ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Δημιουργία Ταξινομητή για Κατηγοριοποίηση Δικτυακών Επιθέσεων με Χρήση Νευρωνικών και Βαθιών Νευρωνικών Δικτύων

Διπλωματική Εργασία

Του

Σταματίου Β. Κουρκούτα

Επιβλέπων: Βασίλειος Μάγκλαρης
Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος, 2017

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Δημιουργία Ταξινομητή για Κατηγοριοποίηση Δικτυακών Επιθέσεων με Χρήση Νευρωνικών και Βαθιών Νευρωνικών Δικτύων

Διπλωματική Εργασία

Του

Σταματίου Β. Κουρκούτα

Επιβλέπων: Βασίλειος Μάγκλαρης
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 7η Νοεμβρίου 2017.

…….	…….	…….
Μάγκλαρης Β.	Κοζύρης Ν.	Συκάς Ε.
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Νοέμβριος, 2017

……..
Σταμάτιος Κουρκούτας
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Copyright © Κουρκούτας Σταμάτιος, 2017
Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, η 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 αναφέρονται προς το συγγραφέα. Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν το συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Η εκπόνηση της παρούσας διπλωματικής εργασίας έγινε στα πλαίσια της φοίτησής μου στο τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου. Θα ήθελα να ευχαριστήσω τον καθηγητή μου κ. Βασίλη Μάγκλαρη για τη δυνατότητα που μου έδωσε να ασχοληθώ με το συγκεκριμένο, πολύ ενδιαφέρον θέμα καθώς και την εμπιστοσύνη που μου έδειξε κατά την εκπόνησή του. Παράλληλα θα ήθελα να ευχαριστήσω τον κ. Αδάμ Παυλίδη για τη βοήθεια και την καθοδήγηση στα διάφορα στάδια της διπλωματικής μου εργασίας. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την συμπαράσταση και τη στήριξη τους όλα αυτά τα χρόνια.

Περίληψη

Στη σημερινή εποχή όπου το διαδίκτυο και οι εφαρμογές του είναι μέρος της καθημερινότητάς μας, η δικτυακή ασφάλεια είναι θέμα υψίστης σημασίας. Καθημερινά παρατηρούνται πολλές διαφορετικές δικτυακές επιθέσεις, όπως για παράδειγμα επιθέσεις Άρνησης Υπηρεσίας (Denial of Service, DoS) καθώς και προσπάθειες εισβολής σε διάφορα δίκτυα με στόχο την άντληση πληροφοριών.

Παράλληλα υπάρχει αυξημένο ερευνητικό ενδιαφέρον σχετικά με τις τεχνικές μηχανικής μάθησης και πιο συγκεκριμένα με τα Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks). Η χρήση των Βαθιών Νευρωνικών Δικτύων περιλαμβάνει ένα μεγάλο εύρος εφαρμογών όπως για παράδειγμα την ανάλυση και επεξεργασία εικόνων, την αναγνώριση φωνής και την επεξεργασία φυσικής γλώσσας.

Αντικείμενο της παρούσης διπλωματικής εργασίας είναι η μελέτη και η χρήση διαφορετικών Νευρωνικών Δικτύων με στόχο την κατηγοριοποίηση της δικτυακής κίνησης σε ένα υπό εξέταση δίκτυο σε καλόβουλη, θεμιτή κίνηση ή σε κακόβουλη, αθέμιτη κίνηση, η οποία είναι μέρος μιας επίθεσης που έχει ως στόχο την παρεμπόδιση των παρεχόμενων υπηρεσιών από το δίκτυο. Ειδικότερα επιδιώκουμε την κατάταξη της κακόβουλης κίνησης σε τέσσερις υποκατηγορίες οι οποίες είναι η Πλημμύρα ICMP πακέτων (ICMP Flood), η Πλημμύρα TCP πακέτων με σημαία SYN (TCP SYN Flood), η Πλημμύρα UDP πακέτων (UDP Flood) και η επίθεση Σάρωσης Θυρών (Port Scanning).

Η μελέτη επικεντρώθηκε σε Νευρωνικά Δίκτυα αλλά και Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks) και οι τύποι δικτύων που εξετάστηκαν είναι τα Νευρωνικά Δίκτυα Πολλών Επιπέδων (Multi-Layer Perceptrons, MLPs), τα Νευρωνικά Δίκτυα Ανάδρασης (Recurrent Neural Networks, RNN) και τα LSTM (Long Short-Term Memory).

Τα αποτελέσματα έδειξαν ότι τα Νευρωνικά και τα Βαθιά Νευρωνικά Δίκτυα μπορούν με πολύ καλή ακρίβεια να κατηγοριοποιήσουν την κίνηση σε καλόβουλη ή σε μια από τις παραπάνω κατηγορίες επιθέσεων. Ο μηχανισμός που υλοποιήθηκε για την εξαγωγή των συμπερασμάτων μπορεί να επεκταθεί περαιτέρω για τη δημιουργία ενός εργαλείου ανίχνευσης και απόρριψης κακόβουλης κίνησης της προαναφερθείσας μορφής από ένα δίκτυο.

Λέξεις κλειδιά: Ανίχνευση Επιθέσεων, Επιθέσεις Άρνησης Υπηρεσίας, Νευρωνικά Δίκτυα, Βαθιά Νευρωνικά Δίκτυα, Netflow, Ταξινόμηση Δικτυακής κίνησης

Abstract

Nowadays, Internet and its applications are part of our everyday lives and as a result network security is an issue of significant importance. Everyday a large amount of network attacks, such as Denial of Service (DOS) attacks or efforts of intrusion into different networks, with the goal of stealing information, are observed.

Simultaneously, there is an increasing research interest in the field of Machine Learning and more specifically an interest in Deep Neural Networks. Deep Neural Networks are used in many varying applications and fields such as image analysis, voice recognition and natural language processing.

The main purpose of the diploma thesis is the studying and the use of different Neural Networks in order to classify the internet traffic of an internet network into legitimate, wanted traffic or malicious, unwanted traffic which is part of an attack that aims to prevent the services provided by the network. More specifically the classification of malicious traffic into four distinct subcategories is pursued which are ICMP Flood, TCP SYN Flood, UDP Flood and Port Scanning.

The research focused on the following types of Neural and Deep Neural Networks, Multi-Layer Perceptrons (MLP), Recurrent Neural Networks (RNN) and Long Short-Term Memory Neural Networks (LSTM).

The results demonstrated that Neural and Deep Neural Networks are highly capable of classifying internet traffic into the five aforementioned categories, achieving almost perfect classification accuracy. The mechanism that was created in order to draw these conclusions can be expanded towards the creation of a tool suited for the recognition and rejection of malicious traffic observed in an internet network.

Keywords: Attack Detection, Denial of Service Attacks, Neural Networks, Deep Neural Networks, Netflow, Internet Traffic Classification

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Περιεχόμενα

1 Εισαγωγή

1.1 Το ερευνητικό πρόβλημα

Ένα από τα σημαντικότερα προβλήματα που καλούνται να λύσουν σήμερα οι διαχειριστές των δικτύων είναι αυτό της δικτυακής ασφάλειας. Ένα δίκτυο γίνεται πολύ συχνά θύμα μιας κυβερνοεπίθεσης της οποίας ο δράστης έχει ως απώτερο στόχο να βγάλει εκτός υπηρεσίας έναν δικτυακό πόρο ώστε οι νόμιμοι χρήστες να μην μπορούν να τον χρησιμοποιήσουν. Κάνουμε λόγο λοιπόν για επιθέσεις Άρνησης Παροχής Υπηρεσίας (Denial of Service, DoS), οι οποίες μπορεί να προέρχονται είτε από ένα μηχάνημα, είτε από περισσότερα οπότε χαρακτηρίζονται και κατανεμημένες (Distributed Denial of Service, DDoS). Άλλες πάλι φορές ως επίθεση μπορεί να χαρακτηριστεί η προσπάθεια κάποιου τρίτου να βρει τρωτά σημεία στο δίκτυο με σκοπό την περαιτέρω εκμετάλλευσή τους και την άρνηση συγκεκριμένων υπηρεσιών σε επόμενο στάδιο. Τέτοια επίθεση είναι η Port Scanning κατά την οποία ο δράστης σαρώνει όλες τις θύρες ενός ή περισσότερων μηχανημάτων προκειμένου να βρει κάποια ‘’ανοιχτή’’ θύρα που αντιστοιχεί σε κάποια τρωτή υπηρεσία του συστήματος.

Όσο αυξάνεται ο αριθμός των συσκευών που συνδέονται στο διαδίκτυο παρατηρείται, σύμφωνα με το [1], ένας αυξανόμενος αριθμός επιθέσεων σαν αυτές που προαναφέραμε. Όπως είναι φυσικό έχουν προταθεί και πολλοί διαφορετικοί τρόποι και τεχνικές πρόληψης, αναγνώρισης και αντιμετώπισής τους.

Παράλληλα η ανάγκη για όλο και ‘’εξυπνότερες’’ εφαρμογές έχει οδηγήσει σε ένα αυξημένο ερευνητικό ενδιαφέρον γύρω από τη μηχανική μάθηση. Υπάρχουν πολλές διαφορετικές τεχνικές μηχανικής μάθησης οι οποίες εφαρμόζονται σε διάφορους τομείς για τη δημιουργία εφαρμογών με δυνατότητες όπως για παράδειγμα ανάλυση και επεξεργασία εικόνων, αναγνώριση φωνής και επεξεργασία φυσικής γλώσσας. Τα Τεχνητά Νευρωνικά Δίκτυα (Artificial Neural Networks) είναι μια από τις πιο διαδεδομένες τεχνικές μηχανικής μάθησης και όταν έχουμε Τεχνητά Νευρωνικά Δίκτυα με περισσότερα από ένα κρυφά επίπεδα κάνουμε λόγο για Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks). Ένα από τα πεδία που μελετάται η προσφορά των Βαθιών Νευρωνικών Δικτύων είναι και το πρόβλημα της αναγνώρισης και της κατηγοριοποίησης δικτυακών επιθέσεων που προαναφέραμε.

1.2 Σκοπός της Εργασίας

Στην παρούσα διπλωματική εργασία εξετάζεται η δυνατότητα των Βαθιών Νευρωνικών Δικτύων να κατηγοριοποιήσουν την κίνηση ενός δικτύου σε καλόβουλη ή κακόβουλη. Ειδικότερα διακρίνουμε πέντε διαφορετικές κατηγορίες, τέσσερις κατηγορίες επιθέσεων και την κατηγορία ‘’καλόβουλη κίνηση’’, βλέπε και ενότητες 2.1.1 και 2.1.2, οι οποίες είναι:

- Πλημμύρα ICMP (ICMP Flood ή Ping Flood)
- Πλημμύρα πακέτων TCP SYN (SYN Flood)
- Πλημμύρα UDP (UDP Flood)
- Σάρωση Θυρών (Port Scanning)
- Καλόβουλη κίνηση (Legitimate traffic)

Στο πλαίσιο της εργασίας προσπαθούμε να ανακαλύψουμε το είδος και τις παραμέτρους ενός Βαθιού Νευρωνικού Δικτύου που οδηγούν στην καλύτερη κατηγοριοποίηση της δικτυακής κίνησης σε μια από τις παραπάνω κατηγορίες. Για το λόγο αυτό κάνουμε χρήση

διαφορετικών ειδών Βαθιών Νευρωνικών Δικτύων όπως είναι τα Νευρωνικά Δίκτυα Πολλών Επιπέδων (Multi-Layer Perceptrons, MLPs), τα Νευρωνικά Δίκτυα Ανάδρασης (Recurrent Neural Networks, RNN) και τα LSTM (Long Short-Term Memory). Ιδιαίτερη προσοχή δίνεται στα χαρακτηριστικά των δεδομένων εκπαίδευσης που χρησιμοποιούμε καθώς προσπαθούμε να προτείνουμε τη χρήση αυτών και μόνο αυτών που συντελούν σε όσο το δυνατόν καλύτερη ταξινόμηση της δικτυακής κίνησης σε μια από τις παραπάνω κατηγορίες. Τέλος προτείνεται ένα Βαθύ Νευρωνικό Δίκτυο το οποίο βάσει των πειραματικών αποτελεσμάτων κρίνουμε ως καταλληλότερο για την κατηγοριοποίηση που θέλουμε να επιτύχουμε.

1.3 Δομή της Εργασίας

Η διπλωματική εργασία οργανώνεται σε πέντε κεφάλαια. Στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο που απαιτείται για την εκπόνηση της εργασίας σχετικά με τις δικτυακές επιθέσεις και την ασφάλεια δικτύων αλλά και οι απαραίτητες γνώσεις για τη μηχανική μάθηση και τα Νευρωνικά Δίκτυα. Στο τρίτο κεφάλαιο παρουσιάζεται η προτεινόμενη αρχιτεκτονική του συστήματος που θα χρησιμοποιηθεί για την πειραματική μελέτη και την εξαγωγή συμπερασμάτων. Στο επόμενο κεφάλαιο παρουσιάζονται αναλυτικά όλα τα εργαλεία που χρησιμοποιήθηκαν για την υλοποίηση του συστήματος που προτείνεται και γίνετε ανάλυση των επιμέρους λεπτομερειών υλοποίησης. Τέλος παρουσιάζονται τα αποτελέσματα των επιμέρους πειραμάτων που υλοποιήθηκαν και τα γενικά συμπεράσματα της εργασίας.

2 Θεωρητικό Υπόβαθρο

2.1 Ασφάλεια Δικτύων – Δικτυακή κίνηση

Στην ενότητα αυτή παρουσιάζουμε τη βασική θεωρία, γύρω από τη δικτυακή ασφάλεια, τις δικτυακές επιθέσεις και την καταγραφή δικτυακής κίνησης, η οποία απαιτείται για την κατανόηση της παρούσας διπλωματικής εργασίας.

2.1.1 Είδη Δικτυακών Επιθέσεων

Όπως αναφέραμε και στην εισαγωγική ενότητα οι επιθέσεις στο διαδίκτυο είναι ένα πολύ συχνό χαρακτηριστικό. Δεν έχουν όλες όμως τα ίδια χαρακτηριστικά. Ακολούθως αναφέρουμε κάποιες βασικές κατηγορίες δικτυακών επιθέσεων σύμφωνα με τα [2], [3], [4], [5], [6] και [7].

1) Διάδοση Κακόβουλου Λογισμικού (Malware)

Κακόβουλο λογισμικό (Malware) ονομάζεται ένα πρόγραμμα, το οποίο εισάγεται σε ένα σύστημα κρυφά, συνήθως μέσω αρχείων κατεβασμένων από το διαδίκτυο, με την πρόθεση να προκληθεί βλάβη στην εμπιστευτικότητα, ιδιωτικότητα, ακεραιότητα ή διαθεσιμότητα των δεδομένων, των εφαρμογών ή του λειτουργικού συστήματος και γενικά να προκληθεί ενόχληση του θύματος. Σε αυτήν την κατηγορία ανήκουν:

- Ο Ιός (Virus) και ο Μακροϊός (Macro-virus)
- Το Σκουλήκι (Worm)
- Ο Δούρειος Ίππος (Trojan Horse)
- Το Περαστικό Κατέβασμα (Drive-by download)
- Λογισμικό που συλλέγει πληροφορίες από ένα σύστημα και τις μεταδίδει σε ένα άλλο (Spyware)
- Λογισμικό που έχει εγκατασταθεί στο σύστημα, ενεργοποιείται την κατάλληλη χρονική στιγμή και προκαλεί επιθέσεις σε ένα άλλο σύστημα (Zombie ή Bot)
- Γεννήτρια μεγάλου όγκου δεδομένων με σκοπό την επίθεση σε δικτυωμένους υπολογιστές (Flooder)
- Συλλογή από εργαλεία (Rootkit), τα οποία εγκαθίστανται και δίνουν δικαιώματα διαχειριστή (root access) σε μη εξουσιοδοτημένους, από το σύστημα, χρήστες.
- Κώδικας που εκμεταλλεύεται μια συγκεκριμένη αδυναμία του συστήματος (Exploit) και κώδικας που εκμεταλλεύεται μια άγνωστη προηγουμένως αδυναμία του συστήματος (Zero-day Exploit).
- Λογική Βόμβα (Logic Bomb), δηλαδή κώδικας που ενεργοποιεί την εκτέλεση κακόβουλου λογισμικού όταν γίνει αληθής μια συνθήκη.

2) Επίθεση Άρνησης Υπηρεσίας (Denial of Service, DoS)

Μία τέτοια επίθεση έχει ως στόχο να βλάψει τη διαθεσιμότητα μιας παρεχόμενης υπηρεσίας ενός δικτύου. Η επίθεση στοχεύει κρίσιμους πόρους, χωρίς τους οποίους η υπηρεσία δεν μπορεί να παρέχεται απρόσκοπτα. Ανάλογα με τον πόρο που στοχεύει η επίθεση DoS, μπορεί να καταταχθεί σε κάποιες επιμέρους υποκατηγορίες, σύμφωνα με τα [4], [5] και [6].

- **Ογκομετρική Επίθεση (Volumetric Attack) με στόχο το εύρος ζώνης:**

Ο πόρος που προσπαθεί να εξαντλήσει μια επίθεση αυτού του είδους είναι είτε το εύρος ζώνης της σύνδεσης του δικτύου-θύματος με το υπόλοιπο διαδίκτυο, είτε το εύρος ζώνης των εσωτερικών συνδέσεων του δικτύου-θύματος. Αυτό επιτυγχάνεται στέλνοντας προς το δίκτυο-θύμα πολύ μεγάλο όγκο δικτυακής κίνησης με αποτέλεσμα να τον κατακλύζει. Για αυτό το λόγο οι επιθέσεις αυτές ονομάζονται και επιθέσεις Πλημμύρας (Flood). Το μέγεθος των επιθέσεων αυτής της κατηγορίας μετριέται σε bits ανά δευτερόλεπτο (bps).

