Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης
NETMODE (Network Management & Optimal Design Laboratory)

Χρήση eXplainable Artificial Intelligence (XAI) για την Επεξήγηση Ταξινομητών Ανίχνευσης Κίνησης από Domain Generation Algorithms (DGA)

Μελέτη και υλοποίηση

Διπλωματικη Εργασια

της

ΜΥΡΣΙΝΗΣ ΦΙΛΙΠΠΑ

Επιβλέποντες: Συμεών Παπαβασιλείου, Καθηγητής Ε.Μ.Π.
Βασίλειος Μάγκλαρης, Ομότιμος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης
NETMODE (Network Management & Optimal Design Laboratory)

Χρήση eXplainable Artificial Intelligence (XAI) για την Επεξήγηση Ταξινομητών Ανίχνευσης Κίνησης από Domain Generation Algorithms (DGA)

Μελέτη και υλοποίηση

Διπλωματικη Εργασια

της

ΜΥΡΣΙΝΗΣ ΦΙΛΙΠΠΑ

Επιβλέποντες: Συμεών Παπαβασιλείου, Καθηγητής Ε.Μ.Π.
Βασίλειος Μάγκλαρης, Ομότιμος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 18 Ιουλίου 2024

(Υπογραφή)	*(Υπογραφή)*	*(Υπογραφή)*
.....	.....	.....
Συμεών Παπαβασιλείου	Ελένη Στάη	Γεώργιος Στάμου
Καθηγητής Ε.Μ.Π.	Επίκουρη Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2024

Εθνικο Μετσοβιο Πολυτεχνειο
Σχολη Ηλεκτρολογων Μηχανικων και Μηχανικων Υπολογιστων
Τομεας Επικοινωνιων, Ηλεκτρονικης και Συστηματων Πληροφορικης
NETMODE (Network Management & Optimal Design Laboratory)

ΔΗΛΩΣΗ ΜΗ ΛΟΓΟΚΛΟΠΗΣ ΚΑΙ ΑΝΑΛΗΨΗΣ ΠΡΟΣΩΠΙΚΗΣ ΕΥΘΥΝΗΣ

Με πλήρη επίγνωση των συνεπειών του νόμου περί πνευματικών δικαιωμάτων, δηλώνω ενυπογράφως ότι είμαι αποκλειστικός συγγραφέας της παρούσας Πτυχιακής Εργασίας, για την ολοκλήρωση της οποίας κάθε βοήθεια είναι πλήρως αναγνωρισμένη και αναφέρεται λεπτομερώς στην εργασία αυτή. Έχω αναφέρει πλήρως και με σαφείς αναφορές, όλες τις πηγές χρήσης δεδομένων, απόψεων, θέσεων και προτάσεων, ιδεών και λεκτικών αναφορών, είτε κατά κυριολεξία είτε βάσει επιστημονικής παράφρασης. Αναλαμβάνω την προσωπική και ατομική ευθύνη ότι σε περίπτωση αποτυχίας στην υλοποίηση των ανωτέρω δηλωθέντων στοιχείων, είμαι υπόλογος έναντι λογοκλοπής, γεγονός που σημαίνει αποτυχία στην Πτυχιακή μου Εργασία και κατά συνέπεια αποτυχία απόκτησης του Τίτλου Σπουδών, πέραν των λοιπών συνεπειών του νόμου περί πνευματικών δικαιωμάτων. Δηλώνω, συνεπώς, ότι αυτή η Πτυχιακή Εργασία προετοιμάστηκε και ολοκληρώθηκε από εμένα προσωπικά και αποκλειστικά και ότι, αναλαμβάνω πλήρως όλες τις συνέπειες του νόμου στην περίπτωση κατά την οποία αποδειχθεί, διαχρονικά, ότι η εργασία αυτή ή τμήμα της δεν μου ανήκει διότι είναι προϊόν λογοκλοπής άλλης πνευματικής ιδιοκτησίας.

(Υπογραφή)

.....
Μυρσίνη Φίλιππα
Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών

18 Ιουλίου 2024

Περίληψη

Τα τελευταία χρόνια οι κυβερνοεπιθέσεις έχουν σημειώσει ραγδαία αύξηση και γίνονται ολοένα πιο ισχυρές και σύνθετες, με τα botnets να αποτελούν τη βάση της πλειοψηφίας αυτών. Οι σύγχρονες υλοποιήσεις botnets βασίζονται στους Αλγορίθμους Παραγωγής Ονομάτων (Domain Generation Algorithms - DGAs) για την απόκρυψη της ταυτότητας των Command & Control (C&C) servers με σκοπό να γίνει δυσκολότερη η εξάρθρωση τους. Τα bots και οι C&C servers εκτελούν περιοδικά τον αλγόριθμο με ένα κοινό seed γνωστό μόνο σε αυτούς και δημιουργούν ένα σύνολο ονομάτων εκ των οποίων μικρό υποσύνολο εκχωρείται στις διευθύνσεις IP των C&C servers μέσω του Domain Name System - DNS. Τα bots εκτελούν DNS queries μέχρι να λάβουν ως απάντηση μία διεύθυνση IP για κάποιο από τα καταχωρημένα ονόματα και να εδραιώσουν επικοινωνία με κάποιον C&C server. Η περιοδική αλλαγή των ονομάτων που εκχωρούνται στους C&C servers καθιστά τον εντοπισμό τους από παραδοσιακά συστήματα ασφαλείας, όπως το blacklisting αναποτελεσματικά, καθώς μετά από ένα μικρό χρονικό διάστημα τα ονόματα αυτά αποσύρονται και δεν επαναχρησιμοποιούνται. Οι υλοποιήσεις με μεθόδους Μηχανικής Μάθησης (Machine Learning) για τον εντοπισμό τέτοιων ονομάτων αποτελούν, πλέον, μία από τις δημοφιλέστερες προσεγγίσεις καθώς προσφέρουν καλή απόδοση και ανίχνευση σε πραγματικό χρόνο, οι οποίες όμως παραμένουν μη ερμηνεύσιμες (δεν κατανοούμε τον τρόπο με τον οποίο παίρνουν αποφάσεις), με αποτέλεσμα να αντιμετωπίζονται με επιφυλακτικότητα από τους διαχειριστές δικτύων.

Στην παρούσα διπλωματική εργασία, παρουσιάζουμε δύο Random Forest ταξινομητές, έναν δύο κλάσεων binary, που κατηγοριοποιεί τα ονόματα σε καλόβουλα και κακόβουλα (παραγόμενα από DGA) και έναν πολλών κλάσεων multiclass, που κατηγοριοποιεί τα ονόματα σε καλόβουλα και 54 διαφορετικές οικογένειες DGA. Για την εκπαίδευση και αξιολόγηση των ταξινομητών χρησιμοποιήσαμε δημοφιλή σύνολα δεδομένων, συγκεκριμένα τα καλόβουλα ονόματα επιλέχθηκαν από τη λίστα Tranco, ενώ τα κακόβουλα από το DGArchive. Χρησιμοποιήσαμε μεθόδους eXplainable Artificial Intelligence (XAI) για την αποτίμηση της επίδρασης των χαρακτηριστικών (features) στις αποφάσεις των δύο ταξινομητών. Για το σκοπό αυτό, χρησιμοποιήσαμε τις οπτικοποιήσεις που προσφέρει η XAI μέθοδος SHapley Additive exPlanations (SHAP). Επιπλέον, με αφορμή τη πληροφορία για τη διάρκεια ζωής των ονομάτων που παρέχει το DGAchive, εκτιμήσαμε πως μεταβάλλεται η απόδοση των δύο ταξινομητών με την εμφάνιση νέων οικογενειών DGA και κακόβουλων ονομάτων με τη πάροδο του χρόνου (το χρονικό διάστημα για το οποίο είχαμε δεδομένα είναι τα έτη 2010 έως 2019), εκπαιδεύοντας τους δύο ταξινομητές με ονόματα του έτους 2010 και κατόπιν αξιολογώντας το με δεδομένα των ακόλουθων ετών (2011-2019). Στόχος μας, ήταν η σύγκριση των δύο ταξινομητών ως προς την απόδοση και τις ερμηνείες τους.

Λέξεις Κλειδιά

Ασφάλεια Δικτύων, Σύστημα Ονοματοδοσίας Τομέων (DNS), Domain Generation Algorithm (DGA), Μηχανική Μάθηση, Δέντρα Αποφάσεων, Τυχαίο Δάσος, eXplainable Artificial Intelligence (XAI), SHapley Additive exPlanations (SHAP)

Abstract

In recent years, cyberattacks have significantly increased and become progressively more powerful and complex, with botnets forming the foundation of the majority of these attacks. Modern botnet implementations rely on Domain Generation Algorithms (DGAs) to hide the identities of Command & Control (C&C) servers, making them harder to detect. Both bots and C&C servers periodically run the algorithm with a shared seed known only to them, generating a set of names from which a small subset is assigned to the IP addresses of C&C servers via the Domain Name System (DNS). The bots execute DNS queries until they receive an IP address for one of the registered names and establish communication with a C&C server. The periodic change of names assigned to C&C servers renders traditional security systems like blacklisting ineffective, as these names are withdrawn after a short period and not reused. Machine Learning (ML) implementations for detecting such names have become one of the most popular approaches as they have good performance and also offer real-time detection, though they remain uninterpretable (the way they make decisions is not understood), leading to skepticism from network administrators.

In this thesis, we present two Random Forest classifiers, a binary classifier that categorizes domain names as benign or malicious (DGA-generated), and a multiclass classifier that categorizes domain names as benign or into 54 different DGA families. For training and evaluating the classifiers, we used popular datasets, specifically benign names were selected from the Tranco list and malicious names from DGArchive. We employed Explainable Artificial Intelligence (XAI) methods to assess the impact of features on the decisions of the two classifiers. To this end, we used visualizations provided by the SHapley Additive exPlanations (SHAP) XAI method. Additionally, leveraging the information about the lifespan of the domain names provided by DGArchive, we estimated how the performance of the two classifiers changes with the emergence of new DGA families and malicious names over time (the time period for which we had data is the years 2010 to 2019). We trained the two classifiers with names from the year 2010 and then evaluated them with data from the following years (2011-2019). Our goal was to compare the performance and interpretability of the two classifiers.

Keywords

Cybersecurity, Domain Name System (DNS), Domain Generation Algorithm (DGA), Machine Learning, Decision Trees, Random Forest, eXplainable Artificial Intelligence (XAI),

SHapley Additive exPlanations (SHAP)

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον Ομότιμο Καθηγητή ΕΜΠ κ. Βασίλειο Μάγκλαρη για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο NETMODE.

Επίσης, θα ήθελα να ευχαριστήσω ιδιαιτέρως τον Διδάκτωρ Νίκο Κωστόπουλο για την καθοδήγησή του και την υποστήριξη του όλους αυτούς τους μήνες.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου, τον Τάσο, τους φίλους μου και όλη μου την οικογένεια για την υποστήριξη τους όλα αυτά τα χρόνια.

Αθήνα, Ιούλιος 2024

Μυρσίνη Φίλιππα

Περιεχόμενα

Κατάλογος Σχημάτων

Κεφάλαιο 1

Εισαγωγή

Στην εποχή μας, η τεχνολογία και το Διαδίκτυο αποτελούν αναπόσπαστο κομμάτι της καθημερινότητας μας. Η συντριπτική πλειοψηφία του παγκόσμιου πληθυσμού έχει πλέον πρόσβαση στο Διαδίκτυο. Η μαζική αυτή χρήση έχει δημιουργήσει μία τεράστια βιομηχανία, όπου μέρα με τη μέρα αναπτύσσει ολοένα και περισσότερες ανταγωνιστικές υπηρεσίες. Αυτή όμως η μαζική χρήση των υπηρεσιών, προσελκύει επίσης κακόβουλους χρήστες που βρίσκουν την ευκαιρία να εκμεταλλευτούν την ελεύθερη και αφιλτράριστη διακίνηση πληροφορίας προς όφελος τους. Τα τελευταία χρόνια, οι κακόβουλες δραστηριότητες, όπως η υποκλοπή προσωπικών δεδομένων (π.χ. τραπεζικοί κωδικοί) ή οι κατανεμημένες επιθέσεις άρνησης υπηρεσιών (Distributed Denial of Service Attacks), έχουν αυξηθεί ραγδαία. Έτσι, οι ερευνητές ασφαλείας καλούνται να αναπτύξουν νέες αποδοτικότερες μεθόδους για την ανίχνευση και εξουδετέρωση επιθέσεων που γίνονται ολοένα πιο ισχυρές και σύνθετες.

Το Σύστημα Ονοματοδοσίας Τομέων (Domain Name System - DNS) αποτελεί ένα ιεραρχικό και κατανεμημένο σύστημα ονοματοδοσίας το οποίο είναι απαραίτητο για την ορθή και ομαλή λειτουργία του Διαδικτύου, καθώς αντιστοιχίζει ονόματα (π.χ. ιστοσελίδων) με δικτυακές πληροφορίες (π.χ. διευθύνσεις IP). Λόγω της καθολικής αποδοχής και χρήσης της υπηρεσίας, οι πολιτικές που εφαρμόζονται στα firewalls των δικτύων δεν μπλοκάρουν μηνύματα DNS. Το γεγονός αυτό σε συνδυασμό με την ευκολία εκχώρησης ονομάτων εκμεταλλεύονται οι επιτιθέμενοι στις σύγχρονες υλοποιήσεις botnet (δίκτυα μολυσμένων συσκευών που χρησιμοποιούνται σε πληθώρα επιθέσεων). Με τη βοήθεια των Αλγορίθμων Παραγωγής Ονομάτων (Domain Generation Algorithms - DGAs) παράγουν περιοδικά πληθώρα ονομάτων και ένα μικρό υποσύνολο αυτών εκχωρείται από τους Command & Control (C&C) servers για να μπορεί να εδραιωθεί επικοινωνία για τον έλεγχο των bots του δικτύου. Αυτή η συνεχής εναλλαγή ονομάτων, καθιστά ιδιαίτερα δύσκολο τον εντοπισμό των C&C servers και άρα και την εξάρθρωση των botnets, καθώς απαιτείται γνώση του συνόλου των ονομάτων που παράγονται από τον αλγόριθμο αυτό ή αντίστοιχα του αλγόριθμου του ίδιου (Reverse Engineering), το οποίο πρόκειται για πολύ δύσκολη και χρονοβόρα διαδικασία.

Στοχεύοντας, λοιπόν, στην επίλυση του παραπάνω προβλήματος, οι ερευνητές ασφαλείας έστρεψαν τις προσπάθειες τους σε υλοποιήσεις με μεθόδους Μηχανικής Μάθησης (Machine Learning - ML). Οι αλγόριθμοι ML χρησιμοποιούνται συχνά σε προβλήματα ανίχνευσης ανωμαλιών (anomaly detection) και άρα τους καθιστά κατάλληλους για την ανίχνευση αλ-

γοριθμικά παραγόμενων ονομάτων. Συνήθως, οι υλοποιήσεις αυτές βασίζονται σε στατιστικά χαρακτηριστικά που προκύπτουν άμεσα από το domain name που είναι εύκολο και γρήγορο να υπολογιστούν και επιτυγχάνουν καλή απόδοση και ανίχνευση σε πραγματικό χρόνο (real-time detection). Τα μοντέλα αυτά, όμως, λειτουργούν ως μαύρα κουτιά (black-box models), δηλαδή δεν γνωρίζουμε με ποιο τρόπο παίρνουν αποφάσεις, το οποίο δημιουργεί επιφυλακτικότητα στην αφομοίωση και χρήση από τα συστήματα ασφαλείας των δικτύων. Για το λόγο αυτό, δημιουργείται η ανάγκη για επεξηγησιμότητα των μοντέλων αυτών, το οποίο θα βοηθήσει στην κατανόηση και αποδοχή των αποφάσεων από τους διαχειριστές των δικτύων, στο debugging και τη βελτίωση των μοντέλων από τους developers, αλλά και στην επιβεβαίωση συμμόρφωσης με κανονισμούς νομικών οντοτήτων (π.χ. συμμόρφωση με το γενικό κανονισμό για την προστασία δεδομένων - GDPR).

1.1 Αντικείμενο Διπλωματικής

Στην παρούσα διπλωματική εργασία, στόχος είναι η εκπαίδευση και αξιολόγηση ταξινομητών για την ανίχνευση κακόβουλων δραστηριοτήτων που σχετίζονται με botnets και τη χρήση Domain Generation Algorithms (DGA), αξιοποιώντας δεδομένα DNS (Domain Name System), με έμφαση στη χρήση τεχνικών Μηχανικής Μάθησης και Επεξηγήσιμης Τεχνητής Νοημοσύνης (eXplainable Artificial Intelligence - XAI). Ειδικότερα, υλοποιήθηκαν δύο ταξινομητές, ένας δύο κλάσεων (binary), όπου κατηγοριοποιεί τα ονόματα σε κακόβουλα και καλόβουλα και ένας πολλών (multiclass), που κατηγοριοποιεί τα ονόματα σε καλόβουλα και 54 διαφορετικές οικογένειες DGA. Οι ταξινομητές είναι βασισμένοι στον αλγόριθμο Random Forest (αλγόριθμος που βασίζεται σε δέντρα αποφάσεων) και η εκπαίδευση και αξιολόγηση τους έγινε σε δημοφιλή σύνολα δεδομένων (Tranco [3] για τα καλόβουλα ονόματα και DGArchive [4] για τα κακόβουλα). Στη συνέχεια, χρησιμοποιήσαμε τη μέθοδο SHAP (SHapley Additive exPlanations), η οποία αποτελεί μία από τις δημοφιλέστερες μεθόδους επεξηγηματικής τεχνητής νοημοσύνης (eXplainable Artificial Intelligence - XAI) για την αποτίμηση της επίδρασης των χαρακτηριστικών (features) στις αποφάσεις των δύο ταξινομητών. Τέλος, με αφορμή τη πληροφορία για τη διάρκεια ζωής των ονομάτων που παρέχει το DGAchive, εκτιμήσαμε πώς μεταβάλλεται η απόδοση των δύο ταξινομητών με την εμφάνιση νέων οικογενειών DGA και κακόβουλων ονομάτων με τη πάροδο του χρόνου (το χρονικό διάστημα για το οποίο είχαμε δεδομένα είναι τα έτη 2010 έως 2019). Τα παραπάνω, διεξήχθησαν με σκοπό τη σύγκριση τόσο της απόδοσης όσο και των ερμηνειών των δύο ταξινομητών.

1.2 Οργάνωση Κειμένου

Η συνέχεια της εργασίας οργανώνεται στα ακόλουθα 4 Κεφάλαια και τη Βιβλιογραφία:

- **Κεφάλαιο 2**: Αφορά το θεωρητικό υπόβαθρο της παρούσας διπλωματικής. Παρουσιάζονται και αναλύονται έννοιες σχετικές με το αντικείμενο μελέτης, όπως Botnets, DGA, DNS και XAI.
- **Κεφάλαιο 3**: Περιγράφεται τη πειραματική διαδικασία, από την επιλογή και την προεπεξεργασία των δεδομένων, μέχρι την εκπαίδευση και αξιολόγηση των μοντέλων

και την δημιουργία διαγραμμάτων με τη βοήθεια της SHAP για την ερμηνεία των μοντέλων.

- **Κεφάλαιο 4**: Παρουσιάζονται, αναλύονται και αξιολογούνται τα αποτελέσματα των πειραμάτων που εκτελέσαμε.
- **Κεφάλαιο 5**: Σύνοψη της διαδικασίας και των συμπερασμάτων που προκύψανε. Επίσης, αναφέρονται οι ενδεχόμενες μελλοντικές επεκτάσεις και στόχοι της παρούσας υλοποίησης.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό, παρουσιάζεται και εξηγείται αναλυτικά το θεωρητικό υπόβαθρο, το οποίο είναι αναγκαίο για την κατανόηση του αντικειμένου που μελετάται στα πλαίσια της παρούσας διπλωματικής.

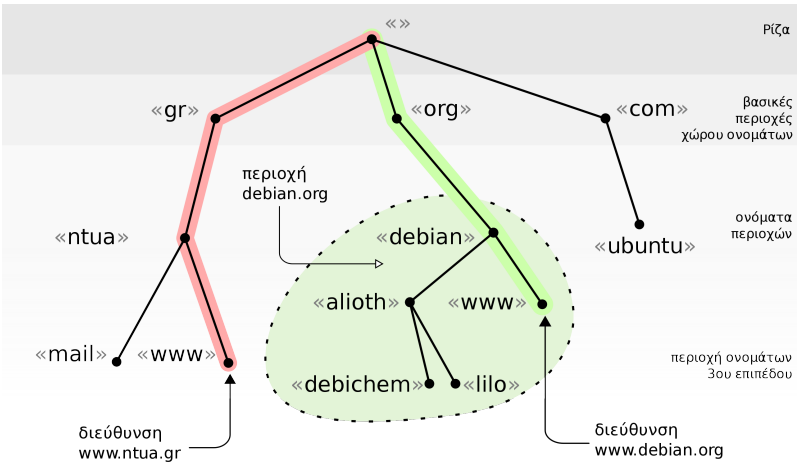
2.1 Σύστημα Ονοματοδοσίας Τομέων - Domain Name System (DNS)

Το Domain Name System [5] αποτελεί στην ουσία τον «τηλεφωνικό κατάλογο» του Internet. Όταν θέλουμε να πλοηγηθούμε σε έναν ιστότοπο πληκτρολογούμε το domain name του ιστότοπου που επιθυμούμε στον web browser, όπως για παράδειγμα www.google.com. Στην πραγματικότητα, όμως, αυτό που απαιτείται για την πρόσβαση μας στον ιστότοπο είναι η διεύθυνση IP του. Ο ρόλος, λοιπόν, του DNS είναι η αντιστοίχιση ενός hostname (για παράδειγμα www.example.com) σε μία διεύθυνση IP (φιλική για τον υπολογιστή, για παράδειγμα 192.168.1.4). Η καθολικότητα της αποδοχής και της χρήσης της υπηρεσίας αυτής την καθιστά απαραίτητη για τη λειτουργία του Διαδικτύου και για το λόγο αυτό η κίνηση DNS δεν φιλτράρεται αυστηρά από τα συστήματα ασφαλείας. .

Η διαδικασία αντιστοίχησης ενός hostname με μία διεύθυνση IP που αναφέρθηκε προηγουμένως, λειτουργεί αναφορικά ως εξής: Ένας χρήστης πληκτρολογεί ένα domain name π.χ. «example.com» για να αποκτήσει πρόσβαση σε μία ιστοσελίδα. Στη συνέχεια το ερώτημα αυτό λαμβάνεται από έναν αναδρομικό επιλυτή DNS (Recursive DNS server), ο οποίος με τη σειρά του υποβάλλει το ερώτημα σε έναν διακομιστή ρίζας (DNS root nameserver). Ο root server επιστρέφει στον resolver τη διεύθυνση του Top Level Domain (TLD) DNS server στον οποίο ανήκει το domain για το οποίο έγινε το ερώτημα. Στην περίπτωση μας, επιστρέφεται ο «.com» TLD nameserver, ο οποίος απαντάει στον resolver με την IP διεύθυνση του domain name που ζητήθηκε αρχικά. Τέλος, η διεύθυνση αποστέλλεται στο χρήστη και η επιθυμητή σελίδα φορτώνεται.

Στο Σχήμα 2.1, μπορούμε να εύκολα να οπτικοποιήσουμε την ιεραρχία των μερών/ετικετών τα οποία απαρτίζουν ένα domain name και την οποία ακολουθούν και οι διακομιστές που αναφέρθηκαν στην παραπάνω διαδικασία για την επίλυση του ονόματος τομέα. Τα Top Level Domains που μπορούμε να δούμε στο Σχήμα είναι τα «gr», «org» και «com». Καλό

θα ήταν σε αυτό το σημείο να αναφέρουμε τις ζώνες (zones). Μία ζώνη αποτελεί ένα υποσύνολο του δέντρου DNS, για την οποία είναι υπεύθυνος ένας DNS server να απαντάει για τις εγγραφές που περιλαμβάνει. Οι servers αυτοί ονομάζονται authoritative DNS servers. διαφορετικά



Σχήμα 2.1: *Ιεραρχία DNS*

2.2 Δίκτυα Μολυσμένων Υπολογιστών Botnet

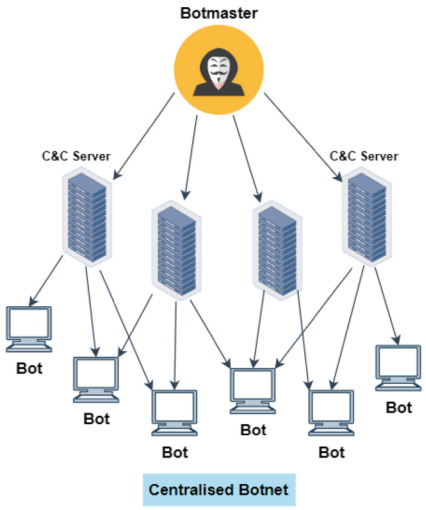
2.2.1 Η δομή ενός Botnet

Ως botnet [6, 7] ορίζεται ένα δίκτυο από συσκευές, οι οποίες έχουν μολυνθεί από κάποιο κακόβουλο λογισμικό (malware). Οι συσκευές αυτές ονομάζονται bots και με τη βοήθεια του malware επιτυγχάνεται ο απομακρυσμένος έλεγχος τους από τον κακόβουλο διαχειριστή (botmaster) του botnet.

Ο έλεγχος των bots επιτυγχάνεται μέσω των Command and Control (C&C) servers. Ο διαχειριστής δίνει εντολή στα bots, με κατανεμημένο τρόπο, μέσω των C&C server με σκοπό την πραγματοποίηση μίας συντονισμένης επίθεσης, η οποία ταυτόχρονα παρέχει ανωνυμία για τον botmaster. Ένα botnet μπορεί να αποτελείται από μόλις μερικές χιλιάδες έως και εκατομμύρια bots [8] το οποίο καθιστά την επίθεση ιδιαίτερα ισχυρή.

Μερικές από τις κυριότερες χρήσεις των botnet είναι οι εξής [9]:

- **Επιθέσεις κατανεμημένης άρνησης παροχής υπηρεσιών (Distributed Denial of Service - DDoS)**: Κατακλυσμός ενός ενός server (π.χ. που φιλοξενεί έναν ιστότοπο) με τεράστιο όγκο κίνησης με σκοπό να καταστεί μη διαθέσιμος για τους χρήστες.
- **Spamming**: Μαζική αποστολή spam μηνυμάτων ηλεκτρονικού ταχυδρομείου (email), τα οποία μπορεί να περιέχουν απόπειρες για phising ή αρχεία κακόβουλου λογισμικού.



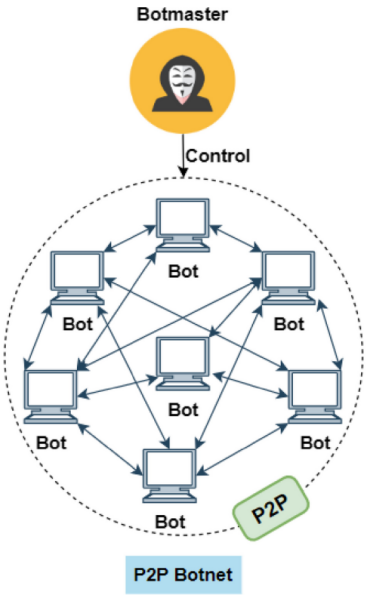
Σχήμα 2.2: *Η δομή ενός botnet*

- **Υποκλοπή πληροφοριών**: όπως αριθμούς πιστωτικών καρτών, κωδικούς πρόσβασης, προσωπικά δεδομένα κλπ
- **Cryptojacking**: Μόλυνση συσκευών με κακόβουλο λογισμικό με σκοπό τη χρήση των υπολογιστικών πόρων για εξόρυξη κρυπτονομισμάτων.

2.2.2 DGA-based Botnets

Όπως αναφέρθηκε παραπάνω και όπως μπορούμε να δούμε και στο Σχήμα 2.2 τα botnets βασίζουν τη λειτουργία τους σε έναν ή περισσότερους C&C servers με τους οποίους τα bots εγκαθιστούν έναν επικοινωνιακό δίαυλο. Οι μηχανισμοί επικοινωνίας που χρησιμοποιούνται είναι βασισμένοι σε πρωτόκολλα, όπως Internet Relay Chat (IRC) και Hypertext Transfer Protocol (HTTP). Οι κεντροποιημένες αυτές δομές έχουν το μειονέκτημα του μεμονωμένου σημείου αποτυχίας (single point of failure), καθώς ο εντοπισμός και το μπλοκάρισμα του C&C server αρκούν για να χάσει ο botmaster τον έλεγχο του botnet. Αυτό το πρόβλημα προσπάθησαν οι εν δυνάμει επιτιθέμενοι να επιλύσουν με τοπολογίες Peer to Peer (P2P), όπου τα bots εναλλάσσονται στον ρόλο του C&C server και του bot όπως μπορούμε να δούμε και στο Σχήμα 2.3. Η τοπολογία αυτή προσφέρει μεγάλη ανθεκτικότητα, αλλά ταυτόχρονα είναι ιδιαιτέρως πολύπλοκα, με αποτέλεσμα να μη χρησιμοποιούνται συχνά [10].

Στο πρόβλημα της έλλειψης στιβαρότητας των κεντροποιημένων τοπολογιών όπου οι C&C server έχουν σταθερές διευθύνσεις IP και της πολυπλοκότητας των P2P τοπολογιών, έρχονται να δώσουν λύση τα DGA-based botnets. Στη δομή αυτή, κάθε bot εκτελεί έναν αλγόριθμο παραγωγής ονομάτων (Domain Generation Algorithm - DGA), τα οποία στη συνέχεια προσπαθεί να επιλύσει εκτελώντας DNS queries έως ότου να επιλυθεί η IP του C&C server και να εδραιωθεί η επικοινωνία τους. Εκμεταλλεύονται με αυτό τον τρόπο την καθολική αποδοχή και χρήση της υπηρεσίας DNS για να αποφεύγεται το φιλτράρισμα (firewalls) της κίνησης (καθώς θεωρείται γενικά νόμιμη) και την ευκολία καταχώρησης (register) νέων ονομάτων προς αποφυγή του black listing. Όμως, η μέθοδος των DGA διατηρεί ένα ευάλωτο σημείο εκ κατασκευής. Η κατάχρηση του πρωτοκόλλου DNS από τα bots αφήνει μία πληθώρα ανεπίλυτων ερωτημάτων (NXDomain - Non Existent Domain response) στην προσπάθεια να



Σχήμα 2.3: *Αποκεντροποιημένη δομή botnet*

επιλυθεί η IP του C&C server. Χαρακτηριστικά παραδείγματα DGA-based botnet αποτελούν τα Cryptolocker [11], Conficker [12] και Kraken [13]. Αναλυτικά τη λειτουργία και τα είδη των DGA θα δούμε στην ακόλουθη Ενότητα.

