ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

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

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΙΧΑΗΛΙΔΗΣ ΜΑΡΙΟΣ

Επιβλέπων: Βασίλειος Μάγκλαρης
Ομότιμος Καθηγητής Ε.Μ.Π.

Αθήνα, Μάρτιος 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΑΤΜΗΜΑΤΙΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΠΙΣΤΗΜΗ ΔΕΔΟΜΕΝΩΝ ΚΑΙ ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

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

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΜΙΧΑΗΛΙΔΗΣ ΜΑΡΙΟΣ

Επιβλέπων: Βασίλειος Μάγκλαρης
Ομότιμος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 28/03/2023

.....	.....	.....
Βασίλειος Μάγκλαρης	Ευστάθιος Συκάς	Γεώργιος Στάμου
Ομότιμος Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Μάρτιος 2023

Copyright © Μιχαηλίδης Μάριος, 2023
Με επιφύλαξη παντός δικαιώματος. All rights reserved.
Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

……..

ΜΙΧΑΗΛΙΔΗΣ ΜΑΡΙΟΣ
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών
Μάρτιος 2023

Περίληψη

Στόχος της παρούσας διπλωματικής είναι η μελέτη και η υλοποίηση αλγορίθμων επεξήγησης Τεχνητής Νοημοσύνης (eXplainable AI – XAI) για την ανίχνευση κακόβουλων ονομάτων που παράγονται από Domain Generation Algorithms με σκοπό την παραπλάνηση των administrators του Domain Name System (DNS) σε επιθέσεις από botnets. Το μεγαλύτερο ποσοστό των botnets χρησιμοποιούν Domain Generation Algorithms, για να αποκρύπτουν την ταυτότητα τους μέσω της περιοδικής εκτέλεσης των DGAs και συνεπώς την περιοδική αλλαγή του Domain Name που εκχωρείται στον C&C server. Ο κύριος στόχος αυτών των αλγορίθμων είναι η παραγωγή ενός μεγάλου ψευδοτυχαίου συνόλου ονομάτων τομέα και έπειτα η χρήση ενός υποσυνόλου αυτού για τον έλεγχο και την επικοινωνία μεταξύ του C&C server και των bots. Μ' αυτόν τον τρόπο οι botmasters ενισχύουν την δομή του botnet και καθιστούν ιδιαίτερα απαιτητική την διαδικασία εντοπισμού του C&C και την αποκοπή του από τα bots καθώς το σύστημα λόγω των περιοδικών αλλαγών είναι ανθεκτικό σε παραδοσιακά συστήματα ασφαλείας όπως το blacklisting. Παρά το γεγονός πως οι μέθοδοι μηχανικής και βαθιάς μάθησης γίνονται ολοένα και πιο δημοφιλείς στην αντιμετώπιση αυτού του προβλήματος και παρουσιάζουν εξαιρετικά αποτελέσματα όσον αφορά την ακρίβεια, εντούτοις παραμένουν «un-interpretable» (μη επεξηγήσιμες) και δύσκολες για τους ερευνητές να κατανοήσουν πως προκύπτουν οι αποφάσεις και οι προβλέψεις τους.

Στοχεύοντας στην επίλυση του παραπάνω προβλήματος, παρουσιάζουμε διάφορα μοντέλα μηχανικής μάθησης, δίνοντας έμφαση σε ταξινομητές βαθιάς μηχανικής μάθησης (Multilayer Perceptron - MLP, Long Short-Term Memory) ώστε να επεξηγήσουμε και να ερμηνεύσουμε τα χαρακτηριστικά που καθόρισαν την κατηγοριοποίηση των ονομάτων τομέα σε πραγματικά ή κακόβουλα. Η μελέτη μας ακολουθεί δύο διακριτές μεθοδολογίες αξιοποίησης της πληροφορίας του δείγματος μας. Στην πρώτη, εξαγάγαμε στατιστικά χαρακτηριστικά για τα domain names, όπως το μέγεθος του ονόματος κ.ά. και χρησιμοποιήσαμε Δέντρα Αποφάσεων (ακρίβεια 93%) για την ανάλυση του προβλήματος, καθώς τα αποτελέσματα τους θεωρητικά είναι intrinsically-explainable. Με τα ίδια χαρακτηριστικά, εκπαιδεύσαμε ένα MLP (ακρίβεια 90%) το οποίο μας έδωσε την δυνατότητα να αναλύσουμε αποδοτικότερα τόσο τις σωστές, όσο και τις λανθασμένες αποφάσεις του μοντέλου μας. Αντίθετα, στην δεύτερη, αξιοποιήσαμε το δίκτυο LSTM (ακρίβεια 99%), ώστε να μπορέσουμε να εκμεταλλευτούμε την πραγματική ακολουθία των αλφαριθμητικών χαρακτήρων ενός ονόματος τομέα και όχι απλά στατιστικά χαρακτηριστικά. Συνεπώς, με αυτή την προσέγγιση μας δίνεται η δυνατότητα να αντιληφθούμε τα n-grams που καθοδηγούν τις αποφάσεις του δικτύου μας. Τέλος, επεκτείναμε την υλοποίηση με LSTM με μια multi-class προσέγγιση, όπου χρησιμοποιήσαμε κάποιες από τις οικογένειες DGA, ώστε να γίνει ένα πρώτο βήμα στην ουσιαστικότερη κατανόηση του τρόπου παραγωγής τους, καθώς εντοπίζουμε τα n-grams που ωθούν τα μοντέλα μας όχι μόνο να ανιχνεύσουν κακόβουλη δικτυακή κίνηση, αλλά και από ποιον αλγόριθμο προήλθε.

Παράλληλα, χρησιμοποιούμε το framework του SHAP (SHapley Additive exPlanations) ώστε να ερμηνεύσουμε τα χαρακτηριστικά με την περισσότερη επιρροή στα μοντέλα μας και να ποσοτικοποιήσουμε την συνεισφορά τους σε κάθε παράδειγμα ξεχωριστά (τοπική επεξήγηση) και στο σύνολο του δείγματος μας (γενική επεξήγηση). Συγκρίνοντας τις δύο μεθοδολογίες αναγνωρίζουμε ότι τα στατιστικά χαρακτηριστικά ναι μεν είναι πιο εύκολα κατανοητά από τον άνθρωπο, ωστόσο κρύβουν παθογένειες που βασίζονται στην μη αξιοποίηση του πραγματικού ονόματος τομέα, κάτι που επιλύεται ουσιαστικά από την δεύτερη μεθοδολογία (χρήση n-grams), η οποία παρέχει υψηλότερη ακρίβεια αλλά και βαθύτερα αποτελέσματα για την κατανόηση της λειτουργίας των DGA.

Λέξεις-Κλειδιά: Botnet, Domain Generation Algorithm (DGA), Βαθιά Μηχανική Μάθηση, eXplainable AI (XAI), SHAP

Abstract

The aim of this thesis is to study and implement eXplainable AI algorithms for the detection of malicious names generated by Domain Generation Algorithms in order to mislead the Domain Name System in botnet attacks. Nowadays, most botnets use Domain Generation Algorithms to hide their identity, through the periodic change of the Domain Name assigned to the C&C Server. The main goal of these algorithms is a large pseudo-random set of domain names and then use a subset of that for control and communication between the C&C server and the bots. In this way, botmasters strengthen the infrastructure of the botnet and make the process of identifying C&C and cutting it off from the bots particularly demanding, as the system, due to periodic changes, is resilient to traditional security systems such as blacklisting. Despite the fact that, machine and deep learning methods are becoming more popular in dealing with this problem and they show excellent results in terms of accuracy, yet they remain un-interpretable and difficult for researchers to understand how their decisions and predictions are made.

Aiming to solve the above problem, we present various machine learning models, emphasizing into deep machine learning classifiers (Multilayer Perceptron – MLP, Long Short-Term Memory - LSTM) in order to explain and interpret the features that determined the classification of domain names in legit or malicious. Our study follows two distinct methodologies of using the information of our sample. In the first one, we extracted manually statistical features about the domain names, such as the length of the domain name, etc. and we used Decision Tree (accuracy 93%) to analyze the problem, as its results are theoretically intrinsically-explanatory. Using the same features, we trained an MLP (accuracy 90%) which enables us to more efficiently analyze and explain both the correct and incorrect decisions of our model. In the second one, we leveraged the LSTM network (accuracy 99%) to be able to exploit the actual sequence of alphanumeric characters of a domain name and not just statistical features. Therefore, with this approach we are given the possibility to perceive the n-grams that guide the decisions of our network. Finally, we extended the LSTM implementation with a multi-class approach, where we used some of the DGA families, in order to take a first step in a more substantial understanding of how they are generated, as we identify the n-grams that drive our models not only to detect malicious traffic network, but also from which algorithm it generated.

We use the SHAP (SHapley Additive exPlanations) framework to interpret the most influential features in our models and quantify their contribution to each instance individually (local explanation) and to our sample as a whole (global explanation). By comparing the two methodologies, we recognize that the statistical features are easier to understand by humans, but they hide weaknesses based on not using the real domain name, which is essentially solved by the second methodology, which provides higher accuracy and deeper results to understand the operation of DGAs

Keywords: Botnet, Domain Generation Algorithm (DGA), Deep Learning, eXplainable AI (XAI), SHAP

Ευχαριστίες

Θα ήθελα καταρχάς να ευχαριστήσω τον Ομότιμο Καθηγητή Ε.Μ.Π. Βασίλειο Μάγκλαρη που μου ανέθεσε την παρούσα διπλωματική εργασία και μου έδωσε την ευκαιρία να την εκπονήσω στο εργαστήριο NETMODE (Network Management & Optimal Design Laboratory). Εν συνεχεία, θα ήθελα να ευχαριστήσω τον Καθηγητή κ. Συκά Ευστάθιο και τον Καθηγητή κ. Στάμου Γεώργιο για την τιμή που μου έκαναν να συμμετάσχουν στην τριμελή εξεταστική επιτροπή της εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω ιδιαιτέρως τον ερευνητή και Υποψήφιο Διδάκτωρ κ. Κωστόπουλο Νικόλαο που ήταν αρωγός καθ' όλη την διάρκεια διεξαγωγής της συγκεκριμένης διπλωματικής με τις πολύτιμες συμβουλές και υποδείξεις του.

Τέλος, θα ήθελα να ευχαριστήσω τους γονείς μου για την συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια, αλλά και τους φίλους και συμφοιτητές μου Δοντά Σπυρίδων και Σαντοριναίο Χριστόδουλο για την συνεχή υποστήριξη και όλα αυτά τα υπέροχα φοιτητικά χρόνια.

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1 – Εισαγωγή 14
1.1 Αντικείμενο της Διπλωματικής – Εισαγωγικά Στοιχεία 14
1.2 Οργάνωση Κειμένου 17
Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο 19
2.1 Η δομή ενός botnet και οι κακόβουλες δραστηριότητές του 19
2.2 DGA-based botnets 20
2.3 Domain Generation Algorithms (DGA) 21
2.4 Σύστημα Ονοματοδοσίας Τομέων – Domain Name System (DNS) 23
2.5 Μηχανισμοί ανίχνευσης ονομάτων από DGA κι η ανάγκη της επεξηγησιμότητας (explainability) 26
2.6 eXplainability Artificial Intelligence (XAI) 28
Κεφάλαιο 3 – Συναφής Βιβλιογραφία 37
3.1 Ανασκοπική (Retrospective) Ανίχνευση DGA 40
3.2 Real-Time Ανίχνευση DGA με Μηχανική Μάθηση 41
3.3 Χρήση eXplainability AI για τον εντοπισμό DNS κίνησης μέσω DGA 43
3.4 Συνεισφορά της Παρούσας Διπλωματικής 45
Κεφάλαιο 4 – Dataset και Μεθοδολογία Επιλεγμένων Μοντέλων Μηχανικής Μάθησης 46
4.1 Binary Dataset 46
4.2 Multiclass Dataset 46
4.3 Decision Tree – Binary Classification 47
4.4 Multilayer Perceptron – Binary Classification 49
4.5 LSTM – Binary and Multiclass Classification 51
Κεφάλαιο 5 – Πειραματική Αξιολόγηση 58
5.1 Κριτήρια Αξιολόγησης των Μοντέλων Μάθησης 58
5.2 Decision Tree, Intrinsically-Explainable – Binary Classification 59
5.3 MLP, SHAP Explainability – Binary Classification 61
5.4 LSTM, SHAP Explainability – Binary Classification 65
5.5 LSTM, SHAP Explainability – Multiclass Classification 69
Κεφάλαιο 6 – Συμπεράσματα και Μελλοντική Μελέτη 73
6.1 Ανασκόπηση και Συμπεράσματα 73
6.2 Μελλοντική Μελέτη 74
Κεφάλαιο 7 – Βιβλιογραφία 75

ΚΕΦΑΛΑΙΟ 1 – Εισαγωγή

1.1 Αντικείμενο της Διπλωματικής – Εισαγωγικά Στοιχεία

Στην εποχή της ψηφιοποίησης, η τεχνολογία είναι καθημερινά ολοένα και πιο διαθέσιμη. Οι περισσότεροι άνθρωποι είναι πλέον «εξαρτημένοι» από ψηφιακές υπηρεσίες και γενικότερα ο αριθμός αυτών που είναι συνδεδεμένοι στο διαδίκτυο αυξάνεται ταχύτατα. Ακόμη και σε επίπεδο επιχειρήσεων αν αναλογιστούμε την κατάσταση, όλοι προσφέρουν κάποιου είδους πληροφορία μέσω του ιστού. Χαρακτηριστικά σύμφωνα με έρευνες της Cisco, πάνω από το 70% του παγκόσμιου πληθυσμού θα είναι συνδεδεμένοι στο διαδίκτυο εντός του 2023[1]. Αυτή η μαζική εισροή χρηστών προφανώς έχει οδηγήσει από την μία στην παροχή όλο και πιο ανταγωνιστικών υπηρεσιών, αλλά δυστυχώς και στην δημοσίευση και διακίνηση αμφιλεγόμενης πληροφορίας στο διαδίκτυο. Συνεπώς, αυτό δίνει χώρο σε κακόβουλους χρήστες να προσπαθήσουν να επωφεληθούν αυτής της μαζικής έκρηξης και εξάπλωσης του διαδικτύου, με στόχο να διενεργήσουν διάφορες κακόβουλες δραστηριότητες, όπως την υποκλοπή προσωπικών δεδομένων (π.χ. τραπεζικοί κωδικοί) ή κατανεμημένες επιθέσεις άρνησης υπηρεσιών (Distributed Denial of Service Attacks), οι οποίες σύμφωνα με την ίδια πηγή [1] διπλασιάστηκαν από το 2018 μέχρι το 2023. Ωστόσο, όσο κι αν οι κυβερνοεπιθέσεις γίνονται πιο ισχυρές και πιο σύνθετες, οφείλουμε να βρούμε νέους τρόπους να τις ανιχνεύσουμε και να τις περιορίσουμε.

Περνώντας σταδιακά προς το αντικείμενο αυτής της διπλωματικής, το Σύστημα Ονοματοδοσίας Τομέων ευρέως γνωστό ως Domain Name System (DNS) είναι ένα ιεραρχικό και κατανεμημένο σύστημα ονοματοδοσίας για δίκτυα υπολογιστών, απαραίτητο για την ομαλή λειτουργία του internet, καθώς περιέχει αντιστοιχίες ονομάτων και αριθμητικών διευθύνσεων IPv4, IPv6, υπηρεσιών ή άλλων δικτυακών πόρων και τυγχάνει καθολικής αποδοχής και χρήσης. Γίνεται εύκολα αντιληπτό ότι η κίνηση που διέρχεται μέσω του DNS είναι καίριας σημασίας για πολλά συστήματα ασφαλείας και όχι μόνο, καθώς σε κάθε εφαρμογή πρέπει να αντιστοιχιστεί ένα domain name πριν γίνει οποιαδήποτε άλλη σύνδεση. Εξού και η υπηρεσία ονοματοδοσίας αποτελεί κυρίαρχο στόχο για επίδοξους hackers ή κακόβουλους χρήστες, καθώς τους επιτρέπει πρωτογενή πρόσβαση σε δίκτυα και δεδομένα για υποκλοπή [2]. Για αυτό τον σκοπό υπάρχουν συστήματα ανίχνευσης εισβολής (IDS) τα οποία εντοπίζουν ύποπτες δραστηριότητες και παράγουν ειδοποιήσεις όταν εντοπίζουν κάτι.

Η τεχνολογία των botnets έχει γίνει το πρωταρχικό μέσο για τους συντονιστές κυβερνοεπιθέσεων ώστε να εκτελέσουν τις κακόβουλες δραστηριότητες τους, όπως οι επιθέσεις DDoS, η αποστολή spam κ.ά. Μάλιστα, σύγχρονες έρευνες καταδεικνύουν την ύπαρξη botnets που αποτελούνται από εκατομμύρια μολυσμένες μονάδες (bots), αποτυπώνοντας το μέγεθος της απειλής[3]. Οι κακόβουλοι διαχειριστές (botmasters) επιλέγουν την χρήση Domain Generation Algorithms, μια πολύ ευέλικτη μέθοδο για

την επικοινωνία στα botnets, με στόχο την επικοινωνία του C&C (Command and Control) Server με τα bots του, ώστε να μπορεί να τα ελέγχει και να στέλνει εντολές με κατανεμημένο τρόπο. Στόχος κάθε botmaster είναι η απόκρυψη κυρίως της δικής του ταυτότητας αλλά και των bots του, ώστε να μπορέσει να λειτουργήσει το botnet του και να μην ανιχνευθεί από αμυντικούς μηχανισμούς ανίχνευσης κακόβουλης δικτυακής κίνησης. Οι Domain Generation Algorithms (DGA) λαμβάνουν ως είσοδο ένα seed, το οποίο είναι κοινό στα bots και στον C&C και εκτελούνται περιοδικά σε όλους τους κόμβους του botnet παράγοντας ένα ψευδο-τυχαίο σύνολο από domain names. Ακολούθως, εκχωρούνται (register) στους C&C servers ορισμένα από τα παραγόμενα domain names, με τον φθηνότερο δυνατό τρόπο και με ελαστικούς κανόνες εκχώρησης. Τα bots με την σειρά τους εκτελούν DNS queries μέχρι να γίνουν resolve οι διευθύνσεις και να συνδεθούν στον C&C, διαδικασία η οποία παράγει πολλά NXDomain responses. Αυτό το σχήμα λειτουργίας των botnets με την εκμετάλλευση των DGAs καθιστά αναποτελεσματικά τα παραδοσιακά συστήματα ασφαλείας, διότι είναι στατικά, όπως το blacklisting των domain names ενός C&C server μόλις εντοπιστεί, ενώ στην πραγματικότητα το domain name του αλλάζει περιοδικά οπότε αδυνατούν να τον ανιχνεύσουν έγκαιρα. Ένας ακόμη κλασσικός μηχανισμός άμυνας είναι το reverse engineering για την ανακατασκευή όσο είναι δυνατό του DGA που χρησιμοποιείται από μια μολυσμένη συσκευή. Στόχος του είναι τόσο η μελέτη του τρόπου παραγωγής των κακόβουλων ονομάτων όσο και η πρόβλεψη μελλοντικών, ωστόσο είναι ιδιαίτερα χρονοβόρα τεχνική και απρόβλεπτη, εάν το seed του DGA δεν μπορεί να προσδιοριστεί εύκολα, π.χ. Tweets. Ιδανικά, θα θέλαμε η ανίχνευση της κακόβουλης δικτυακής κίνησης να εκτελείται σε μεμονωμένο επίπεδο ονομάτων, ώστε να μπορεί να αποκόπτεται αμεσότερα.

Στοχεύοντας στην επίλυση του παραπάνω προβλήματος, οι ερευνητές που ασχολούνται με την κυβερνοασφάλεια και τις επιθέσεις στο διαδίκτυο έστρεψαν αρχικά την προσοχή τους και τις προσπάθειες τους σε μεθόδους μηχανικής μάθησης. Κάτι τέτοιο είναι απόλυτα λογικό καθώς οι αλγόριθμοι μηχανικής μάθησης χρησιμοποιούνται συχνά σε προβλήματα ανίχνευσης ανωμαλιών (anomaly detection) σε network εφαρμογές. Έτσι είχαμε απλούστερα μοντέλα αλλά επεξηγήσιμα για μικρό αριθμό δειγμάτων. Οι παραπάνω υλοποιήσεις βασίζονται στην εξαγωγή στατιστικών χαρακτηριστικών που επιλέγει ο εκάστοτε ερευνητής για την μελέτη των domain names. Οι σύγχρονες απαιτήσεις ωστόσο, δημιούργησαν την ανάγκη για μεγαλύτερη ακρίβεια και καλύτερα αποτελέσματα, συνεπώς αναπτύχθηκαν πιο σύνθετα, πιο ακριβά και σαφώς πιο περίπλοκα μοντέλα για την ανίχνευση κακόβουλης δικτυακής κίνησης, τα οποία δυστυχώς είναι μη επεξηγήσιμα λόγω της πολυπλοκότητας τους. Τέτοιου είδους ανιχνευτές βασίζονται σε τεχνικές Supervised Deep Learning και εκμεταλλεύονται κατά κόρον την πραγματική αλφαριθμητική ακολουθία ενός ονόματος και όχι στατιστικά χαρακτηριστικά που έχει επιλέξει ο άνθρωπος, με στόχο την κατηγοριοποίηση κάθε δειγματικού στοιχείου ως πραγματικό κι έγκυρο (legit) είτε ως κακόβουλο το οποίο έχει παραχθεί από DGA. Η αξιοποίηση της πραγματικής ακολουθίας έφερε μεγάλη επιτυχία στην χρήση τέτοιου είδους μοντέλων καθώς μειώθηκε σημαντικά η ικανότητα των κακόβουλων χρηστών να ξεγελάσουν τις

μεθόδους ανίχνευσης συγκριτικά με την περίπτωση των εξαγόμενων στατιστικών χαρακτηριστικών που επιλέγει ο άνθρωπος και συνεπώς δεν έχουν την δυνατότητα να βελτιστοποιούν τους DGA που δεν εντοπίζονται με βάση αυτά τα χαρακτηριστικά. Επομένως, τα Deep Learning μοντέλα επιτυγχάνουν εξαιρετική ακρίβεια, όπως θα δούμε τόσο στην βιβλιογραφία όσο και στην πράξη, ωστόσο αποτελούν black-box συστήματα που δεν είναι ερμηνεύσιμα από τον άνθρωπο. Κι εδώ υπεισέρχεται η ανάγκη για επεξηγησιμότητα (explainability) για την μελέτη μεθόδων και μηχανισμών για ανίχνευση DGA. Αυτή η ανάγκη αναφέρεται στους operators του δικτύου για να μπορούν να απαντούν στο ερώτημα γιατί ανιχνεύθηκε ως DGA, στους προγραμματιστές που αναπτύσσουν τα μοντέλα για να τα ελέγχουν και να κάνουν debugging και τέλος σε νομικές οντότητες (legal entities) ώστε να βεβαιώνονται ότι συμμορφώνονται με τον γενικό κανονισμό για την προστασία δεδομένων (GDPR).

Εφόσον λοιπόν, έχουμε μοντέλα μάθησης που είναι πολλά υποσχόμενα για την επίλυση του προβλήματος, οφείλουμε να τα πλαισιώσουμε και με τους καλύτερους αλγορίθμους επεξηγησιμότητας, ώστε να κατανοήσουμε τις αποφάσεις και τις προβλέψεις που προκύπτουν από την Τεχνητή Νοημοσύνη, δηλαδή να μεταφερθούμε από ένα black-box σ' ένα white-box. Οι περισσότερες εργασίες σχετικά με την επεξηγησιμότητα μέχρι και σήμερα έχουν εστιάσει κυρίως σε κλάδους όπως η όραση υπολογιστών και η επεξεργασία φυσικής γλώσσας, ωστόσο σταδιακά γίνονται βήματα και στον χώρο της κυβερνοασφάλειας, ώστε να βελτιστοποιήσουν οι ειδικοί τις αποφάσεις τους σύμφωνα με την γνώμη του μοντέλου. Για την παρούσα διπλωματική εργασία θα χρησιμοποιήσουμε το framework του SHAP (SHapley Additive exPlanations), το οποίο συνδυάζει τόσο την τοπική επεξηγησιμότητα (local explainability) ανά δειγματικό στοιχείο, όσο και την γενικότερη στο επιλεγμένο δείγμα μας (global explainability) με στόχο και των 2 την ερμηνεία των αποτελεσμάτων. Οι τοπικές επεξηγήσεις εξηγούν τους λόγους του γιατί το μοντέλο έλαβε κάποιες αποφάσεις για συγκεκριμένη είσοδο, ενώ το global explainability υποδεικνύει τα σημαντικά χαρακτηριστικά που επηρέασαν σε ευρύτερο βαθμό τις κατηγοριοποιήσεις του μοντέλου. Πολύ σύντομα, να αναφέρουμε ότι το framework του SHAP βασίζεται στις Shapley τιμές που θα αναλύσουμε εκτενέστερα παρακάτω, οι οποίες αναφέρονται στον υπολογισμό της οριακής συνεισφοράς κάθε χαρακτηριστικού στην λειτουργία του μοντέλου.

Πιο συγκεκριμένα, στην παρούσα διπλωματική εργασία θα αναπτύξουμε και θα παρουσιάσουμε δυαδικούς ανιχνευτές για έγκυρα και κακόβουλα ονόματα. Η διαδρομή ξεκινάει από ένα απλούστερο μοντέλο με Δέντρο Αποφάσεων, με την χρήση στατιστικών χαρακτηριστικών, όπου θα μελετηθούν τα αποτελέσματα που λόγω του τρόπου λειτουργίας της μεθόδου αναμένουμε να είναι intrinsically-explainable (εγγενώς επεξηγήσιμο). Ωστόσο, στην πράξη για μεγάλα μεγέθη και τυχαιότητα δεν είναι ερμηνεύσιμο ένα Δέντρο Αποφάσεων και αδυνατεί να δώσει και τοπική επεξηγησιμότητα. Περνώντας στην Βαθιά Μάθηση που θεωρούμε ότι είναι και η κύρια συνεισφορά μας, παρουσιάζουμε αρχικά ένα MLP μοντέλο που βασίζεται στα ίδια χαρακτηριστικά με το Decision Tree. Παρατηρούμε εξίσου καλή ακρίβεια και σε αυτή την περίπτωση, και σαφώς καλύτερη διαχείριση του όγκου των δεδομένων που μας

βοηθάει με την αξιοποίηση του SHAP να έχουμε μια πρώτη γεύση για την ερμηνεία των αποτελεσμάτων, σαφώς ισχυρότερη συγκριτικά με τα Δέντρα Αποφάσεων. Δυστυχώς όμως, ακόμη κι έτσι θα παρατηρήσουμε ότι το μοντέλο πάσχει από False Negative περιπτώσεις, όπου κατηγοριοποιεί ως έγκυρο όνομα τομέα κάποιο κακόβουλο. Αυτό προκύπτει διότι τα στατιστικά χαρακτηριστικά είναι επιλεγμένα από τον άνθρωπο και προφανώς δεν μπορούν να καλύψουν όλες τις περιπτώσεις όπως θα δούμε στα πειράματα μας. Κλείνοντας το κομμάτι των δυαδικών ανιχνευτών δικτυακής κίνησης, στην προσπάθεια επίλυσης όλων των παραπάνω υλοποιούμε ένα LSTM δίκτυο το οποίο αξιοποιεί την συμβολοσειρά του κάθε domain name, αναλυμένη σε επικαλυπτόμενα bigrams και trigrams αντίστοιχα. Η ακρίβεια του μοντέλου αγγίζει το 99%, το οποίο ήταν κάτι που αναμέναμε, αλλά το ιδιαίτερα θετικό κι ελπιδοφόρο έρχεται από την εφαρμογή του SHAP πάνω σε αυτό το μοντέλο που μας επιτρέπει να αναγνωρίσουμε τα πιο σημαντικά βάσει επιρροής διγράμματα και τριγράμματα. Έτσι, μπορούμε να αρχίσουμε να κατανοούμε καλύτερα τον τρόπο λειτουργίας των DGAs, να τους ανιχνεύουμε αποδοτικότερα και να φτιάξουμε μηχανισμούς φιλτραρίσματος με βάση τις αποφάσεις του νευρωνικού δικτύου μας. Τέλος, εφόσον η παραπάνω υλοποίηση εμφανίστηκε πολλά υποσχόμενη, αποφασίσαμε να την επεκτείνουμε με μια multi-class προσέγγιση, όπου απομονώνουμε ένα υποσύνολο από τις οικογένειες των DGA και πλέον κατηγοριοποιούμε μεταξύ των κακόβουλων αλγορίθμων παραγωγής ονομάτων τομέα. Στόχος αυτής της προσπάθειας, δεν ήταν αποκλειστικά η ακρίβεια (99%) αλλά η δυνατότητα παραγωγής ενός επεξηγήσιμου μοντέλου που μας αναλύει και ερμηνεύει ποια χαρακτηριστικά (εδώ bigrams και trigrams) ώθησαν το νευρωνικό να κατηγοριοποιήσει τους εκάστοτε αλγορίθμους, συνεπώς τι ήταν αυτό που τους ξεχώρισε και επηρέασε σημαντικά για την επιλογή του καθενός από την ταξινόμηση.

1.2 Οργάνωση Κειμένου

Το υπόλοιπο της διπλωματικής αυτής οργανώνεται σε 5 ακόμη κεφάλαια και την βιβλιογραφία:

- Το κεφάλαιο 2 περιέχει το σχετικό υπόβαθρο αυτής της διπλωματικής, όπου παρουσιάζονται και εξηγούνται οι έννοιες του αντικειμένου που μελετάμε (botnets, DGA, DNS, XAI).
- Στο κεφάλαιο 3 παρουσιάζεται η συναφής βιβλιογραφία και η σχετική έρευνα που προηγήθηκε της συγγραφής του παρόντος τόμου, καθώς και η συνεισφορά της παρούσας εργασίας συγκριτικά με την μέχρι σήμερα βιβλιογραφία.
- Στο κεφάλαιο 4 περιγράφουμε τα πειραματικά δεδομένα που χρησιμοποιήσαμε (dataset) και την απαραίτητη προ-εξεργασία που έγινε. Ακολούθως, αναλύουμε την μεθοδολογία για την υλοποίηση των ανιχνευτών (Decision Tree, MLP, LSTM), όπου εξηγούμε την αρχιτεκτονική τους και το πειραματικό μας σύστημα, δηλαδή την σύνδεση των black-box μοντέλων με τα επεξηγήσιμα μοντέλα που μας παρέχει το framework του SHAP.

- Το κεφάλαιο 5 αποτελείται από την παρουσίαση, την αξιολόγηση και τον σχολιασμό των πειραμάτων μας για την εξαγωγή ποσοτικών αλλά και ποιοτικών συμπερασμάτων.

- Το κεφάλαιο 6 αποτελεί τον επίλογο της διπλωματική, με μια σύνοψη των όσων μελετήσαμε και των συμπερασμάτων που προκύπτουν. Επίσης, αναφέρονται και μελλοντικές ιδέες προς έρευνα και υλοποίηση ως επέκταση της παρούσας εργασίας.