- **Επίθεση Πρωτοκόλλου (Protocol Attack), με στόχο κάποιον υλικό πόρο ενός μηχανήματος:**

Μια τέτοια επίθεση στοχεύει σε έναν συγκεκριμένο υλικό πόρο ενός συγκεκριμένου μηχανήματος στο υπό επίθεση δίκτυο και προσπαθεί να τον θέσει εκτός λειτουργίας. Ένας τέτοιος πόρος μπορεί να είναι για παράδειγμα η διαθέσιμη μνήμη RAM ή ο επεξεργαστής και η χρησιμοποίηση αυτού, ενός μηχανήματος που λειτουργεί ως εξυπηρετητής (server). Σε αυτήν την κατηγορία, η επίθεση βασίζεται σε κάποιο συγκεκριμένο δικτυακό πρωτόκολλο. Ανάλογα με την λειτουργία την οποία επιτελεί το πρωτόκολλο αυτό, η επίθεση αποκτά συγκεκριμένα χαρακτηριστικά και προσπαθεί να προκαλέσει τη λάθος λειτουργία του πρωτοκόλλου, με σκοπό την εξάντληση υλικών πόρων. Αναφέρουμε ως παράδειγμα την επίθεση Πλημμύρας TCP SYN, βλέπε ενότητα 2.1.2. Επειδή βασικό χαρακτηριστικό και αυτών των επιθέσεων είναι ο μεγάλος όγκος δικτυακής κίνησης που λαμβάνει το θύμα, μπορούν και αυτές οι επιθέσεις να θεωρηθούν ογκομετρικές (Volumetric Attacks) με την ευρύτερη έννοια. Το μέγεθος όμως των επιθέσεων αυτής της κατηγορίας μετριέται σε πακέτα ανά δευτερόλεπτο.

- **Επίθεση βασιζόμενη σε κάποιο πρωτόκολλο του επιπέδου εφαρμογής (Application Layer Attack):**

Η επίθεση αυτής της κατηγορίας αποτελείται συνήθως από φαινομενικά καλόβουλα και αθώα αιτήματα, στόχος των οποίον είναι να απενεργοποιήσουν (crash) έναν εξυπηρετητή. Στην κατηγορία αυτή περιλαμβάνονται μικρές και αργές επιθέσεις (low-and-slow attacks), πλημμύρες GET και POST μηνυμάτων (GET/POST floods), επιθέσεις που σημαδεύουν αδυναμίες και τρωτά σημεία των λογισμικών Apache, Windows και OpenBSD και άλλα. Το μέγεθος τέτοιων επιθέσεων μετριέται σε αιτήματα ανά δευτερόλεπτο (requests per second).

Επειδή οι επιθέσεις DoS απαιτούν συνήθως την αποστολή πολύ μεγάλου όγκου δεδομένων, τις περισσότερες φορές, η πηγή μιας επίθεσης DoS δεν είναι ένας συγκεκριμένος υπολογιστής ή δίκτυο αλλά πολλοί διαφορετικοί υπολογιστές, οι οποίοι ανήκουν και σε διαφορετικά δίκτυα. Τότε κάνουμε λόγο για Κατανεμημένη Επίθεση Άρνησης Υπηρεσίας (Distributed Denial of Service, DDoS). Επίθεση DDoS μπορεί να συμβεί από δίκτυα με ‘’μολυσμένους’’, από κακόβουλο λογισμικό, υπολογιστές, όπως Bots, οπότε κάνουμε λόγο για Botnet και Flooders που περιγράψαμε προηγουμένως. Σε αυτή την περίπτωση καθίσταται δυσκολότερη και η ανεύρεση του πραγματικού θύτη της επίθεσης.

3) Άλλες κακόβουλες ενέργειες όπως Σάρωση Θυρών (Port Scanning)

Αν και η Σάρωση Θυρών (Port Scanning) δεν πρόκειται για μία άμεσης μορφής επίθεση, όπως οι παραπάνω, είναι μια ενέργεια με κακόβουλες προθέσεις, τις περισσότερες φορές. Σύμφωνα και με το [7], ο επιτιθέμενος στέλνει προς το θύμα πακέτα TCP ή UDP, αλλάζοντας σε κάθε πακέτο τη θύρα προορισμού (Destination Port) του πρωτοκόλλου μεταφοράς. Με τον τρόπο αυτό σαρώνει όλες, ή τις πιο σημαντικές θύρες ενός υπολογιστή-θύματος και στη συνέχεια εξετάζει μία προς μία τις απαντήσεις από κάθε θύρα. Έτσι ο θύτης προσπαθεί να βρει την κατάσταση στην οποία βρίσκεται η κάθε θύρα, δηλαδή αν είναι ανοικτή ή κλειστή και επιπλέον συλλέγονται πληροφορίες για το είδος του λειτουργικού συστήματος του θύματος και δεδομένα τα οποία πιθανόν να χρησιμοποιηθούν μελλοντικά όπως πιθανές αδυναμίες ή τρωτές υπηρεσίες του συστήματος.

Πιο συγκεκριμένα αναφέρουμε ως παράδειγμα τη μέθοδο σάρωσης TCP SYN scan. Κατά τη διάρκεια αυτής, σε TCP πακέτο του θύτη με σημαία SYN που αποστέλνεται προς μια θύρα προορισμού του θύματος, ο θύτης μπορεί να λάβει ως απάντηση τα ακόλουθα:

- Πακέτο με σημαίες SYN και ACK, το οποίο υποδεικνύει ότι η θύρα είναι ανοιχτή (Open ή Accepted). Αν ληφθεί πακέτο SYN, χωρίς όμως τη σημαία ACK, η θύρα και πάλι θεωρείται ανοιχτή. Αυτό οφείλεται σε ένα πολύ σπάνιο χαρακτηριστικό του TCP γνωστό στη διεθνή βιβλιογραφία ως simultaneous open ή split handshake connection.
- Πακέτο με σημαία RST, δηλαδή η σύνδεση αρνείται, το οποίο υποδεικνύει ότι η θύρα δεν ακούει ή είναι κλειστή (Closed ή Denied ή Not Listening).
- Μήνυμα ICMP Destination Unreachable (type 3, code 0, 1, 2, 3, 9, 10, or 13) ή να μη λάβει καθόλου απάντηση, ακόμη και μετά από αρκετές αναμεταδόσεις, το οποίο υποδεικνύει ότι πρόκειται για φιλτραρισμένη θύρα (Filtered, Dropped ή Blocked).

Άλλες μέθοδοι Σάρωσης Θυρών, οι οποίες αναφέρονται στο [7] και δε θα αναλυθούν εδώ περαιτέρω είναι τα: TCP connect scan, UDP scans, SCTP INIT scan, TCP NULL, FIN, and Xmas scans, TCP ACK scan, TCP Window scan και TCP Maimon scan.

2.1.2 Παραδείγματα Επιθέσεων Άρνησης Υπηρεσίας (Denial of Service, DoS)

Σε αυτήν την υποενότητα παρουσιάζουμε τις πιο συχνές επιθέσεις Άρνησης Υπηρεσίας (Denial of Service, DoS) σύμφωνα με τα [4], [5] και [6], κάθε μία από τις οποίες ανήκει φυσικά σε μια από τις κατηγορίες που περιγράψαμε στην ενότητα 2.1.1.

Αρχικά παρουσιάζουμε 4 διαφορετικές επιθέσεις πλημμύρας, οι οποίες αποκτούν αυτόν τον χαρακτηρισμό καθώς προσπαθούν να κατακλύσουν το θύμα με κάποιας μορφής δικτυακή κίνηση.

1) Πλημμύρα ICMP (ICMP Flood ή Ping Flood)

Πρόκειται για Ογκομετρική Επίθεση DDoS με στόχο το εύρος ζώνης των δικτυακών συνδέσεων του θύματος. Γίνεται χρήση του πρωτοκόλλου Internet Control Message Protocol (ICMP). Κατά τη διάρκεια της επίθεσης αποστέλνονται πολλά μηνύματα ICMP echo request, γνωστά και ως ping, με εξαιρετικά γρήγορο ρυθμό, προς το δίκτυο του θύματος χωρίς να περιμένει απάντηση ο θύτης, ξέροντας όμως ότι πιθανώς θα υπάρξει απάντηση με ίσο αριθμό πακέτων. Αν το θύμα απαντήσει στα μηνύματα αυτά η επίθεση ενισχύεται καθώς καταναλώνεται ακόμη περισσότερο εύρος ζώνης, τόσο εισερχόμενο όσο και εξερχόμενο.

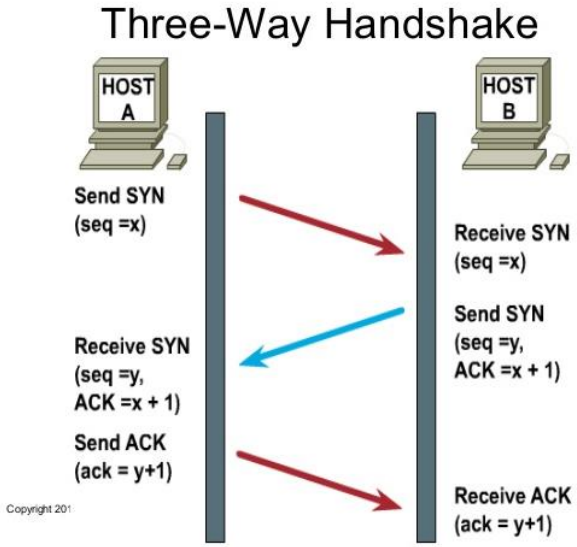
Για να επιτύχει αυτή η επίθεση θα πρέπει ο επιτιθέμενος, αν δεν είναι κατανεμημένη η επίθεση, να διαθέτει μεγαλύτερο εύρος ζώνης (bandwidth) από το θύμα, δηλαδή η σύνδεσή του με το διαδίκτυο να είναι πιο γρήγορη σε σχέση με του θύματος.

Ο κύριος τρόπος αντιμετώπισης τέτοιων επιθέσεων είναι η χρήση firewall που αποκόπτει μηνύματα ping προερχόμενα από διευθύνσεις IP εκτός του δικτύου.

2) Πλημμύρα TCP SYN (SYN Flood)

Πρόκειται για DDoS Επίθεση Πρωτοκόλλου (Protocol Attack), με στόχο να καταναλώσει τους πόρους ενός εξυπηρετητή (server)-θύματος και να τον καταστήσει ανίκανο να ανταποκριθεί. Η επίθεση αυτή εκμεταλλεύεται μια γνωστή αδυναμία στην ακολουθία σύνδεσης TCP, η οποία ακολουθία ονομάζεται τριπλή χειραψία (three-way handshake).

Παρουσιάζουμε εδώ σύντομα τη διαδικασία three-way handshake του πρωτοκόλλου TCP, βλέπε και σχήμα 1. Αρχικά ο πελάτης (client), host A στο σχήμα, οποίος επιθυμεί να συνδεθεί με τον εξυπηρετητή (server), host B στο σχήμα, στέλνει ένα πακέτο TCP, με σημαία SYN. Ο εξυπηρετητής λαμβάνει το πακέτο αυτό και με τη σειρά του, απαντάει με TCP πακέτο, με σημαίες SYN και ACK, αν επιτρέπει στον πελάτη τη σύνδεση με αυτόν ή με TCP πακέτο, με σημαίες ACK και RST, αν δεν επιτρέπει στον πελάτη τη σύνδεση με αυτόν. Τέλος για να ολοκληρωθεί η τριπλή χειραψία ο πελάτης λαμβάνει το πακέτο και στην περίπτωση που του επιτρέπει ο εξυπηρετητής τη σύνδεση, γίνεται η επιβεβαίωση, με την αποστολή, από τον πελάτη, ενός πακέτου TCP με σημαία ACK. Έτσι η σύνδεση έχει ολοκληρωθεί και μπορεί να αρχίσει η ανταλλαγή δεδομένων.



Σχήμα 1: Η τριπλή χειραψία (three-way handshake) του πρωτοκόλλου TCP.
Πηγή [8]

Κατά την επίθεση με Πλημμύρα SYN, ο επιτιθέμενος επιχειρεί να συνδεθεί στον εξυπηρετητή, πολλές φορές σε μικρό χρονικό διάστημα, στέλνοντας πακέτα TCP με σημαία SYN, χωρίς όμως να στέλνει επιβεβαίωση της σύνδεσης στη συνέχεια, με πακέτα με σημαία ACK. Ο εξυπηρετητής δεσμεύει χώρο στη μνήμη του καθώς περιμένει, για κάποιο χρονικό διάστημα, την επιβεβαίωση κάθε ημι-ανοιχτής σύνδεσης (half-open connection) και σαν αποτέλεσμα εξαντλεί τους πόρους του και οδηγείται σε μη-αποκρίσιμη κατάσταση (unresponsive).

3) Πλημμύρα UDP (UDP Flood)

Πρόκειται για Ογκομετρική Επίθεση DDoS, παρόμοιας λογικής με την Πλημμύρα ICMP, με στόχο το εύρος ζώνης των δικτυακών συνδέσεων του θύματος. Σε αυτήν την περίπτωση γίνεται η χρήση του πρωτοκόλλου User Datagram Protocol (UDP), ενός δικτυακού πρωτοκόλλου του επιπέδου μεταφοράς, το οποίο δεν απαιτεί σύνδεση (sessionless).

Κατά τη διάρκεια της επίθεσης, ο επιτιθέμενος στέλνει πολλά σε πλήθος πακέτα UDP, σε τυχαίες θύρες προορισμού του θύματος, αναγκάζοντας έτσι το θύμα να ελέγχει επανειλημμένα αν ακούει κάποια εφαρμογή σε κάθε θύρα προορισμού. Όταν δεν υπάρχει κάποια εφαρμογή, η οποία να ακούει στη θύρα προορισμού, το θύμα απαντά με ένα μήνυμα λάθους ICMP Destination Unreachable. Σαν αποτέλεσμα καταναλώνεται το εύρος ζώνης του θύματος αλλά και άλλοι πόροι του. Ο θύτης μάλιστα φροντίζει, τις περισσότερες φορές, να θέσει ψευδείς διευθύνσεις IP αποστολέα (IP spoofing), ώστε να ανωνυμοποιήσει την επίθεση, αλλά και να σιγουρευτεί ότι δεν τον φτάνουν τα πακέτα απάντησης ICMP.

Ο πιο εύκολος, αν και όχι ο πλέον καλύτερος, τρόπος αντιμετώπισης τέτοιων επιθέσεων είναι ο ορισμός ορίων, από το λειτουργικό σύστημα, στο ρυθμό των ICMP απαντήσεων. Άλλες πάλι λύσεις περιλαμβάνουν τη χρήση firewall, τα οποία μπλοκάρουν τα πακέτα UDP. Τέτοιο αυστηρό φιλτράρισμα όμως έχει αντίκτυπο και στη νόμιμη-καλόβουλη κίνηση του δικτύου.

4) Πλημμύρα HTTP (HTTP Flood)

Σε αυτήν την DDoS Επίθεση Επιπέδου Εφαρμογής (Application Layer Attack), ο θύτης αποστέλλει φαινομενικά καλόβουλα αιτήματα HTTP GET ή POST σε έναν εξυπηρετητή ή μία εφαρμογή. Σε κάθε αίτημα επιδιώκεται να διαμοιραστούν όσο το δυνατόν περισσότεροι πόροι του θύματος. Έτσι η επίθεση, με τα αιτήματα αυτά, προσπαθεί να αναγκάσει τον εξυπηρετητή-θύμα να εκτελέσει πολύπλοκες υπολογιστικά διαδικασίες οι οποίες χρησιμοποιούν όλους τους διαθέσιμους πόρους. Επιθέσεις τέτοιου είδους αν και είναι ογκομετρικές σε ένα βαθμό, δεν απαιτούν τόσους πόρους και εύρος ζώνης (bandwidth) από την πλευρά του θύτη, όπως άλλου είδους πλημμύρες. Απαιτούν όμως καλή γνώση του συστήματος του θύματος για να είναι αποδοτικές.

Συνήθως τέτοιες DDoS επιθέσεις προέρχονται από ένα botnet ή ‘’zombie army”, δηλαδή υπολογιστές με δυνατότητα σύνδεσης στο διαδίκτυο, οι οποίοι έχουν μολυνθεί από κακόβουλο λογισμικό (malware), όπως Δούρειους Ίππους (Trojan Horses).

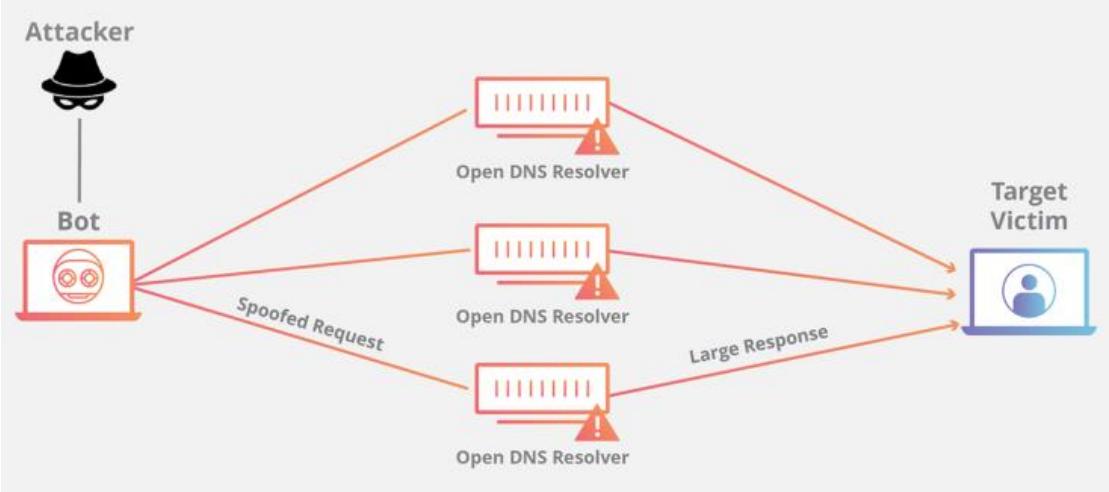
Αυτού του είδους η πλημμύρα δε χρησιμοποιεί λάθος σχηματισμένα πακέτα, με ψευδείς διευθύνσεις αποστολέα (IP spoofing), ούτε κάνει χρήση τεχνικών ανάκλασης (reflection techniques). Καθώς τα αιτήματα HTTP είναι νόμιμα και δε διαφέρουν σημαντικά από την καλόβουλη κίνηση, η αναγνώριση και η αντιμετώπιση μιας HTTP Πλημμύρας πρόκειται, σύμφωνα με το [4], για μια από τις πιο προχωρημένες δοκιμασίες ασφάλειας για τους εξυπηρετητές και τις εφαρμογές, χωρίς να εκμεταλλεύεται κάποια ιδιαίτερη παράβλεψη ή αδυναμία του συστήματος. Παραδοσιακές τεχνικές αναγνώρισης επιθέσεων με έλεγχο του ρυθμού εισερχόμενων αιτημάτων δεν είναι τόσο αποτελεσματικές. Η πιο αποδοτική μέθοδος αντιμετώπισης της επίθεσης περιλαμβάνει, σύμφωνα με το [4], τη χρήση βάσεων με πληροφορίες φήμης των διευθύνσεων IP (IP reputation) και παρακολούθηση δικτυακής κίνησης (traffic profiling).

