Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Ανίχνευση Phishing URLs με χρήση Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ Β. ΡΟΥΣΣΗΣ
ΠΑΝΑΓΙΩΤΗΣ Δ. ΣΟΥΛΙΩΤΗΣ

Επιβλέπων : Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Συνεπιβλέπων : Γεώργιος Σιόλας
Εργαστηριακό και Διδακτικό Προσωπικό Ε.Μ.Π.

Αθήνα, Ιούνιος 2021

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Συστημάτων Τεχνητής Νοημοσύνης και Μάθησης

Ανίχνευση Phishing URLs με χρήση Μηχανικής Μάθησης

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΔΗΜΗΤΡΙΟΣ Β. ΡΟΥΣΣΗΣ
ΠΑΝΑΓΙΩΤΗΣ Δ. ΣΟΥΛΙΩΤΗΣ

Επιβλέπων: Ανδρέας-Γεώργιος Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

Συνεπιβλέπων : Γεώργιος Σιόλας
Εργαστηριακό και Διδακτικό Προσωπικό Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24η Ιουνίου 2021.

.....	.....	.....
Ανδρέας-Γεώργιος Σταφυλοπάτης Καθηγητής Ε.Μ.Π.	Στέφανος Κόλλιας Καθηγητής Ε.Μ.Π.	Γεώργιος Στάμου Αναπληρωτής Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2021

Δημήτριος Β. Ρούσσης

Παναγιώτης Δ. Σουλιώτης

Διπλωματούχοι Ηλεκτρολόγοι Μηχανικοί και Μηχανικοί Υπολογιστών Ε.Μ.Π.

Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας είναι η μελέτη του φαινομένου του Phishing με χρήση Μηχανικής Μάθησης.

Αρχικά, συλλέγουμε ενεργά URLs από διάφορες ανοιχτές πηγές, από τα οποία εξάγουμε τα χαρακτηριστικά που τα περιγράφουν. Με τον τρόπο αυτό δημιουργούμε το σύνολο δεδομένων που θα χρησιμοποιηθεί για την εκπαίδευση και την αξιολόγηση της ικανότητας διαφόρων αλγορίθμων επιβλεπόμενης μηχανικής μάθησης και αρχιτεκτονικών βαθιάς μάθησης, να διακρίνουν τα legitimate URLs από τα phishing.

Κατόπιν, τα μοντέλα μηχανικής μάθησης που θεωρούνται κατάλληλα για την επίλυση του προβλήματος (Αλγόριθμος k-Κοντινότερων Γειτόνων, Πολυστρωματικό Perceptron, Τυχαία Δάση, Ενίσχυση Κλίσης, Συνελικτικά Νευρωνικά Δίκτυα), δίνουν προβλέψεις για την ταξινόμηση οποιουδήποτε URL. Η ανάπτυξη ενός επιτυχημένου σχήματος ψηφοφορίας συνδυάζει τις επιμέρους προβλέψεις προσφέροντας μια ολοκληρωμένη τελική πρόβλεψη.

Τέλος, δημιουργούμε μια διαδικτυακή εφαρμογή που ενσωματώνει το σύστημα πρόβλεψης, ανιχνεύοντας αν το URL που εισάγει ο χρήστης είναι phishing.

Λέξεις κλειδιά

ανίχνευση, phishing, url, μηχανική μάθηση, επιβλεπόμενη μάθηση, νευρωνικά δίκτυα, αλγόριθμος k-κοντινότερων γειτόνων, πολυστρωματικό perceptron, τυχαία δάση, ενίσχυση κλίσης, βαθιά μάθηση, συνελικτικά νευρωνικά δίκτυα, μοντέλα, ταξινόμηση, σχήμα ψηφοφορίας, διαδικτυακή εφαρμογή, server

Abstract

The objective of the current diploma thesis is the research of the Phishing phenomenon using Machine Learning.

First, we collect active URLs from various open sources and we extract the characteristics that describe them, in order to create the dataset that will be used to train and evaluate the ability of different machine learning algorithms and architectures of deep learning to classify correctly legitimate and phishing URLs.

Next, the machine learning models, that are suitable for the problem (k-Nearest Neighbors, Multi-Layer Perceptron, Random Forest, Gradient Boosting, Convolutional Neural Networks), give predictions about the classification of any URL given. The development of a successful voting scheme offers a complete final estimation.

Finally, we create a web application which integrates the prediction system and detects if the URL given by the user is phishing.

Key words

detection, phishing, url, machine learning, supervised learning, neural networks, k-nearest neighbors, multi-layer perceptron, random forest, gradient boosting, deep learning, convolutional neural networks, models, classification, voting scheme, web application, server

Ευχαριστίες

Θα θέλαμε να ευχαριστήσουμε τον Καθηγητή Ε.Μ.Π. κ. Σταφυλοπάτη Ανδρέα-Γεώργιο για την τιμή να μας εμπιστευτεί την εκπόνηση της παρούσας εργασίας, καθώς και τους κ. Κόλλια Στέφανο, Καθηγητή Ε.Μ.Π. και κ. Στάμου Γεώργιο, Αναπληρωτή Καθηγητή Ε.Μ.Π. για την συμμετοχή τους στην εξεταστική επιτροπή. Επίσης θα θέλαμε να ευχαριστήσουμε θερμά τον κ. Σιόλα Γεώργιο, Ε.ΔΙ.Π. Ε.Μ.Π. για τη συνεχή υποστήριξη και καθοδήγηση του σε όλα τα στάδια εκπόνησης της παρούσας εργασίας.

Ευχαριστούμε επίσης τον ~okeanos-knossos για την παραχώρηση χρήσης τεχνολογικών υποδομών που συνέβαλλαν σημαντικά στην ανάπτυξη της παρούσας εργασίας.

Τέλος, ευχαριστούμε ιδιαίτερα τις οικογένειες μας για την αμέριστη υπομονή και υποστήριξη που επέδειξαν καθ’ όλη τη διάρκεια των σπουδών μας.

Δημήτριος Β. Ρούσσης & Παναγιώτης Δ. Σουλιώτης,

Αθήνα 24η Ιουνίου 2021

Περιεχόμενα

Κατάλογος πινάκων

Κατάλογος εικόνων

Κεφάλαιο 1

Εισαγωγή

1.1 Phishing

Η περίοδος της πανδημίας της COVID19 επέφερε κολοσσιαία αύξηση στον όγκο των τραπεζικών και γενικότερα ηλεκτρονικών συναλλαγών που πραγματοποιούνται καθημερινά. Ωστόσο, σημαντικό μέρος του πληθυσμού σε όλες τις χώρες του κόσμου δεν ήταν τόσο εξοικειωμένο με τα τεχνολογικά μέσα. Το γεγονός αυτό έδωσε την ευκαιρία σε αρκετούς κακόβουλους χρήστες του Διαδικτύου να τους εξαπατήσουν αποκομίζοντας μεγάλα οικονομικά οφέλη. Μία από τις μεθόδους που χρησιμοποίησαν είναι το phishing.

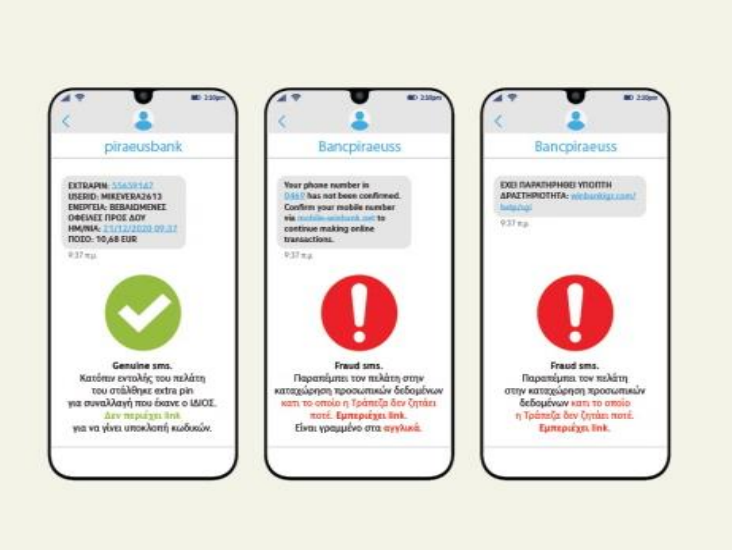
Το phishing δεν γεννήθηκε μέσα στην πανδημία αλλά γιγαντώθηκε μέσα σε αυτήν. Με τις τράπεζες κλειστές για φυσική επικοινωνία με το κοινό, η δημιουργία ηλεκτρονικής τραπεζικής (e-banking) αποτέλεσε μονόδρομο για ιδιώτες και επιχειρήσεις και άρα εφαλτήριο και για κακόβουλες ενέργειες. Ωστόσο, δεν σχετίζεται αμιγώς με τραπεζικές συναλλαγές.

Το phishing αν και πρωτοπαρουσιάστηκε σε συνέδριο της Hewlett-Packard το 1987, ξεκίνησε στην πράξη το 1995 με τις πρώτες phishing επιθέσεις να σχετίζονται με την τότε μεγαλύτερη διαδικτυακή υπηρεσία επικοινωνίας AOL [1]. Οι phishers υποδύονταν υπαλλήλους της AOL και επικαλούμενοι πρόβλημα στο λογαριασμό του χρήστη, του ζητούσαν τους κωδικούς του και τους αριθμούς των τραπεζικών του λογαριασμών. Με την πάροδο του χρόνου το phishing εξελίχθηκε και εκτός από τις επιθέσεις σε τραπεζικούς λογαριασμούς, χρησιμοποιήθηκε και για στοχευμένες επιθέσεις εναντίον κυβερνήσεων. Ο στόχος ήταν η υποκλοπή απόρρητων πληροφοριών για τη διοίκηση του κράτους.

Το Phishing (ελληνιστί Ηλεκτρονικό Ψάρεμα Στοιχείων) αποτελεί μια ευρεία διαδεδομένη τεχνική κατά την οποία κακόβουλοι δημιουργούν πλαστές ιστοσελίδες που προσομοιάζουν με τις επίσημες ιστοσελίδες νόμιμων οργανισμών/εταιρειών/τραπεζών. Στη συνέχεια στέλνουν μηνύματα (emails ή sms) ή δημιουργούν παραπλανητικές διαφημίσεις τα οποία περιέχουν ένα σύνδεσμο (link) στο παραπλανητικό Ενιαίο Εντοπιστή Πόρων (Uniform Resource Locator, συντ. URL) που έχουν δημιουργήσει. Σε αυτές τις ιστοσελίδες ζητείται στους χρήστες να συμπληρώσουν διάφορα απόρρητα προσωπικά και οικονομικά δεδομένα όπως όνομα χρήστη, κωδικούς πρόσβασης, στοιχεία τραπεζικών καρτών κ.ά. Κύριοι λόγοι που επικαλούνται τα περισσότερα phishing μηνύματα είναι πρόβλημα στο λογαριασμό του χρήστη, επιβεβαίωση εκτέλεσης ή ακύρωση συναλλαγής (η οποία ουδέποτε έχει πραγματοποιηθεί από το χρήστη), ενέργεια αναβάθμισης υπηρεσίας, επιβεβαίωση τραπεζικών δεδομένων.

Η αντιμετώπιση του φαινομένου του Phishing δεν είναι καθόλου εύκολη. Οι εταιρείες που αναπτύσσουν λογισμικό προστασίας υπολογιστών (antivirus) φροντίζουν να καταγράφουν αυτές τις ιστοσελίδες στις βάσεις δεδομένων τους ως επισφαλείς και οι αρμόδιες αστυνομικές αρχές να μεριμνούν για την καταστολή τους. Παρόλα αυτά, οι ιστοσελίδες phishing δεν μένουν ενεργές για

πολύ καιρό. Ξεφυτρώνουν σαν τα μανιτάρια και η λειτουργία τους κρατά λίγα μόνο 24ωρα. Η δύναμη του phishing κρύβεται στην αληθοφάνεια του περιεχομένου που προβάλει, στοχεύοντας να πείσει το χρήστη να μην ελέγξει καθόλου το URL που επισκέπτεται, καθώς και στην έλλειψη γνώσεων και προσοχής του θύματος.



Εικόνα 1.1: Παράδειγμα Phishing SMS από την Τράπεζα Πειραιώς (Πηγή: [2])

Το θύμα δεν παρατηρεί τα ίχνη του phishing που είναι η παραλλαγμένη διεύθυνση email, sms, URL. Συχνά υπάρχουν αναγραμματισμοί, λάθος σύνταξη ή ορθογραφία, χρήση ειδικών χαρακτήρων (πχ -,@,#,$,% κ.ά.). Η οπτική εξαπάτηση του θύματος είναι εξίσου σημαντική, γι' αυτό ο σχεδιασμός, οι εικόνες, τα κείμενα και τα λογότυπα των οργανισμών/εταιρειών/τραπεζών που υποδύονται είναι σχεδόν πανομοιότυπα με τα πραγματικά.

Βέβαια όσο εξελίσσεται η αντιμετώπιση του φαινομένου, εξελίσσεται και το ίδιο το phishing, εφευρίσκοντας νέα τεχνάσματα.

Εικόνα 1.2: Δείγμα phishing URL που ομοιάζει με το πραγματικό της Τράπεζας Πειραιώς (Πηγή: [2])

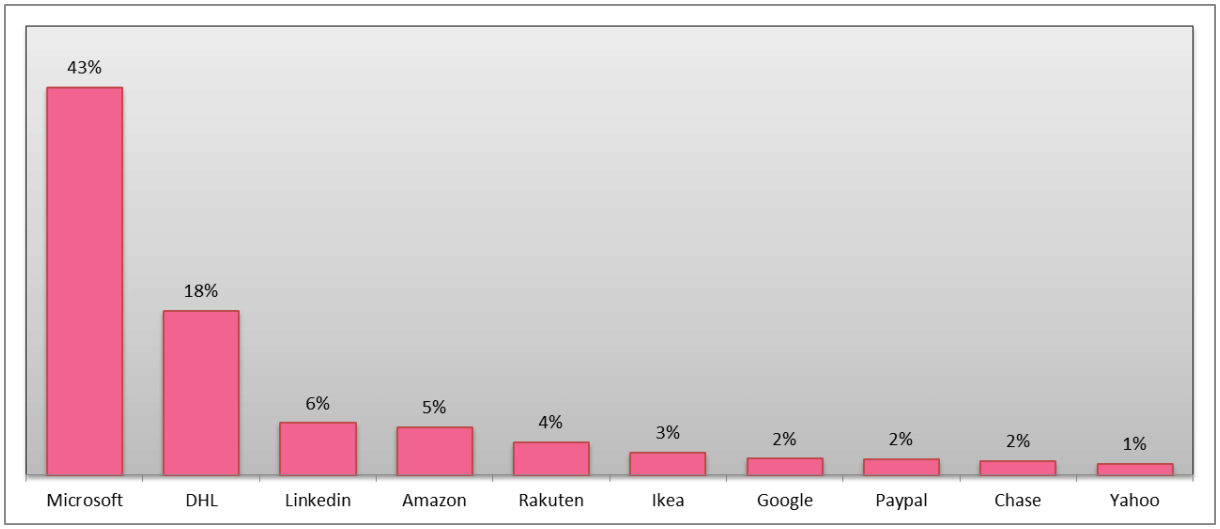
Σύμφωνα με το FBI, το phishing ήταν το συνηθέστερο είδος κυβερνοεγκλήματος το 2020, καθώς διπλασιάστηκαν σε συχνότητα τα περιστατικά phishing από 114.702 το 2019, σε 241.324 το 2020 [3]. Σύμφωνα πάλι με το FBI, ο αριθμός είναι 11 φορές μεγαλύτερος σε σχέση με το 2016.

Πιο αναλυτικά, το 75% των επιχειρήσεων το 2020 αντιμετώπισε κάποιο είδος phishing επίθεσης. Ο αριθμός φαντάζει τεράστιος αλλά δε θα πρέπει να ξεχνάμε ότι δηλώνει τις απόπειρες phishing και όχι τις επιτυχείς επιθέσεις, οι οποίες ανήλθαν στο διόλου ευκαταφρόνητο 44% κατά μέσο όρο σε όλο τον κόσμο και στο υψηλό 74% στις ΗΠΑ.

Η συχνότητα των επιθέσεων διαφέρει ανάλογα το μέγεθος της επιχείρησης/οργανισμού. Οι 3 κλάδοι που δέχθηκαν τις περισσότερες επιθέσεις phishing με βάση το πλήθος των εργαζομένων στις επιχειρήσεις/οργανισμούς είναι:

- 1-249 εργαζόμενοι: Επιχειρήσεις Φροντίδας Υγείας και Φαρμακευτικές, Εκπαίδευση, Βιομηχανία
- 250-999 εργαζόμενοι: Κατασκευές, Επιχειρήσεις Φροντίδας Υγείας και Φαρμακευτικές, Επιχειρήσεις παροχής υπηρεσιών
- 1.000 και περισσότεροι εργαζόμενοι: Τεχνολογία, Επιχειρήσεις Φροντίδας Υγείας και Φαρμακευτικές, Βιομηχανία

Το μεγαλύτερο μέρος των επιθέσεων phishing έγινε μέσω emails (96%). Σύμφωνα με έρευνα [4], το top 10 των brands που χρησιμοποιήθηκαν σε phishing επιθέσεις το 4^{ο} τρίμηνο του 2020 είναι:

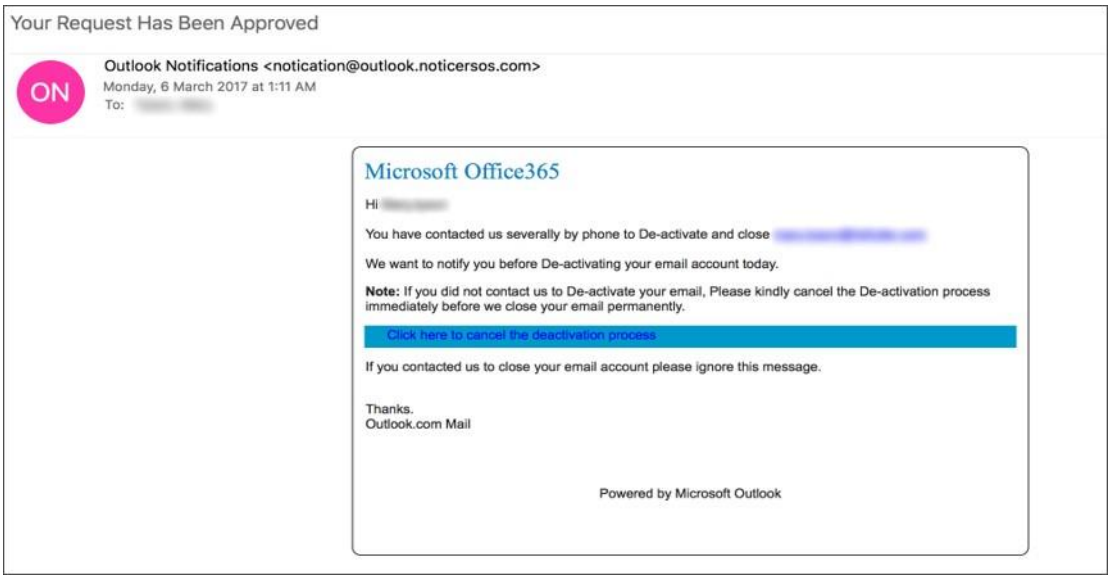


Εικόνα 1.3: Το top 10 των brands που χρησιμοποιήθηκαν σε phishing επιθέσεις το 4^{ο} τρίμηνο του 2020 (Πηγή: [4])

Όπως βλέπουμε πρόκειται για διεθνή πασίγνωστα Brands, στα οποία πάρα πολλοί άνθρωποι διατηρούν λογαριασμό και τα εμπιστεύονται. Αυτός είναι και ο λόγος που τα χρησιμοποίησαν οι κακόβουλοι, προκειμένου να πείσουν και τελικά να εξαπατήσουν τα θύματα τους αποσπώντας τους πολύτιμα δεδομένα [5]. Το top5 των δεδομένων που αποκομίστηκαν από phishing επιθέσεις είναι:

- Διαπιστευτήρια (Credentials, passwords, usernames, κωδικοί pin)
- Προσωπικά δεδομένα (ονοματεπώνυμο, διεύθυνση κατοικίας/εργασίας, διεύθυνση ηλεκτρονικού ταχυδρομείου)
- Απόρρητα δεδομένα επιχειρήσεων (πληροφορίες για πωλήσεις προϊόντων, στρατηγικές προώθησης και marketing)
- Ιατρικά δεδομένα (πληροφορίες για αγωγές υγείας, φάρμακα,

συνταγογραφήσεις, ασφαλιστικές καλύψεις)
- Τραπεζικά δεδομένα (αριθμοί λογαριασμών, πληροφορίες χρεωστικής/πιστωτικής κάρτας)



Εικόνα 1.4: Δείγμα phishing email που υποδύεται ότι προέρχεται από την Microsoft και συγκεκριμένα από την υπηρεσία Office365 (Πηγή: [5])

Αναμφίβολα, επίκαιρα γεγονότα που απασχολούν όλο τον κόσμο δε θα μπορούσαν να μην χρησιμοποιηθούν κι αυτά στην τεχνική του phishing. Όπως ήταν λοιπόν αναμενόμενο, phishing επιθέσεις σχετιζόμενες με την COVID19 ξεκίνησαν τον Μάρτιο του 2020 και εκτοξεύθηκαν στην κορυφή την 3^{η} και 4^{η} εβδομάδα του Απριλίου. Καθόλου παράξενο, καθώς την περίοδο εκείνη η μία μετά την άλλη, όλες οι χώρες έβλεπαν τα κρούσματα να γιγαντώνονται και οδηγούνταν σε καθολικά lockdown, επιβάλλοντας τη διακοπή κάθε επιχειρηματικής δραστηριότητας με φυσική παρουσία, αναγκάζοντας έτσι τους εργαζόμενους να δουλεύουν από το σπίτι τους, συχνά με τους προσωπικούς τους υπολογιστές και χωρίς την άμεση προστασία που παρέχουν τα firewalls και γενικότερα τα συστήματα ασφαλείας της επιχείρησης/οργανισμού τους.

Κυριότερος στόχος των επιθέσεων για το 2020 ήταν η απόσπαση χρημάτων, καθώς οι phishing επιθέσεις σχετιζόμενες με πληρωμές αυξήθηκαν κατά 112% μεταξύ του πρώτου και δεύτερου τριμήνου του έτους.

Οι προβλέψεις για το μέλλον παρουσιάζονται ιδιαίτερα ανησυχητικές, καθώς τον Απρίλιο του 2021 σημειώθηκαν πολύ μεγάλες διαρροές-υποκλοπές προσωπικών δεδομένων από λογαριασμούς χρηστών σε μέσα κοινωνικής δικτύωσης (Facebook, LinkedIn) [6] [7]. Το γεγονός αυτό θα επιτρέψει σε διάφορους κακόβουλους να αναλύσουν τα προφίλ των χρηστών και να δημιουργήσουν πιο στοχευμένες, πιο προσωποποιημένες phishing επιθέσεις καθιστώντας πιο δύσκολη την ανίχνευση τους από τους χρήστες. Η απειλή αυτή σε συνδυασμό με το μεγάλο πλήθος τραπεζικών συναλλαγών που πραγματοποιούνται διαδικτυακά καθιστά την ασφάλεια σημαντική (ίσως και κυρίαρχη) προτεραιότητα των χρηματοπιστωτικών ιδρυμάτων για το 2021 [8]. Σύμφωνα με την Deloitte Center for Financial Services Global Outlook Survey 2020, το 71% των προέδρων τραπεζών αναμένει αύξηση των δαπανών στον τομέα της κυβερνοασφάλειας (cybersecurity).

1.2 Στόχος της εργασίας

Στόχος της παρούσας εργασίας είναι η μελέτη του φαινόμενου του phishing, η συλλογή και η εξαγωγή χαρακτηριστικών από ενεργά phishing URLs με σκοπό τη δημιουργία ενός dataset (σύνολο δεδομένων), που θα χρησιμοποιηθεί για να εκπαιδεύσουμε και να αξιολογήσουμε την ικανότητα διάφορων ταξινομητών μηχανικής μάθησης να επιτύχουν να διακρίνουν τα αυθεντικά (legitimate) URLs από τα phishing. Κατόπιν, σκοπεύουμε να δημιουργήσουμε μια διαδικτυακή εφαρμογή στην οποία ο χρήστης θα εισάγει ένα URL και αυτή με βάση τους ταξινομητές που έχουμε εκπαιδεύσει θα του εμφανίζει μια ποσοστιαία πρόβλεψη για την κατηγορία στην οποία ανήκει αυτό το URL (legitimate ή phishing).

Η ύπαρξη μιας τέτοιας εφαρμογής σαφώς δεν στοχεύει στο να εντοπίσει κάθε phishing URL και να αποτρέψει κάθε απόπειρα εξαπάτησης μέσω phishing. Μπορεί όμως να αποτελέσει ένα ισχυρό εργαλείο στα χέρια των χρηστών του διαδικτύου και προσφέρει επίσης τη δυνατότητα στη μηχανική μάθηση να δοκιμάσει τις δυνατότητες της και στο πεδίο της ασφάλειας του κυβερνοχώρου.

1.3 Μεθοδολογία της εργασίας

Πρόκειται για ένα end-to-end (από άκρο σε άκρο) project. Αρχικά θα μελετήσουμε ποια είναι τα χαρακτηριστικά εκείνα που διακρίνουν τα URLs (λεξικολογικά χαρακτηριστικά, χαρακτηριστικά domain, χαρακτηριστικά του κώδικα της σελίδας και των επικεφαλίδων HTTP) προκειμένου να δημιουργήσουμε τα σύνολα δεδομένων εκπαίδευσης και ελέγχου, τα οποία θα χρησιμοποιήσουμε στην εξερεύνηση των διαφόρων αλγορίθμων επιβλεπόμενης μάθησης πειραματιζόμενοι και με αρχιτεκτονικές βαθιάς μάθησης. Η δημιουργία κάθε μοντέλου μηχανικής μάθησης που θα θεωρηθεί κατάλληλο για την επίλυση του προβλήματος θα προσφέρει και μια πρόβλεψη για κάθε URL που θα του δίνεται ως είσοδος. Ο συνδυασμός των διαφόρων προβλέψεων με την ανάπτυξη ενός επιτυχημένου voting scheme (συστήματος ψηφοφορίας) θα προσφέρει μια ολοκληρωμένη τελική πρόβλεψη. Το σύστημα πρόβλεψης που θα δημιουργηθεί θα ενθυλακωθεί σε μια διαδικτυακή εφαρμογή όπου ο χρήστης θα μπορεί απλά να εισάγει το URL που επιθυμεί και να λάβει πρόβλεψη για την κατηγορία που αυτό ανήκει (legitimate-phishing).

1.4 Δομή της εργασίας

Η υπόλοιπη εργασία διαρθρώνεται ως εξής. Στο Κεφάλαιο 2 περιγράφεται το θεωρητικό πλαίσιο της μηχανικής μάθησης, των μοντέλων και του voting scheme που χρησιμοποιήθηκαν. Στο Κεφάλαιο 3, παρουσιάζεται το πρακτικό κομμάτι της εργασίας και πιο συγκεκριμένα η κατασκευή της συλλογής δεδομένων από τα δεδομένα που συλλέχθηκαν. Αναλύονται οι διάφορες αρχιτεκτονικές μοντέλων μηχανικής μάθησης και τα αποτελέσματα των δοκιμών που έγιναν με αυτές. Επίσης, επιλέγεται και παραμετροποιείται το είδος του voting scheme. Στο Κεφάλαιο 4 παρουσιάζεται η δημιουργία και η ανάπτυξη της εφαρμογής. Τέλος, στο Κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα στο σύνολο ελέγχου, δίνονται παραδείγματα από τη χρήση της εφαρμογής και αναφέρονται πιθανές μελλοντικές κατευθύνσεις-επεκτάσεις.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Μηχανική μάθηση

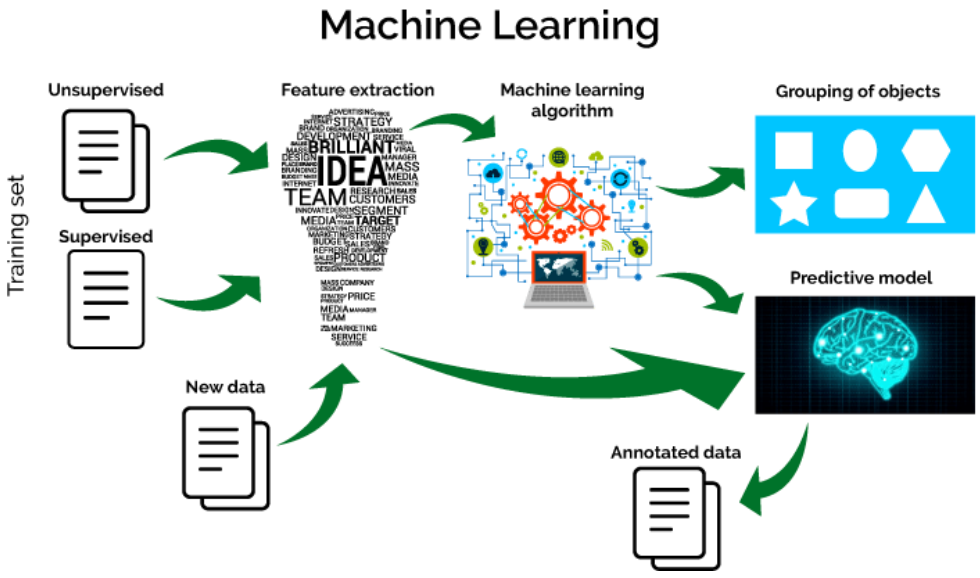
2.1.1 Ορισμός Μηχανικής Μάθησης

Η Μηχανική Μάθηση (Machine Learning) αποτελεί πεδίο της Επιστήμης των Υπολογιστών και υπάγεται στον κλάδο της Τεχνητής Νοημοσύνης (Artificial Intelligence). Ασχολείται με την ανάπτυξη αλγορίθμων που μπορούν να μαθαίνουν (εκπαιδεύονται) από τα δεδομένα χωρίς να έχουν ρητά προγραμματιστεί και να κάνουν προβλέψεις σχετικά με αυτά [9] [10]. Ο Tom M. Mitchell το 1997 πρότεινε έναν πιο επίσημο ορισμό που χρησιμοποιείται ευρέως: «Ένα πρόγραμμα υπολογιστή λέγεται ότι μαθαίνει από εμπειρία E ως προς μια κλάση εργασιών T και ένα μέτρο επίδοσης P, αν η επίδοσή του σε εργασίες της κλάσης T, όπως αποτιμάται από το μέτρο P, βελτιώνεται με την εμπειρία E» [11].

Η συμβολή της στο πεδίο της ανάλυσης δεδομένων είναι καθοριστική, καθώς επιτρέπει σε ερευνητές, αναλυτές, μηχανικούς να αναπτύξουν πολύπλοκα μοντέλα που θα φανερώσουν άγνωστες συσχετίσεις στα δεδομένα οδηγώντας στην εξαγωγή χρήσιμων συμπερασμάτων και τελικά στη λήψη αξιόπιστων αποφάσεων [11].

Η Μηχανική Μάθηση διακρίνεται σε 3 βασικά είδη:

- Επιβλεπόμενη Μάθηση (Supervised Learning)
- Μη-Επιβλεπόμενη Μάθηση (Unsupervised Learning)
- Ενισχυτική Μάθηση (Reinforcement Learning)



Εικόνα 2.1: Διαγραμματική απεικόνιση της ροής εργασιών της Μηχανικής Μάθησης (Πηγή: [9])

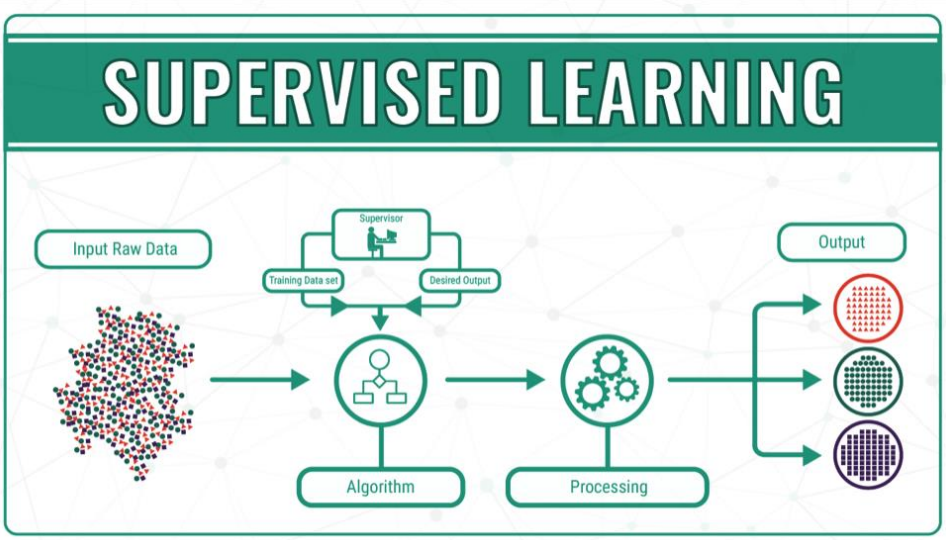
2.1.2 Είδη Μηχανικής Μάθησης

Η ταξινόμηση του μοντέλου σε κάποιο από τα τρία μεγάλα είδη γίνεται με βάση τη φύση του εκπαιδευτικού «σήματος».

Στην επιβλεπόμενη μηχανική μάθηση (ή αλλιώς μάθηση με επίβλεψη), το σύστημα δέχεται ως είσοδο τα δεδομένα ή αλλιώς δείγματα (samples) μαζί με την επιθυμητή τους έξοδο (label). Τα δείγματα διακρίνονται από τα χαρακτηριστικά τους (features). Τόσο τα χαρακτηριστικά των δειγμάτων όσο και τα δεδομένα μπορούν να έχουν αριθμητική ή κατηγορική τιμή. Στόχος του συστήματος είναι να μάθει να αντιστοιχίζει σωστά τα δείγματα με τις εξόδους τους.

Τα κυριότερα προβλήματα αυτού του είδους είναι:

- Ταξινόμηση (Classification): Δίνοντας ως είσοδο τα χαρακτηριστικά ενός δείγματος, το μοντέλο πρέπει να αναγνωρίσει σε ποια κλάση ανήκει το δείγμα που δόθηκε. Σε αυτή τη μορφή προβλήματος επιβλεπόμενης μάθησης ανήκει και το θέμα με το οποίο ασχολούμαστε στην παρούσα διπλωματική εργασία.
- Παλινδρόμηση (Regression): Παρόμοιο πρόβλημα με την ταξινόμηση με τη διαφορά ότι η έξοδος είναι αριθμητική τιμή.

