνονται τα χαρακτηριστικά υπάρχουν αυξομειώσεις στην απόδοση κυρίως των μετρικών F1 Score, F2 Score και AUC Score. Ίσως και οι «5 Διαστάσεις» να εμφανίζουν αποδόσεις λίγο χειρότερες από τις αναμενόμενες, αλλά από 6 εως 10 διαστάσεις τα αποτελέσματα στις μετρικές F2 Score και AUC Score είναι πανομοιότυπα.

3.6.3 Σύγκριση Απόδοσης με Random Forest ταξινομητή προεπιλεγμένων 7 διαστάσεων

Τέλος, στον πίνακα 3.6.3.1 παρουσιάζονται αναλυτικά οι τιμές όλων των μετρικών του Random Forest ταξινομητή του πειράματος 3.2.5 στον οποίο είχαν προεπιλεχθεί οι 7 διαστάσεις χαρακτηριστικών όπως και η μέση τιμή των 3 πειραμάτων για τον Random Forest ταξινομητή 3.6.2, στον οποίο οι 7 διαστάσεις των χαρακτηριστικών είχαν δημιουργηθεί αξιοποιώντας έναν autoencoder.

Πίνακας 3.6.3.1: Σύγκριση απόδοσης R.F. Μοντέλου 7 διαστάσεων πειράματος 3.2.5 με 3.6.2

	Μοντέλο R.F. 7 Διαστάσεων με Default Υπερπαραμέτρους	**Μοντέλο R.F. με Autoencoder για Μείωση σε 7 Διαστάσεις**	**Διαφορά**
Accuracy	0.9965	0.9956	-0.0009
Precision	0.9437	0.9123	-0.0314
Recall	0.9454	0.9550	+0.0096
F1 Score	0.9445	0.9331	-0.0114
F2 Score	0.945	0.9461	+0.0011
AUC Score	0.9718	0.9760	+0.0042

Αυτό που παρατηρείται είναι πως όσο αφορά τις μετρικές που ενδιαφέρουν περισσότερο τη δική μας περίπτωση, Η χρήση του autoencoder βελτίωσε ελάχιστα την απόδοση των F2 Score και AUC Score συγκριτικά με τις προεπιλεγμένες διαστάσεις. Από την άλλη, στην μετρική Precision και F1 Score τα αποτελέσματα σε αυτή την πειραματική διαδικασία είναι χειρότερα από προηγουμένως.

Το συγκεκριμένο μας δείχνει πως οι 7 διαστάσεις που είχαν προεπιλεχθεί φαίνονται ως αρκετά σωστές επιλογές καθώς η απόδοση τους είναι αρκετά ψηλή. Όμως η υλοποίηση ενός απλού autoencoder φαίνεται ότι μπορεί να βελτιώσει την απόδοση, οπότε η εύρεση ενός βέλτιστου autoencoder πιθανό να βελτιώσει την απόδοση του μοντέλου μειώνοντας ακόμη περισσότερο τον αριθμό των διαστάσεων.

Κεφάλαιο 4: Συμπεράσματα - Βελτιώσεις

4.1 Συμπεράσματα από Πειραματικές Διαδικασίες

Μέσα από τις πειραματικές διαδικασίες έχει υλοποιηθεί μια σειρά από μετρήσεις και συγκρίσεις διαφόρων μοντέλων, υπερπαραμέτρων ,χαρακτηριστικών όπως και συνδυασμός όλων αυτών με στόχο τον όσο το δυνατό καλύτερο εντοπισμό ανωμαλιών σε ένα δίκτυο. Η έρευνα βασίστηκε σε τεχνικές επιβλεπόμενης μάθησης – Supervised Learning εκτός στην περίπτωση του πειράματος 5, που αξιοποιήθηκαν οι autoencoders και αποτελούν κομμάτι του κλάδου της μη επιβλεπόμενης μάθησης – Unsupervised Learning

Τα συμπεράσματα τα οποία προέκυψαν σε κάθε πείραμα είναι τα εξής:

4.1.1 Συμπεράσματα Πείραματος 1: Μοντέλα Επιβλεπόμενης Μηχανικής Μάθησης

Η Πειραματική διαδικασία 1 επαλήθευσε το συμπέρασμα της έρευνας 2.2, η οποία κατέληξε στο ότι το Random Forest είναι ο ιδανικός ταξινομητής για το σύνολο δεδομένων UNSW-NB15 που τυγχάνει επεξεργασίας. Από όλα τα μοντέλα μάθησης που δοκιμάστηκαν στο πείραμα στα 5 υποσύνολα δεδομένων, το Random Forest είχε σε όλες τις περιπτώσεις τη πιο ψηλή απόδοση σε F2 Score και AUC Score πρωτίστως αλλά και σε Accuracy και F1 Score ως δευτερεύοντα κάτι που το καθιστά τον ιδανικότερο ταξινομητή στην κατηγορία της επιβλεπόμενης μάθησης.