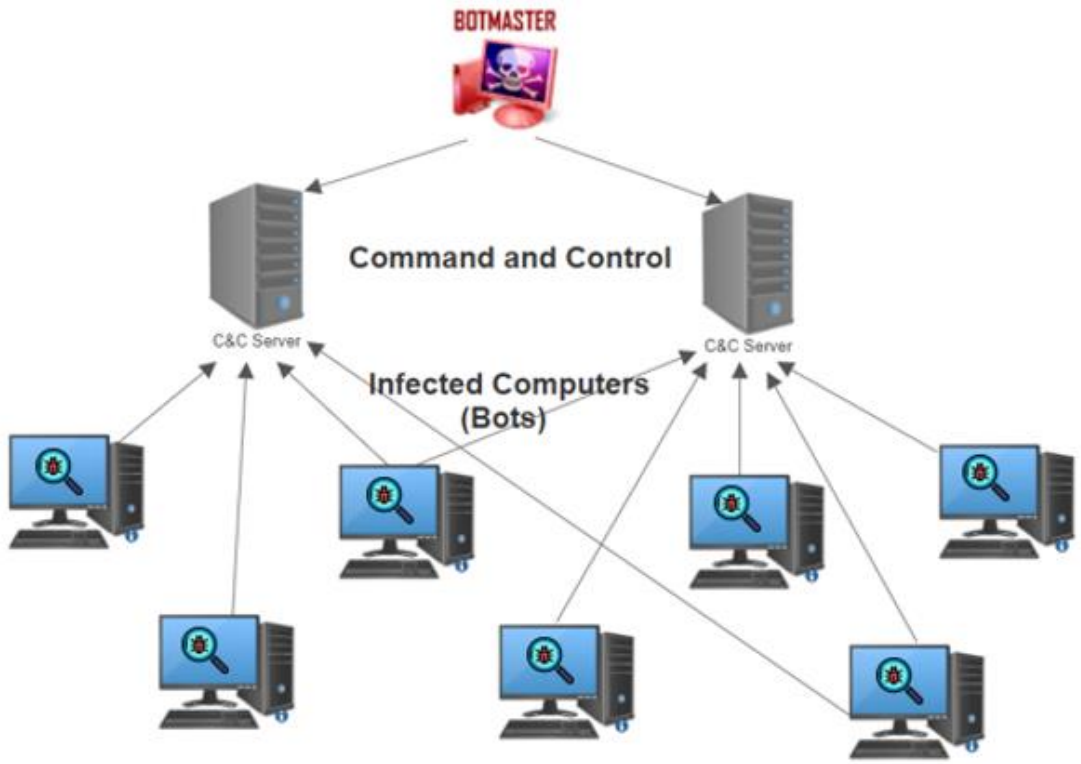
Κεφάλαιο 2 – Θεωρητικό Υπόβαθρο

2.1 Η δομή ενός botnet και οι κακόβουλες δραστηριότητές του

Ως botnet (συντομογραφία του όρου "robot-network") ορίζεται ένα δίκτυο υπολογιστών που έχουν μολυνθεί από κακόβουλο λογισμικό (malware) και βρίσκονται υπό τον έλεγχο ενός κακόβουλου επιτιθέμενου διαχειριστή γνωστού ως botmaster. Κάθε μηχάνημα υπό τον έλεγχο του ονομάζεται bot. Από έναν κεντρικό κόμβο, και πιο συγκεκριμένα τον Command and Control (C&C) server ο επιτιθέμενος μπορεί να δώσει εντολή, με κατανεμημένο τρόπο, σε κάθε υπολογιστή του botnet του ώστε να πραγματοποιήσει ταυτόχρονα μια συντονισμένη κυβερνοεπίθεση [4]. Η κλίμακα ενός botnet, το οποίο μπορεί να αποτελείται από εκατομμύρια bots, επιτρέπει στον εισβολέα να εκτελεί συντονισμένες ενέργειες μεγάλης κλίμακας που προηγουμένως ήταν αδύνατες απλά μολύνοντας έναν υπολογιστή με malware. Δεδομένου ότι τα bots ελέγχονται από τον botmaster και εκτελούν εντολές μέσω των C&C server, μπορούν να λαμβάνουν ενημερώσεις και να αλλάζουν την συμπεριφορά τους, κάνοντας την δυναμική αυτή δομή πολύ-λειτουργική για τον επιτιθέμενο [5].

Χαρακτηριστικά παραδείγματα δράσης botnet είναι:

- Υποκλοπή προσωπικών δεδομένων, όπως τραπεζικοί κωδικοί
- Αποστολή ανεπιθύμητης αλληλογραφίας (spam), δηλαδή την αποστολή μαζικών e-mail από κάθε bot που περιέχουν μολυσμένα αρχεία, με σκοπό κυρίως την εξάπλωση του botnet
- Κατανεμημένες επιθέσεις άρνησης υπηρεσιών (DDoS Attacks), στην πράξη το botnet αξιοποιεί το μέγεθος του για να υπερφορτώσει ένα δίκτυο ή έναν διακομιστή με αιτήματα, καθιστώντας το απρόσιτο στους χρήστες του. Αυτές οι επιθέσεις στοχεύουν οργανισμούς για προσωπικά ή πολιτικά κίνητρα, είτε για εκβιασμό πληρωμής με αντάλλαγμα την διακοπή της επίθεσης
- Ransomware – κρυπτογράφηση των αρχείων και των δεδομένων ενός υπολογιστή και απειλή δημοσιοποίησης τους ή διακοπή πρόσβασης σε αυτά, με σκοπό την απαίτηση αντιτίμου
- Εκμετάλλευση της επεξεργαστικής ισχύος για cryptocurrency mining, προς όφελος του επιτιθέμενου



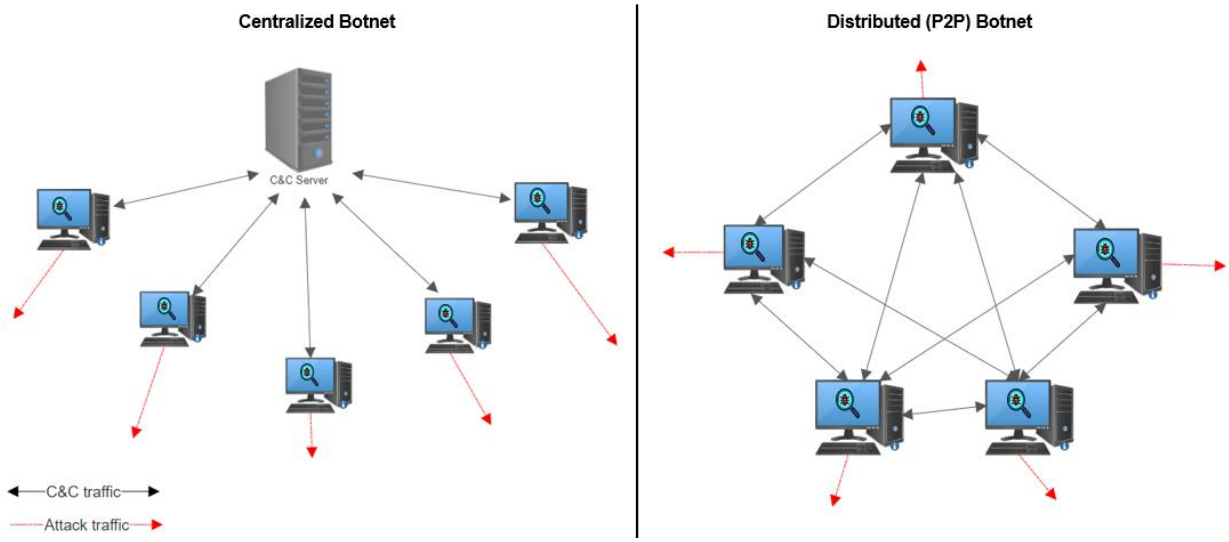
Εικόνα 1: Η δομή ενός botnet

2.2 DGA-based botnets

Τα περισσότερα botnets σήμερα βασίζονται σε έναν ή περισσότερους κεντρικούς διακομιστές εντολών και ελέγχου (C&C server), όπως φαίνεται και στην παραπάνω εικόνα, όπου οι μολυσμένοι υπολογιστές εκτελούν DNS queries με βάση ένα προκαθορισμένο domain name το οποίο κάνει resolve την IP address του C&C, από τον οποίον θα λαμβάνονται οι εντολές. Τέτοιου είδους κεντρικές δομές υποφέρουν από το πρόβλημα του μεμονωμένου σημείου αποτυχίας (single point of failure) γιατί αν εντοπιστεί και μπλοκαριστεί ο C&C server, τότε ο botmaster χάνει τον έλεγχο σε ολόκληρο το botnet. Για να ξεπεραστεί αυτός ο περιορισμός, οι επιτιθέμενοι έχουν δοκιμάσει P2P (peer-to-peer) δομές στο botnet τους, οπού τα bots όπως φαίνεται και στην εικόνα 2 τελούν και χρέη C&C server. Τέτοιες υλοποιήσεις, όπως το Nugache [6], το Waledac [7] και το Alureon (γνωστό ως TLD4) [8], έχουν πιο στιβαρή δομή, η οποία είναι δύσκολο να ανιχνευθεί και να εξαρθρωθεί. Ωστόσο, ευτυχώς είναι ιδιαίτερα περίπλοκα και δύσκολα στην συντήρηση και συνεπώς δεν χρησιμοποιούνται συχνά.

Για να δυσκολέψουν την ανίχνευση οι επιτιθέμενοι έχουν αναπτύξει botnets τα οποία εντοπίζουν τον διακομιστή τους μέσω αυτόματων παραγόμενων ψευδοτυχαίων ονομάτων τομέων. Κάθε bot για να επικοινωνήσει με τον botmaster εκτελεί έναν αλγόριθμο δημιουργίας ονόματος τομέα (Domain Generation Algorithms - DGA), όπου δίνεται ένα τυχαίο seed ως είσοδος (το οποίο είναι κοινό στα bots και τους C&C),

και παράγεται μια λίστα με υποψήφια ονόματα τομέων για τον διακομιστή. Στη συνέχεια, το bot επιχειρεί να κάνει resolve αυτά τα domain names στέλνοντας ερωτήματα DNS έως ότου ένα από αυτά να επιλύσει την διεύθυνση IP του C&C Server, όπου για κάθε αποτυχημένη προσπάθεια παράγεται ένα NXDomain response. Η αξιοποίηση των Domain Generation Algorithms παρέχει ένα σημαντικό επίπεδο ευελιξίας διότι εκτελούνται περιοδικά σε όλους τους κόμβους του botnet, παρακάμπτοντας έτσι παραδοσιακά συστήματα ασφαλείας (π.χ. domain name blacklist), και ακολούθως τα bots εκτελούν κατόπιν DNS queries για να γίνει resolve η νέα διεύθυνση του C&C server που έχει παραχθεί από τον DGA. Χαρακτηριστικά τέτοια botnets είναι το Bobax [9], το Kraken [10] και το Torpig [11].



Εικόνα 2: Centralized vs P2P Botnets

2.3 Domain Generation Algorithms (DGA)

Η μελέτη των αλγορίθμων παραγωγής ονομάτων τομέα, δηλαδή οι Domain Generation Algorithms, βασίζεται κυρίως στην δουλειά των Plohmann et al. [12]. Ως αλγόριθμος ένας DGA χρησιμοποιείται για να δημιουργεί δυναμικά ένα μεγάλο αριθμό φαινομενικά τυχαίων ονομάτων τομέα και στην συνέχεια επιλέγει ένα υποσύνολο για να επιτρέψει την επικοινωνία με του C&C servers. Τα παραγόμενα domain names υπολογίζονται βάσει ενός δοσμένου seed, το οποίο μπορεί να αποτελείται από αριθμητικές σταθερές, την τρέχουσα ημερομηνία και ώρα ή ακόμη και τις τάσεις του Twitter. Το seed αποτελεί κοινό «μυστικό» μεταξύ των botmasters που ελέγχουν τον διακομιστή εντολών και ελέγχου και των bots για τον υπολογισμό κοινού συνόλου ονομάτων τομέα. Με την περιοδική εναλλαγή των χρησιμοποιούμενων ονομάτων καθίστανται αναποτελεσματικές οι παραδοσιακές στατικές μέθοδοι ανίχνευσης όπως το static domain blacklist, όπου δημιουργείται μια λίστα από domain names που χρησιμοποιήθηκαν για κακόβουλη δραστηριότητα. Το δοσμένο seed είναι μείζονος

σημασίας για την παραγωγή των ονομάτων σε κάθε εκτέλεση του DGA, καθώς είναι αυτό που «ορίζει» την τυχαιότητα της δημιουργίας των domain names, περιπλέκοντας έτσι την διαδικασία ανίχνευσης και εξάρθρωσης ενός κακόβουλου δικτύου. Επιπρόσθετα, η πιθανή εξάρτηση του seed από την ημερομηνία και ώρα δυσχεραίνει ακόμη περισσότερο την δουλειά των δικτύων ανίχνευσης κακόβουλου λογισμικού καθώς διαφορετικά domain names παρατηρούνται σε διαφορετικά χρονικά σημεία. Τέλος, ένα σημαντικό πλεονέκτημα που έχει αυτή η δομή είναι η χρήση ονομάτων τομέα βραχείας διάρκειας που γίνονται register, δηλαδή καταχωρούνται στο δίκτυο λίγο πριν γίνουν έγκυρα, οπότε έτσι αποφεύγουν την ανίχνευση από reputation systems. Συνεπώς, όπως γίνεται αντιληπτό η χρήση DGA δημιουργεί μια εξαιρετικά ασύμμετρη κατάσταση μεταξύ επιτιθέμενων (botmasters) και αμυνόμενων (ερευνητές ασφάλειας και νομικές αρχές). Για τους botmasters αρκεί η πρόσβαση σ' ένα όνομα τομέα για τον έλεγχο ή την μετεγκατάσταση των bots τους, ενώ η αντίθετη πλευρά πρέπει να ελέγξει όλα τα domain names για να εξασφαλίσουν ότι εξάρθρωσαν το botnet. Αν σε αυτό, συνυπολογίσουμε ότι οι κακόβουλοι χρήστες έχουν πρόσβαση σε πολλά και κυρίως φθηνά top-level domains (τελευταίο κομμάτι ενός ονόματος τομέα στο διαδίκτυο) για να διαλέξουν, είναι εύκολο να δημιουργήσουν μια παγκόσμιας εμβέλειας απειλή οδηγώντας τις νομικές αρχές στην ανάγκη για καλύτερο συντονισμό και συνεργασία [13]. Στην πράξη όμως, οι οργανισμοί ασφαλείας δυσανασχετούν στο να συνεργαστούν και να ανταλλάξουν τα δεδομένα τους, είτε λόγω αυστηρών πρωτοκόλλων ιδιωτικότητας είτε λόγω εμπορικού ανταγωνισμού και συμφερόντων. Τα βασικά σχήματα παραγωγής των DGA είναι τα παρακάτω:

1. **Arithmetic-based / Alphanumeric DGA:** Υπολογισμός μια ακολουθίας αλφαριθμητικών. Τα DGA που βασίζονται σε αυτό το σχήμα παραγωγής υπολογίζουν μια ακολουθία τιμών που είτε έχουν άμεση αναπαράσταση ASCII είτε έχει χρησιμοποιηθεί κάποιο offset για την δημιουργία του αλφάβητου του DGA. Αυτή η μέθοδος είναι η πιο κοινή (π.χ. DirCrypt).

2. **Hash-based DGA:** Υπολογισμός της δεκαεξαδικής αναπαράστασης ενός hash code, που συνδυάζεται με τους αλγορίθμους κρυπτογράφησης MD5 και SHA256 (π.χ. MoneroDownloader, Bamital).

3. **Wordlist-based DGA:** Συνένωση δύο ή παραπάνω λέξεων, προερχόμενες από διαφορετικές λίστες λέξεων. Αυτή η μέθοδος παραγωγής οδηγεί σε domain names που είναι πιο συγκαλυμμένα καθώς ομοιάζουν σ' ένα βαθμό με έγκυρα. Οι λίστες από τις οποίες αντλούνται οι λέξεις είτε βρίσκονται ενσωματωμένες στο κακόβουλο λογισμικό (malware) ή λαμβάνονται από κάποια δημόσια προσβάσιμη πηγή (π.χ. Matsnu, Suppobox).

4. **Permutation-based DGA:** Μετάθεση των αλφαριθμητικών χαρακτήρων ενός αρχικού domain name (π.χ. Volatile Cedar).

Επιπρόσθετα, σύμφωνα με την έρευνα των Barabosch et al. [14], οι DGAs, μπορούν να διαχωριστούν σε δύο ακόμη κατηγορίες με βάση τις ιδιότητες του seed τους. Αυτές είναι:

1. **Time dependence – Εξάρτηση από τον χρόνο**. Αυτό σημαίνει ότι ενσωματώνεται μια πηγή χρόνου όπως η ώρα του συστήματος του παραβιασμένου host ή η ημερομηνία σε μια απόκριση HTTP.

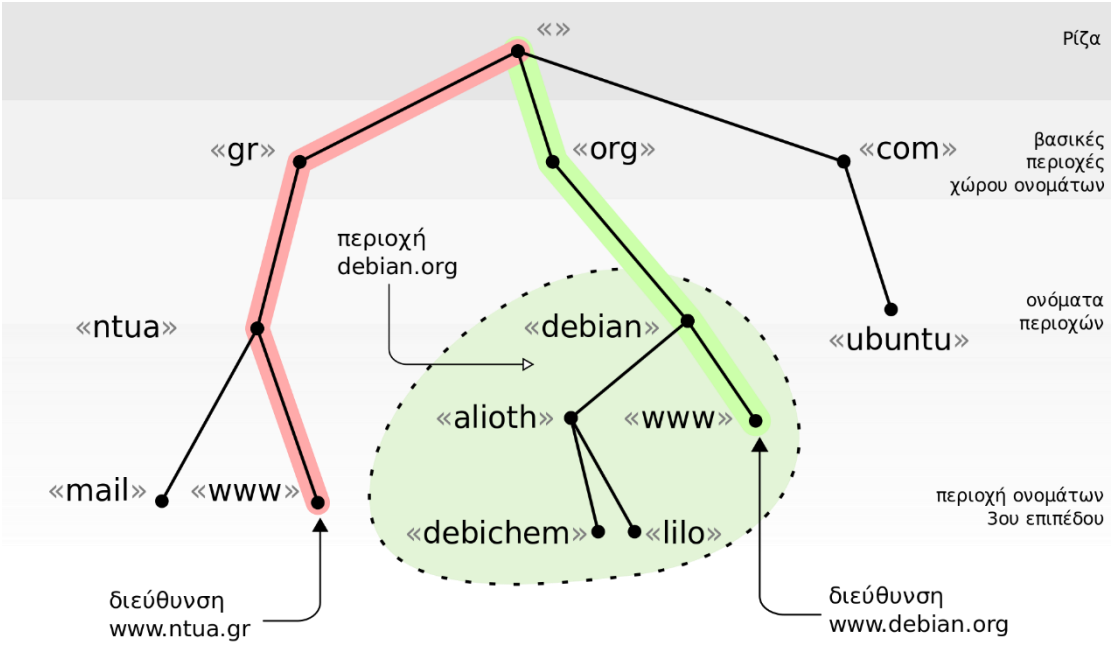
2. **Determinism – Αιτιότητα**. Αφορά την παρατηρησιμότητα και την διαθεσιμότητα των τιμών που δίνονται ως seed [12]. Για την πλειοψηφία των γνωστών DGA, όλες οι απαιτούμενες παράμετροι για την εκτέλεση του DGA είναι γνωστές σε βαθμό που μπορούν να υπολογιστούν όλα τα πιθανά domain names. Ωστόσο, ελάχιστες οικογένειες DGA χρησιμοποιούν απρόβλεπτα αλλά δημόσια προσβάσιμα δεδομένα ως seed [16], π.χ. ν-οστή πιο δημοφιλής είδηση στο Twitter [11].

2.4 Σύστημα Ονοματοδοσίας Τομέων – Domain Name System (DNS)

Όπως έχει γίνει αντιληπτό, οι υλοποιήσεις botnet που αξιοποιούν τους Domain Generation Algorithms χρησιμοποιούν κατά κόρον το πρωτόκολλο DNS (Domain Name System) για την δημιουργία ενός διαύλου επικοινωνίας μεταξύ του C&C server και των bots. Σε αυτή την ενότητα θα μελετήσουμε λίγο τον τρόπο λειτουργίας του DNS. Το σύστημα ονοματοδοσίας τομέων είναι ο τηλεφωνικός κατάλογος του διαδικτύου, καθώς όπως αυτός αντιστοιχίζει ονόματα ανθρώπων και τηλεφωνικούς αριθμούς, έτσι και το DNS αντιστοιχίζει ονόματα υπολογιστών (hostnames) σε διευθύνσεις IP (IP addresses) ή και άλλους δικτυακούς πόρους, και αντίστροφα [16]. Η καθολική αποδοχή και χρήση της υπηρεσίας το καθιστά ζωτικής σημασίας για την απρόσκοπτη λειτουργία του διαδικτύου. Ταυτόχρονα, γίνεται και στόχος επιθέσεων λόγω της πρόσβασης και της γνώσης που προσφέρει, καθώς η κίνηση DNS θεωρείται απαραίτητη και συνεπώς δεν φιλτράρεται ή μπλοκάρεται αυστηρά εξ ορισμού. Γίνεται ξεκάθαρο λοιπόν, ότι οποιαδήποτε δυσλειτουργία του συστήματος είναι πιθανό να παρεμποδίσει την πρόσβαση μεγάλου αριθμού χρηστών στο διαδίκτυο, κάτι το οποίο μπορεί να επιφέρει σημαντικές ζημιές τόσο σε οικονομικό επίπεδο όσο και στην αξιοπιστία των δικτύων.

Περνώντας στον τρόπο λειτουργίας του, οι διακομιστές DNS μετατρέπουν τα ονόματα τομέα σε διευθύνσεις IP, τις οποίες οι υπολογιστές μπορούν να καταλάβουν και να χρησιμοποιήσουν. Δηλαδή, μεταφράζουν την είσοδο ενός χρήστη σ' ένα πρόγραμμα περιήγησης σε κάτι που μπορεί να χρησιμοποιήσει το μηχάνημα για να βρει μια ιστοσελίδα. Αυτή η διαδικασία μετάφρασης και αναζήτησης ονομάζεται DNS Resolution (επίλυση) [17]. Προτού όμως αναλύσουμε τα βήματα της επίλυσης που συμβαίνει εσωτερικά του συστήματος, ας μελετήσουμε την δομή των ονομάτων τομέα.

Τα ονόματα αυτά αποτελούνται από πολλά μέρη που ονομάζονται ετικέτες (labels) και η ιεραρχία διαβάζεται από τα δεξιά προς τ' αριστερά με κάθε τμήμα να δηλώνει μια υποδιαίρεση. Το Top-Level-Domain (TLD) εμφανίζεται μετά την τελευταία «.» στο όνομα. Παραδείγματα TLD αποτελούν τα .com, .org και .edu κ.ά. Ορισμένα μπορεί να υποδηλώνουν έναν κωδικό χώρας ή μια γεωγραφική τοποθεσία όπως .us για τις Ηνωμένες Πολιτείες της Αμερικής ή .ca για τον Καναδά. Κάθε ετικέτα αριστερά από το TLD υποδηλώνει μια υποπεριοχή (subdomain) του τομέα στα δεξιά. Για παράδειγμα, στην διεύθυνση www.ntua.gr, το «ntua» είναι μια υποπεριοχή του .gr.



Εικόνα 3: Η ιεραρχία του DNS

Για την ολοκλήρωση του πολυπόθητου DNS Resolution συμμετέχουν δύο τύποι διακομιστών, οι recursive (ο πρώτος από τους παρακάτω) και οι authoritative (οι 2, 3 και 4 από τους παρακάτω). Η ακόλουθη λίστα περιγράφει του διακομιστές που χρησιμοποιούνται [18]:

1. **Recursive Server - DNS Recursor / Resolver (Αναδρομικός DNS διακομιστής):** Μπορεί να θεωρηθεί ως ένας βιβλιοθηκάριος που του ζητείται να πάει να βρει ένα συγκεκριμένο βιβλίο κάπου σε μια βιβλιοθήκη. Στην πράξη δηλαδή, είναι ο διακομιστής που λαμβάνει ερωτήματα από υπολογιστές μέσω περιηγητών ιστού.

2. **Authoritative Server - Root nameserver (Διακομιστής ονομάτων ρίζας):** Αποτελεί το πρώτο βήμα για την μετάφραση (επίλυση) ονομάτων υπολογιστών (hostnames) σε διευθύνσεις IP. Αναλογικά, θεωρείται το ευρετήριο σε μια βιβλιοθήκη που οδηγεί σε συγκεκριμένες τοποθεσίες

3. **Authoritative Server - TLD nameserver (Διακομιστής ονομάτων TLD):** Αναλογικά με μια βιβλιοθήκη αποτελεί το συγκεκριμένο ράφι που περιέχει τα βιβλία που ψάχνουμε. Φιλοξενεί το τελευταίο τμήμα ενός ονόματος, όπως τα .org, .com που συζητήσαμε παραπάνω

4. **Authoritative nameserver:** Μπορεί να θεωρηθεί ως το λεξικό στο επιλεγμένο ράφι βιβλίων, όπου ένα κλειδί μπορεί να οδηγήσει στον τίτλο του βιβλίου. Πρακτικά, αυτοί οι διακομιστές αποτελούν το τελευταίο βήμα ενός DNS query και αν έχουν πρόσβαση στην εγγραφή που ζητήθηκε, τότε θα επιστρέψει την διεύθυνση IP για το ζητούμενο hostname πίσω στον DNS Recursor που έκανε το αρχικό αίτημα. Ωστόσο, πρέπει να σημειωθεί ότι μπορεί να περιλαμβάνονται και πολλαπλά βήματα των παραπάνω (root, TLD) διακομιστών.

Έχοντας συμπληρώσει πλέον τα κομμάτια του puzzle, μπορούμε να δούμε την διαδικασία του DNS Resolution:

1. Ο χρήστης εισάγει μια διεύθυνση ιστού ή ένα όνομα τομέα σ' ένα πρόγραμμα περιήγησης

2. Το πρόγραμμα περιήγησης στέλνει ένα μήνυμα (recursive DNS query) στο δίκτυο για να βρει σε ποια IP διεύθυνση ή σε ποια διεύθυνση δικτύου αντιστοιχεί ο τομέας.

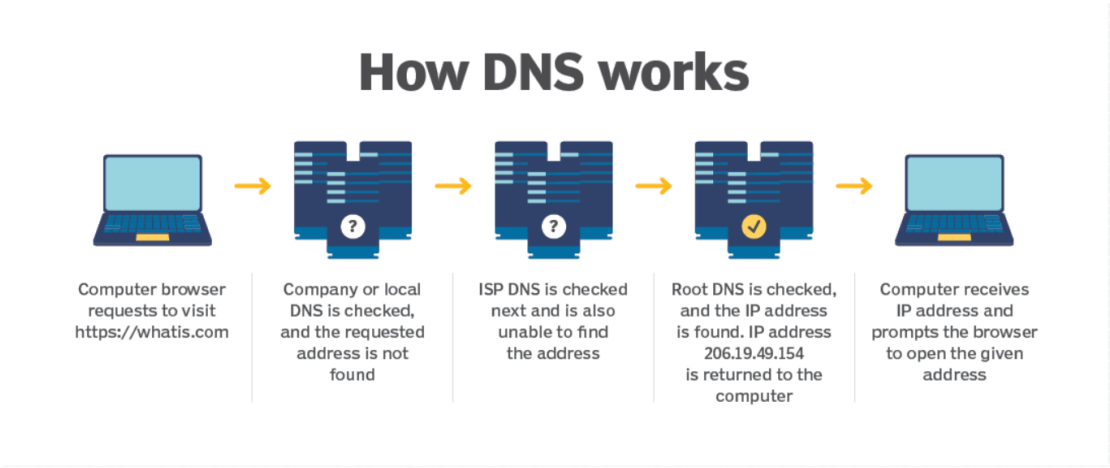
3. Το ερώτημα προωθείται στον DNS Recursor / Resolver και συνήθως χειρίζεται από τον πάροχο υπηρεσιών διαδικτύου (ISP). Αν ο DNS Recursor έχει τη διεύθυνση, θα την επιστρέψει στον χρήστη και αυτός θα λάβει πρόσβαση στην ιστοσελίδα.

4. Αν δεν μπορεί να απαντήσει, τότε ρωτάει τους άλλους διακομιστές με την ακόλουθη σειρά: DNS root nameserver, top-level-domain nameservers και τέλος authoritative name-servers

5. Στην περίπτωση του recursive resolution, οι τρεις διακομιστές που έλαβαν το ερώτημα συνεργάζονται και αναζητούν έως ότου ανακτήσουν την εγγραφή DNS που περιέχει την διεύθυνση IP που ζητήθηκε. Αυτή η πληροφορία αποστέλλεται πίσω στον DNS Recursor και φορτώνεται η ιστοσελίδα που αναζητά ο χρήστης. Οι DNS root nameservers και οι TLD servers κυρίως ανακατευθύνουν ερωτήματα και σπανίως παρέχουν οι ίδιοι την απάντηση.

6. Ο DNS Recursor αποθηκεύει προσωρινά την εγγραφή A DNS για το όνομα τομέα, η οποία περιέχει τη διεύθυνση IP. Την επόμενη φορά που θα λάβει ένα αίτημα για αυτό το domain name, μπορεί να απαντήσει απευθείας στον χρήστη χωρίς να ρωτήσει τους άλλους διακομιστές.

7. Αν το ερώτημα φτάσει στον authoritative nameserver και δεν μπορεί να βρει την απαραίτητη πληροφορία, τότε επιστρέφεται μήνυμα σφάλματος



Εικόνα 4: DNS Resolution

Παρά την σημαντική και εύρωστη λειτουργία του συστήματος ονοματοδοσίας τομέων, έχουν ανακαλυφθεί μερικά τρωτά σημεία του με την πάροδο του χρόνου όπως το DNS cache poisoning [18], τα οποία ξεφεύγουν από τα πλαίσια ανάλυσης αυτής της εργασίας. Επιστρέφοντας στο πρόβλημα μας, και προτού περάσουμε στους μηχανισμούς εντοπισμού κακόβουλων ονομάτων παραγωγής από Domain Generation Algorithms για παραπλάνηση του σε επιθέσεις από botnets, υπενθυμίζουμε ότι αφότου παραχθεί το σύνολο των DGA ονομάτων, ακολουθεί σειριακή αποστολή DNS ερωτημάτων από τα bots, μέχρι κάποιον domain name να γίνει resolver και να εντοπιστεί ο C&C server.

2.5 Μηχανισμοί ανίχνευσης ονομάτων από DGA κι η ανάγκη της επεξηγησιμότητας (explainability)

Όπως είδαμε παραπάνω, οι DGA αξιοποιούνται από τους κακόβουλους για την παραγωγή σύντομης διάρκειας ζωής domain names που εκχωρούνται στον C&C server με στόχο να αποκρύψουν την ταυτότητα του. Με αυτό τον τρόπο παρακάμπτονται παραδοσιακά συστήματα ασφαλείας, τα οποία χρησιμοποιούν στατικές μεθόδους για την ανίχνευση κακόβουλης δικτυακής κίνησης σχετικά με το domain name του C&C server, τον περιορισμό και την εξουδετέρωση του botnet. Πρωταρχική μέθοδος αποτέλεσε το domain name blacklist, σύμφωνα με το οποίο όταν εντοπιζόταν ένα κακόβουλο domain name, αυτό τοποθετούνταν σε μια μαύρη λίστα και ελεγχόταν κάθε φορά μεταγενέστερα. Είναι σχετικά απλό να αντιληφθούμε ότι αυτή η προσέγγιση δεν αποδίδει καρπούς στην περίπτωση της εφαρμογής των DGA, διότι ακόμη κι αν εντοπιστεί ένα τέτοιο όνομα, θα έχει αντικατασταθεί μέχρι να γίνει η εξουδετέρωση.

Έτσι, αντιλαμβανόμαστε καλύτερα αυτό που αναφέραμε νωρίτερα ως ασύμμετρη απειλή [13], καθώς οι κακόβουλοι χρήστες χρειάζονται μόλις ένα domain name για να ενεργοποιήσουν το botnet τους, ενώ οι ερευνητές ασφαλείας πρέπει να ανιχνεύσουν το σύνολο των παραγόμενων ονομάτων για να μπορέσουν να εξαλείψουν την απειλή μιας ύποπτης διαδικτυακής κίνησης. Μια δεύτερη προσέγγιση που δοκιμάστηκε ήταν αυτή του reverse engineering μελετώντας την δομή και τον τρόπο λειτουργίας των αλγορίθμων παραγωγής ονομάτων. Ωστόσο, αυτό αποτελεί μια πολύ χρονοβόρα διαδικασία και συνεπώς μη αποδοτική και μη κλιμακώσιμη.