Στη συνέχεια παρουσιάζουμε δύο επιθέσεις Ενίσχυσης (Amplification Attacks). Ο χαρακτηρισμός Επίθεση Ενίσχυσης προκύπτει καθώς το θύμα δέχεται πολύ μεγαλύτερη

κίνηση από αυτή που στέλνει ο θύτης, γεγονός που επιτυγχάνεται με την εκμετάλλευση κάποιων χαρακτηριστικών λειτουργίας των πρωτοκόλλων που χρησιμοποιεί η επίθεση.

1) DNS Ενίσχυση (DNS Amplification)

Η επίθεση αυτή εκμεταλλεύεται το πρωτόκολλο DNS για να προκαλέσει μια Ογκομετρική Επίθεση DDoS, με σκοπό να εξαντλήσει το εύρος ζώνης των δικτυακών συνδέσεων του θύματος. Ο επιτιθέμενος στέλνει πολλά ερωτήματα DNS προς ανοιχτούς DNS resolvers, χρησιμοποιώντας όμως ως διεύθυνση αποστολής των ερωτημάτων, τη διεύθυνση IP του θύματος (spoofed IP address). Έτσι το θύμα λαμβάνει την απάντηση των DNS resolvers αντί του θύτη, βλέπε και σχήμα 2. Τα ερωτήματα αυτά μάλιστα είναι έτσι κατασκευασμένα ώστε να απαιτούν όσο το δυνατόν μεγαλύτερες απαντήσεις, όπως για παράδειγμα ερωτήματα τύπου ''ANY''. Σαν αποτέλεσμα το δίκτυο του θύματος κατακλύζεται από πακέτα UDP, αρκετά μεγαλύτερου όγκου από την αρχική κίνηση του θύτη και αδυνατεί να προσφέρει απρόσκοπτα τις υπηρεσίες του (Denial of Service).



Σχήμα 2: Σχηματική απεικόνιση μιας επίθεσης DNS Ενίσχυσης.
Πηγή [6]

Η πρόληψη τέτοιων επιθέσεων σύμφωνα με το [6], γίνεται μειώνοντας τον αριθμό των ''φτωχά ρυθμισμένων'' DNS resolvers. Κάθε DNS resolver ιδανικά θα ήταν ρυθμισμένος ώστε να απαντάει μόνο σε ερωτήματα προερχόμενα από έμπιστες πηγές. Επίσης μια καλή μέθοδος πρόληψης είναι η ρύθμιση των Internet Service Providers (ISPs), ώστε να απορρίπτουν κίνηση προερχόμενη από το εσωτερικό τους με ψευδείς IP διευθύνσεις αποστολής (spoofed IP address). Τρόποι αντιμετώπισης μιας επίθεσης ενίσχυσης περιλαμβάνουν συνδυασμό από φιλτράρισμα της εισερχόμενης κίνησης στο δίκτυο και ''overprovisioning'', δηλαδή να υπάρχει τρόπος διαφορετικής δρομολόγησης της κίνησης (additional bandwidth).

2) NTP Amplification

Πρόκειται για Ογκομετρική Επίθεση DDoS, η οποία έχει ως στόχο να κατακλύσει το θύμα με κίνηση UDP, με σκοπό να εξαντλήσει το εύρος ζώνης των δικτυακών συνδέσεών του. Αυτή η επίθεση κάνει χρήση του πρωτοκόλλου Network Time Protocol (NTP).

Το πρωτόκολλο NTP, σύμφωνα με το [4], είναι ένα από τα πιο παλιά δικτυακά πρωτόκολλα, το οποίο χρησιμοποιείται από συσκευές συνδεδεμένες στο διαδίκτυο για

συγχρονισμό των ρολογιών τους. Επίσης παλιότερες εκδόσεις του NTP παρέχουν δυνατότητες monitoring ρωτώντας έναν NTP server για το μετρητή κίνησης. Πιο συγκεκριμένα η εντολή ‘’monlist’’ επιτρέπει την επιστροφή μιας λίστας με τα τελευταία 600 μηχανήματα που συνδέθηκαν και ρώτησαν τον NTP server. Στην πιο βασική της μορφή, η επίθεση περιλαμβάνει την αποστολή αιτημάτων ‘’get monlist’’ από το θύτη προς ένα NTP server, χρησιμοποιώντας όμως ως διεύθυνση αποστολής του αιτήματος τη διεύθυνση IP του θύματος (spoofed IP address). Έτσι το θύμα λαμβάνει την απάντηση του NTP server αντί του θύτη, εξού και ο χαρακτηρισμός επίθεση ανάκλασης (reflection attack).

Καθώς πρόκειται για επίθεση ενίσχυσης το θύμα δέχεται πολύ μεγαλύτερη κίνηση από αυτή που στέλνει ο θύτης, με αναλογία αιτημάτων-απαντήσεων από 1:20 έως 1:200. Έτσι, οποιοσδήποτε αποκτήσει μια λίστα από ανοιχτούς NTP servers μπορεί να δημιουργήσει μια καταστροφική, υψηλού όγκου, μεγάλου εύρους ζώνης DDoS επίθεση.

Δεδομένου του μεγάλου όγκου της και το γεγονός ότι η επίθεση συνίσταται από φαινομενικά νόμιμη κίνηση προερχόμενη από έγκυρους εξυπηρετητές καθίσταται δύσκολη η αντιμετώπισή της. Τρόποι αντιμετώπισης περιλαμβάνουν συνδυασμούς από φιλτράρισμα της εισερχόμενης κίνησης στο δίκτυο και ‘’overprovisioning’’, δηλαδή να υπάρχει τρόπος διαφορετικής δρομολόγησης της κίνησης (additional bandwidth).

Τέλος παρουσιάζουμε κάποιες άλλες συχνά εμφανιζόμενες επιθέσεις Άρνησης Υπηρεσίας, διαφορετικές από τις επιθέσεις Πλημμύρας και Ενίσχυσης που περιγράψαμε μέχρι στιγμής.

Επίθεση Slowloris

Είναι μια DDoS Επίθεση του Επιπέδου Εφαρμογής (Application Layer Attack). Ο επιτιθέμενος στέλνει, με αργό ρυθμό, μικρά κομμάτια από αιτήματα HTTP στον εξυπηρετητή (server) του θύματος, χωρίς να ολοκληρώνεται ποτέ κάποιο αίτημα. Κάθε επόμενο κομμάτι ενός αιτήματος καταφθάνει στον εξυπηρετητή αμέσως πριν περάσει ο χρόνος αναμονής, λήξει και τελικά ακυρωθεί το HTTP αίτημα. Έτσι ο εξυπηρετητής περιμένει όλα τα κομμάτια ενός HTTP αιτήματος να καταφτάσουν και τελικά κρατάει πολλές ανοιχτές ‘’συνδέσεις’’ με αποτέλεσμα να ξεπερνάται η ταυτόχρονη δυνατότητα συνδέσεών του, γεγονός που οδηγεί σε άρνηση χειρισμού νόμιμων αιτημάτων HTTP από κανονικούς χρήστες.

Ping of Death (PoD)

Πρόκειται για DoS Επίθεση Πρωτοκόλλου (Protocol Attack) με κάποια παρόμοια χαρακτηριστικά με την ICMP Πλημμύρα. Στόχος όμως της επίθεσης δεν είναι τόσο το εύρος ζώνης των δικτυακών συνδέσεων του θύματος αλλά οι πόροι κάποιου μηχανήματος. Γίνεται και πάλι χρήση του πρωτοκόλλου ICMP. Κατά τη διάρκεια της επίθεσης αποστέλνονται πολλά πακέτα-θραύσματα (fragments) κακοσχηματισμένων (malformed) μηνυμάτων ICMP echo request (pings).

Το μέγιστο επιτρεπτό μέγεθος πακέτου IP, συμπεριλαμβανομένου και της επικεφαλίδας, είναι 65.535 bytes. Όμως το στρώμα ζεύξης δεδομένων (data link layer) έχει όριο στο μέγιστο μέγεθος πλαισίου (frame), 1500 bytes για παράδειγμα για δίκτυο Ethernet. Σαν αποτέλεσμα IP πακέτα με μεγαλύτερο μέγεθος από 1500 bytes θρυμματίζονται σε μικρότερα πακέτα-θραύσματα (fragments). Τα θραύσματα αυτά επανασυναρμολογούνται σε ένα πακέτο αφού φτάσουν στον τελικό προορισμό. Κατά την επίθεση PoD στέλνονται κακοσχηματισμένα πακέτα-θραύσματα τα οποία, έπειτα από τη συναρμολόγησή τους στον τελικό προορισμό,

οδηγούν στην δημιουργία πακέτων IP με περισσότερα από 65.535 bytes. Σαν συνέπεια σε πολλά συστήματα η επανασυναρμολόγηση αυτή οδηγεί σε προβλήματα όπως memory overflow και crash.

Πλέον αυτή η επίθεση αντιμετωπίζεται επαρκώς από το λειτουργικό των περισσότερων συστημάτων με τη χρήση ελέγχων κατά τη διαδικασία επανασυναρμολόγησης και για το λόγο αυτό δεν είναι τόσο διαδεδομένη όπως στο παρελθόν. Τη θέση της, σύμφωνα με τα [4] και [6], έχει πάρει η συχνότερα εμφανιζόμενη ICMP Πλημμύρα, την οποία περιγράψαμε προηγουμένως.

Άλλοι τρόποι αντιμετώπισης μιας επίθεσης PoD είναι η χρήση firewall που αποκόπτει μηνύματα ping προερχόμενα από διευθύνσεις IP εκτός του δικτύου. Πιο έξυπνοι τρόποι αντιμετώπισης αυτής της επίθεσης περιλαμβάνουν το επιλεκτικό μπλοκάρισμα θρυμματισμένων ping και επιτρέπουν τα καλόβουλα ping να περάσουν ανεμπόδιστα.

Zero-day DDoS Επιθέσεις

Με αυτή την ονομασία αναφερόμαστε σε DDoS επιθέσεις, αγνώστων χαρακτηριστικών, οι οποίες εκμεταλλεύονται νέες αδυναμίες και τρωτότητες των συστημάτων, που προκύπτουν συνήθως έπειτα από την αναβάθμιση ή ενημέρωση κάποιου λογισμικού και δεν έχουν αναγνωρισθεί και καλυφθεί ακόμη.

2.1.3 Παρακολούθηση Δικτυακής κίνησης – Το Πρωτόκολλο Netflow

Το πλήθος των διαφορετικών επιθέσεων που υπάρχουν αναγκάζει τους διαχειριστές των δικτύων να παρακολουθούν τη δικτυακή κίνηση. Υπάρχουν πολλά διαφορετικά πρωτόκολλα παρακολούθησης (monitoring) της κίνησης ενός δικτύου. Ένα από αυτά είναι και το πρωτόκολλο Netflow.

Το πρωτόκολλο Netflow δημιουργήθηκε από τη Cisco και αρχικά χρησιμοποιήθηκε σε δρομολογητές και δικτυακό εξοπλισμό της ίδιας εταιρίας. Πλέον υποστηρίζεται από τις συσκευές των περισσότερων κατασκευαστών δικτυακού εξοπλισμού.

Το πρωτόκολλο αυτό συλλέγει δικτυακή κίνηση, η οποία εξέρχεται ή εισέρχεται σε μια διαπροσωπία (interface) ενός δρομολογητή. Η συλλογή της κίνησης γίνεται με τη μορφή ροών. Μια ροή, σύμφωνα με το [9], ορίζεται ως ακολουθία πακέτων μιας κατεύθυνσης, με κάποια κοινά χαρακτηριστικά, που διέρχονται μέσω μιας συσκευής ενός δικτύου. Η παρακολούθηση της κίνησης με τη μορφή ροών και όχι πακέτων, έχει το πλεονέκτημα ότι μπορεί να ελεγχθεί στο ίδιο χρονικό διάστημα πολύ μεγαλύτερος όγκος δεδομένων και να αποθηκευτεί με μεγαλύτερη ευκολία και μικρότερες απαιτήσεις σε χώρο.

Υπάρχουν πολλές διαφορετικές εκδόσεις του πρωτοκόλλου Netflow. Οι πιο σημαντικές είναι οι εκδόσεις 5 και 9. Στην παρούσα διπλωματική εργασία επικεντρωνόμαστε στην έκδοση 9 που είναι και η πιο διαδεδομένη έκδοση και αυτή που χρησιμοποιήθηκε. Ακολούθως παρουσιάζουμε τα βασικά συστατικά στοιχεία, όπως αυτά αναφέρονται στο [9], ενός συστήματος παρακολούθησης κίνησης με το Netflow και τον τρόπο λειτουργίας του πρωτοκόλλου.

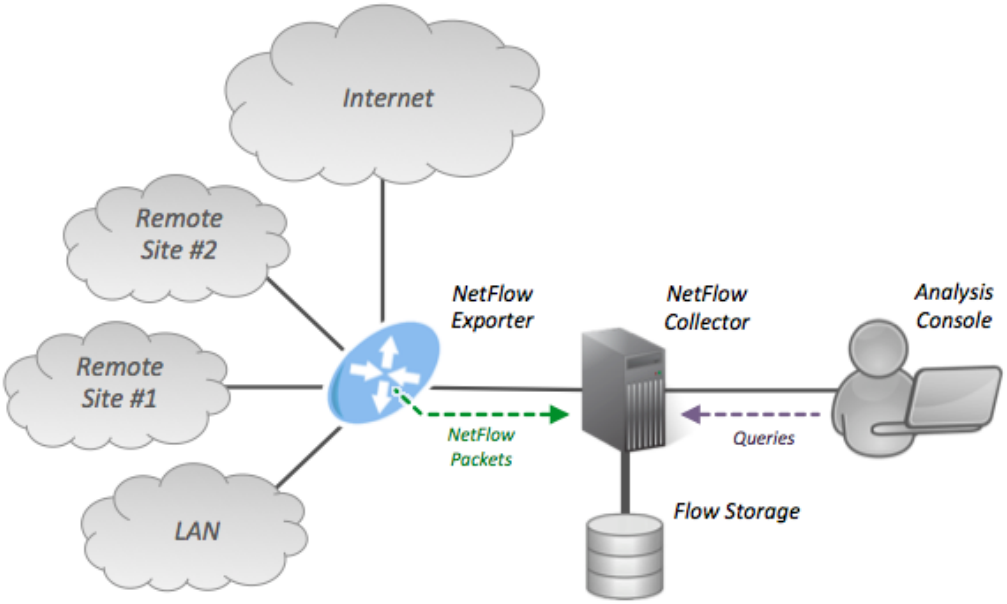
- **Εξαγωγέας Ροών (Flow Exporter):**

Μια συσκευή, για παράδειγμα ένας δρομολογητής, με ενεργοποιημένες τις υπηρεσίες Netflow, η οποία παρακολουθεί τα πακέτα IP που διέρχονται από ένα σημείο παρατήρησης

και δημιουργεί ροές IP από αυτά. Η πληροφορία από αυτές τις ροές εξάγεται, με τη μορφή Εγγραφών Ροής (Flow Records), στο Συλλέκτη Netflow (Netflow Collector).

- **Συλλέκτης Netflow (Netflow Collector):**

Ο Συλλέκτης δέχεται τις εγγραφές ροής από έναν ή περισσότερους Εξαγωγείς ροών, τις επεξεργάζεται και τις αποθηκεύει σε κατάλληλο αποθηκευτικό χώρο. Η αποθήκευση των ροών μπορεί να γίνει όπως αυτές δημιουργούνται ή έπειτα από άθροιση των δεδομένων δύο ή περισσότερων ροών (aggregation).



Σχήμα 3: Αρχιτεκτονική ενός συστήματος συλλογής, επεξεργασίας και αποθήκευσης δικτυακής κίνησης με τη χρήση του Netflow.
Πηγή [10]

Ενεργή και Ανενεργή Ροή IP - Λήξη Ροής:

Μια Ροή IP θεωρείται Ανενεργή όταν δεν έχει παρατηρηθεί στο σημείο παρατήρησης, για χρόνο μεγαλύτερο από κάποιο χρονικό όριο (timeout), κάποιο πακέτο που να εντάσσεται στη ροή αυτή. Αντιθέτως αν καταφτάσει ένα τέτοιο πακέτο εντός του χρονικού ορίου η ροή θεωρείται Ενεργή. Μια ροή εξάγεται από τον Εξαγωγέα προς το Συλλέκτη στις ακόλουθες περιπτώσεις:

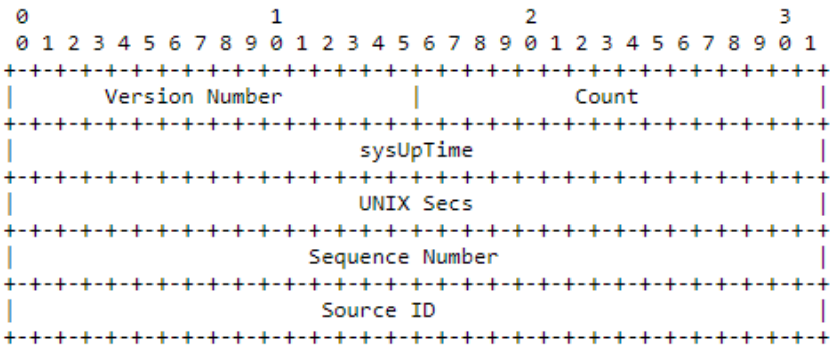
1) Αν ο Εξαγωγέας εντοπίσει το τέλος μιας ροής όπως για παράδειγμα συμβαίνει με μια σημαία FIN ή RST που υποδεικνύει το τέλος μιας σύνδεσης TCP.
2) Αν η ροή είναι ανενεργή για ένα συγκεκριμένο χρονικό διάστημα. Αυτό το χρονικό διάστημα έλλειψης δραστηριότητας συνήθως ρυθμίζεται στον Εξαγωγέα και μπορεί να έχει μέχρι και μηδενική τιμή.
3) Αν η ροή διαρκεί μεγάλο χρονικό διάστημα, έτσι ώστε ο Εξαγωγέας να εξάγει ροές σε τακτά χρονικά διαστήματα. Το χρονικό διάστημα αυτό ορίζεται στον Εξαγωγέα.
4) Αν ο Εξαγωγέας αντιμετωπίσει εσωτερικούς περιορισμούς, όπως για παράδειγμα χαμηλή μνήμη, οπότε και η ροή αναγκάζεται να λήξει πρόωρα.