2.2.3 Domain Generation Algorithms (DGA)

Οι αλγόριθμοι παραγωγής ονομάτων τομέα (Domain Generation Algorithm - DGA) [14] χρησιμοποιούνται για την παραγωγή φαινομενικά τυχαίων domain names. Ένα μικρό υποσύνολο των ονομάτων αυτών γίνεται register και αντιστοιχούν σε έγκυρες διευθύνσεις IP των C&C servers. Τα bots καταφέρνουν να έρθουν σε επικοινωνία με τους διακομιστές ελέγχου κάνοντας DNS request μέχρι να μπορέσουν να επιλύσουν κάποιο από τα καταχωρημένα ονόματα. Τα domain names παράγονται βάσει ενός seed, το οποίο μπορεί να είναι κάποια αριθμητική σταθερά, η τρέχουσα ημερομηνία/ώρα, κάποιο trend στο Twitter είτε κάποιος συνδυασμός των παραπάνω. Ανάλογα με τα χαρακτηριστικά των seed τους, οι DGA, διακρίνονται σε δύο κατηγορίες, ανάλογα με την εξάρτηση τους από το χρόνο (Time-Dependent/Time-Independent) και την αιτιότητα τους (Deterministic/Non-Deterministic)). Αυτό το seed, λοιπόν, εξυπηρετεί την εξασφάλιση παραγωγής κοινού συνόλου ονομάτων από τον botmaster και τα bots. Αυτή η συνεχής εναλλαγή καταχωρημένων ονομάτων δίνει πλεονέκτημα στον επίδοξο επιτιθέμενο έναντι στις παραδοσιακές στατικές τεχνικές ανίχνευσης, όπως τα blacklists (λίστες με domain names που έχουν χρησιμοποιηθεί στο παρελθόν για κακόβουλο σκοπό). Επιπλέον, η εξάρτηση των παραγόμενων ονομάτων από το χρόνο, δυσκολεύει το έργο των δυναμικών συστημάτων ανάλυσης κακόβουλου λογισμικού, καθώς για διαφορετικές χρονικές στιγμές θα παρατηρούνται διαφορετικά ονόματα τομέα, καθιστώντας την εξαγωγή συμπερασμάτων δυσκολότερη. Τέλος, η βραχυπρόθεσμη διάρκεια ζωής των ονομάτων συμβάλει στο να αποφεύγεται η κατάταξη τους σε υπηρεσίες φήμης τομέων (domain reputation services).

Εκτός όμως από τους διαφορετικούς τύπους seeds, υπάρχει και μία κατηγοριοποίηση βάσει τεχνικής που χρησιμοποιεί ο αλγόριθμος για την παραγωγή των domain names και είναι οι εξής:

- **Arithmetic-based DGAs**: Οι αλγόριθμοι αυτοί παράγουν τυχαίες ακολουθίες αριθμών. Τα ονόματα προκύπτουν από τη συνένωση των αναπαραστάσεων ASCII που αντιστοιχούν στους αριθμούς αυτούς ή χαρακτήρων που εντοπίζουν σε κωδικοποιημένους πίνακες, οι οποίοι αποτελούν το αλφάβητο του DGA. Πρόκειται για τη συνηθέστερη κατηγορία DGA.
- **Hash-based DGAs**: Τα ονόματα προκύπτουν ως η δεκαεξαδική αναπαράσταση κατακερματισμένων (hashed) αλφαριθμητικών συμβολοσειρών. Αλγόριθμοι κατακερματισμού που έχουν παρατηρηθεί είναι οι SHA256 και MD5.
- **Wordlist-based DGAs**: Τα ονόματα παράγονται από συνένωση τυχαίων λέξεων οι οποίες επιλέγονται είτε από κάποια πηγή (π.χ. λεξικό) είτε βρίσκονται απευθείας ενσωματωμένες στο κακόβουλο λογισμικό. Τα domain names αυτά μοιάζουν λιγότερο τυχαία με αποτέλεσμα να καθίσταται δυσκολότερη η ανίχνευση τους.
- **Permutation-based DGAs**: Δημιουργείται ένα τυχαίο domain name το οποίο στη συνέχεια αντιμετατίθεται για να παραχθούν τα υπόλοιπα ονόματα τομέα.

2.3 Μέθοδοι εντοπισμού DGA-based botnet

Εξηγήσαμε, λοιπόν, πώς οι επίδοξοι επιτιθέμενοι χρησιμοποιούν τους αλγορίθμους αυτούς για την παραγωγή domain names μικρής διάρκειας ζωής τα οποία ανατίθενται δυναμικά στους C&C servers με σκοπό την απόκρυψη της πραγματικής τους ταυτότητας. Με αυτό το τρόπο οι επιτιθέμενοι απέκτησαν μεγάλο πλεονέκτημα απέναντι στους ερευνητές ασφαλείας, καθώς πλέον για την εξάρθρωση ενός botnet δεν αρκεί ο εντοπισμός των ονομάτων των C&C servers, αφού μέχρι να γίνει το μπλοκάρισμα τους, τα ονόματα αυτά θα έχουν αντικατασταθεί. Έτσι, οι ερευνητές ασφαλείας καλούνται να αναπτύξουν νέες μεθόδους για την ανίχνευση και εξουδετέρωση των επιθέσεων αυτών.

Έχει καταστεί σαφές πως τα παραδοσιακά συστήματα ασφαλείας, τα οποία χρησιμοποιούν στατικές μεθόδους είναι αναποτελεσματικά απέναντι σε τέτοιου είδους επιθέσεις. Για παράδειγμα, η μέθοδος του domain name blacklisting, όπου όταν ανακαλύπτεται ένα κακόβουλο domain name, προστίθεται σε μία μαύρη λίστα, η οποία χρησιμοποιείται για μελλοντικό φιλτράρισμα. Στη συνέχεια ακολούθησε μία προσπάθεια η οποία επιστράτευε το reverse engineering για την ανάλυση των ονομάτων με σκοπό την ανακάλυψη του τρόπου λειτουργίας των αλγορίθμων παραγωγής ονομάτων τομέων. Η προσέγγιση αυτή όμως είναι ιδιαιτέρως περίπλοκη και χρονοβόρα (λόγω της πολυπλοκότητας των αλγορίθμων αυτών), το οποίο την καθιστά μη αποδοτική για ανίχνευση σε πραγματικό χρόνο αλλά και μη κλιμακώσιμη. Άλλες ασύγχρονες υλοποιήσεις βασίζονται σε στατιστικά δεδομένα όπως τα DNS responses, με τα οποία όμως μπορούμε να εξάγουμε συμπεράσματα μόνο μετά την επίθεση.

Για παράδειγμα, η εμφάνιση απότομης αύξησης των NXDomain Responses, μπορεί να χαρακτηριστεί ως επίθεση μόνο κατόπιν παρατήρησης μεγάλου όγκου κίνησης του δικτύου, το οποίο όμως δεν μπορεί να εφαρμοστεί σε όλα τα DGA, διότι κάποια παράγουν μικρό αριθμό ονομάτων.

Οι τεχνικές που αναπτύσσονται τα τελευταία χρόνια, βασίζονται αποκλειστικά στα χαρακτηριστικά των domain names, αντιμετωπίζοντάς τα σαν μία απλή αλληλουχία χαρακτήρων. Τα χαρακτηριστικά αυτά μπορεί να είναι πολύ απλά, όπως για παράδειγμα, το μήκος του ονόματος, το πλήθος των ψηφίων που περιέχει, η συχνότητα εμφάνισης χαρακτήρων ή/και λέξεων κλπ, ενώ σε πολλές υλοποιήσεις χρησιμοποιούνται και χαρακτηριστικά όπως η εντροπία και τα N-grams [15] ή και πιο σύνθετα features. Με βάση, λοιπόν, την εξαγωγή των χαρακτηριστικών αυτών από ένα όνομα γίνεται η ταξινόμηση του σε καλόβουλο/έγκυρο όνομα και κακόβουλο/αλγοριθμικό παραγόμενο όνομα. Τέτοιες προσεγγίσεις έχουν υλοποιηθεί με μεθόδους Μηχανικής Μάθησης (όπως Δέντρα Αποφάσεων) ή Βαθιάς Μηχανικής Μάθησης (όπως Νευρωνικά Δίκτυα) και παρουσιάζουν καλή ακρίβεια, αλλά με προβλήματα στην κατανόηση της διαδικασίας παραγωγής των προβλέψεων.

2.4 Μηχανική Μάθηση (Machine Learning)

Η Μηχανική Μάθηση (Machine Learning - ML) [16] πρόκειται για ένα κλάδο της επιστήμης των υπολογιστών που ασχολείται με την ανάπτυξη και μελέτη αλγορίθμων που έχουν την ικανότητα να μαθαίνουν από δεδομένα και κατόπιν να τα γενικεύουν και έτσι να εκτελούν εργασίες χωρίς να τους δοθούν ρητές οδηγίες από τον προγραμματιστή. Οι αλγόριθμοι αυτοί εκπαιδεύονται ουσιαστικά στον να βρίσκουν σχέσεις και μοτίβα στα δεδομένα που τους δίνονται και στη συνέχεια χρησιμοποιούν αυτήν τη πληροφορία για να κάνουν προβλέψεις, να ταξινομούν (classification) ή να ομαδοποιούν (clustering) δεδομένα.

2.4.1 Βασικές έννοιες Μηχανικής Μάθησης

Τα μοντέλα Machine Learning διακρίνονται σε τρεις βασικές κατηγορίες με βάση το τρόπο εκμάθησης του μοντέλου. Οι κατηγορίες αυτές είναι οι εξής:

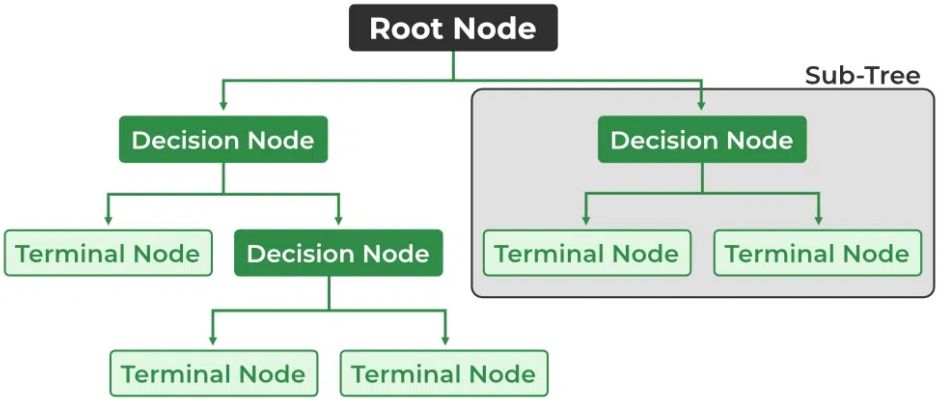
- **Supervised Learning**: Σε αυτούς τους αλγορίθμους δίνεται ένα σύνολο δεδομένων (είσοδοι) καθώς και οι επιθυμητές έξοδοι για τα δεδομένα αυτά κατά την εκπαίδευση του μοντέλου. Κατά την εκπαίδευση οι αλγόριθμοι προσπαθούν να δημιουργήσουν ένα κανόνα αντιστοίχισης εισόδων-εξόδων, τον οποίο γενικοποιούν για την εφαρμογή σε άγνωστα δεδομένα. Παραδείγματα επιβλεπόμενης μάθησης αποτελούν τα δέντρα αποφάσεων (decision trees) και τα Support Vector Machines - SVMs.
- **Unsupervised Learning**: Στην περίπτωση αυτή δεν απαιτείται τα δεδομένα εκπαίδευσης να έχουν ετικέτες (labeled dataset). Οι αλγόριθμοι αυτοί, εντοπίζουν κοινά σημεία στα δεδομένα που τους δίνονται και τα ομαδοποιούν σε μικρότερα υποσύνολα με κοινά χαρακτηριστικά. Είναι ιδιαίτερα χρήσιμοι για ομαδοποίηση δεδομένων

(clustering), ανίχνευση ανωμαλιών (anomaly detection) και μείωση διαστάσεων (dimensionality reduction). Παραδείγματα μη-επιβλεπόμενης μάθησης αποτελούν τα K-means clustering και οι Αυτοκωδικοποιητές (Autoencoders).

- **Reinforcement learning**: Ο πράκτορας (agent) αφήνεται να αλληλεπιδράσει με το περιβάλλον έχοντας λάβει έναν στόχο και κάποιες οδηγίες (πολιτικές) για να επιτύχει το στόχο αυτό. Με κάθε ενέργεια που κάνει λαμβάνει μία ανταμοιβή (είτε θετική όταν πλησιάζει το στόχο είτε αρνητική όταν απομακρύνεται από το στόχο) και το περιβάλλον μεταβαίνει σε μία νέα κατάσταση. Με κάθε ανατροφοδότηση (ανταμοιβή και νέα κατάσταση) ενημερώνει τη πολιτική του έχοντας ως στόχο τη μεγιστοποίηση της αθροιστικής ανταμοιβής (επιδιώκει τις θετικές ανταμοιβές ή την ελαχιστοποίηση της ζημιάς). Η ενισχυτική εκμάθηση χρησιμοποιείται κυρίως για την εκμάθηση εκτέλεσης εργασιών από ρομπότ.

2.4.2 Δέντρα Αποφάσεων (Decision Trees)

Τα δέντρα αποφάσεων (Decision Trees) [17] αποτελούν έναν από τους απλούστερους και δημοφιλέστερους αλγορίθμους επιβλεπόμενης μάθησης και βρίσκουν εφαρμογή σε πληθώρα προβλημάτων. Πρόκειται για μια γραφική αναπαράσταση όλων των πιθανών λύσεων που οδηγούν σε μία απόφαση με βάση συγκεκριμένες συνθήκες/κανόνες (if-then rules).



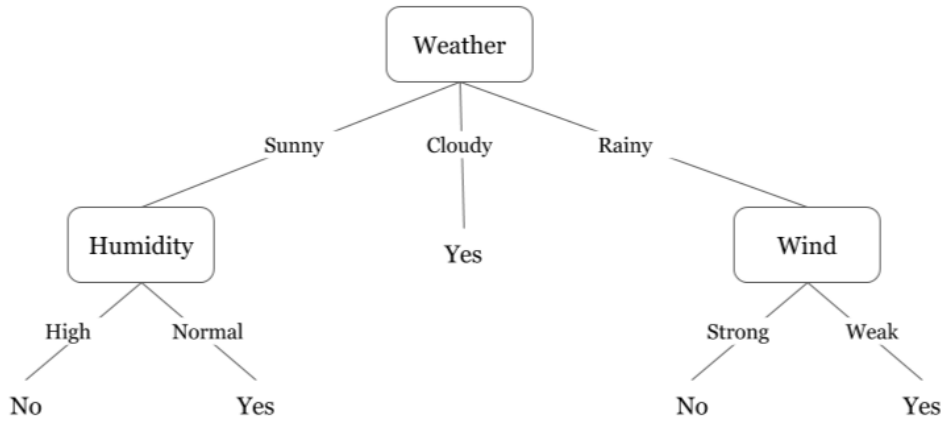
Σχήμα 2.4: *Βασική μορφή δέντρου αποφάσεων*

Όπως μπορούμε να δούμε και στο Σχήμα 2.4 ο αλγόριθμος ξεκινάει από τον κόμβο ρίζα (root node), ο οποίος αντιπροσωπεύει το σύνολο δεδομένων το οποίο καλείται να ταξινομήσει. Κάθε ενδιάμεσος κόμβος (internal/decision node) αναπαριστά μία συνθήκη ελέγχου για κάποιο από τα χαρακτηριστικά (features) εισόδου και οι ακμές/κλαδιά (γραμμές που συνδέουν τους κόμβους) αναπαριστούν τις δυνατές τιμές του χαρακτηριστικού αυτού. Οι τιμές των φύλλων του δέντρου (leaf/terminal nodes) αντιστοιχεί στην κλάση στην οποία ταξινομήθηκε το εκάστοτε δείγμα. Για να δούμε σε τι κλάση έχει ταξινομηθεί ένα δείγμα μετά την ολοκλήρωση του αλγορίθμου, αρκεί να διασχίσουμε το δένδρο μέχρι να καταλήξουμε στο φύλλο που αντιστοιχεί στο δείγμα που μας ενδιαφέρει. Να σημειώσουμε σε αυτό το σημείο πως τα δέντρα αποφάσεων είναι επεξηγήσιμα, καθώς παρέχουν πληροφορίες για το feature

importance. Επιπλέον, για μεγάλο πλήθος features το explainability είναι περιορισμένο καθώς γίνονται πολύ περίπλοκα και γενικότερα είναι λιγότερο ακριβή σε σχέση με τις μεθόδους που θα δούμε στην Ενότητα 2.5.

Για καλύτερη κατανόηση των παραπάνω παρουσιάζεται το παράδειγμα της εικόνας 2.5. Στο παράδειγμα αυτό, καλούμαστε να αποφασίσουμε, με βάση κάποια χαρακτηριστικά του καιρού (όπως Υγρασία, Καιρός, Αέρας) αν μπορούμε να πάμε για hiking. Ας υποθέσουμε πως για τη σημερινή μέρα έχουμε το εξής δειγματικό σημείο:

$(Weather = Rainy, Humidity = High, Wind = Strong)$



Σχήμα 2.5: *Πάραδειγμα Δέντρου Αποφάσεων: Μπορώ να πάω για Hiking [1]*

Διασχίζοντας, λοιπόν, το δέντρο για το παραπάνω δείγμα καταλήγουμε στο δεύτερο φύλλο από τα αριστερά το οποίο μας λέει πως οι καιρικές συνθήκες της ημέρας δεν είναι κατάλληλες για να πάμε για hiking.

Γενικά, το παραπάνω παράδειγμα αλλά και το κάθε δέντρο μπορεί επίσης να αναπαρασταθεί από μία διάζευξη συζεύξεων, για την κάθε κλάση, όπου η κάθε σύζευξη αντιστοιχεί σε ένα μονοπάτι το οποίο ξεκινάει από τη ρίζα του δέντρου και καταλήγει σε ένα φύλλο της κλάσης αυτής. Για παράδειγμα, για το παραπάνω δέντρο αποφάσεων, η κλάση «Yes (Μπορώ να πάω για hiking)» μπορεί να εκφραστεί ως εξής:

$$(Weather = Sunny \cap Humidity = Normal) \cup (Weather = Cloudy) \cup (Weather = Windy \cap Wind = Weak)$$

Η παραπάνω έκφραση προκύπτει από τη μέθοδο του αθροίσματος γινομένων (Sum Of Product - SOP method) η οποία είναι επίσης γνωστή και ως κανονική συζευκτική μορφή (Disjunctive Normal Form) και αποτελεί τον τρόπο κατασκευής ενός δέντρου αποφάσεων. Η μεγαλύτερη πρόκληση κατά την κατασκευή ενός decision tree είναι ο προσδιορισμός των χαρακτηριστικών (feature) που θα ελεγχθούν σε κάθε κόμβο του δέντρου. Η διαδικασία

επιλογής του χαρακτηριστικού αυτού είναι γνωστή και ως «attribute selection» και οι δύο πιο δημοφιλείς μετρικές για την απόφαση αυτή είναι το Information Gain και το Gini Index.

Information Gain

Το κέρδος πληροφορίας (Information Gain) είναι μία από τις δύο δημοφιλέστερες μετρικές που χρησιμοποιούνται για τον καθορισμό του χαρακτηριστικού που πρέπει να χρησιμοποιηθεί για τον διαχωρισμό των δεδομένων σε κάθε εσωτερικό κόμβο ενός δέντρου αποφάσεων. Ο υπολογισμός του κέρδους πληροφορίας γίνεται με τη χρήση της εντροπίας (Entropy) η οποία χρησιμοποιείται για τον προσδιορισμό της τυχαιότητας ενός τυχαίου συνόλου δειγματικών σημείων. Αν θεωρήσουμε ένα σύνολο δεδομένων S, με N κλάσεις, τότε η εντροπία του ορίζεται ως εξής:

$$E(S) = -\sum_{i=1}^{N} p_i \log_2(p_i) \tag{2.1}$$

όπου:

p_i: είναι η πιθανότητα να ταξινομηθεί ένα τυχαίο δειγματικό σημείο στην κλάση i.

Για παράδειγμα αν θεωρήσουμε ένα σύνολο δεδομένων το οποίο αποτελείται από τις κλάσεις 1,2 και 3, τότε έχουμε:

$$E(S) = -(p_1 \log_2 p_1 + p_2 \log_2 p_2 + p_3 \log_2 p_3) \tag{2.2}$$

όπου

p_1, p_2, p_3: είναι η πιθανότητα να ταξινομηθεί ένα τυχαίο δειγματικό σημείο στις κλάσεις 1,2 και 3 αντίστοιχα.

Με τη βοήθεια τώρα της εντροπίας ορίζουμε και το κέρδος πληροφορίας παρακάτω:

$$\text{Gain}(S, A) = \text{Entropy}(S) - \sum_{v \in \text{Values}(A)} \left(\frac{|S_v|}{|S|} \cdot \text{Entropy}(S_v) \right) \tag{2.3}$$

όπου

S: το σύνολο δεδομένων,

A: ένα από τα χαρακτηριστικά του συνόλου,

S_v: ένα υποσύνολο του S,

v: μία τιμή που μπορεί να πάρει το χαρακτηριστικό A και $Values(A)$ το σύνολο των τιμών αυτών.

Η κατασκευή, λοιπόν, ενός δέντρου αποφάσεων με τη βοήθεια του κέρδους πληροφορίας ξεκινάει με το σύνολο των δεδομένων εκπαίδευσης στο κόμβο ρίζα και με χρήση του Information Gain καθορίζεται ποιο χαρακτηριστικό αντιστοιχίζεται σε κάθε εσωτερικό κόμβο (Κανένα από τα μονοπάτια που ξεκινάνε από τη ρίζα και καταλήγουν σε ένα φύλλο δεν πρέπει να περιέχει κάποιο χαρακτηριστικό δύο φορές).

Gini Index

Ο δείκτης Gini (Gini Index) πρόκειται για μία μετρική που χρησιμοποιείται για να μετρήσει πόσο συχνά, ένα τυχαία επιλεγμένο στοιχείο ταξινομείται λανθασμένα. Όσο χαμηλότερη η τιμή του δείκτη, τόσο μικρότερη και η πιθανότητα να ταξινομηθεί εσφαλμένα.

$$Gini(t) = 1 - \sum_{i=1}^{N} p_i^2 \tag{2.4}$$

όπου:
S: το σύνολο δεδομένων,
N: ο αριθμός των κλάσεων,
p_i: το ποσοστό των δειγμάτων που ανήκουν στη κλάση i για τον κόμβο t.

Στα δέντρα αποφάσεων, η μετρική αυτή χρησιμοποιείται για την αξιολόγηση μίας διακλάδωσης, μετρώντας τη διαφορά του δείκτη Gini του κόμβου-γονέα και του σταθμισμένου (weighted) δείκτη Gini των κόμβων-παιδιών. Έχει το πλεονέκτημα ότι είναι υπολογιστικά ταχύτερη από άλλες μετρικές ομογένειας των δεδομένων, όμως έχει την τάση να προτιμάει διακλαδώσεις που δημιουργούν κόμβους-παιδιά ίσου μεγέθους, ακόμα και αν αυτό δεν ευνοεί την ακρίβεια του μοντέλου.

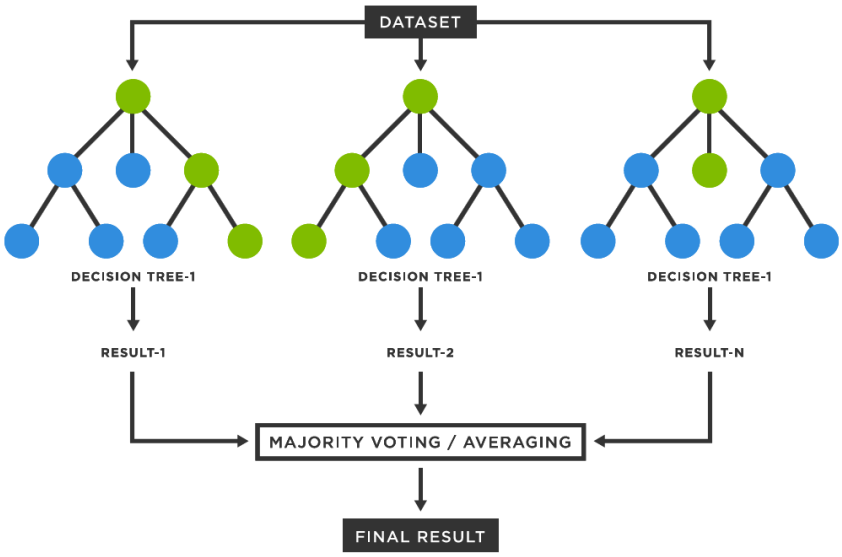
2.4.3 Τυχαίο Δάσος (Random Forest)

Το Τυχαίο Δάσος (Random Forest) [18] πρόκειται για έναν ευρέως εδραιωμένο αλγόριθμο επιβλεπόμενης μάθησης ο οποίος αναπτύχθηκε από τους Leo Breiman et al. [19]. Ο αλγόριθμος αυτός αποτελεί μία επέκταση της μεθόδου bagging ή boostrap aggregating, που είναι μία από τις πιο γνωστές μεθόδους εκμάθησης συνόλου (ensemle learning). Οι μέθοδοι που ανήκουν στην κατηγορία αυτοί επιστρατεύουν ένα σύνολο ταξινομητών (στην προκειμένη περίπτωση δέντρα αποφάσεων) και οι προβλέψεις τους συγκεντρώνονται για να τον προσδιορισμό του δημοφιλέστερου αποτελέσματος.

Ο αλγόριθμος Random Forest, αποτελείται από ένα σύνολο δέντρων αποφάσεων (δάσος), όπου το κάθε ένα από αυτά κατασκευάζεται από ένα δείγμα δεδομένων το οποίο εξάγεται από ένα σύνολο εκπαίδευσης με αντικατάσταση (που σημαίνει ότι κάθε στοιχείο μπορεί να επιλεχθεί περισσότερες από μία φορές). Ο συνδυασμός των προβλέψεων όλων αυτών των δέντρων οδηγεί σε ένα όχι μόνο πιο ακριβές μοντέλο αλλά και λιγότερο επιρρεπές στην υπερπροσαρμογή (overfitting). Επιπλέον, για το κάθε δέντρο επιλέγεται ένα τυχαίο υποσύνολο χαρακτηριστικών (features). Με αυτό τον τρόπο ο αλγόριθμος καταφέρνει να εντοπίσει διαφορετικά μοτίβα και σχέσεις μεταξύ των δεδομένων, το οποίο προσφέρει επίσης ακρίβεια αλλά και σταθερότητα στις προβλέψεις του μοντέλου.

Όσον αφορά την τελική πρόβλεψη του μοντέλου, υπάρχουν δύο βασικές μέθοδοι συνδυασμού των προβλέψεων του κάθε δέντρου που απαρτίζει το τυχαίο δάσος:

- **Ψηφοφορία (Voting)**: Ο αλγόριθμος χρησιμοποιεί ένα μηχανισμό ψηφοφορίας όπου



Σχήμα 2.6: *Αλγόριθμος Random Forest*

το κάθε δέντρο παρέχει την κλάση την οποία προέβλεψε και η τελική πρόβλεψη είναι η κλάση την οποία ψήφισε η πλειοψηφία των δέντρων. Η μέθοδος αυτή χρησιμοποιείται για προβλήματα ταξινόμησης.

- **Μέσος όρος (Averaging)**: Στη περίπτωση αυτή, το κάθε δέντρο παρέχει στον αλγόριθμο την αριθμητική πρόβλεψη που έχει κάνει και το τελικό αποτέλεσμα προκύπτει από τον μέσο όρο των προβλέψεων όλων των δέντρων του τυχαίου δάσους. Η τεχνική αυτή εφαρμόζεται για προβλήματα παλινδρόμησης (regression problems).

Την παραπάνω διαδικασία μπορούμε επίσης να δούμε και στο Σχήμα 2.6 όπου φαίνεται πως από το αρχικό σύνολο δεδομένων κατασκευάζονται τα διαφορετικά δέντρα των οποίων οι προβλέψεις συνδυάζονται (ανάλογα με τον τύπο του προβλήματος) με τους παραπάνω τρόπους για την εξαγωγή του τελικού αποτελέσματος.

Τα σημαντικότερα πλεονεκτήματα του αλγορίθμου τυχαίου δάσους είναι ότι μειώνει το ρίσκο υπερπροσαρμογή, όπως αναφέρθηκε και προηγουμένως, προσφέρει ευελιξία καθώς μπορεί να διαχειριστεί τόσο προβλήματα ταξινόμησης όσο και παλινδρόμησης και τέλος, χρησιμοποιείται για την εκτίμηση της σημαντικότητας των χαρακτηριστικών. Όπως είναι όμως λογικό, είναι πιο περίπλοκος και απαιτεί περισσότερο χρόνο και πόρους από ένα απλό δέντρο αποφάσεων.

2.4.4 K-means Clustering

Ο K-means clustering πρόκειται για έναν αλγόριθμο μη επιβλεπόμενης μάθησης και παρουσιάστηκε αρχικά από τον Stuart Lloyd [20]. Ο στόχος του αλγορίθμου αυτού είναι να ομαδοποιήσει/συσταδοποιήσει (clustering) ένα σύνολο δεδομένων χωρίς ετικέτα (unlabeled dataset).

Ο αλγόριθμος ξεκινάει τη διαδικασία συσταδοποίησης αρχικοποιώντας τυχαία K σημεία στο χώρο. Τα σημεία αυτά ονομάζονται μέσα (means) ή κέντρα συστάδων (cluster centroids). Στη συνέχεια, κάθε στοιχείο του συνόλου ανατίθεται στη συστάδα η οποία ορίζεται από το κοντινότερο κέντρο του. Η απόσταση αυτή συνήθως υπολογίζεται με χρήση της Ευκλείδιας απόστασης. Έπειτα, ανανεώνει τις συντεταγμένες του κέντρου, υπολογίζοντας τον μέσο όρο των στοιχείων που ανατέθηκαν στη συστάδα που ορίζει ο μέσος. Τα βήματα της ανάθεσης και της ανανέωσης των συντεταγμένων επαναλαμβάνονται είτε έως ότου τα κέντρα πάψουν να αλλάζουν σημαντικά είτε μέχρι να γίνει ένας ορισμένος αριθμός επαναλήψεων. Όταν επέλθει μία από τις δύο παραπάνω συνθήκες ο αλγόριθμος τερματίζει και επιστρέφει τα τελικά κέντρα καθώς και την ανάθεση των δεδομένων στη κάθε συστάδα [21].