Σε αυτό το σημείο, εισήλθε η ιδέα της αξιοποίησης στατιστικών χαρακτηριστικών των ονομάτων, όπως το πλήθος των ψηφίων, η μέγιστη ακολουθία συμφώνων κ.ά., με στόχο την κατηγοριοποίηση των domain names σε δύο κλάσεις, τα έγκυρα και τα αλγοριθμικά παραγόμενα. Σε τέτοιου είδους προβλήματα ανίχνευσης ανωμαλιών παραδοσιακά τα μοντέλα μηχανικής μάθησης παρουσίαζαν αξιόλογα αποτελέσματα. Όπως θα δούμε και στα πλαίσια των πειραμάτων αυτής της διπλωματικής, αρχικά δοκιμάστηκαν μεθοδολογίες με την χρήση Δέντρων Αποφάσεων, τα οποία ωστόσο όσο μεγαλώνει ο όγκος των δεδομένων και αυξάνεται η τυχαιότητα δεν είναι ερμηνεύσιμα και κατανοητά, παρότι επιτυγχάνουν καλή ακρίβεια. Με την ανάγκη, λοιπόν, για την ανάπτυξη αποτελεσματικότερων μεθοδολογιών όσον αφορά την ακρίβεια στραφήκαμε σε μοντέλα βαθιάς μηχανικής μάθησης αξιοποιώντας τα στατιστικά χαρακτηριστικά των domain names. Σε γενικές γραμμές τα μοντέλα βαθιάς μηχανικής μάθησης παρουσιάζουν μεγαλύτερη ακρίβεια σε σχέση με τα δέντρα, ωστόσο στην προσπάθεια αυτή γίνονται περίπλοκα, πιο ακριβά και μη επεξηγήσιμα. Επιπρόσθετα, η βαθιά μηχανική μάθηση έδωσε την δυνατότητα για να γίνει το επόμενο βήμα ώστε να αξιοποιηθεί η ακολουθία των αλφαριθμητικών χαρακτήρων ενός domain name οδηγώντας όπως θα δούμε και στα πειράματα μας σε ακόμη υψηλότερη ακρίβεια και περισσότερες δυνατότητες επεξηγησιμότητας, καθώς μελετάται το ίδιο το domain name ως έχει, χωρίς ενδιάμεση εξαγωγή χαρακτηριστικών.

Με βάση τα παραπάνω, προκύπτει η ανάγκη για την πολυπόθητη επεξηγησιμότητα (explainability) για την μελέτη μεθόδων και μηχανισμών για ανίχνευση DGA. Η πρώτη πλευρά που επιζητά περαιτέρω κατανόηση στην λήψη αποφάσεων ενός μοντέλου είναι οι ίδιοι οι χειριστές τους δικτύου (network operators), οι οποίοι καλούνται να απαντούν στο ερώτημα αφενός αν ένα domain name αποτελεί προϊόν κακόβουλης δικτυακής κίνησης, αλλά και γιατί ανιχνεύθηκε ως τέτοιο, ισχυροποιώντας έτσι τις γνώσεις τους για την ανίχνευση τέτοιων περιπτώσεων. Η δεύτερη πλευρά που χρειάζεται την δυνατότητα του explainability είναι οι προγραμματιστές (developers) του εκάστοτε συστήματος ασφαλείας, ώστε να μπορούν συνεργατικά με τους network operators να ελέγχουν το μοντέλο και να κάνουν το απαραίτητο debugging. Οι 2 παραπάνω πλευρές (network operators και developers) μπορεί και να ταυτίζονται κάποιες φορές. Η τρίτη και τελευταία πλευρά είναι οι νομικές οντότητες (Legal Entities), όπου οι αποφάσεις τους οφείλουν να συμμορφώνονται με τον γενικό κανονισμό για την προστασία δεδομένων, για την περίπτωση χρήσης αλγορίθμων σε κρίσιμες υποδομές.

2.6 eXplainability Artificial Intelligence (XAI)

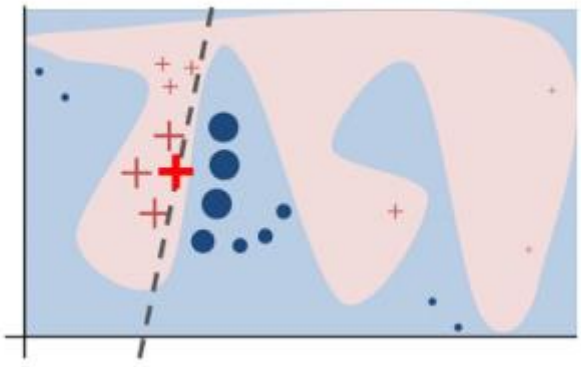
Η ερμηνεία πολύπλοκων μοντέλων μας βοηθά να κατανοήσουμε πως και γιατί ένα μοντέλο λαμβάνει μια απόφαση και ποια χαρακτηριστικά ήταν σημαντικά για να καταλήξουμε σε αυτό το συμπέρασμα. Αυτό βοηθά στο να ξεπεράσουμε ηθικές ανησυχίες και ζητήματα εμπιστοσύνης σχετικά με την χρήση μηχανικής και βαθιάς μάθησης στη λήψη αποφάσεων. Οι αλγόριθμοι που θα σημειώσουμε σε αυτή την ενότητα είναι post-hoc, δηλαδή αναλύουν τα αποτελέσματα μετά το πέρας της εκπαίδευσης του μοντέλου και model-agnostic, που σημαίνει ότι η μεθοδολογία μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο. Εφόσον, εκπαιδευτεί το οποιοδήποτε επεξηγήσιμο μοντέλο μπορεί να αντιμετωπιστεί ως black box για την λειτουργία του. Συνεπώς, αποκτάμε ευελιξία διότι δεν χρειάζεται να αναπτύξουμε διαφορετικό πλαίσιο αξιολόγησης για κάθε τύπο μοντέλου, καθώς μπορούμε να συγκρίνουμε πολλά μοντέλα χρησιμοποιώντας τις ίδιες μετρήσεις και να έχουμε έτσι αξιόπιστη σύγκριση τους.

Η προσέγγιση ερμηνείας του μοντέλου χωρίζεται σε δύο κατηγορίες, την τοπική επεξηγησιμότητα (local explainability) και την καθολική επεξηγησιμότητα (global explainability). Η πρώτη εξετάζει μια είσοδο και προσπαθεί να βρει και να εξηγήσει γιατί το μοντέλο λαμβάνει μια συγκεκριμένη απόφαση, ενώ η δεύτερη μας δίνει την δυνατότητα να κατανοήσουμε το μοντέλου στην συνολική του δομή [20].

Ο πρώτος αλγόριθμος που θα αναλύσουμε είναι ο Permutation Feature Importance (PFI) [21]. Αυτός μετρά την μείωση της απόδοσης του μοντέλου (π.χ. Root-Mean-Square-Error - RMSE), αφού ανακατέψουμε τις τιμές ενός χαρακτηριστικού και επομένως εξηγεί ποια χαρακτηριστικά οδηγούν λανθασμένα την απόδοση του μοντέλου [22]. Με απλούστερα λόγια, ένα χαρακτηριστικό δεν είναι και τόσο σημαντικό εάν από το shuffling των τιμών του αυξάνεται το σφάλμα του μοντέλου, επειδή σε αυτή την περίπτωση βασίστηκε στο χαρακτηριστικό για την καλύτερη πρόβλεψη. Από την άλλη, ένα χαρακτηριστικό δεν είναι και τόσο σημαντικό εάν το shuffling των τιμών του αφήνει σχεδόν αμετάβλητο το σφάλμα του μοντέλου, διότι σε αυτή την περίπτωση το μοντέλο αγνοεί το χαρακτηριστικό για την πρόβλεψη [23]. Ο PFI όπως γίνεται αντιληπτό συνδέεται άμεσα με το σφάλμα του μοντέλου, κάτι το οποίο δεν είναι πάντα αυτό που ψάχνουμε. Ακόμη, δεν ενδείκνυται για μοντέλα με συσχετισμένα χαρακτηριστικά (correlated features), καθώς μετά το ανακάτεμα μπορεί να οδηγήσει σε παραπλανητικές ερμηνείες. Εκτός από την αδυναμία να επεξηγήσει μοντέλα με έντονα συσχετισμένα χαρακτηριστικά, είναι ακατάλληλος για την εξήγηση μοντέλων χρονοσειρών (time-series models).

Ο δεύτερος αλγόριθμος που θα παρουσιάσουμε είναι γνωστός ως LIME (Local Interpretable Model-agnostic Explanations) [24]. Ουσιαστικά, το LIME είναι ένας model-agnostic αλγόριθμος, που σημαίνει ότι μπορεί να εφαρμοστεί σε οποιαδήποτε μοντέλα με βάση την υπόθεση ότι κάθε τέτοιο πολύπλοκο μοντέλο μπορεί να αντιμετωπιστεί ως black box. Σε αντίθεση με τον PFI που εξηγεί ποια χαρακτηριστικά

καθοδηγούν την ακρίβεια του μοντέλου, το LIME (όπως και το SHAP που θα δούμε παρακάτω) προσπαθεί να εξηγήσει ποια χαρακτηριστικά συνεισέφεραν περισσότερο στις εκτιμήσεις και τις προβλέψεις του μοντέλου. Είναι σημαντικό να σημειωθεί σε αυτό το σημείο ότι το LIME είναι observation-specific. Αυτό σημαίνει ότι προσπαθεί να κατανοήσει τα χαρακτηριστικά που επηρεάζουν την πρόβλεψη γύρω από ένα συγκεκριμένο δειγματικό στοιχείο (local explainability). Η ιδέα που οδήγησε σε αυτή την προσέγγιση είναι ότι η καθολική επεξηγησιμότητα είναι δύσκολο να επιτευχθεί, συγκριτικά με την τοπική προσέγγιση ενός black-box μοντέλου.



Εικόνα 5: Decision Boundary of black-box model [24]

Στην πράξη το decision boundary (όριο απόφασης) ενός μοντέλου μπορεί να φαίνεται πολύ περίπλοκο, όπως διαγράφεται από το μπλε-ροζ φόντο στην εικόνα 5, το περιεχόμενο της οποίας θα αναλυθεί παρακάτω. Ωστόσο, μελέτες [25], [26] έχουν δείξει ότι αν εστιάσουμε σ΄ ένα συγκεκριμένο δειγματικό στοιχείο και την γειτονιά γύρω του, ανεξάρτητα από την πολυπλοκότητα του μοντέλου σε καθολικό επίπεδο, το decision boundary σε μια τοπική περιοχή μπορεί να είναι πολύ πιο απλό και στην πραγματικότητα ακόμη και γραμμικό. Εδώ, ας παρατηρήσουμε λίγο καλύτερα την εικόνα 5. Το μπλε-ροζ φόντο όπως είπαμε αποτελεί το όριο απόφασης ενός περίπλοκου μοντέλου. Ο έντονος κόκκινος σταυρός είναι το επιλεγμένο δειγματικό στοιχείο προς εξήγηση. Το LIME δημιουργεί νέα δειγματικά στοιχεία (απλά κόκκινα) γύρω από το επιλεγμένο instance και εφαρμόζει το μοντέλο σε αυτή την γειτονία λαμβάνοντας τις αντίστοιχες προβλέψεις. Τέλος, χρησιμοποιεί ένα γραμμικό μοντέλο για να εξηγήσει την πρόβλεψη του μοντέλου. Η μαύρη διακεκομμένη γραμμή της εικόνας υποδεικνύει την συμπεριφορά του μοντέλου στην συγκεκριμένη γειτονιά. Τα παραγόμενα δειγματικά στοιχεία ζυγίζονται με βάση την απόσταση τους από το επεξηγούμενο δειγματικό στοιχείο. Τα βάρη καθορίζονται από μια συνάρτηση πυρήνα (kernel function), η οποία παίρνει την ευκλείδεια απόσταση και το kernel width (απόσταση από το κέντρο των δεδομένων) ως είσοδο και εξάγει την βαθμολογία σπουδαιότητας (importance score) για κάθε παραγόμενο παράδειγμα. Η επεξηγησιμότητα που προσφέρει το LIME είναι σχετικά απλή. Ωστόσο, απαιτεί τον καθορισμό μιας γειτονιάς, που μπορεί να είναι πολύ χειραγωγήσιμο δεδομένο. Έτσι, οδηγείται το επεξηγήσιμο μοντέλο σε αστάθειες καθώς εξαρτάται από το πλήθος των παραγόμενων instances και το kernel width, το οποίο καθορίζει πόσο μεγάλη ή μικρή είναι η γειτονιά.

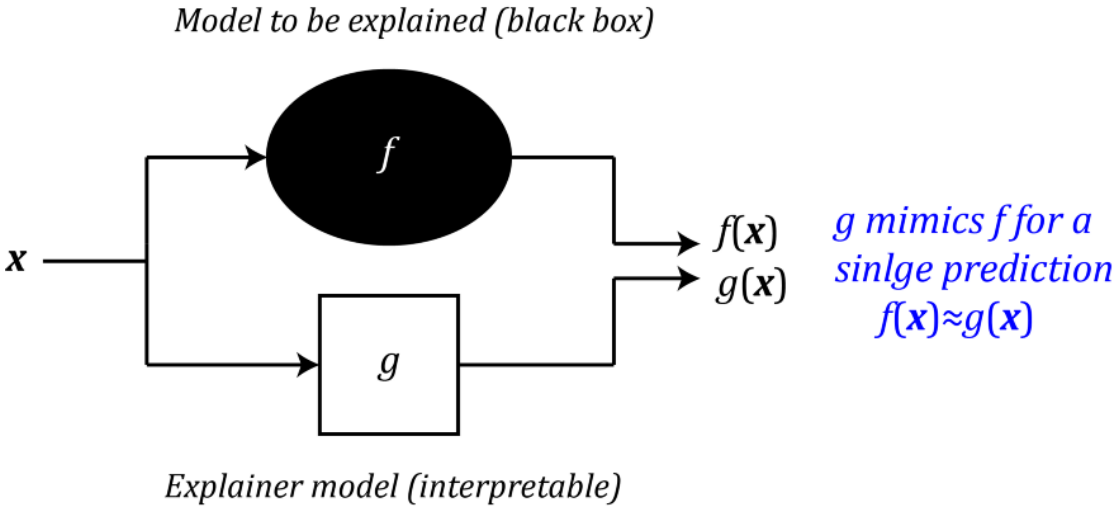
Η επιλογή γειτονιάς είναι καθοριστικής σημασίας ώστε να μην δημιουργηθεί μεροληψία προς την συνολική εξήγηση. Το LIME ως μέθοδος βασίζεται στο ότι τα δειγματικά στοιχεία θα έχουν γραμμική σχέση έστω σε τοπικό επίπεδο, αλλά δεν υπάρχει σχετική θεωρία που να το καθιστά σίγουρο αυτό, συνεπώς είναι απλά μια υπόθεση. Επομένως, ένα μη γραμμικό μοντέλο τόσο συνολικά όσο και τοπικά δεν μπορεί να επεξηγηθεί επαρκώς με την χρήση του LIME, παρά μόνο υπό τις προϋποθέσεις της ορθά επιλεγμένης γειτονιάς και της τοπικής γραμμικότητας.

Οι παραπάνω αλγόριθμοι που παρουσιάσαμε στερούνται θεωρητικής βάσης ως προς την αποτελεσματικότητα τους. Γι' αυτό στην παρούσα εργασία χρησιμοποιούμε το SHAP (SHapley Additive exPlanations) το οποίο έχει σημαντικό θεωρητικό υπόβαθρο στη θεωρία παιγνίων και μπορεί να χρησιμοποιηθεί για την επεξηγησιμότητα των μοντέλων. Όπως θα αναλύσουμε παρακάτω στην παρούσα εργασία, το SHAP επιτυγχάνει τόσο τοπική όσο και καθολική επεξηγησιμότητα, η οποία είναι και η σημαντικότερη για την ευρεία κατανόηση των αποφάσεων ενός μοντέλου. Το SHAP είναι ένας αλγόριθμος που δημοσιεύτηκε πρόσφατα (2017) [28] και αποτελεί την state-of-the-art τεχνική για επεξηγησιμότητα μοντέλων, καθώς στην πράξη μπορεί να επιτύχει αποτελεσματικά το reverse engineering σε τέτοιου είδους black box μοντέλα. Ως συνολικότερο πλαίσιο (framework) παρέχει υπολογιστικά εργαλεία για τον υπολογισμό των τιμών Shapley [29], μια έννοια της θεωρίας παιγνίων που χρονολογείται στην δεκαετία του 1950. Οι τιμές αυτά χρησιμοποιούνται στο game theory για να καθορίσουν πόσο συμβάλει ένας παίχτης στην επιτυχία σ' ένα παιχνίδι συνεργασίας. Ας το αναλύσουμε λίγο περαιτέρω αυτό, για να αντιληφθούμε πως ξεκινώντας από την θεωρία παιγνίων καταλήγουμε στην επεξηγησιμότητα μοντέλων μηχανικής μάθησης.

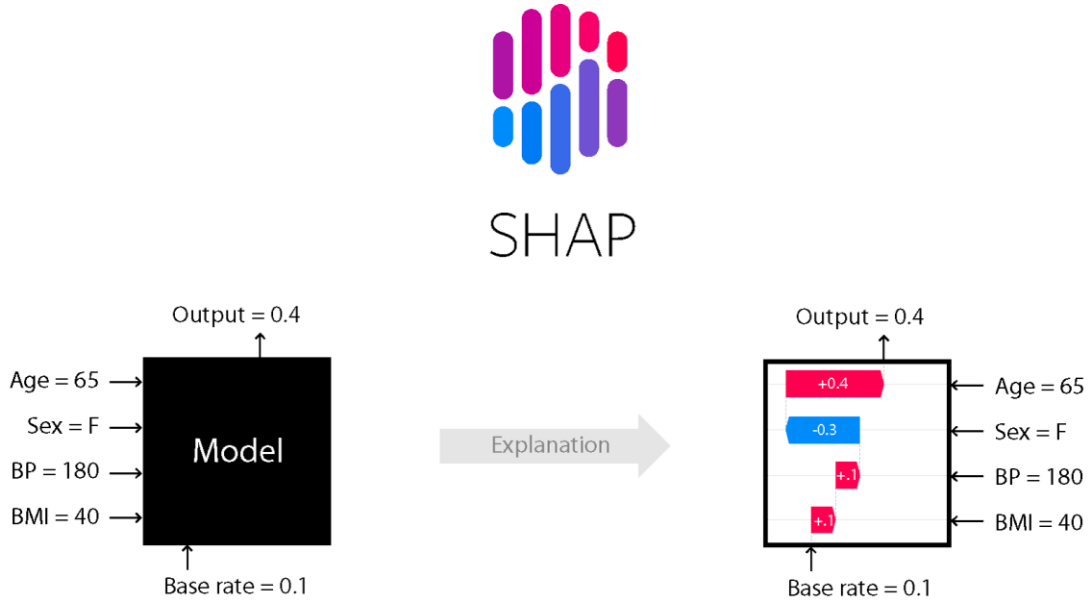
Το ερώτημα που οφείλουμε συνεπώς να απαντήσουμε αρχικά είναι: «Ποια είναι η σχέση μεταξύ της θεωρίας παιγνίων και της ερμηνείας της μηχανικής μάθησης;». Αντί, λοιπόν, να έχουμε ένα πρόβλημα μηχανικής μάθησης όπου εκπαιδεύουμε ένα μοντέλο με πολλαπλά χαρακτηριστικά για την δημιουργία προβλέψεων, ας φανταστούμε ένα παιχνίδι όπου κάθε χαρακτηριστικό («παίχτης») συνεργάζεται με τα υπόλοιπα για να επιτευχθεί μια πρόβλεψη («σκορ»). Η ερμηνεία της μηχανικής μάθησης απλοποιείται στο ερώτημα: «Πως και πόσο συνέβαλε κάθε παίχτης (χαρακτηριστικό) στο σκορ (πρόβλεψη);» [23]. Η συνεισφορά κάθε χαρακτηριστικού δίνεται από την τιμή Shapley, η οποία υπολογίζει πόσους πόντους θα κερδίζαμε ή θα χάναμε αν παίζαμε αυτό το παίγνιο με αυτό το χαρακτηριστικό. Ουσιαστικά, η τιμή Shapley εξηγεί την κατανομή των προβλέψεων με βάση το κάθε χαρακτηριστικό.

Προτού περάσουμε σ' ένα σύντομο παράδειγμα για τα Shapley Values ας δούμε δύο σημαντικές δυνατότητες του SHAP framework. Κάποια μοντέλα, όπως τα Δέντρα Αποφάσεων (Decision Trees) είναι εκ κατασκευής ερμηνεύσιμα. Ωστόσο, εδώ, η επεξηγησιμότητα αναφέρεται σε black-box μοντέλα με στόχο την κατανόηση από τον άνθρωπο των προβλέψεων αυτών των μοντέλων. Το SHAP είναι model-agnostic evaluation framework, δηλαδή μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο χρησιμοποιώντας τις ίδιες μετρικές. Πιο συγκεκριμένα, έχει πρόσβαση μόνο στα

δεδομένα εισόδου και στην πρόβλεψη του μοντέλου που επιθυμούμε να ερμηνεύσουμε. Ακόμη, είναι post-hoc μέθοδος, δηλαδή εφαρμόζεται μετά την εκπαίδευση του αρχικού μοντέλου. Στην πράξη το SHAP framework προσπαθεί να μιμηθεί όσο καλύτερα γίνεται το πραγματικό μοντέλο για κάθε instance. Συνεπώς, το ερμηνεύσιμο μοντέλο (g) παρέχει τις ίδιες προβλέψεις με το προς ερμηνεία μοντέλο για συγκεκριμένο δειγματικό στοιχείο. Για να μπορέσει λοιπόν το SHAP να επεξηγήσει οποιοδήποτε μοντέλο μηχανικής μάθησης κατ' αυτόν τον τρόπο αξιοποιεί τις τιμές Shapley.



Εικόνα 6:Από το black-box στο ερμηνεύσιμο μοντέλο (1)



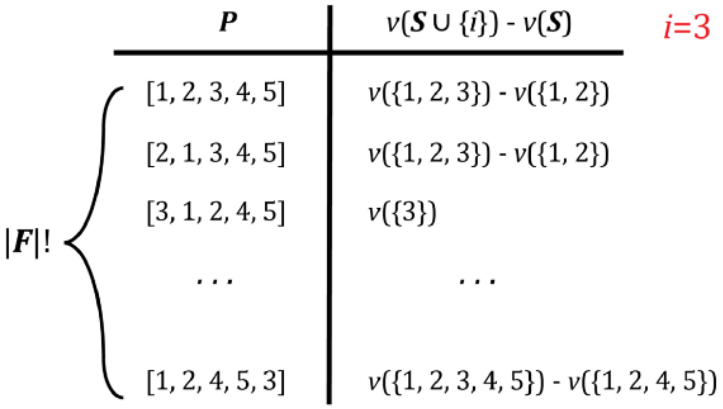
Εικόνα 7: Από το black-box στο ερμηνεύσιμο μοντέλο (2)

Περνώντας στον ορισμό της τιμής Shapley ϕ_i, είναι η μέση συνεισφορά ενός χαρακτηριστικού για την πρόβλεψη σε κάθε πιθανό συνδυασμό των χαρακτηριστικών.

Πιο συγκεκριμένα, ας υποθέσουμε ότι στο παίγνιο μας έχουμε Μ παίχτες, τότε ας ονομάσουμε ***F*** το σύνολο των παιχτών (χαρακτηριστικών), ***F*** = {1, 2, …, Μ}. Ως συμμαχία ***S*** (θα αναφέρεται ως coalition παρακάτω) ορίζουμε ένα υποσύνολο του ***F*** (***S*** ⊆ ***F***), συμπεριλαμβανομένου και του κενού συνόλου, όταν δηλαδή το coalition δεν περιέχει κανένα χαρακτηριστικό. Αν λοιπόν έχουμε 3 χαρακτηριστικά, τότε τα πιθανά coalitions είναι τα εξής: { ∅, {1}, {2}, {3}, {1, 2}, {1, 3}, {2, 3}, {1, 2, 3} }. Κατόπιν, ορίζουμε μια συνάρτηση *ν*, η οποία αντιστοιχεί κάθε coalition σ' ένα πραγματικό αριθμό και ονομάζεται χαρακτηριστική συνάρτηση (characteristic / target function). Επομένως, για κάθε coalition ***S***, ορίζεται η ποσότητα *ν(**S**)*, η οποία καθορίζει την αξία του. Το ερώτημα που τίθεται πλέον προς απάντηση είναι να ορίσουμε τον πιο δίκαιο τρόπο για να «μοιραστεί» η συνεισφορά των παιχτών. Στην πραγματικότητα για να έχουμε μια δίκαιη αξιολόγηση της συνεισφοράς του παίχτη / χαρακτηριστικού {i} θα πρέπει να σχηματίσουμε όλες τις δυνατές μεταθέσεις του ***F*** και να υπολογίσουμε την συνεισφορά του {i} σε καθεμιά από αυτές και κατόπιν να υπολογίσουμε τον μέσο όρο αυτών των συνεισφορών. Εξίσου σημαντικό να σημειωθεί σε αυτό το σημείο είναι ότι η χαρακτηριστική συνάρτηση *ν*, παίρνει ως όρισμα ένα coalition και όχι ένα συγκεκριμένο permutation. Το coalition είναι ένα σύνολο (set), επομένως η σειρά των στοιχείων σε αυτό δεν παίζει ρόλο, αντίθετα με το permutation που είναι μια διατεταγμένη συλλογή στοιχείων. Για παράδειγμα, αν είχαμε 5 παίχτες, μια πιθανή μετάθεση είναι η [3, 1, 2, 4, 5], όπου ο 3 είναι ο πρώτος και ο 5 είναι ο τελευταίος παίχτης. Για κάθε μετάθεση αυτών, η σειρά των στοιχείων μπορεί να αλλάξει την συμβολή τους στο συνολικό κέρδος, ωστόσο το συνολικό κέρδος, δηλαδή η αποτίμηση της χαρακτηριστικής συνάρτησης για την αξία του coalition θα είναι η ίδια, καθώς εξαρτάται μόνο από τα στοιχεία και όχι από τη σειρά τους. Επί παραδείγματι, *ν*(coalition of [5, 3, 2, 4]) = *ν*({2, 3, 4, 5}). Επομένως, για κάθε αντιμετάθεση ***P***, πρέπει πρώτα να υπολογίσουμε την συνεισφορά του coalition των παιχτών προτού προστεθεί το χαρακτηριστικό {i} που θέλουμε να ερμηνεύσουμε. Ας ονομάσουμε αυτή την συμμαχία ***S***. Κατόπιν, πρέπει να υπολογίσουμε την αξία της συμμαχίας που δημιουργείται με την προσθήκη του {i} στο ***S***, κι έτσι προκύπτει ένα νέο coalition το ***S***.U{i}. Τώρα, η συνεισφορά του παίχτη {i}, δηλαδή η Shapley τιμή του ϕ_i είναι:

$$\phi_i = \frac{1}{|F|!}\sum_{P}\big(v(S \cup \{i\}) - v(S)\big)$$

Το σύνολο των permutations του συνόλου των χαρακτηριστικών είναι |***F***|!, οπότε διαιρούμε το άθροισμα των συνεισφορών με αυτό, για να λάβουμε την μέση συνεισφορά του χαρακτηριστικού {i}.



$$\phi_i = \frac{1}{|\boldsymbol{F}|!}\sum_{\boldsymbol{P}}(v(\boldsymbol{S}\cup\{i\}) - v(\boldsymbol{S}))$$

Εικόνα 8: Υπολογισμός Shapley Value από permutations

Όπως φαίνεται από την παραπάνω εικόνα, κάποιες μεταθέσεις έχουν ακριβές την ίδια συνεισφορά καθώς τα coalitions ***S*** και ***S***.U{i} είναι τα ίδια, κατ' αντιστοιχία. Επομένως, αρκεί να υπολογίσουμε τις διακριτές τιμές των συνεισφορών και να τις πολλαπλασιάσουμε με το πλήθος εμφάνισης τους. Για να το επιτύχουμε αυτό, πρέπει να υπολογίσουμε πόσα permutations μπορούν να σχηματιστούν για κάθε coalition. Εξαιρώντας το στοιχείο {i} που θέλουμε να ερμηνεύσουμε, παραμένει το σύνολο ***F*** – {i}, και τώρα το ***S*** είναι ένα coalition αυτού, δηλαδή (***S*** ⊆ ***F*** – {i}). Η αξία κάθε μετάθεσης του ***S*** (πλήθος permutations: |***S***|!) είναι *ν(**S**)* και προσθέτοντας το χαρακτηριστικό {i} στο τέλος κάθε τέτοιας μετάθεσης του ***S***, λαμβάνουμε συνεισφορά *ν(**S***.U{i}*)*, καθώς όλα τα στοιχεία θα συμμετέχουν στην συμμαχία ***S***.U{i}. Παραμένουν ακόμη |***F***| - |***S***| - 1 χαρακτηριστικά στο σύνολο των παιχτών μας, τα οποία μπορούν να σχηματίσουν (|***F***| - |***S***| - 1)! μεταθέσεις. Ως αποτέλεσμα λοιπόν, έχουμε |***S***|! * (|***F***| - |***S***| - 1)! τρόπους για να σχηματίσουμε τις μεταθέσεις του αρχικού συνόλου ***F***, στο οποίο το χαρακτηριστικό {i} έρχεται μετά από μια μετάθεση του ***S*** και κατόπιν ακολουθούν οι υπόλοιποι παίχτες μετά το {i}. Εκτελώντας την παραπάνω διαδικασία για κάθε σύνολο ***F*** – {i}, παίρνουμε το άθροισμα των συνεισφορών του παίχτη {i} για όλες τις μεταθέσεις του αρχικού ***F***, κι έτσι έχουμε:

$$\sum_{S\subseteq F-\{i\}} |S|!\,(|F|-|S|-1)!\,\big(v(S\cup\{i\})-v(S)\big)$$

Τέλος, για να υπολογίσουμε την μέση συνεισφορά του {i}, πρέπει να διαιρέσουμε τον παραπάνω όρο με το σύνολο όλων των πιθανών μεταθέσεων του ***F*** και τελικά λαμβάνουμε:

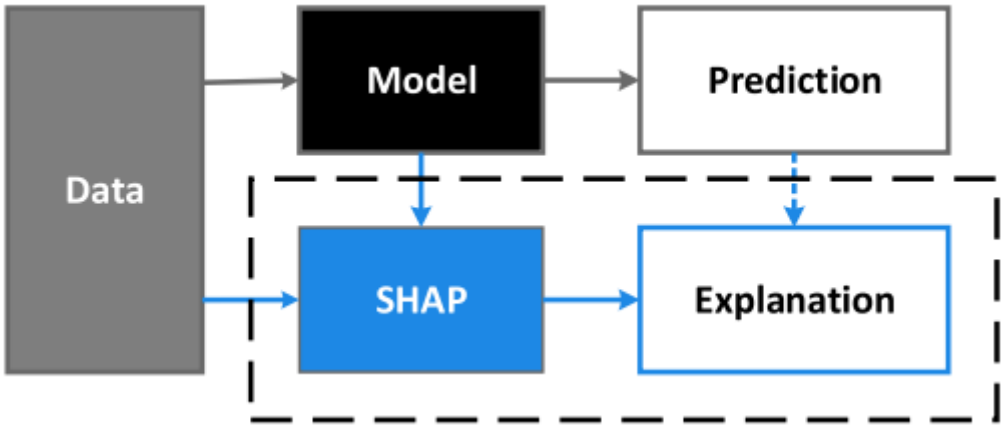
$$\phi_i = \sum_{S\subseteq F-\{i\}} \frac{|S|!\,(|F|-|S|-1)!}{|F|!}\big(v(S\cup\{i\})-v(S)\big)$$

Εδώ, λοιπόν, ϕ_i ονομάζεται η τιμή Shapley του χαρακτηριστικού {i}, η οποία είναι η μέση συνεισφορά του {i} σε όλες τις μεταθέσεις του $\boldsymbol{F}$. Αποδεικνύεται ότι αυτό είναι το μαθηματικό δίκαιο μερίδιο του παίχτη {i} στο συνολικό κέρδος όλων των παιχτών του $\boldsymbol{F}$.