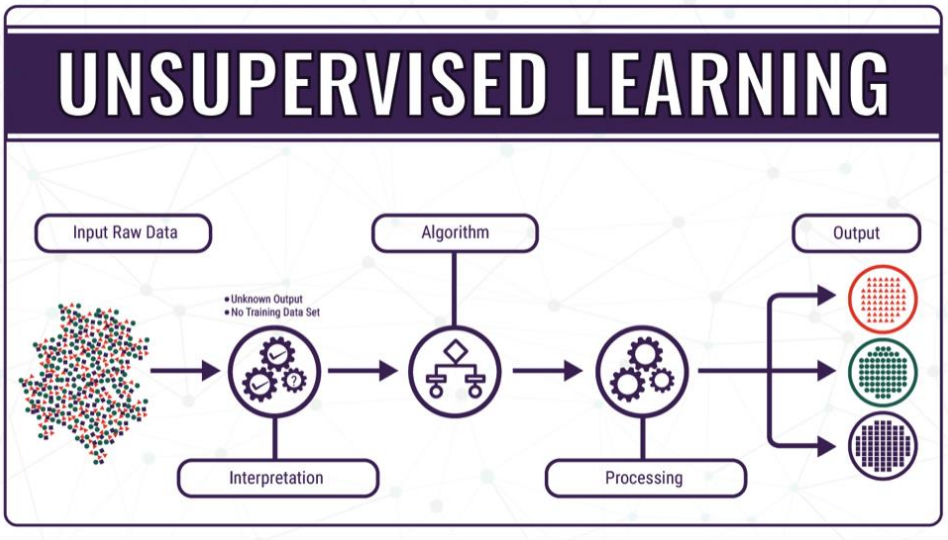


Εικόνα 2.2: Διαγραμματική απεικόνιση λειτουργίας της Επιβλεπόμενης Μάθησης (Πηγή: https://datafloq.com/read/machine-learning-explained-understanding-learning/4478)

Στην μη-επιβλεπόμενη μηχανική μάθηση (ή αλλιώς μάθηση χωρίς επίβλεψη), το σύστημα δέχεται ως είσοδο τα δεδομένα χωρίς όμως την επιθυμητή έξοδο. Το σύστημα οφείλει επομένως να ανακαλύψει τη δομή των δεδομένων εισόδου αναζητώντας για όμοια χαρακτηριστικά μεταξύ των δειγμάτων προκειμένου να καταφέρει να τα ομαδοποιήσει.

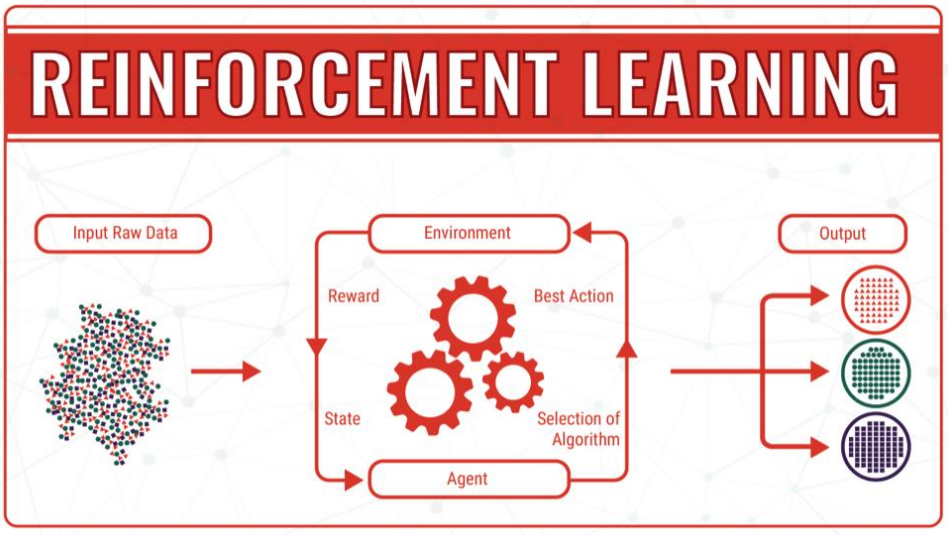
Τα κυριότερα προβλήματα αυτού του είδους είναι:

- Ανάλυση συσχετίσεων (Association Analysis): Αναζήτηση πιθανών μοτίβων συσχέτισης μεταξύ των διαφορετικών αντικειμένων.
- Ομαδοποίηση (Clustering): Οργάνωση των διαφορετικών στοιχείων ενός συνόλου δεδομένων σε διακριτές ομάδες, με βάση κοινά τους χαρακτηριστικά.



Εικόνα 2.3: Διαγραμματική απεικόνιση λειτουργίας της Μη-Επιβλεπόμενης Μάθησης (Πηγή: https://datafloq.com/read/machine-learning-explained-understanding-learning/4478)

Στην ενισχυτική μηχανική μάθηση, το σύστημα αλληλεπιδρά διαρκώς με το περιβάλλον του λαμβάνοντας ερεθίσματα, τα οποία το ενημερώνουν για την πρόοδο του βοηθώντας το να βελτιώσει τη συμπεριφορά του. Χρησιμοποιείται κυρίως σε προβλήματα λήψης αποφάσεων (πχ σχεδιασμός επιχειρηματικής στρατηγικής, αυτόματος έλεγχος κίνησης ρομπότ κ.ά.).



Εικόνα 2.4: Διαγραμματική απεικόνιση λειτουργίας της Ενισχυτικής Μάθησης (Πηγή: https://datafloq.com/read/machine-learning-explained-understanding-learning/4478)

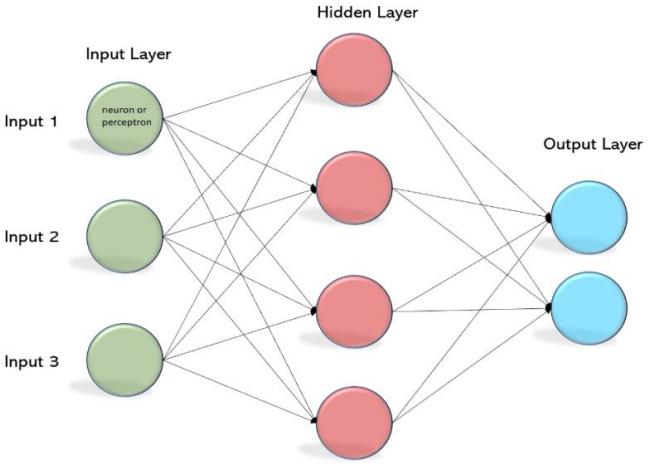
2.1.3 Αλγόριθμος k-Κοντινότερων Γειτόνων - KNN

Ο αλγόριθμος k-Κοντινότερων Γειτόνων (KNN) είναι μια μέθοδος ταξινόμησης με βάση την απόσταση που πρωτοαναπτύχθηκε το 1951 [12]. Πρόκειται για μη-παραμετροποιημένη μέθοδο μάθησης αφού δεν κάνει καμία υπόθεση για την κατανομή των δεδομένων και είναι επίσης οκνηρή (lazy) αφού χρησιμοποιεί όλα τα δεδομένα για την ταξινόμηση νέων δειγμάτων [13].

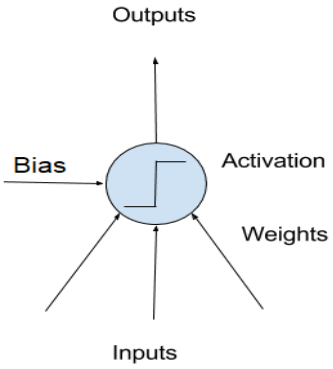
Χρησιμοποιείται τόσο για ταξινόμηση δεδομένων όσο και για παλινδρόμηση. Και στις δύο περιπτώσεις η έξοδος επιλέγεται ως η κλάση των k πιο κοντινών δειγμάτων εκπαίδευσης στο σύνολο δεδομένων. Ο υπολογισμός της απόστασης του νέου σημείου πρέπει να γίνει ως προς όλα τα άλλα σημεία του συνόλου δεδομένων ώστε να βρεθούν οι κοντινότεροι γείτονες. Υπάρχει λοιπόν η περίπτωση, τα χαρακτηριστικά των δεδομένων να έχουν διαφορετική κατανομή και συνεπώς να δίνεται μεγαλύτερη σημασία σε κάποια έναντι άλλων, με αποτέλεσμα η κανονικοποίηση των χαρακτηριστικών των δεδομένων να αυξήσει την ακρίβεια δραματικά. Επιπροσθέτως για να αποφύγουμε τις ισοψηφίες συνήθως επιλέγεται μονή τιμή για το k. Η επιλογή της βέλτιστης τιμής για το k είναι εξαρτώμενη των δεδομένων και η αύξηση της μπορεί να μειώσει την επίδραση του θορύβου αλλά κάνει τα όρια επιλογής λιγότερο διακριτά, μιας και η κλάση με τα περισσότερα δείγματα θα υπερισχύει.

2.1.4 Multi-Layer Perceptron (MLP)

Ο Multi-Layer Perceptron (MLP) είναι ένας αλγόριθμος επιβλεπόμενης μάθησης πρόσθιας τροφοδότησης που μαθαίνει μία συνάρτηση $f:R^{inp} \rightarrow R^{out}$, και εκπαιδεύεται στο σύνολο δεδομένων. Το inp είναι ο αριθμός των χαρακτηριστικών των δεδομένων εισόδου και το out ο αριθμός των κλάσεων που τοποθετούνται τα δεδομένα. Πέρα από τα 2 παραπάνω επίπεδα, υπάρχει και τουλάχιστον 1 ενδιάμεσο επίπεδο που καλείται κρυφό επίπεδο (hidden layer).



Εικόνα 2.5: Ένα απλό MLP δίκτυο με 3 νευρώνες εισόδου (χαρακτηριστικά), 4 νευρώνες στο μοναδικό κρυφό επίπεδο και 2 νευρώνες εξόδου (κλάσεις ταξινόμησης των χαρακτηριστικών) (Πηγή: https://becominghuman.ai/multi-layer-perceptron-mlp-models-on-real-world-banking-data-f6dd3d7e998f)



Κάθε νευρώνας ονομάζεται perceptron και συνδέεται με όλους του επόμενου επιπέδου μέσω βαρών, ενώ η συνάρτηση ενεργοποίησης υπάρχει σε όλους τους νευρώνες πέρα από αυτούς του επιπέδου εισόδου.

Εικόνα 2.6: Ένας νευρώνας (perceptron) που φαίνονται όλα τα συστατικά του, πιο συγκεκριμένα τα βάρη (weights), η συνάρτηση ενεργοποίησης (activation function) καθώς και η πόλωση (bias). (Πηγή: https://machinelearningmastery.com/neural-networks-crash-course/)

Η ιδέα του αλγορίθμου υπάρχει από το 1980, ωστόσο το ενδιαφέρον για αυτόν ανανεώθηκε πρόσφατα λόγω της μεγάλης επιτυχίας της βαθιάς μάθησης (κυρίως λόγω της χρήσης πολλών κρυφών επιπέδων).

Η διαδικασία της μάθησης συμβαίνει όταν ανανεώνονται τα βάρη των νευρώνων μετά την επεξεργασία κάθε δεδομένου εισόδου και βασίζεται στην μέτρηση του σφάλματος στην έξοδο (προβλεπόμενη έξοδος) και την σύγκριση του με την επιθυμητή έξοδο. Αναπαριστούμε το σφάλμα για το n-οστό δεδομένο και για τον νευρώνα i ως:

$$e_i(n) = d_i(n) - y_i(n)$$

όπου d είναι η πραγματική τιμή και y η προβλεπόμενη [14].

Τα βάρη των κόμβων ρυθμίζονται με την ελαχιστοποίηση της παρακάτω συνάρτησης:

$$E(n) = \frac{1}{2}\sum_j e_j^2(n)$$

Για να υπολογιστεί αυτό αποτελεσματικά χρησιμοποιείται ο αλγόριθμος του back-propagation, ο οποίος αποτελεί γενίκευση της μεθόδου ελαχίστων τετραγώνων [14]. Ο αλγόριθμος αυτός υπολογίζει την παράγωγο του σφάλματος ως προς κάθε βάρος του δικτύου με τον κανόνα της αλυσίδας, σε 1 επίπεδο κάθε φορά, ξεκινώντας από το τελευταίο επίπεδο και συνεχίζοντας προς τα πίσω ώστε να αποφευχθούν περιττοί υπολογισμοί ενδιάμεσων όρων στον κανόνα της αλυσίδας. Έτσι καθίσταται εφικτή η χρήση μεθόδων παραγώγων (gradient methods) για την εκπαίδευση MLP δικτύων. Άλλες παραδοχές που χρησιμοποιούνται είναι η gradient descent ή η stochastic gradient descent που υπολογίζουν τοπικά ελάχιστα [15]. Χρησιμοποιώντας την μέθοδο κατάβασης κλίσης (gradient descent), η μεταβολή των βαρών των νευρώνων γίνεται με βάση τον παρακάτω τύπο :

$$\Delta_{w_{ij}}(n) = -\eta \frac{\partial E(n)}{\partial u_j(n)} y_i(n)$$

όπου η είναι ο ρυθμός μάθησης και y_i η έξοδος του νευρώνα [14]. Αποδεικνύεται ότι:

$$-\frac{\partial \mathcal{E}(n)}{\partial u_j(n)} = e_j(n)\varphi'(u_j(n))$$

όπου φ είναι η συνάρτηση ενεργοποίησης [14].

Ωστόσο υπάρχει η πιθανότητα να δημιουργηθεί το πρόβλημα της εξαφανιζόμενης κατάβασης κλίσης (vanishing gradient descent problem), όπου η παράγωγος είναι τόσο πολύ μικρή που αποτρέπει τον νευρώνα από το να αλλάξει τιμή. Αντίστοιχα, υπάρχει και το δυϊκό πρόβλημα, όπου εάν η παράγωγος της συνάρτησης ενεργοποίησης επιτρέπει μεγάλες τιμές μπορεί να εμφανιστεί το πρόβλημα της εκρηγνυόμενης κλίσης κατάβασης (exploding gradient descent problem) όπου οι ενημερώσεις των βαρών είναι μεγάλες. Τα παραπάνω προβλήματα μπορούν να περιοριστούν με πολλούς τρόπους όπως Long Short-Term Memory (LSTM), Residual networks καθώς και με χρήση άλλης συνάρτησης ενεργοποίησης όπως η ReLU (rectifiers).

2.1.5 Τυχαία δάση – Random Forest

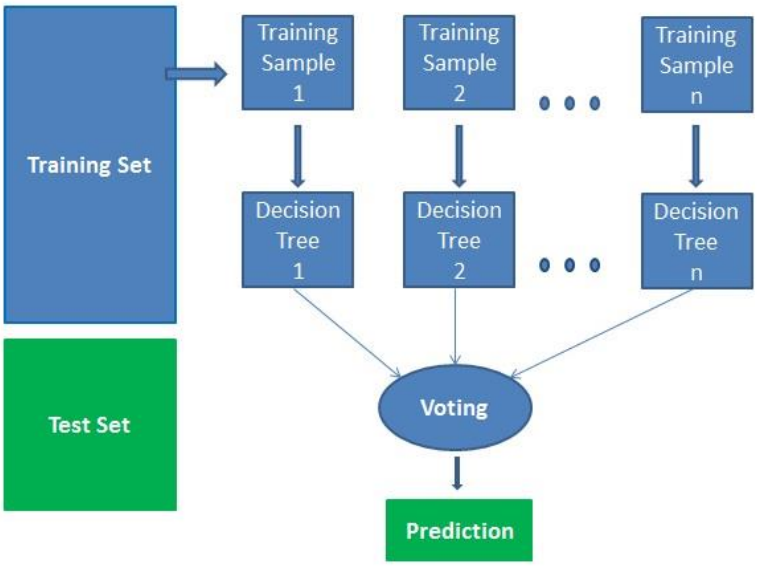
Τα Τυχαία Δάση (Random Forest) είναι μία τεχνική Δένδρων Απόφασης και Μάθησης Συνόλου (ensemble learning) που χρησιμοποιείται σε πληθώρα εφαρμογών, είτε ταξινόμησης είτε παλινδρόμησης ή και άλλων μεθόδων.

Τα βήματα της τεχνικής είναι τα εξής [16]:

1. Επέλεξε τυχαία δείγματα από το σύνολο δεδομένων.
2. Για κάθε ένα από τα δείγματα φτιάξε ένα δένδρο απόφασης και πάρε μία πρόβλεψη από αυτό.
3. Πάρε μία ψήφο για κάθε πρόβλεψη.
4. Επέλεξε τη πρόβλεψη με τις περισσότερες ψήφους ως την τελική.

Το μοντέλο ονομάζεται δάσος επειδή αποτελείται από πολλά δένδρα και καλείται τυχαίο λόγω του τυχαίου τρόπου δημιουργίας κάθε δένδρου. Η ανάπτυξη της τεχνικής οφείλεται στον Leo Breiman, ο οποίος περιέγραψε την δημιουργία ενός τέτοιου δάσους από ασυσχέτιστα δέντρα μέσω μιας διαδικασίας τύπου CART [17], συνδυασμένη με τυχαιοποιημένη βελτιστοποίηση κόμβων και ενσακκίσεως (bagging). Το bagging οδηγεί σε καλύτερη επίδοση του μοντέλου αφού μειώνει την διακύμανση του, χωρίς να αυξάνεται η πόλωση (bias). Αυτό σημαίνει ότι παρότι το κάθε δένδρο ξεχωριστά είναι ευαίσθητο στον θόρυβο των δεδομένων εισόδου, αντιθέτως ο μέσος όρος πολλών δένδρων δεν είναι, υποθέτοντας ωστόσο ότι τα δένδρα είναι ασυσχέτιστα μεταξύ τους. Η παραπάνω διαδικασία αφορά τα δένδρα. Η μόνη αλλαγή στα τυχαία δάση είναι ότι χρησιμοποιούν ένα τροποποιημένο αλγόριθμο μάθησης για δένδρα που επιλέγει ένα τυχαίο υποσύνολο των χαρακτηριστικών (feature bagging).

Η χρήση του μέσου όρου στα μεμονωμένα δένδρα επιφέρει βελτίωση στην ορθότητα (accuracy) και τον έλεγχο ως προς την υπερπροσαρμογή (overfitting). Το τελευταίο επιτυγχάνεται κυρίως μέσω της ικανότητας που έχει να εκτιμάει την σπουδαιότητα-αξία κάθε χαρακτηριστικού. Ο Random Forest χρησιμοποιεί την gini ή αλλιώς MDI (mean decrease in impurity) για να υπολογίσει τη βαρύτητα κάθε χαρακτηριστικού. Η gini ουσιαστικά υπολογίζει πόσο καλά εφαρμόζει το μοντέλο ή μειώνεται το accuracy όταν αφαιρείται μια μεταβλητή. Όσο πιο μεγάλη η μείωση, τόσο πιο σημαντική είναι η μεταβλητή [16].



Εικόνα 2.7: Διαγραμματική απεικόνιση λειτουργίας τεχνικής Random Forest (Πηγή: [16])

2.1.6 Ενίσχυση κλίσης – Gradient Boosting

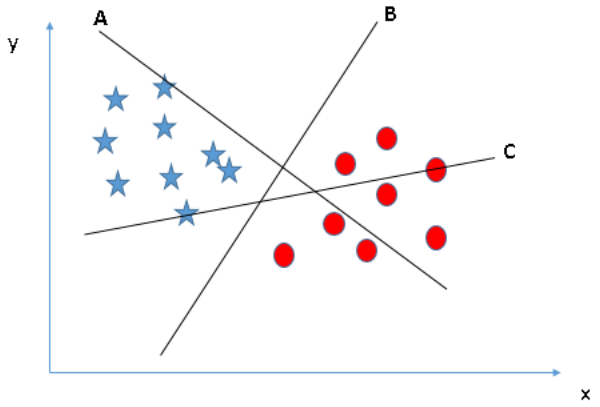
Η ενίσχυση κλίσης αποτελεί μια μέθοδο μηχανικής μάθησης και συγκεκριμένα Μάθησης Συνόλου, που χρησιμοποιείται τόσο σε προβλήματα ταξινόμησης όσο και σε προβλήματα παλινδρόμησης. Η λειτουργία της έγκειται στο συνδυασμό πολλών αδύναμων μοντέλων πρόβλεψης, κυρίως δένδρων αποφάσεων (Decision Trees).

Στηρίζεται στην τεχνική του Boosting, κατά την οποία σύμφωνα με τον Breiman [18], συνδυάζονται πολλοί «αδύναμοι» μαθητευόμενοι, δηλαδή μαθητευόμενοι με σφάλμα λίγο καλύτερο από αυτό της τυχαίας επιλογής και προκύπτει ένας ισχυρός μαθητευόμενος, δηλαδή ένας ταξινομητής που είναι πολύ κοντά στην πραγματική ταξινόμηση. Επιτυγχάνεται μείωση της πόλωσης και της διακύμανσης (variance).

Στο Gradient Boosting, σε κάθε επανάληψη προστίθεται ένα νέο αδύναμο μοντέλο στο σύνολο και εκπαιδεύεται με βάση το σφάλμα όλου του μέχρι τότε συνόλου. Η διαδοχική προσθήκη νέων μοντέλων αποσκοπεί στην μείωση του σφάλματος και σταματάει όταν δεν επιτυγχάνεται πλέον περαιτέρω βελτίωση.

2.1.7 Μηχανές διανυσμάτων υποστήριξης - SVM

Οι μηχανές διανυσμάτων υποστήριξης - SVM (Support Vector Machine) είναι ένα σύνολο μοντέλων επιβλεπόμενης μάθησης που χρησιμοποιείται για επίλυση προβλημάτων ταξινόμησης, παλινδρόμησης, καθώς και για την ανίχνευση των ακραίων τιμών. Στη μέθοδο αυτή βάζουμε κάθε δεδομένο στον n-διάστατο χώρο, όπου n είναι ο αριθμός των χαρακτηριστικών με τιμή του κάθε χαρακτηριστικού την τιμή που έχει το δεδομένο για αυτό το χαρακτηριστικό. Ύστερα προσπαθούμε να βρούμε υπερεπίπεδα που διαχωρίζουν καλύτερα τις κλάσεις και όχι μια υποκειμενική κατανομή για τα δεδομένα όπως κάνουν άλλα μοντέλα [19].



Εικόνα 2.8: Δεδομένα που ανήκουν σε 2 κλάσεις και μερικές ευθείες που προσπαθούν να τα κατηγοριοποιήσουν. Τα δεδομένα φαίνεται να έχουν χαρακτηριστικά, τα x και y.
(Πηγή: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/)

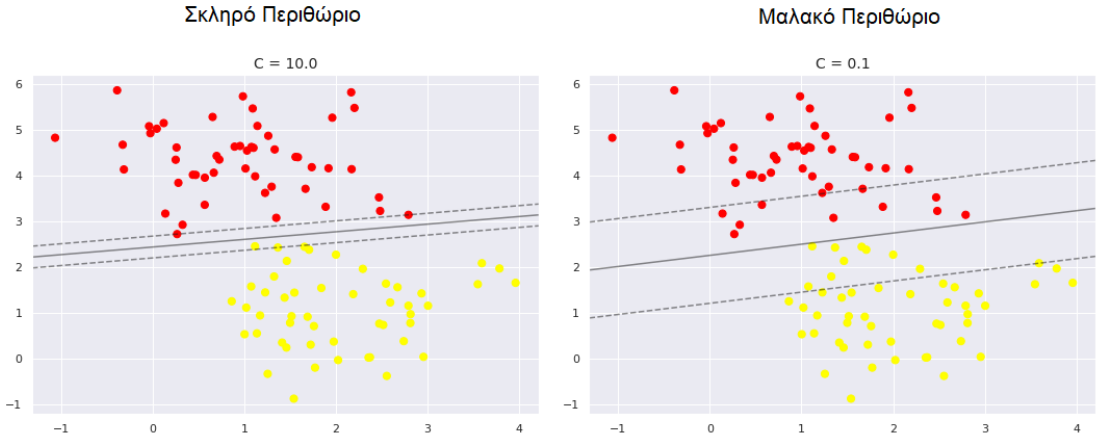
Ωστόσο το πλήθος των υπερεπιπέδων που χωρίζουν τα δεδομένα είναι άπειρο και συνεπώς πρέπει να βρεθεί ένας τρόπος για να επιλέξουμε το

βέλτιστο. Ο SVM επιλέγει το επίπεδο το οποίο κρατάει το μέγιστο περιθώριο μεταξύ των κλάσεων. Μεγιστοποιώντας το περιθώριο υπάρχουν ισχυρές ενδείξεις ότι νέα δεδομένα θα ταξινομούνται σωστά. Έτσι στην εικόνα 2.8 θα επέλεγε την ευθεία Β.

Τα δεδομένα τα οποία βρίσκονται κοντινότερα στο υπερεπίπεδο ονομάζονται διανύσματα υποστήριξης (Support Vectors) μιας και είναι τα μόνα που επηρεάζουν την θέση του υπερεπιπέδου. Χρησιμοποιώντας μόνο αυτά τα σημεία μπορούμε να μεγιστοποιήσουμε το περιθώριο του ταξινομητή, ενώ η διαγραφή τους θα άλλαζε την θέση του βέλτιστου υπερεπιπέδου.

Ωστόσο ο SVM δε λειτουργεί καλά μόνο σε γραμμικά διαχωρίσιμα δεδομένα. Επιτρέπει να κατηγοριοποιηθούν και μη γραμμικά διαχωρίσιμα δεδομένα με την χρήση πυρήνων (kernels) μέσω των οποίων γίνεται μία έμμεση αντιστοιχία των δεδομένων σε χώρους χαρακτηριστικών υψηλών ή ακόμα και άπειρων διαστάσεων [20].

Ένας άλλος τρόπος για κατηγοριοποίηση δεδομένων που μπορεί να έχουν αλληλοεπικαλύψεις με δεδομένα διαφορετικών κλάσεων μπορούν να προσεγγιστούν από τον SVM μαλακώνοντας το περιθώριο διαχωρισμού.



Εικόνα 2.9: Στην εικόνα φαίνονται 2 περιπτώσεις, μία μαλακού και μία σκληρού περιθωρίου. Στο σκληρό περιθώριο παρατηρούμε ότι τα διανύσματα υποστήριξης είναι αυτά που είναι κοντινότερα στην ευθεία διαχωρισμού. Στην δεύτερη εικόνα παρατηρούμε ότι τα διανύσματα υποστήριξης δεν είναι αυτά που βρίσκονται κοντινότερα στην ευθεία διαχωρισμού και συνεπώς μπορούν να βρεθούν σημεία εντός του περιθωρίου. Έτσι ο SVM προσαρμόζεται καλύτερα σε καινούρια δεδομένα. (Πηγή: https://colab.research.google.com/drive/1_D-ylXhiJ5AP2BBFRNwQL0y1cF_VkkN2#scrollTo=a9WoArNEsH2N&line=13&uniqifier=1)

2.2 Βαθιά Μάθηση – Deep Learning

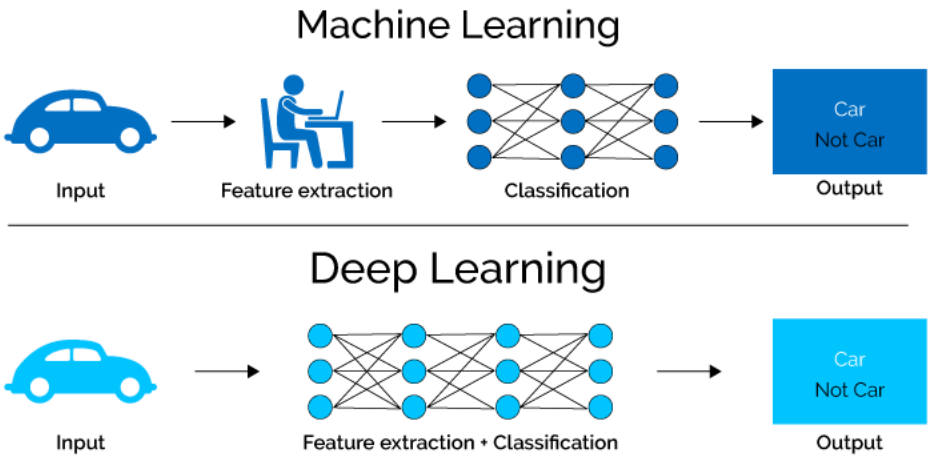
2.2.1 Ορισμός Βαθιάς Μάθησης

Η Βαθιά Μάθηση αποτελεί υποσύνολο της Μηχανικής Μάθησης. Η διαφορά έγκειται στο γεγονός ότι η Μηχανική Μάθηση χρησιμοποιεί αλγορίθμους σχεδιασμένους για συγκεκριμένες εργασίες, ενώ η βαθιά μάθηση είναι περισσότερο μια αναπαράσταση δεδομένων που βασίζεται σε πολλαπλά επίπεδα ενός πίνακα, όπου κάθε επίπεδο χρησιμοποιεί έξοδο από το προηγούμενο επίπεδο ως είσοδο [21].

Η Βαθιά Μάθηση δεν αποτελεί καινούρια εφεύρεση, καθώς οι πρώτοι αλγόριθμοι της χρονολογούνται από την δεκαετία του 1980 και το πρόβλημα

της αναγνώρισης χειρόγραφων ψηφίων των Αμερικανικών Ταχυδρομείων [22]. Ωστόσο, η ανάπτυξη της τα τελευταία χρόνια έχει γνωρίσει εκρηκτική αύξηση, συμβάλλοντας στην επίλυση προβλημάτων που η τεχνητή νοημοσύνη δεν μπορούσε να αντιμετωπίσει επί σειρά ετών. Πλέον, η εφαρμογή της είναι ευρεία σε πολλούς επιστημονικούς κλάδους. Το μεγάλο αρνητικό της Βαθιάς Μάθησης είναι ότι απαιτεί μεγάλη υπολογιστική ισχύ και ως εκ τούτου χρειάζεται περισσότερο χρόνο εκπαίδευσης σε σχέση με τους κλασσικούς αλγορίθμους μηχανικής μάθησης.

Υπάρχουν πολλές διαφορετικές αρχιτεκτονικές στη Βαθιά Μάθηση με τις κυριότερες να είναι τα Βαθιά Νευρωνικά Δίκτυα, τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks) και τα Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks).



Εικόνα 2.10: Σύγκριση Μηχανικής-Βαθιάς Μάθησης (Πηγή: [21])

2.2.2 Συναρτήσεις ενεργοποίησης – Activation functions

Στη μηχανική μάθηση, η συνάρτηση ενεργοποίησης είναι η συνάρτηση που καθορίζει την έξοδο του νευρώνα-perceptron. Με μη γραμμικές συναρτήσεις ενεργοποίησης είναι εφικτή η μάθηση σε πιο περίπλοκα προβλήματα με χρήση μικρού αριθμού νευρώνων. Οι συναρτήσεις ενεργοποίησης χρησιμοποιούνται όταν είναι απαραίτητη η κανονικοποίηση της εξόδου του νευρώνα ή η εισαγωγή μη γραμμικότητας στο σύστημα. Οι πιο σημαντικές εξ αυτών είναι:

Δυαδική βηματική: Πρόκειται για μη γραμμική συνάρτηση, μη παραγωγίσιμη που μετατρέπει θετική έξοδο σε 1.

$$f(x) = \begin{cases} 0 \; if \; x < 0 \\ 1 \; if \; x \geq 0 \end{cases}$$

Σιγμοειδής συνάρτηση: Είναι παραγωγίσιμη, μη γραμμική συνάρτηση με πεδίο ορισμού το σύνολο των πραγματικών και πεδίο τιμών [0,1]. Χρησιμοποιείται συνήθως στο επίπεδο εξόδου νευρώνα για κατηγοριοποίηση δυαδικού προβλήματος, αφού πρόκειται για συμμετρική συνάρτηση ως προς το 0.

$$S(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}}$$

Softmax: Η συνάρτηση αυτή χρησιμοποιείται κυρίως ως συνάρτηση νευρώνα εξόδου για κατηγοριοποίηση σε πολλές κλάσεις. Συγκεκριμένα, δέχεται ως είσοδο ένα διάνυσμα και επιστρέφει ένα διάνυσμα με κανονικοποιημένες εξόδους για κάθε κλάση.

$$Softmax(y_i) = \frac{e^{y_i}}{\sum_{i=1}^{N} e^{y_i}}$$

Συνάρτηση υπερβολικής εφαπτομένης: Πρόκειται για παραγωγίσιμη, μη γραμμική συνάρτηση και ισούται με τον λόγο της συνάρτησης υπερβολικού ημιτόνου προς το υπερβολικό συνημίτονο.

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

ReLU (Rectified Linear Unit): Η συνάρτηση αυτή είναι η πιο συνηθισμένη συνάρτηση ενεργοποίσης στα νευρωνικά δίκτυα. Αυτό οφείλεται στο γεγονός ότι είναι απλή στον υπολογισμό καθώς και η παράγωγος της έχει σταθερή τιμή: 1 για θετικές τιμές εισόδου και 0 για αρνητικές. Η χρήση της σε ένα βαθύ νευρωνικό δίκτυο αποτρέπει την εμφάνιση του προβλήματος «vanishing gradient», που εμφανίζει συχνά η σιγμοειδής συνάρτηση. Το πρόβλημα αυτό αναφέρεται στην τάση της παραγώγου ενός νευρώνα να τείνει στο 0 για μεγάλες τιμές εισόδου. Ωστόσο η τιμή 0 για τις αρνητικές τιμές εισόδου μπορεί να εμφανίσει το πρόβλημα «dying ReLU», όπου τα βάρη του νευρώνα ενημερώνονται έτσι ώστε ο νευρώνας να μην μπορέσει να ενεργοποιηθεί από κανένα δεδομένο εισόδου, δίνοντας πάντα παράγωγο 0. Αυτό αντιμετωπίζεται με κάποιες παραλλαγές της ReLU όπως η ELU ή η Leaky ReLU [23].

$$ReLU(x) = \max(0\,, x)$$

Η πληθώρα των συναρτήσεων ενεργοποίησης οφείλεται στα διαφορετικά χαρακτηριστικά που παρέχει η κάθε μία. Πέρα από τις εμπειρικές επιδόσεις, οι συναρτήσεις ενεργοποίησης στηρίζονται σε μαθηματικές ιδιότητες, οι κυριότερες από τις οποίες είναι :

1) Nonlinear – Μη γραμμικότητα: Όταν η συνάρτηση ενεργοποίησης είναι μη γραμμική τότε ένα νευρωνικό δίκτυο 2 στρωμάτων αποδεικνύεται ότι είναι μία καθολική προσέγγιση συνάρτησης.
2) Range – Πεδίο τιμών: Όταν το πεδίο τιμών της συνάρτησης ενεργοποίησης είναι πεπερασμένο τότε μέθοδοι βασισμένες στην παράγωγη τείνουν να είναι πιο σταθερές. Απεναντίας, όταν το πεδίο τιμών είναι άπειρο η εκπαίδευση γίνεται πιο αποδοτικά γιατί μεταβάλλονται τα περισσότερα βάρη.
3) Continuously differentiable – Συνεχώς διαφορίσιμες: Αυτή η ιδιότητα είναι επιθυμητή για την ενεργοποίηση μεθόδων βελτιστοποίησης παραγώγου (gradient-based optimization methods). Για παράδειγμα, η ReLU δεν είναι διαφορίσιμη.