Παρακάτω μπορούμε να δούμε τον ψευδοκώδικα του αλγορίθμου όπου σαν είσοδο δέχεται το σύνολο δεδομένων που καλείται να συσταδοποιήσει καθώς και το πλήθος τον συστάδων και επιστρέφει τις συστάδες καθώς και τα κέντρα τους:

Αλγοριθμος 2.1: *K-means clustering*

1: **procedure** K-means (σύνολο δεδομένων S, πλήθος συστάδων - K)
2: Αρχικοποίηση K κέντρων επιλέγοντας k τυχαία σημεία του συνόλου δεδομένων.
3: **repeat**
4: Υπολογισμός $dist(x, \mu_i)$, $\forall x \in S$ με κάθε κέντρο μ_i των συστάδων c_i.
5: Ανάθεση κάθε σημείου στο πιο κοντινό κέντρο μ_i.
6: Υπολογισμός του νέου κέντρου της κάθε συστάδας c_i
7: **until** Νέα κέντρα ισούνται με τα προηγούμενα ή ολοκληρωθεί ο αριθμός κάποιων επαναλήψεων
8: **return** Συσταδοποιημένα δεδομένα και κέντρα συστάδων
9: **end procedure**

Ο βασικότερος στόχος της συσταδοποίησης K-means είναι να χωρίσει τα δεδομένα σε Κ συστάδες, με τρόπο τέτοιο ώστε τα στοιχεία της κάθε συστάδας να μοιάζουν μεταξύ τους αλλά όχι με αυτά που ανήκουν στις υπόλοιπες. Αυτό επιτυγχάνεται ελαχιστοποιώντας τις αποστάσεις των σημείων εντός της συστάδας ενώ ταυτόχρονα προσπαθεί να μεγιστοποιήσει τις αποστάσεις μεταξύ των διαφορετικών συστάδων. Η επιλογή του βέλτιστου K αποτελεί ένα αρκετά δύσκολο πρόβλημα. Μία από τις δημοφιλέστερες μεθόδους εύρεσης του βέλτιστου K είναι η μέθοδος Elbow [22].

2.5 eXplainable Artificial Intelligence (XAI)

Καθώς η Τεχνητή Νοημοσύνη γίνεται μέρα με τη μέρα όλο και πιο προηγμένη αλλά και προσιτή σε όλους μας, δημιουργείται η ανάγκη για κατανόηση του πώς ένας αλγόριθμος καταλήγει στο αποτέλεσμα που μας δίνει.Τα μοντέλα μηχανικής μάθησης συχνά αναφέρονται και ως «μαύρα κουτιά», το οποία είναι αδύνατο να ερμηνευθούν. Τα μοντέλα αυτά μαθαίνουν απευθείας από τα δεδομένα και για αυτό το λόγο ακόμα και οι μηχανικοί ή οι επιστήμονες δεδομένων που τα αναπτύσσουν δεν μπορούν να γνωρίζουν ή να κατανοήσουν με ποιο τρόπο και γιατί ο αλγόριθμος καταλήγει σε κάποιο αποτέλεσμα. Αυτό, λοιπόν, αποτελεί και το

βασικότερο λόγο που υπάρχουν αμφιβολίες για την ενσωμάτωση της Τεχνητής Νοημοσύνης σε διάφορους κλάδους της κοινωνίας μας στους οποίους ενδεχομένως θα μπορούσε να φανεί ιδιαιτέρως χρήσιμη. Για παράδειγμα, στον κλάδο της ιατρικής θα ήταν εξαιρετικά χρήσιμο να υπάρχει ένα μοντέλο που να αναλύει τις εξετάσεις των ασθενών και να μπορεί να αποφανθεί για το αν πάσχουν από κάποια ασθένεια ή όχι. Και έχουν αναπτυχθεί πολλά τέτοια μοντέλα, που όμως η εφαρμογή τους, δεν είναι ακόμα πραγματικότητα καθώς δεν υπάρχει η απαραίτητη διαφάνεια στο μοντέλο για τη λήψη μίας απόφασης που αφορά μία ανθρώπινη ζωή. Αντίστοιχα, στον τραπεζικό κλάδο μπορεί να χρησιμοποιηθεί για την απόφαση του αν κάποιος δικαιούται δάνειο ή όχι. Πάλι, όμως, υπάρχει το πρόβλημα της διαφάνειας και επιπλέον το μοντέλο μπορεί να είναι είτε θετικά είτε αρνητικά προκατειλημμένο (biased) απέναντι σε κάποια κατηγορία ανθρώπων, το οποίο θα οδηγούσε σε αδικίες. Δημιουργείται, λοιπόν, η ανάγκη κατανόησης των μοντέλων AI για τη καλύτερη λήψη αποφάσεων, ώστε να γίνει ένα εργαλείο που σε συνδυασμό με την ανθρώπινη κριτική σκέψη και εμπειρία, να οδηγεί στο βέλτιστο αποτέλεσμα. Στο σημείο αυτό, έρχεται η προσπάθεια ανάπτυξης μεθόδων επεξηγηματικότητας των μοντέλων αυτών.

Οι αλγόριθμοι επεξηγηματικότητας [23, 24] χωρίζονται σε κατηγορίες βάσει δύο κύριων σημείων αποφάσεων όσον αφορά τη προσέγγιση. Το πρώτο έχει να κάνει με το εάν είναι απαραίτητη η γνώση των ιδιαίτερων χαρακτηριστικών του μοντέλου ή όχι: model-agnostic και model-specific. Στη πρώτη περίπτωση, με την οποία θα ασχοληθούμε, το μοντέλο αντιμετωπίζεται σαν «μαύρο κουτί» και δεν απαιτείται κάποια γνώση των εσωτερικών διαδικασιών για την παραγωγή επεξηγήσεων. Η προσέγγιση αυτή προσφέρει ευελιξία στους προγραμματιστές καθώς μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο Στη δεύτερη περίπτωση, απαιτείται γνώση των εσωτερικών διεργασιών του μοντέλου και για το λόγο αυτό, περιορίζονται στην επεξήγηση συγκεκριμένων μοντέλων. Το δεύτερο σημείο απόφασης αφορά το εύρος εφαρμογής των επεξηγήσεων: τοπική επεξηγηματικότητα (local explainability) και καθολική επεξηγηματικότητα (global explainability). Η πρώτη παρέχει επεξηγήσεις για την έξοδο του μοντέλου που αφορά μία συγκεκριμένη είσοδο, ενώ η δεύτερη δίνει τη δυνατότητα για μία πιο ολιστική κατανόηση του μοντέλου.

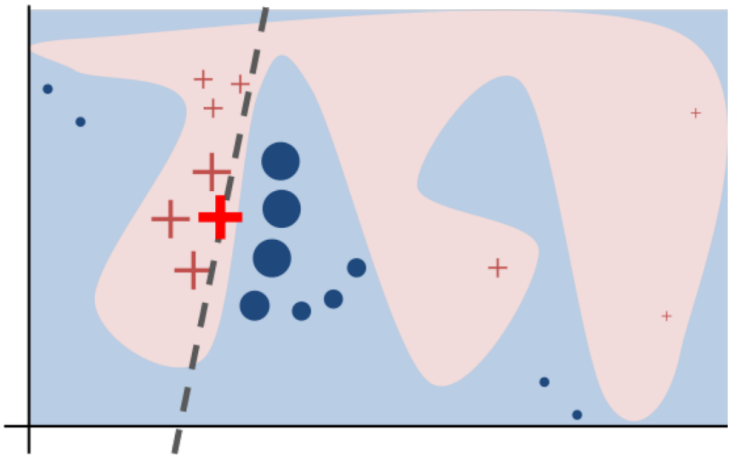
2.5.1 Permutation Feature Importance (PFI)

Η μέθοδος Permutation Feature Importance (PFI) προσπαθεί να απαντήσει στην ερώτηση: Ποια είναι τα σημαντικότερα χαρακτηριστικά του μοντέλου μου; Η μέθοδος αυτή ουσιαστικά υπολογίζει την αύξηση του σφάλματος πρόβλεψης (και άρα τη μείωση της απόδοσης του μοντέλου) μετά την αντιμετάθεση των τιμών ενός χαρακτηριστικού. Με αυτό τον τρόπο μπορούμε να δούμε πόσο ρόλο παίζει (ή δεν παίζει) ένα χαρακτηριστικό για την τελική απόφαση του μοντέλου. Δηλαδή, ένα χαρακτηριστικό είναι σημαντικό εάν μετά την αναδιάταξη το σφάλμα του μοντέλου αυξάνεται σημαντικά, καθώς αυτό επισημαίνει πως το μοντέλο αρχικά είχε βασιστεί στο χαρακτηριστικό αυτό για πραγματοποιήσει την πρόβλεψη. Αντιθέτως, ένα χαρακτηριστικό θα θεωρηθεί ασήμαντο εάν το σφάλμα επηρεαστεί ελάχιστα ή και καθόλου καθώς υποδεικνύει πως εξ αρχής το μοντέλο το είχε αγνοήσει. Το PFI, λοιπόν, έχει άμεση συσχέτιση με το σφάλμα του μοντέλου, το οποίο δεν είναι απαραίτητα αρνητικό,

αλλά πολλές φορές δεν είναι αυτό που χρειαζόμαστε. Για παράδειγμα, μπορεί να μας ενδιαφέρει η ανθεκτικότητα της εξόδου του μοντέλου όσο μεταβάλλονται τα χαρακτηριστικά. Σε αυτή τη περίπτωση, δεν μας ενδιαφέρει η μείωση της απόδοσης του μοντέλου κατά την μεταβολή ενός χαρακτηριστικού, αλλά κατά πόσο μπορεί να ερμηνευθεί η διακύμανση της εξόδου από κάθε χαρακτηριστικό. Επιπλέον, η μέθοδος αυτή δεν μπορεί να εφαρμοστεί σε μοντέλα μη επιβλεπόμενης μάθησης, καθώς απαιτείται γνώση της πραγματικής εξόδου για τον υπολογισμό του PFI. Τέλος, μπορεί τα αποτελέσματα της να είναι παραπλανητικά εάν υπάρχουν χαρακτηριστικά με μεγάλη συσχέτιση (correlated), καθώς ανακατεύοντας το ένα να επηρεάζεται ταυτόχρονα και το άλλο.

2.5.2 Local Interpretable Model-agnostic Explanations (LIME)

Μία ακόμα γνωστή μέθοδος XAI είναι η LIME (Local Interpretable Model-agnostic Explanations) η οποία πρωτοπαρουσιάστηκε από τους Ribeiro et al. [2]. Η βασική ιδέα είναι ότι χρησιμοποιούνται τοπικά υποκατάστατα μοντέλα (local surrogate models), τα οποία εκπαιδεύονται έτσι ώστε να προσεγγίσουν τις προβλέψεις του υποκείμενου μοντέλου. Τα τοπικά υποκατάστατα μοντέλα είναι ερμηνεύσιμα μοντέλα που χρησιμοποιούνται για την εξήγηση μεμονωμένων προβλέψεων black-box μοντέλων. Σε αντίθεση με τη μέθοδο PFI το LIME δεν προσφέρει καθολική επεξηγηματικότητα του μοντέλου αλλά τοπική, κάτι που μπορεί να γίνει αντιληπτό και από το όνομα της μεθόδου αυτής. Ο λόγος είναι ότι για ένα πολύπλοκο black-box μοντέλο είναι ευκολότερο να εξάγουμε τοπικές εξηγήσεις από ότι καθολικές.



Σχήμα 2.7: *Παράδειγμα διαισθητικής κατανόησης της LIME [2]*

Για να εξηγήσουμε διαισθητικά πως λειτουργεί η LIME παραθέτουμε το παράδειγμα του Σχήματος 2.7 από τη δημοσίευση των Ribeiro et al. Στο Σχήμα αυτό, το μπλε και ροζ φόντο που βλέπουμε εκφράζει τη συνάρτηση απόφασης (δηλαδή τις δύο διαφορετικές κλάσεις του) ενός black-box μοντέλου. Η συνάρτηση αυτή όπως φαίνεται και στο παραπάνω σχήμα είναι πολύ περίπλοκη για να προσεγγιστεί με ένα γραμμικό μοντέλο. Ο έντονος κόκκινος σταυρός είναι το επιλεγμένο δειγματικό σημείο προς επεξήγηση. Στη συνέχεια, παίρνει τυχαία δειγματικά σημεία, υπολογίζει τις προβλέψεις με χρήση της συνάρτησης απόφασης (σταυρός για ροζ και κύκλος για μπλε) και έπειτα σταθμίζει (weighs) τα σημεία αυτά ανάλογα με

την εγγύτητα στο δείγμα προς επεξήγηση (όσο πιο κοντά στον έντονο κόκκινο σταυρό τόσο μεγαλύτερος ο κύκλος ή σταυρός αντίστοιχα). Τέλος, με βάση τα παραπάνω προκύπτει η διακεκομμένη γραμμή του σχήματος εκφράζει το επεξηγήσιμο μοντέλο από το οποίο εξάγουμε τοπική επεξήγηση για το επιλεγμένο δειγματικό σημείο. Το ερμηνεύσιμο αυτό μοντέλο θα πρέπει να είναι μία καλή προσέγγιση των προβλέψεων του μοντέλου τοπικά, χωρίς αυτό να σημαίνει ότι είναι και καλή συνολική προσέγγιση. Αυτή η μετρική ονομάζεται και τοπική πιστότητα (local fidelity) και μας δίνει πληροφορίες για την αξιοπιστία των τοπικών επεξηγήσεων. Η LIME αποτελεί μία από τις λίγες μεθόδους που λειτουργούν για δεδομένα σε πίνακες, κείμενο και εικόνες.

2.5.3 SHapley Additive exPlanations (SHAP)

Μία από τις δημοφιλέστερες μεθόδους επεξήγησης μοντέλων μηχανικής μάθησης αποτελεί η SHAP (SHapley Additive exPlanations), η οποία παρουσιάστηκε από τους Lundberg and Lee [25] και είναι βασισμένη στα shapley values της θεωρίας παιγνίων. Η μέθοδος αυτή αποτελεί μία από τις state-of-the-art τεχνικές στον τομέα της επεξηγηματικότητας μοντέλων μηχανικής μάθησης και μπορεί να επιτύχει τόσο τοπική όσο και καθολική επεξηγηματικότητα.

Shapley Values

Για την κατανόηση της λειτουργίας της SHAP είναι απαραίτητος ο ορισμός των τιμών Shapley. Στη θεωρία παιγνίων (game theory), οι τιμές Shapley χρησιμοποιούνται για να υπολογίσουν τη συμβολή του κάθε παίκτη στη νίκη ενός συνεργατικού παιχνιδιού (cooperative game). Πήραν το όνομά τους από τον Lloyd Shapley, ο οποίος εισήγαγε την έννοια το 1953.

Ας υποθέσουμε ότι έχουμε ένα παίγνιο με n παίκτες και ας θεωρήσουμε το σύνολο παικτών N, όπου $N = 1, 2, ..., n$. Ας θεωρήσουμε επίσης, μία συμμαχία (coalition) παικτών S, όπου $S \subseteq N$, στο οποίο περιλαμβάνεται και το κενό σύνολο, δηλαδή μία συμμαχία στην οποία δεν συμμετέχει κανένας παίκτης. Αν, για παράδειγμα, υποθέσουμε πως έχουμε 3 παίκτες, όλες οι δυνατές συμμαχίες είναι: $\{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. Έπειτα, ορίζουμε μία συνάρτηση v, η οποία αντιστοιχεί κάθε συμμαχία S με έναν πραγματικό αριθμό. Άρα η ποσότητα $v(S)$ εκφράζει την αξία της συμμαχίας S. Αυτό που απομένει τώρα, είναι να υπολογιστεί η συνεισφορά του κάθε παίκτη με τον δικαιότερο τρόπο. Για τον υπολογισμό, λοιπόν, της συνεισφοράς ενός παίκτη i, βρίσκουμε όλες τις δυνατές μεταθέσεις του συνόλου S (δηλαδή όλες τις δυνατές διατάξεις με τις οποίες μπορεί να έχουν μπει οι παίκτες στο παιχνίδι). Στη συνέχεια, υπολογίζεται η οριακή συνεισφορά (marginal contribution) του παίκτη i όταν εντάσσεται σε κάθε μία από αυτές. Τέλος, για την συνολικής συνεισφοράς του παίκτη i, υπολογίζεται ο σταθμισμένος μέσος όρος των παραπάνω συνεισφορών. Με αυτό τον τρόπο εξασφαλίζεται η δίκαιη κατανομή της συνεισφοράς των παικτών, καθώς έχουν ληφθεί υπόψη όλα τα δυνατά σενάρια με τα οποία θα μπορούσαν να έχουν συνεισφέρει σε μία συμμαχία. Η παραπάνω διαδικασία και επομένως ο υπολογισμός της τιμής Shapley για έναν παίκτη i δίνεται από την παρακάτω μαθηματική σχέση:

$$\phi_i(v) = \sum_{S \subseteq N\{i\}} \frac{|S|!(n-|S|-1)!}{n!} \left(v(S \cup \{i\}) - v(S)\right) \tag{2.5}$$

όπου:
$S \subseteq N\{i\}$: όλα τα δυνατά υποσύνολα παικτών που δεν περιέχουν τον παίκτη i και
$|S|$: το πλήθος τον παικτών που ορίζουν τη συμμαχία S.

Εναλλακτικά, η παραπάνω σχέση μπορεί να γραφεί και ως εξής:

$$\phi_i(v) = \frac{1}{n!} \sum_{R} \left(v(P_i^R \cup \{i\}) - v(P_i^R)\right) \tag{2.6}$$

όπου:
R: μία διάταξη των παικτών και
P_i^R: το σύνολο των παικτών που προηγούνται του παίκτη i στη διάταξη R .

Για καλύτερη κατανόηση των παραπάνω, παραθέτουμε στη συνέχεια, ένα απλό παράδειγμα παιγνίου [26, 27] για τον υπολογισμό της συνεισφοράς των παικτών του. Ας υποθέσουμε ότι υπάρχουν 3 εργοστάσια παραγωγής γαντιών τα οποία αποφασίζουν να συνεργαστούν ώστε καθένα από αυτά να παράγει είτε μόνο δεξιόχειρα γάντια είτε μόνο αριστερόχειρα γάντια. Για να μπορέσουν να πωληθούν και άρα να έχουν αξία πρέπει τα γάντια να είναι σε ζευγάρια. Τα γάντια που δεν έχουν ζευγάρι έχουν μηδενική αξία. Αν θεωρήσουμε πως τα τα εργοστάσια 1 και 2 παράγουν αριστερόχειρα γάντια ενώ το εργοστάσιο 3 δεξιόχειρα προκύπτει η εξής συνάρτηση αξίας:

$$v(S) = \begin{cases} 1, & \text{αν } S \in \{\{1,3\},\{2,3\},\{1,2,3\}\} \\ 0, & \text{αν } S \in \{\{1\},\{2\},\{3\},\{1,2\}\} \end{cases} \tag{2.7}$$

Ο στόχος τώρα, λοιπόν, είναι να μπορέσουμε να μοιράσουμε με το δικαιότερο τρόπο τις εισπράξεις στα 3 εργοστάσια. Στον Πίνακα 2.1 μπορούμε να δούμε την οριακή συνεισφορά του κάθε εργοστασίου για τις 6 διαφορετικές διατάξεις. Ενδεικτικά, για το Εργοστάσιο 1, οι τιμές του παρακάτω πίνακα προκύπτουν ως εξής: Στις διατάξεις {1,2,3} και {1,3,2} δεν προηγείται κάποιος του εργοστασίου 1, επομένως το marginal contribution (στη συνέχεια θα αναφέρεται ως MC για συντομία) είναι $v(\{1\}) - v(\emptyset) = 0$. Αντίστοιχα για τη διάταξη {2,1,3} έχουμε $v(\{1,2\}) - v(\{2\}) = 0$, για τη {2,3,1} και {3,2,1} είναι $v(\{1,2,3\}) - v(\{2,3\}) = 1 - 1 = 0$ και τέλος, για τη {3,1,2} έχουμε $v(\{1,3\}) - v(\{3\}) = 1 - 0 = 1$.

Από τον Πίνακα 2.1 και τη Σχέση 2.6 προκύπτουν οι τιμές Shapley και άρα η συνεισφορά των τριών εργοστασίων:

$$\phi_1(v) = \frac{1}{6} \cdot 1 = \frac{1}{6} = \phi_2(v) \text{ και } \phi_3(v) = \frac{1}{6} \cdot 4 = \frac{2}{3} \tag{2.8}$$

Επιπλέον, είναι σημαντικό να σημειωθεί πως οι τιμές Shapley είναι η μόνη μέθοδος που

Διάταξη	**Εργοστάσιο 1**	**Εργοστάσιο 2**	**Εργοστάσιο 3**
{1,2,3}	0	0	1
{1,3,2}	0	0	1
{2,1,3}	0	0	1
{2,3,1}	0	0	1
{3,1,2}	1	0	0
{3,2,1}	0	1	0

Πίνακας 2.1: *Οριακή συνεισφορά του κάθε εργοστασίου για την κάθε δυνατή διάταξη*

ικανοποιεί τέσσερεις πολύ σημαντικές ιδιότητες, των οποίων ο συνδυασμός θα μπορούσε να θεωρηθεί ο ορισμός μίας δίκαιης κατανομής. Οι ιδιότητες αυτές είναι οι εξής:

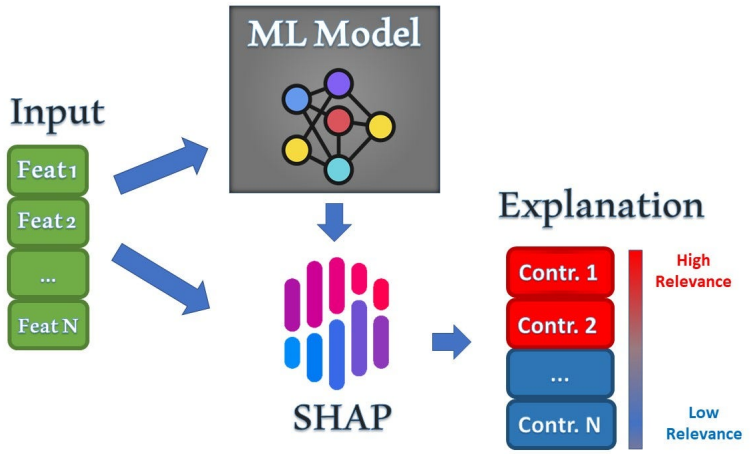
- **Efficiency**: Το άθροισμα της συνεισφοράς (δηλαδή της τιμής Shapley) όλων των παικτών, πρέπει να ισούται με τη διαφορά της πρόβλεψης για το x και του μέσου όρου. Δηλαδή $\sum_{j=1}^{n} \hat{\phi}_j = f(x) - E_x(\hat{f}(x))$, όπου $\hat{f}$ η συνάρτηση πρόβλεψης του μοντέλου.
- **Symmetry**: Εάν δύο παίκτες έστω i και j συνεισφέρουν το ίδιο σε κάθε δυνατή συμμαχία, τότε και η συνολική συνεισφορά τους θα είναι ίση. Δηλαδή αν $v(S \cup \{i\}) = v(S \cup \{j\})$ για κάθε συμμαχία S, τότε θα ισχύει $\phi_i = \phi_j$.
- **Dummy**: Αν ένας παίκτης i δεν προσθέτει καμία αξία, σε καμία από τις συμμαχίες όταν προστίθεται σε αυτές, τότε η συνολική συνεισφορά του παίκτη πρέπει να ισούται με μηδέν. Δηλαδή, $v(S \cup \{i\}) = v(S)$, για κάθε συμμαχία S, τότε $\phi_i = 0$.
- **Additivity**: Αν θεωρήσουμε πως ένας παίκτης i συμμετέχει σε δύο διαφορετικά παίγνια με συναρτήσεις u και v. Τότε η συνολική συνεισφορά του παίκτη αυτού θα ισούται με το άθροισμα της συνεισφοράς του παίκτη i στο καθένα από αυτά. Δηλαδή, αν $\phi_i(u)$ η συνεισφορά του στο πρώτο παίγνιο και $\phi_i(v)$ η συνεισφορά του στο δεύτερο, τότε θα ισχύει ότι $\phi_i(u + v) = \phi_i(u) + \phi_i(v)$.

Η αναλογία ενός συνεργατικού παιχνιδιού και ενός μοντέλου μηχανικής μάθησης μπορεί να γίνει αν θεωρήσουμε ως «παίκτες» τα χαρακτηριστικά που συνεργάζονται για την εξαγωγή μίας πρόβλεψης και ως «νίκη ή επιβράβευση» την πρόβλεψη αυτή. Με αυτό τον τρόπο ανάγουμε ένα περίπλοκο πρόβλημα, δηλαδή «Πώς μπορούμε να ερμηνεύσουμε ένα μοντέλο μηχανική μάθησης μαύρου κουτιού;», σε ένα απλούστερο, δηλαδή «Πόσο και πώς συνέβαλε ο κάθε παίκτης-χαρακτηριστικό στη νίκη-πρόβλεψη;». Και η απάντηση στην ερώτηση αυτή δίνεται από την τιμή Shapley του κάθε χαρακτηριστικού-παίκτη.

Από τα Shapley Values στη SHAP

Είναι πλέον κατανοητό, πως οι τιμές Shapley είναι εξαιρετικά χρήσιμες για την ερμηνεία μοντέλων μαύρου κουτιού, καθώς έχουν και σημαντικό θεωρητικό υπόβαθρο. Ένα όμως από τα μεγαλύτερα μειονεκτήματα τους είναι η απαίτηση μεγάλου υπολογιστικού χρόνου.

Ο υπολογισμός είναι τόσο δαπανηρός γιατί υπάρχουν 2^N πιθανοί συνασπισμοί των N χαρακτηριστικών και επιπλέον κάθε απουσιάζον χαρακτηριστικό πρέπει να προσομοιωθεί με τη δημιουργία κάποιων τυχαίων δειγμάτων γεγονός που αυξάνει τη διακύμανση των τιμών Shapley. Ο στόχος της SHAP είναι να εξηγήσει μία πρόβλεψη x υπολογίζοντας (με χρήση των τιμών Shapley) την συμβολή κάθε χαρακτηριστικού του μοντέλου στη πρόβλεψη αυτή. Η καινοτομία που φέρνει η μέθοδος αυτή, είναι ότι αναπαριστά τις επεξηγήσεις των τιμών Shapley ως μία γραμμικό μοντέλο (additive feature attribution method). Η προσέγγιση με το γραμμικό μοντέλο μας συνδέει με τη προσέγγιση της LIME που αναφέρθηκε στην παραπάνω ενότητα. Στην Εικόνα 2.8, μπορούμε να δούμε συνοπτικά τη διαδικασία που ακολουθείται για εξαγωγή επεξηγήσεων με χρήση της SHAP.



Σχήμα 2.8: *Σκελετός λειτουργίας της SHAP*

Η SHAP, ορίζει την επεξήγηση ενός δειγματικού στοιχείου x ως εξής:

$$g(z') = \phi_0 + \sum_{j=1}^{M} \phi_j z'_j \tag{2.9}$$

όπου:

g: το μοντέλο επεξήγησης,

$z' \in \{0,1\}^M$: το coalition vector (διάνυσμα συνασπισμού) ή απλοποιημένα χαρακτηριστικά, όπως αναφέρεται στη δημοσίευση των Lundberg and Lee (2017). Στο διάνυσμα αυτό μία καταχώρηση 1 υποδεικνύει ότι η αντίστοιχη τιμή χαρακτηριστικού είναι παρούσα, ενώ για 0 ότι είναι απούσα,

M: το μέγιστο μέγεθος συνασπισμού (δηλαδή το πλήθος των χαρακτηριστικών),

z'_j: είναι 0 ή 1 ανάλογα με το αν το χαρακτηριστικό j υπάρχει ή όχι για το δείγμα x,

ϕ_j: η τιμή Shapley του χαρακτηριστικού j για το δειγματικό στοιχείο που εξετάζουμε και

ϕ_0: το null output του μοντέλου, δηλαδή η έξοδος του μοντέλου, όταν απουσιάζουν όλα τα χαρακτηριστικά (τιμή ανεξάρτητη των χαρακτηριστικών).

Η παραπάνω σχέση μπορεί να απλοποιηθεί, αν για το δειγματικό στοιχείο x που μας ενδιαφέρει, θεωρήσουμε το coalition vector x' όπου όλα τα χαρακτηριστικά είναι παρόντα (δηλαδή ένα διάνυσμα μόνο με άσους). Έτσι προκύπτει:

$$g(x') = \phi_0 + \sum_{j=1}^{M} \phi_j \tag{2.10}$$

Αναφέραμε προηγουμένως, ότι οι τιμές Shapley ικανοποιούν τις ιδιότητες Efficiency, Symmetry, Dummy και Additivity. Οι τιμές SHAP ικανοποιούν επίσης τις παραπάνω ιδιότητες, αλλά επιπλέον στη δημοσίευση των Lundberg and Lee [25] περιγράφονται κάποιες ακόμα οι ακόλουθες τρεις επιθυμητές ιδιότητες:

- **Local Accuracy**: Αν θεωρήσουμε πως η είσοδος x και η απλοποιημένη είσοδος x' είναι σχεδόν ίδιες, δηλαδή αν $x' \approx x$, τότε το μοντέλο f και το μοντέλο επεξήγησης g θα πρέπει να παράγουν σχεδόν την ίδια πρόβλεψη, δηλαδή $g(x') \approx f(x)$.
- **Missingness**: Αν ένα χαρακτηριστικό j απουσιάζει, δηλαδή $x'_j = 0$, τότε η συνεισφορά του στο αποτέλεσμα του μοντέλου θα πρέπει να είναι μηδενική, δηλαδή $\phi_j = 0$. Αυτό υπονοεί ότι ο μόνος τρόπος για να επηρεάσει ένα χαρακτηριστικό την πρόβλεψη είναι η παρουσία του και όχι η απουσία του.
- **Consistency**: Αν η συνεισφορά ενός χαρακτηριστικού αλλάξει, τότε η επιρροή του χαρακτηριστικού αυτού ως προς το μοντέλο δεν μπορεί να έχει αντίθετο αποτέλεσμα. Δηλαδή, αν έχουμε ένα νέο μοντέλο, όπου ένα χαρακτηριστικό επηρεάζει περισσότερο με θετικό τρόπο το μοντέλο, τότε δεν μπορεί να μειωθεί η απόδοση του νέου μοντέλου.