2.1.4 Δομή των πακέτων του Πρωτοκόλλου Netflow v9

Τα Πακέτα Εξαγωγής του Netflow ενθυλακώνονται σε UDP δεδομενογράμματα (datagrams). Όμως το Netflow version 9 σχεδιάστηκε ώστε να είναι ανεξάρτητο από το πρωτόκολλο μεταφοράς που χρησιμοποιείται. Έτσι μπορεί να λειτουργήσει και πάνω από πρωτόκολλα όπως το Stream Control Transmission Protocol (SCTP). Ακολουθεί η δομή των πακέτων του πρωτοκόλου Netflow v9, όπως αυτή παρουσιάζεται επίσημα στο [9].

- **Πακέτο Εξαγωγής (Export Packet):**

Πακέτο Εξαγωγής ονομάζεται ένα πακέτο που πηγάζει στον Εξαγωγέα και μεταφέρει τις Εγγραφές Ροής στον Συλλέκτη. Αποτελείται από μια επικεφαλίδα, η οποία βρίσκεται πάντοτε στην αρχή του πακέτου και περιέχει βασικές πληροφορίες όπως την έκδοση του Netflow, το πλήθος των εγγραφών, οι οποίες περιέχονται στο πακέτο και τον αριθμό σειράς (Sequence Number), βλέπε πίνακα 1. Επίσης ένα πακέτο Εξαγωγής περιέχει ένα πλήθος από Flowsets, βλέπε πίνακα 2.

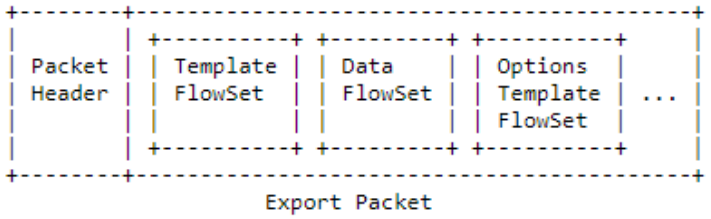


Πίνακας 1: Δομή της Επικεφαλίδας ενός Export Packet του πρωτοκόλλου Netflow v9 και τα πεδία που περιέχει.
Πηγή [9]

- **Σύνολο Εγγραφών (FlowSet):**

Πρόκειται για μία συλλογή από Εγγραφές Ροής (Flow Records) με παρόμοια δομή. Ο Εξαγωγέας ομαδοποιεί Εγγραφές Ροής και τις στέλνει όλες μαζί στον Συλλέκτη με τη μορφή FlowSets. Υπάρχουν τρία διαφορετικά είδη FlowSet:

1) Template FlowSet, το οποίο περιέχει ένα σύνολο από Template Records, βλέπε πίνακα 3.
2) Data FlowSet, το οποίο περιέχει ένα σύνολο από Data Records ή Options Data Records, βλέπε πίνακα 4.
3) Options Template FlowSet, το οποίο περιέχει ένα σύνολο από Options Template Records, βλέπε πίνακα 5.

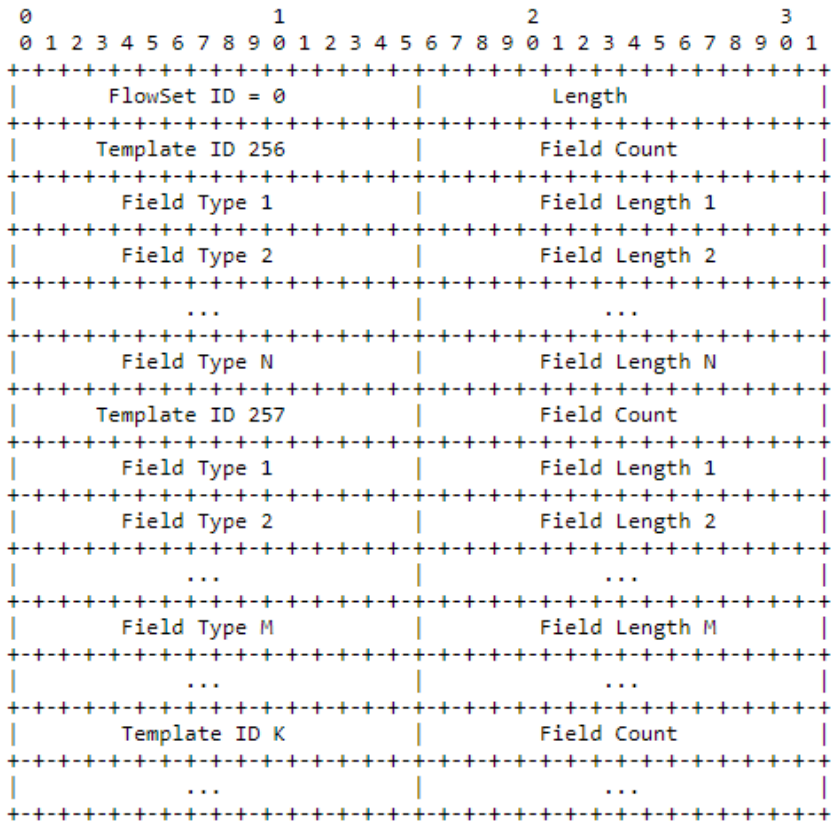


Πίνακας 2: Δομή ενός Export Packet του πρωτοκόλλου Netflow v9, το οποίο περιέχει ένα πλήθος από FlowSets.
Πηγή [9]

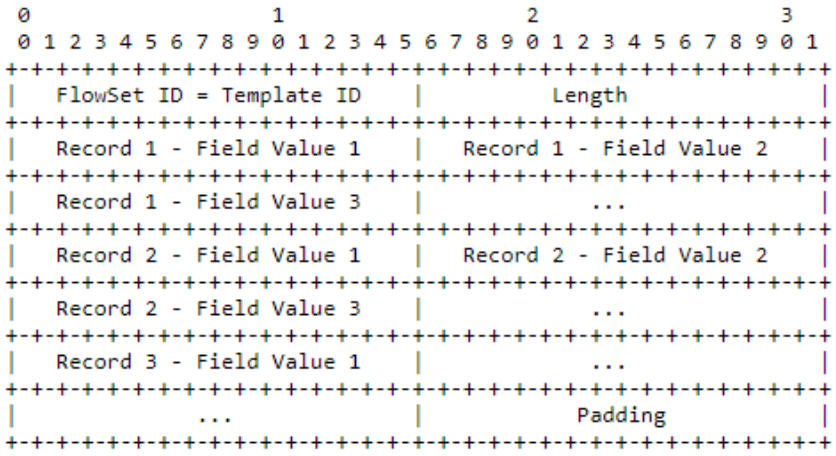
- **Εγγραφή Ροής (Flow Record):**

Πρόκειται για εγγραφή που περιέχει πληροφορίες για μια ροή IP, που παρατηρήθηκε σε ένα σημείο παρατήρησης, από τον Εξαγωγέα. Μία Εγγραφή Ροής έχει μία από τις παρακάτω ειδικές μορφές:

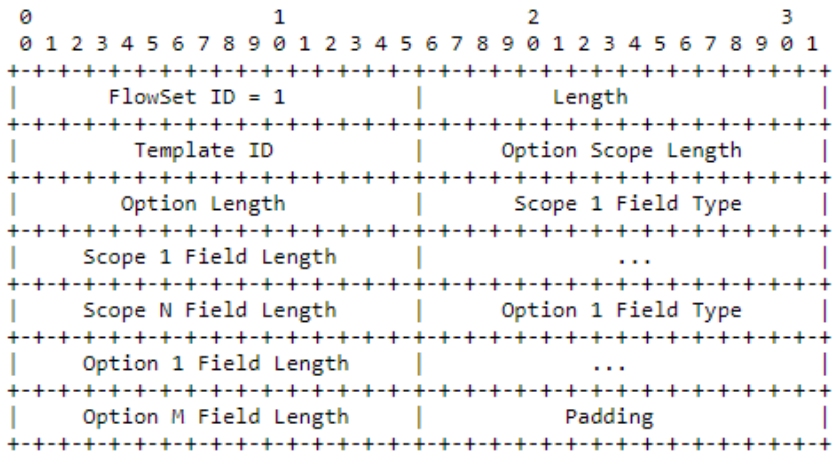
1) Template Record, η οποία ορίζει τη δομή και την ερμηνεία των πεδίων που περιέχονται σε μια Data Record, βλέπε πίνακα 3.
2) Data Record, η οποία περιέχει τις τιμές των παραμέτρων μιας ροής IP, όπως αυτές ορίζονται σε μια Template Record, βλέπε πίνακα 4.
3) Options Template Record, η οποία ορίζει τη δομή και την ερμηνεία κάποιων επιπρόσθετων πεδίων, τα οποία περιέχονται σε μία Options Data Record, βλέπε πίνακα 5.
4) Options Data Record, η οποία περιέχει τις τιμές των επιπρόσθετων πεδίων, όπως αυτά ορίζονται σε μία Options Template Record, βλέπε πίνακα 4.



Πίνακας 3: Δομή ενός Template FlowSet ενός Export Packet του πρωτοκόλλου Netflow v9. Στο FlowSet αυτής της μορφής περιέχονται Template Records, K τέτοιες στον πίνακα αυτόν, οι οποίες καθορίζουν τον τύπο και το μήκος όλων των μεταβλητών που περιέχονται σε K Data Records σε ένα Data FlowSet που ακολουθεί. Έτσι κάθε Template Record περιέχει και ένα Template ID που βοηθά στην αντιστοίχιση με τα Data Records. Το Template FlowSet βελτιώνει την ελαστικότητα του πρωτοκόλλου Netflow καθώς επιτρέπει στον Collector να επεξεργάζεται Data Records με διαφορετικά πεδία, τα οποία περιγράφονται στο αντίστοιχο Template FlowSet.
Πηγή [9]



Πίνακας 4: Δομή ενός Data FlowSet ενός Export Packet του πρωτοκόλλου Netflow v9. Στο FlowSet αυτής της μορφής περιέχονται Data Records ή Options Data Records, 3 τέτοιες εγγραφές στον πίνακα αυτόν, οι οποίες καθορίζουν την τιμή όλων των μεταβλητών που περιεγράφηκαν σε κάποιο Template FlowSet ή Options Template FlowSet αντίστοιχα, το οποίο προηγήθηκε στο ίδιο Export Packet ή σε προγενέστερο Export Packet που έχει σταλεί ήδη στο συλλέκτη.
Πηγή [9]



Πίνακας 5: Δομή ενός Options Template FlowSet ενός Export Packet του πρωτοκόλλου Netflow v9. Στο FlowSet αυτής της μορφής περιέχονται Options Template Records, οι οποίες καθορίζουν τον τύπο και το μήκος όλων των ‘’επιπρόσθετων’’ μεταβλητών οι οποίες περιέχονται σε Data Records σε ένα αντίστοιχο Data FlowSet που ακολουθεί. Με τις Options Data Records, ενός Data FlowSet, αναφερόμαστε σε κάποιες ειδικές μεταβλητές που δε σχετίζονται άμεσα με τις παρατηρούμενες ροές IP, αλλά με τη ρύθμιση της λειτουργίας του Netflow και το πώς αυτό επεξεργάζεται συγκεκριμένα δεδομένα. Τέτοιες μεταβλητές για παράδειγμα μπορεί να είναι ο ρυθμός δειγματοληψίας μιας διαπροσωπίας (interface).
Πηγή [9]

2.2 Μηχανική Μάθηση και Νευρωνικά Δίκτυα

Σε αυτή την ενότητα παρουσιάζεται το θεωρητικό υπόβαθρο για τη μηχανική μάθηση και πιο συγκεκριμένα για τα Νευρωνικά Δίκτυα, τα Βαθιά Νευρωνικά Δίκτυα και τα σημαντικότερα είδη αυτών, τα βασικά τους δομικά στοιχεία, τις μεθόδους εκπαίδευσης αυτών και τον τρόπο χρήσης τους.

2.2.1 Τεχνικές Μηχανικής Μάθησης

Γενικά υπάρχουν πολλές διαφορετικές μέθοδοι και τεχνικές Μηχανικής Μάθησης. Όλες όμως εντάσσονται σε κάποιες ευρύτερες κατηγορίες. Η πρώτη διάκριση μπορεί να γίνει, όπως φαίνεται και στο σχήμα 1, σε Επιβλεπόμενη Μάθηση (Supervised Learning) και Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning).

Επιβλεπόμενη Μάθηση (Supervised Learning):
Κατά τη χρήση τεχνικών Επιβλεπόμενης Μάθησης, για ένα πρόβλημα κατηγοριοποίησης κάποιων δεδομένων (data), γνωρίζουμε εκ των προτέρων τον αριθμό από τις επιμέρους κατηγορίες στις οποίες θέλουμε να κατατάσσονται αυτά τα δεδομένα. Επίσης για τη φάση της εκπαίδευσης του μηχανισμού, ο οποίος εκτελεί την κατηγοριοποίηση, πρέπει να έχουμε στη διάθεση μας δείγματα (samples) για τα οποία γνωρίζουμε ήδη την κατηγορία στην οποία ανήκουν. Βάση αυτών των δειγμάτων, κάνοντας χρήση επαναληπτικών μεθόδων, το σύστημα βελτιώνει όλο και περισσότερο το σφάλμα κατά την κατηγοριοποίηση διαφορετικών δεδομένων. Βλέπε και ενότητες 2.2.5 έως 2.2.7

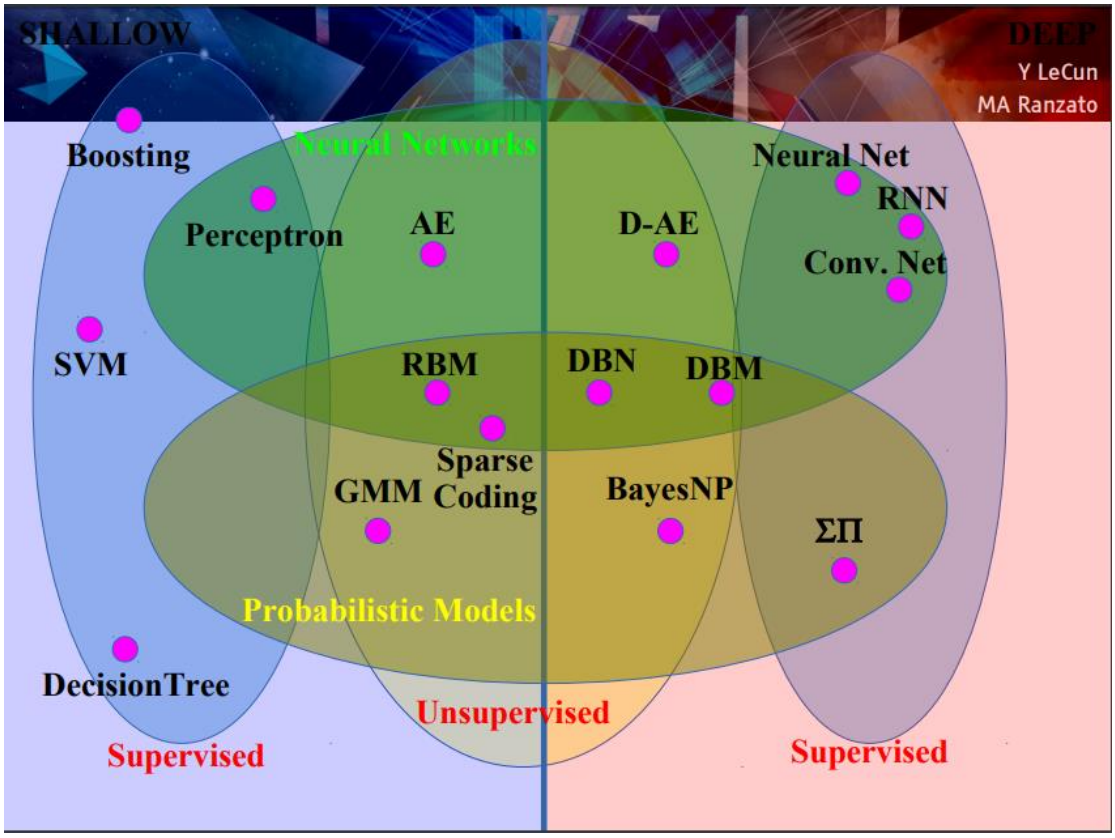
Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning):
Αντίθετα κατά τη χρήση τεχνικών Μη Επιβλεπόμενης Μάθησης στόχος είναι η ανακάλυψη πιθανής δομής που μπορεί να κρύβεται πίσω από μη χαρακτηρισμένα δεδομένα. Δηλαδή δε γνωρίζουμε εκ των προτέρων πόσες και ποιες είναι οι κατηγορίες που θέλουμε να χωρίσουμε τα δεδομένα που διαθέτουμε και είναι αρμοδιότητα του μηχανισμού να προσδιορίσει τις επιμέρους κατηγορίες, στις οποίες μπορούν να χωριστούν τα δεδομένα, βάσει των διαφορετικών χαρακτηριστικών που διαθέτουν. Εφόσον τα παραδείγματα τα οποία χρησιμοποιούνται δεν είναι χαρακτηρισμένα, δεν υπάρχει σφάλμα ή σήμα ανταμοιβής για να αξιολογηθούν οι πιθανές λύσεις.

Ένας άλλος βασικός χαρακτηρισμός των τεχνικών μηχανικής μάθησης, που φαίνεται και στο σχήμα 1, είναι σε Ρηχή Μάθηση (Shallow Learning) και σε Βαθιά Μάθηση (Deep Learning).

Ρηχή Μάθηση (Shallow Learning):
Οι Ρηχές τεχνικές Μάθησης βασίζονται, για την εξαγωγή αποτελεσμάτων και προβλέψεων, κυρίως στα χαρακτηριστικά που δίνονται στο σύστημα μέσω των δεδομένων εισόδου.

Βαθιά Μάθηση (Deep Learning):
Αντίθετα οι Βαθιές τεχνικές Μάθησης προσπαθούν από τα δεδομένα εισόδου να εξάγουν καλύτερες αναπαραστάσεις αυτών, με καινούρια, πιο σύνθετα και συνδυαστικά χαρακτηριστικά. Ο κατασκευαστής του συστήματος εδώ, δεν καλείται υποχρεωτικά να έχει κάποια ιδιαίτερη και βαθύτερη γνώση για τα δεδομένα εισόδου. Το ίδιο το σύστημα έχει τη δυνατότητα να εξάγει τέτοια γνώση (Feature Engineering) για το σκοπό που θέλει να επιτύχει.