2.2.3 Αλγόριθμοι Βελτιστοποίησης

Για να προσαρμόζονται σωστά τα βάρη ενός νευρωνικού δικτύου πρέπει να υπάρχει μια συνάρτηση κόστους την οποία προσπαθούμε να μεγιστοποιήσουμε ή να ελαχιστοποιήσουμε. Πρόκειται για μια συνάρτηση Loss(θ), όπου θ το διάνυσμα των παραμέτρων του νευρωνικού δικτύου. Η βελτιστοποίηση της

συνάρτησης κόστους γίνεται κατά την εκπαίδευση του δικτύου με ανανέωση των παραμέτρων. Συνεπώς η επιλογή του σωστού αλγορίθμου βελτιστοποίησης συμβάλει στην αποτελεσματική εκπαίδευση του δικτύου αλλά και την σύγκλιση προς ένα τοπικό ή ολικό ελάχιστο ή μέγιστο.

Gradient Descent Algorithm (Αλγόριθμος Καθόδου Κλίσης)

Πρόκειται για αλγόριθμο βελτιστοποίησης για ελαχιστοποίηση μιας συνάρτησης κόστους μέσω επαναληπτικής κίνησης προς την κατεύθυνση της πιο απότομης κατάβασης που ορίζεται από το αρνητικό της κλίσης. Η κλίση υπολογίζεται με την μέθοδο της οπίσθιας διάδοσης (backpropagation) όπου τα βάρη ανανεώνονται από τα τελευταία layers (στρώματα) του δικτύου προς τα πρώτα, ελαχιστοποιώντας την συνάρτηση κόστους. Ο αλγόριθμος, παρά την ευκολία υπολογισμού του, δεν ενδείκνυται για μεγάλα δίκτυα μιας και απαιτεί τον υπολογισμό της κλίσης πάνω σε ολόκληρο το δίκτυο, συνεπώς συγκλίνει αργά. Η ανανέωση των βαρών του δικτύου γίνεται σύμφωνα με τον παρακάτω τύπο:

$$\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{old} - \eta * \frac{1}{n} * \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{n} L(\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\theta}_{old}), y_i)$$

όπου η είναι μια πραγματική τιμή που ονομάζεται ρυθμός μάθησης, n είναι το πλήθος των δειγμάτων x_i του συνόλου εκπαίδευσης, y_i η μεταβλητή εξόδου και h η συνάρτηση που προσεγγίζει την συνάρτηση εξόδου.

Stochastic Gradient Descent Algorithm (Αλγόριθμος Στοχαστικής Καθόδου Κλίσης)

Πρόκειται για παραλλαγή του Gradient Descent Algorithm με μοναδική διαφορά το ότι τα βάρη ανανεώνονται με τον υπολογισμό της κλίσης ενός μόνο δείγματος ή ενός batch (δέσμης) από το σύνολο δεδομένων. Έτσι συγκλίνει πιο γρήγορα. Εάν επιλεχθεί ένα batch αποτελούμενο από m στιγμιότυπα τότε ο τύπος ανανέωσης των βαρών υπολογίζεται ως:

$$\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{old} - \eta * \frac{1}{m} * \nabla_{\boldsymbol{\theta}} \sum_{i=1}^{m} L(\boldsymbol{h}(\boldsymbol{x}_i, \boldsymbol{\theta}_{old}), y_i)$$

Momentum (Ορμή)

Όταν η καμπύλη της συνάρτησης κόστους έχει πολλά τοπικά ελάχιστα τότε η εκπαίδευση αργεί να συγκλίνει ακόμα και με την στοχαστική μέθοδο που συζητήθηκε παραπάνω. Αυτό οφείλεται στις ταλαντώσεις γύρω από τα τοπικά ελάχιστα. Την λύση σε αυτό το πρόβλημα έρχεται να δώσει η μέθοδος momentum. Αντί να χρησιμοποιεί την παράγωγο μόνο από το τρέχον βήμα, αθροίζει τις παραγώγους προηγούμενων βημάτων για να βρει προς ποια κατεύθυνση θα κινηθεί. Η ανανέωση των βαρών ενός μοντέλου μέσω ενός

batch m δειγμάτων με την προσθήκη του momentum στη Στοχαστική Κατάβαση Κλίσης υπολογίζεται ως εξής:

Θέτοντας:

$$\boldsymbol{g} = \frac{\mathbf{1}}{\boldsymbol{m}} * \nabla_{\theta} \sum_{i=1}^{m} \boldsymbol{L}(\boldsymbol{h}(\boldsymbol{x_i}, \boldsymbol{\theta_{old}}), \boldsymbol{y_i})$$

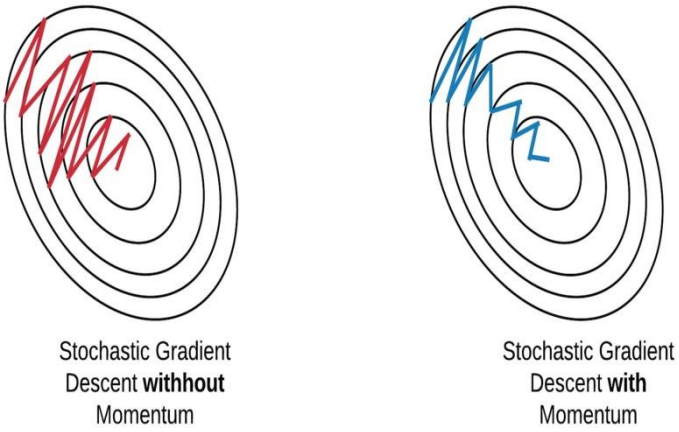
Η εναλλαγή του momentum γίνεται από τον τύπο:

$$\boldsymbol{u}_{new} = \alpha * \boldsymbol{u}_{old} - \eta * \boldsymbol{g}$$

Και έτσι καταλήγουμε στα ανανεωμένα βάρη:

$$\boldsymbol{\theta}_{new} = \boldsymbol{\theta}_{old} + \boldsymbol{u}_{new}$$

όπου α είναι μία παράμετρος με τιμή στο διάστημα [0, 1] και καθορίζει το βάρος που θα δίνεται στο momentum στην επιλογή των βαρών.



Εικόνα 2.11: Στην εικόνα παρουσιάζεται ο αλγόριθμος SGD με και χωρίς την χρήση του momentum. Παρατηρούμε γρηγορότερη σύγκλιση και μείωση των ταλαντώσεων μεταξύ τοπικών βέλτιστων.
(Πηγή: https://eloquentarduino.github.io/2020/04/stochastic-gradient-descent-on-your-microcontroller/)

Adam (Adaptive Moment Estimation)

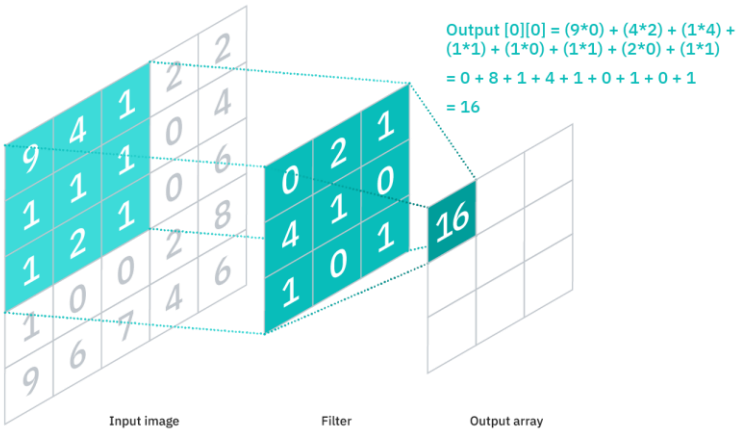
Πρόκειται για αλγόριθμο που βελτιστοποιεί την συνάρτηση κόστους με την προσθήκη momentum. Ο αλγόριθμος βασίζεται στην προσαρμοστική (adaptive) εκτίμηση των ροπών $1^{ης}$ και $2^{ης}$ τάξης δηλαδή τον μέσο όρο και την διακύμανση. Έτσι επιτυγχάνεται η γρηγορότερη σύγκλιση και η αντιμετώπιση προβλημάτων των παραπάνω αλγορίθμων [24].

2.2.4 Συνελικτικά Νευρωνικά Δίκτυα - CNN

Τα Συνελικτικά Νευρωνικά Δίκτυα (CNN) ανήκουν στην κατηγορία των βαθιών νευρωνικών δικτύων με κύρια χρήση την ανάλυση και κατηγοριοποίηση εικόνων. Σε αντίθεση με τα δίκτυα MLP, όπου το κάθε επίπεδο συνδέεται με όλους τους νευρώνες του επόμενου με συνέπεια να γίνονται ευάλωτα στο overfitting, τα CNN εκμεταλλεύονται το ιεραρχικό μοτίβο στα δεδομένα και μαθαίνουν μοτίβα αυξανόμενης πολυπλοκότητας. Βασίζονται επίσης στο γεγονός ότι γειτονικά pixels σχετίζονται μεταξύ τους, σχηματίζοντας μοτίβα τα οποία μπορούν να εντοπιστούν εύκολα. Τα CNN χρησιμοποιούν την πράξη της συνέλιξης, δηλαδή το εσωτερικό γινόμενο (dot product) μήτρας με συρόμενα φίλτρα (kernels) [22] [25] [26].

Τα Συνελικτικά Νευρωνικά Δίκτυα απαρτίζονται από διάφορα στρώματα. Το πρώτο είναι ένα στρώμα εισόδου (Input Layer) όπου περιέχει σαν παράμετρο τις διαστάσεις των δεδομένων. Στην συνέχεια ακολουθεί ένα Συνελικτικό στρώμα (Convolutional Layer), το οποίο αποτελείται από ένα πλήθος φίλτρων και από υπερπαραμέτρους που αφορούν την πράξη της συνέλιξης όπως παραγέμισμα (padding), δρασκελιά (stride). Το σύνολο των εκπαιδευόμενων παραμέτρων προέρχεται μόνο από το γινόμενο του πλήθους και των διαστάσεων των kernels. Το ότι οι παράμετροι είναι λιγότεροι συνεπάγεται ότι μπορούν να δημιουργηθούν πολύ βαθιά δίκτυα, ενώ ταυτόχρονα αποφεύγονται τα προβλήματα vanishing gradient και exploding gradient που εμφανίζονται στα κλασσικά MLP νευρωνικά δίκτυα. Το αποτέλεσμα μετά την πράξη της συνέλιξης ονομάζεται χάρτης ενεργοποίησης (activation map ή feature map) [27].

Υπάρχει η δυνατότητα να στοιβάξουμε πολλά φίλτρα με αποτέλεσμα να δημιουργηθούν τόσα activation maps όσα και το πλήθος των φίλτρων. Ως εκ τούτου, ένα συνελικτικό δίκτυο (ConvNet) αποτελείται από πολλά convolution layers που μεταξύ τους υπάρχουν συναρτήσεις ενεργοποίησης (activation functions). Έτσι επιτυγχάνεται η μάθηση όλο και πιο πολύπλοκων μοτίβων. Ως activation function χρησιμοποιείται κυρίως η ReLU.

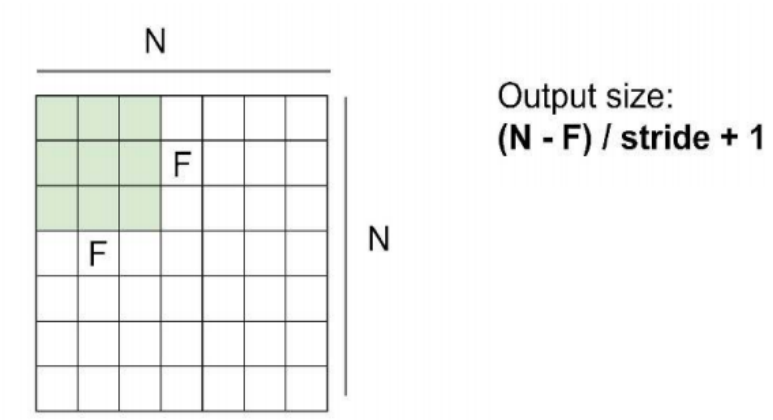


Εικόνα 2.12: Συνέλιξη μεταξύ ενός 5x5 πίνακα με ένα 3x3 φίλτρο (Πηγή: https://www.ibm.com/cloud/learn/convolutional-neural-networks)

Όσον αφορά τις άλλες υπερπαραμέτρους, το stride υποδεικνύει πόσα pixel θα μετακινείται το φίλτρο κάθε φορά. Το padding είναι απαραίτητο γιατί χωρίς αυτό υπάρχει πιθανότητα να χαθεί πληροφορία στα άκρα της εικόνας. Επιπλέον, ελέγχει το χωρικό μέγεθος της εικόνας. Έτσι υπάρχει η δυνατότητα

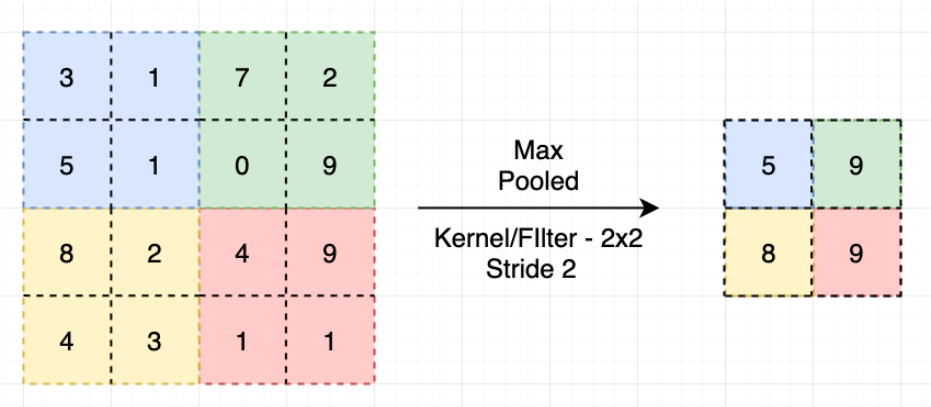
να διατηρήσουμε τις αρχικές διαστάσεις της εικόνας που ονομάζεται συνήθως same padding.

Στην περίπτωση του same padding, για φίλτρα μεγέθους FxF και zero-padding μεγέθους $^{(F-1)}/_{2}$, έχουμε διατήρηση της διάστασης της αρχικής εικόνας στο επόμενο επίπεδο.



Εικόνα 2.13: Υπολογισμός του πίνακα εξόδου, εάν το φίλτρο είναι μεγέθους FxF και η εικόνα μεγέθους NxN, χωρίς χρήση padding (Πηγή: https://eclass.ails.ece.ntua.gr/modules/document/file.php/102/%CE%94%CE%B9%CE%B1%CE%BB%CE%AD%CE%BE%CE%B5%CE%B9%CF%82/CNN.pdf)

Στην συνέχεια ακολουθεί συνήθως ένα pooling layer, το οποίο ουσιαστικά κάνει μια υποδειγματοληψία (downsampling) των εικόνων. Αυτό το layer είναι απαραίτητο για την μείωση των διαστάσεων και εφαρμόζεται σε κάθε activation map ξεχωριστά. Επιπλέον εξάγει τα κυρίαρχα χαρακτηριστικά ενώ ταυτόχρονα λειτουργεί ως καταστολέας θορύβου (noise suppressant) αφαιρώντας τις θορυβώδεις ενεργοποιήσεις. Παρότι με το παραπάνω layer χάνεται αρκετή πληροφορία, υπάρχουν και πλεονεκτήματα που αφορούν την μείωση της πολυπλοκότητας και την μείωση της πιθανότητας για overfit [28] [29].



Εικόνα 2.14: Εφαρμογή Max Pooling στην αρχική εικόνα με ένα 2x2 φίλτρο με stride 2, δηλαδή το φίλτρο σύρεται κατά 2 pixel κάθε φορά. (Πηγή: https://ai.plainenglish.io/pooling-layer-beginner-to-intermediate-fa0dbdce80eb)

Έπειτα, ακολουθεί ένα Fully Connected Layer, δηλαδή ένα MLP δίκτυο με όλους τους νευρώνες του ενός στρώματος να συνδέονται με όλους του επόμενου. Εδώ θα γίνει και η ταξινόμηση των εικόνων μέσω των χαρακτηριστικών που εξάγονται από τα προηγούμενα επίπεδα [30].

Υπάρχουν και κάποια ακόμα στρώματα που χρησιμοποιήσαμε όπως το BatchNormalization layer , το οποίο για κάθε batch διατηρεί την μέση τιμή κοντά στο 0 και την τυπική απόκλιση στο 1. Το συγκεκριμένο στρώμα είναι χρήσιμο γιατί επιλύει το πρόβλημα internal covariance shift δηλαδή το πρόβλημα κατά το οποίο όταν αλλάζει η κατανομή των εισόδων του δικτύου, οι παράμετροι προσαρμόζονται στην νέα κατανομή κατά την εκπαίδευση. Αυτό βοηθάει γιατί είναι ένα βήμα για την σταθεροποίηση της κατανομής των εισόδων [31].

Ένα σημαντικό πρόβλημα των βαθιών νευρωνικών δικτύων είναι ότι τείνουν να κάνουν overfit για μικρό αριθμό δεδομένων εκπαίδευσης. Μια πιθανή λύση στην καταπολέμηση του παραπάνω προβλήματος είναι η χρήση dropout layers. Με τον τρόπο αυτό αφαιρείται ένα ποσοστό των νευρώνων, εννοώντας ότι διαγράφεται προσωρινά μαζί με όλες τις εισερχόμενες και εξερχόμενες συνδέσεις του. Με αυτόν τον τρόπο αφαιρούνται από το δίκτυο οι αλληλεπιδράσεις μεταξύ νευρώνων, κατά τις οποίες ο ένας τείνει να διορθώσει τα λάθη άλλου και συνεπώς οι αλληλεπιδράσεις αυτές δεν γενικεύονται στα δεδομένα ελέγχου [32].

2.2.5 Ανατροφοδοτούμενα Νευρωνικά Δίκτυα - RNN

Τα Ανατροφοδοτούμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks) είναι ένα είδος τεχνητών νευρωνικών δικτύων που χρησιμοποιούν ακολουθιακά δεδομένα ή δεδομένα χρονοσειρών. Ξεχωρίζουν από τα υπόλοιπα βαθιά νευρωνικά δίκτυα, καθώς εμπεριέχουν την έννοια της εσωτερικής κατάστασης (state ή memory), δηλαδή λαμβάνουν πληροφορίες από προηγούμενες εισόδους για να επηρεάσουν την τρέχουσα είσοδο και έξοδο [33].

Στη διαμόρφωση της εξόδου του δικτύου συμβάλλει ωστόσο και ένα διάνυσμα «κρυφής» κατάστασης (hidden state) που αναπαριστά όλες τις παρελθοντικές εισόδους στο δίκτυο. Λόγω αυτού μπορεί το RNN παρόλο που δέχεται την ίδια είσοδο να δώσει διαφορετική έξοδο.

2.2.6 Δίκτυα Μακράς Βραχυπρόθεσμης Μνήμης - LSTM

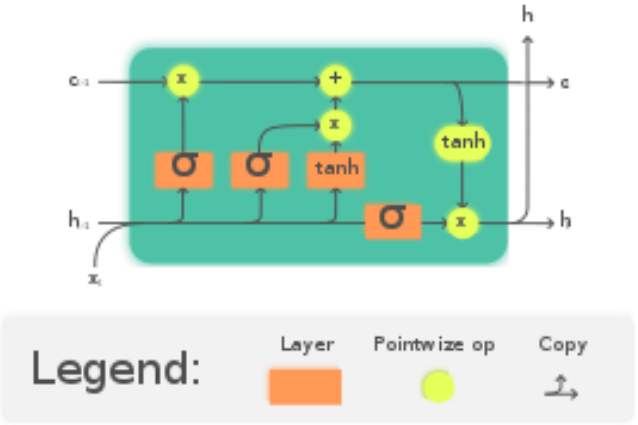
Τα δίκτυα μακράς βραχυπρόθεσμης μνήμης (Long Short-Term Memory) αποτελούν μια δημοφιλή αρχιτεκτονική RNN δικτύου, η οποία προτάθηκε το 1997 από τους Sepp Hochreiter και Jurgen Schmidhuber [34] ως λύση στο πρόβλημα του vanishing gradient. Τα RNN πάσχουν από το πρόβλημα της βραχυπρόθεσμης μνήμης, δηλαδή δυσκολεύονται να μεταφέρουν την πληροφορία όταν υπάρχουν μεγάλες ακολουθίες.

Τα LSTM ωστόσο διαφέρουν, καθώς καταφέρνουν να συνδυάζουν τόσο την μακροπρόθεσμη γνώση (long term) που έχουν αποκτήσει όσο και την βραχυπρόθεσμη πληροφορία (short term) που δέχονται. Αυτό οφείλεται στη δομή τους, καθώς στα hidden layers του νευρωνικού έχουν κύτταρα (cells), που διαθέτουν τρεις πύλες [35]:

- πύλη εισόδου: η οποία αποφασίζει ποιες πληροφορίες είναι σημαντικές για προσθήκη στο τρέχον βήμα
- πύλη εξόδου: η οποία αποφασίζει ποια πρέπει να είναι η επόμενη κρυφή κατάσταση

- πύλη λήθης: η οποία αποφασίζει τι είναι σημαντικό να κρατηθεί από τα προηγούμενα βήματα

Οι πύλες αυτές ελέγχουν την ροή της πληροφορίας που απαιτείται για την πρόβλεψη της εξόδου [33]. Οι διάφορες τροποποιήσεις στις πύλες είναι και ο κύριος λόγος που ο LSTM εμφανίζεται με πληθώρα παραλλαγών.



Εικόνα 2.15: Δομή κελιού LSTM
(Πηγή: https://en.wikipedia.org/wiki/Long_short-term_memory#LSTM_with_a_forget_gate)

2.2.7 Bidirectional - LSTM

Τα δίκτυα μακράς βραχυπρόθεσμης μνήμης διπλής κατεύθυνσης (Bidirectional LSTM) αποτελούν επέκταση των τυπικών LSTM και αποδεικνύονται πιο αποτελεσματικά σε προβλήματα κατηγοριοποίησης ακολουθιών. Αυτή η δομή επιτρέπει στα δίκτυα να έχουν τόσο πληροφορίες προς τα πίσω όσο και προς τα εμπρός σχετικά με την ακολουθία σε κάθε βήμα.

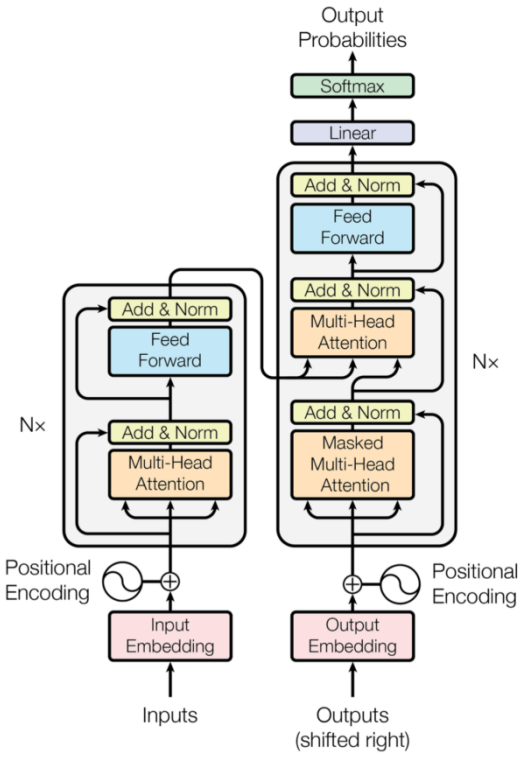
2.2.8 Μετασχηματιστές - Transformers

Οι μετασχηματιστές (transformers) είναι ένα μοντέλο βαθιάς μάθησης που υιοθετεί τον μηχανισμό της προσοχής (attention), δίνοντας βάρη στην επίδραση που έχουν τα διαφορετικά μέρη της εισόδου.

Η προσοχή στην μηχανική μάθηση είναι μια τεχνική που προσπαθεί να μιμηθεί την γνωστική προσοχή (cognitive attention), ενισχύοντας τα σημαντικά μέρη της εισόδου και εξασθενώντας τα υπόλοιπα. Η λογική είναι ότι σε αυτά τα λίγα αλλά σημαντικά δεδομένα θα πρέπει να αφιερωθεί η περισσότερη υπολογιστική ισχύς. Το ποιο μέρος των δεδομένων θα θεωρηθεί πιο σημαντικό από τα άλλα εξαρτάται από το περιεχόμενο της εισόδου και μαθαίνεται με εκπαίδευση με gradient descent.

Οι transformers είναι σχεδιασμένοι να διαχειρίζονται διαδοχικά δεδομένα εισόδου, ωστόσο δεν είναι απαραίτητο να γίνει η επεξεργασία τους με ίδια σειρά. Η λειτουργία προσοχής μπορεί να πλαισιωθεί σε οποιαδήποτε θέση της ακολουθίας εισόδου. Εξαιτίας αυτού του χαρακτηριστικού οι transformers παρέχουν μεγαλύτερο παραλληλισμό από παρόμοια ακολουθιακά μοντέλα.

Είναι ένα μοντέλο με αρχιτεκτονική κωδικοποιητή-αποκωδικοποιητή (encoder-decoder). Στον κωδικοποιητή έρχεται μια ακολουθία που την αντιστοιχίζει σε υψηλότερη διάσταση (n-διάστατος χώρος). Αυτό το αφηρημένο



διάνυσμα εισέρχεται στον αποκωδικοποιητή που το μετατρέπει στην ακολουθία εξόδου. Η ακολουθία εξόδου μπορεί να αποτελείται από προτάσεις άλλης γλώσσας, άλλα σύμβολα, ένα αντίγραφο της αρχικής ακολουθίας κ.ά. [36].

Οι transformers χρησιμοποιούνται κυρίως στο πεδίο της επεξεργασίας φυσικής γλώσσας (NLP) αλλά έχουν και άλλες εφαρμογές όπως την κατανόηση βίντεο.

Εικόνα 2.16: Το μοντέλο ενός transformer. Αριστερά είναι η στοίβα κωδικοποίησης αποτελούμενη από N_x πανομοιότυπα στρώματα. Παρατηρούμε attention layer, feed forward layer, residual blocks[1] και normalization layers. Η έξοδος τροφοδοτείται στην στοίβα αποκωδικοποίησης, όπου πέρα τον 2 υποστρωμάτων προστίθεται και ένα τρίτο που εκτελεί την πράξη masked attention, που αποτρέπει τον αποκωδικοποιητή από το να «κλέψει» κατά την εκπαίδευση. (Πηγή: [36])

2.3 Μετρικές Αξιολόγησης

Η ιδέα δημιουργίας μοντέλων μηχανικής μάθησης βασίζεται σε μία αρχή ανατροφοδότησης. Δημιουργούμε ένα μοντέλο, το εκπαιδεύουμε και παίρνουμε ανατροφοδότηση μέσω διαφόρων μετρικών έως ότου να επιτευχθεί ένα επιθυμητό επίπεδο σε αυτές. Αυτές οι μετρικές ονομάζονται μετρικές αξιολόγησης (evaluation metrics). Η επιλογή τους βασίζεται στην φύση του προβλήματος και οι συνηθέστερες είναι η ορθότητα (Accuracy), η ακρίβεια (Precision) και η ανάκληση (Recall). Για τον ορισμό των παραπάνω μετρικών σε ένα πρόβλημα 2 κλάσεων πρέπει να οριστούν οι έννοιες:

- Αληθές Θετικό (True Positive ή hit): Αριθμός δειγμάτων της πρώτης κλάσης που η πρόβλεψη για αυτά ήταν σωστή.
- Αληθές Αρνητικό (True Negative ή correct rejection): Αριθμός δειγμάτων της δεύτερης κλάσης που η πρόβλεψη για αυτά ήταν σωστή.
- Ψευδές Θετικό (False Positive ή underestimation): Αριθμός δειγμάτων της δεύτερης κλάσης που προβλέφθηκαν λανθασμένα ως δείγματα της πρώτης κλάσης.
- Ψευδές Αρνητικό (False Negative ή overestimation): Αριθμός δειγμάτων της πρώτης κλάσης που προβλέφθηκαν λανθασμένα ως δείγματα της δεύτερης κλάσης.

Οι παραπάνω τιμές ορίζουν τον πίνακα σύγχυσης (confusion matrix) [37]:

[1] residual blocks: Πρόκειται για ένα μπλοκ που πέρα από την μη γραμμική συνάρτηση προσθέτει μία σύνδεση παράκαμψης (skip connection) όπου η τιμή του δεδομένου περνάει ως έχει. Το αποτέλεσμα είναι το άθροισμα των δύο.

		Προβλεπόμενη κλάση	
		Negative	Positive
Πραγματική κλάση	Negative	TN	FP
	Positive	FN	TP

Πίνακας 2.1: Πίνακας σύγχυσης

Κάθε URL που εξετάζουμε θα ανήκει σε μία από αυτές τις τέσσερις κατηγορίες [37]:

- True Positive (TP, σωστά ταξινομημένο Phishing URL)
- True Negative (TN, σωστά ταξινομημένο Legitimate URL)
- False Positive (FP, Legitimate URL λανθασμένα ταξινομημένο ως Phishing URL)
- False Negative (FN, Phishing URL λανθασμένα ταξινομημένο ως Legitimate URL)

Επομένως, οι μετρικές που θα μας απασχολήσουν είναι:

Accuracy: Είναι ο λόγος των δειγμάτων που κατηγοριοποιήθηκαν σωστά στις κλάσεις τους ως προς το σύνολο των δειγμάτων.

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative}$$

Precision: Πρόκειται για τον λόγο των δειγμάτων που προβλέφθηκαν σωστά προς το σύνολο των δειγμάτων που προβλέφθηκαν ως στιγμιότυπα αυτής της κλάσης.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{True\ positive}{Total\ Predicted\ Positive}$$

Recall: Πρόκειται για τον λόγο των δειγμάτων μιας που προβλέφθηκαν σωστά προς όλα τα δείγματα αυτής της κλάσης.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{True\ positive}{Total\ Actual\ Positive}$$

FNR (False Negative Ratio): Πρόκειται για την συμπληρωματική τιμή του recall. Δηλαδή:

$$FNR = 1 - Recall = 1 - \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$\frac{True\ Positive + \ False\ Negative - True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{False\ Negative}{True\ Positive + False\ Negative}$$

2.4 Voting Scheme

Όπως θα δούμε σε επόμενο κεφάλαιο, κάθε ξεχωριστός ταξινομητής δεν μπορεί να είναι βέλτιστος ως προς όλες τις μετρικές, για το λόγο αυτό θα χρησιμοποιήσουμε ένα voting scheme δηλαδή ένα συνδυασμό των ταξινομητών προκειμένου να εξάγουμε τα βέλτιστα χαρακτηριστικά ως προς όλες τις μετρικές απόδοσης.

Σύμφωνα με την Kuncheva [38] υπάρχουν 4 είδη voting scheme ανάλογα με τον τρόπο δημιουργίας τους. Πιο συγκεκριμένα :

- Επίπεδο συνδυασμού (combination level)
- Επίπεδο ταξινόμησης (classifier level)
- Επίπεδο χαρακτηριστικών (feature level)
- Επίπεδο δεδομένων (data level)

Θα επικεντρωθούμε κυρίως στο επίπεδο ταξινόμησης, από το οποίο θα προκύψει η τεχνική που επιλέξαμε να χρησιμοποιήσουμε για την εφαρμογή μας.

Αρχικά θα αναφερθούμε στο simple majority voting scheme που αποτελεί έναν κανόνα απόφασης που επιλέγει μία εκ των πολλών εναλλακτικών βασισμένο στην προβλεπόμενη κλάση με τις περισσότερες ψήφους [39]. Θεωρώντας ότι έχουμε n ανεξάρτητους, ισοδύναμους ειδικούς που λαμβάνουν μοναδική απόφαση σχετικά με την κλάση του μη χαρακτηρισμένου δείγματος, τότε το δείγμα αυτό ταξινομείται στην κλάση όπου υπάρχει απόλυτη πλειοψηφία, δηλαδή απόφαση στην οποία συμφωνούν τουλάχιστον οι μισοί ειδικοί. Έχει αποδειχθεί παρά την απλότητα του αρκετά αποτελεσματικό σε πληθώρα εφαρμογών [40].

Στην συνέχεια, έπεται το Weighed majority voting scheme που αποτελεί γενίκευση του simple majority για ίδια βάρη προς κάθε ταξινομητή. Πιο συγκεκριμένα, η απόφαση του κάθε classifier πολλαπλασιάζεται με ένα βάρος που αντικατοπτρίζει την μεμονωμένη εμπιστοσύνη προς τις αποφάσεις του [39]. Όσο μεγαλύτερη αξιοπιστία έχει ο ειδικός προς τις αποφάσεις του, τόσο μεγαλύτερη η τιμή του βάρους που του ανατίθεται. Το άθροισμα των βαρών είναι ίσο με μονάδα. Επομένως, αν η απόφαση του k^{οστου} ειδικού να ταξινομήσει το άγνωστο δείγμα στην i^{οστή} κλάση δίνεται από το d_{ik} με $0 \leq i \leq m$, όπου m ο αριθμός των κλάσεων, τότε η τελική συνδυαστική απόφαση για ανάθεση στην κλάση I, λαμβάνει την μορφή $d_i^{com} = \sum_{i=1,2,\dots,m} \omega_k * d_{ik}$. Συνεπώς η κλάση y είναι αυτή που επιλέγεται αν το d_y^{com} είναι το μέγιστο [40]. Για να βρεθούν οι βέλτιστες τιμές των βαρών θα πρέπει να ελαχιστοποιούν την συνάρτηση σφάλματος που ορίζεται ως:

$$y \neq \text{true\_label για } max(d_y^{com})$$

Μια συνάρτηση απόφασης είναι βέλτιστη όταν ο παραπάνω τύπος ελαχιστοποιείται στο σύνολο των πιθανών αποφάσεων. Αν υποθέσουμε ανεξαρτησία μεταξύ των ειδικών και επίσης ότι αν η πιθανότητα να επιλέξει την κλάση i είναι p_i τότε η πιθανότητα να επιλέξει οποιαδήποτε άλλη κλάση ισοκατανέμεται σε αυτές, δηλαδή κάθε άλλη κλάση έχει πιθανότητα $\frac{1-p_i}{c-1}$, καταλήγουμε σε μια προσέγγιση ενός Majority Weighted Vote:

$$f^{opt}(x) = sign(\sum_{i=1}^{n} \omega_i * x_i),$$

Όπου τα βάρη ω_i δίνονται από την σχέση:

$$\omega_i = \log\left(\frac{p_i}{1 - p_i}\right), \qquad i \in [n]$$

όπου p_i η πιθανότητα ο ειδικός να επιλέξει την κλάση i [41].