Η τιμή Shapley είναι η μόνη μέθοδος που ικανοποιεί τρεις σημαντικές ιδιότητες:

1. **Symmetry:** Αν για 2 παίχτες i και j ισχύει $\nu(\boldsymbol{S}.\mathrm{U}\{i\}) = \nu(\boldsymbol{S}.\mathrm{U}\{j\})$ για κάθε coalition $\boldsymbol{S}$, το οποίο δεν περιέχει τα i και j, τότε $\phi_i = \phi_j$. Αυτό πρακτικά σημαίνει ότι αν 2 παίχτες προσθέτουν την ίδια συνεισφορά σε κάθε πιθανή συμμαχία, τότε θα έχουν ίδια τελική συνεισφορά
2. **Dummy:** Αν υπάρχει χαρακτηριστικό {i} τέτοιο ώστε $\nu(\boldsymbol{S}) = \nu(\boldsymbol{S}.\mathrm{U}\{i\})$, για κάθε συμμαχία $\boldsymbol{S}$, που δεν περιέχει το {i}, τότε $\phi_i = 0$. Αυτό πρακτικά σημαίνει ότι αν ένας παίχτης δεν προσθέτει αξία σε καμία πιθανή συμμαχία, τότε η τελική συνεισφορά του είναι μηδενική.
3. **Additivity:** Ας υποθέσουμε ότι ορίζουμε δυο διαφορετικές χαρακτηριστικές συναρτήσεις για ένα παίγνιο, έστω u και v, και αντίστοιχα $\phi_i(u)$, $\phi_i(v)$ την συνεισφορά του παίχτη {i}. Τότε ισχύει: $\phi_i(u + v) = \phi_i(u) + \phi_i(v)$. Για να γίνει πιο ξεκάθαρο αυτό, θα βοηθήσει ένα σύντομο παράδειγμα. Ας υποθέσουμε ότι μια ομάδα εργαζομένων εργάζεται σε δύο διαφορετικά έργα και η συνολική απόδοση και η συνεισφορά τους σε κάθε έργο είναι διαφορετική. Στη συνέχεια, αν συνδυάσουμε αυτά τα έργα, η συνεισφορά ενός εργαζομένου στο συνδυασμένο έργο είναι το άθροισμα των συνεισφορών του σε κάθε έργο.

Γίνεται αντιληπτό σε αυτό το σημείο, ότι παρόλο που η τιμή Shapley δίνει σημαντικά αποτελέσματα για την ερμηνεία των black-box μοντέλων και των χαρακτηριστικών που τα καθοδηγούν, απαιτεί πολύ υπολογιστικό χρόνο, καθώς ένας ακριβής υπολογισμός της είναι πολύ κοστοβόρος, διότι αφενός απαιτεί να ελεγχθούν 2^M πιθανά coalitions, αφετέρου η απουσία κάποιου χαρακτηριστικού πρέπει να προσομοιωθεί με την δημιουργία τυχαίων instances, κάτι το οποίο αυξάνει την διακύμανση των εκτιμήσεων των τιμών Shapley. Το SHAP επομένως είναι ένα ενιαίο πλαίσιο που προτάθηκε από τους Lundberg και Lee [28] για την ερμηνεία αποφάσεων μοντέλων. Πιο συγκεκριμένα, εξηγεί την πρόβλεψη ενός δειγματικού στοιχείου x υπολογίζοντας τη συμβολή κάθε χαρακτηριστικού στην πρόβλεψη. Η λειτουργία του φαίνεται συνοπτικά στην παρακάτω εικόνα.



Εικόνα 9: SHAP framework

Μια σπουδαία καινοτομία που έφερε το framework του SHAP είναι ότι η μέθοδος εξήγησης μέσω της τιμής Shapley μπορεί να αναπαρασταθεί ως γραμμικό μοντέλο. Αυτή η οπτική συνέδεσε τον προηγούμενο αλγόριθμο που αναλύσαμε (LIME) και τις Shapley τιμές. Πλέον κάθε τιμή SHAP υπολογίζει πόσο συμβάλει κάθε χαρακτηριστικό του μοντέλου στην τελική απόφαση, είτε θετικά είτε αρνητικά. Δύο σημαντικά οφέλη που προκύπτουν είναι πρώτον ότι η τιμή SHAP μπορεί να υπολογιστεί για οποιοδήποτε μοντέλο (καθώς θυμίζουμε ότι είναι model-agnostic μεθοδολογία), και όχι μόνο για απλά γραμμικά μοντέλα και δεύτερον ότι κάθε εγγραφή έχει τις δικές της τιμές SHAP. Η αξία του τελευταίου θα φανεί στα πειράματα, καθώς έτσι παρέχεται τοπική επεξηγησιμότητα (local explainability), ενώ ο συνδυασμός αυτών παρέχει καθολική επεξηγησιμότητα (global explainability).

Το SHAP καθορίζει την εξήγηση ενός δειγματικού στοιχείου x ως εξής:

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

Όπου:

- g είναι το μοντέλο εξήγησης
- z' είναι το coalition vector (διάνυσμα συμμαχίας) και ισχύει ότι $z' \in \{0, 1\}^M$. Το 1 στο z' σημαίνει ότι τα χαρακτηριστικά στα επιλεγμένα δεδομένα υπάρχουν και είναι ίδια με αυτά των αρχικών δεδομένων, ενώ το 0 υποδηλώνει την απουσία του συγκεκριμένου χαρακτηριστικού
- M είναι το μέγεθος του μέγιστου coalition
- $\phi_j \in R$, είναι η συνεισφορά (τιμή Shapley) του χαρακτηριστικού j για ένα δειγματικό στοιχείο x. Αν το ϕ_j είναι μεγάλος θετικός αριθμός, αυτό σημαίνει ότι το χαρακτηριστικό j έχει σημαντικό θετικό αντίκτυπο στην πρόβλεψη που έκανε το μοντέλο

Στην απλούστερη περίπτωση, που όλα τα features υπάρχουν (μόνο 1 στο coalition vector), ο παραπάνω τύπος για ένα δειγματικό στοιχείο x απλοποιείται:

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

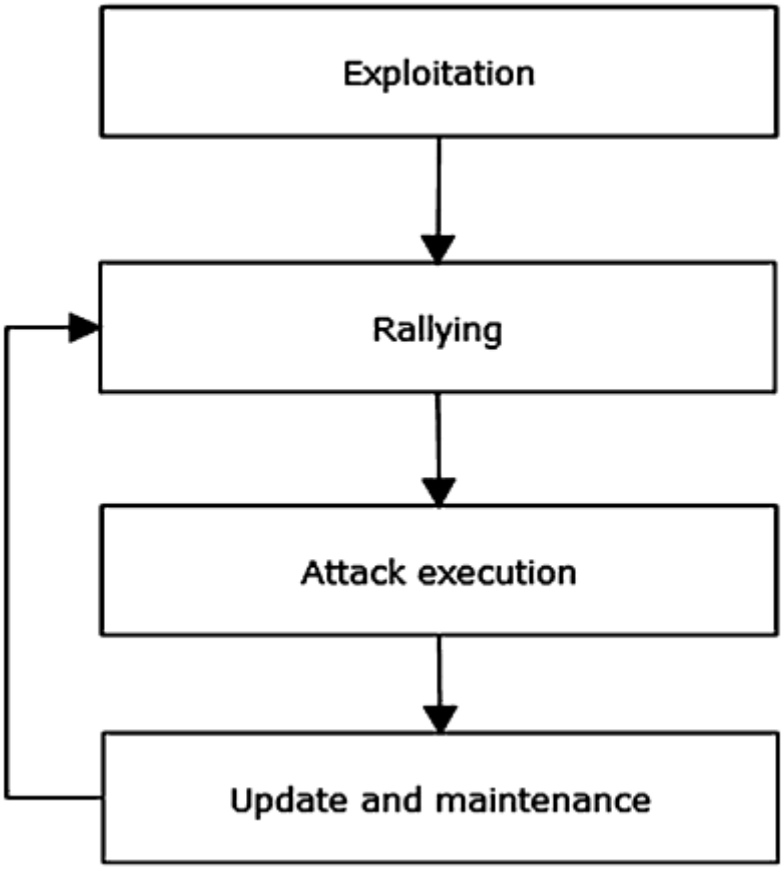
Κλείνοντας αυτή την ενότητα και μαζί και το θεωρητικό υπόβαθρο της παρούσας εργασίας, ας συνοψίσουμε τα πλεονεκτήματα του SHAP κι ας σχολιάσουμε σύντομα τους περιορισμούς του. Συγκριτικά με τις άλλες 2 μεθόδους που αναλύσαμε, το SHAP είναι το μόνο που έχει θεμελιώδεις αρχές, οι οποίες έχουν στηριχθεί στην θεωρία παιγνίων. Οι τιμές Shapley είναι οι μόνες που ικανοποιούν τις τρεις σημαντικές ιδιότητες Symmetry, Dummy, Additivity. Επιπρόσθετα, συνδέει το LIME με τις τιμές Shapley και βοηθά στην ενοποίηση της ερμηνείας των μοντέλων μηχανικής μάθησης κάτω από ένα συγκεκριμένο πλαίσιο. Η διαδικασία επεξήγησης μέσω της τιμής Shapley παρέχει global explainability πολύ αποδοτικότερα συγκριτικά με τις άλλες μεθόδους. Σε περιπτώσεις λοιπόν, που νομικές οντότητες ζητούν εξήγηση για τις αποφάσεις ενός μοντέλου, η τιμή Shapley είναι η μόνη νομικά συμβατή μέθοδος, επειδή βασίζεται σε θεμελιωμένη θεωρία και στην δίκαιη κατανομή. Τελευταίο και σημαντικότερο πλεονέκτημα για τις πρακτικές εφαρμογές είναι ότι αυτοί που το πρότειναν, παρέχουν έναν σχετικά γρήγορο υπολογισμό των SHAP τιμών για μοντέλα μηχανικής μάθησης σε σύγκριση με τον εξαντλητικό υπολογισμό των τιμών Shapley. Ωστόσο, στα μειονεκτήματα του είναι ότι ο υπολογιστικός χρόνος όσο καλές κι αν είναι οι υλοποιήσεις μας αυξάνεται εκθετικά με τον αριθμό των χαρακτηριστικών, κάτι που μας αναγκάζει να χρησιμοποιούμε ένα υποσύνολο των δεδομένων μας. Επίσης, δεν μπορούμε να χειριστούμε κάπως τα dependencies μεταξύ των χαρακτηριστικών, καθώς υπολογίζοντας τα διάφορα permutations θεωρούμε ότι τα χαρακτηριστικά είναι ανεξάρτητα. Ακόμη, οφείλουμε να υπενθυμίσουμε ότι τα μοντέλα δεν είναι πάντα μια καλή αναπαράσταση της πραγματικότητας και συχνά οι άνθρωποι αγαπούν να βρίσκουν μοτίβα που δεν υπάρχουν στην πραγματικότητα. Συνεπώς, η επεξήγηση με την βοήθεια των SHAP τιμών χωρίς γνώση του αντικειμένου μπορεί να οδηγήσει σε ψευδείς ιστορίες, ως προς τι ισχύει και τι όχι. Αυτό μπορεί να προκύπτει είτε από ιδιαιτερότητες των μοντέλων είτε από μεροληψία του ερευνητή που μπορεί κακόβουλα να προσπαθεί να υποστηρίξει κάποιο συμπέρασμα. Κλείνοντας αυτή την ενότητα, θεωρούμε σημαντικό να σημειώσουμε ότι οι XAI αλγόριθμοι και ιδίως το SHAP είναι σημαντικά εργαλεία για να «μετατρέψουμε» τα black-box σε white-box μοντέλα.

Εικόνα 10: Black-box to White-Box

Κεφάλαιο 3 – Συναφής Βιβλιογραφία

Ως τώρα έχουμε αναλύσει το θεωρητικό υπόβαθρο σχετικά με τον ρόλο και την λειτουργία των Domain Generation Algorithms στα σύγχρονα botnets, με στόχο την παραπλάνηση του DNS για την κυκλοφορία κακόβουλης δικτυακής κίνησης. Η μελέτη των DGAs βασίστηκε κυρίως στην έρευνα των Plohmann et al. (2016) [12] αναλύοντας την λειτουργία διαφόρων botnets που χρησιμοποιούν τους παραπάνω αλγορίθμους. Επιπλέον, οι Barabosch et al. (2012) [14] καθόρισαν την ταξινόμηση των DGA βάσει δύο ιδιοτήτων του seed, της χρονικής εξάρτησης (time dependence) και της αιτιότητας (determinism). Ακολούθως, παρουσιάσαμε αλγορίθμους XAI, οι οποίοι επιδιώκουν ερμηνεία black-box μοντέλων. Σε αυτό το κεφάλαιο θα αναλύσουμε σχετική βιβλιογραφία με τις προηγούμενες προσεγγίσεις για την ανίχνευση ονομάτων τομέα αλγοριθμικά παραγόμενων, τις πρώτες προσπάθειες ερμηνείας τέτοιων μοντέλων με χρήση μεθόδων επεξηγησιμότητας και τέλος θα σημειώσουμε την συνεισφορά μας στο αντικείμενο.

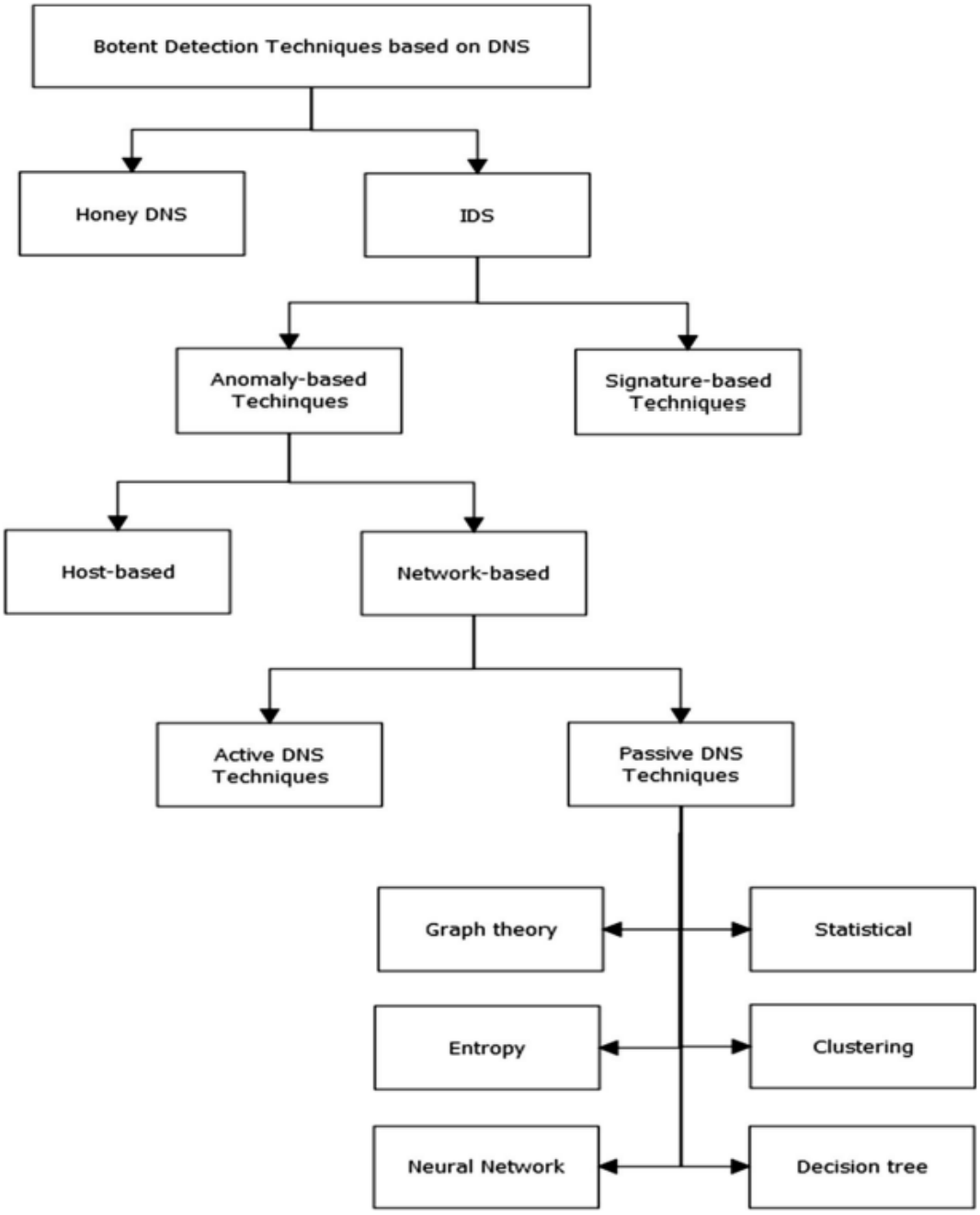
Αρχικά, οι Alieyan et al. (2015) [30] μελέτησαν αναλυτικά τον κύκλο ζωής (εικόνα 9) ενός botnet και παρουσίασαν τις τεχνικές για την ανίχνευση κακόβουλης κίνησης στο DNS από τέτοιες δομές, όπως φαίνεται και στην εικόνα 10 που προέρχεται από την σχετική εργασία τους. Το πρώτο στάδιο του κύκλου αναφέρεται στην μόλυνση του bot (exploitation). Το δεύτερο σχετίζεται με την επιστράτευση του bot μέσω της σύνδεσης του με τον C&C server (rallying). Το τρίτο μέρος του κύκλου ζωής είναι η εκτέλεση της επίθεσης (attack execution) από μια ομάδα bots, όπως καθορίστηκε από τον botmaster. Το τελευταίο κομμάτι του παζλ ολοκληρώνεται με την ενημέρωση και την συντήρηση του εκάστοτε bot (update and maintenance), όπου διαμοιράζονται τα νέα στοιχεία για την παραγωγή του domain name του C&C server, ώστε να γίνει εκ νέου resolve η τοποθεσία του και να δημιουργηθεί ξανά επικοινωνία μεταξύ του διακομιστή και του bot (rallying).



Εικόνα 11: Ο κύκλος ζωής ενός botnet

Οι τεχνικές εντοπισμού domain names που έχουν παραχθεί από Domain Generation Algorithms χωρίζονται σε 2 κατηγορίες

1. Ανασκοπικές (Retrospective). Αποτελεί πιο παραδοσιακή τεχνική, όπου αρχικά συλλέγεται η κίνηση και εκ των υστέρων γίνεται η ανίχνευση των bots συγκρίνοντας ερωτήματα DNS ανάμεσα στους hosts. Πιο συγκεκριμένα, αυτές οι τεχνικές βασίζονται στην ομοιότητα των NXDomain απαντήσεων που επιστρέφονται. Σύντομα, να αναφέρουμε ότι NXDomain response προκύπτει όταν δεν μπορεί να γίνει resolve ένα DNS Query. Αυτές οι τεχνικές βασίζονται στην ιδέα ότι τα bots που θα ανήκουν στην ίδια ομάδα και συνεπώς θα εκτελούν τον ίδιο DGA θα λαμβάνουν συχνά αποκρίσεις σφάλματος NXDomain στην προσπάθεια τους να αναγνωρίσουν τον C&C server και μάλιστα θα είναι αρκετά όμοιες αυτές μεταξύ τους. Εκμεταλλευόμενοι αυτή την ιδιότητα, οι ερευνητές μελέτησαν τις σχετικές αποκρίσεις, εξάγοντας στατιστικά χαρακτηριστικά για την κατηγοριοποίηση ονομάτων τομέα ανάμεσα σε έγκυρων ή αλγοριθμικά παραγόμενων
2. Σε πραγματικό χρόνο (Real-Time). Είναι πιο μοντέρνα τεχνική, όπου γίνεται χρήση μεθόδων μηχανικής μάθησης για την ταξινόμηση των μηνυμάτων DNS ένα προς ένα. Εδώ, αξιοποιούνται χαρακτηριστικά που σχετίζονται και προκύπτουν από την μορφή των domain names.



Εικόνα 12: Προσεγγίσεις Ανίχνευσης Botnet βάσει κίνησης DNS

3.1 Ανασκοπική (Retrospective) Ανίχνευση DGA

Μια πρώτη εργασία σχετικά με το αντικείμενο υλοποιήθηκε από τους McGrath και Gupta (2008) [31], οι οποίοι, με στόχο να ταξινομήσουν ονόματα τομέα σε γνήσια ή κακόβουλα για phishing, μελέτησαν IP διευθύνσεις, εγγραφές "Whois" και στατιστικά λεξιλογικά χαρακτηριστικά των ονομάτων. Η αρχική τους παρατήρηση κατέληξε στο ότι κάθε κλάση παρουσιάζει διαφορετικά χαρακτηριστικά όσον αφορά τους χαρακτήρες. Πιο συγκεκριμένα, όπως κι εμείς θα παρατηρήσουμε στα δικά μας πειράματα, τα κακόβουλα domain names είναι συντομότερα από τα έγκυρα και παρουσιάζουν σημαντικές διαφορές στις αλφαβητικές κατανομές τους. Αποτέλεσε μια σπουδαία εργασία καθώς ήταν ένα πρώτο βήμα προς στην εύρεση χαρακτηριστικών με στόχο το φιλτράρισμα phishing μηνυμάτων που προκύπτουν από την κίνηση ενός botnet. Όπως αναφέραμε στην εισαγωγή του συγκεκριμένου κεφαλαίου, η μέθοδος της ανασκοπικής ανίχνευσης domain name παραγόμενων από DGA βασίζεται κυρίως στην ανάλυση των NXDomain responses. Η πρώτη εργασία που εκμεταλλεύεται την ιδιότητα ότι τα DNS Queries που αφορούν ονόματα τομέα αλγοριθμικά παραγόμενα θα είναι ως επί το πλείστον αποκρίσεις NXDomain και μάλιστα αρκετά παρόμοιες μεταξύ τους καθώς οι μολυσμένες συσκευές του botnet χρησιμοποιούν τον ίδιο DGA είναι των Antonakakis et al. (2012) [4], με το σύστημα ανίχνευσης που ονόμασαν Pleiades. Εστιάζοντας μόνο σε τέτοιες αποκρίσεις μείωσαν το υπολογιστικό και το χρονικό κόστος για το σύστημά τους. Χρησιμοποιώντας ένα σύνολο στατιστικών χαρακτηριστικών όπως το μήκος των domain names, την κατανομή συχνότητας των χαρακτήρων κ.ά. σε συνδυασμό με διάφορες τεχνικές ταξινόμησης κατάφεραν να ομαδοποιήσουν domain names με παρόμοια χαρακτηριστικά και να προσδιορίσουν από ποιον DGA έχουν παραχθεί. Αναλυτικότερα, το σύστημα τους (Pleiades) λαμβάνει υπόψιν τα πιθανά DGA domain names που έχουν οδηγήσει σε NXDomain response και τα συγκρίνει τόσο με λίστες έγκυρων domain names όσο ήδη υπαρκτών κακόβουλων ονομάτων. Αφότου ομαδοποίησαν τα domain names κατάλληλα, ανέπτυξαν μια μέθοδο για να ανιχνεύουν τον C&C server και τελικά παρουσίασαν σημαντική επιτυχία καθώς μόλις σε διάστημα 15 μηνών κατάφεραν να ταυτοποιήσουν 12 DGAs, εκ των οποίων μόνο οι μισοί ήταν ήδη γνωστοί, με ακρίβεια που ξεπερνούσε το 95%. Ένα χρόνο μετά, το 2013 η έρευνα των Zhou et al. εστίασε και αυτή στις αποκρίσεις NXDomain και την ομαδοποίηση των DGA domain names, αυτή την φορά όμως με κριτήριο τον χρόνο ζωής του εκάστοτε ονόματος και την μέτρηση του πλήθους των DNS Queries που αντιστοιχούν στο κάθε domain name. Σε μια προσπάθεια κλιμακωσιμότητας της επίλυσης του ζητήματος της ανίχνευσης αλγοριθμικά παραγόμενων ονομάτων, οι Nguyen et al. (2015) [33] παρουσίασαν ένα σύστημα, το οποίο με την χρήση Collaborative Filtering και Density-Based Clustering μπορεί να ανιχνεύσει κακόβουλη κίνηση ενός botnet, αναλύοντας τ' αρχεία καταγραφής κίνησης DNS καθώς και νέες εκδόσεις ενός DGA, κάτι το οποίο αποτελεί τροχοπέδη σε μεθόδους reverse engineering. Παρότι αυτή η ιδέα φάνταζε ιδανική, δυστυχώς στην πράξη απέδωσε υψηλά ποσοστά ψευδώς θετικών (false positive) και ψευδώς αρνητικών (false negative) εγγραφών. Ο λόγος, όπως αναφέραμε και στην εισαγωγή

του κεφαλαίου είναι ότι προτού γίνει η ανάλυση των ονομάτων τομέων πρέπει να καταγραφεί ολόκληρη η κίνηση DNS που δημιουργείται από τους χρήστες. Μια προσπάθεια επίλυσης του παραπάνω προβλήματος παρουσιάστηκε στην έρευνα των Kwon et al. (2015) [34], οι οποίοι παρουσίασαν μια κλιμακώσιμη προσέγγιση, το PsyBog, αντιστοιχίζοντας τις διάφορες μετρικές τους στις IP διευθύνσεις και όχι στα domain names. Ο λόγος που επέλεξαν αυτή την πρακτική είναι διότι το πλήθος των διευθύνσεων IP παραμένει σταθερό, ενώ με την εξέλιξη του διαδικτύου το πλήθος των domain names αυξάνεται εκθετικά. Συγκριτικά με την προσπάθεια των Nguyen et al. [33], κατάφεραν ακρίβεια της τάξης του 95%, μείωσαν το ποσοστό των ψευδώς θετικών και αρνητικών εγγραφών και εντόπισαν 23 άγνωστες και 26 γνωστές οικογένειας κακόβουλου λογισμικού. Παρότι τα αποτελέσματα τους είναι ικανοποιητικά σε θεωρητικό επίπεδο, στην πράξη ένα τέτοιο σύστημα δεν μπορεί να εφαρμοστεί αποτελεσματικά διότι παρακάμπτεται πολύ εύκολα από τους κακόβουλους με την χρήση ενός VPN (Virtual Private Network), εφόσον η μέθοδός τους χρησιμοποιεί για την ταξινόμηση τις IP διευθύνσεις. Συνεπώς, για ακόμη μια φορά αντιλαμβανόμαστε ότι η διαδικασία εντοπισμού των bots μπορεί να αποβεί δύσκολη και χρονοβόρα. Πιο επιτυχημένες προσπάθειες με την μέθοδο της ανασκοπικής ανίχνευσης επιτεύχθηκαν αρχικά από το BotDigger το 2016 [35], το οποίο αναλύοντας την συχνότητα των DNS Queries και κάποια στατιστικά λεξιλογικά χαρακτηριστικά εντόπισε τα bots του Kraken malware [10] και των Conficker Bots [13]. Ακολούθως, το σύστημα DBod [36] αναπτύχθηκε από τους Wang et al. (2016) , σύμφωνα με τους οποίους, μπορεί να ανιχνεύει κακόβουλη δικτυακή κίνηση χωρίς προηγούμενη εκπαίδευση. Αξιοποιώντας τις αποκρίσεις NXDomain και εξάγοντας στατιστικά στοιχεία για την κίνηση DNS, πετυχαίνει ακρίβεια της τάξεως του 99%.

3.2 Real-Time Ανίχνευση DGA με Μηχανική Μάθηση

Όπως προκύπτει από την έρευνα των Krishnan et al. [37] και από την μελέτη των παραπάνω ερευνών, η retrospective προσέγγιση έχει μεγάλες απαιτήσεις σε χρόνο και την ανάγκη ύπαρξης εκ των προτέρων του συνόλου των ονομάτων τομέα. Συνεπώς, στις πραγματικές εφαρμογές παρουσιάζει μειωμένη απόδοση αυτή η προσέγγιση είτε ακόμη μπορεί να μην είναι και εφικτή, ειδικά αν θυμηθούμε ότι ο C&C server αλλάζει περιοδικά domain name. Τοιουτοτρόπως, οι ερευνητές στράφηκαν στον εντοπισμό και την κατηγοριοποίηση της κίνησης ενός botnet βασιζόμενοι στα domain names αυτά καθαυτά. Μια από τις πρώτες προσπάθειες προς αυτή την κατεύθυνση έκαναν οι Krishnan et al. (2013) [37], οι οποίοι παρουσίασαν μια μέθοδο που ανέλυε τα μοτίβα DNS κίνησης για το κάθε bot ξεχωριστά βασιζόμενη στις NXDomain αποκρίσεις. Ακολούθως, οι Raghuram et al. (38) δοκίμασαν ένα πιθανοτικό μοντέλο (generative probabilistic modeling) αξιοποιώντας τις κατανομές του αλφαβήτου των χαρακτήρων που προκύπτουν για τα έγκυρα και τα αλγοριθμικά παραγόμενα domain names.

Σε τέτοιου είδους προβλήματα ανίχνευσης ανωμαλιών οι αλγόριθμοι μηχανικής μάθησης ευδοκιμούν και δοκιμάζονται ολοένα και περισσότερο. Αποτελεί μια πιο μοντέρνα προσέγγιση, καθώς το μοντέλο εκπαιδεύεται από ένα σύνολο δεδομένων γνήσιων και DGA domain names και μπορεί να λειτουργήσει ως ταξινομητής μεταξύ αυτών των δύο κλάσεων. Έτσι, το αποτέλεσμα επαφίεται στην ικανότητα της μηχανής να διακρίνει την δικτυακή κίνηση. Προφανώς, από την μια αυτό οδήγησε σε σαφώς πιο περίπλοκα μοντέλα, καθότι είναι και black-box, από την άλλη όμως αυτή η προσέγγιση κατέστησε δυσκολότερη την ερμηνεία για τον ερευνητή. Σε αντίθεση με την ανασκοπική ανίχνευση, τα μοντέλα μηχανικής μάθησης προσφέρουν σαφέστατα πιο scalable υλοποιήσεις, ταχύτερη ανίχνευση καθώς μπορούν να εστιάσουν και να προβλέψουν ένα δειγματικό στοιχείο μεμονωμένα. Με την χρήση εξαγώγιμων στατιστικών χαρακτηριστικών δοκιμάστηκαν υλοποιήσεις με SVM [39], [40] πετυχαίνοντας ακρίβεια μεγαλύτερη του 95%. Αντίστοιχα, οι ερευνητές υλοποίησαν διάφορες λύσεις με Hidden Markov Models και Bayesian Networks [41], οι οποίες δεν έφεραν τα αναμενόμενα αποτελέσματα. Επιπρόσθετα, ενδιαφέρον παρουσιάστηκαν έρευνες με την χρήση Random Forest [42], το οποίο εν γένει είναι ένα ερμηνεύσιμο μοντέλο, πετυχαίνοντας υψηλή ακρίβεια, ωστόσο με αδυναμία ερμηνείας από τον άνθρωπο όσο μεγάλωνε το σύνολο των δεδομένων, όπως θα δείξουμε κι εμείς στα πειράματα μας.