Τέλος μια ομαδοποίηση κάποιων τεχνικών Μηχανικής Μάθησης μπορεί να γίνει βάσει του κατασκευάσματος και των μαθηματικών μεθόδων που χρησιμοποιεί η τεχνική. Οπότε κάνουμε λόγο για τεχνικές που χρησιμοποιούν Πιθανοτικά Μοντέλα (Probabilistic Models) από την κλασσική θεωρία πιθανοτήτων και Νευρωνικά Δίκτυα (Neural Networks). Η παρούσα διπλωματική εργασία ασχολείται με τα Νευρωνικά Δίκτυα, των οποίων τα χαρακτηριστικά και η λειτουργία αναλύονται στις ακόλουθες ενότητες.

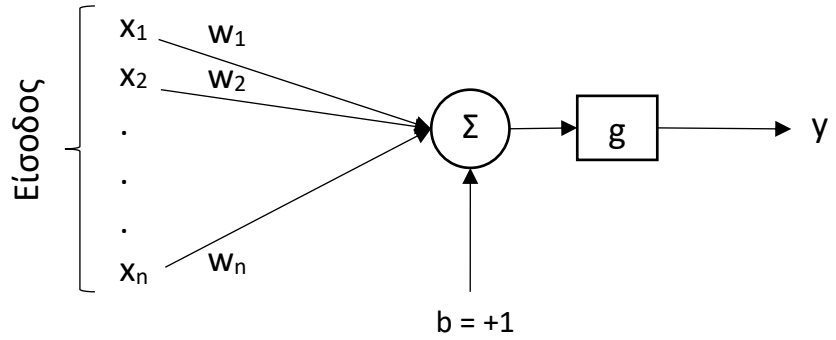


Σχήμα 4: Τεχνικές Μηχανικής Μάθησης και η κατηγοριοποίηση τους.
Πηγή [11]

2.2.2 Ο Νευρώνας (Perceptron)

Τα τεχνητά Νευρωνικά Δίκτυα (Neural Networks) είναι δίκτυα που αποτελούνται από απλούς υπολογιστικούς κόμβους, διασυνδεδεμένους μεταξύ τους. Το βασικό συστατικό τους στοιχείο είναι ο τεχνητός Νευρώνας (Perceptron), σχήμα 2. Κάθε νευρώνας δέχεται ως είσοδο ένα διάνυσμα $x_1,x_2,\ldots,x_n$ και ένα σταθερό όρο b (bias). Έπειτα συνδυάζει γραμμικά το διάνυσμα εισόδου πολλαπλασιάζοντάς κάθε στοιχείο του με αντίστοιχο βάρος $w_1,w_2,..,w_n$ που ο νευρώνας αποδίδει σε κάθε είσοδο και προσθέτοντας στο τελικό άθροισμα τον όρο b. Έπειτα αυτός ο γραμμικός συνδυασμός τροφοδοτείται σε μια συνάρτηση g, η οποία ονομάζεται Συνάρτηση Ενεργοποίησης (Activation Function) και το αποτέλεσμα αυτής παράγεται στην έξοδο y του νευρώνα. Οπότε έχουμε:

$$y = g\left(\sum_{i=1}^{n} w_i x_i + b\right)$$



Σχήμα 5: Τεχνητός Νευρώνας (Perceptron).

2.2.3 Συναρτήσεις Ενεργοποίησης (Activation Functions)

Ο κύριος λόγος ύπαρξης της συνάρτησης ενεργοποίησης είναι ώστε να μετασχηματίσει τη σχέση του διανύσματος εισόδου του νευρώνα με την έξοδο σε μια μη γραμμική σχέση. Με αυτόν τον τρόπο ο συνδυασμός πολλών νευρώνων μαζί, βλέπε επόμενες ενότητες, αποκτά τη δυνατότητα να περιγράψει διαφορετικές γραμμικές αλλά και μη γραμμικές συναρτήσεις. Σε αυτήν την ενότητα αναλύονται οι πιο ευρέως χρησιμοποιούμενες συναρτήσεις ενεργοποίησης σύμφωνα με το [13].

1) Γραμμική (Linear)

Όταν παραλείπεται η χρήση κάποιας Συνάρτησης Ενεργοποίησης οπότε η έξοδος του νευρώνα παραμένει ένας γραμμικός συνδυασμός του διανύσματος εισόδου.

2) Σιγμοειδής (Sigmoid)

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Η συνάρτηση αυτή χρησιμοποιείται κυρίως σε νευρώνες που ανήκουν στο επίπεδο εξόδου (Output Layer), βλέπε ενότητα 2.2.4.

3) Softmax

$$softmax(\boldsymbol{Z})_i = \frac{e^{z_i}}{\sum_{k=1}^{K} e^{z_k}}, \gamma\iota\alpha\ i = 1, \dots, K$$

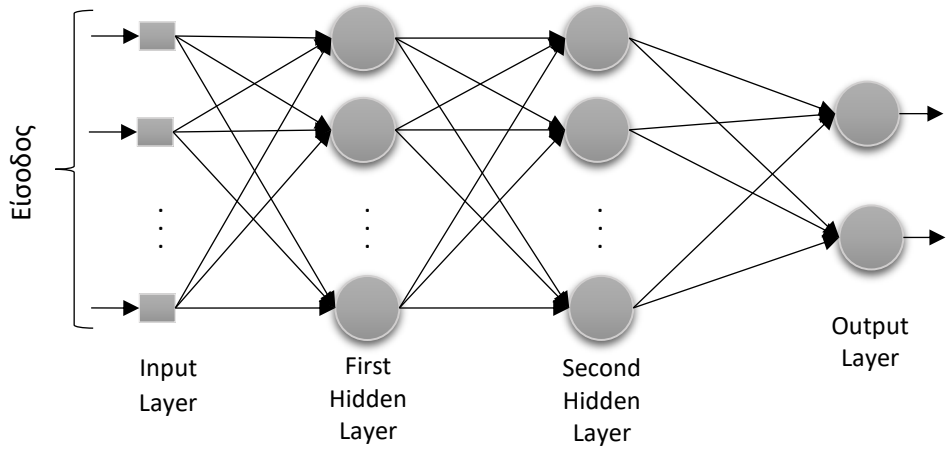
4) Rectified Linear Unit (ReLU)

$$y = \max(0, z)$$

2.2.4 Νευρώνας Πολλών Επιπέδων (Multilayer Perceptron, MLP)

Σε προηγούμενη ενότητα αναλύσαμε το Νευρώνα (Perceptron), ως το βασικό δομικό στοιχείο των Νευρωνικών Δικτύων. Πολλοί Νευρώνες συνδυάζονται μεταξύ τους οργανωμένοι σε επίπεδα (layers) και δημιουργούν ένα Νευρωνικό Δίκτυο.

Μια από τις πιο βασικές μορφές νευρωνικών δικτύων είναι ο Νευρώνας ή Νευρωνικό Δίκτυο Πολλών Επιπέδων (Multilayer Perceptron, MLP), σχήμα 3. Αρχικά το διάνυσμα εισόδου που ονομάζεται και Επίπεδο Εισόδου (Input Layer) τροφοδοτείται σε κάθε έναν από τους νευρώνες του πρώτου κρυφού Επιπέδου (Hidden Layer). Κρυφά Επίπεδα (Hidden Layers) ονομάζονται τα ενδιάμεσα επίπεδα νευρώνων του Δικτύου. Η έξοδος κάθε νευρώνα ενός κρυφού επιπέδου τροφοδοτείται εν συνεχεία σε κάθε νευρώνα του επόμενου επιπέδου. Τέλος η έξοδος κάθε νευρώνα του τελευταίου κρυφού επιπέδου τροφοδοτείται σε ένα ακόμα επίπεδο νευρώνων, το επίπεδο εξόδου (Output Layer).



Σχήμα 6: Παράδειγμα MLP με δυο κρυφά επίπεδα και 2 εξόδους στο επίπεδο εξόδου.

Αναγνωρίζουμε λοιπόν ως βασικές ιδιότητες των MLP δικτύων τις ακόλουθες:

- Η ροή της πληροφορίας είναι αποκλειστικά από τα αριστερά προς τα δεξιά (Feedforward Network), δεν έχουμε δηλαδή ανάδραση μεταξύ των διαφορετικών νευρώνων και επιπέδων.
- Υπάρχει πλήρη συνδεσιμότητα μεταξύ διαδοχικών επιπέδων (fully connected), δηλαδή η έξοδος κάθε νευρώνα ενός επιπέδου τροφοδοτείται σε κάθε νευρώνα του επόμενου επιπέδου. Το διάνυσμα που εισέρχεται σε ένα επίπεδο πολλαπλασιάζεται

με τα βάρη των συνδέσεων κάθε νευρώνα. Τα βάρη όλων των νευρώνων ενός επιπέδου μπορούν να τοποθετηθούν σε ένα πίνακα βαρών W ο οποίος τελικά πολλαπλασιάζεται με το διάνυσμα εισόδου του επιπέδου. Η έξοδος του παραπάνω πολλαπλασιασμού τροφοδοτείται στις συναρτήσεις ενεργοποίησης των νευρώνων, οι οποίες για ένα επίπεδο είναι η ίδια συνάρτηση για λόγους ομοιομορφίας και το αποτέλεσμα μας δίνει τις εξόδους του επιπέδου.

- Απαραίτητη προϋπόθεση είναι να υπάρχει ένα τουλάχιστον κρυφό επίπεδο με μη γραμμική συνάρτηση ενεργοποίησης στους νευρώνες.

Όπως προαναφέραμε, η επιλογή μη γραμμικών συναρτήσεων ενεργοποίησης για τους επιμέρους νευρώνες ενός MLP, δίνει τη δυνατότητα στο MLP μάθει και να προσομοιάσει τη λειτουργία και μη γραμμικών συναρτήσεων. Σύμφωνα μάλιστα με το Universal Approximation Theorem, προσφορά των Kurt Hornik, Maxwell Stinchcombe και Halbert White στο [12], ανεξάρτητα από το ποια συνάρτηση προσπαθούμε να μάθουμε, γνωρίζουμε ότι ένα μεγάλο MLP θα είναι σε θέση να την αναπαραστήσει. Ωστόσο δεν υπάρχει εγγύηση ότι ο αλγόριθμος μάθησης, βλέπε επόμενες ενότητες, θα είναι σε θέση να μάθει και να συγκλίνει στη συνάρτηση αυτή.

Καθώς τα MLP έχουν πολλά επίπεδα (>=2), σε τουλάχιστον ένα από τα οποία εφαρμόζω μη-γραμμικό μετασχηματισμό των χαρακτηριστικών (εισόδου), ανήκουν στην ευρύτερη κατηγορία των Βαθιών Νευρωνικών Δικτύων (Deep Neural Networks). Τέλος η χρήση MLP είναι μία από τις τεχνικές Επιβλεπόμενης Μάθησης (Supervised Learning).

2.2.5 Συνάρτηση Σφάλματος ή Κόστους – Maximum Likelihood Estimation

Η Επιβλεπόμενη Μηχανική Μάθηση των Νευρωνικών Δικτύων αποτελείται από την ακόλουθη διαδικασία. Το δίκτυο δέχεται δεδομένα εισόδου, για τα οποία γνωρίζουμε την κατάταξή τους, και προσπαθεί να προσομοιάσει τη λειτουργία μιας συνάρτησης, η οποία για τη δοσμένη είσοδο παράγει την αντίστοιχη έξοδο. Μέσω επαναληπτικών μαθηματικών διαδικασιών επιχειρεί λοιπόν να ‘’μάθει’’ τα κατάλληλα βάρη W του κάθε νευρώνα ώστε να προσεγγίσει την συνάρτηση που αναφέραμε με όσο το δυνατόν μικρότερο σφάλμα.

Συνάρτηση Σφάλματος:

Ορίζουμε ως συνάρτηση κόστους ή σφάλματος (cost function, loss function, error function) μια συνάρτηση η οποία εκφράζει την διαφοροποίηση της εξόδου του νευρωνικού δικτύου για μια είσοδο X σε σχέση με την επιθυμητή/ιδανική έξοδο Y. Η διαδικασία με την οποία τα νευρωνικά δίκτυα ‘’μαθαίνουν’’ περιλαμβάνει την ελαχιστοποίηση της τιμής της συνάρτησης σφάλματος.

Διαδικασία Εύρεσης κατάλληλης Συνάρτησης Σφάλματος για κάθε μοντέλο – νευρωνικό δίκτυο:

Στη βιβλιογραφία έχουν προταθεί διάφορες συναρτήσεις ως κατάλληλες συναρτήσεις σφάλματος. Μια από τις πρώτες προσεγγίσεις είναι η χρήση της νόρμας L1 ή μέσου απόλυτου σφάλματος (Mean Absolute Error).

$$J = \frac{\sum_{i=1}^{n} |y_{data} - y_{model}|}{n}$$

Μια από τις καλύτερες προσεγγίσεις για την εξαγωγή κατάλληλης συνάρτησης σφάλματος είναι η διαδικασία Maximum Likelihood Estimation ή Log Loss ή Cross-Entropy.

Έστω ότι έχουμε στη διάθεσή μας m δείγματα, X = {x(1), . . . , x(m)}, από δεδομένα (data) των οποίων γνωρίζουμε την κατηγορία στην οποία ανήκουν, δημιουργώντας έτσι μια κατανομή πιθανοτήτων $p_{data}(x)$. Ορίζουμε ως $p_{model}(x; \theta)$ μια παραμετρική οικογένεια από κατανομές πιθανοτήτων πάνω στον χώρο X με δείκτη το θ. Το $p_{model}(x; \theta)$ απεικονίζει κάθε ρύθμιση (configuration) x σε έναν πραγματικό αριθμό που προσπαθεί να ‘‘εκτιμήσει’’ το $p_{data}(x)$. Ορίζεται ως maximum likelihood estimator θ_{ML} για το θ, σύμφωνα με το [13]:

$$\theta_{ML} = \arg_\theta \max p_{model}(X; \theta)$$
$$= \arg_\theta max \prod_{i=1}^{m} p_{model}(x_i; \theta)$$

Για να αποκτήσουμε έναν ισοδύναμο maximum likelihood estimator με μια πιο βολική μορφή λογαριθμούμε την παραπάνω σχέση και έχουμε:

$$\theta_{ML} = \arg_\theta max \sum_{i=1}^{m} \log p_{model}(x_i; \theta)$$

Τέλος καθώς το argmax δεν αλλάζει όταν διαιρούμε με μια σταθερά, μπορούμε να διαιρέσουμε τον παραπάνω τύπο με m και έχουμε την ακόλουθη μορφή για τον maximum likelihood estimator:

$$\theta_{ML} = \arg_\theta \max E_{x \sim p_{data}} \log p_{model}(x; \theta)$$

Ένας τρόπος να ερμηνεύσουμε τον παραπάνω τύπο είναι βλέποντας τον σαν μείωση της διαφοροποίησης μεταξύ της εμπειρικής κατανομής p_{data}, που ορίζεται από τα δεδομένα μας, και από την κατανομή του μοντέλου p_{model}. Αυτή λοιπόν η διαφοροποίηση μεταξύ των δύο κατανομών εκφράζεται με την KL απόκλιση (KL divergence) που ορίζεται σύμφωνα με το [13] ως:

$$D_{KL}(p_{data} || p_{model}) = E_{x \sim p_{data}}[log p_{data}(x) - \log p_{\text{model}}(x)]$$

Για να ελαχιστοποιήσουμε αυτή την απόκλιση μεταξύ των αποτελεσμάτων του μοντέλου μας και των εμπειρικών δεδομένων αρκεί να ελαχιστοποιήσουμε τελικά το δεύτερο όρο του παραπάνω τύπου, τον οποίο μπορούμε να ελέγξουμε αλλάζοντας τις παραμέτρους του νευρωνικού μας δικτύου. Τον όρο αυτόν τον ονομάζουμε J και είναι μια κατάλληλη συνάρτηση κόστους ή σφάλματος (cost function, loss function, error function) για το μοντέλο μας.

$$J = -E_{x \sim p_{data}}[\log p_{\text{model}}(x)]$$

Η διαδικασία που περιγράψαμε αναφέρεται στη διεθνή βιβλιογραφία και στο [13] ως Maximum Likelihood Estimation ή αλλιώς Cross-entropy μεταξύ των δεδομένων εκπαίδευσης και της κατανομής που προκύπτει από το μοντέλο - νευρωνικό δικτύου που χρησιμοποιούμε.

Η ακριβής μορφή της συνάρτησης κόστους ή σφάλματος (cost function, loss function, error function) που προκύπτει από την παραπάνω διαδικασία εξαρτάται από το επίπεδο εξόδου του εκάστοτε νευρωνικού δικτύου και άρα την κατανομή πιθανοτήτων που ακολουθεί το διάνυσμα εξόδου Y.

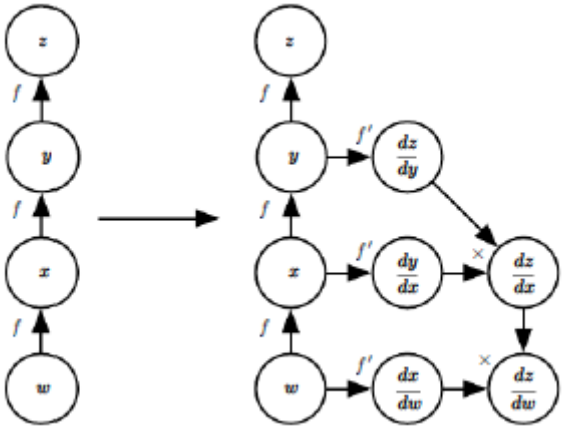
- Για γραμμική συνάρτηση ενεργοποίησης (linear output unit) των νευρώνων στο επίπεδο εξόδου έχουμε Κανονική ή Γκαουσιανή κατανομή (Gaussian Distribution). Ο τύπος της συνάρτησης σφάλματος που προκύπτει τελικά είναι ο τύπος του μέσου τετραγωνικού σφάλματος (Mean Squared Error, MSE):

$$J = \frac{\sum_{i=1}^{n}(y_{data} - y_{model})^2}{n}$$

- Για σιγμοειδή συνάρτηση ενεργοποίησης (sigmoid output unit) των νευρώνων στο επίπεδο εξόδου έχουμε κατανομή Bernouli.
- Για softmax output, που είναι και το ποιο διαδεδομένο output layer για ταξινομητές, έχουμε κατανομή Multinomial, που είναι γενίκευση της κατανομής Bernouli.