Τέλος, υπάρχει και ο Naïve Bayes Combiner όπου υπολογίζει τα βάρη προσεγγίζοντας την από κοινού κατανομή πιθανότητας για κάθε κλάση με ένα σύνολο απαντήσεων των ειδικών. Ξεκινώντας από το θεώρημα του Bayes έχουμε για τα χαρακτηριστικά ότι:

$$\mathrm{P}(c \mid f_1, \dots, f_v) = \frac{p(c) * \mathrm{P}(f_1, \dots, f_v \mid c)}{\mathrm{P}(f_1, \dots, f_v)}$$

όπου f_i τα χαρακτηριστικά και c η μεταβλητή που αφορά την κλάση. Υποθέτοντας ανεξαρτησία μεταξύ των χαρακτηριστικών και συμβολίζοντας με $Z = \mathrm{P}(f_1, \dots, f_m)$, έχουμε από τον προηγούμενο τύπο:

$$\mathrm{P}(c \mid f_1, \dots, f_v) = \frac{1}{Z} p(c) * \prod_{i=1}^{v} p(f_i|c)$$

Παρατηρούμε ότι το Z είναι ένας πολλαπλασιαστικός παράγοντας και είναι ανεξάρτητος της μεταβλητής κλάσης c. Παίρνοντας ως τυχαίες μεταβλητές το σύνολο των απαντήσεων των ταξινομητών $\{e_1, \dots, e_k\}$ αντί των χαρακτηριστικών καταλήγουμε στην εξίσωση:

$$\mathrm{P}(c \mid e_1, \dots, e_k) = \frac{1}{Z} p(c) * \prod_{i=1}^{k} p(e_i|c)$$

Λαμβάνοντας υπόψη την σχέση:

$$\mathrm{P}(c, e_1, \dots, e_k) = \mathrm{P}(c \mid e_1, \dots, e_k) * Z$$

δηλαδή αντικαθιστώντας την δεσμευμένη πιθανότητα με την από κοινού, από τον προηγούμενο τύπο συμπεραίνουμε:

$$\mathrm{P}(c, e_1, \dots, e_k) = p(c) * \prod_{i=1}^{k} p(e_i|c)$$

Συνεπώς τα βάρη συσχετίζονται με την μεταβλητή κλάσης u με την σχέση:

$$\omega(e_1, \dots, e_k) = p(c = u) * \prod_{i=1}^{k} p(e_i|c = u)$$

Έτσι υπολογίζεται η κλάση $\hat{c}$ του μη χαρακτηρισμένου δείγματος x ως:

$$\hat{c} = \max_{u \in C} \sum_{i} \omega_u * r_{i,u}$$

$$r_{i,u} = \begin{cases} 1, \alpha\nu\ o\ \tau\alpha\xi\iota\nu o\mu\eta\tau\acute{\eta}\varsigma\ E_i\ \kappa\alpha\tau\alpha\chi\omega\rho\epsilon\acute{\iota}\ \tau o\ x\ \sigma\tau\eta\nu\ \kappa\lambda\acute{\alpha}\sigma\eta\ u \\ 0, \alpha\lambda\lambda o\acute{\upsilon} \end{cases}$$

Άρα δοθέντος κάθε δείγματος εισόδου x και συνόλου απαντήσεων των ειδικών $\{e_1, \dots, e_k\}$ υπολογίζονται τα βάρη $\omega_u(e_1, \dots, e_k)$, 1 ≤u ≤ m και η τελική απόφαση παίρνεται με βάση την εξίσωση του $\hat{c}$ [42].

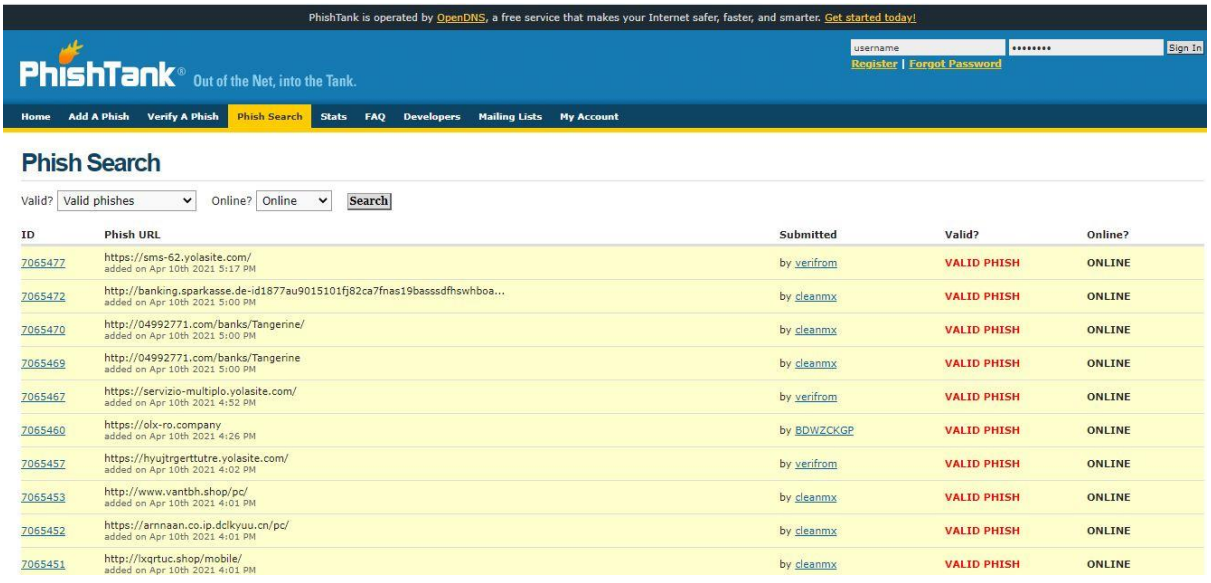
Κεφάλαιο 3

Ανίχνευση Phishing URLs

3.1 Συλλογή δεδομένων

3.1.1 Συλλογή πρωτογενών δεδομένων

Πρωταρχικό βήμα της εργασίας μας αποτέλεσε η συλλογή των κατάλληλων δεδομένων, στην περίπτωση μας URLs, προκειμένου να δημιουργήσουμε το dataset που θα χρησιμοποιηθεί για την εκπαίδευση των μοντέλων μας. Η συλλογή των legitimate URLs έγινε στις 24-3-2021 από τα αρχεία "cc_1.txt", "cc_2.txt", "cc_3.txt" που υπάρχουν διαθέσιμα στο Github [43]. Αντίστοιχα, τα phishing URLs συλλέχθηκαν στις 24-3-2021 από το αρχείο "phishing-links-ACTIVE.txt" στο Github [44], καθώς και χειροκίνητα στις 25-3-2021 τα πρώτα 110 αποτελέσματα από την ιστοσελίδα PhishTank [45]. Το PhishTank είναι μια anti-phishing ιστοσελίδα και στην ουσία αποτελεί μια βάση δεδομένων που ανανεώνεται συνεχώς με εγγραφές για Phishing URLs. Η πρόσβαση στα δεδομένα του είναι ελεύθερη και δωρεάν.



Εικόνα 3.1: PhishTank – Βάση Δεδομένων για Phishing URLs (Πηγή: [45])

3.1.2 Κατασκευή της συλλογής δεδομένων

Αρχικά, φροντίσαμε να ελέγξουμε ότι τα URLs μας είναι ενεργά (active), διαγράφοντας όσα δεν ανταποκρίθηκαν. Τα URLs δεν μπορούν ωστόσο να προσφέρουν άμεσα στην εκπαίδευση των μοντέλων. Για το λόγο αυτό κληθήκαμε να εξάγουμε κάποια χαρακτηριστικά που τα διακρίνουν και τα

καθιστούν ξεχωριστά. Με βάση το paper των Grega Vrbančič, Iztok Fister Jr., Vili Podgorelec [46], τα URLs χωρίζονται σε επιμέρους τμήματα όπως φαίνεται στην Εικόνα 3.2 και συγκεκριμένα στα Domain, Directory, File και Parameters.

Εικόνα 3.2: Διαχωρισμός ενός URL σε επιμέρους τμήματα (Πηγή: [46])

Σε κάθε τμήμα μετράμε το πλήθος κάποιων ειδικών χαρακτήρων (πχ -,#,@ κ.ά.), το μέγεθος του τμήματος κι ελέγχουμε αν εμφανίζονται κάποιες συγκεκριμένες λέξεις σε συγκεκριμένα τμήματα (πχ "client", "server", "script" κ.ά.), αν υπάρχει IP ή email στο τμήμα του Domain, καθώς και το πλήθος των φωνήεντων στο Domain. Επιπλέον, σύμφωνα με το paper με τίτλο "Phishing Websites Features" των Rami M. Mohammad, Fadi Thabtah, Lee McCluskey [47], υπάρχουν χαρακτηριστικά που βασίζονται σε εξωτερικές υπηρεσίες (WHOIS[2], HTTPS[3] Protocol, SSL[4] certificate κ.ά.). Επίσης, βασιζόμενοι στο άρθρο του SingTat στο Medium [48], φροντίσαμε να ελέγχουμε την ύπαρξη ή να μετράμε το πλήθος των εμφανίσεων κάποιων συγκεκριμένων HTTP Headers (πχ cookie, Strict-Transport-Security κ.ά.).

Πιο συγκεκριμένα καταλήξαμε στα παρακάτω χαρακτηριστικά τα οποία εξάγαμε από κάθε URL:

- **check_ssl**: Έλεγχος για ύπαρξη έγκυρου (valid) SSL πρωτοκόλλου (τιμές 0 False - 1 True)
- **qty_redirect**: Αριθμός ανακατευθύνσεων μέχρι την τελική ιστοσελίδα (αριθμητική τιμή)
- **url_shortened**: Έλεγχος αν έχει συντομευθεί το μήκος του αρχικού URL (τιμές 0 False - 1 True)
- **favicon**: Έλεγχος αν το favicon φορτώνεται από εξωτερικό domain (τιμές 0 False - 1 True)
- **dns_record**: Έλεγχος για ύπαρξη DNS εγγραφής για το domain στο WHOIS Database (τιμές 0 True - 1 False)
- **iFrame**: Το iFrame είναι μια ετικέτα (tag) HTML που χρησιμοποιείται για την εμφάνιση μιας πρόσθετης ιστοσελίδας πάνω από την ιστοσελίδα που χρησιμοποιεί ο χρήστης (τιμές 0 False - 1 True)
- **rightClick**: Έλεγχος εάν το δεξί κλικ είναι απενεργοποιημένο στην σελίδα (για αποτροπή εμφάνισης του πηγαίου κώδικα) (τιμές 0 True - 1 False)
- **onmouseover**: Έλεγχος αν το event onmouseover αλλάζει το status bar (για εμφάνιση ψεύτικου URL στο status bar) (τιμές 0 True - 1 False)
- **check_URL_of_anchor**: Τα anchors είναι τα a tags της HTML. Αν τα tags

[2] Το WHOIS είναι ένα πρωτόκολλο ερωτήσεων και απόκρισης που χρησιμοποιείται ευρέως για την αναζήτηση βάσεων δεδομένων που αποθηκεύουν τους εγγεγραμμένους χρήστες ή ανάδοχους ενός πόρου Διαδικτύου
Πηγή: https://en.wikipedia.org/wiki/WHOIS

[3] Το HTTPS χρησιμοποιείται στην πληροφορική για να δηλώσει μία ασφαλή δικτυακή σύνδεση HTTP. Ένας σύνδεσμος που αρχίζει με το πρόθεμα HTTPS υποδηλώνει ότι θα χρησιμοποιηθεί κανονικά το πρωτόκολλο HTTP, αλλά η σύνδεση θα γίνει σε διαφορετική πόρτα και τα δεδομένα θα ανταλλάσσονται κρυπτογραφημένα.
Πηγή: https://el.wikipedia.org/wiki/HTTPS

[4] SSL (Secure Sockets Layer): Διεθνές Standard στον τομέα της Ασφάλειας Δικτύων Υπολογιστών που επιτρέπει την κρυπτογραφημένη επικοινωνία μεταξύ ενός Web Browser και ενός Web Server.

αυτά οδηγούν σε διαφορετικό domain ή δεν οδηγούν σε καμία ιστοσελίδα τότε υπάρχει υποψία για phishing. (real εκφράζει percentage)

- **sfh**: Έλεγχος για το εάν το action ενός form tag (μιας φόρμας) δεν πυροδοτεί καμία δράση ή πυροδοτεί κάποια δράση σε διαφορετικό Domain (τιμές 0 False - 1 True)
- **double_slash**: Ύπαρξη της συμβολοσειράς "//" παραπάνω από 1 φορά μέσα στο URL (τιμές 0 False - 1 True)
- **qty_dot_url**: Πλήθος χαρακτήρα "." σε ολόκληρο το URL (αριθμητική τιμή)
- **qty_hyphen_url**: Πλήθος χαρακτήρα "-" σε ολόκληρο το URL (αριθμητική τιμή)
- **qty_questionmark_url**: Πλήθος χαρακτήρα "?" σε ολόκληρο το URL (αριθμητική τιμή)
- **qty_at_url**: Πλήθος χαρακτήρα "@" σε ολόκληρο το URL (αριθμητική τιμή)
- **qty_hashtag_url**: Πλήθος χαρακτήρα "#" σε ολόκληρο το URL (αριθμητική τιμή)
- **qty_dollar_url**: Πλήθος χαρακτήρα "$" σε ολόκληρο το URL (αριθμητική τιμή)
- **qty_percent_url**: Πλήθος χαρακτήρα "%" σε ολόκληρο το URL (αριθμητική τιμή)
- **TLD_length**: Πλήθος χαρακτήρων TLD[5] (αριθμητική τιμή)
- **TLD_count**: Πλήθος sub-TLDs (αριθμητική τιμή)
- **URL_length**: Πλήθος χαρακτήρων σε ολόκληρο το URL (αριθμητική τιμή)
- **email_in_url**: Εμφάνιση διεύθυνσης ηλεκτρονικού ταχυδρομείου μέσα στο URL (τιμές 0 False - 1 True)
- **word_script_in_url**: Εμφάνιση της λέξης "script" μέσα στο URL (τιμές 0 False - 1 True)
- **check_https_in_url**: Εμφάνιση της λέξης "https" μέσα στο URL (τιμές 0 False - 1 True)
- **qty_dot_domain**: Πλήθος χαρακτήρα "." στο τμήμα του Domain (αριθμητική τιμή)
- **qty_hyphen_domain**: Πλήθος χαρακτήρα "-" στο τμήμα του Domain (αριθμητική τιμή)
- **qty_dollar_domain**: Πλήθος χαρακτήρα "$" στο τμήμα του Domain (αριθμητική τιμή)
- **qty_percent_domain**: Πλήθος χαρακτήρα "%" στο τμήμα του Domain (αριθμητική τιμή)
- **count_vowels**: Πλήθος φωνήεντων στο τμήμα του Domain (αριθμητική τιμή)
- **Domain_length**: Πλήθος χαρακτήρων στο τμήμα του Domain (αριθμητική τιμή)
- **Ip_in_domain**: Εμφάνιση IP στο τμήμα του Domain (τιμές 0 False - 1 True)
- **Client_or_Server_in_domain**: Εμφάνιση της λέξης "client" ή/και "server" στο τμήμα του Domain (τιμές 0 False - 1 True)
- **check_age_of_domain**: Ημέρες από την εγγραφή του Domain στην WHOIS Database (αριθμητική τιμή)
- **days_till_expiration_domain**: Ημέρες μέχρι τη λήξη του SSL Certificate

[5] TLD (Top Level Domain): Ένα top-level domain είναι το τελευταίο κομμάτι ενός ονόματος τομέα στο Διαδίκτυο.
Πηγή: https://el.wikipedia.org/wiki/Top-level_domain

(αριθμητική τιμή)

- **qty_dot_directory**: Πλήθος χαρακτήρα "." στο τμήμα του Directory (αριθμητική τιμή)
- **qty_hyphen_directory**: Πλήθος χαρακτήρα "-" στο τμήμα του Directory (αριθμητική τιμή)
- **qty_questionmark_directory**: Πλήθος χαρακτήρα "?" στο τμήμα του Directory (αριθμητική τιμή)
- **qty_at_directory**: Πλήθος χαρακτήρα "@" στο τμήμα του Directory (αριθμητική τιμή)
- **qty_slash_directory**: Πλήθος χαρακτήρα "/" στο τμήμα του Directory (αριθμητική τιμή)
- **qty_hashtag_directory**: Πλήθος χαρακτήρα "#" στο τμήμα του Directory (αριθμητική τιμή)
- **qty_dollar_directory**: Πλήθος χαρακτήρα "$" στο τμήμα του Directory (αριθμητική τιμή)
- **qty_percent_directory**: Πλήθος χαρακτήρα "%" στο τμήμα του Directory (αριθμητική τιμή)
- **directory_length**: Πλήθος χαρακτήρων στο τμήμα του Directory (αριθμητική τιμή)
- **qty_dot_File**: Πλήθος χαρακτήρα "." στο τμήμα του File (αριθμητική τιμή)
- **qty_hyphen_File**: Πλήθος χαρακτήρα "-" στο τμήμα του File (αριθμητική τιμή)
- **qty_at_File**: Πλήθος χαρακτήρα "@" στο τμήμα του File (αριθμητική τιμή)
- **qty_hashtag_File**: Πλήθος χαρακτήρα "#" στο τμήμα του File (αριθμητική τιμή)
- **qty_dollar_File**: Πλήθος χαρακτήρα "$" στο τμήμα του File (αριθμητική τιμή)
- **qty_percent_File**: Πλήθος χαρακτήρα "%" στο τμήμα του File (αριθμητική τιμή)
- **File_length**: Πλήθος χαρακτήρων στο τμήμα του File (αριθμητική τιμή)
- **qty_dot_params**: Πλήθος χαρακτήρα "." στο τμήμα του Params (αριθμητική τιμή)
- **qty_hyphen_params**: Πλήθος χαρακτήρα "-" στο τμήμα του Params (αριθμητική τιμή)
- **qty_at_params**: Πλήθος χαρακτήρα "@" στο τμήμα του Params (αριθμητική τιμή)
- **qty_underline_params**: Πλήθος χαρακτήρα "_" στο τμήμα του Params (αριθμητική τιμή)
- **qty_hashtag_params**: Πλήθος χαρακτήρα "#" στο τμήμα του Params (αριθμητική τιμή)
- **qty_dollar_params**: Πλήθος χαρακτήρα "$" στο τμήμα του Params (αριθμητική τιμή)
- **qty_percent_params**: Πλήθος χαρακτήρα "%" στο τμήμα του Params (αριθμητική τιμή)
- **params_length**: Πλήθος χαρακτήρων στο τμήμα του Params (αριθμητική τιμή)
- **tld_params**: Έλεγχος εάν στο τμήμα των παραμέτρων υπάρχει κάποιο εκ των TLDs (2 αν υπάρχει στη λίστα των top10 κακόβουλων TLDs όπως αυτές σημειώθηκαν τον Μάρτιο του 2021 [49], 1 αν υπάρχει στη λίστα με τα TLDs, 0 εάν δεν υπάρχει στη λίστα με τα TLDs)
- **count_params**: Πλήθος παραμέτρων που παίρνουν τιμή (αριθμητική τιμή)
- **cookie**: Έλεγχος εάν το HTTP header προσθέτει κάποιο cookie (τιμές 0 False - 1 True)

- **strict_trans_sec**: Έλεγχος για ύπαρξη HTTP header που ένα website ενημερώνει τον browser ότι θα προσπελαστεί αυστηρά με HTTPS Protocol (τιμές 0 False - 1 True)
- **a_tags_count**: Πλήθος a tags στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **form_tags_count**: Πλήθος form tags στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **email_tags_count**: Πλήθος εμφάνισης της λέξης "email" στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **pass_tags_count**: Πλήθος εμφάνισης της λέξης "password" στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **hidden_tags_count**: Πλήθος hidden tags στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **actions_tags_count**: Πλήθος action tags στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **signin_tags_count**: Πλήθος εμφάνισης της λέξης "sign in" στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **signup_tags_count**: Πλήθος εμφάνισης της λέξης "sign up" στον κώδικα HTML της ιστοσελίδας (αριθμητική τιμή)
- **phishing**: ετικέτα (label) για το είδος του URL (0 legitimate – 1 phishing)

Φροντίσαμε να ελέγξουμε για διπλότυπα (duplicates) στα δείγματα, με βάση τις τιμές σε όλα τα features εκτός της στήλης "phishing" που αποτελεί το label. Ωστόσο, δεν εντοπίστηκαν διπλότυπα στο dataset.

3.1.3 Τελική μορφή δεδομένων

Ακολούθησε ανάλυση των τιμών των δειγμάτων, από την οποία προέκυψε ότι 7 χαρακτηριστικά και συγκεκριμένα τα:

- qty_dollar_domain
- qty_percent_domain
- qty_questionmark_directory
- qty_hashtag_directory
- qty_dollar_directory
- qty_hashtag_File
- qty_dollar_File

έπρεπε να αφαιρεθούν, καθώς για καθένα από αυτά όλα σχεδόν τα δείγματα είχαν την ίδια τιμή, επομένως δεν συνέβαλλαν στη διαδικασία της ταξινόμησης.

Κατόπιν, προχωρήσαμε στο διαχωρισμό του dataset σε train dataset (σύνολο εκπαίδευσης) και test dataset (σύνολο ελέγχου). Ο διαχωρισμός αυτός κρίθηκε απαραίτητος, καθώς το train dataset οφείλει να έχει ισοκατανεμημένες τις 2 κλάσεις προκειμένου να διευκολύνει την εκπαίδευση των μοντέλων. Αντίθετα, το test dataset πρέπει να αποτελεί μικρογραφία του πραγματικού κόσμου. Για το λόγο αυτό η κατανομή του test dataset είναι 80% legitimate URLs και 20% phishing URLs. Κατά το σχηματισμό των 2 dataset, φροντίσαμε να ανακατέψουμε (shuffle) τα δείγματα των 2 κλάσεων. Εν τέλει, το train dataset έχει συνολικά 53998 δείγματα (27000 legitimate – 26998 phishing) και το test dataset έχει συνολικά 15005 δείγματα (12004 legitimate – 3001 phishing). Τα 2 dataset είναι σε μορφή αρχείου csv και βρίσκονται διαθέσιμα στο GitHub: https://github.com/souliotispanagiotis/PhishTrap/tree/main/dataset.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	check_ssl,qty_redirects,url_shortened,favicon,dns_record,iFrame,rightClick,onmouseover,check_URL_of_anchor,sfh,double_slash,qty_dot_url,qty_													
2	0,0,1,2,1,1,0,0,0,0,0.6597222222222222,0,0,2,0,0,0,0,0,0,3,1,37,0,0,1,2,0,6,20,0,0,0,113,0,0,0,1,0,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,144,0,1,0,0,1,0,0,0													
3	1,1,1,0,1,0,0,0,0,0,-1.0,0,0,2,0,1,0,0,0,0,0,3,1,78,0,0,1,2,0,4,15,0,0,0,140,0,0,0,2,0,8,0,0,0,0,7,0,0,0,0,0,0,0,38,0,1,0,1,820,1,0,0,0,20,0,0,0													
4	2,2,1,0,1,0,0,0,0,0,0,0,0,0,0,0,4,6,0,0,0,0,0,0,3,1,87,0,0,1,3,5,20,53,0,0,0,29,0,1,0,1,0,8,1,0,0,0,17,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,1,0,0,0,1													
5	3,3,0,0,1,0,0,0,0,0,0,0,0,0,0,0,4,2,1,0,0,0,0,2,1,105,0,0,0,3,1,7,30,0,0,0,-1,0,1,0,1,0,8,1,0,0,0,17,0,0,0,0,0,0,0,41,0,1,0,0,1,0,0,0,1,0,0,0,1													
6	4,4,0,0,0,0,0,0,0,0,0,0,0,0,0,1,8,3,2,0,0,0,0,2,1,162,0,0,1,1,0,4,14,0,0,0,-1,4,0,0,3,0,35,1,0,0,0,11,2,3,0,0,0,0,0,93,1,2,0,0,0,1,0,0,0,0,0,0,1													
7	5,5,1,0,0,0,0,1,0,0,0.075187969924812,0,0,1,2,0,0,0,0,0,3,1,51,0,0,1,1,0,2,9,0,0,5573,33,0,2,0,3,0,33,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,133,2,0,0,2,3,0,0,0													
8	6,6,1,0,1,1,0,1,0,0,0.5360824742268041,0,0,2,0,1,0,0,0,0,2,1,70,0,0,1,2,0,8,22,0,0,0,9,0,0,0,1,0,8,0,0,0,0,7,0,0,0,0,0,0,0,23,0,2,0,0,97,2,0,0,0,0,0,0,0													
9	7,7,1,0,0,0,0,0,0,0,0,0,0,0,0,0,2,4,0,0,0,0,0,2,1,76,0,0,1,1,1,7,17,0,0,0,85,0,0,0,1,0,8,1,3,0,0,42,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1													
10	8,8,1,0,1,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,3,1,22,0,0,1,1,1,3,13,0,0,0,213,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1													

Εικόνα 3.3: Άποψη της μορφής του Dataset

Συνοπτικά, τα βήματα σχηματισμού του dataset περιγράφονται στον παρακάτω αλγόριθμο:

```
First, collect URLs (both legitimate and phishing)

Then, for each URL run in parallel:
        check if the Webpage is active:
        if True:
                Get the Features
                Append row in csv

Delete duplicates from csv
        (duplicates are considered the rows with
        same value in all features except label)

For each feature find the Variance
Delete features with almost-zero Variance (Same value for all Data)

Create the Final csv

From the Final csv create the train csv (50% legit - 50% phishing distribution)
        and the test csv (80% legit - 20% phishing distribution)
        shuffling the data in each one
```

3.2 Επεξεργασία Δεδομένων

Τα δεδομένα, για να μπορούν οι ταξινομητές να εκπαιδευτούν αποδοτικά, πρέπει να περάσουν από κάποια επεξεργασία. Συνηθέστερες μέθοδοι για τον μετασχηματισμό των δεδομένων είναι η κανονικοποίηση (Normalization) ή αλλιώς κλιμάκωση χαρακτηριστικών (Feature Scaling), η επιλογή χαρακτηριστικών (Feature Selection) μειώνοντας έτσι τη διαστατικότητα και η εξισορρόπηση μέσω προσθήκης ή αφαίρεσης ήδη υπαρχόντων δεδομένων για να εξισωθεί ο αριθμός των δειγμάτων σε κάθε κλάση. Λόγω του ότι είχαμε ήδη ισορροπημένες τις δύο κλάσεις στο train dataset μας θα ασχοληθούμε με τις δύο πρώτες μεθόδους.

3.2.1 Κλιμάκωση Χαρακτηριστικών - Feature Scaling

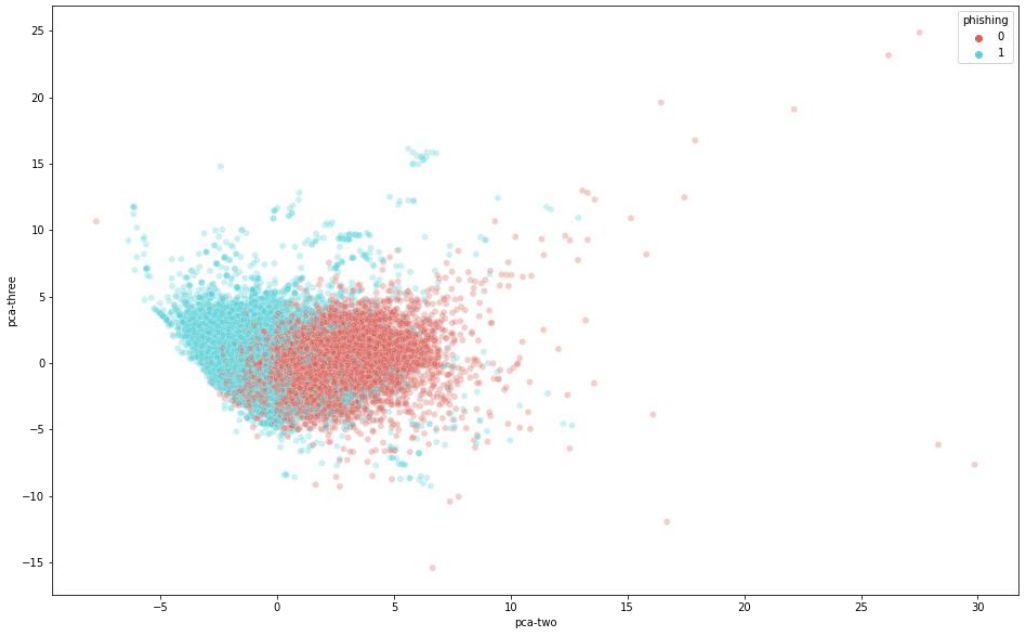
Κλιμακώνοντας τα χαρακτηριστικά επιτυγχάνεται σε πολλές περιπτώσεις καλύτερη επίδοση και ειδικά σε αλγορίθμους που βασίζονται στην απόσταση, όπως ο kNN. Επιπλέον με την χρήση της κλιμάκωσης επιταχύνεται η σύγκλιση σε μεθόδους κατάβασης κλίσης (gradient descent). Τέλος η χρήση της είναι καθοριστική σε περιπτώσεις που χρησιμοποιείται κανονικοποίηση ως μέρος της συνάρτησης σφάλματος.

Πιο συγκεκριμένα θα ασχοληθούμε με τον Standard Scaler, όπου μετατρέπει την κατανομή των χαρακτηριστικών των δεδομένων σε κανονική με μέση τιμή 0 και τυπική απόκλιση 1 με βάση τον παρακάτω τύπο:

$$z = \frac{x - \mu}{\sigma}$$

όπου μ και σ η μέση τιμή και η τυπική απόκλιση αντίστοιχα υπολογισμένες στο σύνολο εκπαίδευσης για κάθε χαρακτηριστικό ξεχωριστά. Για την πρόβλεψη στο σύνολο ελέγχου, τα χαρακτηριστικά των δειγμάτων θα πρέπει να μετατραπούν στην κατανομή των δεδομένων εκπαίδευσης.

Τέλος υπάρχει και η κλιμάκωση μεγίστου (Max Scaler), όπου ουσιαστικά διαιρούμε τα χαρακτηριστικά των δειγμάτων με την μέγιστη τιμή τους. Η μέγιστη τιμή υπολογίζεται στο σύνολο εκπαίδευσης. Πρακτικά κάνουμε αυτήν την κλιμάκωση των χαρακτηριστικών για να επιταχύνουμε τους υπολογισμούς, ειδικά σε βαθιά νευρωνικά δίκτυα.



Εικόνα 3.4: Δεδομένα εκπαίδευσης με χρήση standard scaler και της τεχνικής οπτικοποίησης t-SNE[6]. Τα 63 χαρακτηριστικά των δεδομένων μετατρέπονται σε 2 διαστάσεις με την χρήση PCA, που θα συζητήσουμε αργότερα. Η τεχνική t-SNE χρησιμοποιείται για οπτικοποίηση δεδομένων υψηλών διαστάσεων μετατρέποντας τις ομοιότητες μεταξύ data points σε από κοινού (joint) πιθανότητες και προσπαθεί να ελαχιστοποιήσει την Kullback-Leibler απόκλιση.

3.2.2 Επιλογή Χαρακτηριστικών - Feature Selection

Με αυτή την τεχνική μετράμε την τυπική απόκλιση των χαρακτηριστικών και εάν εμφανίζουν μηδενική ή αμελητέα διακύμανση τότε σημαίνει ότι τα δείγματα έχουν ίδιες ή παρόμοιες τιμές για αυτό το χαρακτηριστικό. Αυτά τα χαρακτηριστικά μπορούν να αφαιρεθούν μιας και δεν συμβάλλουν σημαντικά στην διαδικασία εκπαίδευσης και ταξινόμησης αλλά ταυτόχρονα δυσκολεύουν

[6] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

τον υπολογισμό του ορίου απόφασης και συνεπώς μπορούν με ασφάλεια να απορριφθούν [50].

3.2.3 Μέθοδος Ανάλυσης Κυρίων Συνιστωσών - PCA

Η μέθοδος Ανάλυσης Κυρίων Συνιστωσών (Principal Component Analysis, συντ. PCA) είναι ένας τρόπος μείωσης της διαστατικότητας, κρατώντας όσο πιο δυνατόν μεγαλύτερο ποσοστό της διακύμανσης των δεδομένων.

Πρόκειται για αλγόριθμο μη επιβλεπόμενης μάθησης, που επιλύει το πρόβλημα της κατάρας της διαστατικότητας (curse of dimensionality), όπου όταν αυξάνονται οι διαστάσεις των δεδομένων αυξάνεται σημαντικά ο όγκος του χώρου με αποτέλεσμα τα δεδομένα να γίνονται αραιά (sparse).

Η i-οστή συνιστώσα μπορεί να υπολογιστεί ως η ορθογώνια κατεύθυνση των πρώτων i-1 συνιστωσών που μεγιστοποιούν την διακύμανση των προβαλλόμενων (projected) δεδομένων [51].

3.3 Εκπαίδευση Μοντέλων – Εύρεση Βέλτιστων Υπερπαραμέτρων

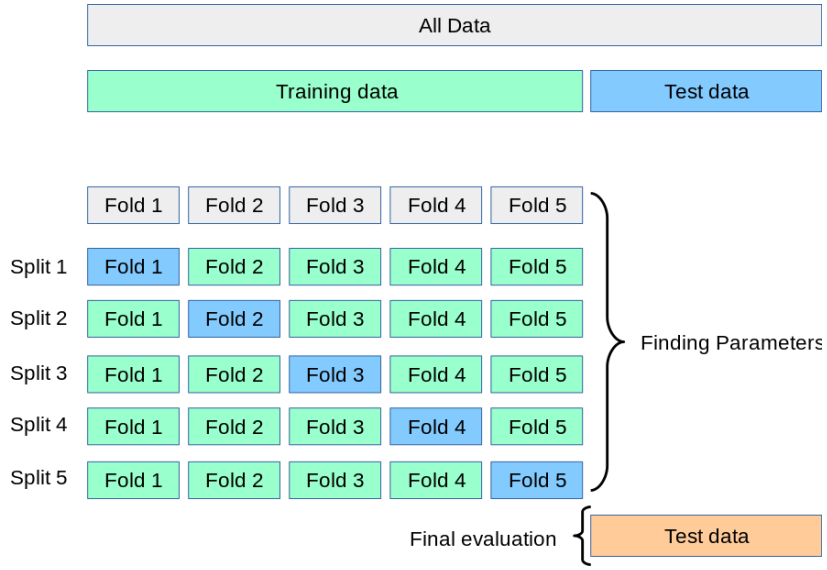
Για τη δημιουργία και την εκπαίδευση των μοντέλων μηχανικής μάθησης χρησιμοποιήθηκαν οι βιβλιοθήκες scikit learn[7] και Keras Framework[8] της Google.