Το Decision Tree ήταν με διαφορά το γρηγορότερο μοντέλο ως προς χρόνο εκπαίδευσης και πρόβλεψης του συνόλου δεδομένων. Τα Logistic Regression , Random Forest και K-Nearest Neighbors χρειάστηκαν περισσότερο χρόνο εκπαίδευσης, αλλά όχι σε βαθμό που αυτό να επηρεάσει αρνητικά την επιλογή των μοντέλων από των χρήστη ως ιδανικούς ταξινομητές. Το τελευταίο ισχύει για το SVC μοντέλο, καθώς ο χρόνος εκπαίδευσης του ήταν πάνω από 10 φορές μεγαλύτερος σε σχέση με τα υπόλοιπα μοντέλα χωρίς αυτό να αντικατοπτρίζεται στην απόδοση του, αφού ήταν ένα από τα χειρότερα μοντέλα σε όλες τις μετρικές.

4.1.2 Συμπεράσματα Πείραματος 2: Voting Classifier

Από όλες τις μετρήσεις με τη χρήση του Soft , του Hard και του Custom Voting, μόνο το AUC Score στην περίπτωση του Custom Voting είχε πιο ψηλή τιμή συγκριτικά με το Random Forest στο πείραμα 1. Όμως το Custom Voting παρουσίασε πολλά False Positive αποτελέσματα και υστερούσε πολύ στις μετρικές που εξαρτώνται από αυτά, με χαρακτηριστικότερο παράδειγμα το Precision. Στη Soft Voting υλοποίηση εμφανίζονται

ψηλότερες αποδόσεις συγκριτικά με το Hard Voting σε όλες τις μετρικές αλλά και συγκριτικά με το Custom Voting (με εξαίρεση το AUC Score), χωρίς όμως οι τιμές να ξεπερνούν τις τιμές των μετρικών του Random Forest.

Η εισαγωγή βαρών στον Soft Voting Classifier βοήθησε την αύξηση της απόδοσης όλων των μετρικών, πλησίασαν την απόδοση του Random Forest αλλά ποτέ δεν την ξεπέρασαν σε καμία μετρική. Γενικά συμπεραίνουμε πως ο Voting Classifier εξομαλύνει το χάσμα μεταξύ των αποδόσεων όλων των μοντέλων, δημιουργώντας έναν ταξινομητή με απόδοση περίπου τον μέσο όρο όλο αυτών.

4.1.3 Συμπεράσματα Πείραματος 3: Υπερπαραμέτροι σε Random Forest

Η πειραματική διεργασία 3 φανέρωσε πως η εισαγωγή των υπερπαραμέτρων σε ένα Random Forest μοντέλο αυξάνει την απόδοση των μετρικών που έχουμε θέσει ως στόχο, παρόλο που οι μετρικές αυτές είχαν ήδη αρκετά ψηλή απόδοση. Αφού είχαν πρώτα εντοπιστεί από ένα προεπιλεγμένο σύνολο τιμών οι βέλτιστες τιμές, η εκπαίδευση ενός νέου Random Forest μοντέλου με τις συγκεκριμένες υπερπαραμέτρους έφερε αύξηση 1.5% στο F2 Score (από 0.945 σε 0.9598) και 2% στο AUC Score, το οποίο ξεπέρασε το 99% (από 0.9718 σε 0.9912)

Η τιμή του κατωφλίου επίσης φαίνεται να επηρεάζει κατά πολύ την απόδοση των μοντέλων. Οι βέλτιστες υπερπαραμέτροι υπολογίστηκαν με τιμή κατωφλίου 0.5 και πράγματι έδιναν καλύτερα αποτελέσματα από το μοντέλο που εκπαιδεύτηκε χωρίς υπερπαραμέτρους. Όμως για τιμές κατωφλίου μικρότερες από 0.35 , το F2 Score είναι ψηλότερο στο Random Forest χωρίς υπερπαραμέτρους συγκριτικά με το Random Forest μοντέλο που είχε. Αν όμως η εύρεση των τιμών των υπερπαραμέτρων γινόταν με διαφορετική τιμή κατωφλίου, τότε σχεδόν βέβαια οι βέλτιστες τιμές θα ήταν διαφορετικές και τα αποτελέσματα θα ήταν αρκετά αλλαγμένα.