2.2.6 Διάδοση σφάλματος προς τα πίσω – Αλγόριθμος Backpropagation

Υπάρχουν πολλοί αλγόριθμοι εκπαίδευσης ενός νευρωνικού δικτύου. Κοινό όλων αυτών είναι ο υπολογισμός των πρώτων παραγώγων της συνάρτησης σφάλματος (Loss function) ως προς τα βάρη των νευρώνων του τελευταίου επιπέδου, και η διάδοση αυτών των παραγώγων, με τη χρήση του κανόνα της αλυσίδας, στα προηγούμενα επίπεδα του δικτύου που εκπαιδεύουμε, ώστε να έχουμε προσαρμογή των βαρών του νευρωνικού δικτύου και επομένως μείωση της τιμής της συνάρτησης σφάλματος. Ο αλγόριθμος που φροντίζει για αυτήν την ‘’προς τα πίσω’’ διάδοση της παραγώγου ονομάζεται backpropagation και βασίζεται στον κανόνα της αλυσίδας (chain rule) της μαθηματικής ανάλυσης. Το πως θα χρησιμοποιηθεί η παράγωγος αυτή από τα προηγούμενα επίπεδα για να προσαρμόσει τα βάρη του νευρωνικού δικτύου είναι ευθύνη του αλγορίθμου εκπαίδευσης που χρησιμοποιούμε, ο οποίος δεν πρέπει να συγχέεται με τον αλγόριθμο backpropagation.



Σχήμα 7: Σχηματική απεικόνιση της προς τα πίσω διάδοσης της παραγώγου μιας μεταβλητής z σε προηγούμενα επίπεδα ενός νευρωνικού δικτύου χρησιμοποιώντας τον κανόνα της αλυσίδας. Πηγή [13]

2.2.7 Αλγόριθμοι Εκπαίδευσης ή Βελτιστοποίησης Νευρωνικών δικτύων

Απαραίτητη φάση σε κάθε τεχνική Μηχανικής Μάθησης είναι η Εκπαίδευση. Έτσι και στα Νευρωνικά Δίκτυα έχουμε τη διαδικασία της εκπαίδευσης κατά την οποία ο αλγόριθμος

εκπαίδευσης δέχεται στην είσοδο τη συνάρτηση σφάλματος, την οποία έχουμε υπολογίσει πάνω στα δεδομένα εκπαίδευσης που διαθέτουμε, βλέπε ενότητα 2.2.5, και προσαρμόζει, σε κάθε επανάληψη, όλο και περισσότερο, τα βάρη των νευρώνων του δικτύου με σκοπό την ελαχιστοποίηση του σφάλματος εξόδου. Ακολουθούν οι πιο συχνά χρησιμοποιούμενοι αλγόριθμοι εκπαίδευσης.

1) Στοχαστική Κατάβαση Δυναμικού (Stochastic Gradient Descend)

Είναι η πιο ευρέως χρησιμοποιούμενη τεχνική Εκπαίδευσης στα Νευρωνικά Δίκτυα. Πρόκειται για μια επέκταση της μεθόδου Gradient Descent. Σε κάθε βήμα ο αλγόριθμος παίρνει ένα μικρό δείγμα m' (minibatch), από τα διαθέσιμα δεδομένα (samples) του συνόλου εκπαίδευσης, και προσπαθεί να προσεγγίσει τη μέθοδο Gradient Descent η οποία χρειάζεται κανονικά όλα τα δεδομένα. Έτσι επιτυγχάνεται μεγαλύτερη ταχύτητα εκπαίδευσης καθώς η συνάρτηση κόστους υπολογίζεται για πολύ λιγότερα δεδομένα.

Δομή Αλγορίθμου

Είσοδος:

L, συνάρτηση κόστους
θ, αρχικές τιμές βαρών των νευρώνων
ε, ρυθμός μάθησης

Βήματα μίας επανάληψης ή εποχής:

I. Υπολογισμός των παραγώγων της συνάρτησης κόστους ως προς τα βάρη των νευρώνων κάνοντας και χρήση του αλγορίθμου backpropagation, βλέπε ενότητα 2.2.6.

$$g \leftarrow \frac{1}{m'} \nabla_{\theta} \sum_{i=1}^{m'} L(y_{data}, y_{model}, \theta)$$

II. Υπολογισμός του κανόνα ενημέρωσης των βαρών κάνοντας χρήση του ρυθμού μάθησης ε και των παραγώγων g:

$$\Delta\theta \leftarrow -\varepsilon g$$

III. Ενημέρωση των βαρών θ:

$$\theta \leftarrow \theta + \Delta\theta$$

Όπως φαίνεται και από τον αλγόριθμο τα δεδομένα εκπαίδευσης του δικτύου παρουσιάζονται σε αυτό ανά εποχές. Ο αλγόριθμος τερματίζει, είτε όταν το σφάλμα που μας δίνει η συνάρτηση κόστους γίνει ικανοποιητικά μικρό, είτε έπειτα από ένα προκαθορισμένο πλήθος επαναλήψεων.

2) Στοχαστική Κατάβαση Δυναμικού με Ορμή (SGD with Momentum)

Πρόκειται για παραλλαγή του αλγορίθμου SGD με την προσθήκη ενός επιπλέον όρου, αυτόν της ορμής. Ο όρος της ορμής μας δίνει τη δυνατότητα να επιταχύνουμε τη μάθηση, καθώς πλέον, δε βασιζόμαστε μόνο στο ρυθμό μάθησης και τις παραγώγους του τρέχοντος βήματος για το πόσο γρήγορα, δηλαδή σε πόσες επαναλήψεις, θα προσεγγίσουμε τα

επιθυμητά βάρη, αλλά λαμβάνουμε υπόψη μας και τις παραγώγους g των προηγούμενων επαναλήψεων, όπως φαίνεται παρακάτω.

Δομή Αλγορίθμου

Είσοδος:

L, συνάρτηση κόστους
θ, αρχικές τιμές βαρών των νευρώνων
ε, ρυθμός μάθησης
α, παράμετρος του όρου της ορμής, α ανήκει [0, 1)

Βήματα μίας επανάληψης ή εποχής:

I. Υπολογισμός των παραγώγων της συνάρτησης κόστους ως προς τα βάρη των νευρώνων κάνοντας και χρήση του αλγορίθμου backpropagation, βλέπε ενότητα 2.2.6.

$$g \leftarrow \frac{1}{m'} \nabla_{\theta} \sum_{i=1}^{m'} L(y_{data}, y_{model}, \theta)$$

II. Υπολογισμός του κανόνα ενημέρωσης των βαρών:

$$u \leftarrow \alpha u - \varepsilon g$$

III. Ενημέρωση των βαρών θ:

$$\theta \leftarrow \theta + u$$

Σύμφωνα με το [13] η έρευνα σχετικά με τους αλγορίθμους εκπαίδευσης οδήγησε σύντομα στο συμπέρασμα, ότι ο ρυθμός μάθησης είναι μια παράμετρος (hyperparameter) αρκετά ευαίσθητη και δύσκολο να τεθεί, γεγονός που οδήγησε στη δημιουργία αλγορίθμων εκπαίδευσης με Προσαρμοζόμενο Ρυθμό Μάθησης (Adaptive Learning Rates).

3) Αλγόριθμος Adaptive Gradient (AdaGrad)

Προτάθηκε στο [14] και είναι επέκταση της μεθόδου SGD. Ο αλγόριθμος αυτός προσαρμόζει τον ρυθμό μάθησης, ατομικά για κάθε ξεχωριστή παράμετρο του δικτύου, κλιμακώνοντας κάθε μία από αυτές αντιστρόφως ανάλογα με την τετραγωνική ρίζα του αθροίσματος των τετραγώνων των παρελθοντικών της τιμών. Έτσι παράμετροι με μεγάλη μερική παράγωγο της συνάρτησης σφάλματος αποκτούν ρυθμό μάθησης με μεγάλη μείωση ενώ αντιθέτως, παράμετροι με μικρή μερική παράγωγο της συνάρτησης σφάλματος αποκτούν ρυθμό μάθησης με μικρότερη μείωση.

Δομή Αλγορίθμου

Είσοδος:

L, συνάρτηση κόστους
θ, αρχικές τιμές βαρών των νευρώνων
ε, ρυθμός μάθησης
δ, μικρή σταθερά, ίσως 10^{-7}, για αριθμητική σταθερότητα

Αρχικοποίηση:

r ← 0, μεταβλητή συσσώρευσης παραγώγων

Βήματα μίας επανάληψης ή εποχής:

I. Υπολογισμός των παραγώγων της συνάρτησης κόστους ως προς τα βάρη των νευρώνων κάνοντας και χρήση του αλγορίθμου backpropagation, βλέπε ενότητα 2.2.6.

$$g \leftarrow \frac{1}{m'} \nabla_{\theta} \sum_{i=1}^{m'} L(y_{data}, y_{model}, \theta)$$

II. Συσσώρευση τετραγωνισμένων παραγώγων:

$$r \leftarrow r + g * g$$

III. Υπολογισμός του κανόνα ενημέρωσης των βαρών:

$$\Delta\theta \leftarrow -\frac{\varepsilon}{\delta + \sqrt{r}} g$$

IV. Ενημέρωση των βαρών θ:

$$\theta \leftarrow \theta + \Delta\theta$$

4) Αλγόριθμος Root Mean Square Propagation (RMSProp)

Προτάθηκε από τον Hinton το 2012 και είναι τροποποίηση του αλγορίθμου AdaGrad. Ο αλγόριθμος αυτός αλλάζει τη συσσώρευση των παραγώγων των παρελθοντικών τιμών σε έναν κινητό μέσο όρο με βάρη, με τη χρήση του ρυθμού μείωσης ρ. Έτσι κατά την εκπαίδευση δε λαμβάνεται υπόψιν το μακρινό παρελθόν των παραγώγων και αποφεύγεται και το πρόβλημα της ταχείας εξασθένισης του ρυθμού μάθησης.

Δομή Αλγορίθμου

Είσοδος:

L, συνάρτηση κόστους
θ, αρχικές τιμές βαρών των νευρώνων
ε, ρυθμός μάθησης
ρ, ρυθμός μείωσης
δ, μικρή σταθερά, συνήθως 10^{-6}, για αριθμητική σταθερότητα διαιρέσεων με μικρούς αριθμούς

Αρχικοποίηση:

r ← 0, μεταβλητή συσσώρευσης παραγώγων

Βήματα μίας επανάληψης ή εποχής:

I. Υπολογισμός των παραγώγων της συνάρτησης κόστους ως προς τα βάρη των νευρώνων κάνοντας και χρήση του αλγορίθμου backpropagation, βλέπε ενότητα 2.2.6.

$$g \leftarrow \frac{1}{m'} \nabla_\theta \sum_{i=1}^{m'} L(y_{data}, y_{model}, \theta)$$

II. Συσσώρευση τετραγωνισμένων παραγώγων:

$$r \leftarrow \rho r + (1-\rho) g * g$$

III. Υπολογισμός του κανόνα ενημέρωσης των βαρών:

$$\Delta\theta \leftarrow -\frac{\varepsilon}{\sqrt{\delta + r}} g$$

IV. Ενημέρωση των βαρών θ:

$$\theta \leftarrow \theta + \Delta\theta$$

5) Αλγόριθμος Adaptive Moments (Adam)

Προτάθηκε στο [15]. Είναι ένας ακόμη αλγόριθμος προσαρμοζόμενου ρυθμού μάθησης, ο οποίος συνδυάζει τη χρήση ορμής, τη χρήση δυο μεταβλητών συσσώρευσης παραγώγων αλλά και διορθώσεις ‘’προκατάληψης’’ (bias corrections) αυτών.

Δομή Αλγορίθμου

Είσοδος:

L, συνάρτηση κόστους

θ, αρχικές τιμές βαρών των νευρώνων

ε, ρυθμός μάθησης, προτεινόμενος: 0,001

ρ_1, ρ_2, ρυθμοί μείωσης, προτεινόμενοι: 0,9 και 0,999 αντίστοιχα

δ, μικρή σταθερά για αριθμητική σταθερότητα, προτεινόμενη: 10^{-8}

Αρχικοποίηση:

s ← 0, μεταβλητή πρώτης στιγμής

r ← 0, μεταβλητή δεύτερης στιγμής

t ← 0, μεταβλητή χρονικού βήματος

Βήματα μίας επανάληψης ή εποχής:

I. Υπολογισμός των παραγώγων της συνάρτησης κόστους ως προς τα βάρη των νευρώνων κάνοντας και χρήση του αλγορίθμου backpropagation, βλέπε ενότητα 2.2.6.

$$g \leftarrow \frac{1}{m'} \nabla_\theta \sum_{i=1}^{m'} L(y_{data}, y_{model}, \theta)$$

II. Ενημέρωση των ‘’προκατειλημμένων’’ μεταβλητών πρώτης και δεύτερης στιγμής:

$$s \leftarrow \rho_1 s + (1-\rho_1) g$$
$$r \leftarrow \rho_2 r + (1-\rho_2) g * g$$

III. Διόρθωση ‘’προκατάληψης’’ (bias) των μεταβλητών πρώτης και δεύτερης στιγμής:

$$\hat{s} \leftarrow \frac{s}{1-\rho_1^t}$$
$$\hat{r} \leftarrow \frac{r}{1-\rho_2^t}$$

IV. Υπολογισμός του κανόνα ενημέρωσης των βαρών:

$$\Delta\theta \leftarrow -\varepsilon \frac{\hat{s}}{\sqrt{\hat{r}}+\delta} g$$

V. Ενημέρωση των βαρών θ:

$$\theta \leftarrow \theta + \Delta\theta$$

2.2.8 Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks, CNN)

Στην ενότητα 2.2.4 παρουσιάσαμε το βασικότερο είδος Βαθιών Νευρωνικών Δικτύων με πολλά επίπεδα, τα MLP. Σε αυτά υπάρχει πλήρη συνδεσιμότητα μεταξύ διαδοχικών επιπέδων (fully connected), δηλαδή η έξοδος κάθε νευρώνα ενός επιπέδου τροφοδοτείται σε κάθε νευρώνα του επόμενου επιπέδου. Ένα Βαθύ Νευρωνικό Δίκτυο δε χρειάζεται να είναι πλήρως συνδεδεμένο για οποιαδήποτε εφαρμογή και οποιοδήποτε είδος δεδομένων εισόδου. Μάλιστα παραλείποντας συνδέσεις (sparse connections) μεταξύ των επιπέδων πολλές φορές επιταχύνουμε τη διαδικασία εκπαίδευσης και βελτιώνουμε τα αποτελέσματα. Έτσι έχουμε τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks, CNN).

Τα CNN είναι ένα είδος Νευρωνικών Δικτύων τα οποία, σε τουλάχιστον 1 επίπεδο νευρώνων, χρησιμοποιούν την πράξη της συνέλιξης για τον υπολογισμό της εξόδου του επιπέδου και όχι τον πολλαπλασιασμό πινάκων όπως τα MLP. Δηλαδή το διάνυσμα εισόδου συνελίσεται με τα βάρη των συνδέσεων του νευρώνα και το αποτέλεσμα αυτό δίνεται στην συνάρτηση ενεργοποίησης του νευρώνα. Αξίζει να σημειωθεί εδώ, ότι πολλές βιβλιοθήκες λογισμικού μηχανικής μάθησης, υλοποιούν μια παρόμοια πράξη, την cross-correlation, αλλά την ονομάζουν συνέλιξη (convolution), χωρίς όμως να αλλάζει κάτι στη γενική ιδέα των Συνελικτικών Νευρωνικών Δικτύων.

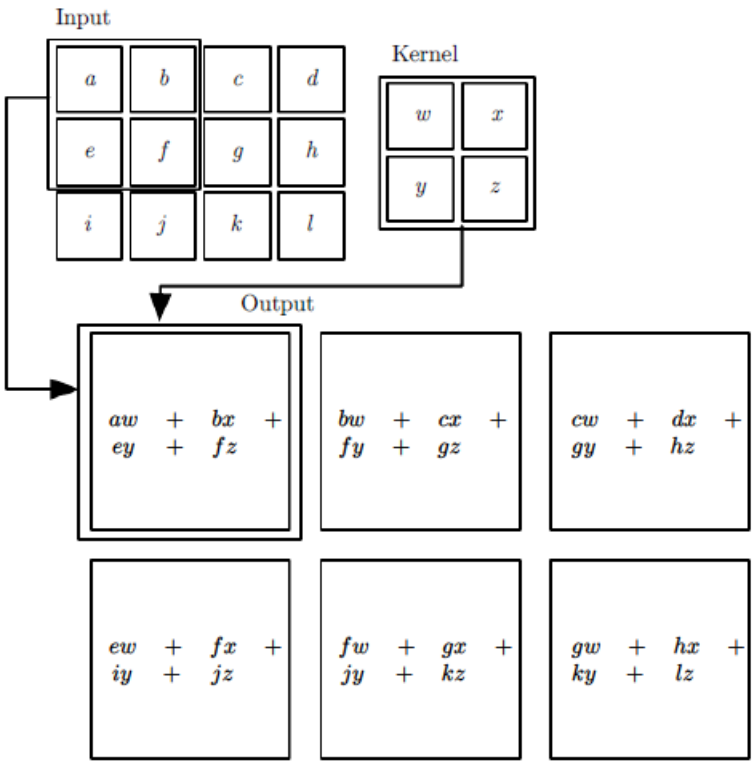
Ακολουθεί η παρουσίαση της πράξης της συνέλιξης:

Συνέλιξη συνεχών συναρτήσεων x, w: $s(t) = (x * w)(t) = \int x(a)w(t-a)da$

Συνέλιξη διακριτών συναρτήσεων x, w: $s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t-a)$

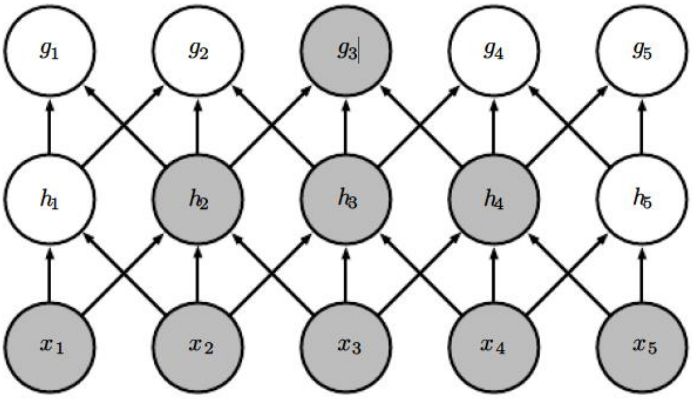
Το x είναι το διάνυσμα εισόδου ενός επιπέδου. Το δεύτερο όρισμα της συνέλιξης, αντίστοιχο με τον πίνακα βαρών W των MLP, ονομάζεται πυρήνας (kernel). Η έξοδος του επιπέδου s ονομάζεται χάρτης χαρακτηριστικών (feature map).