3.3.1 GridSearchCV

Για την εύρεση βέλτιστων παραμέτρων χρησιμοποιήθηκε η τεχνική GridSearchCV. Πρόκειται για εξαντλητική αναζήτηση στις καθορισμένες τιμές για έναν εκτιμητή. Ωστόσο αν χρησιμοποιήσουμε μόνο το σύνολο δεδομένων στην εκπαίδευση τότε είναι πολύ πιθανό να γίνει overfitting σε αυτά , δηλαδή οι παράμετροι να προσαρμοστούν πολύ στα δεδομένα εισόδου και να είναι αδύνατον να συμπεράνουν τα δεδομένα ελέγχου. Για να λυθεί αυτό το πρόβλημα, παρουσιάζονται τα δεδομένα επικύρωσης (validation split) τα οποία αποτελούν ένα ποσοστό των δεδομένων εκπαίδευσης που χρησιμοποιούνται για την αξιολόγηση του μοντέλου κατά την εκπαίδευση.

Έτσι χρησιμοποιείται και η τεχνική Cross Validation με 4-Folds, όπου είναι μία διαδικασία επαναλαμβανόμενης δειγματοληψίας από το σύνολο δεδομένων για την αξιολόγηση μοντέλων. Το σύνολο δεδομένων σπάει σε 4 groups όπου τα 3 εξ αυτών χρησιμοποιούνται για την μάθηση του εκτιμητή ενώ το τελευταίο για την αξιολόγηση και τον υπολογισμό μετρικών. Συνεπώς θεωρούμε κάθε φορά ένα validation split 25%. Αυτή η διαδικασία επαναλαμβάνεται κυκλικά μέχρι και τα 4 groups να χρησιμοποιηθούν ως μέρος αξιολόγησης.

[7] https://scikit-learn.org/stable/index.html
[8] https://keras.io/



Εικόνα 3.5: Αναπαράσταση της διαδικασίας Cross Validation, όπου αρχικά χωρίζουμε το σύνολο των δεδομένων σε δεδομένα εκπαίδευσης και δεδομένα ελέγχου. Τα δεδομένα εκπαίδευσης χωρίζονται σε 5 folds όπου τα 4 από αυτά χρησιμοποιούνται στην εκπαίδευση και το 5ο στην αξιολόγηση του μοντέλου. Αυτή η διαδικασία αξιολόγησης επαναλαμβάνεται 5 φορές μέχρι να χρησιμοποιηθούν όλα τα folds σαν δεδομένα αξιολόγησης. Τέλος αφού βρεθούν οι παράμετροι του βέλτιστου ως προς κάποια μετρική μοντέλου, το αξιολογούμε στο σύνολο ελέγχου. (Πηγή: https://scikit-learn.org/stable/modules/cross_validation.html)

3.3.2 Επιλογή Μετρικής

Η μετρική που επιλέξαμε να παρακολουθούμε κυρίως στα πειράματα μας είναι η FNR, την οποία προσπαθήσαμε να ελαχιστοποιήσουμε. Την προτιμήσαμε έναντι του συμπληρωματικού του Recall γιατί θέλουμε να τονίσουμε το γεγονός ότι προσπαθούμε να ελαχιστοποιήσουμε τα Phishing URLs που ταξινομούνται λανθασμένα ως legitimate. Φυσικά μεριμνάμε να ελέγχουμε ότι οι μετρικές Accuracy και Precision παραμένουν σε ικανοποιητικά υψηλό επίπεδο.

3.3.3 Μοντέλο KNN

Για το μοντέλο του KNN[9] χρησιμοποιήσαμε τις τεχνικές επεξεργασίας δεδομένων Feature Scaling (Standard Scaler), Feature Selection και PCA. Συγκεκριμένα, οι υπερπαράμετροι που χρησιμοποιήθηκαν φαίνονται στον πίνακα 3.1.

	Υπερ-Παράμετροι	**Περιγραφή**	**Τιμές**
Variance Threshold	Threshold	Κατώφλι	[0, 0.05, 0.1, 1]

[9] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

PCA	n_components	Πλήθος συνιστωσών	[25, 29, 33, 37, 45, 49, 53, 57, 59, 61, 63]
	svd_solver	Επιλυτής	['full', 'arpack', 'randomized']
	tol	Ανεκτικότητα	[1e-07, 1e-05, 'auto']
	whiten	Μετασχηματίζει το διάνυσμα εισόδου σε διάνυσμα λευκού θορύβου	[True, False]

Πίνακας 3.1: Υπερπαράμετροι Variance Threshold και PCA για το μοντέλο KNN

Όσον αφορά τις υπερπαραμέτρους του μοντέλου KNN δοκιμάσαμε αυτές που φαίνονται στον πίνακα 3.2.

Υπερπαράμετροι	**Περιγραφή**	**Τιμές**
n_neighbors	Πλήθος γειτόνων	[1, 5, 9, 13, 17, 21, 25, 29, 33, 37, 41, 45, 49, 53, 57]
algorithm	Αλγόριθμος υπολογισμού κοντινότερου γείτονα	['ball_tree', 'kd_tree', 'brute']
leaf_size	Πλήθος φύλλων Χρησιμοποιείται μόνο στους αλγορίθμους ball_tree και kd_tree	[5, 10, 30, 50]
metric	Μετρική απόστασης	['minkowski', 'euclidean', 'manhattan']
weights	Συνάρτηση βαρών που χρησιμοποιείται στην πρόβλεψη	['uniform', 'distance']
p	Παράμετρος ύψωσης σε δύναμη για την μετρική Minkowski	[1, 2, 3, 6, 9, 20, 40]

Πίνακας 3.2: Υπερπαράμετροι KNN

Από την εφαρμογή του GridsearchCV για cv=4 και με γνώμονα τη βελτίωση της μετρικής Recall, πήραμε ως βέλτιστες υπερπαραμέτρους τις τιμές του πίνακα 3.3.

Υπερπαράμετροι	**Τιμές**
selector_threshold	0
pca_n_components	57
pca_whiten	True
pca_svd_solver	Randomized
pca_tol	1e-07

n_neighbors	5
algorithm	kd_tree
leaf_size	30
metric	manhattan
weights	distance
p	6

Πίνακας 3.3: Βέλτιστες Υπερπαράμετροι για το μοντέλο του KNN

Με τις τιμές αυτές το μοντέλο για το σύνολο ελέγχου δίνει τα αποτελέσματα που φαίνονται στον πίνακα 3.4.

Πίνακας Σύγχυσης		
	Predicted Class	
Actual Class	10812	1192
	187	2814
Μετρικές		
Accuracy	90.81 %	
Precision	70.24 %	
FNR	6.23 %	

Πίνακας 3.4: Αποτελέσματα μοντέλου KNN στο σύνολο ελέγχου

3.3.4 Μοντέλο MLP

Για το μοντέλο του MLP[10] χρησιμοποιήσαμε Standard Scaler και την τεχνική επεξεργασίας δεδομένων Feature Selection. Συγκεκριμένα, οι υπερπαράμετροι που χρησιμοποιήθηκαν φαίνονται στους πίνακες 3.5.

	Υπερ-Παράμετροι	**Περιγραφή**	**Τιμές**
Variance Threshold	Threshold	Κατώφλι	[0, 0.1]

Πίνακας 3.5: Υπερπαράμετροι Variance Threshold για το μοντέλο MLP

Όσον αφορά τις υπερπαραμέτρους του μοντέλου MLP δοκιμάσαμε αυτές που φαίνονται στον πίνακα 3.6.

[10] https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html

Υπερπαράμετροι	Περιγραφή	Τιμές
hidden_layer_sizes	Πλήθος νευρώνων στο κρυφό επίπεδο	[2000, 2500, 3000]
activation	Συνάρτηση ενεργοποίησης στο κρυφό επίπεδο	['relu', 'tanh', 'sigmoid']
alpha	Παράμετρος ποινής L2	[0.001, 0.01, 0.1]
learning_rate	Τρόπος μεταβολής ρυθμού μάθησης για ενημέρωση βαρών	['adaptive', 'constant']
learning_rate_init	Ρυθμός Μάθησης	[0.1, 1, 10]
solver	Επιλυτής για βελτιστοποίηση βαρών	['adam','sgd']
momentum	Ορμή για ενημέρωση της κατάβασης κλίσης – Χρησιμοποιείται μόνο στον sgd solver	[0.5, 0.8, 0.9, 0.99]

Πίνακας 3.6: Υπερπαράμετροι MLP

Από την εφαρμογή του GridsearchCV για cv=4 και με γνώμονα τη βελτίωση της μετρικής Recall, πήραμε ως βέλτιστες υπερπαραμέτρους τις τιμές του πίνακα 3.7.

Υπερπαράμετροι	Τιμές
selector_threshold	0
hidden_layer_sizes	2500
activation	relu
alpha	0.01
learning_rate	adaptive
learning_rate_init	0.1
solver	sgd
momentum	0.8

Πίνακας 3.7: Βέλτιστες Υπερπαράμετροι για το μοντέλο του MLP

Με τις τιμές αυτές το μοντέλο για το σύνολο ελέγχου δίνει τα αποτελέσματα που φαίνονται στον πίνακα 3.8.

Πίνακας Σύγχυσης		
	Predicted Class	
Actual Class	11212	792
	140	2861

Μετρικές	
Accuracy	93.79 %
Precision	78.32 %
FNR	4.67 %

Πίνακας 3.8: Αποτελέσματα μοντέλου MLP στο σύνολο ελέγχου

3.3.5 Μοντέλο Random Forest

Όσον αφορά τις υπερπαραμέτρους του μοντέλου Random Forest[11] δοκιμάσαμε αυτές που φαίνονται στον πίνακα 3.9, χρησιμοποιώντας Standard Scaler.

Υπερπαράμετροι	Περιγραφή	Τιμές
n_estimators	Πλήθος δέντρων στο δάσος	[300, 400, 450, 500, 550, 600, 700, 800]
criterion	Συνάρτηση υπολογισμού της ποιότητας του κλαδέματος	['gini', 'entropy']
bootstrap	Όταν είναι ενεργοποιημένο παίρνουμε δείγματα από το αρχικό σύνολο εκπαίδευσης μέσω αντικατάστασης ώστε να παραχθούν πολλαπλά σύνολα εκπαίδευσης	[True, False]
max_depth	Μέγιστο βάθος μεμονωμένων εκτιμητών	[100, 200, 300, 400, 500]
max_features	Το πλήθος των χαρακτηριστικών που λαμβάνονται υπόψη για την εύρεση του καλύτερου κλαδέματος	[0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95, 1.0]
min_samples_leaf	Ελάχιστος αριθμός δειγμάτων που πρέπει να βρίσκονται σε κόμβο φύλλων	[1, 2, 3, 4, 5]
min_samples_split	Ελάχιστος απαιτούμενος αριθμός δειγμάτων για το διαχωρισμό ενός εσωτερικού κόμβου	[1, 2, 3, 4, 5]
oob_score	Όταν είναι ενεργοποιημένο χρησιμοποιούνται δείγματα out-of-bag στην εκτίμηση του σκορ γενίκευσης - Χρησιμοποιείται μόνο όταν το bootstrap είναι ενεργοποιημένο.	[True, False]

[11] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

warm_start	Όταν είναι ενεργοποιημένο επαναχρησιμοποιεί την προηγούμενη λύση	[True, False]

Πίνακας 3.9: Υπερπαράμετροι Random Forest.

Σημειώνεται ότι δεν χρησιμοποιήθηκε καμία τεχνική επεξεργασίας δεδομένων καθότι ο αλγόριθμος διαθέτει ενσωματωμένη την τεχνική max_features.

Από την εφαρμογή του GridsearchCV για cv=5 και με γνώμονα τη βελτίωση της μετρικής Recall, πήραμε ως βέλτιστες υπερπαραμέτρους τις τιμές του πίνακα 3.10.

Υπερπαράμετροι	**Τιμές**
n_estimators	450
criterion	Gini
bootstrap	False
max_depth	300
max_features	0.15
min_samples_leaf	1
min_samples_split	2
oob_score	False
warm_start	False

Πίνακας 3.10: Βέλτιστες Υπερπαράμετροι για το μοντέλο του Random Forest

Με τις τιμές αυτές το μοντέλο για το σύνολο ελέγχου δίνει τα αποτελέσματα που φαίνονται στον πίνακα 3.11.

Πίνακας Σύγχυσης		
	Predicted Class	
Actual Class	11378	626
	140	2861
Μετρικές		
Accuracy	94.90 %	
Precision	82.05 %	
FNR	4.67 %	

Πίνακας 3.11: Αποτελέσματα μοντέλου Random Forest στο σύνολο ελέγχου

3.3.6 Μοντέλο Gradient Boosting

Όσον αφορά τις υπερπαραμέτρους του μοντέλου Gradient Boosting[12] δοκιμάσαμε αυτές που φαίνονται στον πίνακα 3.12, χρησιμοποιώντας Standard Scaler.

Υπερπαράμετροι	Περιγραφή	Τιμές
n_estimators	Πλήθος εκτιμητών	[500, 1000, 3000, 5000, 7500, 10000, 15000, 17000]
criterion	Συνάρτηση υπολογισμού της ποιότητας του κλαδέματος	['friedman_mse', 'mse']
learning_rate	Ρυθμός συρρίκνωσης	[0.001, 0.005, 0.01, 0.05, 0.1]
loss	Συνάρτηση σφάλματος προς βελτιστοποίηση	['deviance', 'exponential']
max_depth	Μέγιστο βάθος μεμονωμένων εκτιμητών	[40, 50, 60, 70, 80]
max_features	Το πλήθος των χαρακτηριστικών που λαμβάνονται υπόψιν για την εύρεση του καλύτερου κλαδέματος	[10, 20, 30, 40, 50, 60, 63]
min_samples_leaf	Ελάχιστος αριθμός δειγμάτων που πρέπει να βρίσκονται σε κόμβο φύλλων	[1, 2, 3, 4, 5]
min_samples_split	Ελάχιστος απαιτούμενος αριθμός δειγμάτων για το διαχωρισμό ενός εσωτερικού κόμβου	[100, 150, 200, 250, 300, 350, 400]
subsample	Το κλάσμα των δειγμάτων που χρησιμοποιούνται για την εκπαίδευση μεμονωμένων εκπαιδευόμενων	[0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
warm_start	Όταν είναι ενεργοποιημένο επαναχρησιμοποιεί την προηγούμενη λύση	[True, False]

Πίνακας 3.12: Υπερπαράμετροι Gradient Boosting

Σημειώνεται ότι δεν χρησιμοποιήθηκε καμία τεχνική επεξεργασίας δεδομένων καθότι ο αλγόριθμος διαθέτει ενσωματωμένη την τεχνική max_features.

Από την εφαρμογή του GridsearchCV για cv=5 και με γνώμονα τη βελτίωση της μετρικής Recall, πήραμε ως βέλτιστες υπερπαραμέτρους τις τιμές του πίνακα 3.13.

[12] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

Υπερπαράμετροι	Τιμές
n_estimators	15000
criterion	mse
learning_rate	0.005
loss	deviance
max_depth	60
max_features	60
min_samples_leaf	3
min_samples_split	250
subsample	0.8
warm_start	True

Πίνακας 3.13: Βέλτιστες Υπερπαράμετροι για το μοντέλο του Gradient Boosting

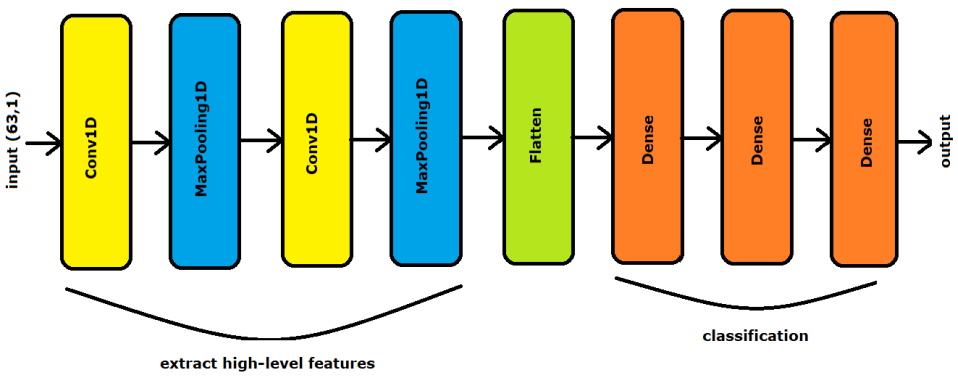
Με τις τιμές αυτές το μοντέλο για το σύνολο ελέγχου δίνει τα αποτελέσματα που φαίνονται στον πίνακα 3.14.

Πίνακας Σύγχυσης		
	Predicted Class	
Actual Class	11481	523
	112	2889
Μετρικές		
Accuracy	95.77 %	
Precision	84.67 %	
FNR	3.73 %	

Πίνακας 3.14: Αποτελέσματα μοντέλου Gradient Boosting στο σύνολο ελέγχου

3.3.7 Μοντέλο CNN

Με βάση το paper των Yerima και Alzaylaee [52], ξεκινήσαμε με την αρχιτεκτονική που φαίνεται στην εικόνα 3.6 δοκιμάζοντας περισσότερες τιμές για τις υπερπαραμέτρους, εφαρμόζοντας και Max Scaler. Λόγω του ότι δεν έχουμε εικόνες αλλά features θα χρησιμοποιήσουμε μονοδιάστατα στρώματα.



Εικόνα 3.6: Πρώιμη αρχιτεκτονική του CNN μοντέλου

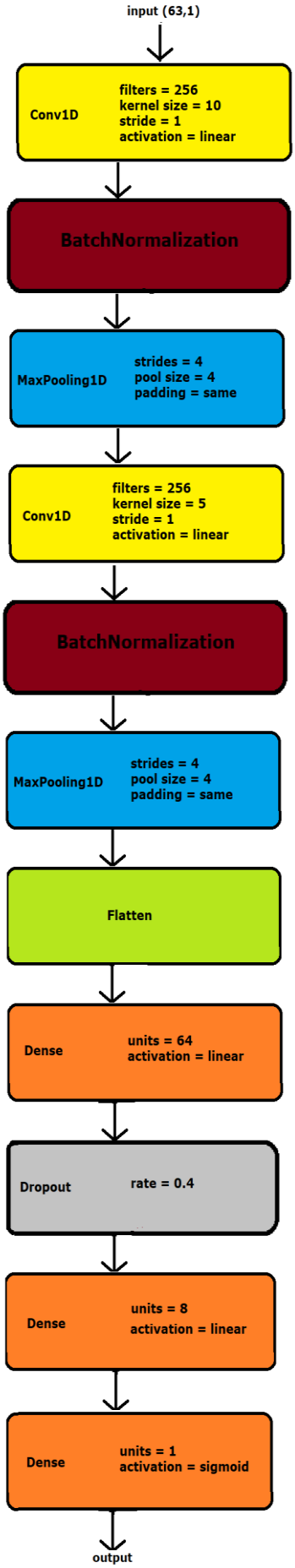
Οι υπεραπαράμετροι που λάβαμε υπόψη στην παραπάνω αρχιτεκτονική[13] παρουσιάζονται στον πίνακα 3.15.

	Υπερπαράμετροι	Περιγραφή	Τιμές
Conv1D	filters	Αριθμός φίλτρων	[8,16,32,64,128,256,512]
	kernel_size	Μέγεθος φίλτρων	[5,10]
	stride	Δρασκελιά	[1,2]
	activation_function	Συνάρτηση Ενεργοποίησης	['relu', 'sigmoid', 'tanh', 'linear', 'elu']
MaxPooling1D	stride	Δρασκελιά	[2,4]
	pool_size	Μέγεθος στρώματος	[2,4]
	padding	Παραγέμισμα	['same', 'valid']
Dense	units	Πλήθος νευρώνων	[8,32,64]
	activation_function	Συνάρτηση Ενεργοποίησης	['linear', 'relu', 'tanh']
Παράμετροι εκπαίδευσης	batch_size	Μέγεθος δέσμης	[8,16,32,64,128,256]
	epochs	Εποχές εκπαίδευσης	[50,100,125,150]
	optimizer	Βελτιστοποιητής	['adam', 'sgd']
	learning_rate_optimizer	Ρυθμός Μάθησης του Βελτιστοποιητή	[1e-4, 1e-3, 1e-2]

Πίνακας 3.15: Υπερπαράμετροι και layers για το πρωταρχικό μοντέλο CNN

Κατόπιν, δοκιμάσαμε να προσθέσουμε κι άλλα layers. Συγκεκριμένα, προσθέσαμε Batch Normalization Layer μεταξύ των Conv1D και MaxPooling1D layers και Dropout layer μεταξύ των δύο πρώτων Dense (Πυκνών) layers με τιμές για την παράμετρο rate (ποσοστό) τις εξής: 0.2, 0.4, 0.5, 0.7.

[13] https://keras.io/api/layers/



Χρησιμοποιήσαμε επίσης τα callbacks για μείωση του learning rate αν δεν υπάρχει βελτίωση για 5 εποχές (factor = 0.5 και min_lr = 1e-6) και early_stopping όπου αν για 10 εποχές δεν έχουμε αύξηση της μετρικής σταματάει η εκπαίδευση και επιστρέφονται τα καλύτερα βάρη.

Ως συνάρτηση μέτρησης σφάλματος χρησιμοποιήθηκε η δυαδική διασταυρούμενη εντροπία (binary crossentropy). Επίσης δημιουργήσαμε μία custom μετρική όπου υπολογίζει το άθροισμα του accuracy με το recall πολλαπλασιασμένο με 3. Δηλαδή:

$$metric = accuracy + 3 * recall$$

Αυτή η επιλογή έγινε ώστε να δοθεί περισσότερη σημασία στη μετρική Recall χωρίς ωστόσο να παραγκωνίζουμε την μετρική Accuracy.

Δοκιμάζοντας τις τιμές των παραπάνω υπερπαραμέτρων καθώς και αν θα υπάρχουν ή όχι τα στρώματα Batch Normalization και Dropout καταλήξαμε στην βέλτιστη διάταξη που φαίνεται στην εικόνα 3.7.

Εν τέλει, χρησιμοποιήσαμε τα στρώματα Batch Normalization και Dropout μιας και συμβάλουν θετικά στην καταπολέμηση των προβλημάτων internal covariance shift και overfitting αντίστοιχα.

Πιο αναλυτικά για το pooling layer, επιλέχθηκε για την υπερπαράμετρο padding η τιμή 'same' δηλαδή η διατήρηση του πλήθους των χαρακτηριστικών, για το pool size η τιμή 4 δηλαδή γίνεται επιλογή της μέγιστης τιμής τεσσάρων χαρακτηριστικών κάθε φορά και για το stride η τιμή 4 δηλαδή γίνεται μετακίνηση του max pooling φίλτρου κατά 4 θέσεις (pixel) δεξιά.

Παράλληλα, για το στρώμα της συνέλιξης επιλέχθηκαν να ισούνται τα kernels με 256, μεγέθους 10 και 5 αντίστοιχα, το stride να έχει την τιμή 1 και η συνάρτηση ενεργοποίησης να είναι η γραμμική, δηλαδή να περνάει το αποτέλεσμα της συνέλιξης ως έχει στο επόμενο επίπεδο. Η επιλογή αυτής της συνάρτησης ενεργοποίησης μπορεί να οφείλεται στο γεγονός ότι δεν περιορίζει το πεδίο τιμών, όπως π.χ. κάνουν οι ReLU, sigmoid, tanh με πεδία τιμών [0, +∞), (0,1), (-1,1) αντίστοιχα, ωστόσο λόγω σταθερής παραγώγου δεν βοηθάει στο backpropagation.

Εικόνα 3.7: Βέλτιστη Αρχιτεκτονική του CNN μοντέλου με αναγραφή των βέλτιστων υπερπαραμέτρων κάθε στρώματος

Τέλος, όσον αφορά το Πλήρως Συνδεδεμένο κομμάτι του δικτύου, επιλέχθηκαν 64 και 8 units για τα πρώτα 2 Dense Layers αντίστοιχα με συνάρτηση ενεργοποίησης πάλι την γραμμική. Εδώ χρησιμοποιείται και το Dropout layer όπου συρρικνώνει το πρόβλημα του overfitting. Στο τελευταίο Dense στρώμα γίνεται η ταξινόμηση στις 2 κλάσεις και εάν αν το αποτέλεσμα είναι μικρότερο του 0.5 τότε το δεδομένο ταξινομείται στην πρώτη κλάση αλλιώς ταξινομείται στη δεύτερη.

Με τις τιμές αυτές το μοντέλο για το σύνολο ελέγχου δίνει τα αποτελέσματα που φαίνονται στον πίνακα 3.16.

Πίνακας Σύγχυσης		
	Predicted Class	
Actual Class	10342	1662
	78	2923
Μετρικές		
Accuracy	88.40 %	
Precision	63.75 %	
FNR	2.60 %	

Πίνακας 3.16: Αποτελέσματα μοντέλου CNN στο σύνολο ελέγχου

3.3.8 Άλλα Μοντέλα

Στην ενότητα αυτή θα αναφερθούμε σε 5 μοντέλα για τα οποία αν και προσπαθήσαμε αρκετά για διάφορους λόγους δεν έφεραν τα επιθυμητά αποτελέσματα μη καταφέρνοντας να εξασφαλίσουν FNR κάτω του 7% που είχε τεθεί ως αρχικό threshold. Σημειώνεται ότι στον SVM χρησιμοποιήθηκε Standard Scaler, ενώ στα υπόλοιπα μοντέλα βαθιάς μάθησης Max Scaler.

Μηχανές Διανυσμάτων Υποστήριξης

Παρότι, έγινε δοκιμή όλων των kernels και όλων των παραλλαγών του αλγορίθμου (SVC, NuSVC, LinearSVC) που προσφέρει η βιβλιοθήκη[14], δεν φάνηκε ικανός να ταξινομήσει αποτελεσματικά τα δεδομένα αποτυγχάνοντας να ξεπεράσει το όριο που θέσαμε για την μετρική FNR. Το γεγονός αυτό, σύμφωνα με την εικόνα 3.4, ίσως οφείλεται στο ότι τα δεδομένα στον χώρο 2 διαστάσεων δεν φαίνεται να είναι γραμμικά διαχωρίσιμα και συνεπώς φαίνεται απαραίτητη η χρήση κάποιου kernel, κάτι που καθιστά ακόμα πιο χρονοβόρα την εκπαίδευση.

Σε αυτό συμβάλλει σημαντικά και το μεγάλο πλήθος των δειγμάτων εκπαίδευσης (~54000 δείγματα), καθώς όπως αναφέρει το εγχειρίδιο του scikit-learn όσα περισσότερα τα δείγματα εκπαίδευσης τόσο περισσότερο αυξάνεται ο χρόνος εκπαίδευσης και μάλιστα τετραπλασιάζεται. Το αποτέλεσμα ήταν να μην μπορέσουμε να χρησιμοποιήσουμε όλες τις υπερπαραμέτρους και πιθανώς να αποτύχαμε και γι' αυτόν τον λόγο.

[14] https://scikit-learn.org/stable/modules/classes.html#module-sklearn.svm

LSTM

Πρόκειται για ένα στρώμα επαναλαμβανόμενων νευρωνικών δικτύων κατάλληλο για χρήση σε ταξινόμηση, επεξεργασία και προβλέψεις βασισμένα σε δεδομένα χρονοσειρών.

Με βάση το paper των Chen W., Zhang W., & Su Y., με τίτλο "Phishing Detection Research Based on LSTM" [53], δοκιμάσαμε την προτεινόμενη δομή διευρύνοντας μάλιστα το πλήθος των τιμών των υπερπαραμέτρων. Στον πίνακα 3.17 φαίνονται οι τιμές που δοκιμάσαμε καθώς και το πλήθος των στρωμάτων LSTM.

	Υπερπαράμετροι	**Περιγραφή**	**Τιμές**
LSTM	units	Αριθμός μονάδων	[16, 64, 128, 256]
	number of LSTM layers	Πλήθος στρωμάτων	[1, 3, 5]
	dropout	Ποσοστό μονάδων που θα διαγραφούν κατά τον μετασχηματισμό των εισόδων	[0.2, 0.4, 0.7]
Dense	units	Πλήθος νευρώνων	[8, 64]
	activation_function	Συνάρτηση Ενεργοποίησης	['linear', 'relu', 'tanh']
Παράμετροι εκπαίδευσης	batch_size	Μέγεθος δέσμης	[8,32,128,256]
	epochs	Εποχές εκπαίδευσης	[50,100]
	optimizer	Βελτιστοποιητής	['adam', 'sgd']
	learning_rate_optimizer	Ρυθμός Μάθησης του Βελτιστοποιητή	[1e-4, 1e-3, 1e-2]

Πίνακας 3.17: Υπερπαράμετροι και layers για το μοντέλο LSTM

Τα αποτελέσματα των παραπάνω δοκιμών δεν ήταν ικανοποιητικά ενώ ταυτόχρονα υπήρχαν ταλαντώσεις για την μετρική Recall μεταξύ των τιμών 0 και 1 και συνεπώς δεν μάθαινε να διαχωρίζει τα phishing από τα legit δεδομένα.

Άλλος πιθανός λόγος αποτυχίας είναι ότι τα δεδομένα μας δεν έχουν μορφή χρονοσειρών με αποτέλεσμα να αποτυγχάνει η χρήση απλών LSTM στρωμάτων.

Δοκιμάσαμε, τέλος, την χρήση bidirectional LSTMs στρωμάτων ώστε να ληφθεί υπόψη περιεχόμενο (context) τόσο από το παρελθόν όσο και από το μέλλον, χωρίς ωστόσο αποτέλεσμα καθώς παρουσιαζόταν παρόμοια συμπεριφορά με το απλό LSTM μοντέλο.

Ind-RNN (Independently Recurrent Neural Network)

Οι νευρώνες σε ένα στρώμα RNN είναι πεπλεγμένοι μεταξύ τους με αποτέλεσμα η συμπεριφορά τους να είναι δύσκολο να ερμηνευθεί. Για την αντιμετώπιση αυτού του προβλήματος προτάθηκε το paper των Shuai Li,

Wanqing Li, Chris Cook, Ce Zhu & Yanbo Gao με τίτλο "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN" [54], όπου οι νευρώνες στο ίδιο στρώμα RNN είναι ανεξάρτητοι μεταξύ τους και συνδέονται μεταξύ στρωμάτων. Έτσι επιτρέπεται να μάθει μακροπρόθεσμες εξαρτήσεις καταπολεμώντας προβλήματα των RNNs.

Δοκιμάσαμε 1 μόνο layer για το παραπάνω στρώμα ακολουθούμενο από classification head. Η μόνη υπερπαράμετρος ήταν ο αριθμός των units. Παρατηρήσαμε ότι οι ταλαντώσεις εξαλείφθηκαν, ωστόσο το FNR ήταν της τάξης του 10% με αποτέλεσμα να απορριφθεί.

Θα μπορούσαμε ίσως να δοκιμάσουμε την προσθήκη και άλλων στρωμάτων ή ακόμα και την χρήση του σε ένα βαθύ CNN δίκτυο.

CNN-LSTM

Σε αυτήν την ενότητα ασχοληθήκαμε με την χρήση της βέλτιστης CNN αρχιτεκτονικής που προτάθηκε παραπάνω και την προσθήκη σε αυτής στρωμάτων επαναληπτικών νευρωνικών δικτύων. Δοκιμάσαμε λοιπόν να προσθέσουμε πριν το classification head ένα στρώμα LSTM με μοναδική υπερπαράμετρο τον αριθμό των μονάδων (units) για να δούμε πώς θα μεταβληθούν οι μετρικές στο σύνολο επικύρωσης. Το ότι είχαμε ως βάση το βέλτιστο CNN βοήθησε στη σταθεροποίηση του Recall και την αποφυγή των ταλαντώσεων που είχαμε για τα vanilla επαναληπτικά δίκτυα.

	Υπερπαράμετροι	**Περιγραφή**	**Τιμές**
LSTM	units	Αριθμός μονάδων	[16, 64, 128, 256]
	number of LSTM layers	Πλήθος στρωμάτων	[1]

Πίνακας 3.18: Επιπλέον Υπερπαράμετροι και layers για το μοντέλο CNN-LSTM

Τα αποτελέσματα και σε αυτή την περίπτωση ωστόσο δεν ήταν τα αναμενόμενα. Αυτό πιθανώς να οφείλεται στο γεγονός ότι το LSTM δεν ενδείκνυται για το είδος δεδομένων που χρησιμοποιήσαμε και ταυτόχρονα το πλήθος των δεδομένων να μην είναι ικανοποιητικό για να ολοκληρωθεί επιτυχώς η εκπαίδευση μιας τέτοιας βαθιάς αρχιτεκτονικής.

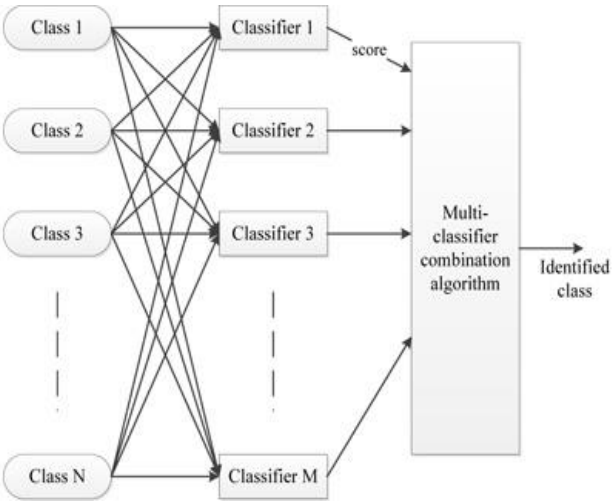
Transformers

Οι transformers χρησιμοποιούνται κυρίως για επεξεργασία φυσικής γλώσσας. Αυτός είναι ένας από τους πιθανούς λόγους που απέτυχαν στο δικό μας πρόβλημα. Πιθανότατα και ο όγκος των δεδομένων μας να είναι αρκετά μικρός για να χρησιμοποιηθεί σε ένα βαθύ δίκτυο όπως αυτό των transformers. Η τεχνική αυτή απορρίφθηκε λόγω του ότι υπερέβαινε κατά πολύ το όριο του μεγίστου FNR που είχαμε θέσει.