Οι παραπάνω έρευνες βασίστηκαν όπως προαναφέραμε, στην εξαγωγή στατιστικών λεξιλογικών χαρακτηριστικών από τα domain names και σε κάθε προσέγγιση φαίνεται να λείπει κάτι από την εξίσωση για να αντιμετωπίσει όσο πληρέστερα γίνεται το πρόβλημα. Κάποιες τεχνικές απαιτούν πολύ χρόνο και υπολογιστικούς πόρους και κάποιες δεν μπορούν να γενικεύσουν ή παρακάμπτονται από τους κακόβουλους. Στην πράξη, όταν εξάγουμε κατ' αυτό τον τρόπο τα χαρακτηριστικά, αν ο διαχειριστής του botnet αντιληφθεί την προσέγγιση μας, τότε είναι εύκολο σ' ένα βαθμό να προσαρμόσει τον DGA που χρησιμοποιεί για να παραπλανήσει το δίκτυο. Η γενίκευση σε τέτοιες περιπτώσεις αποτελεί δύσκολο κατόρθωμα καθώς τα δεδομένα μας βασίζονται κατά κόρον σε γνωστούς DGA και όχι σε νέους που εμφανίζονται. Σε αντίθεση λοιπόν με τα παραπάνω, και την εισαγωγή της επιβλεπόμενης βαθιάς μηχανικής μάθησης στο παιχνίδι (Supervised Deep Learning), οι ερευνητές δοκιμάσαν να ταξινομήσουν τα domain names αξιοποιώντας είτε features αυτόματα παραγόμενα από τα μοντέλα, είτε την ακολουθία χαρακτήρων του domain name ως έχει. Την πρώτη και πολύ σημαντική έρευνα προς αυτή την κατεύθυνση παρουσίασαν οι Woodbridge et al. [43] το 2016, όπου χρησιμοποιήσαν ένα δίκτυο LSTM που χειρίζεται τα domain names σε επίπεδο χαρακτήρα, κάτι αντίστοιχο με την δική μας υλοποίηση. Η προσπάθεια τους έδειξε εξαιρετικά αποτελέσματα όσον αφορά την ακρίβεια καθώς δοκιμάστηκε σε περιβάλλον ανίχνευσης πραγματικού χρόνου. Παρόμοια αποτελέσματα με την χρήση LSTM παρουσίασαν οι Palak et al. (2020) [44]. Αξιοποιώντας την δυναμική των Long Short-Term Memory Networks οι Tran et al. [45] επέκτειναν το πρόβλημα από την δυαδική ταξινόμηση στην μελέτη μιας multiclass κατηγοριοποίησης, με στόχο να αξιολογήσουν την σημαντικότητα κάθε DGA αλγορίθμου. Το μοντέλο τους προτείνει έναν τρόπο χειρισμού των imbalances

(ανισορροπιών) που υπάρχουν στην ανίχνευση DGA-based botnets, ένα πρόβλημα που αποτελεί πολύ συχνό φαινόμενο στον πραγματικό κόσμο, καθώς ένας DGA μπορεί να παράγει πολλά ονόματα και κάποιος σαφώς λιγότερα, ανάλογα και με την τεχνική υλοποίησης του botnet. Αυτή η έρευνα μας ενέπνευσε κι εμάς να παρουσιάσουμε μια multiclass προσέγγιση του προβλήματος με στόχο την επεξηγησιμότητα, όπως θα αναλύσουμε και παρακάτω. Ακολούθως, οι Yu et al. [46] στην έρευνα τους το 2017 δοκίμασαν να συγκρίνουν ένα Convolutional Neural Network με τις υπάρχοντες λύσεις με LSTM δίκτυα. Η έρευνα τους έδειξε ότι τα τελευταία έχουν καλύτερη απόδοση κι έδειξαν ότι εκτός από την δυαδική κατηγοριοποίηση, με τις κατάλληλες τροποποιήσεις τα μοντέλα αυτά μπορούν να προβλέψουν και από ποια κατηγορία DGA έρχεται το κάθε domain name, θέτοντας αυστηρά όρια για το ποσοστό των ψευδώς χαρακτηρισμένων domain names ως γνήσια, ώστε μπορεί να χρησιμοποιηθεί το σύστημα τους σε περιβάλλον πραγματικής δικτυακής κίνησης. Ωστόσο, παρουσίασε αδυναμίες στην μελέτη των word-list based DGAs, που όπως σημειώσαμε και στο θεωρητικό υπόβαθρο είναι οι δυσκολότεροι προς ανίχνευση, καθώς χρησιμοποιούν συνδυασμούς πραγματικών υπαρκτών λέξεων. Κλείνοντας αυτή την ενότητα, διάφορες έρευνες εστίασαν στην ανίχνευση domain names που έχουν προέλθει από wordlist-based DGAs. Χαρακτηριστικά παραδείγματα αποτελούν η έρευνα των Liu at al. (2018) [47], η οποία βασίστηκε στην συχνότητα εμφάνισης συγκεκριμένων λέξεων, η έρευνα των Sai Charan et al. (2020) [48] οι οποίοι χρησιμοποιήσαν Ensemble και GAN (Generative Adversarial Networks) και τέλος η έρευνα των Curtin et al. [49], οι οποίοι χρησιμοποιώντας ένα LSTM δίκτυο, χρησιμοποιώντας ως χαρακτηριστικό την ομοιότητα μεταξύ ονομάτων που δημιουργήθηκαν από wordlist-based DGAs κι ενός συνόλου αγγλικών λέξεων, επιτυγχάνοντας ποσοστά ψευδώς ταξινομημένων παρατηρήσεων χαμηλότερα του 1%.

3.3 Χρήση eXplainability AI για τον εντοπισμό DNS κίνησης μέσω DGA

Όπως έχουμε αναφέρει και στην εισαγωγή μας, και όπως φάνηκε από τις παραπάνω ενότητες η ανάγκη για μεγαλύτερη ακρίβεια και καλύτερα αποτελέσματα ώθησε τους ερευνητές να αφήσουν απλούστερα αλλά επεξηγήσιμα μοντέλα και ταυτόχρονα να αναπτύξουν περίπλοκα και ακριβά σε υπολογιστικό κόστος μοντέλα, τα οποία δυστυχώς είναι μη επεξηγήσιμα. Αυτό συμβαίνει διότι, η λειτουργία των black-box μοντέλων δεν μας επιτρέπει ως ερευνητές να κατανοήσουμε επαρκώς τι τα οδηγεί σε προβλέψεις και αποφάσεις που λαμβάνουν, παρότι επιτυγχάνουν σημαντική ακρίβεια. Τους λόγους που αυτό είναι μείζονος σημασίας τους έχουμε τονίσει στα παραπάνω κεφάλαια και θα αναδειχθούν και στον σχολιασμό των πειραμάτων μας. Όπως σημειώνουν οι Wang et al. στην έρευνα τους [50], δεν υπάρχουν πολλές εργασίες για ερμηνεία μοντέλων που να αφορούν την μελέτη δικτυακής κίνησης, καθώς οι περισσότερες εστιάζονται σε άλλους τομείς όπως όραση υπολογιστών, επεξεργασία

φυσικής γλώσσας ακόμη και βιολογία. Ωστόσο, οι αλγόριθμοι XAI και πιο συγκεκριμένα το SHAP είναι «jack of all trades», καθώς ως ένα model-agnostic framework μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο και να το ερμηνεύσει χρησιμοποιώντας τις ίδιες μετρικές. Ωστόσο, η έλλειψη τέτοιων ερευνών ευρύτερα στον κλάδο της κυβερνοασφάλειας δημιουργούν αδυναμίες κατά την πρακτική χρήση των διαφόρων μεθόδων, διότι οι ερευνητές αν δεν είναι σε θέση να κατανοήσουν την κρίση του εκάστοτε μοντέλου δεν μπορούν εύκολα να προχωρήσουν σε βελτιώσεις. Στην μελέτη τους οι Wang et al. [50] χρησιμοποίησαν το framework του Shapley Additive exPlanations) για να ερμηνεύσουν αλγορίθμους, σχετιζόμενους με συστήματα ανίχνευσης δικτυακής εισβολής (Intrusion Detection Systems). Οι έρευνες που συνδέουν XAI αλγορίθμους με τον εντοπισμό domain names που έχουν παραχθεί από DGA είναι περιορισμένες και σε πρωταρχικό στάδιο ακόμη, καθώς έχουν ασχοληθεί μόνο με μοντέλα μηχανικής μάθησης, αλλά όχι βαθιάς μηχανικής μάθησης όπως επιχειρούμε εμείς στην παρούσα εργασία. Μια πρώτη προσπάθεια έγινε από τους Dricher et al. (2021) [51], οι οποίοι πρότειναν το EXPLAIN, όπως ονόμασαν το σύστημα τους, το οποίο βασίζεται σε χαρακτηριστικά εξαγόμενα από τον άνθρωπο για την ερμηνεία των αποτελεσμάτων μιας Multiclass κατηγοριοποίησης των αλγοριθμικά παραγόμενων domain names. Μάλιστα συνέκριναν τα αποτελέσματα της μεθόδου τους με το FANCI (Feature-based Automated NXDomain Classification and Intelligence) [52], το οποίο είναι ένα σύστημα για την κατηγοριοποίηση των domain names, το οποίο χρησιμοποιεί SVM (Support Vector Machine) και RF (Random Forest) ταξινομητές, καθώς τα ίδια χρησιμοποίησαν και οι Dricher et al. [51]. Επιπλέον, σύμφωνα με την μελέτη τους η επιλογή των χαρακτηριστικών απαιτεί πολύ μεγαλύτερη προσπάθεια σε σύγκριση με τη χρήση ταξινομητών βαθιάς μάθησης, όπου όλες οι πληροφορίες απλώς κωδικοποιούνται και παρέχονται στο μοντέλο και ουσιαστικά αυτό μαθαίνει από μόνο του. Γι' αυτό τον λόγο επέλεξαν το πλαίσιο EXPLAIN που ανέπτυξαν να επικεντρωθεί στην επιλογή των χαρακτηριστικών, στο feature engineering και στην βελτιστοποίηση των υπερπαραμέτρων. Οι Suryotrisongko et al. (2021) [53] παρατηρώντας για χρόνια καταγραφές DNS ερωτημάτων και μελετώντας την λειτουργία και τον εντοπισμό DGA-based botnets, πρότειναν ένα πλαίσιο που συνδυάζει τους XAI αλγορίθμους (LIME, SHAP, Counterfactual κ.ά.) με το OSINT (open-source intelligence), το οποίο αναφέρεται στην συλλογή και ανάλυση δεδομένων από ανοιχτές πηγές για την παραγωγή actionable intelligence, δηλαδή γνώσης που μπορεί να χρησιμοποιηθεί για την δράση σε πραγματικές εφαρμογές. Πιο συγκεκριμένα, για την ανίχνευση κίνησης που βασίζεται σε DGA χρησιμοποίησαν Random Forest ταξινομητές με ακρίβεια της τάξης του 96%. Στόχος του συνδυασμού XAI αλγορίθμων με το OSINT είναι η ύπαρξη αποδεικτικών στοιχείων για την επικύρωση του προτεινόμενου υπολογιστικού πλαισίου τους, ώστε να ενισχυθεί η προσπάθεια των αναλυτών κυβερνοασφάλειας σε επιχειρήσεις ασφαλείας, αξιοποιώντας ένα αυτοματοποιημένο κι επεξηγήσιμο framework. Τοιουτοτρόπως, περιορίζεται η χειροκίνητη παρέμβαση στην κρίσιμη λήψη αποφάσεων, εφόσον τόσο οι ερευνητές όσο και οι νομικές οντότητες έχουν αποδεικτικά στοιχεία για τις προβλέψεις του μοντέλου. Τέλος, οι Piras et al. (2022) [54] παρουσίασαν το framework EXPOSURE, για την ερμηνεία DGA ανιχνευτών από δεδομένα DNS δικτυακής κίνησης. Στην πράξη, συνέλεξαν έγκυρων και αλγοριθμικά

παραγόμενων domain names, τα οποία ταξινόμησαν και αξιολόγησαν με διάφορα μοντέλα: Decision Trees (DT), Support-Vector Machine (SVM), Ada-Boost (ADA), K-Nearest Neighbors KNN και Random Forest (RF). Κατόπιν, με την βοήθεια του SHAP παρουσίασαν τα αποτελέσματα των παραπάνω μεθόδων για τοπική και καθολική επεξηγησιμότητα.

3.4 Συνεισφορά της Παρούσας Διπλωματικής

Στην παρούσα εργασία θα παρουσιαστούν παρακάτω Decision Tree, Multilayer Perceptron και LSTM μοντέλα με στόχο την επίλυση του προβλήματος ανίχνευσης κακόβουλης δικτυακής κίνησης παραγόμενη από DGA-based botnets. Εξ όσων γνωρίζουμε, XAI αλγόριθμοι για το συγκεκριμένο πρόβλημα έχουν εφαρμοστεί σε μοντέλα μηχανικής μάθησης, όπως αναλύσαμε και στην παραπάνω ενότητα, ωστόσο επιθυμούμε να κάνουμε την πρώτη προσπάθεια για ερμηνεία ταξινομητών βαθιάς μηχανικής μάθησης (MLP, LSTM) επεκτείνοντας έτσι την σχετική βιβλιογραφία. Θα παρουσιάσουμε κυρίως το πρόβλημα της δυαδικής ταξινόμησης και στο τέλος μια προσπάθεια multiclass προσέγγισης, έχοντας πάντα ως στόχο την επεξηγησιμότητα. Θεωρούμε μείζονος σημασίας την δυνατότητα της κατανόησης από τον άνθρωπο την διαδικασία λήψης αποφάσεων black-box μοντέλων.

Κεφάλαιο 4 – Dataset και Μεθοδολογία Επιλεγμένων Μοντέλων Μηχανικής Μάθησης

4.1 Binary Dataset

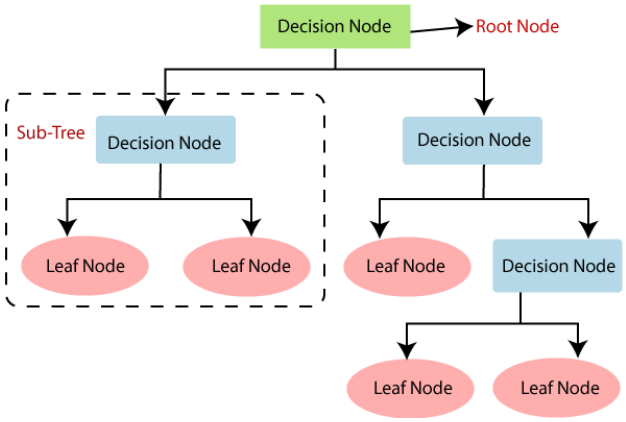
Παραδοσιακά ένα dataset το οποίο θα χρησιμοποιηθεί για την εκπαίδευση κάποιο μοντέλου χωρίζεται σε 2 μέρη, το train-set και το validation-set. Τα ποσοστά διαμέρισης που επιλέξαμε στην παρούσα εργασία είναι 80% για το training και 20% για το testing. Στο μεγαλύτερο εύρος των πειραμάτων ασχολούμαστε με την δυαδική ταξινόμηση των domain names ανάμεσα σε έγκυρα (legit) και κακόβουλα. Για την πρώτη κατηγορία αντλήσαμε δεδομένα από τις συλλογές Alexa Top 1 Million και Cisco Top 1 Million, όπου ύστερα από την σχετική εκκαθάριση και την αφαίρεση των διπλότυπων (duplicates) καταλήξαμε με 1.5 εκατομμύριο εγγραφές από επίσημα κι έγκυρα domain names από τις παραπάνω δημοφιλείς λίστες. Από την άλλη, αντλήσαμε τα αλγοριθμικά παραγόμενα δεδομένα από το Netlab OpenData Project [55], το οποίο περιέχει συνολικά περίπου 1 εκατομμύριο εγγραφές, οι οποίες προέρχονται από 63 διαφορετικές οικογένειες Domain Generation Algorithms. Αυτά τα ονόματα τομέα χρησιμοποιούν κυρίως ως generation scheme τον υπολογισμό μιας αλφαριθμητικής (alphanumeric) παράστασης κι έχουν προκύψει μέσω reverse-engineering από real data malware. Τον τρόπο που τα αξιοποιήσαμε στην εκάστοτε περίπτωση θα τον αναφέρουμε στις παρακάτω ενότητες αυτού του κεφαλαίου όπου θα παρουσιάσουμε συνοπτικά τα μοντέλα μηχανικής και βαθιάς μάθησης που επιλέξαμε.

4.2 Multiclass Dataset

Όπως, έχουμε αναφέρει και παραπάνω παρουσιάζουμε και μια multiclass προσέγγιση με την εκπαίδευση να έχει γίνει με Long Short-Term Memory Network. Εδώ, ο στόχος είναι η δημιουργία ενός explainable μοντέλου για να κατανοήσουμε τι επηρεάζει και τι όχι για να κατηγοριοποιήσει το μοντέλο ένα instance στην εκάστοτε κατηγορία, δηλαδή ενός μοντέλου όπου με την βοήθεια του SHAP framework προσεγγίζει το γενικό μοντέλο μας. Για αυτό τον λόγο, κρατήσαμε τις 11 οικογένειες Domain Generation Algorithms με τα περισσότερα instances στο δείγμα, από τις συνολικά 63. Σε πληθικότητα αντιστοιχούν περίπου στο 90% του αρχικού δείγματος, αν αναλογιστούμε ότι το support του validation-set (20%) είναι 186.277 δειγματικά στοιχεία, ωστόσο αναλυτικότερες λεπτομέρειες για τα κριτήρια αξιολόγησης της απόδοσης και το μέγεθος του συνόλου των δεδομένων θα δούμε στον σχολιασμό των πειραμάτων και αποτελεσμάτων.

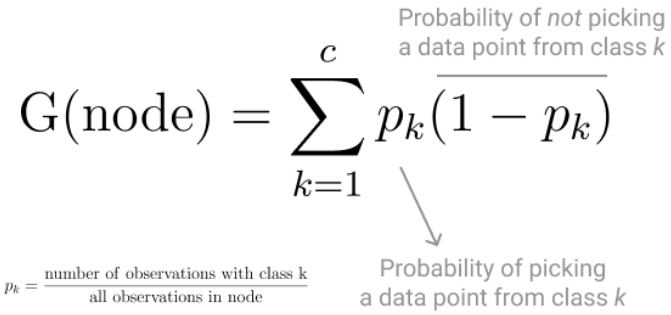
4.3 Decision Tree – Binary Classification

Σε αυτή και τις παρακάτω υποενότητες αυτού του κεφαλαίου θα παρουσιάσουμε συνοπτικά το μοντέλο μάθησης που χρησιμοποιήθηκε για την ανίχνευση κίνησης από DGA-based botnet. Η πρώτη προσέγγιση με Δέντρο Απόφασης χρησιμοποιεί την υλοποίηση της γνωστής βιβλιοθήκης της python, scikit-learn [56]. Το Decision Tree (DT) είναι ένας αλγόριθμος επιβλεπόμενης μηχανικής μάθησης που χρησιμοποιεί ένα σύνολο κανόνων για την λήψη αποφάσεων, παρόμοια με τον τρόπο που οι άνθρωποι λαμβάνουν αποφάσεις. Η ιδέα αυτού του αλγορίθμου είναι η χρήση των χαρακτηριστικών (features) του dataset για να δημιουργήσει ναι/όχι ερωτήσεις και να διακλαδίζει (split) συνεχώς το dataset μέχρι να απομονώσει όλα τα δειγματικά στοιχεία που ανήκουν σε κάθε κλάση. Εξού και προκύπτει αυτή η δενδρική δομή.

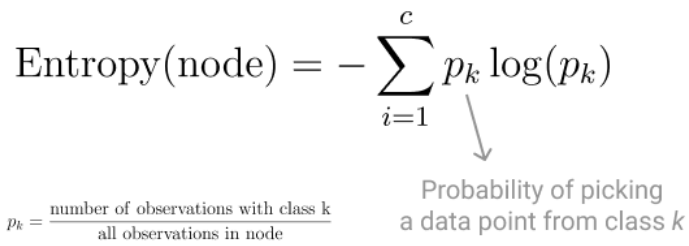


Εικόνα 13: Decision Tree

Με την εκτέλεση κάθε «ερώτησης» προστίθεται ένας κόμβος στο δέντρο, με τον πρώτο να ονομάζεται root node. Μετά το πέρας της διαδικασίας βάσει κάποιων κριτηρίων τερματισμού προκύπτουν οι κόμβοι φύλλα (leaf nodes), οι οποίοι είναι οι τελευταίοι που δημιουργήθηκαν. Οι δύο πιο κλασσικές μετρικές που χρησιμοποιούνται στα Δέντρα Αποφάσεων για την αξιολόγηση του split με βάση την καθαρότητα των κόμβων που προκύπτουν είναι το Gini Impurity και η Εντροπία (Entropy), οι οποίες συγκρίνουν την κατανομή της κλάσης πριν και μετά το split.



Εικόνα 14: Gini Impurity για έναν κόμβο



Εικόνα 15: Εντροπία (Entropy) ενός κόμβου

Το μεγάλο πλεονέκτημα του Decision Tree είναι η δυνατότητα ερμηνείας λόγω της οπτικοποίησής του, καθώς φαίνεται η σειρά των κανόνων που εφαρμόστηκαν για την εκάστοτε πρόβλεψη. Ωστόσο, αυτό αποτελεί ταυτόχρονα και την αδυναμία του, καθώς με την αύξηση του όγκου των δεδομένων και την τυχαιότητα που εισάγεται αλλοιώνονται τα αποτελέσματα του και είναι αδύνατο να οπτικοποιηθεί με τρόπο ικανό να μελετηθεί από άνθρωπο, καθιστώντας σαφώς δυσκολότερη την τοπική ερμηνεία (local explainability). Συνεπώς, αντιλαμβανόμαστε ότι υπάρχει ένα trade-off ανάμεσα στην ερμηνευσημότητα (interpretability) και στην απόδοση. Μπορούμε σχετικά εύκολα να οπτικοποιήσουμε και να ερμηνεύσουμε ένα μικρό δέντρο, ωστόσο αυτό θα έχει υψηλή διακύμανση. Δηλαδή, μια μικρή αλλαγή στο training dataset μπορεί να οδηγήσει σ' ένα εντελώς διαφορετικό δέντρο και πιθανώς και διαφορετικές προβλέψεις. Από την άλλη, ένα μεγάλο δέντρο με πολλούς κόμβους και πολλαπλά splits παράγει καλύτερη ταξινόμηση. Δυστυχώς όμως στην πράξη, απλά μαθαίνει και απομνημονεύει το training dataset και αδυνατεί να ταξινομήσει δεδομένα που δεν έχει δει προηγουμένως. Επομένως, θεωρούμε ότι η επιλογή να χρησιμοποιήσουμε Decision Tree είναι ένα καλό benchmark για να αντιληφθούμε την εγγενή επεξηγησιμότητα του αλλά και τις πιθανές παθογένειές του, όπως την αδυναμία τοπικής εξήγησης συνεισφοράς του εκάστοτε χαρακτηριστικού, χρησιμοποιώντας το Gini Index και την εντροπία για να αξιολογήσουμε τους κόμβους του δέντρου.

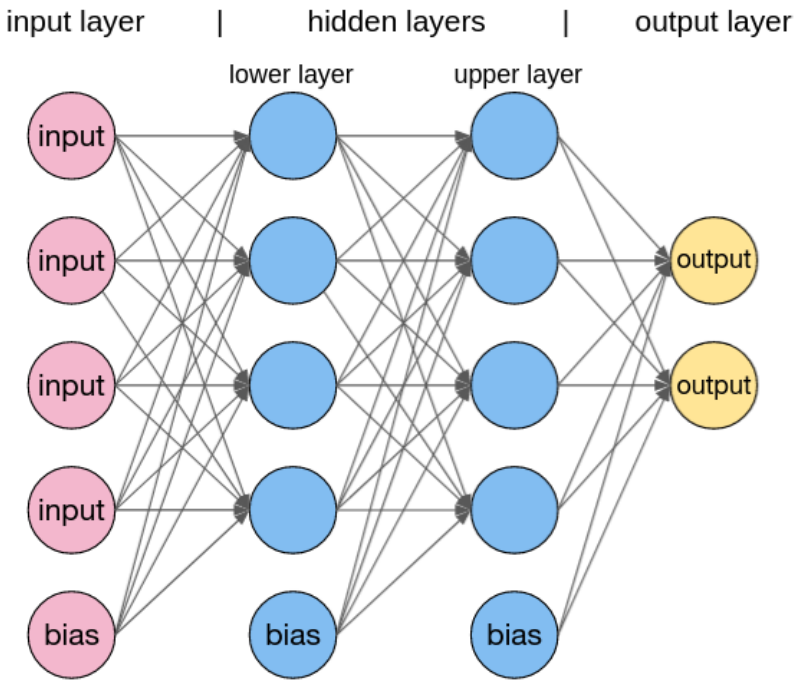
Όσον αφορά τα δικά μας πειράματα εξαγάγαμε 13 χαρακτηριστικά βασισμένα στο σύνολο των δεδομένων μας, τα οποία χρησιμοποιήθηκαν για την εκπαίδευση του Δέντρου Αποφάσεων, με βάση τα οποία αυτό έφτιαξε τους rule-based splits. Τα χαρακτηριστικά αυτά είναι τα εξής:

1. Πλήθος DNS Labels (τμήματα domain name)
2. Πλήθος συμφώνων
3. Μέγιστη ακολουθία συμφώνων
4. Ελάχιστη ακολουθία συμφώνων
5. Πλήθος φωνηέντων
6. Μέγιστη ακολουθία φωνηέντων
7. Ελάχιστη ακολουθία φωνηέντων
8. Πλήθος ψηφίων
9. Μέγιστη ακολουθία ψηφίων
10. Ελάχιστη ακολουθία ψηφίων
11. Πλήθος hyphens ("-")

12. Μέγιστη ακολουθία hyphens
13. Ελάχιστη ακολουθία hyphens

4.4 Multilayer Perceptron – Binary Classification

Ακολούθως, περνώντας στην πρώτη προσέγγιση με μοντέλο βαθιάς μηχανικής μάθησης χρησιμοποιούμε το Multilayer Perceptron με στόχο την διερεύνηση της δυνατότητας εξήγησης που μας παρέχει. Αυτό, θα εκπαιδευθεί με βάση τα ίδια 13 χαρακτηριστικά που χρησιμοποιήσαμε για το Decision Tree. Κατόπιν της εκπαίδευσης του MLP, επιλέγουμε τυχαία 1000 δειγματικά στοιχεία τα οποίο φορτώνουμε στο explainable model, και με την βοήθεια του DeepExplainer που παρέχεται από την βιβλιοθήκη SHAP [57], υπολογίζονται τα SHAP values που θα χρησιμοποιήσουμε για την υλοποίηση και παρουσίαση των σχετικών διαγραμμάτων μας με στόχο την τοπική αλλά και καθολική επεξηγησιμότητα των αποφάσεων του black-box μοντέλου βαθιάς μηχανικής μάθησης. Το SHAP χρειάζεται δειγματοληψία, διότι ελέγχει διάφορους συνδυασμούς των features και συνεπώς με την προσθήκη όλο και περισσότερων στοιχείων στο explainable μοντέλο του, ο χρόνος αυξάνεται εκθετικά. Πιο συγκεκριμένα, ο DeepExplainer αποτελεί μια βελτιωμένη έκδοση του αλγορίθμου DeepLIFT (Deep SHAP) [28], όπου προσεγγίζονται τα conditional expectations των τιμών SHAP χρησιμοποιώντας μια επιλογή από το σύνολο των δεδομένων που χρησιμοποιήθηκαν για εκπαίδευση. Ενσωματώνοντας όλο και περισσότερα δεδομένα στην δειγματοληψία, αποδείχτηκε ότι οι εκτιμήσεις του Deep SHAP προσεγγίζουν τις τιμές SHAP, έτσι ώστε αυτές να αθροίζουν στη διαφορά μεταξύ της αναμενόμενης εξόδου του μοντέλου στα δειγματοληπτημένα στοιχεία και στην τρέχουσα έξοδο του μοντέλου (f(x) - E[f(x)]). Για την υλοποίηση του MLP χρησιμοποιήσαμε το keras API [58], από την βιβλιοθήκη του TensorFlow [59].



Εικόνα 16: Multilayer Perceptron

Το MLP είναι ένα τεχνητό νευρωνικό δίκτυο τροφοδοσίας (feedforward artificial neural network) που παράγει ένα σύνολο εξόδων από ένα σύνολο εισόδων, με την μεσολάβηση κάποιων hidden layers. Ας δούμε σε σύντομα βήματα τον αλγόριθμο που ακολουθεί:

1. Οι είσοδοι του ωθούνται προς τα εμπρός μέσω του MLP υπολογίζοντας το dot product μεταξύ της εισόδου και των βαρών που υπάρχουν μεταξύ του στρώματος εισόδου και κρυφού επιπέδου. Το αποτέλεσμα αυτής της πράξης αποδίδει μια τιμή στο κρυφό στρώμα, Ωστόσο, αυτή δεν προωθείται ως έχει.

2. Το MLP χρησιμοποιεί συναρτήσεις ενεργοποίησης (activation function) σε κάθε ένα από τα υπολογιζόμενα layers τους. Πιο γνωστές τέτοιες συναρτήσεις είναι η ReLU [60], η sigmoid και η tanh. Εφαρμόζουμε σε κάποια από αυτές τις συναρτήσεις το αποτέλεσμα του βήματος 1, και αυτό που βρίσκουμε αποτελεί την έξοδο του layer στο οποίο βρισκόμαστε.

3. Μόλις η έξοδος του hidden layer υπολογιστεί μέσω της συνάρτησης ενεργοποίησης, προωθείται στο επόμενο στρώμα στο MLP παίρνοντας το dot product με τα αντίστοιχα βάρη.

4. Επαναλαμβάνονται τα βήματα 2 και 3 έως ότου φτάσουμε στο επίπεδο εξόδου.

5. Στο επίπεδο εξόδου, οι υπολογισμοί είτε θα χρησιμοποιηθούν για έναν αλγόριθμο backpropagation που αντιστοιχεί στη συνάρτηση ενεργοποίησης που επιλέχθηκε για το MLP (στην περίπτωση εκπαίδευσης - training) είτε θα ληφθεί απόφαση με βάση την έξοδο (στην περίπτωση ελέγχου - testing).