Kernel SHAP

Για να αντιμετωπιστεί το πρόβλημα της υπολογιστικής απαίτησης των τιμών Shapley, οι Lundberg and Lee [25] αναπτύξανε και παρουσιάσανε τον Kernel SHAP, ένα μέσο προσέγγισης των τιμών Shapley που απαιτεί πολύ μικρότερο αριθμό δειγμάτων. Αυτό επιτυγχάνεται δίνοντας στο μοντέλο διάφορες διατάξεις των χαρακτηριστικών του δείγματος που καλούμαστε να επεξηγήσουμε (με απουσιάζοντα χαρακτηριστικά). Όμως, η πλειοψηφία των μοντέλων μηχανικής μάθησης, δεν επιτρέπει απλά να αγνοήσουμε κάποιο χαρακτηριστικό. Για το λόγο αυτό, ορίζουμε ένα σύνολο background (background dataset), το οποίο αποτελείται από ένα σύνολο αντιπροσωπευτικών δειγματικών σημείων του συνόλου εκπαίδευσης του μοντέλου. Στη συνέχεια, το σύνολο αυτό χρησιμοποιείται για να καλύψει τις θέσεις των απουσιαζόντων τιμών των χαρακτηριστικών στις διάφορες διατάξεις. Έπειτα υπολογίζεται ο μέσος όρος των εξόδων όλων των συνθετικών δειγμάτων που δημιουργήθηκαν με χρήση του συνόλου background για κάθε μία από τις διατάξεις. Θεωρούμε το μέσο όρο των συνθετικών εξόδων της κάθε διάταξης ως την έξοδο της διάταξης, οι οποίες χρησιμοποιούνται για τον υπολογισμό βαρών και την αναγωγή του προβλήματος σε μία γραμμική παλινδρόμηση (Linear Regression). Τέλος, οι τιμές Shapley προκύπτουν ως οι συντελεστές του γραμμικού μοντέλου.

Ο Kernel SHAP αποτελεί μία από τις δημοφιλέστερες και γνωστότερες μεθόδους της SHAP, καθώς είναι model-agnostic και άρα εφαρμόζεται σε οποιοδήποτε μοντέλο μηχανικής μάθησης.

Tree SHAP

Η μέθοδος Tree SHAP προτάθηκε από τους Lundberg and Lee το 2018 [28] και αποτελεί μία model-specific εκδοχή της SHAP για μοντέλα μηχανικής μάθησης βασισμένα σε δέντρα (όπως δέντρα αποφάσεων, τυχαία δάση κλπ). Αποτελεί μία γρηγορότερη εναλλακτική για τον Kernel SHAP, όταν καλούμαστε να επεξηγήσουμε μοντέλα βασισμένα σε δέντρα.

Ο λόγος που η μέθοδος αυτή είναι ταχύτερη της Kernel είναι ότι βασίζεται στα βασικά γνωρίσματα των δέντρων. Όμως, η μέθοδος Tree SHAP ορίζει τη συνάρτηση αξίας χρησιμοποιώντας την υπό συνθήκη συνεισφορά αντί της οριακής. Το πρόβλημα που δημιουργεί η προσέγγιση αυτή, είναι ότι χαρακτηριστικά που δεν έχουν επιρροή στο μοντέλο μπορεί να προκύψουν με μη μηδενική συνεισφορά. Αυτό μπορεί να συμβεί, εάν ένα χαρακτηριστικό που δεν έχει επιρροή, έχει έντονη συσχέτιση με ένα χαρακτηριστικό που επηρεάζει τη πρόβλεψη του μοντέλου [29, 30]. Για το λόγο αυτό, εάν υπάρχουν χαρακτηριστικά με έντονες συσχετίσεις, δίνεται η δυνατότητα ορισμού συνόλου background, το οποίο «σπάει» τις εξαρτήσεις μεταξύ των χαρακτηριστικών σύμφωνα με τους κανόνες που υπαγορεύει η περιστασιακή συμπερασματολογία (casual inference) (Janzing et al. 2019) [29]. Παρέχοντας σύνολο background, έχουμε μείωση της ταχύτητας σε σχέση με όταν εκμεταλλεύεται απλά τα χαρακτηριστικά των δέντρων, αλλά εξακολουθεί να είναι ταχύτερη από την Kernel SHAP.

Κεφάλαιο 3

Σύνολα Δεδομένων και Μεθοδολογία

Στο κεφάλαιο αυτό θα μιλήσουμε για τα δεδομένα που χρησιμοποιήσαμε και τη μεθοδολογία [31] που ακολουθήσαμε κατά την εκτέλεση των πειραμάτων μας. Αρχικά, θα δούμε την προέλευση των δεδομένων μας, την προεπεξεργασία την οποία υπέστησαν και τα χαρακτηριστικά τα οποία επιλέξαμε να χρησιμοποιήσουμε. Έπειτα, θα δούμε τα εργαλεία που χρησιμοποιήθηκαν κατά τη διαδικασία εκπαίδευσης και αξιολόγησης των μοντέλων (binary και multiclass classification με χρήση του Random Forest Classifier). Τέλος, θα αναλύσουμε τα εργαλεία που επιλέχθηκαν για την ερμηνεία των μοντέλων, καθώς και τον τρόπο που ερμηνεύουμε την πληροφορία που παρέχουν οι οπτικοποιήσεις της SHAP.

3.1 Δεδομένα

Για τη διεξαγωγή των πειραμάτων ήταν απαραίτητο ένα σύνολο δεδομένων που περιέχει τόσο έγκυρα/καλόβουλα (legit/benign) domain names, όσο και κακόβουλα (malicious) και συγκεκριμένα DGA-generated ονόματα τομέα. Για την πρώτη κατηγορία τα δεδομένα αντλήθηκαν από το Tranco List [3], το οποίο πρόκειται για μία κατάταξη (ranking) του ενός εκατομμυρίου (top 1M) δημοφιλέστερων ιστότοπων, η οποία έχει βασιστεί στην έρευνα των Pochat et al. [32]. Τα αλγοριθμικά παραγόμενα ονόματα τα πήραμε από το DGArchive [4], το οποίο περιλαμβάνει πάνω από 100 DGA οικογένειες. Το πλήθος των ονομάτων για κάθε μία από τις οικογένειες αυτές ποικίλει, με κάποιες να έχουν μόλις κάποιες δεκάδες αναγνωρισμένα ονόματα και άλλες να έχουν δεκάδες εκατομμύρια. Επιπλέον, το αρχείο αυτό μας παρέχει πληροφορίες για τη διάρκεια ζωής των ονομάτων αυτών, καθώς όπως αναφέραμε και στην Ενότητα 2.2.2 τα ονόματα αυτά χρησιμοποιούνται από τους επιτιθέμενους για μικρό χρονικό διάστημα για την αποφυγή ανίχνευσης των C&C servers.

Από τη λίστα Tranco επιλέξαμε για το πείραμα μας τα εκατό χιλιάδες δημοφιλέστερα ονόματα, τα οποία ελέγξαμε για ύπαρξη διπλοτύπων και επιπλέον τα συγκρίναμε με τη λίστα των κακόβουλων αλγοριθμικά παραγόμενων ονομάτων για να επιβεβαιώσουμε πως δεν υπάρχουν κοινά ονόματα στις δύο λίστες.

Για τα αλγοριθμικά παραγόμενα ονόματα προμηθευτήκαμε από το DGArchive την τελευταία έκδοση, που περιέχει δεδομένα μέχρι και το τέλος του 2019. Η έκδοση αυτή περιλαμβάνει συνολικά 93 οικογένειες. Κάποιες από τις οικογένειες αποτελούνται από εξαιρετικά

μικρό πλήθος ονομάτων, όπως αναφέραμε και προηγουμένως, και στην περίπτωση της ταξινόμησης πολλών κλάσεων αυτό δημιουργεί μεγάλη ανισορροπία μεταξύ των κλάσεων. Για το λόγο αυτό και για λόγους ομοιογένειας μεταξύ των πειραμάτων ταξινόμησης δύο και πολλών κλάσεων, αγνοούμε τις οικογένειες με πλήθος ονομάτων μικρότερο των 3.000 και έτσι απομένουν 55 από τις 93 οικογένειες. Το νούμερο αυτό επιλέχθηκε μετά από ένα πλήθος δοκιμών, όπου φάνηκε ότι αγνοώντας τις κλάσεις με έως και 3.000 ονόματα βελτιώνει την απόδοση του μοντέλου, ενώ αγνοώντας κλάσεις με παραπάνω ονόματα δεν παρατηρήθηκε επιπλέον βελτίωση στην απόδοση του μοντέλου.

3.1.1 Binary Dataset

Για τη δημιουργία του συνόλου δεδομένων του μοντέλου ταξινόμησης δύο κλάσεων (binary classification) παίρνουμε 100.000 από τα έγκυρα ονόματα (top 100K από Tranco list) στα οποία βάζουμε ετικέτα μηδέν (0). Για τα κακόβουλα δεδομένα, επιλέγουμε τυχαία 10.000 ονόματα από κάθε μία από τις οικογένειες, εφόσον υπάρχουν, διαφορετικά παίρνουμε όλα τα ονόματα της οικογένειας (εάν έχει λιγότερα από 10.000). Στα κακόβουλα δεδομένα δίνουμε την ετικέτα ένα (1). Το σύνολο δεδομένων, χωρίζεται σε σύνολο εκπαίδευσης (train set) και σύνολο αξιολόγησης (test set). Το ποσοστό διαμέρισης που επιλέχθηκε είναι 80% για την εκπαίδευση και 20% για την αξιολόγηση. Στο σύνολο εκπαίδευσης εφαρμόζουμε υπερδειγματοληψία (oversampling) με χρήση της μεθόδου SMOTE (Ενότητα 3.1.3) για την εξισορρόπηση των δύο κλάσεων.

3.1.2 Multiclass Dataset

Για τη δημιουργία του συνόλου δεδομένων του μοντέλου ταξινόμησης πολλών κλάσεων (multiclass classification), επιλέγουμε 20.000 τυχαία ονόματα από τα 100K έγκυρα (top 100K από Tranco list), τα οποία τώρα έχουν ετικέτα «tranco». Για τις υπόλοιπες κλάσεις επιλέγουμε 20.000 τυχαία ονόματα από κάθε οικογένεια, εφόσον υπάρχουν, και όλα τα ονόματα της οικογένειας διαφορετικά, τα οποία έχουν ετικέτα το όνομα του αλγορίθμου DGA που αντιστοιχεί στη κάθε μία. Οι ετικέτες αυτές στη συνέχεια αντιστοιχίζονται στους αριθμούς 0 έως 54 (mapping), καθώς αυτό απαιτείται για την ορθή λειτουργία του μοντέλου. Χρησιμοποιούμε και πάλι τα ποσοστά διαμέρισης που χρησιμοποιήσαμε για το binary dataset (80%:training - 20%:test) και εφαρμόζουμε SMOTE oversampling στο σύνολο εκπαίδευσης.

3.1.3 SMOTE (Synthetic Minority Over-sampling Technique)

Κατά την εκπαίδευση μοντέλων μηχανικής μάθησης, ερχόμαστε συχνά αντιμέτωποι με μη-ισορροπημένα σύνολα δεδομένων, όπου μία ή περισσότερες κλάσεις έχουν σημαντικά μεγαλύτερο ή μικρότερο πλήθος δειγμάτων από τις υπόλοιπες. Αυτό μπορεί να δημιουργήσει πρόβλημα, καθώς πολλοί αλγόριθμοι μηχανικής μάθησης, όπως τα δέντρα αποφάσεων, μεροληπτούν προς τη κυρίαρχη κλάση και τείνουν να αγνοούν την κλάση που μειονεκτεί. Για το λόγο αυτό, δημιουργείται η ανάγκη εξισορρόπησης του συνόλου εκπαίδευσης, ώστε να αποφεύγεται η δημιουργία προκαταλήψεων από το μοντέλο.

Οι τεχνικές που χρησιμοποιούνται για την εξισορρόπηση συνόλων δεδομένων χωρίζονται σε δύο βασικές κατηγορίες:

- **Υπερδειγματοληψία (Oversampling)**: Αυξάνει το πλήθος των δειγμάτων της κλάσης που μειονεκτεί (π.χ. Random Oversampling, SMOTE oversampling).
- **Υποδειγματοληψία (Downsampling)**: Μειώνει το πλήθος των δειγμάτων που υπερτερεί (π.χ. Random Downsampling, Cluster-based downsampling).

Στην υλοποίηση μας επιλέξαμε να κάνουμε υπερδειγματοληψία για να μη χαθεί χρήσιμη πληροφορία από τη κυριαρχούσα κλάση. Η πιο απλή μέθοδος υπερδειγματοληψίας είναι η τυχαία υπερδειγματοληψία, όπου δημιουργεί τυχαία διπλότυπα των δειγμάτων της μικρότερης κλάσης, το οποίο μπορεί να οδηγήσει στην υπερπροσαρμογή (overfitting) του μοντέλου. Αυτό, λοιπόν, μπορεί να αποφευχθεί εφαρμόζοντας υπερδειγματοληψία με χρήση της μεθόδου SMOTE (Synthetic Minority Over-sampling Technique) [33] που χρησιμοποιήσαμε στην υλοποίηση μας.

Η μέθοδος SMOTE παράγει νέα συνθετικά δειγματικά σημεία για την υπερδειγματοληψία της μικρότερης κλάσης. Τα συνθετικά αυτά δείγματα δημιουργούνται ως εξής: Επιλέγεται ένα τυχαίο δειγματικό σημείο O από την κλάση που μειονεκτεί και βρίσκει τους k-κοντινότερους γείτονες (k-nearest neighbours) του O που ανήκουν στην ίδια κλάση. Στη συνέχεια, το O συνδέεται με μία ευθεία γραμμή με τους γείτονες αυτούς, και με χρήση ενός τυχαίου παράγοντα κλιμάκωσης (scaling factor) $z \in [0, 1]$, τοποθετείται ένα νέο σημείο σε απόσταση $z \cdot 100\%$ από το δείγμα O για κάθε μία από τις ευθείες. Τα σημεία αυτά, είναι τα νέα συνθετικά δείγματα. Η διαδικασία επαναλαμβάνεται μέχρι να επιτευχθεί το επιθυμητό πλήθος συνθετικών δειγμάτων.

Στην υλοποίηση μας χρησιμοποιούμε τη συνάρτηση SMOTE της βιβλιοθήκης imbalanced-learn [34] με τις default τιμές των παραμέτρων (k_neighbors=5).

3.1.4 Χρονολογικός διαχωρισμός δεδομένων

Αναφέραμε προηγουμένως, πως το DGArchive μας παρέχει πληροφορίες για τη διάρκεια ζωής των ονομάτων που περιέχει. Στο αρχείο περιλαμβάνονται ονόματα από το 2010 έως και το 2019. Δημιουργήσαμε, λοιπόν, δέκα νέα σύνολα δεδομένων, όπου το καθένα από αυτά περιέχει ονόματα τα οποία χρησιμοποιήθηκαν τις χρονιές 2010 έως 2019, καθώς και ένα τυχαίο υποσύνολο έγκυρων ονομάτων από το Tranco List ανεξάρτητα από το έτος. Τα σύνολα αυτά, τα χρησιμοποιήσαμε για να ελέγξουμε την ανθεκτικότητα των μοντέλων μας (binary και multiclass) ανά τα χρόνια. Για το σκοπό αυτό, εκτελέσαμε κάποια πειράματα και για τα δύο μοντέλα, στα οποία τα εκπαιδεύσαμε μόνο με δεδομένα που χρονολογούνται το 2010 και κατόπιν τα αξιολογήσαμε τόσο με την ίδια τη χρονιά (training set: 2010 - test set:2010), όσο και με τις ακόλουθες χρονιές (training set:2010 - test set: 2011-2019).

Κατά τον διαχωρισμό των δεδομένων, παρατηρήσαμε πως 10 από τις 55 οικογένειες εμφανίζονται και στις δέκα χρονολογίες (με διαφορετικά ονόματα κάθε χρονιά). Οι υπόλοιπες

οικογένειες εμφανίζονται σταδιακά τις ακόλουθες χρονιές (μετά το 2010). Το 2015 παρατηρήθηκε η μεγαλύτερη αύξηση εμφάνισης νέων οικογενειών DGA. Για κάθε έτος έχουν συμπεριληφθεί όλες οι οικογένειες για τις οποίες έχουμε δεδομένα για το έτος αυτό.

3.2 Εξαγωγή Χαρακτηριστικών (Feature Extraction)

Όπως αναφέραμε και στην Ενότητα 2.3, τα μοντέλα ανίχνευσης κακόβουλης κίνησης τα τελευταία χρόνια βασίζονται κυρίως σε χαρακτηριστικά (features) που προκύπτουν από τα ονόματα τομέα. Αυτή η προσέγγιση διευκολύνει την ανίχνευση σε πραγματικό χρόνο (real-time detection) καθώς δεν απαιτείται ανάλυση δεδομένων από μεγάλα χρονικά παράθυρα, όπως υλοποιήσεις που βασίζονται στον εντοπισμό αυξημένου πλήθους NXDomain Responses που συνήθως οφείλεται σε επιθέσεις από DGA-based botnets, μπορούμε δηλαδή να ταξινομήσουμε ένα όνομα κάθε φορά (κατόπιν εκπαίδευσης του μοντέλου).

Στην υλοποίηση της παρούσας διπλωματικής, λοιπόν, επιλέχθηκαν χαρακτηριστικά που βασίζονται αποκλειστικά στα domain names, καθώς είναι μία προσέγγιση για την οποία δεν απαιτείται χρονοβόρα πρόσβαση σε εξωτερικές βάσεις δεδομένων και ταυτόχρονα δεν υπάρχουν προβλήματα με πρόσβαση σε ευαίσθητη πληροφορία. Συνολικά επιλέξαμε 50 χαρακτηριστικά [35], τα οποία στην πλειοψηφία τους είναι αρκετά απλά τόσο στον υπολογισμό τους όσο και στη κατανόηση τους, αλλά παρέχουν χρήσιμη πληροφορία για τη φύση των ονομάτων.

Στον Πίνακα 3.1 βλέπουμε τα 50 χαρακτηριστικά που χρησιμοποιήσαμε τόσο για τη ταξινόμηση δύο κλάσεων, όσο και για τη ταξινόμηση πολλαπλών κλάσεων, μαζί με μία σύντομη περιγραφή. Οι τιμές των χαρακτηριστικών αυτών, υπολογίζονται κατόπιν αφαίρεσης του Top-Level-Domain (TLD) («.com», «.gr», «.gov» κ.α.) [36]. Τα TLDs δεν είναι αλγοριθμικά παραγόμενα και άρα δεν παρέχουν επιπλέον χρήσιμη πληροφορία στην εκπαίδευση των μοντέλων ταξινόμησης. Η αναγνώριση των έγκυρων επιθεμάτων (suffix) DNS (με σκοπό την αφαίρεση τους) γίνεται με χρήση της δημόσιας λίστας επιθεμάτων του Mozilla (Mozilla public suffix list) [37].

Ας αναλύσουμε, όμως, περαιτέρω τα χαρακτηριστικά 46, 47, 48 και 50, των οποίων ο υπολογισμός αλλά και η χρησιμότητα δεν είναι τόσο ξεκάθαρα.

- **Max_Gap (feature 46)**: Στην ουσία το χαρακτηριστικό αυτό μας δίνει το μέγιστο μήκος των labels που μεσολαβούν (εάν υπάρχουν) μεταξύ του ονόματος και του suffix (ή των suffixes που έχουν αφαιρέσει). Για παράδειγμα αν έχουμε το όνομα «example.for.maxgap.com» μετά την αφαίρεση του suffix «.com» έχουμε τις «ενδιάμεσες» ετικέτες «for» και «maxgap» και άρα η τιμή του χαρακτηριστικού 46 θα ήταν 6 στην περίπτωση αυτή.
- **Reputation (feature 47)**: Χρησιμοποιούμε το χαρακτηριστικό αυτό για την αξιολόγηση της νομιμότητας ενός ονόματος τομέα. Όσο μεγαλύτερη η τιμή του Reputation, τόσο πιθανότερο να είναι νόμιμο το όνομα. Η μέθοδος υπολογισμού του Reputation

No	**Feature Name**	**Περιγραφή**
1	Length	Το μήκος του domain name
2	Max_DeciDig_Seq	Το μήκος της μέγιστης ακολουθίας δεκαδικών ψηφίων του domain name
3	Max_Let_Seq	Το μήκος της μέγιστης ακολουθίας λατινικών γραμμάτων του domain name
4-29	Freq_A - Freq_Z	Η συχνότητα εμφάνισης των λατινικών γραμμάτων (A-Z) στο domain name
30-39	Freq_0 - Freq_9	Η συχνότητα εμφάνισης των δεκαδικών ψηφίων (0-9) στο domain name
40	Spec_Char_Freq	Το πλήθος των χαρακτήρων «-» και «.» στο domain name
41	Ratio_Spec_Char	Ο λόγος του Spec_Char_Freq με το μήκος του domain name
42	DeciDig_Freq	Το συνολικό πλήθος δεκαδικών ψηφίων στο domain name
43	Ratio_DeciDig	Ο λόγος του DeciDig_Freq με το μήκος του domain name
44	Vowel_Freq	Το πλήθος των φωνηέντων στο domain name
45	Vowel_Ratio	Ο λόγος του Vowel_Freq με το μήκος του domain name
46	Max_Gap	Το μήκος της μεγαλύτερης ετικέτας label του domain name
47	Reputation	Το πλήθος των whitelisted N-grams (N=3,..,7) στο domain name
48	Words_Freq	Το πλήθος των υπαρκτών λέξεων στο domain name
49	Words_Mean	Το μέσο μήκος των λέξεων του παραπάνω χαρακτηριστικού
50	Entropy	Η εντροπία του domain name

Πίνακας 3.1: *Πίνακας Χαρακτηριστικών (features)*

score ενός domain name που χρησιμοποιήσαμε βασίζεται στη συχνότητα εμφάνισης N-grams (ακολουθίες N συνεχόμενων χαρακτήρων) τα οποία υπάρχουν σε έγκυρα ονόματα και απουσιάζουν από κακόβουλα ονόματα. Για τον υπολογισμό αυτού του χαρακτηριστικού απαιτείται η κατασκευή ενός whitelist με N-grams που προκύπτουν από ένα σύνολο έγκυρων ονομάτων (στην περίπτωση μας από το Tranco List). Οπότε για τον υπολογισμό του Reputation ενός domain name συγκρίνουμε πόσα από τα N-grams του, περιλαμβάνονται στο whitelist που κατασκευάσαμε. Οι τιμές του N που επιλέξαμε είναι μεταξύ του 3 και του 7, έχουμε δηλαδή αγνοήσει τα unigrams (N=1) και τα bigrams (N=2), καθώς πολλά από αυτά συναντώνται τόσο σε καλόβουλα όσο και σε κακόβουλα ονόματα, με αποτέλεσμα να επηρεάζει τη διαδικασία εκμάθησης του μοντέλου ταξινόμησης [35].

- **Words_Freq (feature 48)**: Το χαρακτηριστικό αυτό υπολογίζει το πλήθος των υπαρκτών λέξεων εντός ενός domain name. Τις λέξεις αυτές τις αντλούμε χρησιμοποιώντας ένα εργαλείο επεξεργασίας φυσικής γλώσσας (Natural Language Processing - NLP). Το εργαλείο αυτό ονομάζεται Wordninja [38] και διαχωρίζει με πιθανοτικό τρόπο συμβολοσειρές σε λέξεις με βάση τη συχνότητα εμφάνισης unigram των λέξεων που εμφανίζονται στην αγγλική Βικιπαίδεια. Λέξεις με λιγότερους από 3 χαρακτήρες (όπως αντωνυμίες και άρθρα) αγνοούνται καθώς δεν συνεισφέρουν στην διαδικασία εκπαίδευσης.

- **Entropy (feature 50)**: Το χρησιμοποιούμε για να εκτιμήσουμε την τυχαιότητα ενός domain name. Για τον υπολογισμό του χαρακτηριστικού χρησιμοποιήσαμε τον τυπικό ορισμό της εντροπίας Shannon:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x) \tag{3.1}$$

όπου:

X: το σύνολο των χαρακτήρων το domain name και

$p(x)$: η συχνότητα εμφάνισης του χαρακτήρα $x \in X$.

Πριν την έναρξη της διαδικασίας εκπαίδευσης του μοντέλου, ελέγχουμε αν υπάρχει έντονη συσχέτιση μεταξύ των χαρακτηριστικών και επιπλέον κανονικοποιούμε τις τιμές τους.

3.2.1 Έλεγχος συσχέτισης χαρακηριστικών (feature correlation)

Για να εντοπίσουμε τυχόν περιττά χαρακτηριστικά που δε συμβάλλουν στη διαδικασία εκμάθησης του μοντέλου, υπολογίζουμε τις κατά ζεύγη συσχετίσεις των χαρακτηριστικών με χρήση του συντελεστή συσχέτισης Pearson (Pearson's Correlation Coefficient - PCC) [39]. Για κάθε πιθανό ζεύγος χαρακτηριστικών, υπολογίζουμε το συντελεστή συσχέτισης Pearson και εάν αυτός ξεπερνάει ένα προκαθορισμένο κατώφλι (το οποίο έχουμε θέσει ίσο με 0.9) [40] για κάποιο ζεύγος, τότε επιλέγεται τυχαία ένα από τα δύο χαρακτηριστικά του ζεύγους αυτού και διαγράφεται από το σύνολο δεδομένων.

Συγκεκριμένα, κατόπιν υπολογισμού των παραπάνω προέκυψε ότι μόνο το χαρακτηριστικό «Ratio_DeciDig» συσχετίζεται έντονα με άλλα χαρακτηριστικά και ως εκ τούτου αφαιρέθηκε από το σύνολο δεδομένων που χρησιμοποιήσαμε στα ακόλουθα πειράματα.

3.2.2 Κανονικοποίηση δεδομένων (Data Scaling)

Η κανονικοποίηση των δεδομένων (data scaling) είναι χρήσιμη όταν τα χαρακτηριστικά είναι σε διαφορετικές κλίμακες, καθώς κάποια χαρακτηριστικά μπορεί να κυριαρχήσουν κατά την διαδικασία εκμάθησης του μοντέλου, απλά και μόνο επειδή η κλίμακα τους είναι μεγαλύτερη και όχι επειδή είναι εκ των πραγμάτων σημαντικά. Έτσι κανονικοποιώντας τις τιμές των χαρακτηριστικά, διασφαλίζεται η δίκαιη συμβολή των χαρακτηριστικών στην εκπαίδευση του μοντέλου.

Μία από τις δημοφιλέστερες μεθόδους κανονικοποίησης (και αυτή που χρησιμοποιήθηκε στην παρούσα διπλωματική) αποτελεί το Min-Max Scaling (Normalization). Η μέθοδος αυτή μετατρέπει όλα τα χαρακτηριστικά ώστε να έχουν την ίδια κλίμακα, μεταξύ 0 και 1. Αυτό επιτυγχάνεται εφαρμόζοντας την ακόλουθη σχέση τόσο στα δεδομένα του training όσο και στου test set:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3.2}$$

όπου:
X: η τιμή που κανονικοποιούμε,
X': η κανονικοποιημένη τιμή που προκύπτει,
X_{min}: η ελάχιστη τιμή του χαρακτηριστικού και
X_{max}: η μέγιστη τιμή του χαρακτηριστικού.

Άλλες μέθοδοι κανονικοποίησης είναι οι Robust Scaling και Log Transformation, οι οποίες συνήθως επιλέγονται όταν τα δεδομένα μας περιέχουν outliers και η Standard Scaling, η οποία προτιμάται όταν χρησιμοποιούμε αλγορίθμους που θεωρούν πως τα δεδομένα είναι κεντραρισμένα γύρω από το μηδέν (π.χ Support Vector Machine, Linear Regression). Επιλέξαμε να κανονικοποιήσουμε τα δεδομένα μας με τη μέθοδο Min-Max καθώς η μέθοδος αυτή διατηρεί αναλλοίωτη την αρχική κατανομή των δεδομένων και επιπλέον τα δεδομένα μας δεν περιμένουμε να έχουν ακραίες τιμές, ούτε υλοποιούμε κάποιον αλγόριθμο που να απαιτεί κανονικοποίηση γύρω από το μηδέν.

3.3 Random Forest Classifier

Το μοντέλο που επιλέξαμε για την ταξινόμηση των ονομάτων σε κακόβουλα (DGA-generated) και μη domain names είναι ο Random Forest Classifier. Πρόκειται για έναν tree-based αλγόριθμο τον οποίο αναλύσαμε και στην Ενότητα 2.4.3. Συνοπτικά, ο αλγόριθμος αυτός κατασκευάζει ένα σύνολο τυχαίων δέντρων αποφάσεων (τυχαίο δάσος). Στη συνέχεια, το καθένα από αυτά προβλέπει τη κλάση ενός δείγματος και η κλάση στην οποία τελικά ταξινομείται το δείγμα, αποφασίζεται μέσω ψηφοφορίας (η κλάση που προέβλεψε η πλειοψηφία των δέντρων, είναι η κλάση που τελικά επιλέγεται).

Ο αλγόριθμος υλοποιήθηκε με τη βιβλιοθήκη scikit-learn [41, 42]. Για την επιλογή των παραμέτρων n_estimators (δηλαδή το πλήθος των δέντρων που απαρτίζουν το δάσος) και max_depth (δηλαδή το μέγιστο βάθος που μπορούν να έχουν τα προηγούμενα δέντρα) εκτελέσαμε Grid Search και τελικά επιλέξαμε 50 δέντρα με μέγιστο βάθος 100 για τη ταξινόμηση δύο κλάσεων (binary classification) και 100 δέντρα με μέγιστο βάθος 100 για την ταξινόμηση πολλών κλάσεων (multiclass classification). Για τις υπόλοιπες παραμέτρους, κρατήσαμε τις προεπιλεγμένες τιμές του scikit-learn. Ενδεικτικά, η προεπιλογή της συνάρτησης για το κριτήριο αξιολόγησης των διακλαδώσεων για τα χαρακτηριστικά είναι ο δείκτης Gini (2.4.2). Η επιλογή των τιμών αυτών για τις παραμέτρους έγινε γιατί ταυτόχρονα έχουμε μία ικανοποιητική απόδοση τόσο για το μοντέλο ταξινόμησης δύο κλάσεων όσο και για το μοντέλο πολλών κλάσεων, χωρίς να είναι χρονικά και υπολογιστικά ιδιαίτερα απαιτητικό.