Εν τέλει, ίσως απαιτείται περαιτέρω έρευνα, εξονυχιστική διερεύνηση των υπερπαραμέτρων και προσαρμογές για την ένταξη του συγκεκριμένου αλγορίθμου στην επίλυση του προβλήματος, η οποία ίσως προέλθει ακόμη και σε συνδυασμό με κάποιο άλλο μοντέλο βαθιάς μηχανικής μάθησης.

3.4 Επιλογή Voting Scheme

Η εφαρμογή πέρα από την πρόβλεψη των μοντέλων, προσπαθεί να κάνει και μια τελική πρόβλεψη για το δοθέν URL, λαμβάνοντας υπόψη τις πιθανότητες που δίνουν όλοι οι classifiers για τις 2 κλάσεις. Σε σημαντικές εφαρμογές που αφορούν ιατρικές διαγνώσεις ή κυβερνοασφάλεια, το κόστος λανθασμένης πρόβλεψης μπορεί να είναι μεγάλο. Συνεπώς ο συνδυασμός των προτάσεων πολλών classifiers μπορεί να δώσει μια πιο αξιόπιστη και σφαιρική πρόταση από έναν μόνο ταξινομητή [55].



Εικόνα 3.8: Συνδυασμός προβλέψεων ταξινομητών (Πηγή: [55])

Οι combiners που θα δούμε δεν απαιτούν κάποια ρύθμιση παραμέτρων μετά την εκπαίδευση των επιμέρους ταξινομητών και δέχονται ως είσοδο μόνο τις πιθανότητες για κατανομή στις 2 κλάσεις του ταξινομητή.

Μια ερώτηση που αναδύεται είναι ποια μέθοδος συνδυασμού πρέπει να χρησιμοποιηθεί. Σύμφωνα με τους Kuncheva L. I. & Rodríguez J. J. [56] η χρήση της «βεβαρυμένης ψήφου πλειοψηφίας» (weighted majority vote), έχει τα καλύτερα αποτελέσματα για προβλήματα με μικρό αριθμό μη ισορροπημένων κλάσεων, όπως το δικό μας.

Προετοιμασία πιθανοτήτων

Αρχικά θεωρούμε σύνολο κλάσεων $\Omega = \{\omega_1, \dots, \omega_c\}$ και ένα σύνολο από L ταξινομητές. Συμβολίζουμε με s_i την πρόταση που έκανε ο ταξινομητής i ($s_i \in \Omega$) . Συνεπώς μας ενδιαφέρει η πιθανότητα:

$$P(\, \omega_\kappa \, \nu\alpha \, \epsilon\acute{\iota}\nu\alpha\iota \, \eta \, \sigma\omega\sigma\tau\acute{\eta} \, \kappa\lambda\acute{\alpha}\sigma\eta \mid s_1, \dots, s_L \,) \; \gamma\iota\alpha \; k = 1, \dots, c$$

και για συντομία $P(\omega_k|\boldsymbol{s})$. Θεωρώντας ότι οι ταξινομητές παίρνουν αποφάσεις ανεξάρτητα από την ετικέτα της κατηγορίας οδηγούμαστε στον επόμενο τύπο:

$$P(\omega_k \mid \boldsymbol{s}) = \frac{P(\omega_k)}{P(\boldsymbol{s})} \prod_{i=1}^{L} P(s_i \mid \omega_k)$$

Σπάμε το γινόμενο σε 2 γινόμενα ανάλογα με το αποτέλεσμα της πρόβλεψης ω_k. Συμβολίζουμε με I_+^k το σύνολο των δεικτών των ταξινομητών που

πρότειναν ω_k και με I_{-}^{k} το σύνολο των δεικτών των ταξινομητών που πρότειναν οποιαδήποτε άλλη κλάση. Έχουμε λοιπόν:

$$P(\omega_k \mid \boldsymbol{s}) = \frac{P(\omega_k)}{P(\boldsymbol{s})} \prod_{I_{+}^{k}} P(s_i = \omega_k \mid \omega_k) * \prod_{I_{-}^{k}} P(s_i = \overline{\omega_k} \mid \omega_k)$$

Το σχήμα ψήφων που προτείνεται παραπάνω υποθέτει ότι αν υπάρχουν διαφορετικά ατομικά accuracies δηλαδή όταν:

$$P(s_i = \omega_k \mid \omega_k) = p_i \text{ (τύπος Α)}$$
$$\kappa\alpha\iota$$
$$P(s_i = \omega_j \mid \omega_k) = \frac{1-p_i}{c-1} \text{ (τύπος Β)}$$

τότε το σχήμα weighted majority vote είναι το βέλτιστο. Από τον τύπο Β συμπεραίνουμε ότι η πιθανότητα κατανέμεται ομοιόμορφα στις υπόλοιπες κλάσεις. Στην περίπτωση μας έχουμε 2 κλάσεις και συνεπώς ισχύει ο παραπάνω ισχυρισμός.

Αντικαθιστώντας τους τύπους Α και Β στην $P(\omega_k \mid \boldsymbol{s})$ έχουμε:

$$P(\omega_k \mid \boldsymbol{s}) = \frac{P(\omega_k)}{P(\boldsymbol{s})} \times \prod_{i\ \varepsilon\ I_{+}^{k}} p_i \times \prod_{i\ \varepsilon\ I_{-}^{k}} \frac{1-p_i}{c-1}$$
$$= \frac{P(\omega_k)}{P(\boldsymbol{s})} \times \prod_{i\ \varepsilon\ I_{+}^{k}} \frac{p_i\,(c-1)}{1-p_i} \times \prod_{i=1}^{L} \frac{1-p_i}{c-1}$$
$$= \frac{1}{P(\boldsymbol{s})} \times \prod_{i=1}^{L} \frac{1-p_i}{c-1} \times P(\omega_k) \times \prod_{i\ \varepsilon\ I_{+}^{k}} \frac{p_i\,(c-1)}{1-p_i}$$

Λογαριθμίζοντας κατά μέλη έχουμε το εξής:

$$\log(P(\omega_k \mid \boldsymbol{s})) = \log\left(\frac{\prod_{i=1}^{L}(1-p_i)}{P(\boldsymbol{s})(c-1)^L}\right) + log\big(P(\omega_k)\big)$$
$$+ \sum_{i\ \varepsilon |I_{+}^{k}|} \log\left(\frac{p_i}{1-p_i}\right) + |I_{+}^{k}| \times \log(c-1)$$

Ο πρώτος όρος δεν επηρεάζει την απόφαση της κλάσης και συνεπώς αφαιρείται. Αντικαθιστώντας c = 2 και εκφράζοντας τον 3ο όρο ως εξής:

$$\omega_i = log\left(\frac{p_i}{1-p_i}\right), 0 < p_i < 1$$

Καταλήγουμε στην παρακάτω εξίσωση για την πιθανότητα:

$$\log(P(\omega_k \mid \boldsymbol{s}) \propto \underbrace{\log\big(P(\omega_k)\big)}_{class\ constant} + \sum_{i\ \varepsilon |I_{+}^{k}|} \omega_i$$

Η σταθερά $\log(P(\omega_k))$ εξαρτάται από την πιθανότητα εμφάνισης της κάθε κλάσης. Επομένως μπορούμε να θεωρήσουμε ότι αφορά κάποια προκατάληψη από εμάς προς το ποιες κλάσεις αναμένουμε να έρθουν. Επιπλέον μπορεί να υπολογιστεί ως η τιμή η οποία δίνει το καλύτερο Recall στο test-set για το voting scheme.

Δοκιμάζοντας διάφορες τιμές πιθανότητας για την κλάση Legit παίρνουμε τον πίνακα 3.19 όπου φαίνονται οι αντίστοιχες τιμές για Accuracy, Precision και FNR. Παρατηρούμε ότι όσο μικρότερη πιθανότητα δίνουμε για εμφάνιση των Legitimate τόσο μειώνεται το FNR, αλλά ταυτόχρονα χάνουμε σε Precision και Accuracy, δηλαδή παρατηρούμε ένα trade-off.

Πιθανότητα για την κλάση Legit	**Accuracy**	**Precision**	**FNR**
0.005	93.43	76.18	2.30
0.01	93.75	77.17	2.37
0.02	94.02	78.03	2.47
0.05	94.33	79.16	2.77
0.10	**94.53**	**79.90**	**2.93**
0.15	**94.70**	**80.50**	**3.00**
0.20	94.83	80.98	3.10
0.25	94.94	81.4	3.17
0.30	95.02	81.69	3.20
0.35	95.09	81.96	3.23
0.40	95.16	82.21	3.27
0.45	95.23	82.49	3.33
0.50	95.25	82.57	3.37
0.60	95.35	83.04	3.57
0.70	95.48	83.60	3.70
0.80	95.59	84.09	3.83
0.90	95.77	84.92	4.10

Πίνακας 3.19: Εύρεση βέλτιστης τιμής για τη σταθερά κλάσης του voting scheme

Παρατηρώντας τα αποτελέσματα και επιθυμώντας το FNR να παραμείνει κοντά στην απόδοση του βέλτιστου μοντέλου (2.6%) διατηρώντας ωστόσο τα υψηλά επίπεδα Accuracy και Precision των κλασσικών αλγορίθμων μηχανικής μάθησης, επιλέγουμε μία αναλογία 12-88 για Legitimate-Phishing.

Συγκεκριμένα, το voting scheme με αυτή την παραμετροποίηση πετυχαίνει στο σύνολο ελέγχου Accuracy: 94.6 %, Precision: 80.15 % και FNR: 2.96 %.

Αυτή η επιλογή έγινε γιατί έχουμε αρκετά ικανοποιητικά αποτελέσματα μιας και το FNR είναι το βέλτιστο από το αν χρησιμοποιούσαμε 4 εκ των 5 ταξινομητές μόνους τους.

Ταυτόχρονα το Precision είναι σε ικανοποιητικό επίπεδο μιας και η εφαρμογή έχει κύριο σκοπό να μάθει να ανιχνεύει τα Phishing URLs όχι όμως σε βάρος των Legitimate.

Κεφάλαιο 4

Υλοποίηση Διαδικτυακής Εφαρμογής

4.1 Απαιτήσεις Συστήματος

Σε αυτή την ενότητα θα αναλύσουμε όλες τις απαιτήσεις συστήματος που φροντίσαμε να ικανοποιεί η εφαρμογή μας, προκειμένου να είναι εύχρηστη και να χαίρει ομαλής λειτουργίας. Οι απαιτήσεις συστήματος χωρίζονται στις λειτουργικές, που είναι οι υπηρεσίες που πρέπει να παρέχει το σύστημα και στις μη λειτουργικές που αφορούν περιορισμούς στις υπηρεσίες ή στις λειτουργίες του συστήματος.

4.1.1 Λειτουργικές Απαιτήσεις Συστήματος

Οι λειτουργικές απαιτήσεις που φροντίσαμε να ικανοποιεί το σύστημα μας είναι:

- Δίνει στο χρήστη την δυνατότητα με 3 μόνο κινήσεις να λάβει την πρόβλεψη για το URL που εισάγει.
- Ελέγχει αν το URL που εισήγαγε ο χρήστης πληροί τα χαρακτηριστικά ενός έγκυρου URL.
- Ελέγχει αν το URL που εισήχθη αντιστοιχεί σε ενεργή ιστοσελίδα.
- Εμφανίζει λεπτομερώς την πρόβλεψη μαζί με την πιθανότητα της για κάθε εκτιμητή και παρέχει και μια τελική εκτίμηση-αξιολόγηση για το URL που εισήχθη από το χρήστη.
- Προσφέρει άμεση και ευκατανόητη παρουσίαση τόσο των επιμέρους προβλέψεων των ταξινομητών όσο και της τελικής εκτίμησης.
- Επιτρέπει στο χρήστη να προβεί σε νέα καταχώριση URL προς αναζήτηση από την ίδια σελίδα που τυπώθηκε το αποτέλεσμα της προηγούμενης αναζήτησης.
- Παρέχει συγκεντρωτική αναφορά (καρτέλα στατιστικά) για την ταξινόμηση των URLs που έδωσε πρόβλεψη.

4.1.2 Μη Λειτουργικές Απαιτήσεις Συστήματος

Οι μη λειτουργικές απαιτήσεις που φροντίσαμε να ικανοποιεί το σύστημα μας είναι:

- Χρηστικότητα: Το σύστημα είναι πολύ φιλικό στη χρήση. Ο χρήστης χρειάζεται απλά να πληκτρολογήσει ή να κάνει επικόλληση το URL που τον ενδιαφέρει. Σε περίπτωση λανθασμένης πληκτρολόγησης ή ανενεργής ιστοσελίδας ο χρήστης ενημερώνεται άμεσα για να ξαναπροσπαθήσει. Η εφαρμογή είναι γραμμένη στα αγγλικά οπότε μπορεί να χρησιμοποιηθεί

διεθνώς από οποιοδήποτε ηλικιακή ομάδα χωρίς να απαιτεί ιδιαίτερες ή εξειδικευμένες γνώσεις. Ο συνδυασμός του αποτελέσματος μαζί με το κατάλληλο χρώμα (πράσινο-legitimate και κόκκινο-phishing) καθιστά εύκολα αναγνωρίσιμο το αποτέλεσμα. Ομοίως, για την παρουσίαση της τελικής εκτίμησης χρησιμοποιείται έντονη και μεγαλύτερη γραμματοσειρά προκειμένου να ξεχωρίζει εύκολα από τις επιμέρους προβλέψεις των ταξινομητών.

- Απόδοση: Σε μερικά μόνο δευτερόλεπτα, η εφαρμογή φροντίζει να ερευνήσει αν το δοθέν URL αντιστοιχεί σε ενεργή ιστοσελίδα και κατόπιν να εξάγει τα απαραίτητα χαρακτηριστικά που θα χρησιμοποιηθούν στους classifiers, οι οποίοι θα ταξινομήσουν το URL. Σε κάθε άλλη περίπτωση η εφαρμογή ανταποκρίνεται άμεσα υποδεικνύοντας το λάθος στο χρήστη. Το σύστημα λειτουργεί αδιάλειπτα και επιτρέπει πολλές δοσοληψίες-παράλληλες αναζητήσεις. Χάρις στο σύστημα CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) από την Google αποφεύγονται αναζητήσεις από bots που θα δημιουργούσαν μεγάλο φόρτο εργασίας στην εφαρμογή.

- Ιδιωτικότητα: Η εφαρμογή είναι μη κερδοσκοπικού σκοπού και αποτελεί μέρος ερευνητικής εργασίας. Για το λόγο αυτό δεν ζητάει από το χρήστη κανένα προσωπικό στοιχείο, ούτε απαιτεί εγγραφή. Δεν αποθηκεύει επίσης την IP του ή άλλα διαπιστευτήρια. Τα στατιστικά που χρησιμοποιούμε είναι απλά συγκεντρωτικού χαρακτήρα και προκύπτουν από το αποτέλεσμα που προτείνει η εφαρμογή. Δεν είναι προσωπικά για τον κάθε χρήστη.

- Επεκτασιμότητα: Πρόκειται για ένα open-source project με όλο τον κώδικα αναρτημένο στο Github που περιέχει όλες τις απαραίτητες επεξηγήσεις. Χρησιμοποιούμε τεχνολογίες αιχμής που είναι κατασκευασμένες για να επεκτείνονται. Επομένως ο καθένας μπορεί να το επεκτείνει άμεσα προσθέτοντας παραπάνω μοντέλα (ταξινομητές) χωρίς να απαιτείται καμία αλλαγή στον τρόπο υπολογισμού του voting scheme.

- Φορητότητα (portability): Το σύστημα είναι όλο γραμμένο σε γλώσσα Python3 οπότε για την εγκατάσταση του σε τοπικό υπολογιστή ή server (εξυπηρετητή) απαιτείται μόνο η εγκατάσταση της Python3 και των απαραίτητων βιβλιοθηκών.

4.2 Αρχιτεκτονική Εφαρμογής

Η εφαρμογή απαρτίζεται από 2 συντελεστές: το Frontend δηλαδή αυτό που βλέπει ο χρήστης όταν επισκέπτεται την εφαρμογή και το Backend που εξυπηρετεί τα αιτήματα του Frontend.

4.2.1 Υποδομή client (Frontend)

Το περιβάλλον χρήστη (user interface, συντ. UI) αποτελείται από 4 καρτέλες και 5 σελίδες. Οι καρτέλες είναι «URL Detection» (Ανίχνευση URL), «Features» (Χαρακτηριστικά), «Research Report» (Ερευνητική Αναφορά), «Statistics» (Στατιστικά) και οι σελίδες αποτελούνται από την αρχική σελίδα στην οποία ο χρήστης μπορεί να κάνει αναζήτηση για όποιο URL επιθυμεί, μια

σελίδα επεξήγησης των χαρακτηριστικών των URLs που χρησιμοποιήσαμε, μια σελίδα με την έκθεση της έρευνας, μια σελίδα στατιστικών χρήσης της εφαρμογής και τέλος η σελίδα αποτελεσμάτων.
Πιο αναλυτικά:

1) Σελίδα αναζήτησης: Είναι η αρχική σελίδα στην οποία μεταφέρεται ο χρήστης όταν μπαίνει στην εφαρμογή. Εκεί μπορεί να αναζητήσει οποιοδήποτε URL με μόνη προϋπόθεση την συμπλήρωση του CAPTCHA. Υπάγεται στην καρτέλα «URL Detection».

2) Σελίδα χαρακτηριστικών: Η σελίδα όπου περιγράφονται τα χαρακτηριστικά των URLs που υπολογίζονται κατά την εξαγωγή χαρακτηριστικών, πριν εισέλθουν στους εκτιμητές. Υπάγεται στην καρτέλα «Features».

3) Σελίδα έκθεσης έρευνας: σελίδα η οποία περιέχει σύντομη αναφορά των μοντέλων μηχανικής μάθησης που χρησιμοποιούνται στην εφαρμογή δίνοντας σε μορφή εικόνας τα αποτελέσματα των μετρικών και του voting scheme στο σύνολο ελέγχου. Επίσης, υπάρχει μια ανακατεύθυνση (redirect) προς ένα Github αποθετήριο (repository) που δημιουργήσαμε αποκλειστικά για την εφαρμογή, όπου οι χρήστες θα βρουν περισσότερες τεχνικές λεπτομέρειες. Υπάγεται στην καρτέλα «Research Report».

4) Σελίδα στατιστικών: Η σελίδα αυτή περιέχει ένα διάγραμμα πίτας με τα ποσοστά των αποτελεσμάτων των URLs για τα οποία έχει κάνει εκτίμηση η εφαρμογή, καθώς και πόσες φορές έχει χρησιμοποιηθεί η εφαρμογή. Υπάγεται στην καρτέλα «Statistics».

5) Σελίδα αποτελεσμάτων: Στη σελίδα αυτή παρουσιάζεται η εκτίμηση μαζί με την πιθανότητα της προβλεπόμενης κλάσης από κάθε εκτιμητή καθώς και η τελική πρόταση όπως προκύπτει από το voting scheme. Τέλος δίνεται στο χρήστη η δυνατότητα να αναζητήσει ένα νέο URL. Υπάγεται στην καρτέλα «URL Detection».

Τέλος, σε όλες τις παραπάνω σελίδες υπάρχει ένα υποσέλιδο (footer) όπου περιέχει το σύμβολο των πνευματικών δικαιωμάτων καθώς και έναν σύνδεσμο για ανακατεύθυνση προς το GitHub repository που δημιουργήσαμε, όπου οι χρήστες μπορούν να μάθουν περισσότερες τεχνικές πληροφορίες, να βρουν τον κώδικα που δημιουργήσαμε στα πλαίσια της παρούσας διπλωματικής και να μάθουν πώς να επικοινωνήσουν μαζί μας.

Τα πιθανά σενάρια χρήσης της εφαρμογής είναι τα εξής:

1) Ο χρήστης πληκτρολογεί ένα URL και δεν συμπληρώνει το CAPTCHA. Τότε εμφανίζεται ένα σφάλμα αμέσως μετά το πάτημα του κουμπιού Search που ενημερώνει τον χρήστη ότι η συμπλήρωση του CAPTCHA είναι απαραίτητη. Το σφάλμα εμφανίζεται με κόκκινα γράμματα για να γίνει εύκολα ορατό στον χρήστη.

2) Ο χρήστης πληκτρολογεί ένα URL που δεν ξεκινάει με http:// ή

https:// και συμπληρώνει σωστά το CAPTCHA. Τότε το αίτημα αυτό δεν θα σταλθεί στον Server. Ο χρήστης θα ενημερωθεί άμεσα για το σφάλμα του στην ίδια σελίδα με κόκκινα γράμματα μέσω χρήσης της γλώσσας Javascript (συντ. JS).

3) Ο χρήστης πληκτρολογεί ένα URL που ξεκινάει με http:// ή https:// αλλά δεν είναι έγκυρο ως προς την δομή (πχ https://ThisIsWrong). Τότε ο χρήστης ενημερώνεται άμεσα ότι δεν πρόκειται για URL και παρακινείται να ξαναδοκιμάσει.

4) Ο χρήστης πληκτρολογεί ένα σωστά δομημένο URL και συμπληρώνει και το CAPTCHA. Τότε θα του εμφανιστεί μία οθόνη φόρτωσης (loading) που τον ενημερώνει ότι γίνεται ο υπολογισμός των χαρακτηριστικών του δοθέντος URL καθώς και πρόβλεψη από τους εκτιμητές. Όταν τελειώσουν οι παραπάνω υπολογισμοί τότε μεταφέρετε στην σελίδα αποτελεσμάτων.

5) Ο χρήστης πληκτρολογεί ένα ανενεργό URL. Τότε μεταφέρεται στη σελίδα αποτελεσμάτων που τον ενημερώνει γι' αυτό μέσω μηνύματος σφάλματος και του δίνεται η δυνατότητα να αναζητήσει ένα νέο URL.

4.2.2 Υποδομή Server (Backend)

Σε αυτό το σημείο θα αναφερθούμε στον Server, ο οποίος εξυπηρετεί τα αιτήματα του Frontend, εξάγει τα χαρακτηριστικά των URLs και εκτελεί τις προβλέψεις των μοντέλων για το δοθέν URL. Ο Server υλοποιήθηκε χρησιμοποιώντας την βιβλιοθήκη Flask της Python 3, που επιτρέπει στο χρήστη να δημιουργεί έναν Server, να διαχειρίζεται HTTP αιτήματα και να απαντά σε αυτά κάνοντας ανακατεύθυνση στις αντίστοιχες HTML σελίδες. Συγκεκριμένα, επιτρέπουμε μόνο τις μεθόδους GET και POST. Η GET ουσιαστικά δίνει στον χρήστη την σελίδα που ζήτησε, ενώ η POST μεταφέρει τα δεδομένα (στην περίπτωση μας URLs) από τον χρήστη στον Server ο οποίος τα επεξεργάζεται και απαντάει κατάλληλα, εφόσον το δοθέν URL πληροί τα κριτήρια ενός σωστά δομημένου URL.

Πιο συγκεκριμένα, κατά τη δημιουργία του ο Server φορτώνει τα 5 μοντέλα, ορίζει κάποιες μεταβλητές για τα στατιστικά, κάποιες σταθερές για το voting scheme και τέλος δημιουργεί τον standard scaler με βάση τα δεδομένα εκπαίδευσης. Αυτά ορίζονται σε global μεταβλητές γιατί είναι απαραίτητα καθ' όλη τη χρήση της εφαρμογής. Στην συνέχεια, ορίζεται μία συνάρτηση η οποία ανάλογα με την μέθοδο που κλήθηκε η εφαρμογή εκτελεί διαφορετική λειτουργία. Αν κληθεί με GET HTTP μέθοδο αιτήματος επιστρέφει την αρχική σελίδα στον χρήστη στην οποία αυτός μπορεί να εισάγει κάποιο URL προς πρόβλεψη από τους εκτιμητές. Αφού εισαχθεί το URL και πληροί την δομή ενός σωστού URL, ο χρήστης πατώντας το κουμπί Search (Αναζήτηση) στέλνει στον Server ένα POST αίτημα. Ο Server το λαμβάνει και κάνει τα εξής βήματα :

1) Ελέγχει αν πρόκειται για ενεργό URL με βάση το status του HTTP response προς το URL.

2) Εφόσον δεν πρόκειται για ενεργό ενημερώνει άμεσα στον χρήστη. Σε αντίθετη περίπτωση ξεκινάει η διαδικασία εξαγωγής χαρακτηριστικών, που είναι και η πιο χρονοβόρα.

3) Στην συνέχεια τα χαρακτηριστικά αυτά κλιμακώνονται. Σε μία μεταβλητή, τα χαρακτηριστικά δέχονται κανονική κλιμάκωση (Standard Scaling) και χρησιμοποιούνται στα μοντέλα μηχανικής μάθησης ενώ σε άλλη μεταβλητή τα δεδομένα δέχονται κλιμάκωση μεγίστου (Max Scaling) και χρησιμοποιούνται στο μοντέλο CNN.

4) Εν συνεχεία, γίνεται η πρόβλεψη από το κάθε μοντέλο καθώς και από το voting scheme και ανακατευθύνει τον χρήστη στην σελίδα αποτελεσμάτων.

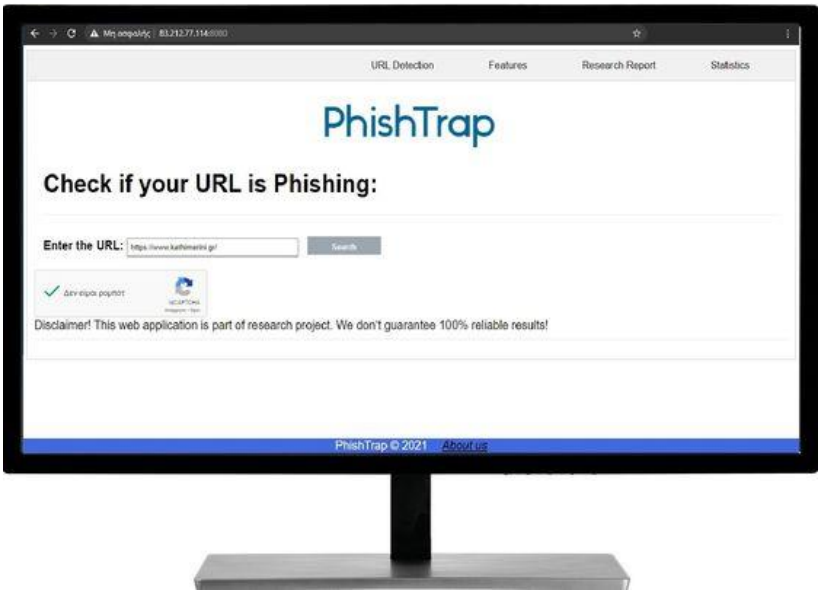
Ο κώδικας της εφαρμογής είναι διαθέσιμος στο Github: https://github.com/souliotispanagiotis/PhishTrap

4.3 Εμφάνιση Εφαρμογής

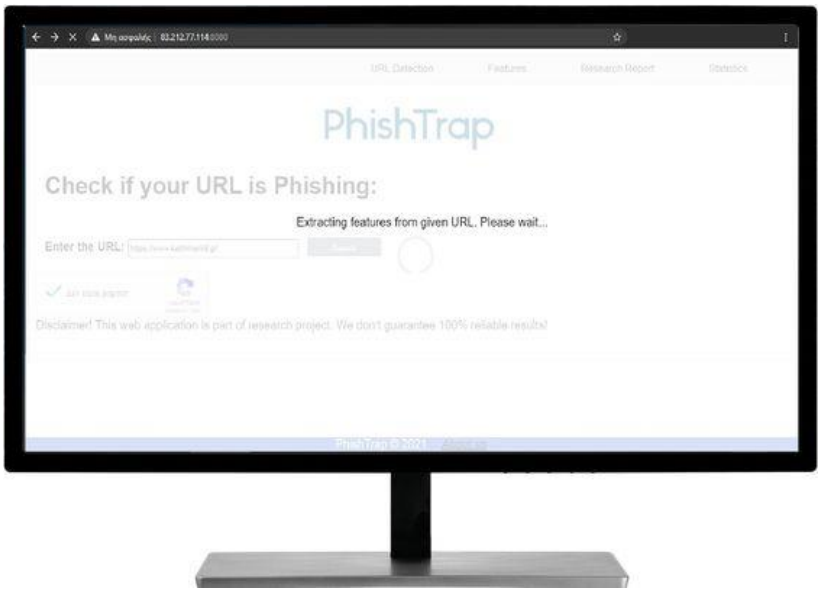
Στην ενότητα αυτή θα παρατεθούν στιγμιότυπα από το πώς εμφανίζεται η διαδικτυακή εφαρμογή σε περιβάλλον υπολογιστή (desktop), κινητού (mobile) και ταμπλέτας (tablet). Έχει δοθεί μεγάλη προσοχή στην εμφάνιση της εφαρμογής ούτως ώστε να είναι εύχρηστη και καλαίσθητη και στοχεύει να αφήνει μια ευχάριστη εμπειρία στον χρήστη.

4.3.1 Οθόνη Desktop

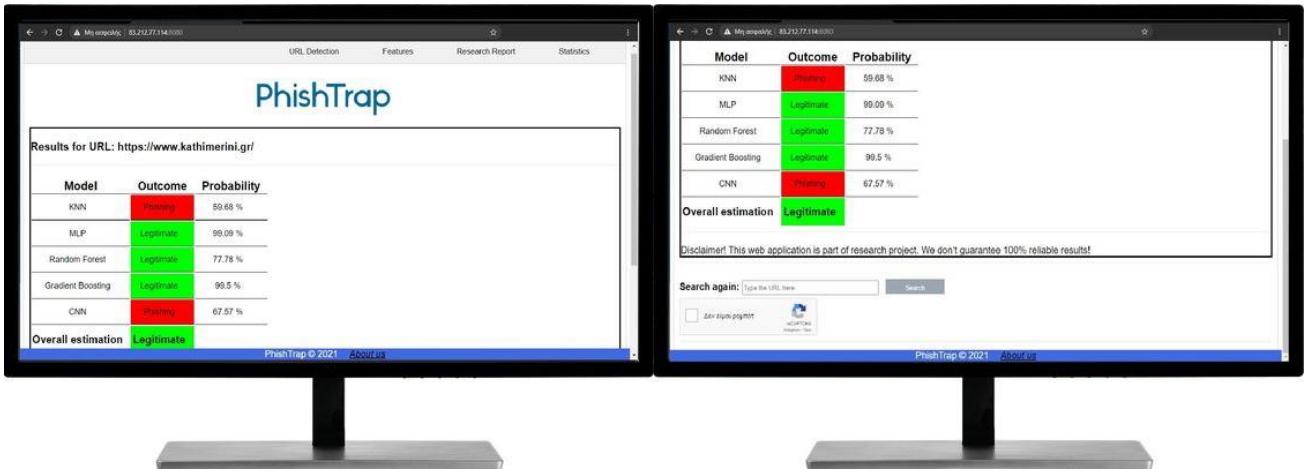
Θα παρουσιαστούν σε εικόνες οι 5 βασικές σελίδες της εφαρμογής. Θα χρησιμοποιήσουμε ως παράδειγμα αναζήτησης τον legitimate ειδησεογραφικό ιστότοπο www.kathimerini.gr.