4.1.4 Συμπεράσματα Πειράματος 4: Μείωση Διαστάσεων Συνόλου Δεδομένων

Το πείραμα 4 έδειξε πως τα κατηγορηματικά χαρακτηριστικά 5.”proto” και 6.”state” δεν συμβάλουν στην αύξηση της απόδοσης του Random Forest μοντέλου και πως η διαφορά στην απόδοση μετά την αφαίρεση τους από το σύνολο δεδομένων κρίνεται αμελητέα. Αυτό συμβαίνει λόγω των πολλών διαφορετικών πιθανόν επιλογών στις κατηγορίες που μπορούν να ανήκουν οι εγγραφές για τα συγκεκριμένα πεδία, (16 και 134 αντίστοιχα) και άρα δεν μπορεί να παρθεί ικανοποιητική πληροφορία μέσα από αυτά τα πεδία.

Όλα τα υπόλοιπα χαρακτηριστικά όπως φαίνεται και από το 3.5.3, έστω και ελάχιστα, επηρεάζουν θετικά την απόδοση του εκπαιδευόμενου μοντέλου και πως πιθανή αφαίρεση τους θα μειώσει τις τιμές όλων των μετρικών.

4.1.5 Συμπεράσματα Πειράματος 5: Χρήση Autoencoders για Dimensionality Reduction

Η πειραματική διεργασία 5 είχε διαφορές σε σχέση με τις προηγούμενες καθώς ήταν η μόνη η οποία δεν βασίστηκε εξολοκλήρου σε Supervised Learning μεθόδους. Σε ένα μεγάλο σύνολο δεδομένων, ο αριθμός των χαρακτηριστικών επηρεάζει σε σημαντικό βαθμό τον χρόνο εκτέλεσης και πρόβλεψης ενός μοντέλου. Η χρήση των autoencoders και η αξιοποίηση του Encoder για την μείωση των διαστάσεων φανέρωσε ότι πως δεν είναι ανάγκη να υπάρξει προεπιλογή συγκεκριμένων χαρακτηριστικών. Η υλοποίηση του autoencoder δεν έγινε με την κατάλληλη παραμετροποίηση έτσι ώστε να βρεθεί η βέλτιστη λύση αλλά ακόμη και έτσι όταν μειώθηκαν οι διαστάσεις σε 7 (όσο δηλαδή και τα πειράματα με τα προεπιλεγμένα μοντέλα) οι αποδόσεις των μετρικών F2 Score και AUC Score σε ένα Random Forest ταξινομητή ήταν λίγο καλύτερες από όσο υπολογίστηκε προηγουμένως. Επίσης στην συγκεκριμένη περίπτωση δεν χρειάστηκε η οποιαδήποτε διαδικασία επιλογής χαρακτηριστικών αλλά ο αλγόριθμος από μόνος του «εκπαιδεύτηκε» ώστε να δημιουργεί αυτά που θεωρεί τα ιδανικά χαρακτηριστικά.

Παράλληλα, με την συγκεκριμένη μεθοδολογία υπάρχει μεγάλη ευελιξία ως προς την επιλογή του τελικού αριθμού χαρακτηριστικών που επιλέγει ο χρήστης. Ο autoencoder πάντα παραμένει ο ίδιος και το μόνο που αλλάζει είναι ο αριθμός των νευρώνων στο μεσαίο στρώμα. Τέλος, το πείραμα γενικά απέδειξε πως ακόμη και σε Supervised Μοντέλα, υπάρχει η δυνατότητα της αξιοποίησης Unsupervised Τεχνικών πριν την εκπαίδευση των ταξινομητών έτσι ώστε να προκύψουν ακόμη καλύτερα αποτελέσματα, όπως για παράδειγμα ένας encoder για τον περιορισμό των διαστάσεων των χαρακτηριστικών.

4.2 Προτάσεις για Μελλοντική Μελέτη - Βελτιώσεις

Ο κλάδος της μηχανικής μάθησης και τα εργαλεία που παρέχει εξελίσσονται εκθετικά και ήδη υπάρχουν άπειρες διαφορετικές προσεγγίσεις ως προς την ανάπτυξη των μοντέλων ανάλογα με τη χρήση. Οι πειραματικές διεργασίες έγιναν με τη χρήση μοντέλων επιβλεπόμενης μηχανικής μάθησης σε συνδυασμό με την εισαγωγή υπερπαραμέτρων και με την προεπιλογή συγκεκριμένων χαρακτηριστικών από το σύνολο δεδομένων.

Κάποιες προτάσεις για Μελλοντική Μελέτη - Βελτιώσεις είναι οι εξής:

- Έλεγχος διαφορετικών τιμών και υπερπαραμέτρων σε Random Forest Μοντέλα για την εύρεση ακόμη καλύτερων τιμών. Στόχος να είναι πάντα όσο το δυνατό πιο

χαμηλός αριθμός των False Negative προβλέψεων, δηλαδή να υπάρχουν ανωμαλίες οι οποίες να μην ανιχνεύονται.