3.4 Αξιολόγηση μοντέλων ταξινόμησης

3.4.1 Πίνακας Σύγχυσης (Confusion Matrix)

Για να μπορέσουμε να ορίσουμε τις μετρικές που χρησιμοποιήσαμε για την αξιολόγηση των μοντέλων ταξινόμησης, πρέπει αρχικά να ορίσουμε τον πίνακα σύγχυσης. Ο πίνακας

σύγχυσης (confusion matrix), πρόκειται για έναν πίνακα που εκφράζει την απόδοση ενός αλγορίθμου ταξινόμησης, μας δείχνει δηλαδή πόσες από τις τιμές ταξινομήθηκαν στη κλάση που έπρεπε και πόσες έχουν ταξινομηθεί σε λάθος κλάση.

Ο πίνακας σύγχυσης ενός μοντέλου ταξινόμησης δύο κλάσεων έχει την ακόλουθη μορφή:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Πίνακας 3.2: *Πίνακας σύγχυσης ταξινόμησης δύο κλάσεων*

όπου:

TP: σωστά ταξινομημένα δειγματικά σημεία DGA

FP: έγκυρα δειγματικά σημεία τα οποία ταξινομήθηκαν ως DGA

TN: σωστά ταξινομημένα έγκυρα δειγματικά σημεία

FN: DGA δειγματικά σημεία τα οποία ταξινομήθηκαν ως έγκυρα

Αντίστοιχα, για ένα μοντέλο ταξινόμησης πολλών κλάσεων, ο πίνακας σύγχυσης προκύπτει ως επέκταση του Πίνακα 3.2 και έχει την ακόλουθη μορφή για ένα πρόβλημα ταξινόμησης 3 κλάσεων:

	Predicted Class 1	Predicted Class 2	Predicted Class 3
Actual Class 1	*TP*	*FP*	*FP*
Actual Class 2	*FN*	*TP*	*FP*
Actual Class 3	*FN*	*FN*	*TP*

Πίνακας 3.3: *Πίνακας Σύγχυσης για ταξινόμηση πολλών κλάσεων*

3.4.2 Μετρικές αξιολόγησης

Για την αξιολόγηση των μοντέλων ταξινόμησης, που θα δούμε αναλυτικά στο Κεφάλαιο 4, χρησιμοποιούμε τέσσερεις μετρικές, οι οποίες αποτελούν από τις συνηθέστερες μετρικές αξιολόγησης στον τομέα της Μηχανικής Μάθησης. Αναλυτικά, οι μετρικές αυτές είναι οι ακόλουθες:

- **Accuracy**: Πρόκειται για τον λόγο των δειγμάτων που έχουν ταξινομηθεί στη σωστή κλάση προς το συνολικό πλήθος όλων των δειγμάτων. Η μετρική αυτή εκφράζει τη συνολική ορθότητα του μοντέλου.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Instances}} \tag{3.3}$$

- **Precision**: Η μετρική αυτή είναι επίσης γνωστή και ως (Positive Predicted Value), καθώς εκφράζει πόσα από τα δείγματα που ταξινομήθηκαν ως Positive άνηκαν στην πραγματικότητα στην κλάση αυτή.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.4}$$

- **Recall**: Η μετρική αυτή είναι επίσης γνωστή ως Sensitivity ή True Positive Rate γιατί εκφράζει πόσα από τα δείγματα που ανήκουν στην κλάση Positive ταξινομήθηκαν όντως στη κλάση αυτή.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.5}$$

- **F1 Score**: Πρόκειται για τον αρμονικό μέσο όρο των μετρικών Precision και Recall. Μας δίνει μία ισορροπημένη εικόνα των απόψεων των δύο παραπάνω μετρικών.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

3.5 Εξαγωγή Επεξηγήσεων

Για την ανάλυση και επεξήγηση των μοντέλων ταξινόμησης των DGA-generated domain names χρησιμοποιήσαμε τη βιβλιοθήκη SHAP [43], της οποίας τη μεθοδολογία είδαμε αναλυτικά στην Ενότητα 2.5.3. Όπως έχουμε αναφέρει και προηγουμένως, πρόκειται για μία model-agnostic μέθοδο η οποία παρέχει τόσο τοπικές όσο και καθολικές επεξηγήσεις για μοντέλα μαύρου κουτιού. Στην παρούσα διπλωματική, χρησιμοποιούμε μία model-specific εκδοχή της SHAP, την TreeSHAP (2.5.3), για εξοικονόμηση υπολογιστικού χρόνου, καθώς τα μοντέλα μας είναι βασισμένα σε δέντρα.

3.5.1 Tree Explainer

Για την επεξήγηση των μοντέλων, είναι απαραίτητη η χρήση ενός explainer. Η SHAP, παρέχει πληθώρα επιλογών (όπως KernelExplainer, TreeExplainer, DeepExplainer κ.α.) με τον KernelExplainer να είναι ο δημοφιλέστερος, καθώς εφαρμόζεται σε όλους τους τύπους μοντέλων (model-agnostic). Όπως αναφέραμε και προηγουμένως, εμείς θα χρησιμοποιήσουμε τον TreeExplainer [44], ο οποίος ειδικεύεται σε μοντέλα βασισμένα σε δέντρα, όπως ο RandomForestClassifier που υλοποιούμε στη παρούσα εργασία.

Η μοναδική παράμετρος που απαιτείται για την κατασκευή του TreeExplainer είναι το εκπαιδευμένο μοντέλο που καλείται να επεξηγήσει. Όπως αναφέραμε, όμως, και στην Ενότητα 2.5.3 υπάρχει η δυνατότητα παροχής συνόλου background (eXplainability Background

Instances (XBIs)) με σκοπό την εξάλειψη οποιασδήποτε συσχέτισης μπορεί να υπάρχει μεταξύ των χαρακτηριστικών. Παρότι έχουμε ελέγξει τα δεδομένα μας για ύπαρξη συσχετίσεων, επιλέγουμε αυτή τη προσέγγιση, για να έχουμε το αντιπροσωπευτικότερο αποτέλεσμα για τις επεξηγήσεις του μοντέλου μας. Το background dataset κατασκευάζεται με εφαρμογή του αλγορίθμου K-means Clustering για K ίσο με 100 στα δεδομένα εκπαίδευσης. Τα 100 δείγματα αποτελεί ένα αποδεκτό και ικανοποιητικό μέγεθος για το σύνολο background δεδομένου του μεγέθους του συνόλου δεδομένων μας. Δοκιμές με περισσότερα δείγματα δεν φάνηκαν να επηρεάζουν ιδιαίτερα τις επεξηγήσεις. Η επιλογή των δειγμάτων αυτών με χρήση του αλγορίθμου K-means Clustering (2.4.4) είναι μία συνήθης τακτική καθώς θέλουμε το σύνολο αυτό να αποτελεί ένα αντιπροσωπευτικό δείγμα του συνόλου δεδομένων μας, το οποίο επιτυγχάνεται παίρνοντας τα κέντρα των 100 συστάδων που προκύπτουν από τον αλγόριθμο K-means.

Για τις επεξηγήσεις απαιτείται ο υπολογισμός των SHAP values ενός συνόλου δειγμάτων (eXplainability Test Instances - XTIs) που αυτή τη φορά επιλέγουμε τυχαία από το test set. Με τη βοήθεια του explainer υπολογίζονται οι τιμές SHAP για τα δειγματικά σημεία αυτά, τα οποία στη συνέχεια χρησιμοποιούμε για την δημιουργία διαγραμμάτων που προσφέρονται από τη SHAP και είναι απαραίτητα για τη διαδικασία ερμηνείας του μοντέλου μας. Η SHAP προσφέρει πληθώρα διαγραμμάτων και οπτικοποιήσεων για να διευκολυνθεί η κατανόηση των ερμηνειών. Από αυτές έχουμε επιλέξει τα SHAP Summary Plots (Bar plots και Beeswarm Plots) για την ερμηνεία του μοντέλου στο σύνολο του και τα SHAP Force Plots για την ερμηνεία μεμονωμένων δειγματικών σημείων.

3.5.2 Καθολικές επεξηγήσεις (global explanations) με SHAP Summary Plots

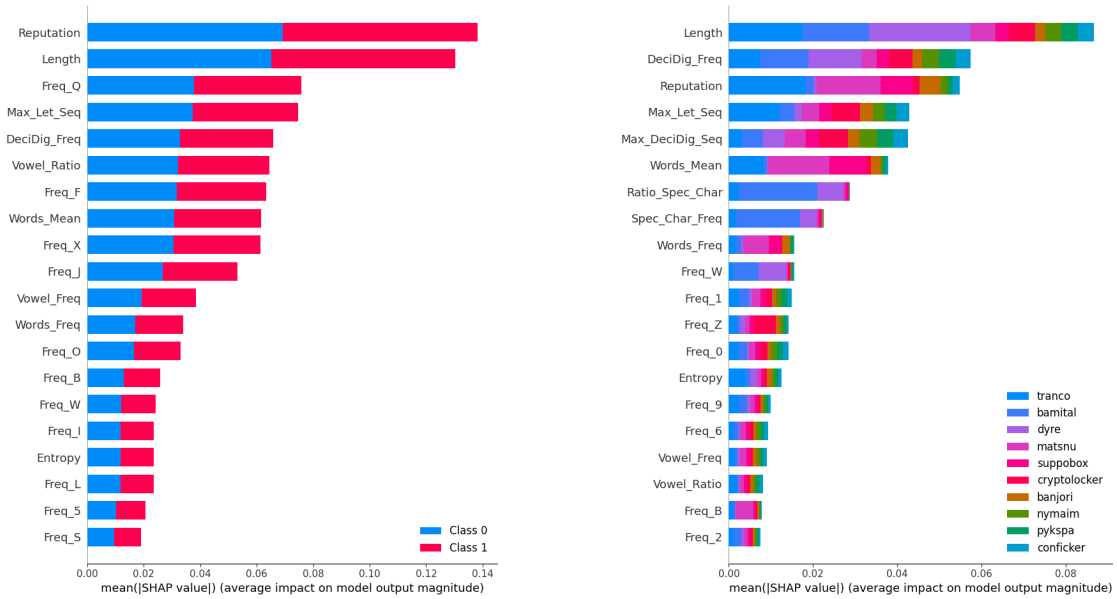
Για να μπορέσουμε να έχουμε μία συνολική εικόνα της επιρροής των χαρακτηριστικών στο μοντέλο μας, χρησιμοποιήσαμε τα SHAP Summary Plots [45]. Η συνάρτηση αυτή μας δίνει την επιλογή να κατασκευάσουμε δύο τύπους διαγραμμάτων, Bar Plot και Beeswarm Plot, τα οποία μας δίνουν μία οπτικοποίηση της κατάταξης των χαρακτηριστικών που ασκούν τη μεγαλύτερη επιρροή στο μοντέλο.

Bar Plot

Το Bar Plot, πρόκειται για μία αρκετά απλή και γενική οπτικοποίηση, η οποία μας προσφέρει μία συνολική εικόνα για το ποια χαρακτηριστικά επηρεάζουν περισσότερο το μοντέλο. Λαμβάνει υπόψη την απόλυτη τιμή των SHAP values, οπότε στην οπτικοποίηση αυτή δεν μπορούμε να δούμε με ποιο τρόπο το εκάστοτε χαρακτηριστικό επηρεάζει την απόφαση του μοντέλου (αν δηλαδή έχει θετική ή αρνητική επιρροή στο μοντέλο).

Στο Σχήμα 3.1 βλέπουμε τα bar plots των δύο μοντέλων μας (δύο κλάσεων και πολλών κλάσεων), τα αποτελέσματα των οποίων θα σχολιαστούν αναλυτικά στο επόμενο Κεφάλαιο.

Όπως μπορούμε να δούμε πρόκειται για μία φθίνουσα κατάταξη των 20 (default τιμή της συνάρτησης) πιο σημαντικών για το μοντέλο χαρακτηριστικών, με το πρώτο να έχει τη μεγαλύτερη επιρροή και το τελευταίο τη μικρότερη. Επιπλέον, μπορούμε να δούμε πως



(α΄) Binary Bar Plot

(β΄) Multiclass Bar Plot

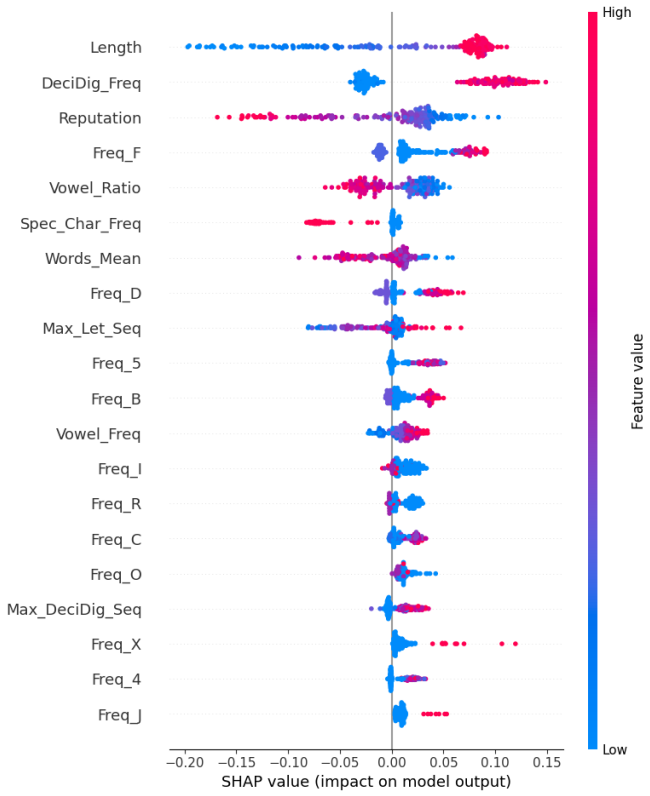
Σχήμα 3.1: *Bar Plots Examples*

κάθε κλάση αντιστοιχεί σε ένα χρώμα, το οποίο μας δίνει επιπλέον πληροφορία για το πόσο κάθε χαρακτηριστικό επηρεάζει μία συγκεκριμένη κλάση. Στην περίπτωση της ταξινόμησης 2 κλάσεων οι μπάρες είναι ισομοιρασμένες στις δύο κλάσεις καθώς όσο ένα χαρακτηριστικό επηρεάζει το μοντέλο να προβλέψει τη μία κλάση τόσο το επηρεάζει στο να μη προβλέψει την άλλη κλάση και καθώς στο συγκεκριμένο διάγραμμα λαμβάνονται υπόψη οι απόλυτες τιμές των SHAP values οι δύο αυτές επιρροές είναι ίσες. Τέλος, όσον αφορά τη ταξινόμηση πολλών κλάσεων, έχουμε επιλέξει να συμπεριλάβουμε στο διάγραμμα αυτό την οπτικοποίηση 10 μόνο κλάσεων, καθώς η οπτικοποιήση και των 55 κλάσεων θα είχε τόση πληροφορία που θα την καθιστούσε αδύνατη να ερμηνευθεί. Η επιλογή των 10 αυτών κλάσεων έγινε επειδή είναι οι μόνες από τις 55 κλάσεις που γνωρίζουμε ότι ανήκουν σε μία από τις 4 κατηγορίες DGA (Arithmetic-based, Hash-based, Wordlist-based και Permutation-based). Γνωρίζοντας, λοιπόν, την κατηγορία στην οποία ανήκουν, διευκολύνεται η διαδικασία ερμηνείας του μοντέλου, καθώς είναι ευκολότερο να κατανοήσουμε γιατί ένα χαρακτηριστικό επηρέασε το μοντέλο να επιλέξει την κλάση αυτή ή το αντίθετο.

Beeswarm Plot

Στο Beeswarm Plot έχουμε και πάλι μία κατάταξη των 20 πιο σημαντικών χαρακτηριστικών του μοντέλου, αλλά επιπλέον μπορούμε να δούμε πως οι διάφορες τιμές των χαρακτηριστικών αυτών επηρεάζουν τη πρόβλεψη του μοντέλου. Σε αυτό το διάγραμμα, κάθε κουκίδα αντιπροσωπεύει ένα δειγματικό σημείο (από τα XTIs) και τοποθετείται ανάλογα με το πόσο θετικά ή αρνητικά επηρεάζει το μοντέλο. Στον οριζόντιο άξονα δηλαδή, έχουμε τις τιμές SHAP και οι θετικές τιμές επηρεάζουν το μοντέλο θετικά, δηλαδή τείνουν να αυξάνουν την έξοδο του μοντέλου, το οποίο στην περίπτωση της ταξινόμησης δύο κλάσεων σημαίνει ότι τείνει προς την τιμή 1 (DGA domain name). Άρα όσο μεγαλύτερη η τιμή SHAP, τόσο περισ-

σότερο έχει συμβάλει το χαρακτηριστικό αυτό στο να επιλεχθεί η κλάση 1 και το αντίθετο.



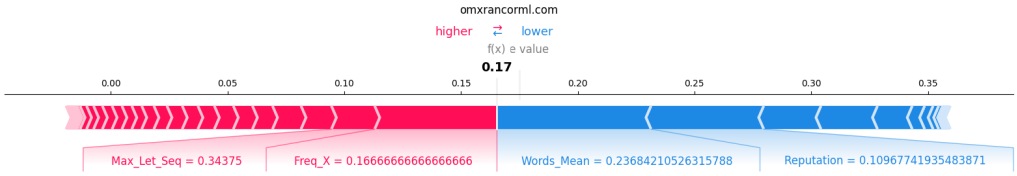
Σχήμα 3.2: *Beeswarm Plot Example*

Επίσης, στο Σχήμα 3.2 βλέπουμε πως στα δεξιά υπάρχει μία παλέτα χρωμάτων όπου τα ψυχρά χρώματα αντιπροσωπεύουν τις χαμηλότερες τιμές των χαρακτηριστικών και τα θερμότερα τις υψηλότερες τιμές των χαρακτηριστικών. Ο συνδυασμός, λοιπόν, των τιμών SHAP στον οριζόντιο άξονα και των χρωμάτων που εκφράζουν τις τιμές των χαρακτηριστικών, μας βοηθάνε να ερμηνεύσουμε πως επηρεάζονται οι προβλέψεις ανάλογα με τις τιμές των χαρακτηριστικών. Όσον αφορά τη ταξινόμηση πολλών κλάσεων, τα beeswarm plots προκύπτουν ανάγοντας το πρόβλημα σε ταξινόμηση δύο κλάσεων. Προφανώς, ένα τέτοιο διάγραμμα θα ήταν αδύνατο να ερμηνευθεί όταν αντιμετωπίζουμε ένα πρόβλημα με μεγάλο αριθμό κλάσεων. Για το λόγο αυτό, τα beeswarm plots είναι ξεχωριστά για τη κάθε μία κλάση, όπου οι μεγαλύτερες τιμές SHAP εκφράζουν ότι το μοντέλο τείνει να προβλέψει τη κλάση αυτή και οι μικρότερες τιμές ότι τείνει να μην προβλέψει την κλάση. Είναι σαν να έχουμε δηλαδή πολλά προβλήματα ταξινόμησης δύο κλάσεων, όπου κάθε φορά οι δύο κλάσεις είναι «Κλάση x» και «Όχι κλάση x».

3.5.3 Τοπικές επεξηγήσεις (local explanations) με SHAP Force Plots

Για την ερμηνεία μεμονωμένων δειγματικών σημείων (τοπικές επεξηγήσεις) χρησιμοποιήσαμε τα SHAP Force Plots [46]. Σε αυτό τον τύπο διαγράμματος βλέπουμε την επιρροή των χαρακτηριστικών στην πρόβλεψη ενός συγκεκριμένου δειγματικού. Στα αριστερά με κόκκινο χρώμα βλέπουμε τα χαρακτηριστικά που είχαν θετική επιρροή, ενώ στα δεξιά με μπλε αυτά που είχαν αρνητική επιρροή ως προς μία συγκεκριμένη κλάση. Η τιμή που βλέπουμε με

έντονο μαύρο χρώμα είναι η τιμή που έχει προβλέψει το μοντέλο για το δεδομένο δείγμα.



Σχήμα 3.3: *Force Plot Example*

Στο Σχήμα 3.3 βλέπουμε το force plot ενός κακόβουλου ονόματος που όμως έχει ταξινομηθεί ως έγκυρο. Το domain name αναγράφεται στην κορυφή του σχήματος (omxrancorml.com). Τα χαρακτηριστικά που άσκησαν τη μεγαλύτερη επιρροή στην απόφαση αντιστοιχούν στα μεγαλύτερα βέλη του Σχήματος. Συγκεκριμένα, τα χαρακτηριστικά Max_Let_Seq και Freq_X είχαν τη μεγαλύτερη θετική επιρροή στο να επιλέξουν τη κλάση 0 (έγκυρα ονόματα), ενώ τα χαρακτηριστικά Words_Mean και Reputation είχαν τη μεγαλύτερη αρνητική επιρροή.

Η τιμή που βλέπουμε στο διάγραμμα αυτό, πρόκειται για μία raw τιμή που επιστρέφει το μοντέλο, η οποία στη συνέχεια μετασχηματίζεται σε ένα χώρο πιθανοτήτων για να μας δώσει την τελική έξοδο, δηλαδή την κλάση στην οποία ανήκει το εκάστοτε δείγμα. Στην περίπτωση της ταξινόμησης δύο κλάσεων είναι αρκετά εύκολο να καταλάβουμε από την τιμή αυτή, την κλάση στην οποία ταξινομήθηκε, αλλά για την ταξινόμηση πολλών κλάσεων δεν είναι τόσο απλό. Αυτό βέβαια, δεν δημιουργεί κάποιο πρόβλημα, καθώς γνωρίζουμε την κλάση την οποία προέβλεψε το μοντέλο για το κάθε δείγμα, όπως και την κλάση στην οποία όντως ανήκει, επομένως μπορούμε να ξέρουμε τη εκφράζει αυτή η τιμή και άρα να αντλήσουμε τη πληροφορία που χρειαζόμαστε από το διάγραμμα.

Κεφάλαιο 4

Αποτελέσματα

Στο κεφάλαιο αυτό θα παρουσιάσουμε και θα σχολιάσουμε αναλυτικά τα αποτελέσματα των πειραμάτων μας. Αρχικά, θα δούμε την απόδοση των μοντέλων ταξινόμησης DGA domain names (binary και multiclass classification), αλλά και κατά πόσο αλλάζει η απόδοση των μοντέλων με την εμφάνιση νέων οικογενειών ανά τα χρόνια. Στη συνέχεια, θα παρουσιάσουμε τα διαγράμματα που κατασκευάσαμε με τη βοήθεια της SHAP. Τα διαγράμματα αυτά μας παρέχουν πληροφορίες για το ποια χαρακτηριστικά άσκησαν τη μεγαλύτερη επιρροή στη διαδικασία κατηγοριοποίησης των ονομάτων σε έγκυρα domain names και DGA domain names. Με αυτό τον τρόπο, αποκτούμε εικόνα για το εάν και ποια διαφορετικά χαρακτηριστικά επηρεάζουν την απόφαση του binary και του multiclass μοντέλου για την ταξινόμηση των ίδιων δειγμάτων.

4.1 Αξιολόγηση μοντέλων

Για την αξιολόγηση των μοντέλων χρησιμοποιήσαμε τις μετρικές που αναφέραμε στην Ενότητα 3.4.2, δηλαδή Accuracy, Precision, Recall και F1 Score. Ο βασικότερος στόχος μας ήταν να συγκρίνουμε τα χαρακτηριστικά που επηρεάζουν τις προβλέψεις ενός binary και ενός multiclass μοντέλου. Για το σκοπό αυτό αρκεστήκαμε σε έναν απλό και γρήγορο αλγόριθμο με ικανοποιητική απόδοση τόσο στην ταξινόμηση δύο κλάσεων όσο και στην ταξινόμηση πολλών κλάσεων, όπως ο Random Forest Classifier.

4.1.1 Binary Classification

Στον Πίνακα 4.1 βλέπουμε τις τιμές των μετρικών απόδοσης για την ταξινόμηση δύο κλάσεων Random Forest (RF binary classification).

Μετρική	**Τιμή**
Accuracy	0.958
Precision	0.976
Recall	0.977
F1 score	0.977

Πίνακας 4.1: *Μετρικές απόδοσης για την ταξινόμηση δύο κλάσεων Random Forest*

Βλέπουμε, λοιπόν, πως το μοντέλο μας επιτυγχάνει accuracy σχεδόν 96% και precision,

recall και F1 score κοντά στο 97%. Πρόκειται για μία αρκετά καλή απόδοση, αλλά υπάρχουν και υλοποιήσεις για την ανίχνευση DGA-generated domain names με απόδοση μεγαλύτερη του 99%, όπως οι [36], [47], οι οποίες όμως είναι εξαιρετικά περίπλοκες και βασίζονται επίσης στα NXDomain Responses. Σε τέτοιου είδους προβλήματα είναι ιδιαιτέρως σημαντικό να ελαχιστοποιούνται οι ψευδείς προβλέψεις. Συγκεκριμένα, στο πρόβλημα ανίχνευσης DGA domain names τα false positive αντιστοιχούν στα έγκυρα ονόματα που έχουν ταξινομηθεί ως κακόβουλα, για να αποφύγουμε την αποκοπή πραγματικής κίνησης, το οποίο μπορεί να ζημιώσει το δίκτυο. Αντίστοιχα, τα false negatives αντιστοιχούν στα κακόβουλα ονόματα που ταξινομήθηκαν ως έγκυρα και επομένως δεν αποκόπτονται με αποτέλεσμα να διακινδυνεύεται η ασφάλεια του δικτύου. Όπως αναφέραμε όμως και προηγουμένως, δεν εστιάσαμε στη βελτιστοποίηση της απόδοσης του μοντέλου και άρα στην ελαχιστοποίηση των ψευδών προβλέψεων. Παρόλα αυτά, η απόδοση είναι αρκετά ικανοποιητική για να αντλήσουμε χρήσιμες πληροφορίες από την ερμηνεία του μοντέλου.

4.1.2 Multiclass Classification

Στον Πίνακα 4.2 βλέπουμε τις τιμές των μετρικών απόδοσης για την ταξινόμηση πολλών κλάσεων Random Forest (RF multiclass classification).

Μετρική	**Τιμή**
Accuracy	0.718
Precision	0.725
Recall	0.716
F1 score	0.700

Πίνακας 4.2: *Μετρικές απόδοσης για την ταξινόμηση πολλών κλάσεων Random Forest*

Η απόδοση του μοντέλου πολλών κλάσεων είναι αισθητά χαμηλότερη, όπως ήταν αναμενόμενο, καθώς καλείται να ταξινομήσει τα δεδομένα σε 55 κλάσεις. Δεδομένου του μεγάλου πλήθους των κλάσεων και της απλότητας του μοντέλου, η απόδοση είναι ικανοποιητική. Στην δημοσίευση των Drichel et al. [48], οπού υλοποιούν διάφορα μοντέλα για ταξινόμηση 92 κλάσεων επιτυγχάνουν αποδόσεις από 35% έως και 82%. Επομένως, η απόδοση σχεδόν ίση με 72% που επιτυγχάνει το μοντέλο μας είναι επαρκής για να βγάλουμε συμπεράσματα για την ταξινόμηση πολλαπλών οικογενειών DGA και να συγκρίνουμε με το μοντέλο ταξινόμησης δύο κλάσεων. Γενικότερα, όμως, το multiclass classification είναι κάπως δευτερεύον πρόβλημα, καθώς ο βασικός στόχος είναι η ανίχνευση των κακόβουλων ονομάτων. Αποτελεί κάτι σαν δεύτερο στάδιο στην ανάλυση μας, όπου παίρνουμε περισσότερες πληροφορίες για τους διάφορους τύπους malware.

4.1.3 Αξιολόγηση μοντέλων στο πέρασμα του χρόνου (2010-2019)

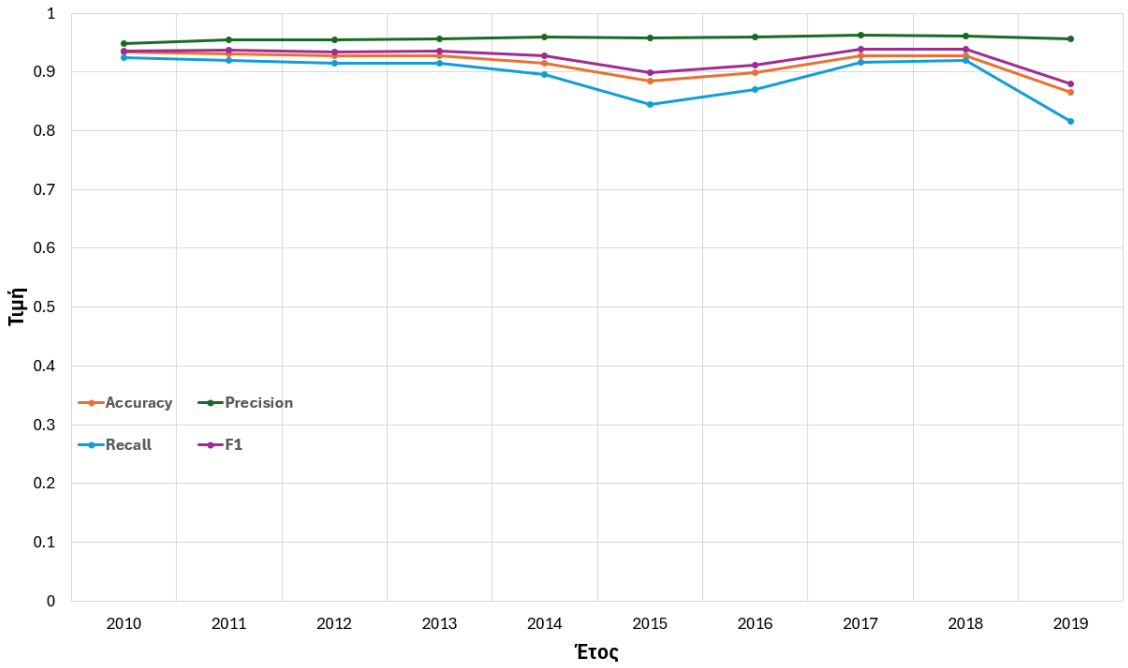
Αναφέραμε προηγουμένως ότι εκμεταλλευτήκαμε το γεγονός ότι το DGArchive προσφέρει πληροφορίες για τη διάρκεια ζωής των ονομάτων για να εκτελέσουμε κάποια πειράματα με σκοπό να δούμε πως συμπεριφέρονται τα μοντέλα μας κατά την εμφάνιση νέων οικογενειών,

δηλαδή αν μπορούν να ανταποκριθούν στην ανίχνευση ονομάτων στα οποία δεν έχουν εκπαιδευτεί.