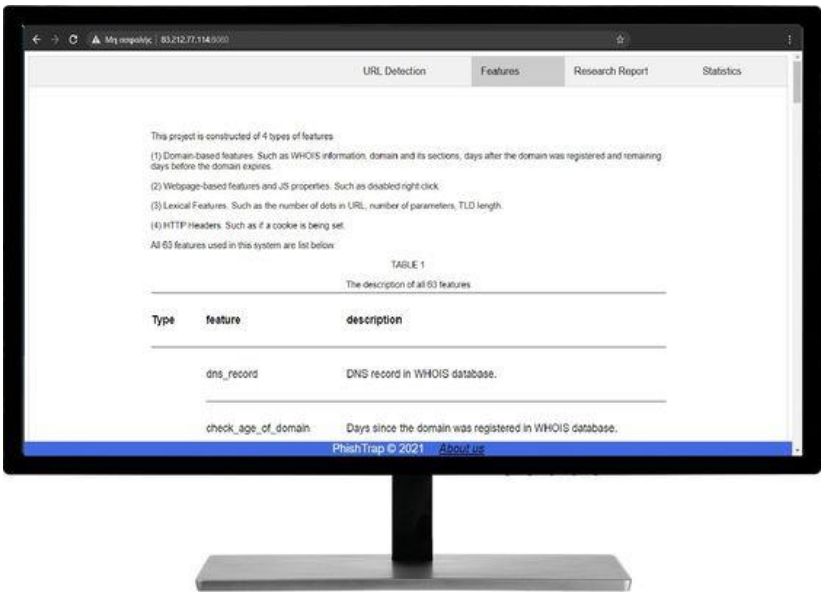
Εικόνα 4.1: Αρχική σελίδα σε περιβάλλον Desktop



Εικόνα 4.2: Οθόνη φόρτωσης σε περιβάλλον Desktop

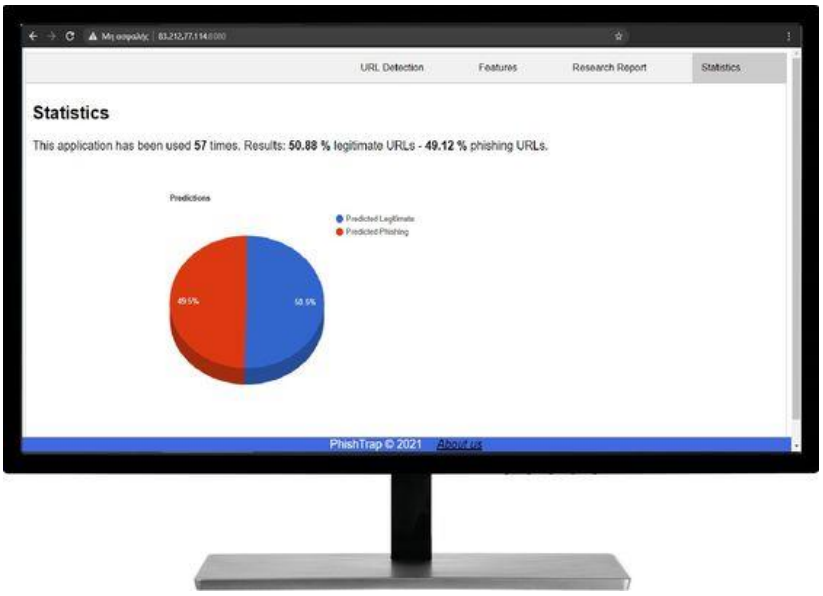


Εικόνα 4.3: Σελίδα παρουσίασης αποτελεσμάτων σε περιβάλλον Desktop



Εικόνα 4.4: Σελίδα χαρακτηριστικών σε περιβάλλον Desktop

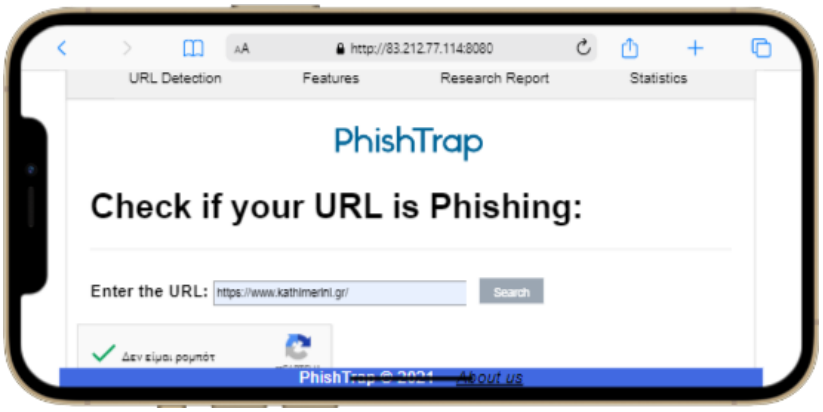
Εικόνα 4.5: Σελίδα Ερευνητικής Αναφοράς σε περιβάλλον Desktop



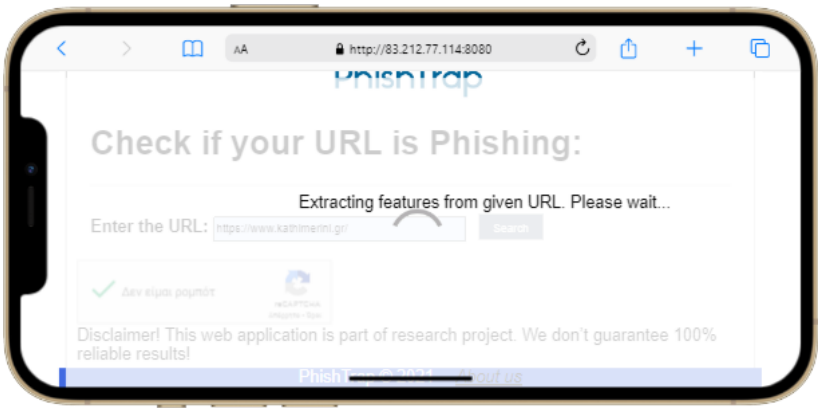
Εικόνα 4.6: Σελίδα Στατιστικών σε περιβάλλον Desktop

4.3.2 Οθόνη Mobile

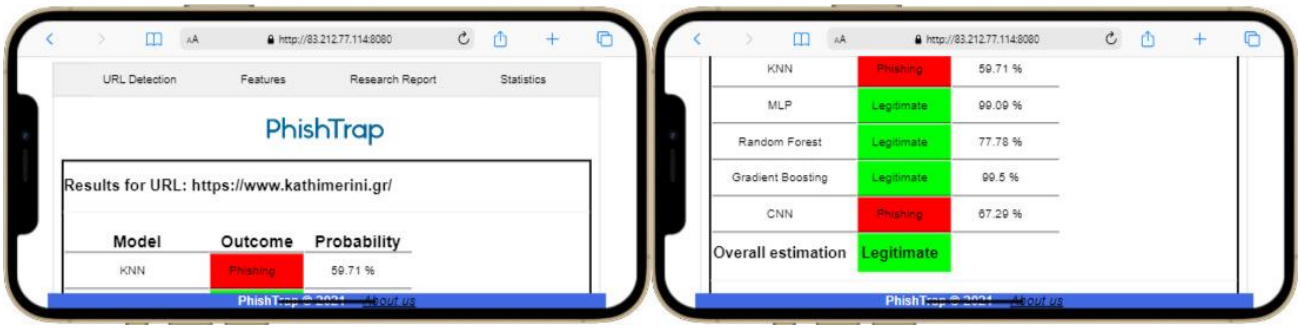
Θα παρουσιαστεί σε εικόνες η βασική λειτουργία της εφαρμογής δηλαδή η καταχώριση ενός έγκυρου URL και η λήψη εκτίμησης για την κλάση που ανήκει, χρησιμοποιώντας πάλι ως παράδειγμα αναζήτησης τον legitimate ειδησεογραφικό ιστότοπο www.kathimerini.gr.



Εικόνα 4.7: Αρχική σελίδα σε περιβάλλον Mobile



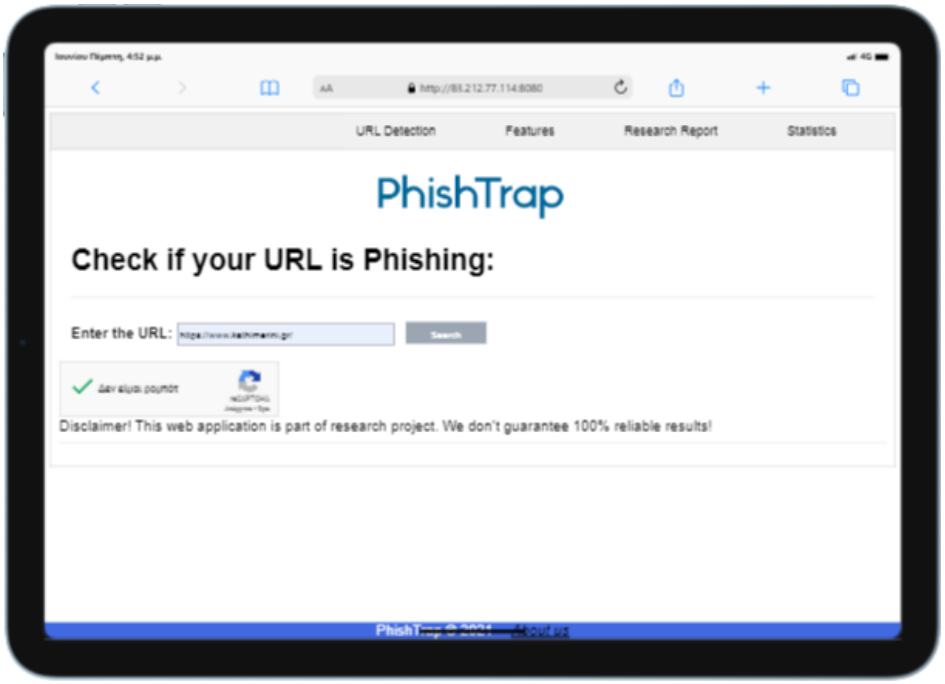
Εικόνα 4.8: Οθόνη φόρτωσης σε περιβάλλον Mobile



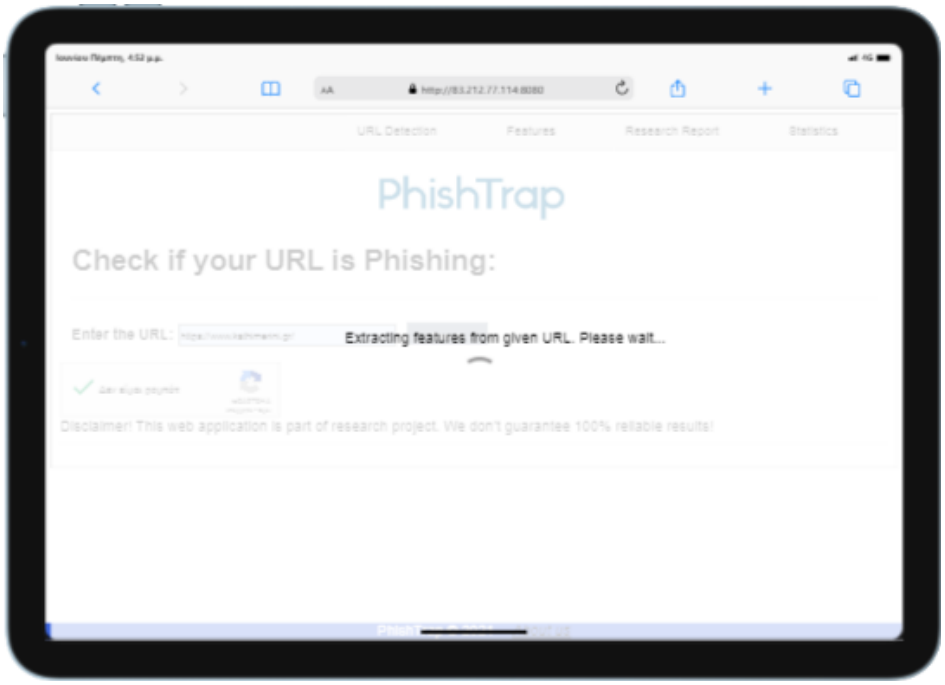
Εικόνα 4.9: Σελίδα παρουσίασης αποτελεσμάτων σε περιβάλλον Mobile

4.3.3 Οθόνη Tablet

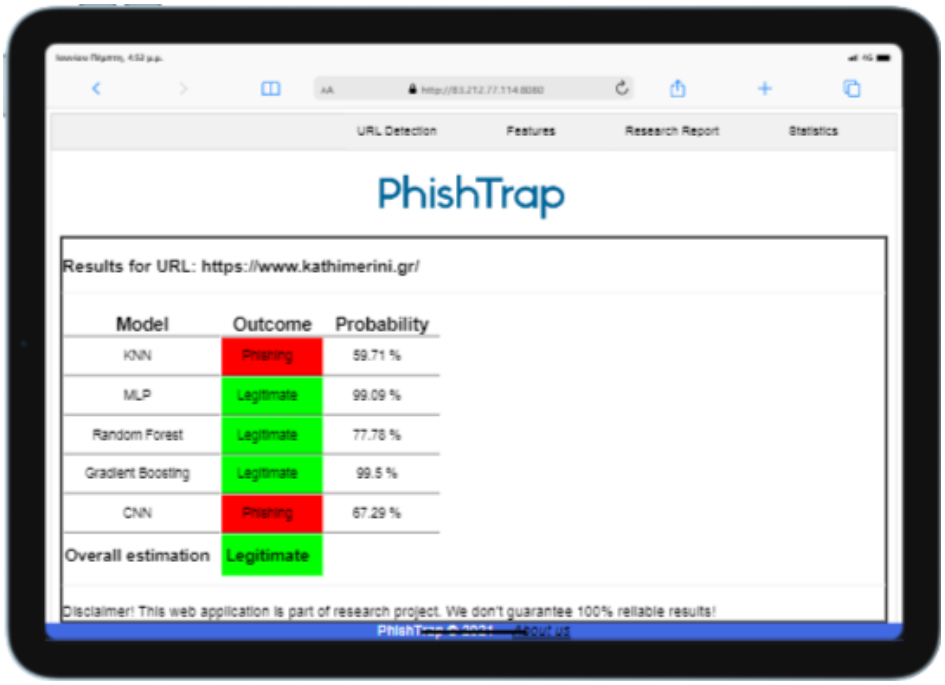
Θα παρουσιαστούν οι αντίστοιχες εικόνες με την προηγούμενη ενότητα.



Εικόνα 4.10: Αρχική σελίδα σε περιβάλλον Tablet



Εικόνα 4.11: Οθόνη φόρτωσης σε περιβάλλον Tablet



Εικόνα 4.12: Σελίδα παρουσίασης αποτελεσμάτων σε περιβάλλον Tablet

4.4 Φιλοξενία Διαδικτυακής Εφαρμογής σε Server

Προκειμένου οι διαδικτυακές εφαρμογές να είναι διαθέσιμες από κάθε είδους συσκευή (desktop, mobile, tablet) και να μην απαιτείται εγκατάσταση από τον χρήστη, απαραίτητη είναι η φιλοξενία τους (deploy) σε έναν Server, ο οποίος θα δώσει τη δυνατότητα να είναι συνεχώς διαθέσιμες σε όλους τους Χρήστες του Διαδικτύου. Ο Server μπορεί να ανήκει είτε σε κάποια εμπορική πλατφόρμα, η οποία προσφέρει αυτή την υπηρεσία επί πληρωμή και συχνά επιτρέπει και την δωρεάν φιλοξενία εφαρμογών με περιορισμένες δυνατότητες και πόρους για δοκιμαστικούς σκοπούς είτε σε κάποιον οργανισμό/ιδιώτη (πχ Πανεπιστημιακό Ίδρυμα).

4.4.1 Εμπορικές Πλατφόρμες

Υπάρχουν πολλές εταιρείες τεχνολογίας που προσφέρουν τέτοιου είδους υπηρεσίες, που διαρθρώνονται συνήθως σε πακέτα με συγκεκριμένη χρονική και χρηματική συνδρομή και δυνατότητες. Οι κυριότερες εμπορικές πλατφόρμες φιλοξενίας διαδικτυακών εφαρμογών τον Ιούνιο του 2021 είναι:

Heroku[15]

Είναι μία πλατφόρμα υπολογιστικού νέφους (cloud) που παρέχεται ως υπηρεσία (Platform as a Service, συντ. PaaS). Υποστηρίζει πολλές γλώσσες προγραμματισμού. Ο χρήστης δεν χρειάζεται να ανεβάσει σε αυτό τα αρχεία της εφαρμογής του αλλά απλά να το συνδέσει με τον λογαριασμό του στο Github και να επιλέξει το αντίστοιχο repository. Στο δωρεάν πακέτο που προσφέρει, δεν απαιτείται τραπεζική κάρτα για εγγραφή ωστόσο υπάρχουν περιορισμοί στις ώρες που είναι διαθέσιμη η εφαρμογή.

[15] https://www.heroku.com/

Ήταν η πρώτη μας επιλογή για φιλοξενία της εφαρμογής μας, ωστόσο περιορισμοί στο επιτρεπόμενο μέγεθος των αρχείων στο Github εμπόδισαν την αποθήκευση των μοντέλων μας με αποτέλεσμα να καταστεί αδύνατη εν τέλει η φιλοξενία της εφαρμογής μας στο Heroku.

Google Cloud[16]

Είναι μία σουίτα υπηρεσιών υπολογιστικού νέφους που προσφέρεται από την Google. Δεν περιορίζεται απλά στην φιλοξενία διαδικτυακών εφαρμογών, αλλά παρέχει πάρα πολλά εργαλεία για Βάσεις Δεδομένων, Big Data, IoT (Internet of Things), Cloud AI (Artificial Intelligence), Networking (Δικτυακά) και άλλες πολλές ακόμη υπηρεσίες. Η ανάπτυξη και η φιλοξενία διαδικτυακών εφαρμογών υποστηρίζεται από το App Engine. Η εγγραφή απαιτεί καταχώριση τραπεζικής κάρτας χωρίς χρέωση ωστόσο και προσφέρει 90 ημέρες δωρεάν χρήσης. Προσφέρει πληθώρα επιλογών για παραμετροποίηση των εφαρμογών, σε συνδυασμό με ένα εύχρηστο UI που διαθέτει μεταξύ άλλων γραμμή εντολών (terminal), εξερευνητή αρχείων (file explorer) και επεξεργαστή κώδικα (code editor). Ο χρήστης μπορεί είτε να συγγράψει απευθείας κώδικα είτε να ανεβάσει στην πλατφόρμα τα αρχεία που επιθυμεί.

Αποτέλεσε την δεύτερη επιλογή μας μετά το Heroku. Έπειτα από αρκετές προσπάθειες αποφασίσαμε ότι το δωρεάν του πρόγραμμα δεν μας ικανοποιεί, καθώς λόγω του μεγέθους των μοντέλων αργούσε πάρα πολύ να στήσει τον Server και παρουσίαζε συχνά καθυστερήσεις και στην εκτέλεση της εφαρμογής.

AWS Elastic Beanstalk[17]

Είναι μία υπηρεσία που προσφέρεται από την Amazon για την φιλοξενία διαδικτυακών εφαρμογών. Εξυπηρετεί αρκετές γλώσσες προγραμματισμού προσφέροντας αρκετά είδη Server. Η εγγραφή απαιτεί τραπεζική κάρτα και μικρή χρέωση. Η χρήση της πλατφόρμας είναι δωρεάν ωστόσο απαιτείται πληρωμή για τη χρήση της εφαρμογής σε πόρους αποθήκευσης και λειτουργίας.

Ερευνώντας τα χαρακτηριστικά του δωρεάν προγράμματος, αποφασίσαμε πως δεν μας ικανοποιεί λόγω του μεγέθους των μοντέλων μας.

4.4.2 ~okeanos-knossos[18]

Αποτελεί μία υποδομή που προσφέρεται ως υπηρεσία (Infrastructure as a Service, συντ. IaaS). Ουσιαστικά πρόκειται για εικονικά μηχανήματα (virtual machines, συντ. VMs) συνδεδεμένα στο διαδίκτυο, τα οποία παραχωρούνται στους χρήστες, που είναι κατά κύριο λόγο οι σπουδαστές και το προσωπικό των Πανεπιστημιακών Ιδρυμάτων της Ελλάδας, προκειμένου να τους βοηθήσουν στην ανάπτυξη των ερευνητικών τους εργασιών.

Αποτέλεσε την τελική μας επιλογή, καθώς προσφέρει απλότητα χρήσης και καλύπτει απόλυτα τις ανάγκες της εφαρμογής μας. Επί της ουσίας, απαιτήθηκε η δημιουργία ενός εικονικού μηχανήματος με λειτουργικό Linux Server και η εγκατάσταση της γλώσσας Python 3 μαζί με τα απαραίτητα πακέτα και βιβλιοθήκες.

[16] https://cloud.google.com/
[17] https://aws.amazon.com/elasticbeanstalk/
[18] https://okeanos-knossos.grnet.gr/home/

Κεφάλαιο 5

Αποτελέσματα και Μελλοντικές Κατευθύνσεις-Επεκτάσεις

5.1 Αποτελέσματα

Σε αυτήν την ενότητα θα παρουσιάσουμε και θα συγκρίνουμε τα αποτελέσματα της εκπαίδευσης των αλγορίθμων μηχανικής μάθησης και της επιλογής και παραμετροποίησης του voting scheme πάνω στο σύνολο ελέγχου που δημιουργήσαμε. Επιπλέον, μέσα από παραδείγματα με URLs που αντιστοιχούν σε πανομοιότυπες εμφανισιακά ιστοσελίδες θα εξερευνήσουμε τις δυνατότητες της εφαρμογής μας.

5.1.1 Αποτελέσματα στο σύνολο ελέγχου

Στον πίνακα 5.1 συνοψίζονται τα μοντέλα μηχανικής μάθησης που αναπτύχθηκαν και χρησιμοποιούνται στο σύστημα πρόβλεψης της εφαρμογής μας, μαζί με τις μετρικές απόδοσης τους στο σύνολο ελέγχου.

	Accuracy	Precision	FNR
KNN	90.81 %	70.24 %	6.23 %
MLP	93.79 %	78.32 %	4.67 %
Random Forest	94.90 %	82.05 %	4.67 %
Gradient Boosting	95.77 %	84.67 %	3.73 %
CNN	88.40 %	63.75 %	2.60 %
Voting	94.60 %	80.15 %	2.96 %

Πίνακας 5.1: Τα μοντέλα μηχανικής μάθησης που χρησιμοποιούνται στο σύστημα πρόβλεψης με τις μετρικές απόδοσης τους στο σύνολο ελέγχου

Όπως βλέπουμε, όλα τα μοντέλα επιτυγχάνουν πολύ υψηλό ποσοστό Accuracy, ικανοποιητικά υψηλό ποσοστό Precision και σαφώς FNR κάτω του 7% που είχε τεθεί ως ελάχιστο threshold.

Πιο συγκεκριμένα, ο Gradient Boosting παρουσιάζει τα καλύτερα αποτελέσματα και στις 3 μετρικές έναντι των άλλων κλασσικών αλγορίθμων μηχανικής μάθησης. Το CNN αποτελεί το καλύτερο μοντέλο όσον αφορά την μετρική FNR. Ωστόσο, παρά το υψηλό του Accuracy, χαρακτηρίζεται από μέτριο Precision.

Το γεγονός αυτό ήταν που μας οδήγησε στην μελέτη και υιοθέτηση ενός voting scheme έναντι της απλής επιλογής του μοντέλου με το χαμηλότερο FNR. Η επιλογή αυτή άλλωστε θα αποδεικνυόταν ολέθρια, καθώς όπως μαρτυρούν οι εικόνες 5.6 και 5.13 το CNN μπορεί λόγω ισχνής πλειοψηφίας της προβλεπόμενης κλάσης να οδηγήσει σε λανθασμένη πρόβλεψη τόσο για legit όσο και για phishing URLs, ενώ ταυτόχρονα όλα τα άλλα μοντέλα θα έδιναν ισχυρές προβλέψεις επικράτησης της άλλης κλάσης.

Με την τεχνική Voting που επιλέξαμε καταφέρνουμε να συνδυάσουμε τα πολύ υψηλά ποσοστά για Accuracy και Precision των κλασσικών αλγορίθμων μηχανικής μάθησης με το πολύ χαμηλό FNR, το οποίο στάθηκε αδύνατο έστω να προσεγγίσουν οι προηγούμενοι και προσφέρει το μοντέλο βαθιάς μάθησης.

Βέβαια, εύλογα θα αναρωτηθεί κάποιος γιατί χρειάζεται ένα πιο περίτεχνο σύστημα voting και όχι απλά μια ψηφοφορία πλειοψηφίας. Η απάντηση δίνεται ευθέως από τη δεξιά στήλη της εικόνας 5.16. Μπορεί η σχετική πλειοψηφία (3 μοντέλα από τα 5 στο σύνολο) να συμφωνεί στην επικράτηση μιας κλάσης, ωστόσο η «μη βεβαιότητα» τους για την πρόβλεψη τους έναντι της ισχυρής γνώμης των άλλων (δύο μοντέλων) θα οδηγούσε με ψηφοφορία πλειοψηφίας σε λανθασμένο αποτέλεσμα.

Σαφώς, δε θα πρέπει να ξεχνάμε όμως ότι πρόκειται για ένα σύστημα πρόβλεψης, που αν και εξασφαλίζει αξιόλογα ποσοστά στις μετρικές αξιολόγησης και παρουσιάζει αξιέπαινα σωστά αποτελέσματα και κατά τη χρήση του, εντούτοις πρόκειται για προβλέψεις και δεν εγγυάται το τελικό αποτέλεσμα.

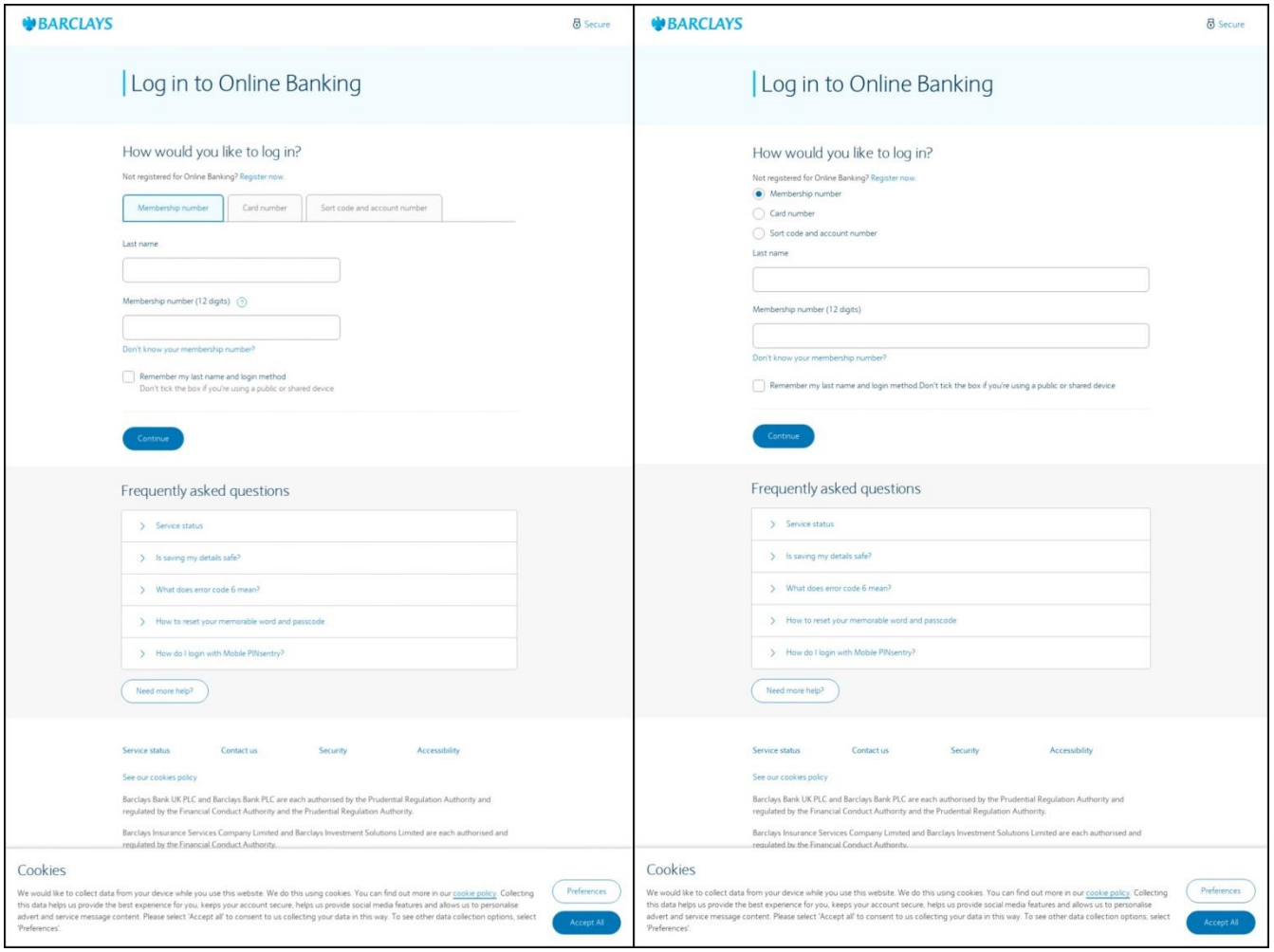
Συγκριτικά με άλλες επιστημονικές εργασίες παρόμοιου περιεχομένου, τα αποτελέσματα είναι πολύ ενθαρρυντικά κυρίως για την βασική μας μετρική FNR. Πιο συγκεκριμένα, για κλασσικούς αλγορίθμους μηχανικής μάθησης παρατηρούμε ότι το μοντέλο Gradient Boosting που έχουμε εκπαιδεύσει είναι πιο αποδοτικό σε σχέση με τα βέλτιστα μοντέλα άλλων μελετών, οι οποίες σημειώνεται ότι χρησιμοποιούν μικρότερα σε πλήθος και χαρακτηριστικά datasets [48] [57]. Σε ότι αφορά την αρχιτεκτονική Βαθιάς Μάθησης, το αποτέλεσμα ως προς το FNR ήταν πολύ κοντά (με διαφορά περίπου 0.5 %) από το βέλτιστο της βιβλιογραφίας [52], με αρκετά χαμηλότερο βέβαια Precision, που αποτελεί βέβαια δευτερεύον στόχο στο πρόβλημα μας. Ταυτόχρονα, το voting scheme εξαλείφει τα μειονεκτήματα του CNN, επιτυγχάνοντας να προσφέρει μια τελική εκτίμηση-αξιολόγηση με πολύ υψηλά ποσοστά Accuracy και Precision σε συνδυασμό με πολύ χαμηλό FNR. Πράγματι, όπως θα δούμε από τα παραδείγματα που ακολουθούν στην επόμενη ενότητα η εφαρμογή καταφέρνει όντως να ανιχνεύει αποτελεσματικά τα phishing URLs μεριμνώντας για την ασφάλεια του χρήστη.

Εν τέλει, κρίνεται πολύ ικανοποιητική η λειτουργία και τα αποτελέσματα της εφαρμογής, η οποία να σημειώσουμε ακόμη ότι καταφέρνει να ανιχνεύσει τις phishing ιστοσελίδες από το πρώτο λεπτό που δημοσιεύονται, σε αντίθεση με τα προγράμματα περιήγησης και τις βάσεις δεδομένων των εταιρειών κυβερνοασφάλειας, που απαιτούν κάποιο χρονικό διάστημα, ίσως και ένα πλήθος καταγγελιών (reports) από χρήστες.

5.1.2 Αποτελέσματα Εφαρμογής

Στην ενότητα αυτή θα δοκιμάσουμε τα αποτελέσματα που δίνει η εφαρμογή σε ζευγάρια legitimate-phishing ιστοσελίδων με σχεδόν πανομοιότυπη εμφάνιση και αρκετά παρόμοιο URL εξετάζοντας παραδείγματα από διάφορους τομείς. Σημειώνεται ότι οι δοκιμές πραγματοποιήθηκαν στο διάστημα 2-7 Ιουνίου 2021 και τα phishing URLs προέρχονται από το PhishTank [45].

Οι τράπεζες αποτελούν ένα από τα πιο αγαπημένα «δολώματα» των phishers. Στην εικόνα 5.1 βλέπουμε 2 σχεδόν πανομοιότυπες εικόνες της σελίδας login (σύνδεσης) της τράπεζας Barclays στο Ηνωμένο Βασίλειο. Η εικόνα στην αριστερή στήλη αποτελεί screenshot της legitimate login σελίδας της Barclays Bank και αντιστοιχεί στο URL https://bank.barclays.co.uk/, ενώ η εικόνα στη δεξιά στήλη αποτελεί screenshot μιας phishing login σελίδας της ίδιας τράπεζας και αντιστοιχεί στο URL https://barclays.co.uk-authid937.com/.



Εικόνα 5.1: Στην αριστερή στήλη παρουσιάζεται η legitimate login σελίδα της Barclays Bank και στη δεξιά η αντίστοιχη phishing.

Η κυριότερη διαφορά όσον αφορά το περιεχόμενο της σελίδας είναι στον τρόπο επιλογής καταχώρισης στοιχείων που στην αριστερή σελίδα δίνεται με καρτέλες και στη δεξιά με radio button. Στο κομμάτι του URL, από το phishing URL απουσιάζει η λέξη "bank" και έχει προστεθεί το τμήμα "authid937".
Στην εικόνα 5.2 παρουσιάζονται τα αποτελέσματα από τις προβλέψεις που

έδωσε η εφαρμογή μας, η οποία κατάφερε επιτυχώς να ταξινομήσει τις 2 ιστοσελίδες στην κατηγορία που ανήκουν.

Results for URL: https://bank.barclays.co.uk/

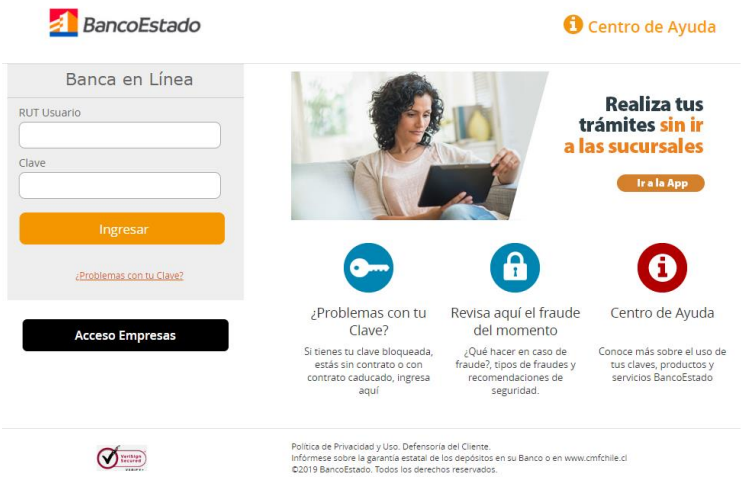
Model	Outcome	Probability
KNN	Legitimate	99.9 %
MLP	Legitimate	100.0 %
Random Forest	Legitimate	68.44 %
Gradient Boosting	Legitimate	100.0 %
CNN	Legitimate	77.23 %
Overall estimation	**Legitimate**	

Results for URL: https://barclays.co.uk-authid937.com/

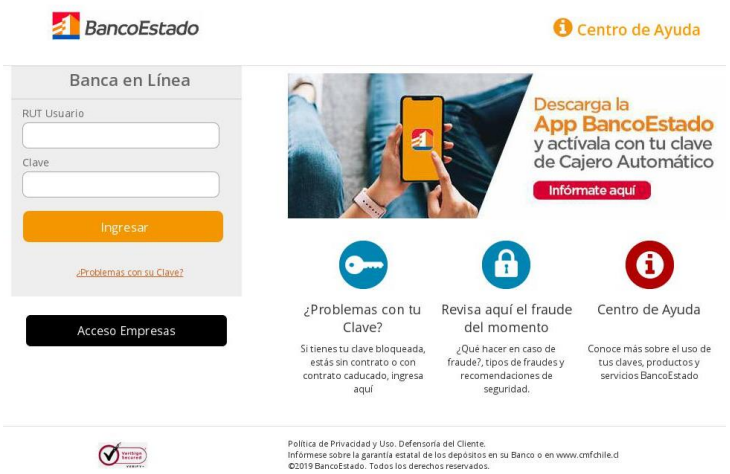
Model	Outcome	Probability
KNN	Phishing	61.37 %
MLP	Phishing	99.98 %
Random Forest	Phishing	68.0 %
Gradient Boosting	Phishing	99.72 %
CNN	Phishing	98.74 %
Overall estimation	**Phishing**	

Εικόνα 5.2: Τα αποτελέσματα της εφαρμογής για τα URLs της Barclays Bank. Στην αριστερή στήλη είναι το legitimate URL και στη δεξιά το phishing URL.

Στόχο δεν αποτελούν μονάχα οι τράπεζες των Δυτικών Χωρών, αλλά όλες οι τράπεζες του κόσμου. Στις εικόνες 5.3 και 5.4 βλέπουμε 2 σχεδόν πανομοιότυπες εικόνες της σελίδας login της τράπεζας Banco Estado στη Χιλή.



Εικόνα 5.3: Legitimate login σελίδα της Banco Estado



Εικόνα 5.4: Phishing login σελίδα της Banco Estado

Η legitimate login σελίδα έχει URL:
https://www.bancoestado.cl/imagenes/comun2008/banca-en-linea-personas.html
και η phishing έχει URL:
https://www.www-bancoestado.cl.traintalk.com.au/pagina/imagenes/comun2008/banca-en-linea-personas.html.