- Εισαγωγή υπερπαραμέτρων σε όλα τα μοντέλα και όχι μόνο στο Random Forest και έλεγχος του πως επηρεάζουν την απόδοση.
- Η επιλογή των χαρακτηριστικών βασίστηκε στη έρευνα 2.2 η οποία και αυτή βασίστηκε στο πρωτόκολλο Netflow της Cisco. Να υπάρξει έλεγχος αν υπάρχουν άλλα χαρακτηριστικά στο σύνολο δεδομένων UNSW-NB15 των οποίων η επιλογή τους θα αυξήσει της απόδοση των μετρικών.
- Έγινε μια πρώτη υλοποίηση ενός autoencoder που ανήκει στην κατηγορία με επιβλεπόμενης μάθησης. Να γίνουν περισσότερες υλοποιήσεις με τη χρήση τεχνικών Μη Επιβλεπόμενης Μηχανικής Μάθησης (Unsupervised Learning) και ημί-επιβλεπόμενης Μηχανικής Μάθησης (Semi-Supervised Learning) και σύγκριση των αποτελεσμάτων. Ένα χαρακτηριστικό παράδειγμα είναι η υλοποίηση GAN (Generative Adversarial Networks) μοντέλων και έλεγχος της αξιοπιστίας τους σε ένα σύνολο δεδομένων το οποίο εμφανίζει ένα μικρό ποσοστό ανωμαλιών.
- Το σύνολο δεδομένων αποτελείται από συνθετικές επιθέσεις και δεν αντικατοπτρίζουν κίνηση σε κανονικό δίκτυο. Να γίνει εφαρμογή των εκπαιδευόμενων μοντέλων και ο εντοπισμός της απόδοσης σε πραγματική κίνηση ενός δικτύου στο οποίο θα υπάρχει και ένας μικρός αριθμός επιθέσεων.

REFERENCES

[1] R. Mohanakrishnan, "What Is a Computer Network? Definition, Objectives, Components, Types, and Best Practices" (2024,May 13) https://www.spiceworks.com/tech/networking/articles/what-is-a-computer-network/ (accessed 4 July,2024)

[2] "Basics of Computer Networking | geeksforgeeks.org" (2024,May 16) https://www.geeksforgeeks.org/basics-computer-networking/ (accessed 4 July,2024)

[3] A. S. Gillis, "What is a network topology?" https://www.techtarget.com/searchnetworking/definition/network-topology#:~:text=What%20is%20a%20network%20topology,with%20switch%20and%20router%20features. (accessed 4 July,2024)

[4] "What Is a Network Protocol, and How Does It Work? | comptia.org" https://www.comptia.org/content/guides/what-is-a-network-protocol#:~:text=A%20network%20protocol%20is%20an,devices%20in%20the%20same%20network. (accessed 4 July,2024)

[5] "IP Address Definition and Explanation | Fortinet.com" https://www.fortinet.com/resources/cyberglossary/what-is-ip-address#:~:text=IP%20Address%20Definition%20and%20Explanation,use%20the%20internet%20to%20communicate. (accessed 4 July,2024)

[6] "What is a Network?" https://fcit.usf.edu/network/chap1/chap1.htm#:~:text=A%20network%20consists%20of%20two,files%2C%20or%20allow%20electronic%20communications. (accessed 4 July,2024)

[7] "What is Network Traffic? | fortinet.com" https://www.fortinet.com/resources/cyberglossary/network-traffic (accessed 4 July,2024)

[8] "What is Network Tomography, and How Does it Work? | extnoc.com" https://www.extnoc.com/learn/networking/network-tomography (accessed 4 July,2024)

[9] P. Vouzis, "What is Network Tomography" (2014, July 8). https://netbeez.net/blog/network-tomography/ (accessed 4 July,2024)

[10] "What is Traceroute: What Does It Do & How Does It Work?| fortinet.com" https://www.fortinet.com/resources/cyberglossary/traceroutes#:~:text=A%20traceroute%20provides%20a%20map,along%20the%20way%2C%20particularly%20routers. (accessed 4 July,2024)

[11] Rui Castro, Mark Coates, Gang Liang, Robert Nowak, Bin Yu. "Network Tomography: Recent Developments." Statistical Science, Vol. 19,No. 3,pp. 499-517, August 2004, doi: https://doi.org/10.1214/088342304000000422

[12] G. Kakkavas, N. Fryganiotis, V. Karyotis and S. Papavassiliou, "Generative Deep Learning Techniques for Traffic Matrix Estimation From Link Load Measurements," in *IEEE Open Journal of the Communications Society*, vol. 5, pp. 1029-1046, 2024, doi: 10.1109/OJCOMS.2024.3358740.