Πιο συγκεκριμένα, η υλοποίηση του δικού μας MLP, ύστερα από μερικές δοκιμές, διαστρωματώνεται ως εξής:

- Input Layer, μήκους 13 (όσα και τα features μας)
- Dense Layer (Hidden), με 128 νευρώνες και συνάρτηση ενεργοποίησης ReLU
- Dropout (για την αποφυγή overfitting)
- Dense Layer (Hidden), με 32 νευρώνες και συνάρτηση ενεργοποίησης ReLU
- Dropout (για την αποφυγή overfitting)
- Output Layer, με 2 νευρώνες (όσες και οι κατηγορίες μας) και συνάρτηση ενεργοποίησης softmax (η οποία ταυτίζεται με την sigmoid στην περίπτωση της δυαδικής κατηγοριοποίησης)
- Ο optimizer που χρησιμοποιήθηκε είναι ο rmsprop

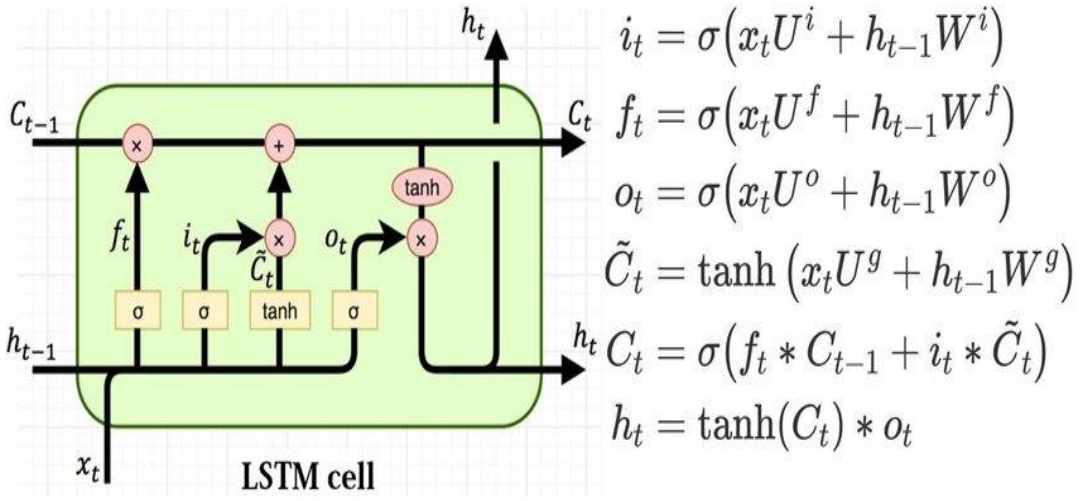
```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Input(shape=[length]))
model.add(tf.keras.layers.Dense(units=128, activation="relu"))
model.add(tf.keras.layers.Dropout(rate=0.5))
model.add(tf.keras.layers.Dense(units=32, activation="relu"))
model.add(tf.keras.layers.Dropout(rate=0.5))
model.add(tf.keras.layers.Dense(units=2, activation="softmax"))
model.compile(loss=tf.losses.BinaryCrossentropy(), optimizer="rmsprop", metrics=[tf.metrics.BinaryAccuracy()])
model.summary()
```

Εικόνα 17: Υλοποίηση Multilayer Perceptron

4.5 LSTM – Binary and Multiclass Classification

Σε αυτή την υποενότητα θα παρουσιάσουμε την υλοποίηση του προβλήματος ώστε να επιτύχουμε την επεξηγησιμότητα βάσει της ακολουθίας των χαρακτήρων στο κάθε domain name και με την χρήση Long Short-Term Memory Network. Για να μπορέσουμε να δούμε πιο βαθιά ως προς την επεξηγησιμότητα και την ερμηνεία των αποτελεσμάτων, σε αυτό το σημείο αντικαθιστούμε την εκπαίδευση που ως τώρα γινόταν με στατιστικά χαρακτηριστικά του κάθε domain name όπως τα εξαγάγαμε, με την χρήση επικαλυπτόμενων (overlapping) n-grams ως features. Ο λόγος αυτής της επιλογής θα φανεί καλύτερα στο κεφάλαιο της πειραματικής αξιολόγησης, όπου θα κατανοήσουμε τις παθογένειες της υλοποίησης με στατιστικά χαρακτηριστικά, όπως για παράδειγμα ότι 2 παρεμφερή domain names (π.χ. permutation ανάμεσα σε δυο χαρακτήρες του ίδιου domain name) μπορεί να έχουν τα ίδια στοιχεία, αλλά το ένα να είναι legit και το άλλο παραγόμενο από DGA. Στα πλαίσια της εργασίας αυτής, η

εκπαίδευση και τα πειράματα έγιναν με bigrams και trigrams, ωστόσο για την παρουσίαση και την συγγραφή χρησιμοποιούνται μόνο τα trigrams, καθότι παρατηρήσαμε ότι δίνουν πιο μεστά αποτελέσματα. Για να κατανοήσουμε λίγο καλύτερα τον όρο overlapping n-grams ας δούμε το παράδειγμα του google.com, το οποίο θα αναλυθεί σε “goo”, “oog”, “ogl”, “gle”, “le.”, “e.c”, “.co”, “com”. Κατόπιν της εκπαίδευσης του LSTM, επιλέγουμε τυχαία 1000 δειγματικά στοιχεία τα οποίο φορτώνουμε στο explainable model, και με την βοήθεια του DeepExplainer που παρέχεται από την βιβλιοθήκη SHAP [57], υπολογίζονται τα SHAP values που θα χρησιμοποιήσουμε για την υλοποίηση και παρουσίαση των σχετικών διαγραμμάτων μας με στόχο σε αυτή την περίπτωση την καθολική επεξηγησιμότητα (global explainability) των αποφάσεων του black-box μοντέλου βαθιάς μηχανικής μάθησης. Η τοπική επεξηγησιμότητα κατανοούμε ότι δεν έχει αξία εδώ, καθώς δεν υπάρχει κοινή βάση σύγκρισης για κάθε domain name, όπως γινόταν με τα στατιστικά χαρακτηριστικά, αλλά το καθένα έχει τον δικό του μοναδικό συνδυασμό n-grams. Για την υλοποίηση του LSTM χρησιμοποιήσαμε το keras API [58], από την βιβλιοθήκη του TensorFlow [59]. Συνεπώς, ο λόγος που χρησιμοποιούμε LSTM είναι για να αξιοποιηθεί η ακολουθία των χαρακτήρων, κάτι το οποίο θα γίνει σαφές παρακάτω στην ανάλυση λειτουργίας ενός τέτοιου δικτύου.

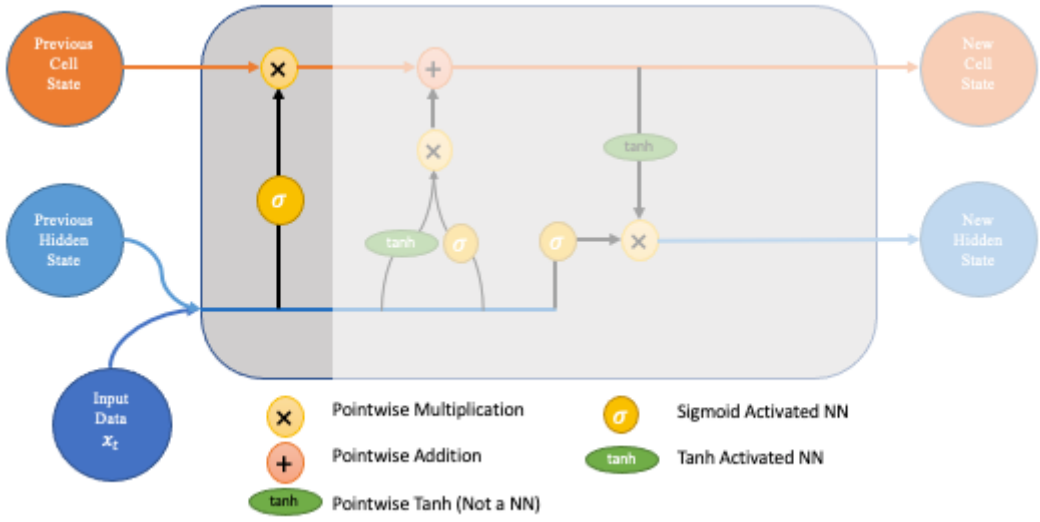


Εικόνα 18: LSTM Memory Block και οι εξισώσεις που το περιγράφουν

Το Long Short-Term Memory μοντέλο αποτελεί ειδική περίπτωση ενός Recurrent Neural Network (RNN) μοντέλου. Τα RNN γενικά χρησιμοποιούνται για εφαρμογές επεξεργασίας κειμένου, γενικότερα για την επεξεργασία συμβολοσειρών και χρονοσειρών, καθώς και για την ανίχνευση εξαρτήσεων μεταξύ χαρακτήρων μιας ακολουθίας, όπως είναι στην περίπτωση μας ένα domain name. Στο RNN μοντέλο, η έξοδος είναι συνάρτηση τόσο της ισχύουσας εισόδου, όσο και των προϋπαρχουσών με ανάδραση, δηλαδή στην περίπτωση μας η έξοδος εξαρτάται κι από προηγούμενα σύμβολα της ακολουθίας. Τοιουτοτρόπως δημιουργείται ένας χάρτης συμφραζομένων για κάθε ακολουθία. Ωστόσο, συνήθη προβλήματα που προκύπτουν λόγω των μακρών αλυσιδωτών αναδρομών (π.χ. από ένα domain name μεγάλου μήκους) στα

παραδοσιακά RNN είναι το vanishing gradient problem, όπου η έξοδος μπορεί να μειώνεται συνεχώς μέχρι που τείνει να εξαφανιστεί και το exploding gradient problem, όπου η έξοδος αυξάνεται συνεχώς. Αυτό αποτελεί τροχοπέδη για την εκμάθηση μακρών αλληλεξαρτήσεων για το εκάστοτε domain name. Σε αυτό το σημείο εισέρχονται τα LSTM μοντέλα, που προσθέτουν το ομώνυμο block. Στην Εικόνα 18 βλέπουμε αυτό το block και τις εξισώσεις που το περιγράφουν. Η δομή του είναι έτσι φτιαγμένη ώστε να επιτρέπει την πρόσβαση και την αποθήκευση ενδιάμεσων καταστάσεων σε περιπτώσεις μακρών ακολουθιών. Πιο συγκεκριμένα, στην δική μας περίπτωση το κελί χρησιμοποιείται για την προσωρινή αποθήκευση συνδυασμών χαρακτήρων ενός domain name για την εξαγωγή χρήσιμων χαρακτηριστικών που βοηθούν στην κατηγοριοποίηση των ονομάτων σε legit ή DGA (binary classification) είτε στον αντίστοιχο DGA από τον οποίο παράχθηκαν (multiclass classification). Συνεπώς, γίνεται κατανοητό ότι ένα LSTM παρουσιάζει ευελιξία ως προς την ανίχνευση και εξαγωγή χρήσιμων συμπερασμάτων με βάση τις εξαρτήσεις των χαρακτήρων μιας ακολουθίας, εξού και το προτείνουμε για την ανίχνευση DGA ονομάτων τομέα. Προτού περάσουμε στην παρουσίαση της αρχιτεκτονικής του δικού μας LSTM, θα αναλύσουμε λίγο περαιτέρω τον τρόπο λειτουργίας των πυλών εστιάζοντας στην κάθε μία ξεχωριστά, όπου θα υπάρχει αντίστοιχη εικόνα για την καθεμία κάνοντας highlight τα αντίστοιχα στοιχεία.

Forget Gate

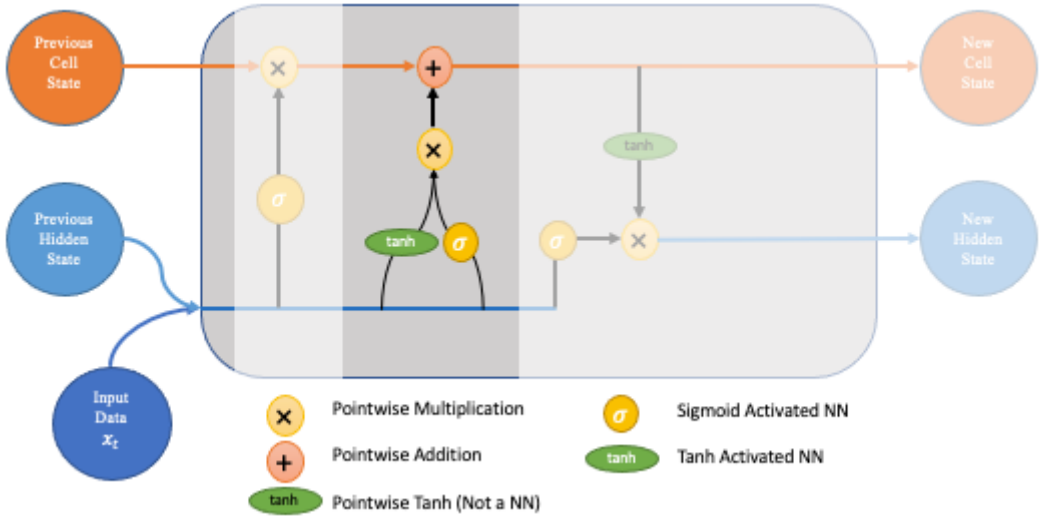


Εικόνα 19: Forget Gate

Το πρώτο βήμα στην διαδικασία υλοποίησης του LSTM cell είναι η πύλη forget. Σε αυτό το σημείο, το block αποφασίζει ποια bits του cell state (μακροπρόθεσμη μνήμη δικτύου) είναι χρήσιμα δεδομένης τόσο της προηγούμενης κρυφής κατάστασης όσο και των νέων δεδομένων εισόδου. Για να γίνει αυτό, η προηγούμενη κρυφή κατάσταση και τα νέα δεδομένα εισόδου τροφοδοτούνται σ' ένα νευρωνικό δίκτυο. Αυτό δημιουργεί ένα διάνυσμα όπου κάθε στοιχείο βρίσκεται στο διάστημα [0, 1], κάτι

το οποίο διασφαλίζεται από την sigmoid activation function. Αυτό το δίκτυο εντός του forget gate είναι εκπαιδευμένο έτσι ώστε το αποτέλεσμα να προσεγγίζει το 0 όταν ένα στοιχείο της εισόδου θεωρείται ασήμαντο και το 1 όταν είναι σχετικό και σημαντικό. Μπορούμε να σκεφτούμε κάθε στοιχείο αυτού του διανύσματος ως ένα φίλτρο που επιτρέπει περισσότερες πληροφορίες καθώς η τιμή πλησιάζει στο 1. Αυτές οι εξαγόμενες τιμές στη συνέχεια πολλαπλασιάζονται κατά σημείο (pointwise multiplication) με την προηγούμενη κατάσταση κελιού. Το αποτέλεσμα αυτής της πράξης επιβεβαιώνει πως οι συνιστώσες του cell state που έχουν θεωρηθεί ασήμαντο από το δίκτυο του forget gate θα πολλαπλασιαστούν μ' έναν αριθμό κοντά στο 0 κι έτσι θα έχουν μικρότερη επιρροή στα επόμενα βήματα. Συνοπτικά, η πύλη forget αποφασίζει ποια κομμάτια της μακροπρόθεσμης μνήμης θα πρέπει να έχουν μικρότερο βάρος, δεδομένης της προηγούμενης κρυφής κατάστασης και της νέας εισόδου δεδομένων στην ακολουθία.

Input Gate



Εικόνα 20: Input Gate

Το βήμα αυτό περιλαμβάνει το νέο δίκτυο μνήμης και την πύλη εισόδου (input gate). Στόχος αυτού είναι να καθορίσει ποιες νέες πληροφορίες πρέπει να προστεθούν στη μακροπρόθεσμη μνήμη του δικτύου (cell state), δεδομένης της προηγούμενης κρυφής κατάστασης και των νέων δεδομένων εισόδου. Τόσο το νέο δίκτυο μνήμης όσο και η πύλη εισόδου είναι νευρωνικά δίκτυα, τα οποία λαμβάνουν τις ίδιες εισόδους, την προηγούμενη κρυφή κατάσταση και τα νέα δεδομένα εισόδου. Ας δούμε τα τρία στάδια υλοποίησης αυτού:

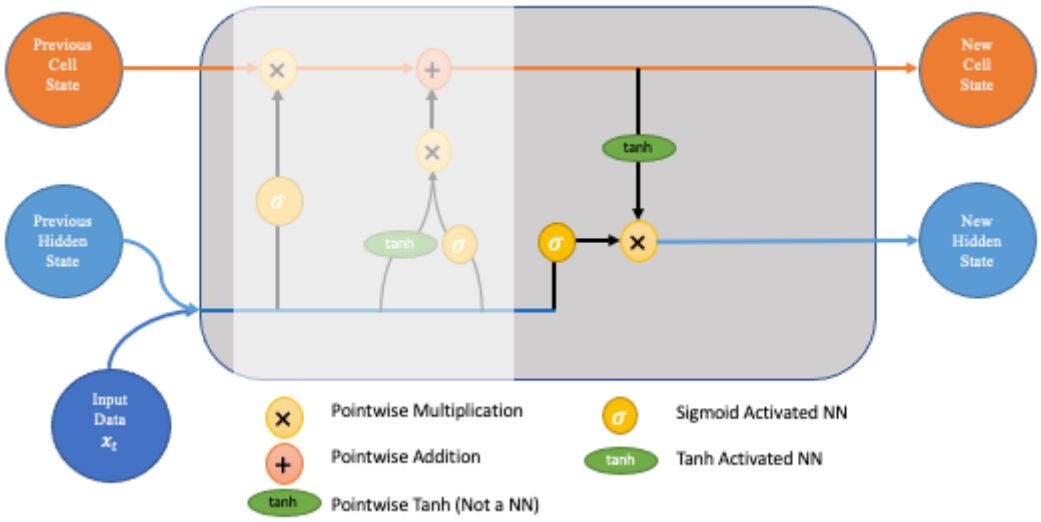
1. Το νέο δίκτυο μνήμης είναι ένα νευρωνικό δίκτυο με συνάρτηση ενεργοποίησης την tanh και έχει εκπαιδευτεί ώστε να συνδυάζει την προηγούμενη κρυφή κατάσταση με τα νέα δεδομένα εισόδου, ώστε να δημιουργεί ένα «διάνυσμα ενημέρωσης μνήμης». Ουσιαστικά, αυτό το διάνυσμα καθορίζει κατά πόσο θα ενημερωθεί κάθε στοιχείο της

μακροπρόθεσμης μνήμης (cell state) του δικτύου. Πρέπει να σημειώσουμε εδώ ότι χρησιμοποιούμε την tanh ως συνάρτηση ενεργοποίησης, επειδή δίνει τιμές στο [-1, 1] και έτσι μπορούμε να αξιοποιήσουμε τις αρνητικές. Αυτές είναι απαραίτητες εάν θέλουμε να μειώσουμε την επίδραση ενός στοιχείου στο cell state.

2. Ωστόσο, στο 1ο στάδιο παραπάνω, όπου δημιουργούμε το νέο διάνυσμα μνήμης, υπάρχει ένα μεγάλο πρόβλημα. Στην πραγματικότητα δεν ελέγχει αν αξίζει να θυμόμαστε τα νέα δεδομένα εισόδου. Εδώ μπαίνει η πύλη εισόδου. Αυτή αποτελεί ένα νευρωνικό δίκτυο με sigmoid activation function που λειτουργεί ως φίλτρο, προσδιορίζοντας ποια στοιχεία του «νέου διανύσματος μνήμης» αξίζει να διατηρηθούν. Αυτό το δίκτυο θα παράγει ένα διάνυσμα τιμών σε [0,1] (λόγω της σιγμοειδούς ενεργοποίησης), επιτρέποντάς του να λειτουργεί ως φίλτρο μέσω του σημειακού πολλαπλασιασμού. Παρόμοια με αυτό που είδαμε στην πύλη λήθης (forget gate), μια έξοδος κοντά στο μηδέν μας λέει ότι δεν θέλουμε να ενημερώσουμε αυτό το στοιχείο της κατάστασης κελιού.

3. Οι έξοδοι των σταδίων 1 και 2 πολλαπλασιάζονται σημειακά. Αυτό έχει ως αποτέλεσμα να ρυθμιστεί το μέγεθος των νέων πληροφοριών που αποφασίσαμε στο στάδιο 2 και να οριστεί στο 0 εάν χρειαστεί. Το συνδυασμένο διάνυσμα που προκύπτει προστίθεται στη συνέχεια στην κατάσταση κυψέλης, με αποτέλεσμα να ενημερώνεται η μακροπρόθεσμη μνήμη του δικτύου.

Output Gate



Εικόνα 21: Output Gate

Τώρα που ολοκληρώθηκαν οι ενημερώσεις στη μακροπρόθεσμη μνήμη του δικτύου, μπορούμε να προχωρήσουμε στο τελευταίο βήμα, την πύλη εξόδου (output gate), καθορίζοντας τη νέα κρυφή κατάσταση. Για να επιτευχθεί αυτό, θα χρειαστούμε

την πρόσφατα ενημερωμένη κατάσταση κελιού, την προηγούμενη κρυφή κατάσταση και τα νέα δεδομένα εισόδου. Στην πράξη, δημιουργούμε ένα φίλτρο, την πύλη εξόδου (output gate), ακριβώς όπως κάναμε στο δίκτυο της πύλης λήθης (forget gate). Οι είσοδοι είναι οι ίδιες (προηγούμενη κρυφή κατάσταση και νέα δεδομένα) και η ενεργοποίηση είναι επίσης σιγμοειδής (αφού θέλουμε κατ' αντιστοιχία οι τιμές των εξόδων να βρίσκονται στο [0,1]). Πιο συγκεκριμένα, θέλουμε να εφαρμόσουμε αυτό το φίλτρο στην πρόσφατα ενημερωμένη κατάσταση κελιού. Αυτό διασφαλίζει ότι εξάγονται μόνο οι απαραίτητες πληροφορίες (αποθηκευμένες στη νέα κρυφή κατάσταση). Ωστόσο, πριν εφαρμόσουμε το φίλτρο, εφαρμόζουμε την tanh στο cell state για να ορίσουμε τις τιμές στο [-1, 1].

Συνοπτικά, τα βήματα της διαδικασίας για την πύλη εξόδου είναι:

▹ Εφαρμόζουμε τη συνάρτηση tanh στην τρέχουσα κατάσταση κελιού για να λάβετε την συμπιεσμένη κατάσταση κελιού, η οποία βρίσκεται τώρα στο [-1,1].

▹ Τροφοδοτούμε την προηγούμενη κρυφή κατάσταση και τα τρέχοντα δεδομένα εισόδου σ' ένα δίκτυο με sigmoid activation function, ώστε να λάβουμε το διάνυσμα φιλτραρίσματος στο [0, 1].

▹ Εφαρμόζουμε αυτό το διάνυσμα φιλτραρίσματος στην κατάσταση συμπιεσμένου κελιού με πολλαπλασιασμό κατά σημείο.

▹ Εξάγουμε την νέα κρυφή κατάσταση

Ωστόσο, η έξοδος του LSTM block εξακολουθεί να είναι μια κρυφή κατάσταση. Και έτσι, για να μετατρέψουμε την κρυφή αυτή κατάσταση στην τελική έξοδο του νευρωνικού δικτύου, πρέπει στην πραγματικότητα να εφαρμόσουμε κάποιο γραμμικό layer μετά το LSTM block.

Πιο συγκεκριμένα, η υλοποίηση του δικού μας LSTM διαστρωματώνεται ως εξής (τα μεγέθη επιλέχθηκαν από μελέτη της βιβλιογραφίας με βάση το rule of thumb που χρησιμοποιείται):

- Embedding Layer
- LSTM Layer, με 128 blocks
- Dropout (για την αποφυγή overfitting)
- Dense Layer, με 32 νευρώνες
- Dropout (για την αποφυγή overfitting)
- Output Layer, με 2 νευρώνες (για το binary classification – sigmoid activation) και 11 νευρώνες (για το multiclass classification – softmax activation)
- Ο optimizer που χρησιμοποιήθηκε είναι ο rmsprop

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Input(shape=[length]))
model.add(tf.keras.layers.Embedding(input_dim=(len(vocab) ** n) + 1, output_dim=4)) # N x LEN -> N x LEN x 4
model.add(tf.keras.layers.Reshape([1, -1])) # N x LEN x 4 -> N x 1 x LEN*4
model.add(tf.keras.layers.LSTM(units=128, activation="relu"))
model.add(tf.keras.layers.Dropout(rate=0.5))
model.add(tf.keras.layers.Dense(units=32, activation="relu"))
model.add(tf.keras.layers.Dropout(rate=0.5))
model.add(tf.keras.layers.Dense(units=2, activation="sigmoid"))
model.compile(loss=tf.losses.BinaryCrossentropy(), optimizer="rmsprop", metrics=[tf.metrics.BinaryAccuracy()])
model.summary()
```

Εικόνα 22: LSTM (Binary Classification)