Η διαφορά στην εμφάνιση έγκειται στην διαφορετική κεντρική εικόνα των 2 σελίδων, η οποία όμως και στις 2 περιπτώσεις έχει κοινό θεματικό άξονα (τον χρήστη να είναι καθισμένος και να χρησιμοποιεί τις υπηρεσίες της τράπεζας με μια έξυπνη συσκευή, κινητό στην μία περίπτωση, tablet στην άλλη). Ο χρήστης πάντως λόγω του ότι τόσο η legit όσο και η phishing ιστοσελίδα διαθέτει ένα σηματάκι ασφαλείας από την ίδια κορυφαία εταιρεία παροχής υπηρεσιών ασφάλειας υπολογιστών, θα μπορούσε πολύ εύκολα να μπερδευτεί και να πιστέψει ότι η phishing σελίδα είναι legit. Σε αυτό συμβάλλει άλλωστε και το μεγάλο μήκος του legit URL, με το οποίο μοιάζει πάρα πολύ και το phishing που διαφέρει κυρίως στη σειρά των λέξεων.

Στις εικόνες 5.5 και 5.6 παρουσιάζονται τα αποτελέσματα από τις προβλέψεις που έδωσε η εφαρμογής μας, η οποία κατάφερε επιτυχώς να ταξινομήσει τις 2 ιστοσελίδες στην κατηγορία που ανήκουν.

Results for URL: https://www.bancoestado.cl/imagenes/comun2008/banca-en-linea-personas.html

Model	Outcome	Probability
KNN	Phishing	80.36 %
MLP	Legitimate	99.77 %
Random Forest	Legitimate	92.22 %
Gradient Boosting	Legitimate	99.93 %
CNN	Legitimate	70.55 %
Overall estimation	**Legitimate**	

Εικόνα 5.5: Αποτέλεσμα της εφαρμογής για την legit login σελίδα της Banco Estado

Results for URL: https://www.www-bancoestado.cl.traintalk.com.au/pagina/imagenes/comun2008/banca-en-linea-personas.html

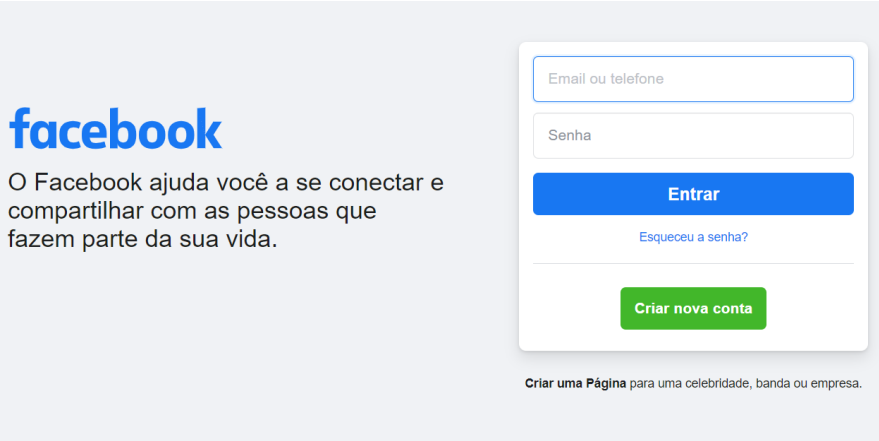
Model	Outcome	Probability
KNN	Phishing	59.33 %
MLP	Phishing	70.31 %
Random Forest	Phishing	58.67 %
Gradient Boosting	Phishing	89.88 %
CNN	Legitimate	54.09 %
Overall estimation	**Phishing**	

Εικόνα 5.6: Αποτέλεσμα της εφαρμογής για την phishing login σελίδα της Banco Estado

Ένας άλλο αγαπημένο δόλωμα είναι τα μέσα κοινωνικής δικτύωσης και κυρίως το Facebook.

Το legit URL του Facebook για την login σελίδα σε γλώσσα Πορτογαλικά Βραζιλίας είναι:
https://pt-br.facebook.com/
Αντίστοιχα, ένα phishing URL είναι:
https://faceboogk.filesusr.com/html/e971c7_8b35b2df874f14d55c6bab3bc4285eb1.html

Αν και διαφέρουν πάρα πολύ στο URL, η εμφάνιση των δύο login σελίδων είναι αρκετά όμοια, όπως φαίνεται στις εικόνες 5.7 και 5.8.



Εικόνα 5.7: Legitimate login σελίδα του Facebook

Εικόνα 5.8: Phishing login σελίδα του Facebook

Στις εικόνες 5.9 και 5.10 παρουσιάζονται τα αποτελέσματα από τις προβλέψεις που έδωσε η εφαρμογή μας, η οποία κατάφερε επιτυχώς να ταξινομήσει τις 2 ιστοσελίδες στην κατηγορία που ανήκουν.

Results for URL: https://pt-br.facebook.com/

Model	Outcome	Probability
KNN	Legitimate	99.9 %
MLP	Legitimate	97.35 %
Random Forest	Legitimate	54.89 %
Gradient Boosting	Legitimate	96.01 %
CNN	Legitimate	54.0 %
Overall estimation	**Legitimate**	

Εικόνα 5.9: Αποτέλεσμα της εφαρμογής για την legit login σελίδα του Facebook

Results for URL: https://faceboogk.filesusr.com/html/e971c7_8b35b2df874f14d55c6bab3bc4285eb1.html

Model	Outcome	Probability
KNN	Phishing	99.9 %
MLP	Phishing	100.0 %
Random Forest	Phishing	95.33 %
Gradient Boosting	Phishing	100.0 %
CNN	Phishing	97.65 %
Overall estimation	**Phishing**	

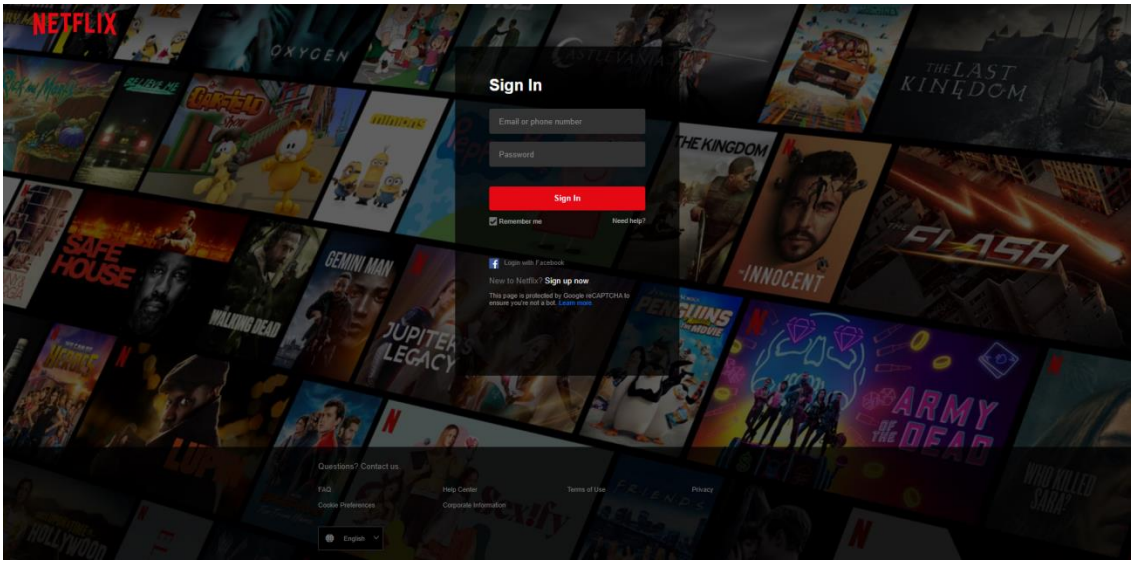
Εικόνα 5.10: Αποτέλεσμα της εφαρμογής για την phishing login σελίδα του Facebook

Την περίοδο της νόσου Covid19 οι εταιρείες διαδικτυακής ενοικίασης και παρακολούθησης (streaming) ταινιών και τηλεοπτικών σειρών γνώρισαν εκρηκτική αύξηση των συνδρομητών τους. Η δημοφιλέστερη εξ' αυτών το Netflix δε θα μπορούσε να μην αποτελέσει στόχο των phishers.

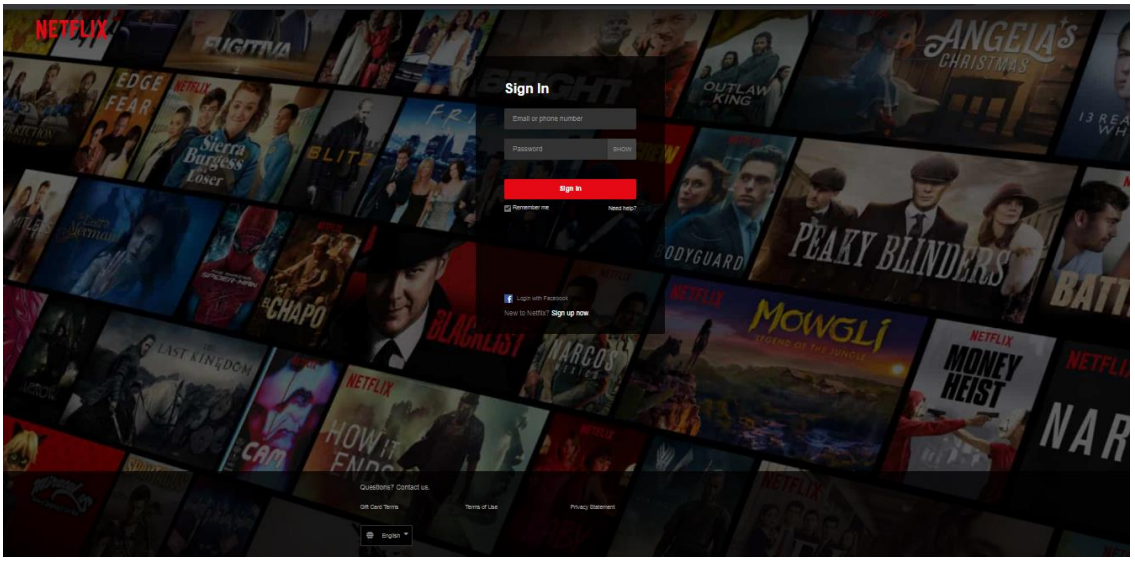
Το legit URL του Netflix για την login σελίδα του στην Ρωσία στα Αγγλικά είναι: https://www.netflix.com/ru-en/login
Αντίστοιχα, ένα phishing URL είναι: http://netflixgm5.temp.swtest.ru/N/login

Παρατηρώντας τις εικόνες 5.11 και 5.12, βλέπουμε ότι οι κυριότερες διαφορές είναι το διαφορετικό κολάζ από εξώφυλλα ταινιών στο background (παρασκήνιο της σελίδας) και οι διαφορετικές επιλογές στο υποσέλιδο της phishing σελίδας. Σε ότι αφορά το URL, το phishing URL έχει λίγους χαρακτήρες παραπάνω και στερείται του πρωτοκόλλου https.

Στην εικόνα 5.13 παρουσιάζονται τα αποτελέσματα από τις προβλέψεις που έδωσε η εφαρμογή μας, η οποία κατάφερε επιτυχώς να ταξινομήσει τις 2 ιστοσελίδες στην κατηγορία που ανήκουν.



Εικόνα 5.11: Legitimate login σελίδα του Netflix



Εικόνα 5.12: Phishing login σελίδα του Netflix

Results for URL: https://www.netflix.com/ru-en/login

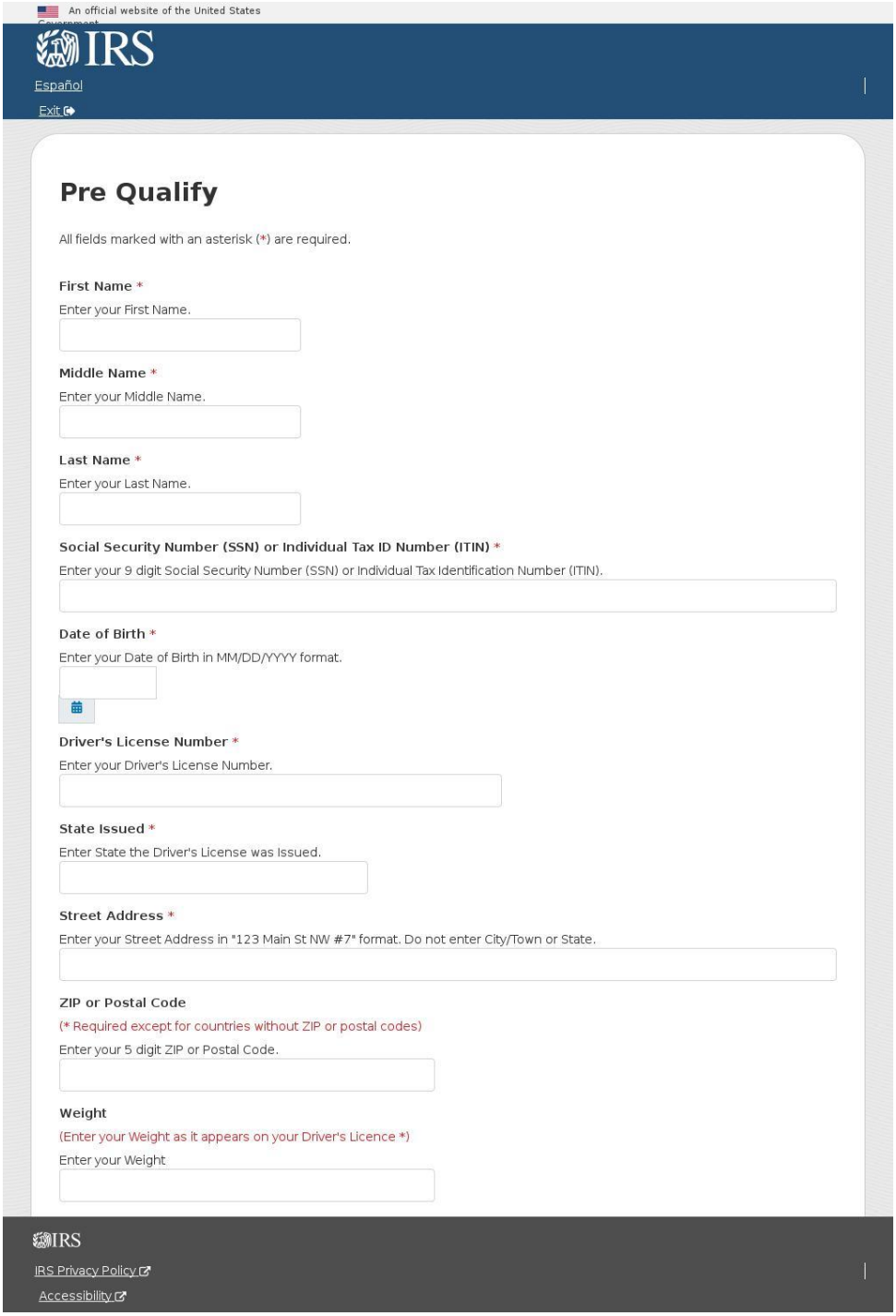
Model	Outcome	Probability
KNN	Legitimate	59.12 %
MLP	Legitimate	97.53 %
Random Forest	Legitimate	87.56 %
Gradient Boosting	Legitimate	100.0 %
CNN	Phishing	70.91 %
Overall estimation	**Legitimate**	

Results for URL: http://netflixgm5.temp.swtest.ru/N/login

Model	Outcome	Probability
KNN	Phishing	80.13 %
MLP	Phishing	99.91 %
Random Forest	Legitimate	59.78 %
Gradient Boosting	Phishing	82.53 %
CNN	Phishing	99.62 %
Overall estimation	**Phishing**	

Εικόνα 5.13: Τα αποτελέσματα της εφαρμογής για τα URLs του Netflix. Στην αριστερή στήλη είναι το legitimate URL και στη δεξιά το phishing URL.

Συχνά επίσης οι phishers μιμούνται κάποια δημόσια υπηρεσία ή οργανισμό δημιουργώντας ψεύτικες ιστοσελίδες για τις συναλλαγές των πολιτών με το κράτος. Μία από αυτές είναι η:
https://lrs-gov.org/coronavirus/notice/pre-qualify.html

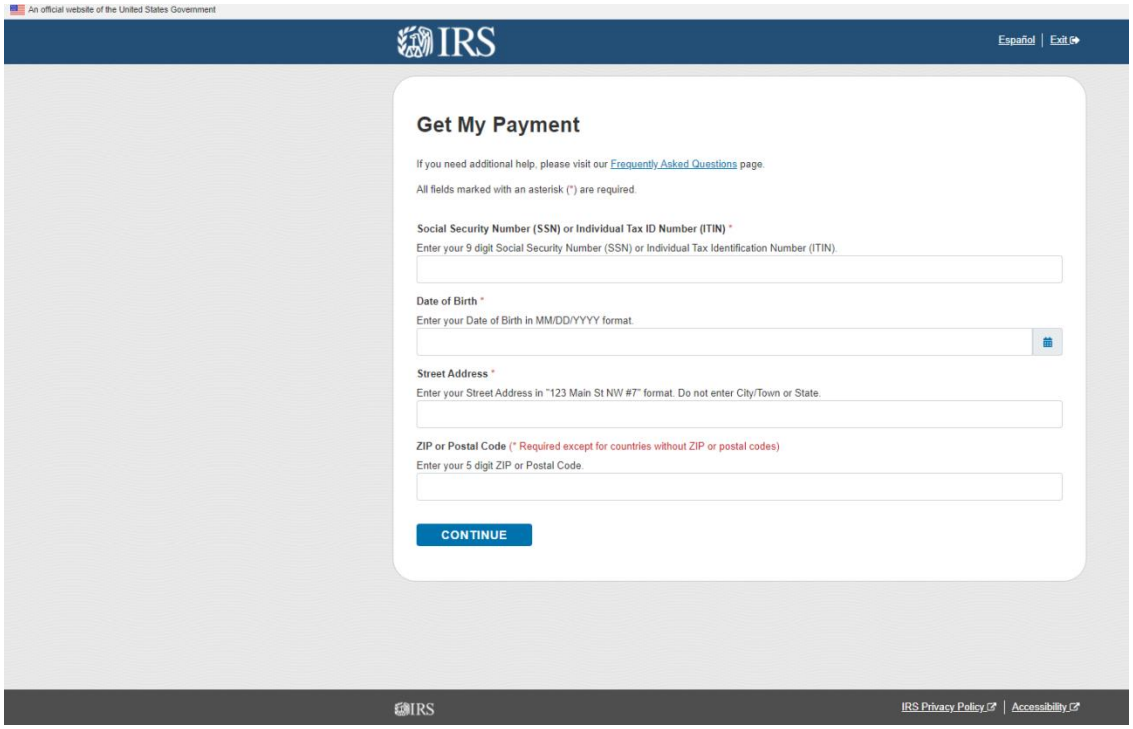


Εικόνα 5.14: Phishing ιστοσελίδα για την κρατική υπηρεσία IRS των ΗΠΑ

Η IRS[19] (Internal Revenue Service - Εσωτερική Υπηρεσία Εσόδων) είναι η δημόσια οικονομική υπηρεσία της ομοσπονδιακής κυβέρνησης των Ηνωμένων Πολιτειών της Αμερικής.

[19] https://www.irs.gov/

Η παραπάνω phishing ιστοσελίδα ζητάει προσωπικά στοιχεία πολιτών μιμούμενη την IRS αντιγράφοντας τη μορφή μιας σχετικής έγκυρης σελίδας της τελευταίας:
https://sa.www4.irs.gov/irfof-wmsp/login



Εικόνα 5.15: Legitimate ιστοσελίδα της κρατικής υπηρεσίας IRS των ΗΠΑ

Το κυριότερο λάθος στο phishing URL έγκειται στο γεγονός ότι αντί για "irs" αναγράφεται "lrs".

Στην εικόνα 5.16 παρουσιάζονται τα αποτελέσματα από τις προβλέψεις που έδωσε η εφαρμογή μας, η οποία κατάφερε επιτυχώς να ταξινομήσει τις 2 ιστοσελίδες στην κατηγορία που ανήκουν.

Results for URL: https://sa.www4.irs.gov/irfof-wmsp/login

Model	Outcome	Probability
KNN	Legitimate	59.25 %
MLP	Legitimate	53.64 %
Random Forest	Legitimate	91.11 %
Gradient Boosting	Legitimate	99.95 %
CNN	Legitimate	66.56 %
Overall estimation	**Legitimate**	

Results for URL: https://lrs-gov.org/coronavirus/notice/pre-qualify.html

Model	Outcome	Probability
KNN	Phishing	99.9 %
MLP	Phishing	99.7 %
Random Forest	Legitimate	67.56 %
Gradient Boosting	Legitimate	98.71 %
CNN	Legitimate	64.41 %
Overall estimation	**Phishing**	

Εικόνα 5.16: Τα αποτελέσματα της εφαρμογής για τα URLs της IRS. Στην αριστερή στήλη είναι το legitimate URL και στη δεξιά το phishing URL.

5.2 Μελλοντικές Κατευθύνσεις – Επεκτάσεις

Μερικές μελλοντικές κατευθύνσεις που θα επεκτείνουν τη μελέτη και την παρουσίαση των αποτελεσμάτων του φαινομένου του Phishing θα μπορούσαν να είναι οι ακόλουθες:

- Εύρεση περισσότερων active phishing ιστοσελίδων για διεύρυνση του dataset.
- Ενδεχομένως διεύρυνση στο πλήθος αλλά και στο είδος των features του dataset, προκειμένου να λαμβάνονται υπόψη και άλλα χαρακτηριστικά των phishing ιστοσελίδων (πχ javascript tags, whois ranking, google index).
- Μελέτη περισσότερων μοντέλων Μηχανικής Μάθησης και ειδικότερα συνδυασμών αλγορίθμων Βαθιάς Μάθησης.
- Παράλληλη προεπεξεργασία των URLs για να μειωθεί ο χρόνος εξαγωγής χαρακτηριστικών. Έτσι θα μειωθεί και ο χρόνος απόκρισης της web εφαρμογής μιας και ο περισσότερος χρόνος καταναλώνεται σε αυτό και όχι στο prediction των μοντέλων.
- Θα μπορούσε να δημιουργηθεί μια βάση δεδομένων, στην οποία θα αποθηκεύονταν τα URLs των phishing ιστοσελίδων μαζί με σημαντικές πληροφορίες όπως το πότε καταχωρήθηκαν ως ενεργές, το είδος τους καθώς και όλα τα δεδομένα που παίρνουμε για αυτά. Περιοδική ανανέωση της βάσης θα ήταν απαραίτητη ώστε να είναι πάντα up to date. Αυτό θα μπορούσε να εξασφαλιστεί σε μεγάλο βαθμό με τη δημιουργία ενός API, το οποίο θα γινόταν trigger (ενεργό) κάθε φορά που κάποιος χρήστης χρησιμοποιούσε την εφαρμογή για να ψάξει ένα URL και ανάλογα με το εάν το URL υπήρχε ήδη στη βάση, είτε θα το εισήγαγε ως νέα εγγραφή ή θα ανανέωνε την ήδη υπάρχουσα στην ημερομηνία που εμφανίστηκε τελευταία φορά ως active. Επίσης, ανά τακτά χρονικά διαστήματα θα συνέλεγε τα URLs που δεν έχουν αναζητηθεί καθόλου στο μεσοδιάστημα και θα ερευνούσε αν είναι active κι αν έχει αλλάξει κάτι στα χαρακτηριστικά τους ενημερώνοντας κατάλληλα την βάση. Ενδιαφέρουσα θα ήταν και μια κατηγοριοποίηση των phishing εγγραφών σε διάφορα είδη (Political, Economical, Social Media, National Security κ.ά).
- Δεδομένου ότι το δημοφιλέστερο πρόγραμμα περιήγησης στο διαδίκτυο είναι το Chrome (65% μερίδιο αγοράς τον Απρίλιο του 2021), θα μπορούσε να δημιουργηθεί ένα Chrome extension (επέκταση), το οποίο θα λειτουργούσε με τον ίδιο τρόπο με την διαδικτυακή εφαρμογή που δημιουργήσαμε με τη διαφορά ότι δε θα χρειαζόταν ο χρήστης να αντιγράφει κάθε φορά το URL που θέλει να εξετάσει και να το δίνει ως είσοδο στην εφαρμογή, αλλά αυτόματα καθώς «επισκέπτεται» το συγκεκριμένο URL θα «πετάγεται» μια pop-up notification (αναδυόμενη ειδοποίηση) σε περίπτωση που το URL αυτό εκτιμάται ότι είναι phishing.

Τέλος, σημαντικό ενδιαφέρον θα παρουσίαζε η εφαρμογή παρόμοιων μοντέλων πρόβλεψης σε θέματα που απασχολούν διαφορετικούς τομείς (πχ. οικονομικός).

Βιβλιογραφία

[1] Wikipedia, https://el.wikipedia.org/wiki/Phishing, "Αποτέλεσμα αναζήτησης Phishing"

[2] Blog Τράπεζας Πειραιώς, https://blog.piraeusbank.gr/el/all-articles/2021/03/ilektroniko-psarema-stoixeion-phishing-osa-prepei-na-kserete, "Όσα πρέπει να γνωρίζεται για το ηλεκτρονικό ψάρεμα στοιχείων"

[3] Maddie Rosenthal, https://www.tessian.com/blog/phishing-statistics-2020/, "Must-Know Phishing Statistics"

[4] CheckPoing Blog, https://blog.checkpoint.com/2021/01/14/brand-phishing-report-q4-2020/, "Brand Phishing Report – Q4 2020"

[5] Chris Taylor, https://blog.trendmicro.com/phishing-starts-inside/, "When Phishing Starts from the Inside"

[6] Jurgita Lapienytė, https://cybernews.com/news/leaker-says-they-are-offering-private-details-of-500-million-facebook-users/, "Facebook data leak"

[7] CyberNews Team, https://cybernews.com/news/stolen-data-of-500-million-linkedin-users-being-sold-online-2-million-leaked-as-proof-2/, "Sale of 500 million LinkedIn accounts with proof"

[8] Δημοσίευση στο Facebook της Τράπεζας Πειραιώς, https://www.facebook.com/winbankPage/posts/10158897806789929, "Facebook data leak"

[9] Δημοσίευση στο csc.com.gr για Μηχανική Μάθηση, https://www.csc.com.gr/machine-learning, "Μηχανική Μάθηση – Τί είναι;"

[10] Koza J. R., Bennett F. H., Andre D. & Keane M. A., "Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming", Artificial Intelligence in Design '96, pp. 151–170, 1996

[11] Tom M Mitchell, "Machine Learning", McGraw-Hill, 1997

[12] Fix Evelyn & Hodges Joseph L., "Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties", USAF School of Aviation Medicine, 1951

[13] Ανάρτηση στο tutorialspoint.com σχετικά με τον αλγόριθμο, kNN, https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_knn_algorithm_finding_nearest_neighbors.htm, "KNN Algorithm - Finding Nearest Neighbors"

[14] Haykin Simon, "Νευρωνικά Δίκτυα και Μηχανική Μάθηση", 3η έκδοση, εκδόσεις Παπασωτηρίου, 2010

[15] Ian Goodfellow, Yoshua Bengio & Aaron Courville, "Deep Learning", Chapter 6, MIT Press, 2016

[16] Avinash Navlani, https://www.datacamp.com/community/tutorials/random-forests-classifier-python#how, "Understanding Random Forests Classifiers in Python"

[17] Leo Breiman, "RANDOM FORESTS", 2001

[18] Leo Breiman, "Arcing Classifier (with Discussion and a Rejoinder by the Author)", Annals of Statistics, vol. 26, no. 3, pp. 801-849, 1998

[19] Rohith Gandhi, https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47, "Support Vector Machine — Introduction to Machine Learning Algorithms"

[20] Cortes C. & Vapnik V., "Support-vector networks", Mach Learn 20, pp. 273–297, 1995

[21] Ed Sperling, https://semiengineering.com/deep-learning-spreads/, "Deep LLearning Spreads"

[22] Y. Lecun et al., "Handwritten digit recognition with a back-propagation network." Morgan Kaufmann, 1990

[23] Stanford University Course 231n Notes, https://cs231n.github.io/neural-networks-1/, "Commonly used activation functions"

[24] Diederik P. Kingma & Jimmy Lei Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION", conference paper at ICLR, 2015

[25] Y. Lecun, L. Bottou, Y. Bengio & P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998

[26] M.V. Valueva, N.N. Nagornov, P.A. Lyakhov, G.V. Valuev & N.I. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation", Mathematics and Computers in Simulation, vol. 177, pp. 232-243, 2020

[27] Hamed Habibi Aghdam & Elnaz Jahani Heravi, "Guide to convolutional neural networks : a practical application to traffic-sign detection and classification", Springer, 2017

[28] Albawi Saad, Abed Mohammed Tareq & ALZAWI Saad., "Understanding of a Convolutional Neural Network", 2017

[29] Ciresan Dan, Ueli Meier, Jonathan Masci, Luca M. Gambardella & Jurgen Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification", Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume, vol. 2, pp. 1237–1242, 2011

[30] IBM Cloud Education, https://www.ibm.com/cloud/learn/convolutional-neural-networks, "Convolutional Neural Networks"

[31] Sergey Ioffe & Christian Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", Proceedings of the 32nd International Conference on Machine Learning, PMLR vol. 37, pp. 448-456, 2015

[32] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever & Ruslan Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting", Journal of Machine Learning Research 15, pp. 1929-1958, 2014

[33] IBM Cloud Education, https://www.ibm.com/cloud/learn/recurrent-neural-networks, "Recurrent Neural Networks"

[34] Sepp Hochreiter & Jurgen Schmidhuber, "LONG SHORT-TERM MEMORY", Neural Computation vol. 9, no. 8, pp.1735-1780, 1997

[35] Michael Phi, https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21, "Illustrated Guide to LSTM's and GRU's"

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin, "Attention Is All You Need", 2017

[37] Chen W., Wang X. A., Zhang W., & Xu C., "Phishing Detection Research Based on PSO-BP Neural Network", Lecture Notes on Data Engineering and Communications Technologies, pp. 990–998, 2018

[38] Kuncheva L., "Diversity in multiple classifier systems", Information Fusion, 2004

[39] Kim H., Kim H., Moon H., & Ahn H., "A weight-adjusted voting algorithm for ensembles of classifiers. Journal of the Korean Statistical Society", vol. 40, no. 4, pp. 437–449, 2011

[40] Rahman A. F. R., Alam H., & Fairhurst M. C., "Multiple Classifier Combination for Character Recognition: Revisiting the Majority Voting System and Its Variations", Document Analysis Systems V, pp. 167–178, 2002

[41] Daniel Berend & Aryeh Kontorovich, "Consistency of weighted majority votes", 2018

[42] De Stefano C., Fontanella F., & Scotto di Freca A., "A Novel Naive Bayes Voting Strategy for Combining Classifiers", 2012 International Conference on Frontiers in Handwriting Recognition, 2012

[43] EbubekirBbr Github Page, https://github.com/ebubekirbbr/phishing_url_detection/tree/master/dataset, "Dataset of legit and phishing URLs"

[44] Mitchell Krog Github Page, https://github.com/mitchellkrogza/Phishing.Database, "Phishing URLs Database"

[45] Phishtank, http://phishtank.org/phish_search.php?valid=y&active=y&Search=Search, "Αναζήτηση Phishing URLs στην βάση του PhishTank"

[46] Grega Vrbancic, Iztok FisterJr. & Vili Podgorelec, "Datasets for Phishing Websites Detection", Data in Brief, 2020

[47] Rami M. Mohammad, Fadi Thabtah & Lee McCluskey, "Phishing Websites Features", 2015

[48] Sing Tat, https://medium.com/swlh/supervised-learning-to-detect-phishing-urls-d0779d360dc8, "Supervised Learning to detect Phishing URLs"

[49] SPAMHAUS, https://www.spamhaus.org/statistics/tlds/, "List of top 10 most abused TLDs"

[50] AMAN1608, https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/, "Feature Selection Techniques in Machine Learning"

[51] Andrzej Maćkiewicz & Waldemar Ratajczak, "Principal components analysis (PCA)", Computers & Geosciences, vol. 9 issue 3, pp. 303-342, 1993

[52] Yerima Suleiman & Alzaylaee Mohammed, "High Accuracy Phishing Detection Based on Convolutional Neural Networks", Third International Conference on Computer Applications & Information Security, 2020

[53] Chen W., Zhang W., & Su Y., "Phishing Detection Research Based on LSTM Recurrent Neural Network", Data Science, pp. 638–645, 2018

[54] Shuai Li, Wanqing Li, Chris Cook, Ce Zhu & Yanbo Gao, "Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN", 2018

[55] M. Mohandes, M. Deriche & S. O. Aliyu, "Classifiers Combination Techniques: A Comprehensive Review", IEEE Access, vol. 6, pp. 19626-19639, 2018

[56] Kuncheva L. I. & Rodríguez J. J., "A weighted voting framework for classifiers ensembles. Knowledge and Information Systems", vol. 38, no. 2, pp. 259–275, 2012

[57] Himanshi Mathur, Vanshika Goel & Vibhu Agrawal, "WhatAPhish: Detecting Phishing Website", 2020

Παράρτημα Α

Ευρετήριο Όρων και Συντμήσεων

Α.1 Ελληνικοί όροι

ΗΠΑ: Ηνωμένες Πολιτείες Αμερικής

Α.2 Αγγλικοί όροι

COVID19: Coronavirus Disease 2019
AOL: America Online
URL: Uniform Resource Locator
FBI: Federal Bureau of Investigation
kNN: k Nearest Neighbors
MLP: Multi Layer Perceptron
LSTM: Long Short-Term Memory
ReLU: Rectifier Linear Unit
CART: Classification And Regression Tree
MDI: Mean Decrease in Impurity
SVM: Support Vector Machines
ELU: Exponential Linear Unit
SGD: Stochastic Gradient Descent
ADAM: Adaptive Moment Estimation
CNN: Convolutional Neural Networks
ConvNet: Convolutional Network
RNN: Recurrent Neural Network
NLP: Natural Language Processing
TP: True Positive
TN: True Negative
FP: False Positive
FN: False Negative
FNR: False Negative Ratio
opt: optimal
IP: Internet Protocol
HTTPS: Hypertext Transfer Protocol Secure
SSL: Secure Sockets Layer
HTTP: Hypertext Transfer Protocol
DNS: Domain Name System
HTML: HyperText Markup Language
TLD: Top Level Domain
t-SNE: t-distributed Stochastic Neighbor Embedding
PCA: Principle Component Analysis
CV: Cross Validation
svd: singular value decomposition
tol: tolerance
oob: out-of-bag
tanh: hyperbolic tangent

lr: learning rate
Ind-RNN: Independently Recurrent Neural Network
CAPTCHA: Completely Automated Public Turing test
UI: User Interface
JS: JavaScript
PaaS: Platform as a Service
IoT: Internet of Things
App: Application
AWS: Amazon Web Services
IaaS: Infrastructure as a Service
VM: Virtual Machine
API: Application Programming Interface
IRS: Internal Revenue Service