Για το σκοπό αυτό, εκπαιδεύσαμε τα μοντέλα με τα παλαιότερα δεδομένα που περιλαμβάνονται στο αρχείο, δηλαδή ονόματα που χρησιμοποιήθηκαν κατά το έτος 2010. Για τη χρονιά αυτή, το αρχείο είχε δεδομένα από 10 διαφορετικές οικογένειες. Στη συνέχεια, αξιολογήσαμε το μοντέλο με δεδομένα της ίδιας χρονιάς που το εκπαιδεύσαμε (δηλαδή το 2010) και με όλες τις ακόλουθες χρονιές. Κάθε χρόνο είχαμε εμφάνιση νέων οικογενειών, με τις 10 οικογένειες του 2010 να υπάρχουν και όλες τις ακόλουθες χρονιές. Αναλυτικά οι οικογένειες που περιλαμβάνονται σε κάθε έτος φαίνονται στον πίνακα τους Παραρτήματος Α΄.

Binary Model

Στο Σχήμα 4.1 βλέπουμε πως επηρεάζονται οι 4 μετρικές αξιολογώντας το μοντέλο ταξινόμησης δύο κλάσεων binary classification model κάθε φορά με τα δεδομένα της επόμενης χρονιάς (2010-2019), ενώ έχει εκπαιδευτεί με δεδομένα από το 2010.



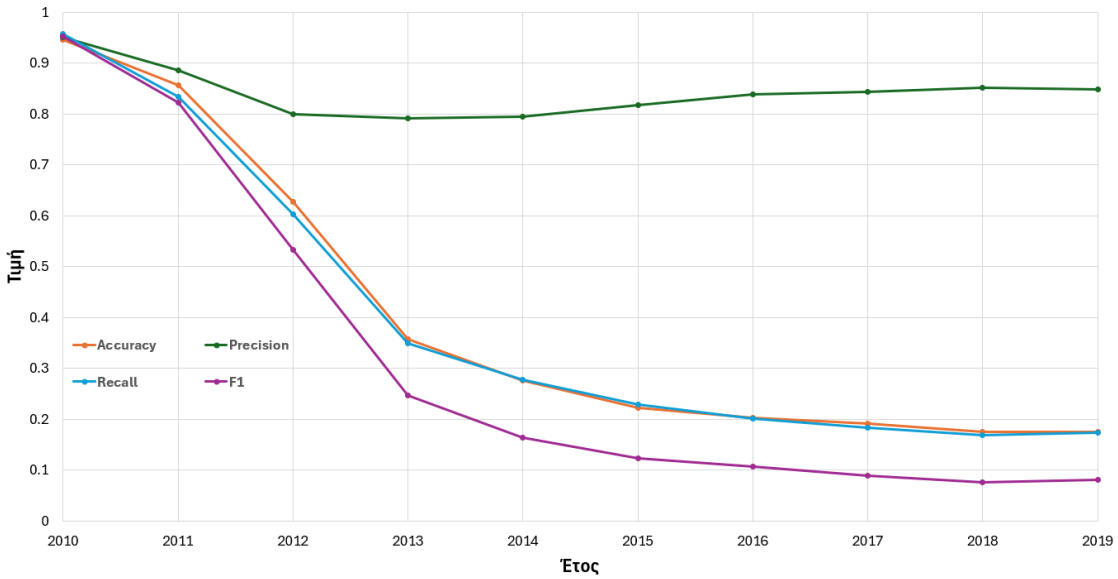
Σχήμα 4.1: *Απόδοση binary μοντέλου (2010-2019)*

Παρατηρούμε ότι όλες οι μετρικές με εξαίρεση το Precision ακολουθούν αντίστοιχες πορείες. Αυτό συμβαίνει γιατί το Precision όπως είδαμε και στη Σχέση 3.4 εξαρτάται από τα False Positives, δηλαδή τα δείγματα που ταξινομήθηκαν ως κακόβουλα ενώ στη πραγματικότητα δεν ήταν. Στο συγκεκριμένο πείραμα είναι αναμενόμενο τα δείγματα αυτά να είναι ελάχιστα καθώς τα έγκυρα ονόματα δεν έχουν κάποια διάρκεια ζωής, οπότε παρότι τα επιλέγουμε τυχαία, είναι πολύ πιθανό να επαναλαμβάνονται και άρα το μοντέλο να τα γνωρίζει και άρα να τα ταξινομεί σωστά. Για το λόγο αυτό, τα False Positives είναι πολύ κοντά στο μηδέν με αποτέλεσμα ο λόγος $\frac{\text{True Positives}}{\text{True Positives+False Positives}}$ να τείνει στο 1.

Όσον αφορά τις υπόλοιπες μετρικές υπάρχει πτώση συγκριτικά με τις αρχικές τιμές. Αυτό ήταν αναμενόμενο καθώς έχουμε εκπαιδεύσει το μοντέλο με δεδομένα από μόλις 10 οικογένειες και το αξιολογούμε με δεδομένα έως και 52 οικογένειες (δεν συνυπάρχουν και οι 55 οικογένειες σε κανένα έτος). Παρατηρούνται κάποιες πιο απότομες πτώσεις για τις χρονιές 2015 και 2019, το οποίο υποδεικνύει ότι τα ονόματα των οικογενειών DGA για εκείνες τις χρονιές διαφέρουν κατά πολύ. Συνολικά, όμως, η πτώση της απόδοσης δεν είναι τόσο δραματική καθώς συγκεκριμένα για το Accuracy, που μας δίνει μία συνολική εικόνα των δειγμάτων που ταξινομήθηκαν σωστά, οι τιμές κυμαίνονται μεταξύ του 0.87 και 0.93, το οποίο μας δείχνει ότι παρά την εμφάνιση μεγάλου πλήθους νέων και διαφορετικών οικογενειών DGA, το μοντέλο εξακολουθεί να έχει αρκετά καλή ικανότητα να αναγνωρίζει ότι ένα όνομα είναι κακόβουλο. Τέλος, παρατηρούμε πως το Recall έχει τις χαμηλότερες τιμές, κάτι που επίσης είναι κατανοητό, καθώς είναι η μετρική που εκφράζει πόσα από τα DGA domain names ταξινομήθηκαν σωστά, το οποίο είναι απόλυτα λογικό να είναι χαμηλότερο από τα υπόλοιπα.

Multiclass Model

Στο Σχήμα 4.2 βλέπουμε πως επηρεάζονται οι 4 μετρικές αξιολογώντας το μοντέλο ταξινόμησης πολλών κλάσεων multiclass classification model κάθε φορά με τα δεδομένα της επόμενης χρονιάς (2010-2019), ενώ έχει εκπαιδευτεί με δεδομένα από το 2010.



Σχήμα 4.2: *Απόδοση multiclass μοντέλου (2010-2019)*

Εδώ βλέπουμε ότι όλες οι μετρικές εκτός του Precision μειώνονται δραματικά. Ο λόγος αυτού είναι ο ίδιος που εξηγήσαμε και παραπάνω στη περίπτωση του μοντέλου ταξινόμησης δύο κλάσεων, με τη μόνη διαφορά ότι σε αυτή τη περίπτωση ο αριθμός των False Positive δεν είναι τόσο κοντά στο μηδέν και άρα ο λόγος δεν τείνει στο ένα. Παρόλα αυτά, οι τιμές τις μετρικής αυτής, διαφέρουν κατά πολύ από τις υπόλοιπες και ο λόγος είναι ο ίδιος με την παραπάνω περίπτωση.

Στην Ενότητα 4.1.2 είδαμε πως η απόδοση του είναι κοντά στο 72%, στο διάγραμμα όμως αυτό, βλέπουμε για τις χρονιές 2010 και 2011 η απόδοση αυτή είναι μεγαλύτερη. Αυτό συμβαίνει, γιατί εκπαιδεύουμε το μοντέλο με δεδομένα του 2010, τα οποία περιέχουν μόλις 10 κλάσεις. Είναι λογικό, λοιπόν, να έχει καλύτερη απόδοση καθώς καλείται να ταξινομήσει 10 και όχι 55 κλάσεις. Αντίστοιχα, στην αξιολόγηση με δεδομένα του 2011 έχει επίσης καλύτερη απόδοση καθώς το 2011 περιέχει 11 κλάσεις και προφανώς είναι χειρότερη από το 2010 καθώς το μοντέλο αδυνατεί να αναγνωρίσει τα δεδομένα που ανήκουν στη νέα αυτή κλάση. Αυτό εξηγεί γενικά τη δραματική πτώση που βλέπουμε στην απόδοση στο πέρας των χρόνων. Κάθε χρονιά, εμφανίζονται ολοένα και περισσότερες νέες οικογένειες DGA και άρα περισσότερες κλάσεις για τις οποίες το μοντέλο δεν έχει εκπαιδευτεί και άρα αδυνατεί να ταξινομήσει σωστά. Συνολικά, παρατηρούμε πως οι ιδιαιτερότητες των διάφορων οικογενειών επηρεάζουν σε μικρό βαθμό τη δυαδική ταξινόμηση, αλλά έχουν μεγάλο αντίκτυπο στην ταξινόμηση πολλών κλάσεων.

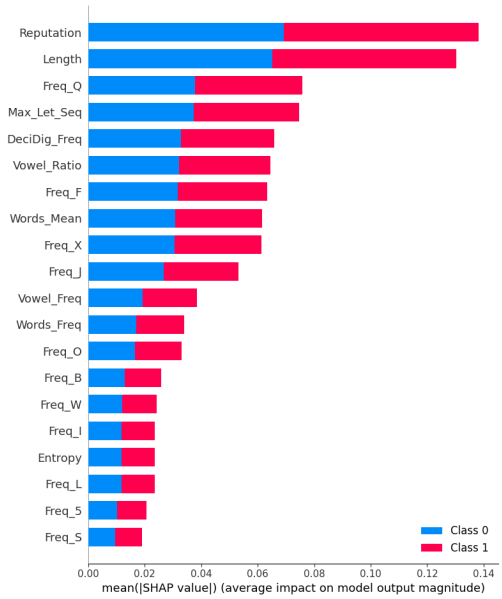
4.2 Καθολικές Επεξηγήσεις (Global Explanations)

Στην Ενότητα αυτή θα παρουσιάσουμε και θα αναλύσουμε τα διαγράμματα που κατασκευάσαμε με τη βοήθεια της SHAP με σκοπό να δούμε ποια χαρακτηριστικά και με ποιο τρόπο επηρεάζουν τις προβλέψεις των μοντέλων μας (binary και multiclass) και εάν ή κατά πόσο διαφέρουν τα χαρακτηριστικά αυτά ανάμεσα στα δύο μοντέλα. Υπενθυμίζουμε πως τα bar plots μας παρέχουν πληροφορία μόνο για το ποια χαρακτηριστικά επηρεάζουν περισσότερο τα μοντέλα και όχι με ποιον τρόπο (πληροφορίες για τον τρόπο που επηρεάζουν παίρνουμε από τα beeswarm plots). Παρόλα αυτά ήταν σημαντικό να τα συμπεριλάβουμε καθώς για τη περίπτωση του μοντέλου ταξινόμησης πολλών κλάσεων είναι το μόνο διάγραμμα που μας δίνει εικόνα για το σύνολο του μοντέλου (τα beeswarm plots στη ταξινόμηση πολλών κλάσεων δίνουν πληροφορία για κάθε μία από τις κλάσεις ξεχωριστά).

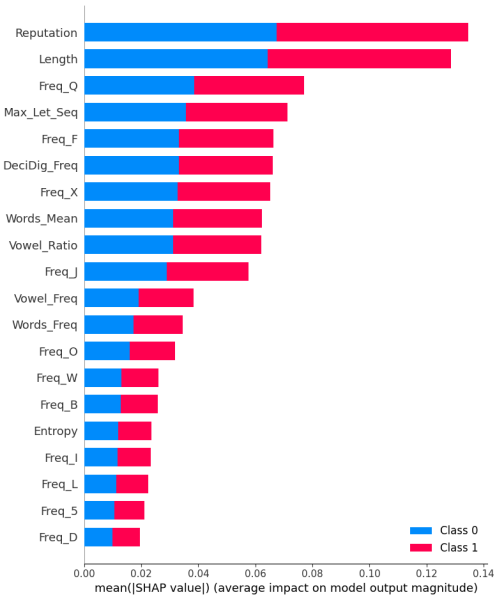
Αρχικά, επειδή τα διαγράμματα αυτά προκύπτουν από ένα μικρό υποσύνολο των δεδομένων αξιολόγησης έχουμε φτιάξει τα διαγράμματα για τέσσερα διαφορετικά τέτοια σύνολα δειγμάτων. Με αυτό τον τρόπο, έχουμε μία συνολικότερη εικόνα των χαρακτηριστικών που επηρεάζουν τα μοντέλα και μπορούμε να δούμε εάν υπάρχει συνοχή στα χαρακτηριστικά που κρίθηκαν σημαντικότερα μεταξύ των τεσσάρων διαφορετικών συνόλων.

4.2.1 Binary Model

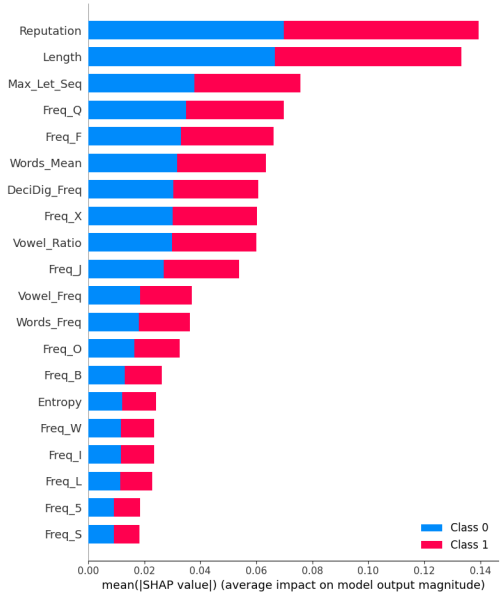
Στο Σχήμα 4.3 βλέπουμε τα διαγράμματα για 4 διαφορετικά σύνολα δειγμάτων. Το κάθε ένα από αυτά έχει δημιουργηθεί από 200 τυχαία δειγματικά σημεία του συνόλου αξιολόγησης (test set). Τα τέσσερα σύνολα δειγμάτων (Version 1 - V1, Version 2 - V2, Version 3 - V3, Version 4 - V4) έχουν επιλεχθεί έτσι ώστε να μην περιέχουν κανένα κοινό όνομα αναμεταξύ τους.



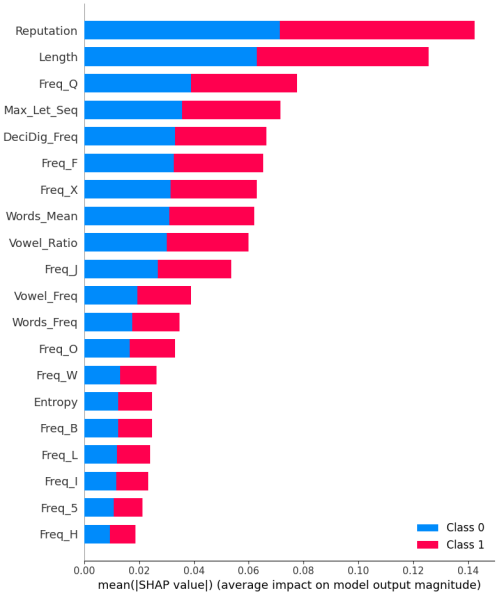
(α') V1



(β') V2



(γ') V3



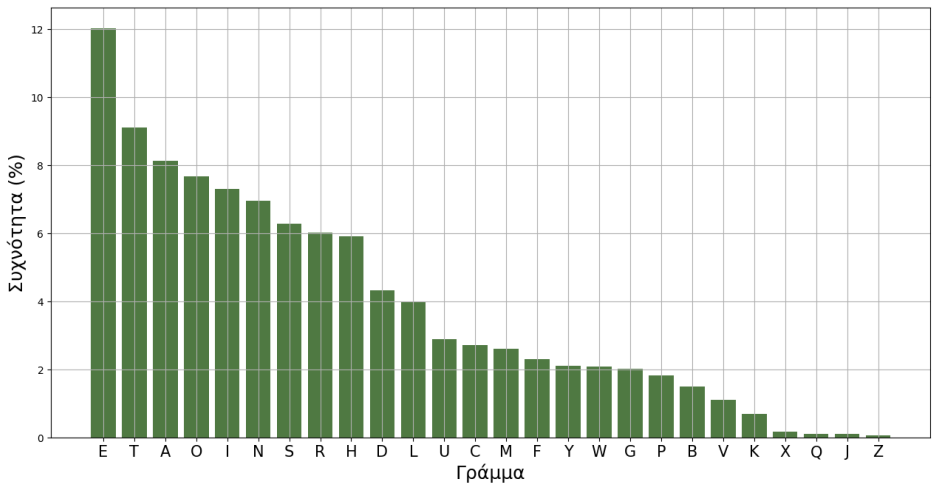
(δ') V4

Σχήμα 4.3: *Bar plots των τεσσάρων διαφορετικών συνόλων δειγμάτων (V1-V4) του μοντέλου ταξινόμησης δύο κλάσεων*

Με μία πρώτη ματιά, βλέπουμε πως τα δύο πιο σημαντικά χαρακτηριστικά (με μεγάλο προβάδισμα σε σχέση με τα υπόλοιπα) είναι τα Reputation και Length και στα τέσσερα διαγράμματα. Το αποτέλεσμα αυτό βγάζει απόλυτο νόημα καθώς το χαρακτηριστικό Reputation, το οποίο βρίσκεται στην κορυφή και των τεσσάρων κατατάξεων, έχει δημιουργηθεί για να εξυπηρετεί ως ένας δείκτης εγκυρότητας των ονομάτων και όπως βλέπουμε εκπληρώνει το σκοπό του με επιτυχία. Στην κατάταξη ακολουθεί το χαρακτηριστικό Length, το οποίο είναι επίσης λογικό να βρίσκεται τόσο ψηλά στην κατάταξη. Πολλά αλγοριθμικά παραγόμενα

ονόματα έχουν συνήθως μεγάλος μήκος και ειδικά τα hash-based DGA ονόματα, επομένως είναι αναμενόμενο να παίζει σημαντικό ρόλο το μήκος των ονομάτων στην ταξινόμησης τους σε κακόβουλα ή όχι ονόματα.

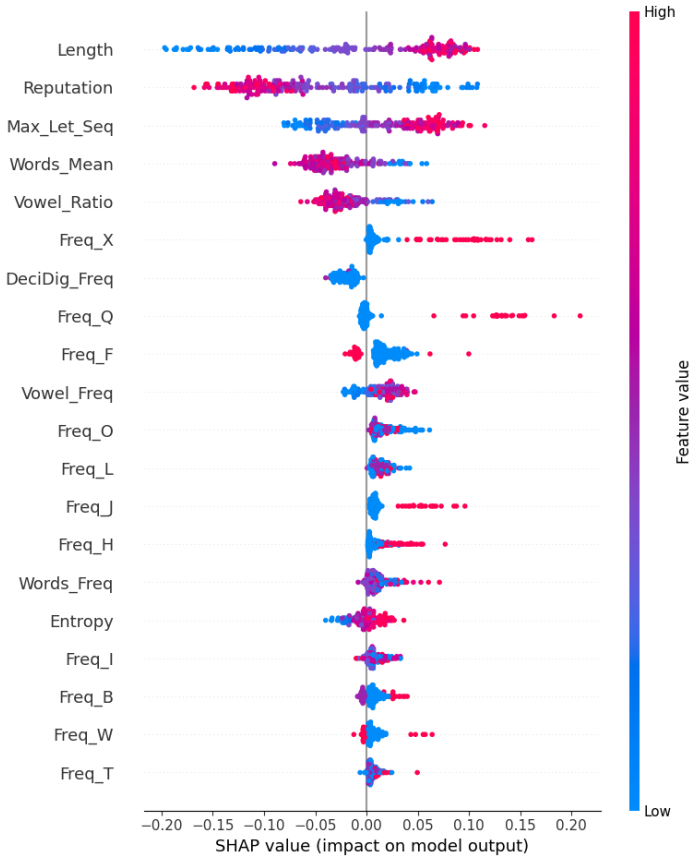
Τα υπόλοιπα χαρακτηριστικά δεν ακολουθούν ακριβώς την ίδια σειρά κατάταξης σε όλες τις περιπτώσεις. Οι διαφορές όμως αυτές δεν είναι σημαντικές καθώς η διαφορά στη θέση της κατάταξης κρίνεται στις μικροδιαφορές των SHAP values (οριζόντιος άξονας). Το σημαντικό είναι ότι για όλα τα δείγματα, τα χαρακτηριστικά που επηρεάζουν περισσότερο την πρόβλεψη του μοντέλου είναι κοινά με μικροδιαφορές στην κατάταξη, το οποίο μας δείχνει ότι το μοντέλο έχει συνοχή και τα ίδια χαρακτηριστικά επηρεάζουν την απόφαση για τα διάφορα δείγματα. Συγκεκριμένα, μπορούμε να δούμε πως παίζουν σημαντικό ρόλο τα Max_Let_Seq, DecDig_Freq, Vowel_Ratio, Words_Mean καθώς και οι συχνότητες διάφορων γραμμάτων. Η μέγιστη ακολουθία γραμμάτων (Max_Let_Seq) είναι λογικό να επηρεάζει σημαντικά το μοντέλο ταξινόμησης καθώς πολλά ονόματα έχουν μεγάλες ακολουθίες γραμμάτων ακατάληπτων και μη. Οι πραγματικές λέξεις και τα ονόματα ιστότοπων (πχ google, twitter) που δεν είναι υπαρκτές λέξεις αλλά χρησιμοποιούνται σε έγκυρα domain names δεν έχουν συνήθως μεγάλο μήκος σε αντίθεση με τα αλγοριθμικά παραγόμενα ονόματα. Αντίστοιχα, το μεγάλο πλήθος αριθμών σε ένα όνομα (DecDig_Freq) είναι ένδειξη κακόβουλου ονόματος, αφού δεν είναι σύνηθες να συναντάμε ονόματα τομέα που να περιέχουν πολλούς αριθμούς. Οι μικρές τιμές του λόγου του πλήθους των φωνηέντων σε σχέση με το συνολικό μήκος του ονόματος (Vowel_Ratio) υποδηλώνει την ύπαρξη πολλών σύμφωνων ή/και αριθμών το οποίο συναντάται πολύ συχνά σε DGA domain names που συνήθως πρόκειται για ακατάληπτες ακολουθίες αλφαριθμητικών χαρακτήρων. Το μικρό πλήθος φωνηέντων στις τυχαίες ακολουθίες αυτές έχει να κάνει με το ότι η πιθανότητα να επιλεγεί ένα φωνήεν στο λατινικό αλφάβητο είναι μόλις 5 στα 26. Το μέσο μήκος λέξεων (Words_Mean) δεν είναι ξεκάθαρο με ποιο τρόπο θα μπορούσε να επηρεάζει το μοντέλο λόγω της ύπαρξης wordlist-based DGA οικογενειών.



Σχήμα 4.4: *Συχνότητα εμφάνισης γραμμάτων σε 40.000 αγγλικές λέξεις*

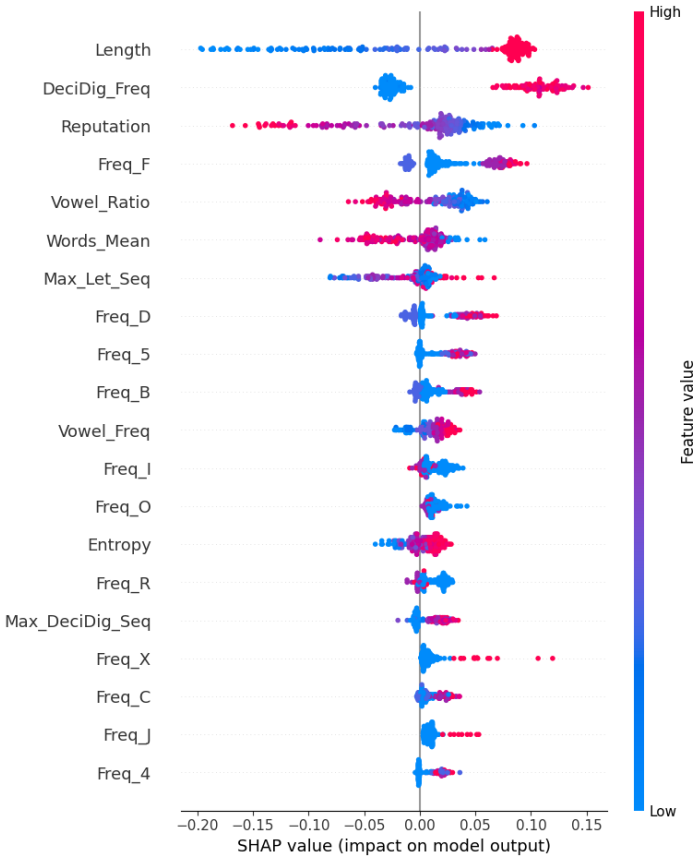
Για να μπορέσουμε να σχολιάσουμε την επιρροή των συχνοτήτων εμφάνισης διάφορων χαρακτήρων, παραθέτουμε το Σχήμα 4.4 με τη συχνότητα εμφάνισης των γραμμάτων της λατινικής αλφαβήτου για ένα δείγμα 40.000 αγγλικών λέξεων [49]. Στα διαγράμματα του Σχήματος 4.3 βλέπουμε πως επηρεάζουν οι συχνότητες των γραμμάτων X, J, Q, W, B και F, τα οποία σε ένα σύνολο σαράντα χιλιάδων λέξεων έχουν συχνότητα εμφάνισης μικρότερη του 2%. Για το λόγο αυτό, μεγάλο πλήθος εμφάνισης των γραμμάτων αυτών σε ένα όνομα, αποτελεί ένδειξη ότι το όνομα αυτό είναι πολύ πιθανό να είναι αλγοριθμικά παραγόμενο. Παρατηρούμε, όμως, ότι επιρροή ασκούν επίσης και κάποιες συχνότητες πιο συνηθισμένων γραμμάτων, όπως το O και το I. Για τις επιρροές αυτών των χαρακτηριστικών είναι δυσκολότερο να κατανοήσουμε το λόγο που μπορεί να επηρεάζουν την έξοδο του μοντέλου χωρίς περαιτέρω πληροφορία. Τέλος, βλέπουμε πως η Εντροπία περιλαμβάνεται στα 20 σημαντικότερα χαρακτηριστικά (από συνολικά 50 χαρακτηριστικά), αλλά κατέχει πολύ χαμηλότερη θέση στην κατάταξη από ότι περιμέναμε, καθώς έχει χρησιμοποιηθεί σε πολλές υλοποιήσεις μοντέλων ανίχνευσης ονομάτων DGA.

Όλα τα παραπάνω, αποτελούν απλώς εικασίες για το πως τα 20 σημαντικότερα χαρακτηριστικά επηρεάζουν στην πραγματικότητα τις προβλέψεις του μοντέλου. Για να επιβεβαιώσουμε ή και να διαψεύσουμε, λοιπόν, τις παραπάνω εικασίες, θα παρουσιάσουμε στη συνέχεια τα beeswarm plots για κάποιες από τις οικογένειες. Επιλέξαμε να εστιάσουμε στις οικογένειες που έχουν αναγνωριστεί ότι ανήκουν σε μία από τις τέσσερεις κατηγορίες των DGA (2.2.3), hash-based, arithmetic-based, wordlist-based, permutation-based, για να είναι ευκολότερο να κρίνουμε αν ο τρόπος που επηρεάζουν τα χαρακτηριστικά στην ταξινόμηση τους είναι λογικός ή όχι. Από τις οικογένειες που πήραμε από το DGArchive, μόνο μία είναι γνωστή ως permutation-based (VolatileCedar) η οποία όμως περιείχε μόλις 500 ονόματα, οπότε δεν έχει συμπεριληφθεί στα πειράματα μας. Επιπλέον, να σημειώσουμε πως τα ακόλουθα διαγράμματα προκύπτουν από διαφορετικά σύνολα δειγμάτων από τα προηγούμενα, καθώς τα προηγούμενα είχαν τυχαία δείγματα από όλες τις οικογένειες, ενώ στα ακόλουθα έχουμε μόνο δείγματα από έγκυρα ονόματα και την εκάστοτε οικογένεια. Επομένως, δεν θα ταυτίζονται οι κατατάξεις των χαρακτηριστικών.



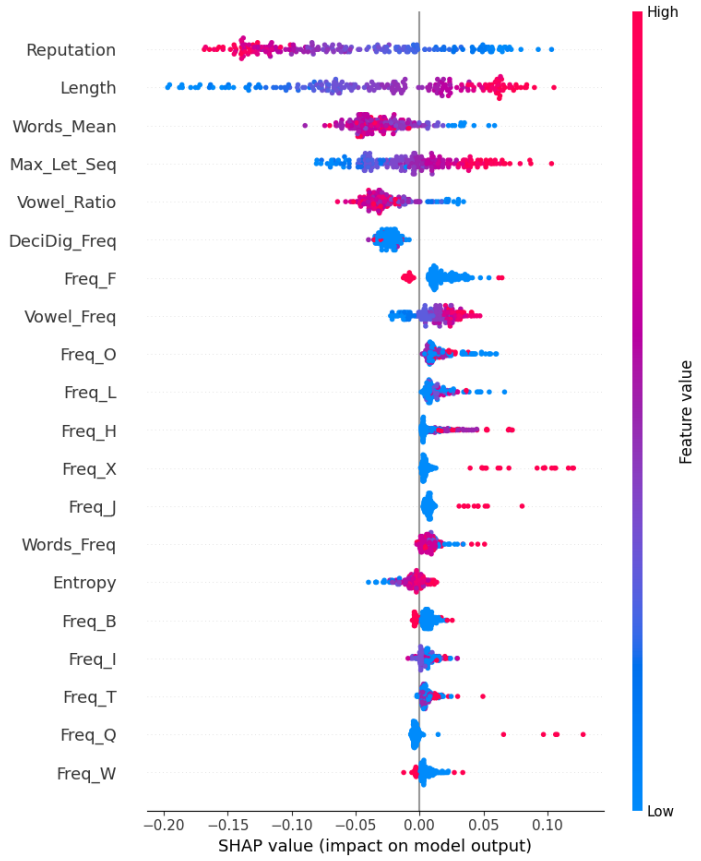
Σχήμα 4.5: *Beeswarm Plot of Binary Model: Banjori (Arithmetic-based DGA)*

Στο Σχήμα 4.5 βλέπουμε το Beeswarm Plot της οικογένειας Banjori η οποία είναι arithmetic-based. Υπενθυμίζουμε συνοπτικά πως σε αυτή τη κατηγορία ανήκουν οι αλγόριθμοι που για την κατασκευή ονομάτων αντιστοιχίζουν τυχαίες ακολουθίες αριθμών με χαρακτήρες ASCII. Αρχικά, παρατηρούμε πως τα μεγάλα μήκη ονομάτων (κόκκινες κουκίδες για Length) οδηγούν το μοντέλο στην επιλογή κακόβουλου ονόματος όπως περιμέναμε ενώ, οι υψηλές τιμές του Reputation οδηγούν τη πρόβλεψη προς τα έγκυρα ονόματα. Επίσης, όπως περιμέναμε οι μεγάλες ακολουθίες γραμμάτων τείνουν προς τα κακόβουλα ονόματα. Ειδικά για τη συγκεκριμένη οικογένεια είναι λογικό να κατέχει τόσο ψηλή θέση στη κατάταξη το χαρακτηριστικό αυτό, αφού πρόκειται για ακατάληπτες ακολουθίες χαρακτήρων. Στην κατάταξη ακολουθεί το Words_Mean, όπου οι μεγάλες τιμές τείνουν να συμβάλουν στην επιλογή έγκυρων ονομάτων. Αυτό βγάζει απόλυτο νόημα καθώς με τον τρόπο που κατασκευάζει ονόματα η οικογένεια Banjori, είναι σχεδόν αδύνατο να υπάρξουν λέξεις, επομένως τα ονόματα αυτά θα έχουν χαμηλό ή και μηδενικό μέσο μήκος λέξης. Επίσης, μπορούμε να δούμε πως οι μεγάλες συχνότητες εμφάνισης γραμμάτων τα οποία δεν εμφανίζονται συχνά σε υπαρκτές λέξεις (όπως X, Q) οδηγούν στην επιλογή κακόβουλου ονόματος.