```
model = tf.keras.Sequential()
model.add(tf.keras.layers.Input(shape=[length]))
model.add(tf.keras.layers.Embedding(input_dim=(len(vocab) ** n) + 1, output_dim=4)) # N x LEN -> N x LEN x 4
model.add(tf.keras.layers.Reshape([1, -1])) # N x LEN x 4 -> N x 1 x LEN*4
model.add(tf.keras.layers.LSTM(units=128, activation="relu"))
model.add(tf.keras.layers.Dropout(rate=0.5))
model.add(tf.keras.layers.Dense(units=32, activation="relu"))
model.add(tf.keras.layers.Dropout(rate=0.5))
model.add(tf.keras.layers.Dense(units=output_length, activation="softmax"))
model.compile(loss=tf.losses.CategoricalCrossentropy(), optimizer="rmsprop", metrics=[tf.metrics.CategoricalAccuracy()])
model.summary()
```

Εικόνα 23: LSTM Multiclass Classification

Το Embedding Layer χρησιμοποιείται για την χαρτογράφηση των n-grams που προκύπτουν από τον συνδυασμό των 40 επιτρεπόμενων συμβόλων του κάθε δειγματικού στοιχείου, σ΄ ένα διάνυσμα 4 διαστάσεων, το οποίο γίνεται flatten σ’ ένα διάνυσμα 1024 στοιχείων (256 * 4 = 1024), όπου 256 είναι το μέγεθος της εισόδου, το οποίο καθορίζεται με το μέγιστο επιτρεπόμενο πλήθος χαρακτήρων ενός domain name. Σημειώνουμε ότι τα 40 σύμβολα που χρησιμοποιούνται, αποτελούνται από 26 πεζά γράμματα του λατινικού αλφαβήτου, τα 10 αριθμητικά ψηφία (0-9), την παύλα (hyphen), την κάτω παύλα, την τελεία (“.”) κι ένα που αντιστοιχεί στο padding, το οποίο προσθέτει όπου χρειάζεται τον κενό χαρακτήρα για να έχουν όλες οι είσοδοι το ίδιο μήκος (256). Το δεύτερο επίπεδο του δικτύου, δηλαδή το LSTM Layer αποτελείται από 128 LSTM blocks, τα οποία λαμβάνουν ως είσοδο τις ακολουθίες που παράγει το embedding layer και καθένα από αυτά παράγει ένα χαρακτηριστικό. Μετά το LSTM layer, ακολουθεί ένα επίπεδο dropout για την αποφυγή του overfitting.

Προτού περάσουμε στην πειραματική αξιολόγηση, ας υπενθυμίσουμε ότι το SHAP είναι ένα post-hoc framework, δηλαδή εφαρμόζεται μετά την εκπαίδευση του νευρωνικού δικτύου. Στην πράξη χρησιμοποιούμε το trained model με επιλεγμένα τυχαία 1000 δειγματικά στοιχεία του, κι αυτό το σύνολο φορτώνεται στον DeepExplainer του SHAP που παράγει τα SHAP values. Περισσότερες πληροφορίες είναι διαθέσιμες στο documentation του shap.DeepExplainer [61].

Κεφάλαιο 5 – Πειραματική Αξιολόγηση

Σε αυτό το κεφάλαιο θα εξεταστεί η αποτελεσματικότητα των παραπάνω μοντέλων που περιγράψαμε για την ανίχνευση DGA domain names και την σωστή κατηγοριοποίηση τους. Κατόπιν, απώτερος στόχος είναι η δυνατότητα ερμηνείας των προβλέψεων των μοντέλων μας, σχολιάζοντας τα αποτελέσματα μας καθώς και πιθανώς περιορισμούς που προκύπτουν.

5.1 Κριτήρια Αξιολόγησης των Μοντέλων Μάθησης

Σε πρώτη φάση πρέπει να ορίσουμε τις μετρικές με τις οποίες αξιολογούνται τα μοντέλα μάθησης, οι οποίες είναι πολύ συνήθεις στον τομέα της Μηχανικής Μάθησης. Αναλυτικότερα, αυτές είναι:

1. **Accuracy:** Αποτελεί την πιο κοινή μετρική απόδοσης και εκφράζει την αναλογία των σωστών προβλέψεων προς τον συνολικό αριθμό προβλέψεων.

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

2. **Precision:** Εκφράζει την αναλογία των σωστών θετικών προβλέψεων προς το σύνολο των θετικών προβλέψεων.

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

3. **Recall:** Εκφράζει την αναλογία σωστών θετικών προβλέψεων προς τις συνολικές προβλέψεις που αφορούν δεδομένα της συγκεκριμένης κλάσης.

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

4. **F1-Score:** Αποτελεί τον σταθμισμένο μέσο όρο του Precision και Recall. Συνεπώς, δίνει έμφαση στις ψευδώς αρνητικές και ψευδώς θετικές προβλέψεις και αποτελεί σημαντική μετρική σε μη ισορροπημένα dataset.

$$F1 - Score = \frac{2\ x\ (Recall\ x\ Precision)}{Recall\ +\ Precision}$$

Για την μελέτη των δικών μας πειραμάτων θα σχολιάζουμε πάντα την κλάση των κακόβουλων domain names που έχουν παραχθεί από DGA. Άρα, ως True Positive (TP) θα χαρακτηρίζουμε την επιτυχή πρόβλεψη ενός Domain Name παραγόμενου από DGA, ενώ ως True Negative, την επιτυχή πρόβλεψη ενός Domain Name ως έγκυρου.

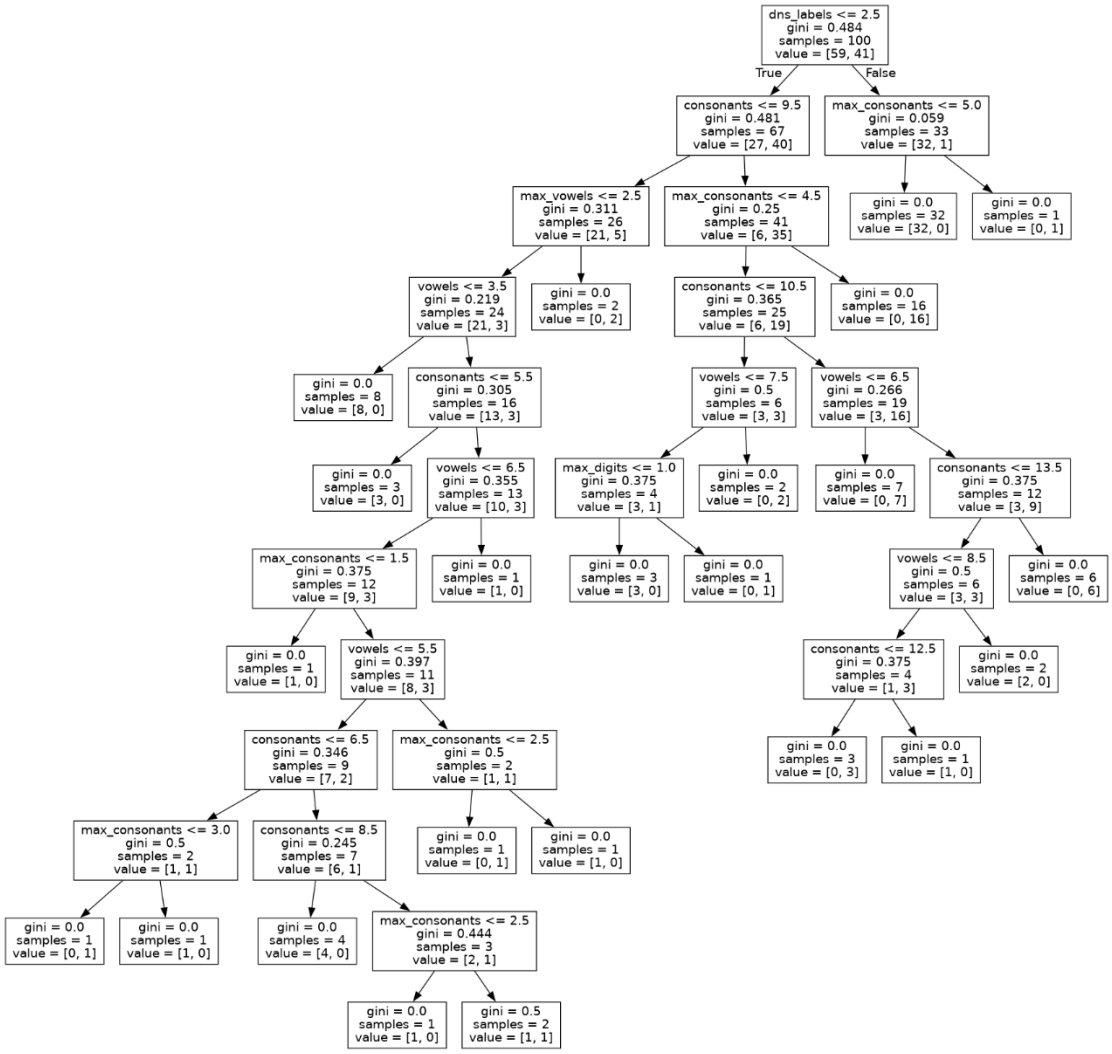
5.2 Decision Tree, Intrinsically-Explainable – Binary Classification

Σε πρώτη φάση, για να έχουμε και κάποιο benchmark για το explainability που μπορεί να προσφέρει ένας XAI αλγόριθμος και πιο συγκεκριμένα το SHAP θα ξεκινήσουμε από την επεξηγησιμότητα βάσει των 13 χαρακτηριστικών που αναφέραμε και παραπάνω με την χρήση Δέντρου Αποφάσεων. Το Decision Tree όπως φαίνεται στην παρακάτω εικόνα επιτυγχάνει accuracy, precision, recall και f1-score γύρω στο 93%. Όσον αφορά την ακρίβεια, είναι αρκετά καλό αποτέλεσμα, ωστόσο σε τόσο κρίσιμα προβλήματα η ύπαρξη ψευδών προβλέψεων από το μοντέλο είναι κρίσιμες, καθώς μπορεί είτε να διαφύγει στο δίκτυο κακόβουλη κίνηση (false negative) που είναι ζήτημα ασφαλείας, είτε να προκληθεί πρόβλημα λειτουργικότητας με την αποκοπή ενός έγκυρου ονόματος τομέα (false positive). Αυτή η αδυναμία θα επιλυθεί παρακάτω με την χρήση LSTM.

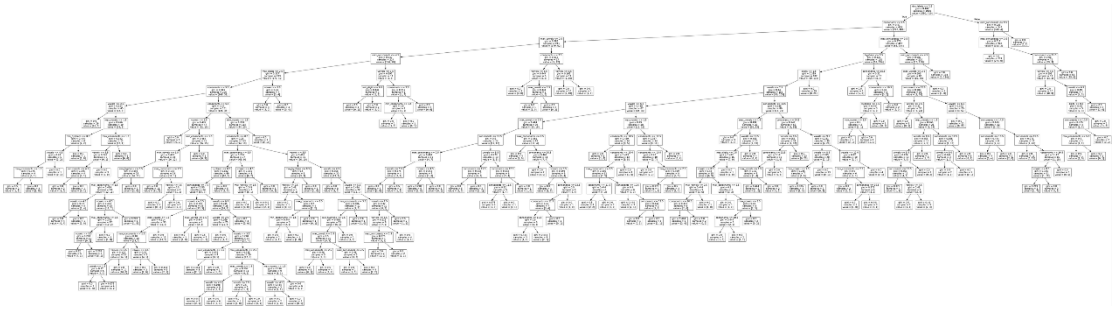
	precision	recall	f1-score	support
0	0.94	0.94	0.94	307360
1	0.90	0.91	0.91	203567
accuracy			0.93	510927
macro avg	0.92	0.92	0.92	510927
weighted avg	0.93	0.93	0.93	510927

Εικόνα 24:Decision Tree Metrics

Τα δέντρα αποφάσεων, όπως έχουμε αναλύσει και παραπάνω, αδυνατούν στην τοπική εξήγηση της συνεισφοράς ενός χαρακτηριστικού, καθώς αν εστιάσουμε σ’ ένα συγκεκριμένο υποδέντρο, δεν είναι εύκολο να αναγνωρίσουμε τους κανόνες από τους οποίους έχει προέλθει. Επιπρόσθετα, παρότι είναι εγγενώς ερμηνεύσιμα, με την αύξηση της πολυπλοκότητας και του μεγέθους των δεδομένων καθίσταται αδύνατη τόσο η μελέτη όσο και η ερμηνεία τους. Πιο συγκεκριμένα, χρησιμοποιήσαμε 100 και 1000 δειγματικά στοιχεία αντίστοιχα, για να παρουσιάσουμε τα αποτελέσματα που προκύπτουν από το Δέντρο Απόφασης.



Εικόνα 25: Decision Tree with 100 instances



Εικόνα 26:Decision Tree with 1000 instances

Στην πρώτη περίπτωση με τα 100 στοιχεία προκύπτει ένα σχετικά επεξηγήσιμο θα λέγαμε μοντέλο, αλλά θεωρούμε μικρό τον αριθμό των στοιχείων που παρουσιάζονται για να αναδείξουμε σημαντική πληροφορία. Όσο για την δεύτερη περίπτωση κατανοούμε ότι είναι αδύνατο οποιοσδήποτε άνθρωπος να μελετήσει το συγκεκριμένο δέντρο και να βγάλει κάποιο συμπέρασμα, όσο κι αν κάνει "zoom" στην εικόνα. Συνεπώς, στην δεύτερη περίπτωση υπάρχει πλήρης αδυναμία κατανόησης και εξήγησης του μοντέλου.

5.3 MLP, SHAP Explainability – Binary Classification

Παρατηρώντας τις αδυναμίες του Decision Tree, αποφασίσαμε να στραφούμε στην βαθιά επιβλεπόμενη μηχανική μάθηση (supervised deep learning) αρχικά με την χρήση ενός Multilayer Perceptron. Αποτελεί ένα από τα βασικά μοντέλα που χρησιμοποιούνται σε περιπτώσεις deep learning κι έτσι επιλέξαμε να το εκπαιδεύσουμε με τα ίδια 13 χαρακτηριστικά, που επιλέξαμε για το Decision Tree. Ας τα υπενθυμίσουμε σε αυτό το σημείο:

1. Πλήθος DNS Labels (τμήματα domain name)
2. Πλήθος συμφώνων
3. Μέγιστη ακολουθία συμφώνων
4. Ελάχιστη ακολουθία συμφώνων
5. Πλήθος φωνηέντων
6. Μέγιστη ακολουθία φωνηέντων
7. Ελάχιστη ακολουθία φωνηέντων
8. Πλήθος ψηφίων
9. Μέγιστη ακολουθία ψηφίων
10. Ελάχιστη ακολουθία ψηφίων
11. Πλήθος hyphens (“-“)
12. Μέγιστη ακολουθία hyphens
13. Ελάχιστη ακολουθία hyphens

Παρατηρούμε παρεμφερή απόδοση του μοντέλου όσον αφορά τα κριτήρια της απόδοσης, που εδώ βρίσκονται γύρω στο 90% με σαφή την ύπαρξη False Negative και False Positive περιπτώσεων, που θα αναλύσουμε και παρακάτω μελετώντας τα διαγράμματα. Στόχος μας είναι να διερευνήσουμε την ευρύτερη δυνατότητα ερμηνείας που παρέχεται και με την βοήθεια τους SHAP framework. Όλα τα διαγράμματα που θα μελετήσουμε παρακάτω, τόσο για την υλοποίηση με MLP, όσο και τις υλοποιήσεις με LSTM αφορούν τα SHAP values για την κλάση των malicious domain names που παράγονται από DGAs. Προτού περάσουμε στα γραφήματα, είναι σημαντικό να σημειωθεί ότι σε αυτό το σημείο αρχίσαμε να έχουμε τον προβληματισμό, ότι ένα legit domain name κι ένα malicious θα μπορούσαν να έχουν τα ίδια χαρακτηριστικά. Για παράδειγμα το google.com γνωρίζουμε ότι είναι legit domain name, θα μπορούσε με κάποιο permutation να προκύψει το ooggle.com, το οποίο έχει σχεδόν ίδια χαρακτηριστικά.

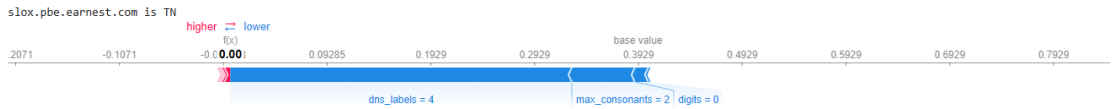
	precision	recall	f1-score	support
0	0.92	0.91	0.91	306952
1	0.86	0.88	0.87	203975
accuracy			0.90	510927
macro avg	0.89	0.89	0.89	510927
weighted avg	0.90	0.90	0.90	510927

Εικόνα 27: MLP metrics

Για την δυνατότητα ερμηνείας με την βοήθεια των SHAP values θα παρουσιάσουμε αρχικά την τοπική επεξηγησιμότητα (local) και ακολούθως την καθολική (global) με τ' αντίστοιχα διαγράμματα. Για την πρώτη περίπτωση επιλέγουμε 4 domain names, ένα από κάθε κατηγορία True Positive/Negative (TP / TN) και False Positive/Negative (FP / FN). Υπενθυμίζουμε ότι το local explainability αναφέρεται στην εξήγηση κάθε μεμονωμένης πρόβλεψης. Κάθε SHAP value ενός feature δρα ως μια «δύναμη» που είτε αυξάνει είτε μειώνει την έξοδο του μοντέλου προσεγγίζοντας την πιθανότητα επιλογής της συγκεκριμένης κλάσης. Αυτό επιτυγχάνεται με την χρήση των Force Plots, όπως ονομάζονται, τα οποία παρέχονται από την βιβλιοθήκη του SHAP [57].

Εικόνα 28: MLP - True Positive Instance

Για να κατανοήσουμε λίγο καλύτερα τι δείχνει το force plot, στο σχήμα τα κόκκινα features ωθούν την πρόβλεψη του μοντέλου προς υψηλότερες τιμές, δηλαδή προς την επιλεγμένη κλάση (εδώ malicious DGA domain names), ενώ τα μπλε απωθούν αντίστοιχα την πρόβλεψη. Αυτές οι δύο δυνάμεις ισορροπούν τελικά στην εκτιμώμενη αποτίμηση του instance, σύμφωνα με την τελική έξοδο του μοντέλου. Το domain name που επιλέχθηκε στην εικόνα 28 είναι το «wshnbyplaywobb.com», το οποίο ταξινομήθηκε επιτυχώς από το MLP ως DGA, αποτελώντας μια True Positive περίπτωση. Ιδιαίτερα χαρακτηριστικά που είχαν την μεγαλύτερη συνεισφορά αποτελούν η μικρή τιμή του DNS Labels και της μέγιστης ακολουθίας των συμφώνων, κάτι αναμενόμενο καθώς δύσκολα υπαρκτές λέξεις έχουν 5 σύμφωνα στην σειρά, όπως έχει το συγκεκριμένο domain. Αυτά μας δίνουν μια πρώτη εικόνα για το τι αναμένουμε και στην συνέχεια να δούμε στο δείγμα μας για τα malicious domain names από DGA.

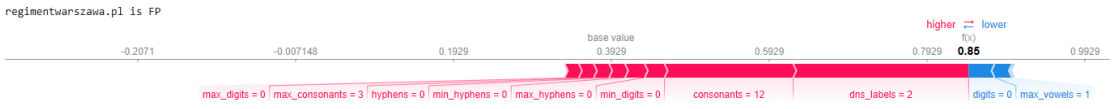


Εικόνα 29: MLP - True Negative Instance

Στην παραπάνω εικόνα 29, έχει επιλεχθεί το domain name «slox.pbe.earnest.com», το οποίο επιτυχώς κατηγοριοποιήθηκε από το MLP ως έγκυρο όνομα τομέα (True Negative). Εδώ, παρατηρούμε ότι υπερτερούν οι μπλε δυνάμεις και το μοντέλο μας απωθείται από την επιλογή της κλάσης των DGA domain names. Ιδιαίτερα χαρακτηριστικά που είχαν την μεγαλύτερη συνεισφορά αποτελούν η μεγαλύτερη τιμή του DNS Labels συγκριτικά με την προηγούμενη περίπτωση και της μικρής τιμής της μέγιστης ακολουθίας των συμφώνων. Αυτά μας δίνουν μια πρώτη εικόνα για το τι αναμένουμε και στην συνέχεια να δούμε στο δείγμα μας για τα legit domain names. Οι παραπάνω περιπτώσεις ήταν οι θεμιτές ως τώρα και μας δώσουν μια επαρκής αρχικά εικόνα σε τοπικό επίπεδο, ωστόσο η βασική παθογένεια της συγκεκριμένης υλοποίησης όπως γίνεται αντιληπτό είναι οι False Negative και οι False Positive περιπτώσεις.

Εικόνα 30: MLP - False Negative Instance

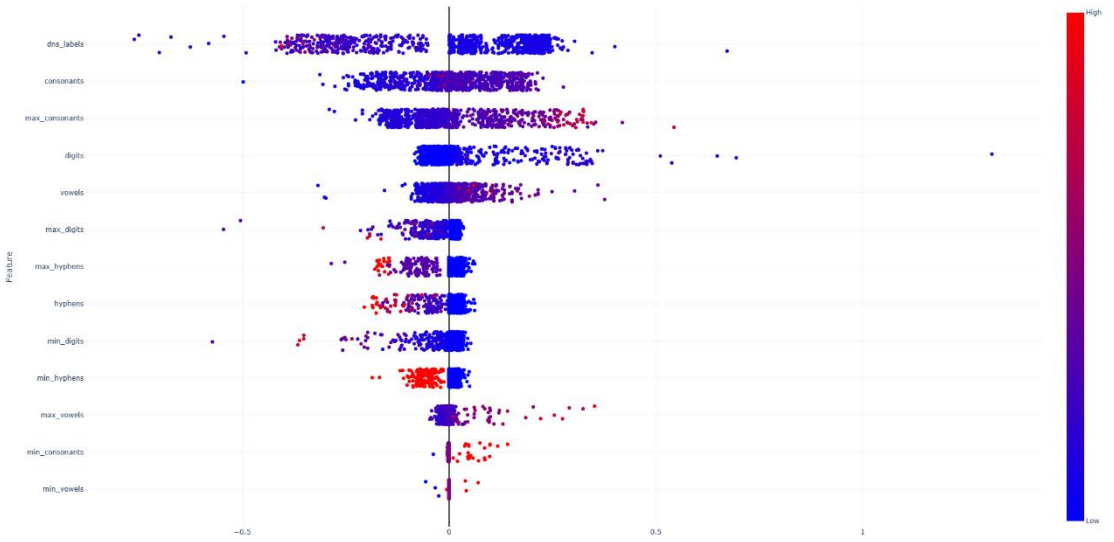
Στην εικόνα 30, έχουμε το domain name «ioblocajanunal.com», για το οποίο προκύπτει λανθασμένη κατηγοριοποίηση από το MLP ως legit domain name (False Negative). Παρατηρούμε μια ισορροπία μεταξύ των δυνάμεων στο σχήμα με το μοντέλο οριακά να επιλέγει να απομακρυνθεί από την κλάση των malicious domain names, καθώς η έξοδος του δικτύου είναι 0.42, όπως φαίνεται στο σχήμα. Αυτό σημαίνει ότι προσεγγιστικά κατατάσσει με πιθανότητα 42% το συγκεκριμένο instance στην κλάση των DGA domain names, κι εφόσον το πρόβλημα είναι δυϊκό, όποια κατηγορία έχει πιθανότητα να επιλεχθεί άνω του 50%, αυτή και επιλέγεται. Πιο συγκεκριμένα, παρατηρούμε την «σύγκρουση» των 2 χαρακτηριστικών που είχαν επικρατήσει ως τώρα, δηλαδή του πλήθους των DNS Labels και της μέγιστης ακολουθίας συμφώνων. Αυτό λοιπόν, είναι κάτι που μας θορυβεί για πιθανές παθογένειες της προσέγγισης με features, καθώς δεν αξιοποιείται η αλληλουχία εμφάνισης των χαρακτήρων. Οι False Negative περιπτώσεις είναι κατά την γνώμη μας οι πιο επικίνδυνες καθώς οδηγούν στο resolution του συγκεκριμένου domain name στο δίκτυο, παρότι είναι malicious, και εκθέτει το δίκτυο σε πιθανές κακόβουλες δραστηριότητες όπως αναλύσαμε στο κεφάλαιο 2 στο θεωρητικό υπόβαθρο. Αντίθετα, προκύπτουν και False Positive προβλέψεις από το μοντέλο μας.



Εικόνα 31: MLP - False Positive Instance

Μια τέτοια περίπτωση είναι το domain name «regimentwarszawa.pl», το οποίο ταξινομείται λανθασμένα από το MLP ως DGA (False Positive). Αναλυτικότερα, εδώ δεν προκύπτει ζήτημα ασφάλειας του δικτύου, αλλά ζήτημα λειτουργικότητας. Σε γενικές γραμμές, δίνουμε προτεραιότητα στην ασφάλεια έναντι της λειτουργικότητας, ωστόσο οφείλουμε να σημειώσουμε την ύπαρξη αυτής της περίπτωσης. Για το συγκεκριμένο παράδειγμα, το μικρό πλήθος των DNS Labels και τα πολλά σύμφωνα μπέρδεψαν το μοντέλο και το οδήγησαν σε εσφαλμένη πρόβλεψη ως DGA domain name. Αν παρατηρήσουμε λίγο καλύτερα το TLD (Top-Level-Domain), αντιλαμβανόμαστε ότι το domain name έχει γίνει register στην Πολωνία (.pl), όπου είναι χαρακτηριστικό της γλώσσας της οι λέξεις πυκνές σε σύμφωνα.

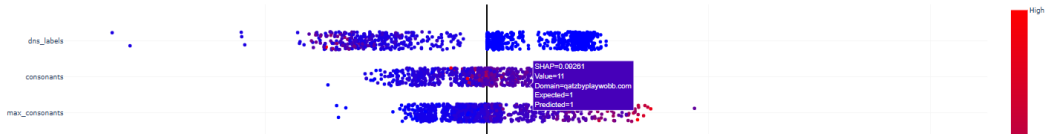
Έχοντας αναλύσει διάφορα παραδείγματα σε τοπικό επίπεδο, είναι το κατάλληλο σημείο να συνθέσουμε όλη αυτή την πληροφορία και να δούμε τι γίνεται σε συνολικότερο επίπεδο. Το global explainability περιγράφει την επίδραση των χαρακτηριστικών στο μοντέλο ελέγχοντας ένα μέρος του συνόλου των δεδομένων και όχι μεμονωμένα δειγματικά στοιχεία. Για την επίτευξη αυτού, το SHAP framework μας παρέχει το Summary Plot.



Εικόνα 32: MLP - Global Explainability

Προτού περάσουμε στην ανάλυση του γραφήματος, ας εξηγήσουμε λίγο τι μας δείχνει το summary plot. Κάθε SHAP value ενός feature αποτελεί μια κουκκίδα στο σχήμα. Στον y-άξονα βρίσκονται τα χαρακτηριστικά που χρησιμοποιήθηκαν και στον x-άξονα οι αντίστοιχες SHAP τιμές τους. Το χρώμα αναπαριστά την τιμή του feature, από το χαμηλό (μπλε) στο υψηλό (κόκκινο). Συνεπώς, έτσι αποκτούμε αίσθηση της κατανομής των SHAP values για κάθε feature. Τα χαρακτηριστικά παρουσιάζονται με σειρά σημαντικότητας όπως προκύπτει από τις SHAP τιμές, αναλόγως αν και πόσο επηρέασαν κάθε πρόβλεψη. Αναλυτικότερα, αναλύσαμε συνολικά 2.5 M δειγματικά στοιχεία, εκ των οποίων παρουσιάζουμε 1000 τυχαία επιλεγμένα. Αρχικά, επιβεβαιώνεται στο σύνολο του δείγματος, αυτό που παρατηρήσαμε για τα DNS Labels

και τα σύμφωνα. Δηλαδή, μικρότερες τιμές (μπλε ανοιχτό) των DNS Labels και μεγάλες τιμές (κόκκινο) στο πλήθος συμφώνων καθώς και στην μέγιστη ακολουθία αυτών οδηγούν το μοντέλο μας να γείρει την πλάστιγγα της εκτίμησης υπέρ του malicious domain name παραγόμενο από DGA. Ένα ιδιαίτερο χαρακτηριστικό που προκύπτει από το summary plot και μπορεί να μας βοηθήσει είναι το πλήθος των hyphens, όπου παρατηρούμε πως η έλλειψη τους (τιμή 0), οδηγεί σε κακόβουλα domain names ενώ οι υψηλότερες τιμές σε legit. Το αντίστροφο, παρατηρούμε όσον αφορά τα ψηφία, καθώς η έλλειψη τους απωθεί το μοντέλο (αρνητικές τιμές SHAP) από την κλάση των αλγοριθμικά παραγόμενων ονομάτων τομέα.



Εικόνα 33: Zooming in MLP - Global Explainability

Κάνοντας λίγο zoom στην εικόνα του summary plot, βλέπουμε ότι το διάγραμμα είναι interactive (με την βιβλιοθήκη plotly [62]) και δίνει όλες τις πληροφορίες που χρειάζεται ο ερευνητής – αναλυτής δικτύου, ώστε να μελετήσει πιθανές ανωμαλίες. Αυτές αναλυτικότερα είναι η τιμή SHAP, η τιμή του χαρακτηριστικού, το domain name, η expected κατηγοριοποίηση και η predicted ταξινόμηση.

5.4 LSTM, SHAP Explainability – Binary Classification

Παρότι, η υλοποίηση του προβλήματος με MLP μας έδωσε μια αρκετά καλή εικόνα για την ερμηνεία του μοντέλου, έκρυβε παθογένειες που μπορούν να προκαλέσουν ζητήματα ασφάλειας ή λειτουργικότητας. Στην προσπάθεια να βελτιώσουμε τις τιμές των κριτηρίων αξιολόγησης που θέσαμε, επιλέξαμε να πάμε σε πιο περίπλοκα μοντέλα βαθιάς μηχανικής μάθησης και συγκεκριμένα το Long Short-Term Memory δίκτυο, για να αξιοποιήσουμε την σειρά των χαρακτήρων ενός Domain Name. Υπενθυμίζουμε εδώ, ότι τα χαρακτηριστικά πλέον είναι τα overlapping bigrams και trigrams. (π.χ. google.com -> “go”, “oo”, “og”, “gl”, “le”, “e.”, “.c”, “co”, “om”). Με βάση τις μετρικές αξιολόγησης του LSTM παρατηρούμε εξαιρετική ακρίβεια της τάξης του 99%, ωστόσο η μεγάλη επιτυχία εδώ είναι η ελαχιστοποίηση των False Positive και False Negative προβλέψεων, όπως μπορούμε να αντιληφθούμε από τις μετρικές Precision και Recall, αντίστοιχα.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	307374
1	0.99	0.98	0.98	203553
accuracy			0.99	510927
macro avg	0.99	0.99	0.99	510927
weighted avg	0.99	0.99	0.99	510927

Εικόνα 34: LSTM metrics

Πρέπει να σημειωθεί σε αυτό το σημείο, ότι πρέπει να ορίσουμε έναν τρόπο επιλογής των σημαντικών n-grams, και πιο συγκεκριμένα trigrams, καθώς αυτά θα παρουσιαστούν στην παρούσα εργασία. Δεδομένου ότι τα feature μας είναι όλα τα πιθανά trigrams, δεν είναι δυνατό να παρουσιαστούν όλα σ' ένα διάγραμμα και γι' αυτό το λόγο επιλέγονται τα 30 σημαντικότερα, με βάση 1000 τυχαία επιλεγμένα δειγματικά στοιχεία. Η επιλογή γίνεται με 2 τρόπους:

- ✓ **Sum of Absolute SHAP Values**, σε αυτή την περίπτωση θέλουμε να επιλέξουμε τα n-grams που συνολικά παίζουν σημαντικό ρόλο με βάση την SHAP τιμή τους, εξού και κρίνουμε την σημαντικότητα του καθενός αθροίζοντας κατ' απόλυτη τιμή τα SHAP Values. Συνεπώς είτε θετικά είτε αρνητικά επιλέγονται αυτά με την μεγαλύτερη επιρροή προς οποιαδήποτε κατεύθυνση.
- ✓ **Sum of SHAP Values**, σε αυτή την περίπτωση θέλουμε να επιλέξουμε τα n-grams που ενισχύουν την πρόβλεψη του μοντέλου μας προς μια κλάση (εδώ malicious domain names από DGA) - τα πιο «θετικά» ως επιρροή. Συνεπώς, θέλουμε να φιλτράρουμε τα n-grams με τις υψηλότερες SHAP τιμές, εξού και επιλέγουμε αυτά με το μεγαλύτερο άθροισμα στο σύνολο του dataset. Τοιουτοτρόπως, ξεχωρίζουν αυτά που ωθούν τις προβλέψεις προς την κλάση των DGA domain names.

Ο παραπάνω διαχωρισμός είναι απαραίτητος να γίνει καθώς εδώ θα παρουσιάσουμε μόνο το global explainability. Λόγω του ότι πλέον βασιζόμαστε σε n-grams, συνεπώς σε ακολουθίες χαρακτήρων, τα force plots που παρέχουν local explainability δεν μπορούν να μας βοηθήσουν ιδιαίτερα διότι οι συνδυασμοί των n-grams είναι της τάξης των χιλιάδων, σε αντίθεση με το περιορισμένο πλήθος των features στο MLP, οπότε κάθε φορά θα παρατηρούμε κάτι εντελώς διαφορετικό και δεν θα υπάρχει μέτρο σύγκρισης μεταξύ των δειγματικών στοιχείων. Στο summary plot, ο χρωματικός κώδικας low-high (μπλε-κόκκινο), θα αναφέρεται πλέον στην τιμή SHAP του εκάστοτε n-gram, καθώς δεν μπορούμε να αποδώσουμε κάποιο άλλο value σε αυτή την περίπτωση. Κλείνοντας τα εισαγωγικά στοιχεία για τα διαγράμματα, αν κάποιο n-gram εμφανιστεί πολλές φορές σ' ένα instance, κρατούμε την μέση τιμή των SHAP values.

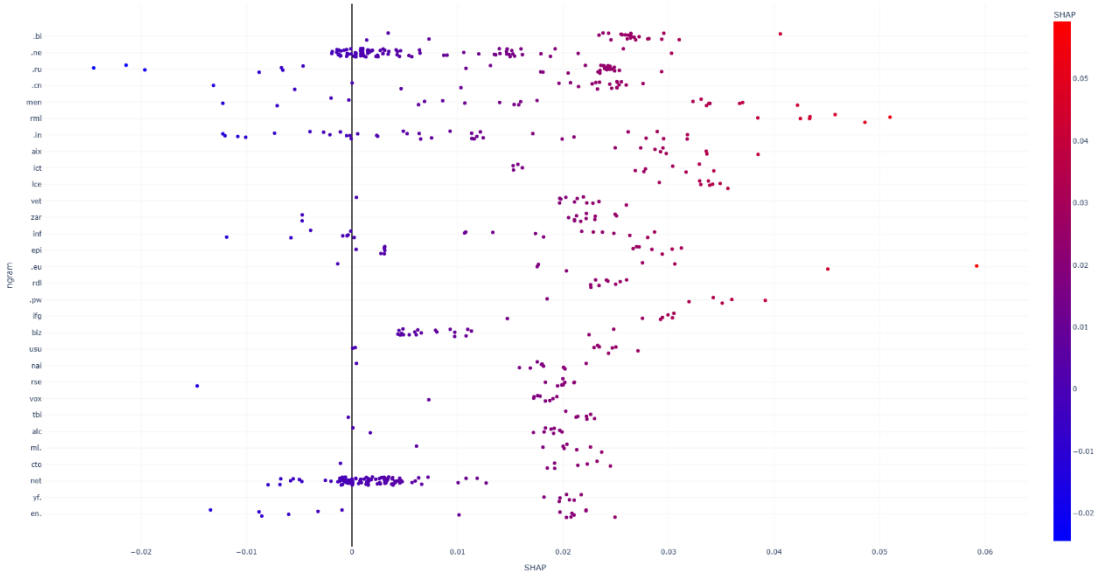
Εικόνα 35: LSTM - Global Explainability (Sum of absolute SHAP Values)

Ας υπενθυμίσουμε σε αυτό το σημείο, ότι ένα top-level domain είναι το τελευταίο κομμάτι ενός ονόματος τομέα στο Διαδίκτυο. Για παράδειγμα στο domain wikipedia.org το top-level domain είναι το .org. Αρχικά, για το διάγραμμα της εικόνας 35, ως κριτήριο επιλογής για την σημαντικότητα των n-grams χρησιμοποιούμε το sum of absolute SHAP values. Ουσιαστικά, επιλέγονται ως επικρατέστερα τα n-grams όπου τα SHAP values τους είναι μεγαλύτερα κατ' απόλυτη τιμή. Δηλαδή, είναι αυτά τα οποία βοήθησαν το μοντέλο είτε να οδηγηθεί σημαντικά στην εκτίμηση του για malicious domain name παραγόμενο από DGA είτε και να απομακρυνθεί από αυτή. Αυτό που αναμένουμε στην συγκεκριμένη περίπτωση είναι να δούμε n-grams:

- ✓ όπως το trigram www, που είναι το συνηθέστερο prefix
- ✓ από πολύ common Top-Level Domains (TLDs) όπως .co, com
- ✓ από TLDs που συχνά είναι πιο εύκολο να γίνει register ένα malicious domain name λόγω πιο «χαλαρών» ελέγχων και συστημάτων ασφαλείας όπως .ru (Ρωσία), .cn (Κίνα), .bi (Burundi), .ne (Νιγηρία)
- ✓ και τέλος, διάφορα n-grams που ανακάλυψε το μοντέλο μας και ανάλογα με την συνεισφορά τους μπορούν να σημειωθούν για την χρήση τους σε κάποια μέθοδο φιλτραρίσματος στην κίνηση ενός δικτύου

Από την παραπάνω προσέγγιση, αποκτούμε μια γενικότερη αίσθηση για την επιρροή των n-grams. Δεν πρέπει να ξεχνάμε όμως, ότι το πρόβλημα που μελετάμε είναι η ανίχνευση κακόβουλης δικτυακής κίνησης που έχει παραχθεί από Domain

Generation Algorithms. Συνεπώς, ο στόχος μας είναι να φιλτράρουμε τα trigrams που μας καθοδηγούν σε αυτά τα domain names.



Εικόνα 36: LSTM - Global Explainability (Sum of SHAP Values)

Υπενθυμίζουμε ότι το κριτήριο Sum of Absolute SHAP Values, αξιολογεί γενικά την συνεισφορά ενός χαρακτηριστικού (εδώ trigram), ενώ το κριτήριο Sum of SHAP Values αξιολογεί την συνεισφορά του χαρακτηριστικού προς μια κλάση (εδώ DGA domain names). Για το διάγραμμα της εικόνας 36, ως κριτήριο επιλογής για την σημαντικότητα των n-grams χρησιμοποιούμε το sum of SHAP values. Ουσιαστικά, επιλέγονται ως επικρατέστερα τα n-grams όπου τα SHAP values τους είναι μεγαλύτερα, δηλαδή αυτά που έχουν τις θετικότερες. Συνεπώς, είναι αυτά τα οποία βοήθησαν το μοντέλο να οδηγηθεί σημαντικά προς την υπόδειξη ενός instance ως malicious domain name προερχόμενο από DGA. Αυτό που αναμένουμε στην συγκεκριμένη περίπτωση είναι να δούμε n-grams:

- ✓ από TLDs που συχνά είναι πιο εύκολο να γίνει register ένα malicious domain name λόγω πιο «χαλαρών» ελέγχων και συστημάτων ασφαλείας όπως .ru και .cn, αντίστοιχα με την πρώτη περίπτωση, ωστόσο εδώ μπορούμε να παρατηρήσουμε περισσότερα τέτοια TLDs, όπως το .bi, το .ne και το .in, τα οποία όλα αντιστοιχούν σε χώρες με πιθανώς πιο ασθενή κριτήρια ασφαλείας