[13] J. Barnard, C. Stryker, "What is anomaly detection" (2023 , 12 December) https://www.ibm.com/topics/anomaly-detection#:~:text=Anomaly%20detection%2C%20or%20outlier%20detection,rest%20of%20a%20data%20set (accessed 4 July,2024)

[14] "Sampling methods, types & techniques| qualtrics.com" https://www.qualtrics.com/en-gb/experience-management/research/sampling-methods/?rid=ip&prevsite=en&newsite=uk&geo=CY&geomatch=uk (accessed 4 July,2024)

[15] "Explaining Exit Polls (aapor.org)" https://aapor.org/wp-content/uploads/2022/12/Explaining-Exit-Polls-508.pdf (accessed 4 July,2024)

[16] Shona McCombes. "Sampling Methods | Types, Techniques & Examples" (2023, June 22). https://www.scribbr.com/methodology/sampling-methods/ (accessed 4 July,2024)

[17] K.C. Claffy, G.C. Polyzos, and H.-W. Braun. "Application of Sampling Methodologies to Network Traffic Characterization", In Proceedings of ACM SIGCOMM'93, San Francisco, CA, pp. 13- 17, September 1993.

[18] Johan de W. Bruwer,N. E.Haydam, "Reducing bias in shopping mall-intercept surveys: the time-based systematic sampling method" (1996). https://sajbm.org/index.php/sajbm/article/view/803/737 (accessed 4 July,2024)

[19] G. Androulidakis, V. Chatzigiannakis, S. Papavassiliou, M. Grammatikou and V. Maglaris, "Understanding and Evaluating the Impact of Sampling on Anomaly Detection Techniques," *MILCOM 2006 - 2006 IEEE Military Communications conference*, Washington, DC, USA, 2006, pp. 1-7, doi: 10.1109/MILCOM.2006.302407.

[20] "How Change Point Detection works | ArcGIS Pro 3.3 "https://pro.arcgis.com/en/pro-app/latest/tool-reference/space-time-pattern-mining/how-change-point-detection-works.htm#:~:text=Change%20point%20detection%20identifies%20time,significantly%20from%20a%20single%20model. (accessed 4 July,2024)

[21] J. McCaffrey. “Anomaly Detection Using Principal Component Analysis (PCA) | THE DATA SCIENCE LAB “ (2021, October 21). https://visualstudiomagazine.com/articles/2021/10/20/anomaly-detection-pca.aspx (accessed 4 July,2024)

[22] “SYN flood attack | cloudflare.com” https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/ (accessed 4 July,2024)

[23] “What is AI? | IBM ” https://www.ibm.com/topics/artificial-intelligence (accessed 4 July,2024)

[24] “What is machine learning (ML)? | IBM ” https://www.ibm.com/topics/machine-learning (accessed 4 July,2024)

[25] Vijay Kanade. “What Is Machine Learning? Definition, Types, Applications, and Trends “ (2022, April 4).https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-ml/ (accessed 4 July,2024)

[26] “10 everyday machine learning use cases| IBM Data and AI Team”(2023, October 16).https://www.ibm.com/blog/10-everyday-machine-learning-use-cases/ (accessed 4 July,2024)

[27] “What is Supervised Learning? | cloud.google.com” https://cloud.google.com/discover/what-is-supervised-learning#:~:text=Supervised%20learning%20is%20a%20category,the%20input%20and%20the%20outputs. (accessed 4 July,2024)

[28] “Getting started with Classification | geeksforgeeks.org” (2024,January 24). https://www.geeksforgeeks.org/getting-started-with-classification/ (accessed 4 July,2024)

[29] V. Kurama, J. Powers,and A. Oppermann. “Regression in Machine Learning: What It Is and Examples of Different Models” (2023,February 28). https://builtin.com/data-science/regression-machine-learning#:~:text=Regression%20is%20a%20supervised%20machine,are%20variance%2C%20bias%20and%20error. (accessed 4 July,2024)

[30] “What is Overfitting? | amazon.com ”https://aws.amazon.com/what-is/overfitting/#:~:text=Overfitting%20occurs%20when%20the%20model,to%20several%20reasons%2C%20such%20as%3A&text=The%20training%20data%20size%20is,all%20possible%20input%20data%20values. (accessed 4 July,2024)

[31] “Regularization in Machine Learning | geeksforgeeks.org” (2024,March 18). https://www.geeksforgeeks.org/regularization-in-machine-learning/ (accessed 4 July,2024)