Σχήμα 4.6: *Beeswarm Plot of Binary Model: Dyre (Hash-based DGA)*

Στο Σχήμα 4.6 βλέπουμε το Beeswarm Plot της οικογένειας Dyre η οποία είναι hash-based. Υπενθυμίζουμε συνοπτικά πως σε αυτή τη κατηγορία ανήκουν οι αλγόριθμοι που τα ονόματα προκύπτουν από την δεκαεξαδική αναπαράσταση κατακερματισμένων αλφαριθμητικών συμβολοσειρών. Βλέπουμε και πάλι πως τα μεγάλα μήκη ονομάτων και οι χαμηλές τιμές Reputation οδηγούν το μοντέλο στην επιλογή κακόβουλου ονόματος. Συγκεκριμένα, η οικογένεια Dyre παράγει ονόματα μήκους 34 χαρακτήρων, το οποίο και είναι σημαντικά μεγαλύτερο από ένα μέσο έγκυρο όνομα, αλλά είναι και προκαθορισμένο, για αυτό το Length έχει τόσο σημαντικό ρόλο στην αναγνώριση της οικογένειας αυτής. Τώρα όμως βλέπουμε ότι η συχνότητα εμφάνισης δεκαδικών αριθμών παίζει πιο σημαντικό και ξεκάθαρο ρόλο, με τις υψηλές συχνότητες να τείνουν έντονα προς τη κακόβουλη κλάση. Πράγμα που βγάζει νόημα καθώς οι οικογένειες που ανήκουν στη συγκεκριμένη κατηγορία περιέχουν πολύ περισσότερους αριθμούς από άλλες οικογένειες DGA πόσο μάλλον από έγκυρα ονόματα. Επιπλέον, βλέπουμε και πάλι πως οι μεγάλες τιμές των χαρακτηριστικών Words_Mean και Vowel_Ratio οδηγούν προς την καλόβουλη κλάση, καθώς είναι ένδειξη έγκυρου ονόματος σε σχέση με τα ονόματα που αποτελούν προϊόντα κατακερματισμού. Η συχνότητα του F κατέχει αρκετά υψηλή θέση στη κατάταξη ενδεχομένως επειδή αποτελεί χαρακτήρα του δεκαεξαδικού συστήματος με γενικά χαμηλή συχνότητα εμφάνισης σε πραγματικές λέξεις (Σχήμα 4.4).



Σχήμα 4.7: *Beeswarm Plot of Binary Model: SuppoBox (Wordlist-based DGA)*

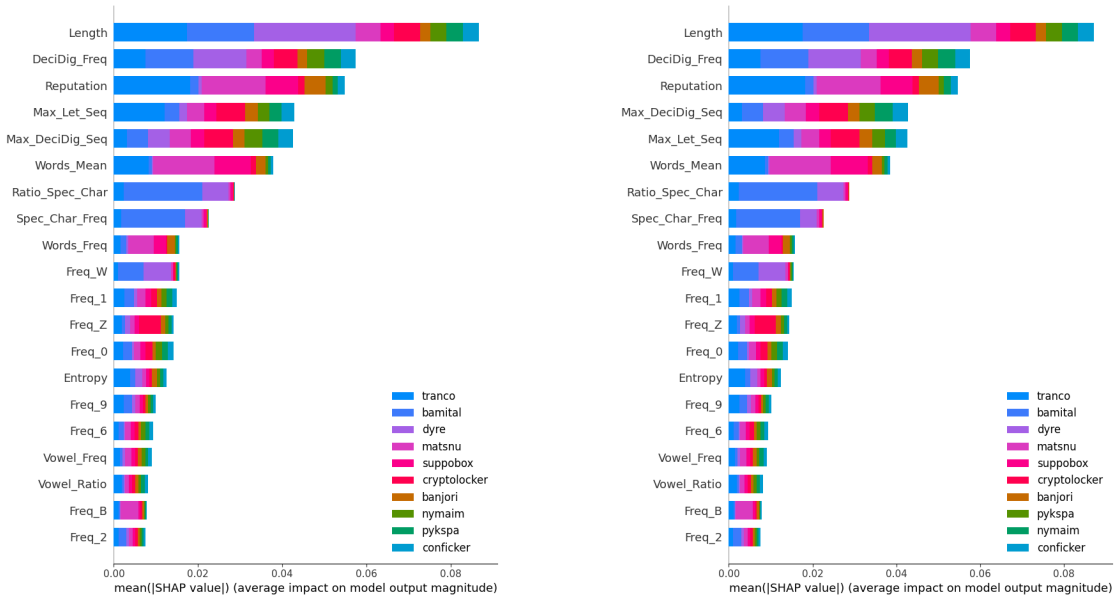
Στο Σχήμα 4.7 βλέπουμε το Beeswarm Plot της οικογένειας SuppoBox η οποία είναι wordlist-based. Υπενθυμίζουμε συνοπτικά πως σε αυτή τη κατηγορία ανήκουν οι αλγόριθμοι που τα ονόματα προκύπτουν από συνένωση τυχαίων λέξεων. Σε αυτή τη περίπτωση βλέπουμε πως το Reputation έχει περάσει στη πρώτη θέση της κατάταξης, το οποίο μάλλον υποδεικνύει λάθη στην ταξινόμηση. Αυτό είναι λογικό να συμβαίνει καθώς τα ονόματα αυτής της κατηγορίας είναι σχεδιασμένα με τρόπο που τα κάνει να μοιάζουν λιγότερο τυχαία. Αφού, λοιπόν, δεν πρόκειται για τυχαίες αλφαριθμητικές ακολουθίες το μοντέλο πρέπει να βασιστεί περισσότερο στις τιμές του Reputation για την ταξινόμηση τους. Βλέπουμε και πάλι τις μεγάλες τιμές του μήκους να έχουν αντίστοιχη επιρροή με τις παραπάνω περιπτώσεις όπως και για το Max_Let_Seq. Τα χαρακτηριστικά Words_Mean και Vowel_Ratio στην περίπτωση ταξινόμησης της οικογένειας αυτής παραπλανούν το μοντέλο. Αυτό συμβαίνει γιατί η πλειοψηφία των οικογενειών DGA αποτελείται από ακατάληπτες ακολουθίες χαρακτήρων, έτσι το μοντέλο εκπαιδεύεται να ταξινομεί ονόματα που τα χαρακτηριστικά τους υποδεικνύουν την ύπαρξη υπαρκτών λέξεων ως έγκυρα. Άρα οι υψηλές τιμές των Words_Mean και Vowel_Ratio που οδηγούν το μοντέλο να επιλέγει έγκυρα ονόματα είναι ο βασικότερος λόγος που τα ονόματα αυτής της οικογένειας ταξινομούνται λανθασμένα στη πλειοψηφία τους. Κάτι που επίσης έρχεται σε αντίθεση με τα προηγούμενα σε αυτή τη περίπτωση είναι η επιρροή του DecDig_Freq, όπου όλες οι τιμές του χαρακτηριστικού οδηγούν στην επιλογή έγκυρου ονόματος, σε αντίθεση με την οικογένεια Dyre. Αυτό συμβαίνει γιατί ένα έγκυρο όνομα είναι πιο πιθανό να περιέχει κάποιον αριθμό σε σχέση με ονόματα οικογενειών wordlist-based,

αφού ο τρόπος με τον οποίο κατασκευάζονται καθιστά αδύνατο να περιέχουν αριθμούς.

4.2.2 Multiclass Model

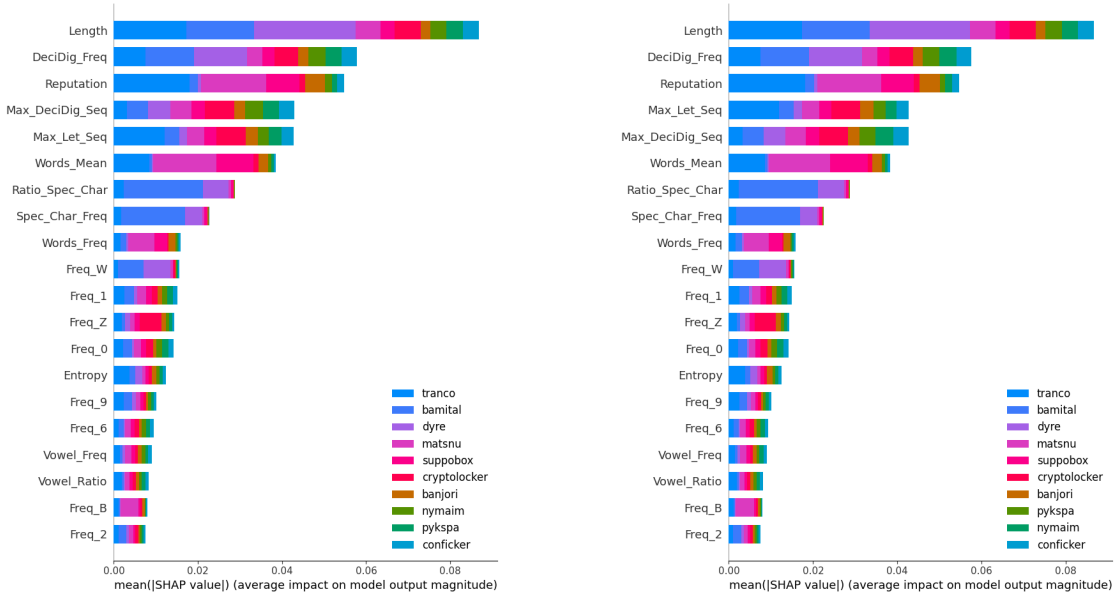
Στο Σχήμα 4.8 βλέπουμε τα διαγράμματα των 4 συνόλων δειγμάτων για το μοντέλο πολλών κλάσεων. Τα σύνολα αυτά είναι κοινά με αυτά των διαγραμμάτων της ταξινόμησης δύο κλάσεων για λόγους συνάφειας και περιέχουν τις ίδιες οικογένειες αλλά διαφορετικά instances αυτών για να δούμε εάν αλλάζουν οι επεξηγήσεις. Λόγω του μεγάλου πλήθους των κλάσεων επιλέξαμε δέκα από τις κλάσεις συμπεριλαμβανομένης της κλάσης των έγκυρων ονομάτων (tranco) για την οπτικοποίηση, αφού διαφορετικά θα ήταν αδύνατο να αντλήσουμε πληροφορία λόγω του συνωστισμού που θα υπήρχε. Σημειώνουμε, πως οι οικογένειες που δεν συμμετέχουν στην οπιτκοποίηση, έχουν συμπεριληφθεί κανονικά στην εκπαίδευση του μοντέλου. Οι οικογένειες αυτές έχουν επιλεχθεί γιατί είναι γνωστό πως ανήκουν σε μία από τις κατηγορίες DGA. Συγκεκριμένα, οι Bamital και Dyre είναι hash-based, οι Matsnu και SuppoBox είναι wordlist-based και τέλος οι Banjori, Cryptolocker, Conficker, Nyaim και Pykspa είναι arithmetic-based.

Στην κορυφή και των τεσσάρων κατατάξεων βλέπουμε το Length, το οποίο φαίνεται να ασκεί μεγαλύτερη επιρροή στις οικογένειες Bamital (σκούρο μπλε) και Dyre (μωβ), οι οποίες είναι hash-based, οπότε έχουν μεγάλα και σταθερά μήκη ονομάτων (Bamital: 32 χαρακτήρες, Dyre: 34 χαρακτήρες). Ακολουθεί το χαρακτηριστικό DecDig_Freq που και πάλι φαίνεται να επηρεάζει κυρίως τις hash-based οικογένειες, λογικό αφού σε σχέση με τις άλλες κατηγορίες οικογενειών έχουν μεγαλύτερο πλήθος αριθμών στα ονόματα που κατασκευάζουν. Έπειτα έχουμε το Reputation, το οποίο κατά κύριο λόγο επηρεάζει το Tranco, αλλά σε αντίστοιχο βαθμό ασκεί επιρροή και στην οικογένεια Matsnu, η οποία είναι wordlist-based. Στην δεύτερη περίπτωση, η επιρροή του χαρακτηριστικού μπορεί να συμβάλει στη παραπλάνηση του μοντέλου, αλλά αυτό δεν μπορούμε να το ξέρουμε χωρίς παραπάνω πληροφορίες. Γενικά, οι κατατάξεις είναι πανομοιότυπες, με εξαίρεση τα χαρακτηριστικά της 4ης και της 5ης θέσης που και πάλι είναι κοινά, απλά αντιστρέφεται η θέση τους σε δύο από τα τέσσερα διαγράμματα. Οι διαφορές αυτές δεν υποδεικνύουν κάποιο πρόβλημα, γιατί ενδεχομένως οφείλονται σε μικροδιαφορές των SHAP values. Βλέπουμε, δηλαδή, ότι στην περίπτωση του μοντέλου ταξινόμησης πολλών κλάσεων, φαίνεται να υπάρχει μεγαλύτερη συνοχή σε σχέση με την ταξινόμηση δύο κλάσεων. Ας σχολιάσουμε, όμως, μερικά ακόμα χαρακτηριστικά που παρουσιάζουν ενδιαφέρον σε αυτά τα διαγράμματα πριν περάσουμε στα beeswarm plots.



(α΄) V1

(β΄) V2



(γ΄) V3

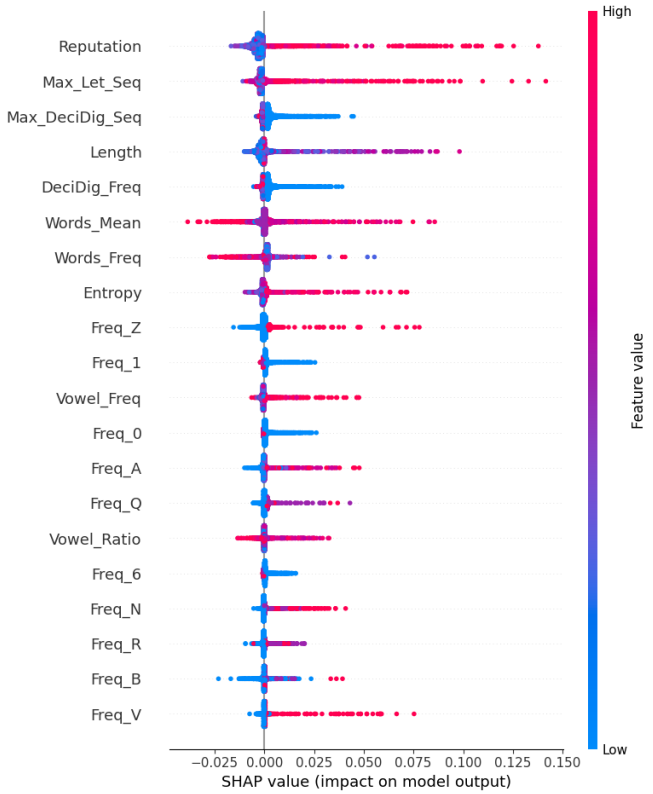
(δ΄) V4

Σχήμα 4.8: *Bar plots των τεσσάρων διαφορετικών συνόλων δειγμάτων (V1-V4) του μοντέλου ταξινόμησης πολλών κλάσεων*

Παρατηρούμε πως το χαρακτηριστικό Words_Mean επηρεάζει κυρίως τις κλάσεις Dyre και Matsnu και λίγο λιγότερο την κλάση Tranco. Είναι απόλυτα λογικό, καθώς είναι οι μόνες οικογένειες που περιέχουν υπαρκτές λέξεις, παρόλα αυτά είναι πολύ πιθανό το γεγονός αυτό να παραπλανεί το μοντέλο και να συμβάλει στη λανθασμένη ταξινόμηση των ονομάτων των οικογενειών αυτών. Επίσης, βλέπουμε πως στη κατάταξη της ταξινόμησης πολλών κλάσεων υπάρχουν τα χαρακτηριστικά που αφορούν τους χαρακτήρες «-» και «.» το οποίο δεν είχε εμφανιστεί καθόλου στη ταξινόμηση δύο κλάσεων και η επιρροή τους φαίνεται να κυριαρχεί

κυρίως στη κλάση Bamital. Η οικογένεια αυτή έχει μόνο ένα label μετά την αφαίρεση των TLD, οπότε πιθανώς η απουσία ειδικών χαρακτήρων να παίζει ρόλο στην ταξινόμηση. Τέλος, βλέπουμε πως στη δεύτερη δεκάδα της κατάταξης τώρα έχουμε κυρίως συχνότητες αριθμών και όχι γραμμάτων, οι οποίες δεν φαίνονται να έχουν σχετικά ισότιμη επιρροή μεταξύ των 10 κλάσεων, επομένως δεν μας δίνουν ιδιαίτερα χρήσιμη πληροφορία.

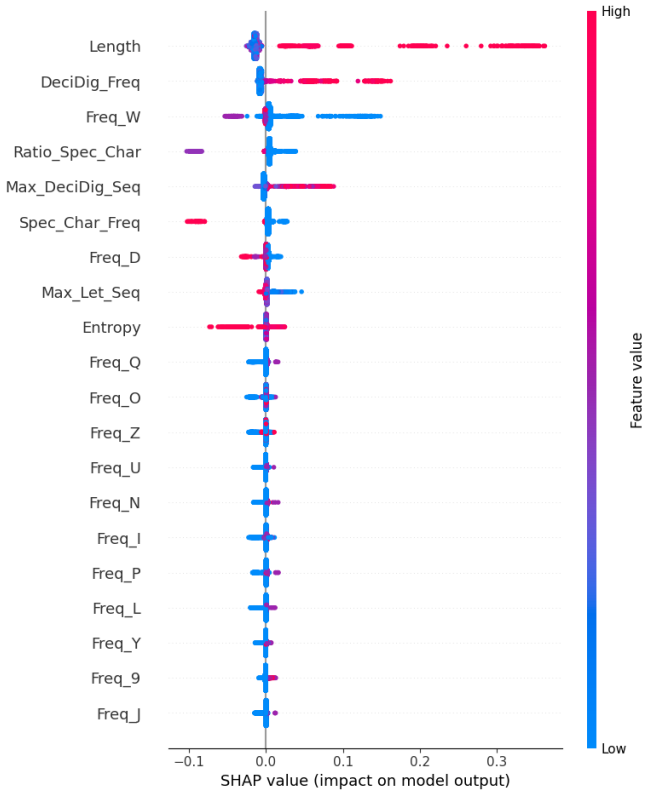
Στη συνέχεια, θα δούμε τα beeswarm plots των οικογενειών που είδαμε και για το μοντέλο ταξινόμησης δύο κλάσεων, δηλαδή Banjori, Dyre και Suppobox. Ας υπενθυμίσουμε σε αυτό το σημείο πως στα beeswarm plots της ταξινόμησης πολλών κλάσεων, όσο μεγαλύτερη η τιμή SHAP, τόσο πιο πολύ επηρεάζει το μοντέλο προς την επιλογή της κλάσης που μελετάμε ενώ το αντίθετο μας δείχνει ότι το μοντέλο τείνει να μη την επιλέξει. Δηλαδή, αντιμετωπίζουμε το διάγραμμα σαν ταξινόμηση δύο κλάσεων με τις κλάσεις αυτές να είναι «Είναι η κλάση X» και «Δεν είναι η κλάση X». Τέλος, να σημειώσουμε πως



Σχήμα 4.9: *Beeswarm Plot of Multiclass Model: Banjori (Arithmetic-based DGA)*

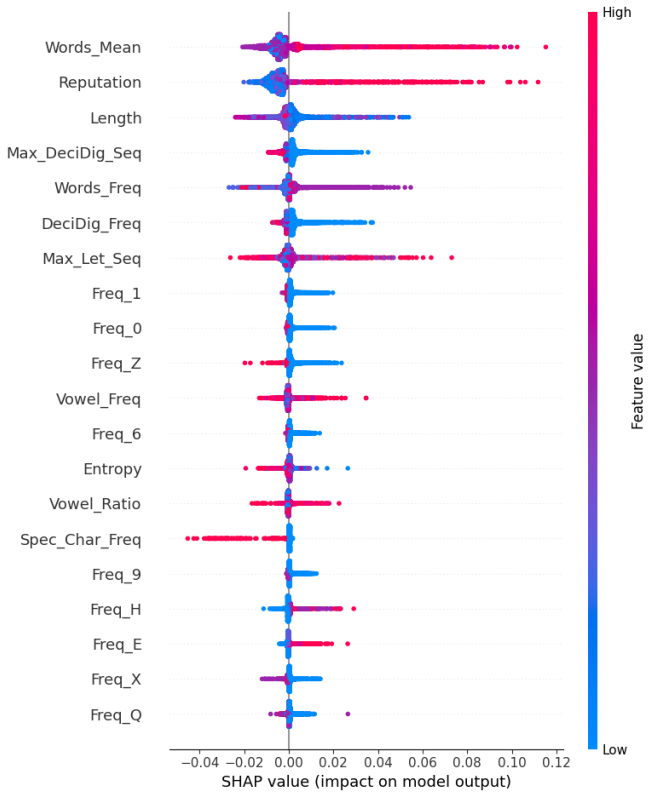
Στο Σχήμα 4.9 βλέπουμε το beeswarm plot της οικογένειας Banjori η οποία είναι arithmetic-based. Το πρώτο πράγμα που παρατηρούμε είναι ότι οι υψηλές τιμές του Reputation τείνουν στην επιλογή της κλάσης αυτής. Το Reputation υπολογίζεται συγκρίνοντας τα N-grams του ονόματος με ένα whitelist από N-grams έγκυρων ονομάτων. Λόγω της φύσης των ονομάτων αυτής της οικογένειας είναι πολύ πιθανό να προκύπτουν N-grams τα οποία ταυτίζονται με αυτά έγκυρων ονομάτων. Παρόλα αυτά, δεν φαίνεται να επηρεάζει ιδιαίτερα αρνητικά την ταξινόμηση των ονομάτων, πράγμα που σημαίνει πως αυτή η πληροφορία σε συνδυασμό με τις τιμές των υπόλοιπων χαρακτηριστικών βοηθάει στην πρόβλεψη του

μοντέλου. Κάτι ενδιαφέρον που παρατηρούμε είναι ότι το Entropy βρίσκεται στα 10 πιο σημαντικά χαρακτηριστικά, κάτι που δεν είχε παρατηρηθεί μέχρι τώρα σε κάποιο άλλο διάγραμμα. Η Εντροπία εκφράζει την τυχαιότητα μιας συμβολοσειράς και καθώς τα ονόματα αυτής της κλάσης προκύπτουν από αντιστοίχηση τυχαίων ακολουθιών αριθμών με χαρακτήρες ASCII, είναι λογικό οι υψηλές τιμές της Εντροπίας να συμβάλουν στην επιλογή της κλάσης.



Σχήμα 4.10: *Beeswarm Plot of Multiclass Model: Dyre (Hash-based DGA)*

Στο Σχήμα 4.10 βλέπουμε το beeswarm plot της οικογένειας Dyre η οποία είναι hash-based. Φαίνεται πως για την επιλογή αυτής της κλάσης αρκεί να έχουν μεγάλο μήκος και μεγάλο πλήθος δεκαδικών αριθμητικών ψηφίων, αφού τα υπόλοιπα χαρακτηριστικά έχουν τιμές SHAP κοντά στο μηδέν. Παρά το γεγονός ότι το μοντέλο βασίζεται σε ελάχιστη πληροφορία για την ταξινόμηση αυτής της οικογένειας, φαίνεται να ταξινομεί τα ονόματα της κλάσης με καλή ακρίβεια. Αυτό οφείλεται στο γεγονός ότι οι hash-based DGA οικογένειες παράγουν ονόματα με δεδομένο μήκος δεκαεξαδικών χαρακτήρων, το οποίο σημαίνει επίσης πως περιέχει μεγάλο πλήθος αριθμών σε σχέση με άλλες οικογένειες. Επίσης, μπορούμε να δούμε πως οι χαμηλές συχνότητες εμφάνισης του γράμματος W οδηγούν στην επιλογή της κλάσης. Αυτό ενδεχομένως οφείλεται στο ότι το γράμμα αυτό δεν περιλαμβάνεται στο δεκαεξαδικό σύστημα, άρα η απουσία του γράμματος αυξάνει το ενδεχόμενο το όνομα να ανήκει στη κλάση αυτή.



Σχήμα 4.11: *Beeswarm Plot of Multiclass Model: SuppoBox (Wordlist-based DGA)*

Στο Σχήμα 4.11 βλέπουμε το beeswarm plot της οικογένειας SuppoBox η οποία είναι wordlist-based. Εδώ σε αντίθεση με τη ταξινόμηση δύο κλάσεων, το μοντέλο ταξινομεί τα ονόματα των wordlist-based οικογενειών με μεγαλύτερη επιτυχία, αφού δεν εκλαμβάνει την ύπαρξη λέξεων και το υψηλό Reputation ως ενδείξεις εγκυρότητας του ονόματος όπως βλέπουμε στο διάγραμμα από τα χαρακτηριστικά Words_Means, Words_Freq και Reputation. Επίσης η απουσία αριθμητικών χαρακτήρων συμβάλει στην επιλογή της κλάσης όπως βλέπουμε από τα χαρακτηριστικά Max_DecDig_Seq, Dec_Dig_Freq και Freq_0,1,6,9. Τέλος, παρατηρούμε πως στο εν λόγω διάγραμμα οι τιμές του μήκους των ονομάτων για τις οποίες το μοντέλο τείνει να ταξινομεί σε αυτή τη κλάση κυμαίνονται σε μεσαίες προς χαμηλές. Αυτό μπορεί να οφείλεται στο γεγονός ότι οι υψηλότερες τιμές του μήκους έχουν συσχετιστεί με τις hash-based κλάσεις (32 ή 34 χαρακτήρες για Bamital και Banjori αντίστοιχα), ενώ το μήκος των ονομάτων της οικογένειας Suppobox κυμαίνεται μεταξύ 8 και 26 [14].

4.3 Τοπικές Επεξηγήσεις (Local Explanations)

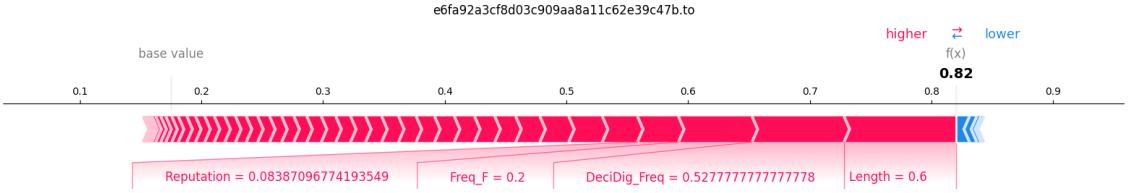
Στην Ενότητα αυτή, θα παρουσιάσουμε και θα αναλύσουμε διαγράμματα που κατασκευάσαμε με τη βοήθεια της SHAP με σκοπό να δούμε ποια χαρακτηριστικά και με ποιο τρόπο επηρεάζουν την πρόβλεψη συγκεκριμένων δειγματικών σημείων.

Θα δούμε τα διαγράμματα μερικών τυχαίων δειγματικών σημείων που έχουν ταξινομηθεί σωστά και μερικών που δεν έχουν ταξινομηθεί σωστά, για να δούμε αν τα χαρακτηριστικά που συνέβαλαν περισσότερο στην εκάστοτε πρόβλεψη, συμπίπτουν με αυτά που είδαμε στην

προηγούμενη ενότητα για την αντίστοιχη οικογένεια DGA.

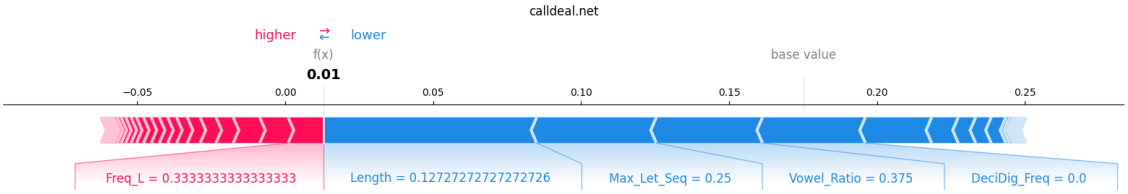
4.3.1 Binary Model

Στο Σχήμα 4.12 βλέπουμε το force plot του δειγματικού σημείου «e6fa92a3cf8d03c909aa8a11c62-e39c47b.to» το οποίο ανήκει στην hash-based οικογένεια Dyre και έχει ταξινομηθεί σωστά ως κακόβουλο όνομα. Υπενθυμίζουμε πως οι τιμές των χαρακτηριστικών έχουν κανονικοποιηθεί στο διάστημα [0,1]. Στο διάγραμμα μπορούμε να δούμε τις τιμές των χαρακτηριστικών που αντιστοιχούν στα μεγαλύτερα κόκκινα βέλη, δηλαδή τα χαρακτηριστικά που συνεισφέρανε περισσότερο στην ταξινόμηση του ονόματος ως DGA. Βλέπουμε, αρχικά, πως τα τέσσερα αυτά χαρακτηριστικά ταυτίζονται με τα τέσσερα πιο σημαντικά του Σχήματος 4.6 γενικά για την οικογένεια αυτή. Συγκεκριμένα, το μεγάλο μήκος του ονόματος, η ύπαρξη πολλών αριθμών και το χαμηλό Reputation είναι ένας συνδυασμός χαρακτηριστικών που παραπέμπει σε αλγοριθμικά παραγόμενο όνομα τόσο για το ανθρώπινο μάτι όσο και για το μοντέλο μας. Φαίνεται, όμως, ότι έπαιξε αρκετό ρόλο και η συχνότητα εμφάνισης του γράμματος F, η οποία παρότι δεν είναι υψηλή (μπορούμε να δούμε ότι το όνομα περιέχει δύο F), σε συνδυασμό με τα υπόλοιπα χαρακτηριστικά παραπέμπει το μοντέλο να ταξινομήσει το όνομα σωστά.