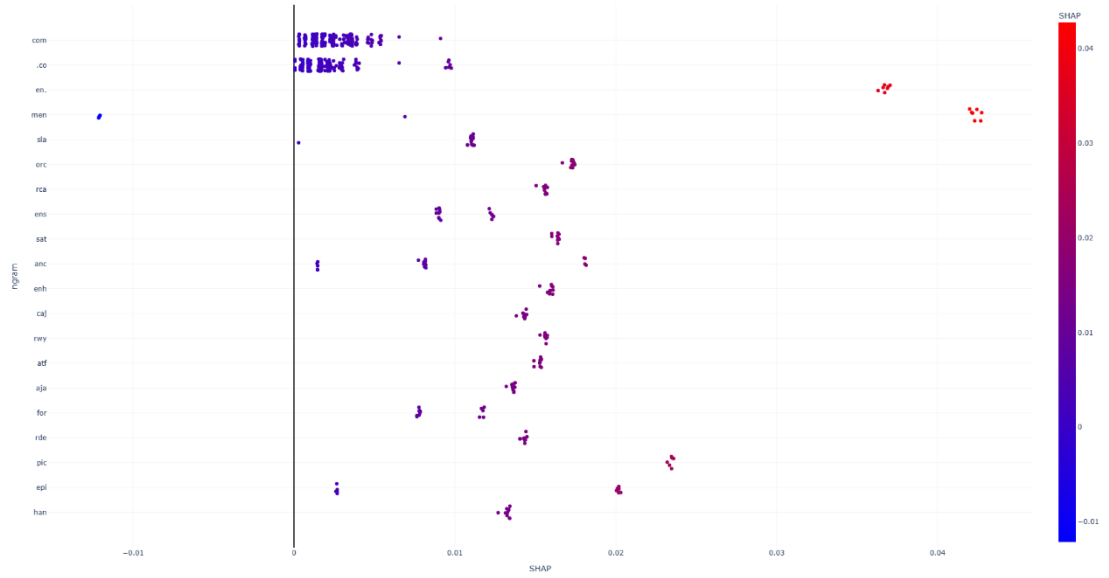
- ✓ και διάφορα n-grams, τα οποία πιθανώς να είναι καθοριστικά για τον DGA από τον οποίο προέρχονται (όπως το zar, από τον bazardoor, όπου θα δούμε παρακάτω) και n-grams τα οποία με το ανθρώπινο μάτι δεν κατανοούμε άμεσα γιατί έπαιξαν σημαντικό ρόλο ώστε το μοντέλο μας να εκτιμήσει ένα malicious domain name, ωστόσο σίγουρα αποτελούν «επικίνδυνες περιπτώσεις» και χρήζουν περαιτέρω διερεύνησης κάθε φορά που τα συναντάμε

5.5 LSTM, SHAP Explainability – Multiclass Classification

Έχοντας υλοποιήσει το LSTM για την περίπτωση της δυϊκής ταξινόμησης, σκεφτήκαμε ότι με ελάχιστες αλλαγές μπορούμε να προσαρμόσουμε το πρόβλημα σε μια multiclass προσέγγιση για την κατηγοριοποίηση των domain names στον DGA από τον οποίο έχουν παραχθεί. Εδώ, ο βασικός στόχος δεν είναι τόσο η ακρίβεια, αλλά η δημιουργία του explainable μοντέλου για να μπορέσουμε να αντλήσουμε πληροφορία σχετικά με τα n-grams (εδώ trigrams) που επηρεάζουν το μοντέλο στην ταξινόμηση των instances στους αντίστοιχους αλγορίθμους από τους οποίους παράχθηκαν. Για τον λόγο αυτό, επιλέχθηκαν οι 11 από τους 63 DGAs, με τα περισσότερα instances στο δείγμα. Για την παραγωγή των διαγραμμάτων, όπως και στις παραπάνω περιπτώσεις επιλέγονται 1000 τυχαία επιλεγμένα δειγματικά στοιχεία. Το LSTM καταφέρνει και σε αυτή την περίπτωση να επιτύχει πολύ ικανοποιητικά αποτελέσματα στις μετρικές accuracy, precision και recall. Ωστόσο, αυτό που μας ενδιαφέρει περισσότερο και θέλουμε να εκμεταλλευτούμε είναι το explainable μοντέλο, καθώς η ακρίβεια σε αυτή την περίπτωση ήταν αναμενομένη λόγω της διαφορετικότητας που υπάρχει μεταξύ των DGA. Συνεπώς, μπορούμε ν' αντιληφθούμε την «δύναμη» του SHAP ως model-agnostic evaluation framework, καθώς μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο και να πάρουμε τις ίδιες μετρικές ως αποτελέσματα. Κατανοούμε ότι η multiclass προσέγγιση είναι ακόμη σε πρωτόλεια μορφή και επιδέχεται πιθανώς βελτιώσεις για την καλύτερη γενίκευση του προβλήματος, για παράδειγμα με την προσθήκη της δυνατότητας ανίχνευσης νεών αλγορίθμων. Ωστόσο, ας επιλέξουμε 3 από τους αλγορίθμους που χρησιμοποιήθηκαν για την εκπαίδευση κι έχουν παραχθεί από τον υπολογισμό μιας αλφαριθμητικής παράστασης (generation scheme).

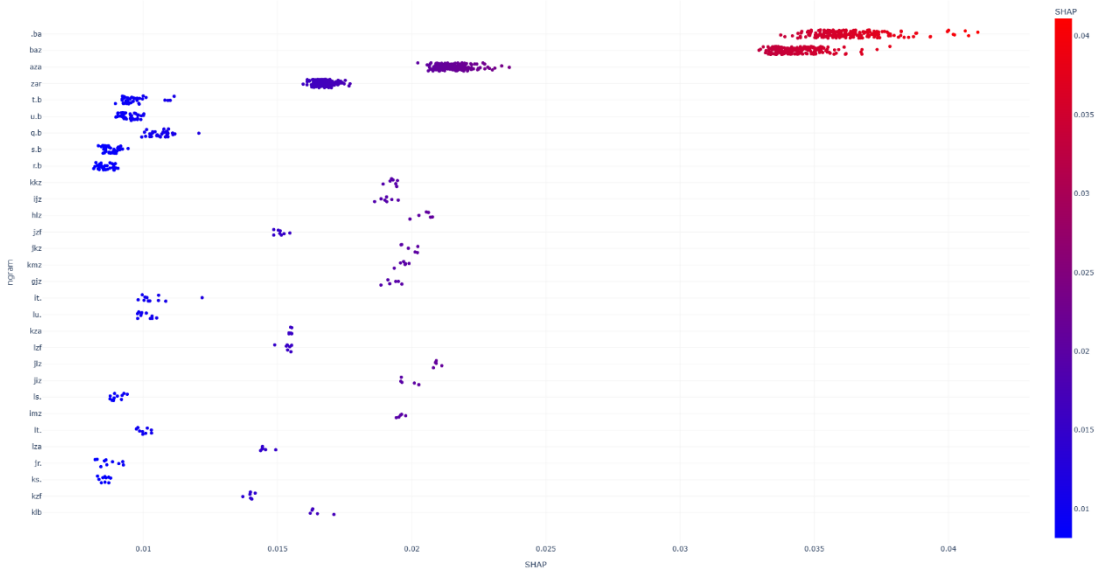
	precision	recall	f1-score	support
banjori	1.00	1.00	1.00	93472
bazardoor	1.00	1.00	1.00	5610
flubot	0.94	1.00	0.97	6031
gameover	1.00	1.00	1.00	2392
mydoom	0.98	1.00	0.99	2003
pykspa_v1	1.00	1.00	1.00	8909
ramnit	0.95	0.66	0.78	3866
ranbyus	0.90	0.97	0.93	2781
rovnix	0.99	1.00	1.00	36289
simda	1.00	0.99	1.00	5653
tinba	0.98	1.00	0.99	19271
accuracy			0.99	186277
macro avg	0.98	0.96	0.97	186277
weighted avg	0.99	0.99	0.99	186277

Εικόνα 37: LSTM - Multiclass Classification



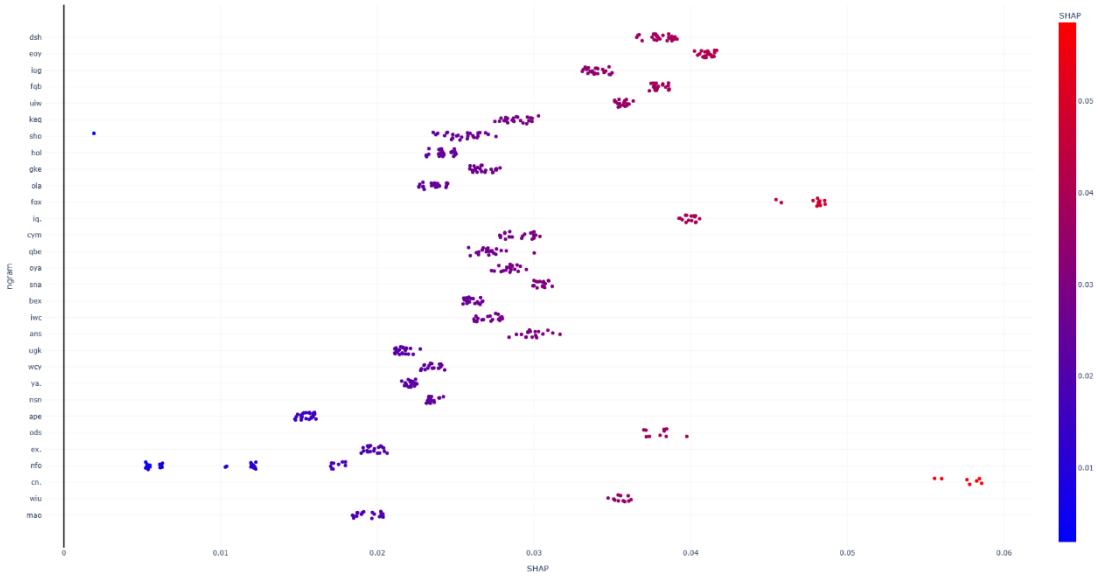
Εικόνα 38: DGA Banjori - Multiclass Analysis

Στην περίπτωση του DGA Banjori (εικόνα 38) παρατηρούμε ότι κυριαρχούν τα trigrams com και .co, κάτι που καθιστά την πρόβλεψη του μοντέλου δυσκολότερη καθώς το .com είναι το πιο σύνηθες TLD. Αυτό, αποδεικνύει και την ασύμμετρη απειλή μεταξύ κακόβουλων και ερευνητών κυβερνοασφάλειας, όπως αναφέραμε και στο θεωρητικό υπόβαθρο, καθώς τα DGA domain names μπορεί να γίνονται register σε ποικίλα top-level-domains και οι αμυνόμενοι οφείλουν να ψάχνουν έναν μεγάλο αριθμό αυτών, ενώ οι διαχειριστές των botnets γνωρίζουν ποιο TLD θα επιλεχθεί ανάλογα τον DGA που χρησιμοποιούν. Στην συγκεκριμένη περίπτωση, το .com ως TLD έχει τον μεγαλύτερο όγκο δεδομένων από έγκυρα domain names, συνεπώς απαιτούν προσοχή τα κακόβουλα που γίνονται register σε αυτό το TLD, διότι γίνονται δυσκολότερα ανιχνεύσιμα. Ακολούθως παρατηρούμε διάφορα trigrams όπως το men και το en., τα οποία όπου εμφανίζονται έχουν πολύ υψηλές τιμές SHAP, που σημαίνει ότι ωθούν την πρόβλεψη προς την κατηγορία του Banjori και προφανώς όταν φιλτράρονται από τα σύστημα μας, είναι θεμιτό να διενεργείται περαιτέρω διερεύνηση. Από το διάγραμμα προκύπτουν και άλλα trigrams με θετικές τιμές SHAP που δεν ξεχωρίζουν τόσο πολύ, ωστόσο μπορούν σίγουρα να βοηθήσουν στην περαιτέρω μελέτη του συγκεκριμένου DGA.



Εικόνα 39: DGA Bazardoor - Multiclass Analysis

Στην περίπτωση του DGA Bazardoor η εκτίμηση και η αξιολόγηση είναι πολύ πιο εύκολες καθώς βλέπουμε τεράστια κυριαρχία των θετικών SHAP value τιμών που αντιστοιχούν στο .ba, στο baz, στο aza και στο zar, το οποίο παρατηρήσαμε και νωρίτερα στο global explainability summary plot κατά την επίλυση του binary προβλήματος. Συνεπώς, η εμφάνιση των παραπάνω trigrams είναι ένα πολύ ισχυρό κριτήριο που ενθαρρύνει το μοντέλο να ανιχνεύσει τον bazardoor.



Εικόνα 40: DGA pykspa_v1

Στην περίπτωση του DGA pykspa_v1 παρατηρούμε ότι δεν υπάρχουν TLDs (Top-Level domains) που να τον καθορίζουν, συνεπώς αναμένουμε έναν DGA που πιθανώς γίνεται register σε διάφορες χώρες και δεν μπορεί να ανιχνευθεί εύκολα, ενδεικτικό της

ασύμμετρης απειλής που αναφέραμε και νωρίτερα. Ωστόσο, από το διάγραμμα προκύπτουν αρκετά trigrams και μάλιστα με αρκετά υψηλές τιμές SHAP, κάτι το οποίο σημαίνει ότι το μοντέλο βοηθήθηκε σημαντικά από αυτά τα trigrams ώστε να κατατάξει τα εκάστοτε instances στον DGA pykspa_v1. Προτείνεται συνεπώς παρακολούθηση και φιλτράρισμα των συγκεκριμένων trigrams, ώστε αν παρατηρηθεί κίνηση που τα εμπεριέχει, να ελεγχθεί περαιτέρω από τα συστήματα ασφαλείας εάν όντως επρόκειτο για DGA και συγκεκριμένα τον pykspa_v1.

Κεφάλαιο 6 – Συμπεράσματα και Μελλοντική Μελέτη

6.1 Ανασκόπηση και Συμπεράσματα

Στην παρούσα διπλωματική εργασία, αρχικά εξετάσαμε την φιλοσοφία και τη δομή των DGA-based botnets, μελετήσαμε τις υπάρχουσες τεχνικές για την ανίχνευση DGA domain names και τις πρώτες προσπάθειες που γίνονται προς την κατεύθυνση της ερμηνείας των μοντέλων που υλοποιούν τέτοιες προσεγγίσεις. Ακολούθως, εκμεταλλευόμενοι το γεγονός ότι η μορφή των domain names που παράγονται από DGA διαφέρει από αυτή των έγκυρων ονομάτων τομέα, εκπαιδεύσαμε ορισμένα μοντέλα, δίνοντας έμφαση στην βαθιά μηχανική μάθηση. Τα τελευταία, ειδικά στην περίπτωση του LSTM παρέχουν ακριβής προβλέψεις για την ανίχνευση κίνησης DGA.

Συνοψίζοντας, ας ξεκινήσουμε από την τελευταία υλοποίηση, όπου αναλύσαμε την multiclass προσέγγιση του προβλήματος, η οποία πετυχαίνει εξαιρετικό accuracy. Αυτό είναι κάπως αναμενόμενο, διότι οι DGA γενικά παρουσιάζουν διαφορές μεταξύ τους και αυτό είναι που τους καθιστά «ενοχλητικούς» κι «επικίνδυνους». Αξίζει να σημειωθεί πως η multiclass προσέγγιση αποτελεί ένα βήμα για την καλύτερη κατανόησή τους αλλά και για την βελτιστοποίηση των συστημάτων ασφαλείας, π.χ. με την χρήση μεθόδων φιλτραρίσματος. Η πληρέστερη εκπαίδευση των μοντέλων (binary, multiclass) σε συνδυασμό με την συνεισφορά των XAI αλγορίθμων βελτιώνουν τα εργαλεία και τις δυνατότητες μας να κατανοήσουμε καλύτερα τόσο τους υπάρχοντες DGA, όσο και το τι καθοδηγεί τα μοντέλα να τους εντοπίζουν αλλά και να ανιχνεύουν νέους. Το τελευταίο αποτελεί μια προσπάθεια γενίκευσης της υλοποίησης του προβλήματος για τον εντοπισμό νέων οικογενειών Domain Generation Algorithms.

Η ανάλυση απλών στατιστικών χαρακτηριστικών των domain names μπορεί να δώσει αποτελέσματα ακρίβειας της τάξης του 90%, ωστόσο κρύβει παθογένειες (π.χ. False Negatives, similar features - legit and DGA domain name). Σημαντική συνεισφορά αποτελεί εδώ το explainability των n-grams, με χρήση LSTM ειδικά στο binary classification, συνδυασμός ο οποίος δεν έχει αναφερθεί στην σχετική βιβλιογραφία που μελετήθηκε. Ο XAI αλγόριθμος SHAP σε συνδυασμό με μοντέλα βαθιάς μηχανικής μάθησης (MLP και LSTM) βοηθούν στην καλύτερη κατανόηση μεθόδων ανίχνευσης κακόβουλης δικτυακής κίνησης από DGA, καθώς το framework του SHAP μας παρέχει ένα σύνολο εργαλείων και γραφημάτων για να ερμηνεύσουμε τα περίπλοκα αλλά ακριβή black-box μοντέλα. Μέσω των πειραμάτων, ελπίζουμε ότι αναδείχθηκε η δυναμική του SHAP ως model-agnostic framework, καθώς μπορεί να εφαρμοστεί σε οποιοδήποτε μοντέλο μηχανικής μάθησης, χρησιμοποιώντας τις ίδιες μετρικές και συνεπώς να προσφέρει συγκρίσιμα αποτελέσματα. Τέλος, είναι σημαντικό να κατανοήσουμε ότι υπάρχει ένα ξεκάθαρο tradeoff μεταξύ της ακρίβειας των μοντέλων με την δυνατότητα και την ευκολία ερμηνείας τους. Όσο πιο περίπλοκα γίνονται τα μοντέλα ανίχνευσης (LSTM) τόσο δυσκολεύεται η επεξηγησιμότητά τους. Αντίθετα, με την χρήση πιο εύκολα κατανοητών χαρακτηριστικών για τον άνθρωπο

στην εκπαίδευση των μοντέλων (MLP), πετυχαίνουμε μια απλούστερη ερμηνεία ως προς την αντίληψη και την κατανόηση, ωστόσο θυσιάζουμε ακρίβεια και ίσως ασφάλεια, καθώς εμφανίζονται ψευδείς προβλέψεις. Τέλος, ένα καταληκτικό σχόλιο, το eXplainability AI (XAI) και οι μέθοδοι που μας παρέχει, μας επιτρέπουν να υλοποιήσουμε αλγορίθμους για να κατανοήσουν τους σύνθετους αλγορίθμους που αναπτύσσουν εσωτερικά τα black-box μοντέλα για την επίλυση περίπλοκων προβλημάτων.

6.2 Μελλοντική Μελέτη

Όσον αφορά τους μελλοντικούς μας στόχους, επιθυμούμε να επεκτείνουμε τις υλοποιήσεις που παρουσιάσαμε στην παρούσα εργασία. Αρχικά, σκοπεύουμε να χρησιμοποιήσουμε περισσότερες οικογένειες DGA, ώστε να βελτιώσουμε την διακριτική ικανότητα των ανιχνευτών μας και να μελετήσουμε περισσότερους τέτοιους αλγορίθμους. Ακόμη, θα θέλαμε να συγκρίνουμε περισσότερα μοντέλα βαθιάς μηχανικής μάθησης, όπως CNN (Convolution Neural Networks) και Bidirectional LSTM (Bi-LSTM), και ιδίως τα τελευταία καθώς όπως φαίνεται η προσέγγιση με την ακολουθία χαρακτήρων του κάθε domain name αποφέρει καλύτερα αποτελέσματα. Σαφώς, ακόμη μια κατεύθυνση επέκτασης της εργασίας μας αποτελεί η υλοποίηση μοντέλων μη επιβλεπόμενης μάθησης (unsupervised learning), π.χ. Autoencoders, για την ανίχνευση κίνησης παραγόμενη από DGA, με αρχικό στόχο τον εντοπισμό συγγενειών μεταξύ των γνωστών DGA και φυσικά τελικό στόχο την ανίχνευση νέων οικογενειών DGA που προκύπτουν ολοένα και συχνότερα. Τέλος, το XAI είναι σίγουρα ένα ιδιαίτερα αναπτυσσόμενο πεδίο στις μέρες μας, εξού και θέλουμε να μελετήσουμε και νεότερους αλγορίθμους που σχετίζονται με το explainability, όπως το counterfactual explanation, το οποίο με βάση τον τρόπο λειτουργίας του υποδεικνύει τι θα πρέπει να είναι διαφορετικό σ’ ένα δειγματικό στοιχείο εισόδου, ώστε ν’ αλλάξει το αποτέλεσμα του AI συστήματος.

Κεφάλαιο 7 – Βιβλιογραφία

[1] Cisco. Cisco Annual Internet Report (2018–2023) White Paper. 2020. url: https: //www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/ annual-internet-report/white-paper-c11-741490.html

[2] Zebin, T., Rezvy, S., & Wang, F. (2022). An Explainable AI-Based Intrusion Detection System for DNS Over HTTPS (DoH) Attacks.
IEEE Transactions on Information Forensics and Security, 17, 2339–2349. https://doi.org/10.1109/tifs.2022.3183390

[3] Rossow, C., Andriesse, D., Werner, T., Stone-Gross, B., Plohmann, D., Dietrich, C.J., And Bos, H. SoK: P2PWNED — Modeling and Evaluating the Resilience of Peer-to-Peer Botnets. In Proceedings of the 34th IEEE Symposium on
Security and Privacy (S&P) (San Francisco, CA, May 2013)

[4] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee and D. Dagon. From throw-away traffic to bots: detecting the rise of DGA- based Malware. 21th USENIX Security Symposium (USENIX Security 12), 2012

[5] What is a Botnet?, Palo Alto Networks, Available online:
What is a Botnet? - Palo Alto Networks

[6] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich. Analysis of the storm and nugache trojans: P2P is here. In USENIX ;login:, vol. 32, no. 6, December 2007

[7] J.Williams. What we know (and learned) from the waledac takedown. http://tinyurl.com/ 7apnn9b, 2010.

[8] S. Golovanov and I. Soumenkov. TDL4 top bot. http://www.securelist.com/en/ analysis/204792180/TDL4_Top_Bot, 2011.

[9] J. Stewart. Bobax trojan analysis. http: //www.secureworks.com/research/ threats/bobax/, 2004.

[10] P. Royal. Analysis of the kraken botnet. http://www.damballa.com/downloads/ r_pubs/KrakenWhitepaper.pdf, 2008.

[11] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna. Your botnet is my botnet: analysis of a botnet takeover. In Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS '09, pages 635–647, New York, NY, USA, 2009. ACM.

[12] D. Plohmann, K. Yakdan, M. Klatt, J. Bader and E. Gerhards-Padilla. A Comprehensive Measurement Study of Domain Generating Malware, 25th USENIX Security Symposium (USENIX Security 16), 2016. pp. 263–278.

[13] Conficker Working Group. Conficker Working Group: Lessons Learned, ConfickerWorking Group, 2011.
http://docplayer.net/16497189-Conficker-working-group-lessons-learned.html

[14] T. Barabosch, A. Wichmann, F. Leder and E. Gerhards-Padilla. Automatic extraction of domain name generation algorithms from current malware. NATO Symposium IST-111 on Information Assurance and Cyber Defense, 2012.

[15] SCHWARZ, D. Bedep's DGA: Trading Foreign Exchange for Malware Domains, 2015. Blog post:
https://asert.arbornetworks.com/bedeps-dga-trading-foreign-exchange-for-malware-domains/.

[16] J. Kurose and K. Ross, Computer Networking: A Top Down Approach Using the Internet, Addison-Wesley Computer Science, 6th Edition, 2013.

[17] domain name system (DNS), TechTarget, Available online:
What is DNS? How Domain Name System works (techtarget.com)

[18] What is DNS? | How DNS works, Cloudflare, Available online:
What is DNS? | How DNS works | Cloudflare

[19] Wu, Hao; Dang, Xianglei; Wang, Lidong; He, Longtao (2016). "Information fusion-based method for distributed domain name system cache poisoning attack detection and identification". IET Information Security. 10 (1): 37–44.

[20] M. Du, N. Liu, and X. Hu, ``Techniques for interpretable machine learning," 2018, arXiv:1808.00033. [Online].
Available: http://arxiv.org/abs/1808.00033

[21] Fisher, Aaron, Cynthia Rudin, and Francesca Dominici. "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously."
https://arxiv.org/abs/1801.01489

[22] Altmann, A., Toloşi, L., Sander, O., & Lengauer, T. (2010). Permutation importance: a corrected feature importance measure. Bioinformatics, 26(10), 1340–1347.

[23] Molnar, C. (2022). Interpretable Machine Learning:
A Guide for Making Black Box Models Explainable (2nd ed.).
christophm.github.io/interpretable-ml-book/

[24] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016, August). "Why should I trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 1135–1144).

[25] Baehrens, D., Schroeter, T., Harmeling, S., Kawanabe, M., Hansen, K., & MÃžller, K. R. (2010). How to explain individual classification decisions. Journal of Machine Learning Research, 11(Jun), 1803–1831.

[26] Alvarez-Melis, D., & Jaakkola, T. S. (2018). On the robustness of interpretability methods. arXiv preprint arXiv:1806.08049.9

[27] Thomas, A. et al., Limitations of Interpretable Machine Learning Methods [Book] (2020): Available online:
Limitations of Interpretable Machine Learning Methods (slds-lmu.github.io)

[28] S. M. Lundberg and S.-I. Lee, ‘‘A unified approach to interpreting model predictions,’’ in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 4765–4774.

[29] L. S. Shapley, ‘‘A value for n-person games,’’ Contributions to Theory Games, vol. 2, no. 28, pp. 307–317, 1953

[30] Kamal Alieyan, Ammar ALmomani, Ahmad Manasrah and Mohammed M. Kadhum. A survey of botnet detection based on DNS., Neural Comput Applic (2017)., 2017.vol. 28. pp. 1541–1558.

[31] D. K. Mcgrath and M. Gupta. "Behind Phishing: An Examination of Phisher Modi Operandi." LEET, vol. 8, 2008.

[32] Y. Zhou, Q. S. Li, Q. Miao and K. Yim. DGA-Based Botnet Detection Using DNS Traffic., Journal of Internet Services and Information Security, 2013. vol. 3. pp. 116–123.

[33] T. D. Nguyen, T. D. CAO and l. G. Nguyen. DGA Botnet detection using Collaborative Filtering and Density-based Clustering. SoICT 2015 Proceedings of the Sixth International Symposium on Information and Communication Technology, 2015.

[34] Jonghoon Kwon, Jehyun Lee, Heejo Lee and Adrian Perrig. PsyBoG:A scalable botnet detection method for large-scale DNS traffic., Computer Networks (The International Journal of Computer and Telecommunications Networking),2016. vol. 97. pp. 48–73.

[35] Han Zhang, Manaf Gharaibeh, Spiros Thanasoulas and Christos Papadopoulos. Bot-Digger: Detecting DGA Bots in a Single Network. Proceedings of the 2016 Network Traffic Measurement and Analysis Conference (TMA), 2016.

[36] Tzy Shiah Wang, Hui Tang Lin, Wei Tsung Cheng and Chang Yu Chen. DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis., Computers and Security, 2016. vol. 64. pp. 1–15.

[37] S. Krishnan, T. Taylor, F. Monrose and J. McHugh. Crossing the threshold: Detecting network malfeasance via sequential hypothesis testing. 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2013.

[38] J. Raghuram, D. J. Miller and G. Kesidis. Unsupervised, low latency anomaly detection of algorithmically generated domain names by generative probabilistic modeling., Journal of Advanced Research, 2014. vol. 5. pp. 423–433.

[39] Yu Chen, Sheng Yan, Tianyu Pang and Rui Chen. Detection of DGA Domains Based on Support Vector Machine. Third International Conference on Security of Smart Cities, Industrial Control System and Communications (SSIC), 2018.

[40] G. Zhang J. Huang and Y. Shen. DGA Domain Name Detection Based on SVM Under Grey Wolf optimization Algorithm. 2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS), 2019.

[41] Hieu Mac, Duc Tran, Van Tong, Linh Giang Nguyen and Hai Anh Tran. DGA Botnet Detection Using Supervised Learning Methods. In SoICT 17: 8th International Symposium on Information and Communication Technology, 2017.

[42] Jan Spooren, Davy Preuveneers, Lieven Desmet, Peter Janssen and Wouter Joosen. Detection of Algorithmically Generated Domain Names used by Botnets: A Dual Arms Race. SAC 2019: The 34th ACM/SIGAPP Symposium On Applied Computing, 2019.

[43] Jonathan Woodbridge, Hyrum S. Anderson, Anjum Ahuja and Daniel Grant. Predicting Domain Generation Algorithms with Long Short-Term Memory Networks, Endgame Inc., 2016. arXiv:1611.00791 [cs.CR] https://arxiv.org/abs/1611.00791

[44] Palak V¼, Sayali Nikam and Ashutosh Bhatia. Detection of Algorithmically Generated Domain Names using LSTM. 2020 12th International Conference on CommunicationSystems Networks (COMSNETS), 2020.

[45] Tran, D., Mac, H., Tong, V., Tran, H. A., & Nguyen, L. G. (2018). A LSTM based framework for handling multiclass imbalance in DGA botnet detection. Neurocomputing, 275, 2401–2413. https://doi.org/10.1016/j.neucom.2017.11.018

[46] Bin Yu, Daniel L. Gray, Jie Pan, Martine De Cock and Anderson C. A. Nascimento. Inline DGA Detection with Deep Networks. 2017 IEEE International Conference on Data Mining Workshops, 2017.

[47] G. Liu, J. Zhai, Y. Dai, Z. Yan, Y. Zou and W. Huang. A Novel Detection Method for Word-Based DGA. International Conference on Cloud Computing and Security (ICCCS 2018), 2018.

[48] Charan, P. S., Shukla, S. K., & Anand, P. M. (2020). Detecting Word Based DGA Domains Using Ensemble Models. Lecture Notes in Computer Science, 127–143. https://doi.org/10.1007/978-3-030-65411-5_7

[49] R. R. Curtin, A. B. Gardner, S. Grzonkowski, A. Kleymenov and A. Mosquera. Detecting DGA domains with recurrent neural networks and side information. 14th International Conference on Availability, Reliability and Security (ARES 2019), 2019.

[50] Wang, M., Zheng, K., Yang, Y., & Wang, X. (2020). An Explainable Machine Learning Framework for Intrusion Detection Systems. IEEE Access, 8, 73127–73141. https://doi.org/10.1109/access.2020.2988359

[51] Drichel, A., Faerber, N., & Meyer, U. (2021). First Step Towards EXPLAINable DGA Multiclass Classification. The 16th International Conference on Availability, Reliability and Security. https://doi.org/10.1145/3465481.3465749

[52] Samuel Schüppen, Dominik Teubert, Patrick Herrmann, and Ulrike Meyer. 2018. FANCI: Feature-Based Automated NXDomain Classification and Intelligence. In USENIX Security Symposium.

[53] H. Suryotrisongko, Y. Musashi, A. Tsuneda and K. Sugitani, "Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing," in IEEE Access, vol. 10, pp. 34613-34624, 2022, doi: 10.1109/ACCESS.2022.3162588

[54] Piras, G., Pintor, M., Demetrio, L., & Biggio, B. (2022). Explaining Machine Learning DGA Detectors from DNS Traffic Data. arXiv preprint arXiv:2208.05285

[55] Netlab DGA Project. https://data.netlab.360.com/dga/, Access Date: 20-12-2022.

[56] Buitinck, L. et al., 2013. API design for machine learning software: experiences from the scikit-learn project. In ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122.

[57] SHAP, GitHub - slundberg/shap: A game theoretic approach to explain the output of any machine learning model.

[58] Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

[59] Martín Abadi, et al.. TensorFlow: Large-scale machine learning on heterogeneous systems,2015. Software available from https://www.tensorflow.org/

[60] Hara, K., Saito, D., & Shouno, H. (2015). Analysis of function of rectified linear unit used in deep learning. 2015 International Joint Conference on Neural Networks (IJCNN). doi:10.1109/ijcnn.2015.7280578

[61] shap.DeepExplainer, shap.DeepExplainer — SHAP latest documentation (shap-lrjball.readthedocs.io) , Access Date: 10/01/2023

[62] Inc., P. T. (2015). Collaborative data science. Montreal, QC: Plotly Technologies Inc. Retrieved from https://plot.ly