[32] “What is unsupervised learning? | IBM” https://www.ibm.com/topics/unsupervised-learning (accessed 4 July,2024)

[33] R.-Q. Chen , G.-H. Shi , W.-L. Zhao ,and C.-H. Liang , “A Joint Model for IT Operation Series Prediction and Anomaly Detection”, Neurocomputing, vol. 448, pp. 130-139, April 2021, doi: https://doi.org/10.1016/j.neucom.2021.03.062

[34] A. Bewtra, “The Ultimate Guide to Semi-Supervised Learning” (2022, July 1). https://www.v7labs.com/blog/semi-supervised-learning-guide#:~:text=Semi%2Dsupervised%20learning%20is%20a,between%20supervised%20and%20unsupervised%20learning. (accessed 4 July,2024)

[35] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, and M. Kloft, “DEEP SEMI-SUPERVISED ANOMALY DETECTION”, Published as a conference paper at ICLR 2020, vol. 2, February 14, 2020, doi: https://doi.org/10.48550/arXiv.1906.02694

[36] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, “GANomaly: Semi-Supervised Anomaly Detection via Adversarial Training”, Durham University, vol. 3, November 13,2018, doi: https://doi.org/10.48550/arXiv.1805.06725

[37] D. Salunke, “SVC (Support Vector Classifier)”(2023 , August 28). https://www.linkedin.com/pulse/svc-support-vector-classifier-dishant-salunke#:~:text=SVC%20(Support%20Vector%20Classifier)%3A%20SVC%20is%20a%20specific%20implementation,data%20points%20into%20different%20classes. (accessed 4 July,2024)

[38] “Derivative of the Sigmoid Function | geeksforgeeks.org” (2024,June 13). https://www.geeksforgeeks.org/derivative-of-the-sigmoid-function/ (accessed 4 July,2024)

[39] “Logistic Regression in Machine Learning | geeksforgeeks.org” (2024,June 20). https://www.geeksforgeeks.org/understanding-logistic-regression/ (accessed 4 July,2024)

[40] “Linear Regression vs Logistic Regression | javatpoint.com” https://www.javatpoint.com/linear-regression-vs-logistic-regression-in-machine-learning (accessed 4 July,2024)

[41] “What is the k-nearest neighbors (KNN) algorithm? | IBM” https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20(KNN,used%20in%20machine%20learning%20today. (accessed 4 July,2024)

[42] “Decision Tree | geeksforgeeks.org” (2024,May 17). https://www.geeksforgeeks.org/decision-tree/ (accessed 4 July,2024)

[43] Pablo Aznar. “Decision Trees: Gini vs Entropy | quantdare.com” (2020, December 2). https://quantdare.com/decision-trees-gini-vs-entropy/ (accessed 4 July,2024)

[44] “Pruning Decision Trees | geeksforgeeks.org” (2024,April 10). https://www.geeksforgeeks.org/pruning-decision-trees/ (accessed 4 July,2024)

[45] “Random Forest Classifier using Scikit-learn | geeksforgeeks.org” (2024,January 31). https://www.geeksforgeeks.org/random-forest-classifier-using-scikit-learn/ (accessed 4 July,2024)

[46] “Voting Classifier | geeksforgeeks.org” (2023,October 11). https://www.geeksforgeeks.org/voting-classifier/ (accessed 4 July,2024)

[47] A.-U.-Rahman, “Understanding Soft Voting and Hard Voting: A Comparative Analysis of Ensemble Learning Methods | medium.com” (2023, August 2). https://medium.com/@awanurrahman.cse/understanding-soft-voting-and-hard-voting-a-comparative-analysis-of-ensemble-learning-methods-db0663d2c008 (accessed 4 July,2024)

[48] Om Pramod, “Model Parameters and Hyperparameters in machine learning | medium.com” (2023, January 14). https://medium.com/@ompramod9921/model-parameters-and-hyperparameters-in-machine-learning-502799f982d7#:~:text=A%20parameter%20is%20a%20variable,before%20the%20training%20process%20begins. (accessed 4 July,2024)

[49] “Hyperparameters of Random Forest Classifier | geeksforgeeks.org” (2021,January 22). https://www.geeksforgeeks.org/hyperparameters-of-random-forest-classifier/ (accessed 4 July,2024)

[50] S. Kumar, “What is label encoding? Application of label encoder in machine learning and deep learning models. | medium.com” (2024, January 13). https://medium.com/@sunnykumar1516/what-is-label-encoding-application-of-label-encoder-in-machine-learning-and-deep-learning-models-c593669483ed (accessed 4 July,2024)

[51] “One Hot Encoding in Machine Learning | geeksforgeeks.org” (2024,March 21). https://www.geeksforgeeks.org/ml-one-hot-encoding/ (accessed 4 July,2024)