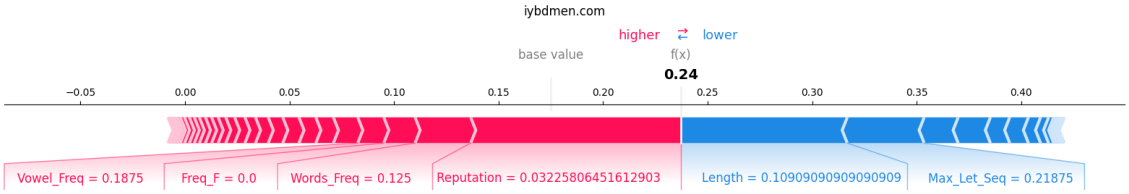
Σχήμα 4.12: *Σωστά ταξινομημένο όνομα της οικογένειας Dyre (hash-based DGA)*

Στο Σχήμα 4.13 βλέπουμε το force plot του δείγματος «calldeal.net» το οποίο ανήκει στην wordlist-based οικογένεια SuppoBox και έχει ταξινομηθεί λανθασμένα ως έγκυρο όνομα. Εδώ βλέπουμε πως το μικρό μήκος του ονόματος, η έλλειψη αριθμών και η αναλογία φωνηέντων παραπλανεί με ιδιαίτερη επιτυχία το μοντέλο το οποίο θεώρησε όνομα αυτό έγκυρο. Το μόνο χαρακτηριστικό που φαίνεται να είχε σχετικά έντονη επιρροή προς την αντίθετη κατεύθυνση είναι η συχνότητα του γράμματος L, το οποίο φυσικά δεν ήταν αρκετό για την ταξινόμηση του ονόματος ως κακόβουλο.



Σχήμα 4.13: *Λανθασμένα ταξινομημένο όνομα της οικογένειας SuppoBox (wordlist-based DGA)*

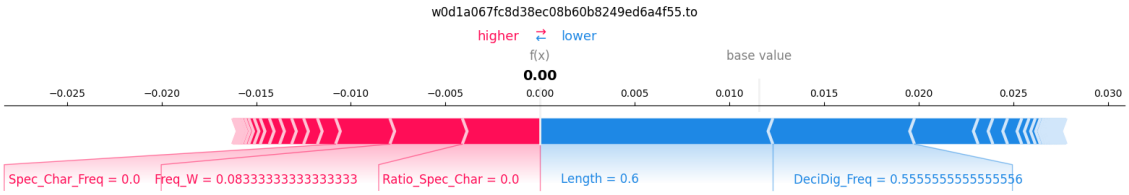
Στο Σχήμα 4.14 βλέπουμε το force plot του δείγματος «iybdmen.com» το οποίο ανήκει στην arithmetic-based οικογένεια Banjori και έχει ταξινομηθεί λανθασμένα ως έγκυρο όνομα. Βλέπουμε πως παρά τη χαμηλή τιμή του Reputation και τη χαμηλή συχνότητα εμφάνισης φωνηέντων, το μικρό μήκος του ονόματος κυριάρχησε λανθασμένα ως ένδειξη εγκυρότητας του ονόματος. Επίσης, παρατηρούμε πως έχει δημιουργηθεί τυχαία η λέξη «men», το γεγονός αυτό όμως δεν φαίνεται να ώθησε το μοντέλο προς τη λάθος κατεύθυνση παρότι το τελικό αποτέλεσμα ήταν λανθασμένο.



Σχήμα 4.14: *Λανθασμένα ταξινομημένο όνομα της οικογένειας Banjori (arithmetic-based DGA)*

4.3.2 Multiclass Model

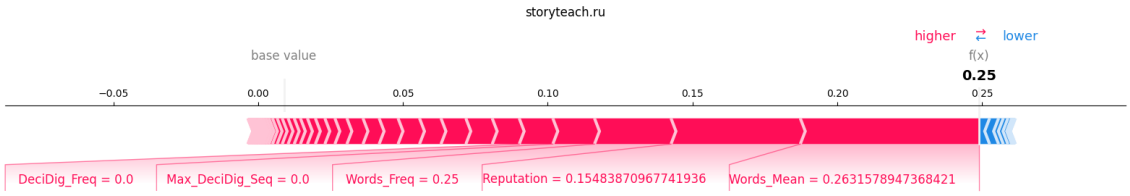
Στο Σχήμα 4.15 βλέπουμε το force plot του δείγματος «w0d1a067fc8d38ec08b60b8249ed6a4f55.to» το οποίο ανήκει στην hash-based οικογένεια Dyre και έχει ταξινομηθεί λανθασμένα στη κλάση WD. Ένα παράδειγμα ονόματος που ανήκει στην κλάση αυτή είναι «wd7bdb20e4d622f6569f3e8503138c859d.win». Το όνομα αυτό ήταν το μοναδικό όνομα αυτής της κλάσης από το σύνολο δειγμάτων που χρησιμοποιήσαμε για την κατασκευή των διαγραμμάτων που ταξινομήθηκε λανθασμένα και είναι λογικό, γιατί αν παρατηρήσουμε το όνομα ξεκινάει με το γράμμα W το οποίο δεν ανήκει στο δεκαεξαδικό σύστημα και άρα δεν είναι λογικό να ανήκει σε αυτή την οικογένεια. Η ύπαρξη αυτού του W φαίνεται και από το διάγραμμα ότι επηρέασε στην εσφαλμένη ταξινόμηση του ονόματος (τα ονόματα της οικογένειας WD ξεκινάνε με τα γράμματα wd).



Σχήμα 4.15: *Λανθασμένα ταξινομημένο όνομα της οικογένειας Dyre (hash-based DGA)*

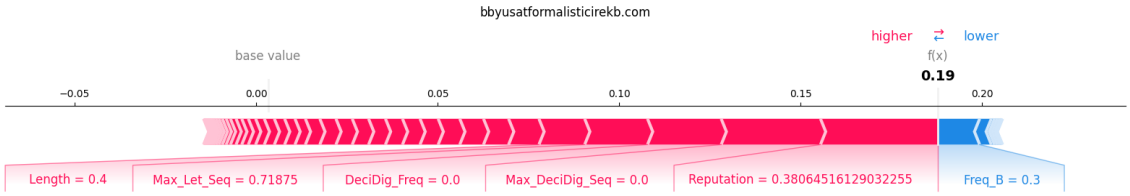
Στο Σχήμα 4.16 βλέπουμε το force plot του δείγματος «storyteach.ru» το οποίο ανήκει στην wordlist-based οικογένεια Suppobox και έχει ταξινομηθεί στην σωστή κλάση. Σε αντίθεση με την ταξινόμηση κλάσεων, τώρα η ύπαρξη λέξεων και η απουσία αριθμών δεν θεωρούνται από το μοντέλο ξεκάθαρες ενδείξεις εγκυρότητας. Αντιθέτως, σε συνδυασμό με

το χαμηλό Reputation του ονόματος, οδηγούν στη σωστή ταξινόμηση του ονόματος.



Σχήμα 4.16: *Σωστά ταξινομημένο όνομα της οικογένειας SuppoBox (wordlist-based DGA)*

Στο Σχήμα 4.17 βλέπουμε το force plot του δείγματος «bbyusatformalisticirekb.com» το οποίο ανήκει στην arithmetic-based οικογένεια Banjori και έχει ταξινομηθεί στην σωστή κλάση. Βλέπουμε και πάλι πως χαρακτηριστικά που στην ταξινόμηση δύο κλάσεων οδήγησαν σε λάθος αποτέλεσμα, τώρα συνεισφέρουν στη σωστή ταξινόμηση του δείγματος. Η απουσία αριθμών, οι μεγάλες ακολουθίες γραμμάτων και το χαμηλό Reputation είναι πληροφορίες είναι πληροφορίες που συνάδουν με τη φύση της οικογένειας και το μοντέλο τις εκμεταλλεύτηκε σωστά.



Σχήμα 4.17: *Σωστά ταξινομημένο όνομα της οικογένειας Banjori (arithmetic-based DGA)*

4.4 Σύγκριση Binary και Multiclass ταξινομητών

Στις προηγούμενες ενότητες, πήραμε μία καλή ιδέα για τα χαρακτηριστικά που συνεισφέρουν περισσότερο στις προβλέψεις των δύο ταξινομητών, συνολικά, για ορισμένες οικογένειες και για κάποια συγκεκριμένα δειγματικά σημεία. Αυτό που παρατηρήσαμε, λοιπόν, κατά την ανάλυση των παραπάνω διαγραμμάτων είναι πως παρότι ο ταξινομητής πολλών κλάσεων δεν έχει την καλύτερη απόδοση φαίνεται να εστιάζει πιο στοχευμένα σε κάποια χαρακτηριστικά για την ταξινόμηση ορισμένων κλάσεων.

Ας μιλήσουμε για παράδειγμα για τις wordlist-based οικογένειες, όπου το binary μοντέλο φαίνεται να υστερεί έντονα στην ανίχνευση τους. Αυτό βέβαια είναι αρκετά λογικό καθώς το μοντέλο αυτό καλείται γενικά να αναγνωρίσει ένα αλγοριθμικά παραγόμενο όνομα και καθώς οι οικογένειες αυτές αποτελούν ένα μικρό μέρος αυτών, το μοντέλο επιτυγχάνει μία πολύ ικανοποιητική απόδοση ακόμα και χωρίς την ανίχνευση των ονομάτων αυτών. Επίσης, τα ονόματα αυτής της κατηγορίας οικογενειών DGA κατασκευάζονται με τρόπο τέτοιο

ώστε να μιμούνται τα γλωσσολογικά χαρακτηριστικά των έγκυρων ονομάτων και να μοιάζουν λιγότερο τυχαία, το οποίο αυξάνει τη δυσκολία αναγνώρισης τους.

Από την άλλη πλευρά, ο ταξινομητής πολλών κλάσεων, οφείλει να εστιάσει στην κάθε μία κλάση κατά την εκπαίδευση του και άρα εστιάζει περισσότερο στις wordlist-based οικογένειες και τις αναγνωρίζει με μεγαλύτερη επιτυχία, όχι όμως άριστα. Αυτό φυσικά οδηγεί στην λανθασμένη ταξινόμηση έγκυρων ονομάτων ως κακόβουλα, το οποίο τελικά έχει αποτέλεσμα την μέτριας απόδοσης ανίχνευση κακόβουλων ονομάτων σε συνδυασμό με misclassified έγκυρα ονόματα, πρόβλημα που δεν αντιμετωπίζαμε στην ταξινόμηση δύο κλάσεων.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντική Μελέτη

5.1 Σύνοψη και Συμπεράσματα

Στην παρούσα διπλωματική εργασία, υλοποιήσαμε δύο ταξινομητές (binary και multiclass) για την κατηγοριοποίηση καλόβουλων και κακόβουλων ονομάτων παραγόμενα από DGAs με σκοπό να συγκρίνουμε τόσο την απόδοση όσο και τις ερμηνείες τους. Για την αξιολόγηση των δύο μοντέλων χρησιμοποιήσαμε τις μετρικές Accuracy, Precision, Recall και F1 Score, ενώ για την αποτίμηση της επίδρασης των χαρακτηριστικών που επηρεάζουν τις αποφάσεις των ταξινομητών, χρησιμοποιήσαμε μία από τις δημοφιλέστερες μεθόδους eXplainable Artificial Intelligece - XAI, την SHAP (SHapley Additive exPlanations), η οποία προσφέρει πληθώρα οπτικοποιήσεων τόσο για τοπικές όσο και καθολικές ερμηνείες των μοντέλα μας.

Ο ταξινομητής δύο κλάσεων που που υλοποιήσαμε με χρήση του Random Forest Classifier επιτυγχάνει ακρίβεια σχεδόν 96% με ελάχιστα False Positives (Precision 98%), αντίστοιχη απόδοση με άλλες υλοποιήσεις [50]. Όσον αφορά το ταξινομητή πολλών κλάσεων, επιτυγχάνουμε μία ακρίβεια της τάξης του 70% το οποίο είναι επίσης μία απόδοση συγκρίσιμη με άλλες δουλειές [48]. Εκμεταλλευτήκαμε, επίσης, το γεγονός ότι το DGArcihve παρέχει πληροφορίες για τη διάρκεια ζωής του κάθε ονόματος για να εκτιμήσουμε τις μεταβολές στην απόδοση των ταξινομητών στο πέρασμα του χρόνου με την εμφάνιση νέων οικογενειών DGA και κακόβουλων ονομάτων. Συγκεκριμένα, τα ονόματα που είχαμε στη διάθεση μας χρονολογούνται μεταξύ του 2010, με δεδομένα από 10 διαφορετικές οικογένειες και του 2019 με δεδομένα από 50 οικογένειες. Για το σκοπό του πειράματος μας, εκπαιδεύσαμε τους δύο ταξινομητές με τα ονόματα του 2010 και κατόπιν τους αξιολογήσαμε με τις ακόλουθες χρονιές (2011-2019). Προέκυψε, λοιπόν, πως η ακρίβεια (accuracy) του binary ταξινομητή πέφτει από το 93,4% στο 86,5%. Διατηρεί, επομένως, μία ικανοποιητική απόδοση παρά την εμφάνιση 40 νέων οικογενειών και πολλών νέων ονομάτων. Στην περίπτωση, του multiclass ταξινομητή η πτώση είναι πολύ πιο αισθητή, όπως και περιμέναμε, καθώς καλείται να ταξινομήσει κλάσεις τις οποίες δεν έχει συναντήσει κατά τη διαδικασία εκπαίδευσης τους (accuracy από 94,6% σε 17,6%).

Όσον αφορά την ανάλυση των οπτικοποιήσεων που κατασκευάσαμε μέσω της SHAP (model-agnostic XAI αλγόριθμος) για την εκτίμηση των επιδραστικότερων χαρακτηριστικών

των δύο μοντέλων παρατηρήσαμε τόσο ομοιότητες όσο και διαφορές ανάμεσα στα δύο μοντέλα, ανάλογα με την οικογένεια που μελετούσαμε. Συγκεκριμένα, για την hash-based οικογένεια Dyre είδαμε πως υπάρχει συμφωνία μεταξύ των δύο ταξινομητών για τα πιο επιδραστικά χαρακτηριστικά (Length, DeciDig_Freq), το οποίο πιθανώς οφείλεται σε χαρακτηριστικές τιμές των χαρακτηριστικών αυτών για την οικογένεια αυτή. Από την άλλη για την arithmetic-based οικογένεια Banjori υπήρχε έντονη ασυμφωνία μεταξύ των επιδραστικότερων χαρακτηριστικών ανάμεσα στους δύο ταξινομητές, παρά το γεγονός ότι και οι δύο ταξινομητές ανιχνεύουν την οικογένεια με αρκετά καλή ακρίβεια. Ιδιαίτερη περίπτωση αποτέλεσε η wordlist-based οικογένεια Suppobox, που ενώ παρατηρήθηκαν ομοιότητες στα πιο επιδραστικά χαρακτηριστικά αυτά είχαν εντελώς διαφορετική επίδραση στους δύο ταξινομητές, με τον binary να ταξινομεί επανειλημμένα τα ονόματα της κλάσης ως καλόβουλα και τον multiclass να τα ανιχνεύει με μεγαλύτερη ακρίβεια από τον binary παρά τη συνολικά χαμηλότερη απόδοση του. Τέλος, παρατηρήσαμε πως η επιλογή διαφορετικών XTIs για την εξαγωγή ερμηνειών δεν επηρεάζει σημαντικά τα αποτελέσματα που προκύπτουν και ειδικά στην περίπτωση του ταξινομητή πολλών κλάσεων φαίνεται να μην επηρεάζουν σχεδόν καθόλου.

5.2 Μελλοντική Μελέτη

Όσον αφορά τους μελλοντικούς μας στόχους, αρχικά, θα θέλαμε να επεκτείνουμε την ανάλυση της παρούσας εργασίας σε νευρωνικά δίκτυα (π.χ. Multi-Layer Perceptrons (MLPs), Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNNs)) για να μπορέσουμε να συγκρίνουμε τα αποτελέσματα των παραπάνω πειραμάτων με άλλα, ενδεχομένως αποτελεσματικότερα, μοντέλα. Επίσης, θα θέλαμε να εμβαθύνουμε περισσότερο στο κομμάτι του XAI, είτε με περισσότερες οπτικοποιήσεις από τη SHAP, είτε με άλλες μεθόδους όπως η LIME και τα Intefrated Gradients, καθώς αποτελεί έναν ιδιαίτερα αναπτυσσόμενο και πολλά υποσχόμενο τομέα. Τέλος, θα θέλαμε να εξετάσουμε τη χρήση Large Language Models - LLMs για τον αποδοτικότερο εντοπισμό ονομάτων.

Παραρτήματα

Παράρτημα Α΄

Οικογένειες DGA που περιλαμβάνονται στα έτη 2010-2019

Έτος	Οικογένειες
2010	bamital, vidro, gozi, murofet, murofetweekly, mydoom, szribi, torpig, conficker, gameover
2011	bamital, vidro, gozi, murofet, murofetweekly, mydoom, szribi, torpig, conficker, sutra, gameover
2012	bamital, sisron, gozi, murofet, murofetweekly, mydoom, vidro, pushdo, nymaim, chinad, szribi, torpig, conficker, blackhole, sutra, gameover
2013	gozi, pushdo, torpig, nymaim, pykspa, gameover, sisron, vidro, murofet, mydoom, suppobox, virut, qakbot, cryptolocker, blackhole, qadars, oderoor, necurs, conficker, chinad, sutra, bamital, murofetweekly, matsnu, szribi
2014	gozi, pushdo, torpig, infy, nymaim, pykspa, gameover, sisron, symmi, vidro, murofet, mydoom, suppobox, virut, qakbot, cryptolocker, blackhole, qadars, oderoor, necurs, conficker, chinad, sutra, bamital, murofetweekly, emotet, dyre, matsnu, szribi
2015	gozi, pushdo, torpig, infy, nymaim, pykspa, gameover, proslikefan, sisron, symmi, vidro, murofet, madmax, mydoom, suppobox, corebot, virut, qakbot, tofsee, cryptolocker, xshellghost, blackhole, qadars, ud2, ekforward, tempedrevetdd, ranbyus, locky, bedep, oderoor, necurs, conficker, chinad, sutra, bamital, murofetweekly, emotet, dyre, modpack, matsnu, szribi, pitou
2016	gozi, pushdo, torpig, infy, nymaim, pykspa, gameover, proslikefan, sisron, symmi, vidro, murofet, goznym, madmax, mydoom, padcrypt, suppobox, corebot, virut, qakbot, tofsee, diamondfox, cryptolocker, xshellghost, blackhole, qadars, ud2, ekforward, tempedrevetdd, ranbyus, locky, mirai, bedep, oderoor, sphinx, necurs, pandabanker, conficker, chinad, sutra, bamital, murofetweekly, emotet, dyre, modpack, matsnu, szribi, pitou

2017	gozi, pushdo, torpig, infy, nymaim, pykspa, gameover, ccleaner, proslikefan, sisron, symmi, murofet, goznym, madmax, mydoom, padcrypt, suppobox, tinynuke, vidro, corebot, virut, qakbot, tofsee, diamondfox, cryptolocker, xshellghost, blackhole, qadars, ud2, ekforward, tempedrevetdd, ranbyus, locky, mirai, bedep, oderoor, sphinx, necurs, pandabanker, conficker, chinad, sutra, bamital, murofetweekly, emotet, wd, dyre, modpack, matsnu, szribi, pitou
2018	gozi, pushdo, torpig, infy, nymaim, pykspa, gameover, ccleaner, proslikefan, sisron, symmi, murofet, madmax, mydoom, padcrypt, tinynuke, suppobox, vidro, corebot, virut, qakbot, tofsee, diamondfox, cryptolocker, xshellghost, blackhole, qadars, ud2, ekforward, tempedrevetdd, ranbyus, locky, mirai, bedep, monerominer, oderoor, sphinx, necurs, pandabanker, conficker, nymaim2, chinad, sutra, bamital, murofetweekly, emotet, wd, dyre, modpack, matsnu, szribi, pitou
2019	gozi, pushdo, torpig, infy, nymaim, qsnatch, pykspa, gameover, ccleaner, proslikefan, sisron, symmi, murofet, madmax, mydoom, padcrypt, tinynuke, suppobox, vidro, corebot, virut, tofsee, diamondfox, cryptolocker, xshellghost, qadars, ud2, ekforward, ranbyus, locky, mirai, monerominer, oderoor, sphinx, necurs, pandabanker, conficker, nymaim2, chinad, sutra, bamital, murofetweekly, emotet, wd, dyre, modpack, matsnu, szribi, pitou, pykspa2

Βιβλιογραφία

[1] *Decision Tree*. https://www.hackerearth.com/practice/machine-learning/machine-learning-algorithms/ml-decision-tree/tutorial/. Ημερομηνία πρόσβασης: 18-05-2024.

[2] Marco Tulio Ribeiro, Sameer Singh και Carlos Guestrin. *" Why should i trust you?" Explaining the predictions of any classifier*. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, σελίδες 1135–1144, 2016.

[3] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński και Wouter Joosen. *Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation*. *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, NDSS 2019, 2019.

[4] *DGArchive Repository Access Portal*. https://shap.readthedocs.io/en/latest/index.html. Ημερομηνία πρόσβασης: 10-06-2024.

[5] *What is DNS? | How DNS works*. https://www.cloudflare.com/learning/dns/what-is-dns/. Ημερομηνία πρόσβασης: 15-05-2024.

[6] Maryam Feily, Alireza Shahrestani και Sureswaran Ramadass. *A Survey of Botnet and Botnet Detection*. *2009 Third International Conference on Emerging Security Information, Systems and Technologies*, σελίδες 268–273, 2009.

[7] Basudev Saha και Ashish Gairola. *Botnet: an overview*. *CERT-In White Paper, CIWP-2005-05*, 240, 2005.

[8] *Botnet: A quick guide to botnets - what they are, how they work and the harm they can cause)*. https://www.f-secure.com/v-descs/articles/botnet.shtml. Ημερομηνία πρόσβασης: 23-06-2024.

[9] Nicholas Ianelli και Aaron Hackworth. *Botnets as a vehicle for online crime*. *CERT Coordination Center*, 1(1):28, 2005.

[10] Gernot Vormayr, Tanja Zseby και Joachim Fabini. *Botnet communication patterns*. *IEEE Communications Surveys & Tutorials*, 19(4):2768–2796, 2017.

[11] Ashley Hansberry, A Lasse και Andrew Tarrh. *Cryptolocker: 2013's most malicious malware*. *Retrieved February*, 9:2017, 2014.

[12] Niall Fitzgibbon και Mike Wood. *Conficker. C: A technical analysis*. *Sophos Labs, Sophos Inc*, 1, 2009.

[13] Paul Royal. *Analysis of the kraken botnet*, 2008.

[14] Daniel Plohmann, Khaled Yakdan, Michael Klatt, Johannes Bader και Elmar Gerhards-Padilla. *A comprehensive measurement study of domain generating malware*. *25th USENIX Security Symposium (USENIX Security 16)*, σελίδες 263–278, 2016.

[15] Alessandro Cucchiarelli, Christian Morbidoni, Luca Spalazzi και Marco Baldi. *Algorithmically generated malicious domain names detection based on n-grams features*. *Expert Systems with Applications*, 170:114551, 2021.

[16] *What is machine learning and how does it work? In-depth guide*. https://www.techtarget.com/searchenterpriseai/definition/machine-learning-ML. Ημερομηνία πρόσβασης: 18-05-2024.

[17] *Decision Tree in Machine Learning*. https://www.geeksforgeeks.org/decision-tree-introduction-example/. Ημερομηνία πρόσβασης: 18-05-2024.

[18] *What is random forest?* https://www.ibm.com/topics/random-forest. Ημερομηνία πρόσβασης: 03-06-2024.

[19] Leo Breiman. *Random forests*. *Machine learning*, 45:5–32, 2001.

[20] S. Lloyd. *Least squares quantization in PCM*. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[21] *The Ultimate Guide to K-Means Clustering: Definition, Methods and Applications*. https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/#:~:text=K%2Dmeans%20is%20a%20centroid,is%20associated%20with%20a%20centroid. Ημερομηνία πρόσβασης: 03-06-2024.

[22] *Elbow Method (clustering)*. https://en.wikipedia.org/wiki/Elbow_method_(clustering). Ημερομηνία πρόσβασης: 23-06-2024.

[23] Christoph Molnar. *A guide for making black box models explainable*. *URL: https://christophm. github. io/interpretable-ml-book*, 2(3):10, 2018.

[24] Leonida Gianfagna και Antonio Di Cecco. *Explainable AI with python*, τόμος 4. Springer, 2021.

[25] Scott M Lundberg και Su In Lee. *A unified approach to interpreting model predictions*. *Advances in neural information processing systems*, 30, 2017.

[26] *Shapley values: an introductory exampleWhat is the Shapley value ?* https://medium.com/the-modern-scientist/what-is-the-shapley-value-8ca624274d5a. Ημερομηνία πρόσβασης: 06-06-2024.

[27] *Shapley Value*. https://en.wikipedia.org/wiki/Shapley_value. Ημερομηνία πρόσβασης: 06-06-2024.

[28] Scott M Lundberg, Gabriel G Erion και Su In Lee. *Consistent individualized feature attribution for tree ensembles*. *arXiv preprint arXiv:1802.03888*, 2018.

[29] Dominik Janzing, Lenon Minorics και Patrick Blöbaum. *Feature relevance quantification in explainable AI: A causal problem*. *International Conference on artificial intelligence and statistics*, σελίδες 2907–2916. PMLR, 2020.

[30] Mukund Sundararajan και Amir Najmi. *The many Shapley values for model explanation*. *International conference on machine learning*, σελίδες 9269–9278. PMLR, 2020.

[31] *Thesis Github Repository*. https://github.com/fmm151/thesis/. Ημερομηνία πρόσβασης: 10-07-2024.

[32] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński και Wouter Joosen. *Tranco: A research-oriented top sites ranking hardened against manipulation*. *arXiv preprint arXiv:1806.01156*, 2018.

[33] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall και W Philip Kegelmeyer. *SMOTE: synthetic minority over-sampling technique*. *Journal of artificial intelligence research*, 16:321–357, 2002.

[34] *DSMOTE Documentation*. https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html. Ημερομηνία πρόσβασης: 10-06-2024.

[35] Nikos Kostopoulos, Dimitris Kalogeras, Dimitris Pantazatos, Maria Grammatikou και Vasilis Maglaris. *SHAP Interpretations of Tree and Neural Network DNS Classifiers for Analyzing DGA Family Characteristics*. *IEEE Access*, 11:61144–61160, 2023.

[36] Samuel Schüppen, Dominik Teubert, Patrick Herrmann και Ulrike Meyer. *{FANCI}: Feature-based automated {NXDomain} classification and intelligence*. *27th USENIX Security Symposium (USENIX Security 18)*, σελίδες 1165–1181, 2018.

[37] *Mozilla Public Suffix List*. https://publicsuffix.org/. Ημερομηνία πρόσβασης: 11-06-2024.

[38] *Wordninja GitHub Repository*. https://github.com/keredson/wordninja/. Ημερομηνία πρόσβασης: 12-06-2024.

[39] Wilhelm Kirch. *Pearson's correlation coefficient*. *Encyclopedia of public health*, 1:1090–1091, 2008.

[40] *Correlation Analysis)*. https://sphweb.bumc.bu.edu/otlt/mph-modules/bs/bs704_correlation-regression/bs704_correlation-regression2.html. Ημερομηνία πρόσβασης: 23-06-2024.

[41] *Random Forest Classifier Documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. Ημερομηνία πρόσβασης: 13-06-2024.

[42] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg και others. *Scikit-learn: Machine learning in Python*. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[43] *SHAP Documentation*. https://shap.readthedocs.io/en/latest/index.html. Ημερομηνία πρόσβασης: 14-06-2024.

[44] *SHAP TreeExplainer Documentation*. https://shap-lrjball.readthedocs.io/en/latest/generated/shap.TreeExplainer.html. Ημερομηνία πρόσβασης: 14-06-2024.

[45] *SHAP Summary Plots Documentation*. https://shap-lrjball.readthedocs.io/en/latest/generated/shap.summary_plot.htmll. Ημερομηνία πρόσβασης: 14-06-2024.

[46] *SHAP Force Plots Documentation*. https://shap.readthedocs.io/en/latest/generated/shap.plots.force.html. Ημερομηνία πρόσβασης: 14-06-2024.

[47] Manos Antonakakis, Roberto Perdisci, Yacin Nadji, Nikolaos Vasiloglou, Saeed Abu-Nimeh, Wenke Lee και David Dagon. *From {Throw-Away} traffic to bots: Detecting the rise of {DGA-Based} malware*. *21st USENIX Security Symposium (USENIX Security 12)*, σελίδες 491–506, 2012.

[48] Arthur Drichel, Nils Faerber και Ulrike Meyer. *First step towards explainable dga multiclass classification*. *Proceedings of the 16th International Conference on Availability, Reliability and Security*, σελίδες 1–13, 2021.

[49] *English Letter Frequency (based on a sample of 40,000 words)*. https://pi.math.cornell.edu/~mec/2003-2004/cryptography/subs/frequencies.html. Ημερομηνία πρόσβασης: 19-06-2024.

[50] Ahmad O Almashhadani, Mustafa Kaiiali, Domhnall Carlin και Sakir Sezer. *MaldomDetector: A system for detecting algorithmically generated domain names with machine learning*. *Computers & Security*, 93:101787, 2020.