Ο πυρήνας συνήθως έχει μέγεθος μικρότερο από το διάνυσμα ή πίνακα της εισόδου, βλέπε σχήμα 5. Όμως συνέλιξη με έναν πυρήνα μικρότερο από την είσοδο σημαίνει ότι το επίπεδο δεν είναι πλήρως συνδεδεμένο με το προηγούμενο αλλά έχουμε αραιές συνδέσεις (sparse connections). Παρόλα αυτά τα βαθύτερα επίπεδα σε ένα CNN συνδέονται έμμεσα με σχεδόν όλα τα στοιχεία της αρχικής εισόδου του νευρωνικού δικτύου, βλέπε σχήμα 6. Οπότε γλιτώνουμε πολυπλοκότητα στους υπολογισμούς αλλά δε χάνουμε αποτελεσματικότητα.



Σχήμα 8: Παράδειγμα Δισδιάστατης Συνέλιξης ενός πίνακα εισόδου με έναν πυρήνα (Kernel)
Πηγή [13]

Σε αντίθεση με ένα fully connected MLP, σε ένα CNN, ανάλογα με το μέγεθος του πυρήνα, έχουμε από κοινού χρήση παραμέτρων (parameter sharing), δηλαδή χρησιμοποιούμε την ίδια παράμετρο για περισσότερες από μία μεταβλητές του μοντέλου. Κάθε βάρος του πυρήνα χρησιμοποιείται σε σχεδόν κάθε στοιχείο της εισόδου καθώς μετακινούμε τον πυρήνα κατά τον υπολογισμό της συνέλιξης, βλέπε σχήμα 5. Ανάλογα με τη μορφή του parameter sharing ένα επίπεδο μπορεί να αποκτήσει ισοδυναμία στη μετάφραση (equivariance to translation), όπως για παράδειγμα σε μια ολίσθηση (shift) εισόδου.



Σχήμα 9: Παράδειγμα νευρώνων σε βαθύτερα επίπεδα ενός CNN, τα οποία συνδέονται έμμεσα με ολόκληρη την είσοδο του δικτύου.
Πηγή [13]

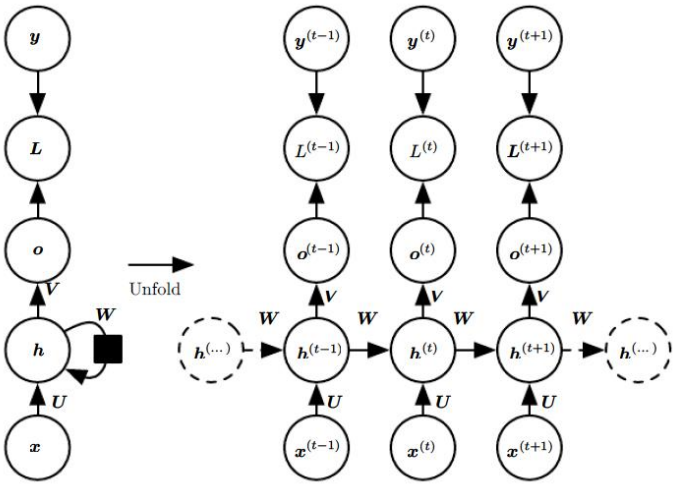
Όλα αυτά τα χαρακτηριστικά καθιστούν τα CNN καταλληλότερα για δεδομένα εισόδου με τοπολογίες πλέγματος, όπου τα δεδομένα έχουν κάποια χωρική συσχέτιση, όπως πχ συμβαίνει με εικόνες. Επίσης τα CNN χρησιμοποιούνται και για εφαρμογές αναγνώρισης φωνής (speech recognition). Οπότε τα CNN δεν κρίνονται ως τα πλέον καταλληλότερα για το είδος ταξινομητή που θέλουμε να δημιουργήσουμε, το οποίο θα δέχεται ως δεδομένα εισόδου καταγεγραμμένη δικτυακή κίνηση.

2.2.9 Νευρωνικά Δίκτυα με Ανάδραση (Recurrent Neural Networks, RNN)

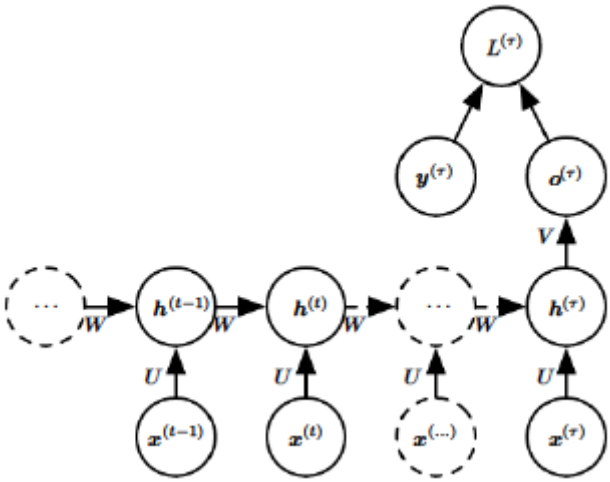
Για τη μελέτη ακολουθιακών δεδομένων εισόδου, όπως για παράδειγμα δεδομένα που αλλάζουν με την πάροδο του χρόνου, πολλές φορές χρησιμοποιούμε Νευρωνικά Δίκτυα με Ανάδραση (Recurrent Neural Networks, RNN). Τα RNN μπορούν να χειριστούν μεγαλύτερου και τις περισσότερες φορές μεταβλητού μήκους ακολουθίες δεδομένων εισόδου, σε αντίθεση με τα Δίκτυα Εμπρόσθιας τροφοδότησης (feedforward networks). Το γεγονός αυτό καθιστά τα RNN κατάλληλα για δεδομένα που εμφανίζουν κάποια χρονική εξάρτηση μεταξύ τους.

Τα RNN συνήθως δουλεύουν σε μικρά δείγματα (minibatches) από ακολουθίες, με μεταβλητό μήκος ακολουθίας τ για κάθε μέλος του minibatch. Σκοπός των RNN είναι να ‘’μάθουν’’ μια αναδρομική συνάρτηση, η οποία εκφράζει τη μετάβαση από μια κατάσταση στην επόμενη, και όχι μια συνάρτηση που δέχεται ολόκληρη την ακολουθία και παράγει το τελικό αποτέλεσμα. Έτσι το νευρωνικό δίκτυο, μετά την εκπαίδευση, μπορεί να ‘’εκτιμήσει’’ ακολουθίες μεταβλητού μήκους και ας μην έχει εκπαιδευτεί με τέτοια μήκη ακολουθιών. Βασικό χαρακτηριστικό όλων των RNN είναι η ύπαρξη της ανάδρασης. Διαφορετικά είδη ανάδρασης και δεδομένων εκπαίδευσης οδηγούν σε διαφορετικά σχήματα RNN. Ακολούθως παρουσιάζονται τα πιο συνηθισμένα είδη σχημάτων.

- **Ανάδραση από τους νευρώνες ενός κρυφού επιπέδου στον εαυτό τους ή σε νευρώνες προηγούμενου κρυφού επιπέδου**

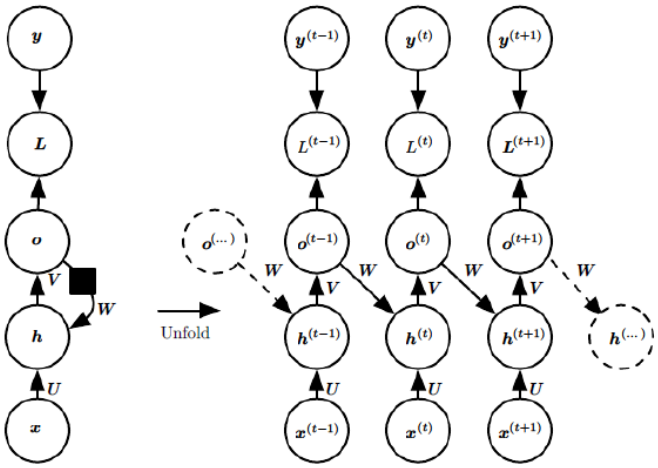


Σχήμα 10: Παράδειγμα RNN με συνδέσεις επανατροφοδότησης των κρυφών μονάδων (hidden units) του επιπέδου h στον εαυτό τους. Τα δεδομένα εκπαίδευσης περιλαμβάνουν ετικέτες y(t) και επομένως εξόδους o(t) σε κάθε βήμα ή χρονική στιγμή t. Στο δεξί μέρος του σχήματος φαίνεται πως ανατροφοδοτείται η κατάσταση των νευρώνων του επιπέδου h με την πάροδο του χρόνου.
Πηγή [13]



Σχήμα 11: Παράδειγμα RNN με συνδέσεις επανατροφοδότησης των κρυφών μονάδων (hidden units) του επιπέδου h στον εαυτό τους. Τα δεδομένα εκπαίδευσης περιλαμβάνουν ετικέτες y(t) και επομένως εξόδους o(t) μόνο στο βήμα ή χρονική στιγμή τ. Δηλαδή το δίκτυο έχει μόνο μια τελική έξοδο για μια ακολουθία δεδομένων που του δίνεται. Τέτοια δίκτυα χρησιμοποιούνται για να δώσουν για παράδειγμα την περίληψη ή τον χαρακτηρισμό μιας ακολουθίας δεδομένων.
Πηγή [13]

- **Ανάδραση από το επίπεδο εξόδου στους νευρώνες ενός κρυφού επιπέδου**



Σχήμα 12: Παράδειγμα RNN με συνδέσεις επανατροφοδότησης από το επίπεδο εξόδου (Output Layer) στις κρυφές μονάδες (hidden units) του επιπέδου h. Τα δεδομένα εκπαίδευσης περιλαμβάνουν ετικέτες y(t) και επομένως εξόδους o(t) σε κάθε βήμα ή χρονική στιγμή t. Στο δεξί μέρος του σχήματος φαίνεται πως ανατροφοδοτείται η κατάσταση των νευρώνων του επιπέδου h με την πάροδο του χρόνου.
Πηγή [13]

Δίκτυα με ανάδραση αυτής της μορφής δεν είναι τόσο ισχυρά όσο τα προηγούμενα αλλά επιτρέπουν την παράλληλη εκπαίδευση μεταξύ των βημάτων, καθώς μπορούμε να χρησιμοποιήσουμε, χωρίς μεγάλο σφάλμα, την τιμή της ετικέτας y(t-1) ως ανάδραση για την εκπαίδευση του βήματος t και δεν απαιτείται να περιμένουμε την τιμή της εξόδου o(t-1).

Αυτή η τεχνική, η οποία καθιστά επιτρεπτή την παράλληλη εκπαίδευση, ονομάζεται Teacher Forcing.

- **Συνδυασμός των δύο προηγούμενων ειδών ανάδρασης**

Όταν έχουμε ανάδραση από τους νευρώνες ενός κρυφού επιπέδου στον εαυτό τους ή σε νευρώνες προηγούμενου κρυφού επιπέδου αλλά και ανάδραση από το επίπεδο εξόδου στους νευρώνες ενός κρυφού επιπέδου.

2.2.10 Βαθιά Νευρωνικά Δίκτυα με Ανάδραση (Deep RNN)

Όπως στα MLP και τα CNN, έτσι και στα RNN μπορεί να έχουμε πολλά κρυφά επίπεδα, σε τουλάχιστον ένα από τα οποία εφαρμόζω μη-γραμμικό μετασχηματισμό των χαρακτηριστικών (εισόδου). Τότε κάνουμε λόγο για Βαθιά Νευρωνικά Δίκτυα με Ανάδραση (Deep RNN). Όμως η έννοια της Βαθιάς Μάθησης στα RNN δεν περιορίζεται μόνο στην προσθήκη περισσότερων κρυφών επιπέδων. Στα RNN με ένα κρυφό επίπεδο έχουμε ροή της πληροφορίας:

1) Από την είσοδο του δικτύου, στο κρυφό επίπεδο, λόγω των εμπρόσθιων συνδέσεων.
2) Από μια κατάσταση, του κρυφού επιπέδου, σε μια χρονική στιγμή (hidden state), στην επόμενη κατάσταση του κρυφού επιπέδου σε αμέσως επόμενη χρονική στιγμή, λόγω συνδέσεων ανατροφοδότησης.
3) Από μια κατάσταση, του κρυφού επιπέδου, σε μια χρονική στιγμή (hidden state), στην έξοδο του δικτύου.

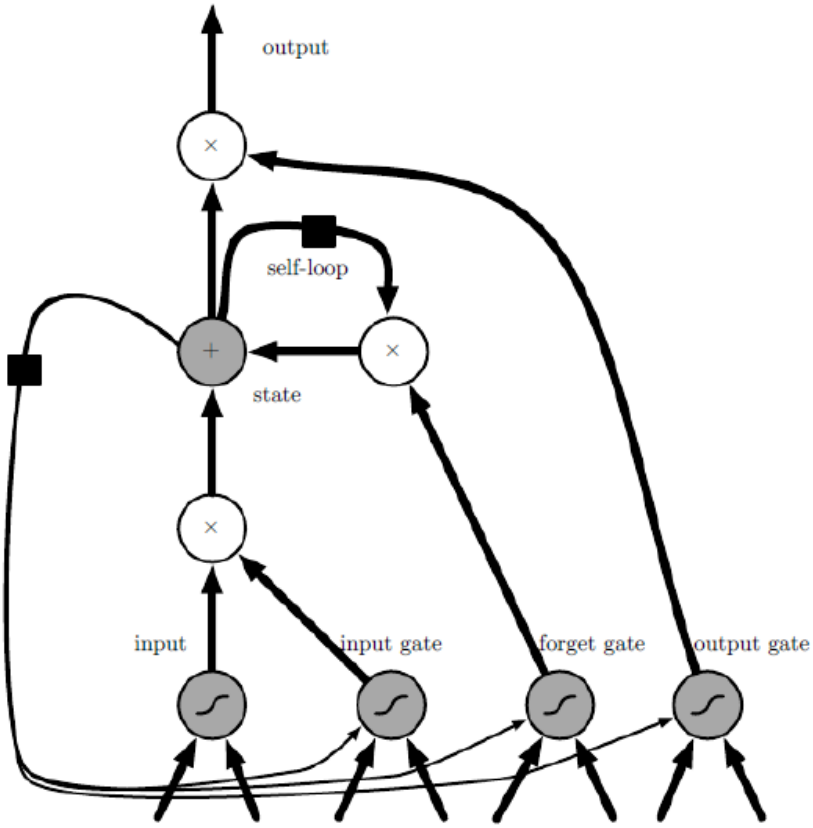
Η πρόσθεση βάθους σε κάθε μία από τις παραπάνω ροές πληροφορίας, είτε προσθέτοντας περισσότερα κρυφά επίπεδα, είτε προσθέτοντας ανάδραση από αρκετά παλιότερες καταστάσεις των νευρώνων στην κατάσταση του τωρινού βήματος-χρονικής στιγμής, οδηγεί σε Βαθύ Νευρωνικό Δίκτυο με Ανάδραση (Deep RNN), με δυνατότητες Βαθιάς Μάθησης, όπως αυτή έχει περιγραφεί στην ενότητα 2.2.1.

Οι μακροπρόθεσμες εξαρτήσεις στα βαθιά RNN, οι οποίες εκφράζονται με τους βρόγχους ανάδρασης, είναι αρκετά απαιτητικές σε υπολογισμούς, καθώς περιλαμβάνουν τη διάδοση παραγώγων πολλά επίπεδα προς τα πίσω. Επομένως ένας πίνακας βαρών, ο οποίος περιλαμβάνει και βάρη βρόγχων ανάδρασης, πολλαπλασιάζεται πολλές φορές με τον εαυτό του κατά τον αλγόριθμο backpropagation. Σαν αποτέλεσμα εκθετικά μικρότερα, ή σπανιότερα εκθετικά μεγαλύτερα, βάρη δίνονται σε νευρώνες, οι οποίοι αντιστοιχούν σε μακροπρόθεσμες αλληλεπιδράσεις. Οπότε έπειτα από τον πολλαπλασιασμό πολλών Ιακωβιανών πινάκων οι παράγωγοι των βαρών τείνουν να εξαφανιστούν, ή σπανιότερα να σκάσουν, κατά τον αλγόριθμο backpropagation. Το πρόβλημα αυτό της εξασθένισης των Βαρών (Weight Decay) και επομένως η αδυναμία έκφρασης μακροπρόθεσμων εξαρτήσεων των δεδομένων εισόδου (Long-Term Dependencies), βλέπε και [13], δυσκολεύει πολλές φορές την εκπαίδευση των Deep RNN που διαθέτουν αναδράσεις.

2.2.11 Δίκτυα Long Short-Term Memory (LSTM)

Το πρόβλημα της εξασθένισης των βαρών των RNN, το οποίο περιγράψαμε στην προηγούμενη ενότητα, λύνεται όσο το δυνατόν καλύτερα, σύμφωνα με το [13], με τη χρήση Αναδρασιακών Νευρωνικών Δικτύων με πύλες (gated RNN). Σε αυτή την κατηγορία ανήκουν και τα Long Short-Term Memory RNN (LSTM RNN). Βασική διαφοροποίησή τους είναι ότι χρησιμοποιούν εσωτερικά, σε κάθε νευρώνα ή κύτταρο (cell) όπως ονομάζεται, λογικές

πύλες, με βάρη, οι οποίες ελέγχουν την ανάδραση. Ακολουθεί η παρουσίαση του κυττάρου των LSTM και οι βασικές εξισώσεις που διέπουν τη λειτουργία του.