[52] W. Fulmyk, “Ordinal Encoding — A Brief Explanation | medium.com” (2023, July 25). https://medium.com/@WojtekFulmyk/ordinal-encoding-a-brief-explanation-a29cf374dbc1#:~:text=Ordinal%20encoding%20is%20a%20preprocessing,that%20expect%20numerical%20input%20features. (accessed 4 July,2024)

[53] “What is Feature Scaling and Why Does Machine Learning Need It? | medium.com” (2023, November 16). https://medium.com/@shivanipickl/what-is-feature-scaling-and-why-does-machine-learning-need-it-104eedebb1c9 (accessed 4 July,2024)

[54] Z. Bobbitt, “How to Normalize Data Between 0 and 1 | statology.org” (2021,April 26). https://www.statology.org/normalize-data-between-0-and-1/ (accessed 4 July,2024)

[55] “Standardized Values: Example | statisticshowto.com” https://www.statisticshowto.com/standardized-values-examples/#:~:text=Step%201%3A%20Identify%20the%20observation,(%CF%83)%20in%20the%20question.&text=Step%202%3A%20Plug%20the%20values,%3D%20520%20%E2%80%93%20420%20%2F%2050. (accessed 4 July,2024)

[56] Sumeet Kumat Agrawal. “Metrics to Evaluate your Classification Model to take the Right Decisions | analyticsvidhya.com” (2024, June 5). https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/ (accessed 4 July,2024)

[57] “Understanding the F-Score and its Significance | giskard.ai” https://www.giskard.ai/glossary/f-score#:~:text=The%20F%2D2%20score%20is,2%20*%20precision%20%2B%20recall). (accessed 4 July,2024)

[58] “How to explain the ROC curve and ROC AUC score? | evidentlyai.com” https://www.evidentlyai.com/classification-metrics/explain-roc-curve#:~:text=The%20ROC%20AUC%20score%20is%20the%20area%20under%20the%20ROC,and%201%20indicates%20perfect%20performance. (accessed 5 July,2024)

[59] A. Chugh, “MAE, MSE, RMSE, Coefficient of Determination, Adjusted R Squared — Which Metric is Better? | medium.com” (2020, December 8). https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e (accessed 4 July,2024)

[60] P. Canovas,“Time Series Forecasting: Error Metrics to Evaluate Model Performance | typethepipe.com” (2019, September 21). https://typethepipe.com/post/energy-forecasting-error-metrics/ (accessed 4 July,2024)

[61] “Residual Standard Error and R2 | evanlray.com” https://www.evanlray.com/stat140_f2018/materials/20181112_residuals/20181112_residual_standard_error_R_squared.pdf (accessed 4 July,2024)

[62] D. Bergmann, and C. Stryker, “What is an autoencoder? | IBM”. (November 23,2023). https://www.ibm.com/topics/autoencoder (accessed 10 July,2024)

[63] D. Birla, “Basics of Autoencoders | medium.com” (March 12,2019)https://medium.com/@birla.deepak26/autoencoders-76bb49ae6a8f (accessed 10 July,2024)

[64] S. Bansal, “How Autoencoders Work. Intro and UseCases | kaggle.com) (2018). https://www.kaggle.com/code/shivamb/how-autoencoders-work-intro-and-usecases (accessed 10 July,2024)

[65]J. Murel, and E. Kavlakoglu, “What is dimensionality reduction? | IBM”. (January 5, 2024).https://www.ibm.com/topics/dimensionality-reduction (accessed 10 July,2024)

[66] Wang, Yasi & Yao, Hongxun & Zhao, Sicheng. (2015). Auto-Encoder Based Dimensionality Reduction. Neurocomputing. 184. 10.1016/j.neucom.2015.08.104.

[67] “The UNSW-NB15 Dataset | UNSW Canberra at ADFA” (2021, 2 June) https://research.unsw.edu.au/projects/unsw-nb15-dataset (accessed 5 July,2024)

[68] Moustafa, Nour, and Jill Slay. "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)." *Military Communications and Information Systems Conference (MilCIS)*, 2015. IEEE, 2015.

[69] Moustafa, Nour, and Jill Slay. "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 dataset and the comparison with the KDD99 dataset." *Information Security Journal: A Global Perspective* (2016): 1-14.

[70] Moustafa, Nour, et al. "Novel geometric area analysis technique for anomaly detection using trapezoidal area estimation on large-scale networks." *IEEE Transactions on Big Data (2017)*.

[71] Moustafa, Nour, et al. "Big data analytics for intrusion detection system: statistical decision-making using finite dirichlet mixture models." *Data Analytics and Decision Support for Cybersecurity. Springer, Cham, 2017.* 127-156.

[72] Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, and Marius Portmann. NetFlow Datasets for Machine Learning-Based Network Intrusion Detection Systems. In Big Data Technologies and Applications: 10th EAI International Conference, BDTA 2020, and 13th EAI International Conference on Wireless Internet, WiCON 2020, Virtual Event, December 11, 2020, Proceedings (p. 117). Springer Nature.

[73] “KDP Cup 1999 Data | kdd.ics.uci.edu”(1999, October 28) https://kdd.ics.uci.edu/databases/kddcup99/kddcup99 (accessed 5 July,2024)

[74] Hassan Zaib. “NSL-KDD | kaggle.com” (2019). https://www.kaggle.com/datasets/hassan06/nslkdd (accessed 5 July,2024)

[75] “Intrusion detection evaluation dataset (CIC-IDS2017) | unb.ca” https://www.unb.ca/cic/datasets/ids-2017.html (accessed 5 July,2024)

[76] “Fuzzing | imperva.com” https://www.imperva.com/learn/application-security/fuzzing-fuzz-testing/ (accessed 4 July,2024)

[77] Shivanshu. “What is Cryptanalysis? Types of Cryptanalysis Attacks | intellipaat.com” (2024 April 12). https://intellipaat.com/blog/what-is-crytanalysis/#:~:text=to%20crack%20cryptosystems.-,What%20is%20a%20Cryptanalytic%20attack%3F,English%20document%20or%20Java%20code. (accessed 4 July,2024)

[78] B. L.-Bergmans, “BACKDOOR ATTACKS | crowdstrike.com” (2023, July 17). https://www.crowdstrike.com/cybersecurity-101/attack-types/backdoor-attack/ (accessed 4 July,2024)

[79] “What is a denial of service attack (DoS) ? | paloaltonetworks.com” https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos (accessed 4 July,2024)

[80] “What Is An Exploit? | fortinet.com” https://www.fortinet.com/resources/cyberglossary/exploit (accessed 4 July,2024)

[81] A. Magnusson ,“What is a Brute Force Attack? Types, Examples & Prevention | strongdm.com” (2024, June 24). https://www.strongdm.com/blog/brute-force-attack (accessed 4 July,2024)

[82] “What Is Cyber Reconnaissance? | sentinelone.com” https://www.sentinelone.com/cybersecurity-101/what-is-cyber-reconnaissance/ (accessed 5 July,2024)

[83] N. Alam, “How Attackers use Shellcodes to Exploit a Vulnerable System | security.packt.com” (2022, June 24). https://security.packt.com/how-attackers-use-shellcodes-to-exploit-a-vulnerable-system/ (accessed 5 July,2024)

[84] B. L.-Bergmans, “WHAT IS A COMPUTER WORM? | crowdstrike.com” (2023, July 31). https://www.crowdstrike.com/cybersecurity-101/malware/computer-worm/ (accessed 4 July,2024)

[85] I. Fosic , D. Zagar , K. Grgic ,and V. Krizanovic, “Anomaly detection in NetFlow network traffic using supervised machine learning algorithms” , Journal of Industrial Information Integration, vol. 33, 100466, June 2023, doi: https://doi.org/10.1016/j.jii.2023.100466

[86]”Netflow | wikipedia.org” (Last Edited 2024,28 June). https://en.wikipedia.org/wiki/NetFlow#:~:text=NetFlow%20records%20are%20traditionally%20exported,configured%20on%20the%20sending%20router. (accessed 5 July,2024)

[87] “Stochastic Gradient Descent Classifier | geeksforgeeks.org” (2023,November 4). https://www.geeksforgeeks.org/stochastic-gradient-descent-classifier/ (accessed 4 July,2024)

[88] “Gaussian Naive Bayes using Sklearn | geeksforgeeks.org” (2023,December 17). https://www.geeksforgeeks.org/gaussian-naive-bayes-using-sklearn/ (accessed 4 July,2024)

[89] “AdaBoost Classification | apmonitor.com” https://apmonitor.com/pds/index.php/Main/AdaBoost (accessed 4 July,2024)

[90] “Pipeline | scikit-learn.org “ https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html (accessed 5 July,2024)

[91] R. O’Connor, “Introduction to Variational Autoencoders Using Keras | assemblyai.com” (January 3, 2022). https://www.assemblyai.com/blog/introduction-to-variational-autoencoders-using-keras/ (accessed 7 July,2024)

[92] “ReLU Activation Function | dremio.com”.https://www.dremio.com/wiki/relu-activation-function/#:~:text=The%20ReLU%20activation%20function%20works,%3D%20max(0%2C%20x) (accessed 8 July,2024)