Σχήμα 13: Εσωτερική δομή του κυττάρου (cell) ενός LSTM.
Πηγή [13]

Η είσοδος ενός κυττάρου περνάει από μια μονάδα τεχνητού νευρώνα όπως στα κλασσικά νευρωνικά δίκτυα, input στο σχήμα 10, και η τιμή του αποτελέσματος αφού περάσει από μια μη-γραμμική συνάρτηση ενεργοποίησης, μπορεί να συσσωρευτεί στο state, όταν το επιτρέπει μια σιγμοειδής πύλη εισόδου, input gate στο σχήμα 10. Η εσωτερική κατάσταση, state, έχει μια γραμμική ανάδραση στον εαυτό της, η οποία ελέγχεται από μια πύλη forget. Τέλος η έξοδος του κυττάρου έχει τη δυνατότητα να αποκόπτεται, μερικώς ή και πλήρως, γεγονός που ελέγχεται από μια πύλη εξόδου, output gate. Όλες οι πύλες, input gate, forget gate και output gate, παίρνουν ως είσοδο την είσοδο του κυττάρου, αλλά και ανάδραση από την προηγούμενη έξοδο του κυττάρου και παράγουν σιγμοειδή έξοδο με εύρος τιμών [0, 1].

Η ιδέα του κυττάρου LSTM είναι ότι εισάγει αναδράσεις στην δική του κατάσταση (self-loops), οι οποίες προσφέρουν μονοπάτια στα οποία η παράγωγος μπορεί να ρέει για μεγάλη διάρκεια, δηλαδή πολλά βήματα, αποφεύγοντας έτσι το πρόβλημα της εξασθένισης των βαρών. Παράλληλα όμως, υπάρχει και η δυνατότητα να ξεχαστεί η παλιά κατάσταση του κυττάρου όταν αυτό κρίνεται σκόπιμο από το δίκτυο.

Εξισώσεις του κυττάρου LSTM ενός Δικτύου:

Έστω $x^{(t)}$ το τωρινό διάνυσμα εισόδου, $h^{(t)}$ το διάνυσμα του τωρινού κρυφού επιπέδου, το οποίο περιέχει τις εξόδους όλων των κυττάρων του LSTM δικτύου, b οι τιμές ‘’προκατάληψης’’ (biases), U τα βάρη που πολλαπλασιάζονται με την είσοδο των πυλών και

W τα βάρη ανάδρασης των πυλών. Χρησιμοποιούμε σύμβολα g, f και o στα διανύσματα b, U και W όταν αναφερόμαστε αντίστοιχα στις πύλες input gate, forget gate και output gate, ενώ δε χρησιμοποιούμε κανένα επιπλέον σύμβολο, όταν αναφερόμαστε στα b, U, W της μονάδας input. Επομένως έχουμε:

Για την input gate του i-οστού κυττάρου:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right)$$

Για τη forget gate του i-οστού κυττάρου:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right)$$

Για την output gate του i-οστού κυττάρου:

$$o_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right)$$

Οπότε η εσωτερική κατάσταση s του i-οστού κυττάρου, σε κάποια χρονική στιγμή t, δίνεται από παρακάτω αναδρομικό τύπο, ο οποίος όπως παρατηρούμε, κάνει χρήση της αμέσως προηγούμενης εσωτερικής κατάστασης του κυττάρου $s^{(t-1)}$ και των εξόδων των πυλών input gate, forget gate καθώς και της εξωτερικής εισόδου του κυττάρου LSTM:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right)$$

Τέλος για την έξοδο του i-οστού κυττάρου έχουμε:

$$h_i^{(t)} = \tanh\left(s_i^{(t)}\right) q_i^{(t)}$$

Τα δίκτυα LSTM έχει δειχθεί ότι μαθαίνουν μακροπρόθεσμες εξαρτήσεις πολύ πιο εύκολα από ότι τα κλασικά RNN. Αυτό το συμπέρασμα προέκυψε αφού εξετάστηκαν οι επιδόσεις τους, αρχικά σε τεχνητά δεδομένα, ειδικά σχεδιασμένα να έχουν μακροπρόθεσμες εξαρτήσεις, βλέπε [16] και [17], και έπειτα σε απαιτητικές εφαρμογές επεξεργασίας ακολουθιών, βλέπε δημοσιεύσεις [18] και [19] του Graves.

2.2.12 Επεξεργασία των Δεδομένων Εκπαίδευσης

Πριν την έναρξη της εκπαίδευσης ενός νευρωνικού δικτύου είναι απαραίτητο τις περισσότερες φορές να φέρουμε τα δεδομένα εκπαίδευσης που διαθέτουμε σε κατάλληλη μορφή. Ακολουθούν κάποιες τεχνικές επεξεργασίας των δεδομένων που διαθέτουμε.

- **Κανονικοποίηση των Δεδομένων Εκπαίδευσης:**

Ένα απαραίτητο στάδιο είναι η κανονικοποίηση των δεδομένων. Στο στάδιο αυτό φροντίζουμε ώστε τα διάφορα πεδία ή χαρακτηριστικά των δεδομένων μας να έχουν το ίδιο εύρος τιμών. Έστω, για παράδειγμα, δεδομένα (samples) με ένα πεδίο ή χαρακτηριστικό την ηλικία ανθρώπων, οπότε θα έχει εύρος περίπου 100 χρόνων, ενώ ένα πεδίο, το οποίο αφορά

το ύψος των ίδιων ανθρώπων, θα έχει εύρος μέχρι κάποια μέτρα. Προκειμένου η μεταβλητή της ηλικίας, καθότι έχει μεγαλύτερες τιμές, να μην επηρεάσει περισσότερο τις τιμές των βαρών κατά την εκπαίδευση του δικτύου απαιτείται κανονικοποίηση των δεδομένων.

Ένας εύκολος τρόπος να κανονικοποιήσουμε οποιαδήποτε δεδομένα είναι η γραμμική κλιμάκωση (Linear Scaling) σε μοναδιαίο εύρος. Για κάθε μεταβλητή χαρακτηριστικού x_i, των δεδομένων X, έχουμε:

$$z_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}$$

Με min και max αναφερόμαστε στην ελάχιστη και μέγιστη τιμή για το εκάστοτε χαρακτηριστικό. Έτσι κάθε χαρακτηριστικό z_i των κανονικοποιημένων δεδομένων Z ανήκει πλέον στο εύρος τιμών [0, 1].

- **Ανάλυση σε Βασικές Συνιστώσες (Principal Component Analysis, PCA)**

Μια άλλη ευρέως χρησιμοποιούμενη επεξεργασία, στην οποία υποβάλουμε τα δεδομένα εκπαίδευσης, είναι η Ανάλυση σε Βασικές Συνιστώσες (Principal Component Analysis, PCA). Η μέθοδος PCA είναι μια στατιστική διαδικασία, η οποία χρησιμοποιεί έναν ορθογώνιο μετασχηματισμό, για να μετατρέψει ένα σύνολο παρατηρήσεων, με πιθανώς συσχετιζόμενες μεταβλητές, σε ένα σύνολο από τιμές γραμμικά ασυσχέτιστων μεταβλητών τις οποίες ονομάζουμε Βασικές Συνιστώσες (Principal Components). Το πλήθος των βασικών συνιστωσών που προκύπτει είναι μικρότερο ή ίσο, από το μικρότερο εκ των δύο, το πλήθος των αρχικών μεταβλητών των παρατηρήσεων και το πλήθος των παρατηρήσεων.

Επομένως η μέθοδος PCA δίνει τη δυνατότητα να παραλείψουμε πεδία και χαρακτηριστικά του συνόλου των παρατηρήσεων, τα οποία δεν προσφέρουν κάποια σημαντική επιπλέον πληροφορία για τα δεδομένα που διαθέτουμε. Αυτό το γεγονός μας επιτρέπει την κατασκευή μικρότερων νευρωνικών δικτύων, με διάνυσμα εισόδου με λιγότερα χαρακτηριστικά, και επομένως λιγότερα βάρη στους νευρώνες του δικτύου, γεγονός που επιταχύνει τη διαδικασία της εκπαίδευσης.

Τέλος, η μέθοδος PCA, εκτός από την ταχύτητα εκπαίδευσης, βελτιώνει κάποιες φορές και την ακρίβεια του νευρωνικού δικτύου, το οποίο εκπαιδεύουμε με τα νέα δεδομένα μικρότερης διάστασης, καθώς αποφεύγεται το υπερβολικό ‘’ταίριασμα’’ (overfitting) πάνω σε κάποια χαρακτηριστικά των οποίων οι πληροφορίες περιέχονται και σε άλλα, συσχετισμένα με αυτά χαρακτηριστικά.

2.2.13 Δυνατότητα γενίκευσης των Νευρωνικών Δικτύων – Η τεχνική Dropout

Συχνά μπορεί να παρατηρηθεί overfitting των νευρωνικών δικτύων πάνω σε συγκεκριμένα δεδομένα εκπαίδευσης. Αυτό σημαίνει ότι το δίκτυο παρόλο που πετυχαίνει καλή ακρίβεια πάνω στα δεδομένα εκπαίδευσης, χάνει τη δυνατότητα γενίκευσης σε άλλα δεδομένα με διαφορετικά χαρακτηριστικά που δεν έχει ξαναδεί.

Μια από τις λύσεις στο πρόβλημα αυτό είναι η τεχνική Dropout. Η τεχνική αυτή εφαρμόζεται κατά την εκπαίδευση των νευρωνικών δικτύων και περιλαμβάνει την πιθανή απενεργοποίηση κάθε νευρώνα ενός κρυφού επιπέδου με μια πιθανότητα p που ανήκει στο [0,1]. Για νευρώνες οι οποίοι απενεργοποιούνται η έξοδός τους τίθεται μηδενική. Έτσι το νευρωνικό δίκτυο τελικά μαθαίνει να μη βασίζεται στα βάρη συγκεκριμένων νευρώνων. Νευρώνες γειτονικοί αυτών που έχουν απενεργοποιηθεί μαθαίνουν πιο γενικά χαρακτηριστικά για να εξισορροπήσουν την απουσία άλλων νευρώνων. Έτσι **το νευρωνικό δίκτυο εσωτερικά μαθαίνει πιο γενικές αναπαραστάσεις των δεδομένων εκπαίδευσης** και αποκτά μεγαλύτερη ικανότητα γενίκευσης και επομένως καλύτερη ακρίβεια (validation accuracy) σε νέα δεδομένα.

3 Αρχιτεκτονική του Συστήματος

Το σύστημα που κατασκευάσαμε για την υλοποίηση των πειραμάτων και την εξαγωγή των συμπερασμάτων αποτελείται από τα ακόλουθα στοιχεία.

- **Σημείο Παρατήρησης – Εξαγωγέας και Συλλέκτης**

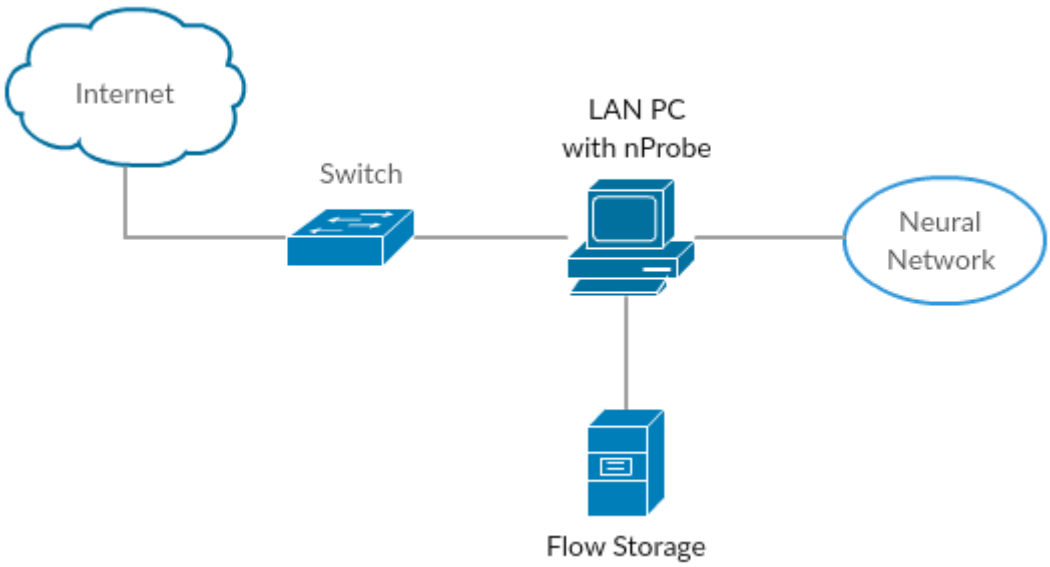
Ένα σημείο παρατήρησης όπου συλλέγεται η δικτυακή κίνηση η οποία διέρχεται. Αυτή η συλλογή γίνεται με τη χρήση του εργαλείου nProbe, βλέπε [20], το οποίο συλλέγει ροές IP, κάνοντας χρήση του πρωτοκόλλου Netflow, βλέπε ενότητες 2.1.3 και 2.1.4. Το σημείο αυτό της παρατήρησης επιλέχτηκε να είναι ένα μηχάνημα του τοπικού δικτύου μας, στο οποίο καταγράφονταν η εισερχόμενη και η εξερχόμενη κίνηση. Το εργαλείο nProbe εκτελεί τη λειτουργία τόσο του Εξαγωγέα όσο και του Συλλέκτη που απαιτείται από ένα σύστημα παρακολούθησης με το πρωτόκολλο Netflow.

- **Επεξεργασία και αποθήκευση ροών Netflow**

Στη συνέχεια οι καταγεγραμμένες ροές που προκύπτουν από το nProbe, αποθηκεύονται σε έναν υπολογιστή του τοπικού δικτύου και επεξεργάζονται από κώδικα που δημιουργήσαμε, ώστε να είναι σε επιθυμητή μορφή για το Νευρωνικό Δίκτυο.

- **Ταξινόμηση Ροών με χρήση Νευρωνικού Δικτύου**

Οι επεξεργασμένες πλέον ροές Netflow τροφοδοτούνται στο Νευρωνικό Δίκτυο που κατασκευάσαμε και βρίσκεται στον ίδιο υπολογιστή. Για την εκτέλεση των πειραμάτων δοκιμάσαμε πολλά διαφορετικά είδη και τοπολογίες Νευρωνικών Δικτύων. Επίσης πειραματιστήκαμε με τον τρόπο με τον οποίο πρέπει να δώσουμε τα δεδομένα στο νευρωνικό δίκτυο και ποια είναι τα καταλληλότερα πεδία που πρέπει να τροφοδοτήσουμε σε αυτό. Στο τελευταίο στάδιο το Νευρωνικό Δίκτυο εκτελεί την κατηγοριοποίηση των ροών σε πέντε διαφορετικές κατηγορίες που θα παρουσιάσουμε αναλυτικά στην επόμενη ενότητα.



Σχήμα 14: Αρχιτεκτονική του συστήματος συλλογής, αποθήκευσης και επεξεργασίας δικτυακής κίνησης που υλοποιήσαμε για την εκτέλεση των πειραμάτων της ενότητας 5.

4 Θέματα Υλοποίησης

4.1 Δεδομένα εκπαίδευσης που χρησιμοποιήθηκαν για κάθε κατηγορία ταξινόμησης

Όπως αναφέραμε στην εισαγωγή της διπλωματικής εργασίας, η ταξινόμηση που επιθυμούμε να εκτελεί το Νευρωνικό Δίκτυο που κατασκευάσαμε είναι μεταξύ των πέντε διαφορετικών κατηγοριών που ακολουθούν:

- Πλημμύρα ICMP (ICMP Flood ή Ping Flood)
- Πλημμύρα TCP πακέτων με σημαία SYN (SYN Flood)
- Πλημμύρα UDP (UDP Flood)
- Σάρωση Θυρών (Port Scanning)
- Καλόβουλη κίνηση (Legitimate traffic)

Για την εκπαίδευση των διαφορετικών Νευρωνικών Δικτύων, βλέπε ενότητες 2.2.5 έως 2.2.7, τα οποία κατασκευάσαμε κατά την εκτέλεση των πειραμάτων, χρησιμοποιήσαμε δικτυακή κίνηση από κάθε μία από τις παραπάνω κατηγορίες. Παρουσιάζουμε λοιπόν τον τρόπο με τον οποίο αντλήσαμε ή κατασκευάσαμε δικτυακή κίνηση για κάθε μία από τις κατηγορίες αυτές.

Πλημμύρα ICMP και Πλημμύρα SYN:

Η δικτυακή κίνηση για αυτές τις δύο κατηγορίες επιθέσεων προήλθε από το αρχείο επίθεσης του 2007 της CAIDA. Η επίθεση διήρκησε περίπου μία ώρα και μέσα σε λίγα λεπτά η κίνηση του δικτύου έφτασε από τα 200 Kbps στα 80 Mbps.

Το αρχείο επίθεσης είναι οργανωμένο σε κομμάτια, τα οποία αντιστοιχούν σε πεντάλεπτα χρονικά διαστήματα καταγραφής κατά τη διάρκεια της επίθεσης. Για τη δημιουργία των ροών Netflow τροφοδοτήσαμε κάποια από αυτά τα κομμάτια της καταγραφής στο εργαλείο nProbe. Για να έχουμε στην έξοδο του nProbe ροές μόνο της μίας κατηγορίας, ‘’Πλημμύρα ICMP’’, ή αντίστοιχα της άλλης κατηγορίας, ‘’Πλημμύρα SYN’’, χρησιμοποιήθηκε κατά την τροφοδοσία των πεντάλεπτων κομματιών φίλτρο καταγραφής στο nProbe, με την επιλογή –f ‘’ip proto 1’’ για ICMP πακέτα ή –f ‘’ip proto 6’’ για TCP πακέτα. Πιο συγκεκριμένα, κατά τη συλλογή TCP πακέτων για την επίθεση ‘’Πλημμύρα SYN’’, το πλήρες φίλτρο που χρησιμοποιήθηκε περιόριζε και τη διεύθυνση αποστολής των TCP πακέτων, σε μερικές μόνο από τις διευθύνσεις που συμμετείχαν στην επίθεση. Αυτό έγινε καθώς οι παραγόμενες σε πλήθος ροές Netflow, που αντιστοιχούν σε πεντάλεπτη καταγραφή της επίθεσης, ήταν υπερβολικά πολλές, περισσότερες από όσες χρειαζόταν η επαρκής εκπαίδευση του νευρωνικού δικτύου. Επιπλέον δε θέλαμε να ξεπεράσουμε τις δυνατότητες καταγραφής της ‘’demo’’ έκδοσης του εργαλείου nProbe που χρησιμοποιήθηκε.

Τέλος κάναμε, χωρίς βλάβη της γενικότητας, τη σύμβαση ότι το δίκτυο που επιθυμούμε να προστατέψουμε από επιθέσεις και να τοποθετηθεί το τελικό νευρωνικό δίκτυο που προτείνουμε βρίσκεται εντός του AS του ΕΜΠ. Επομένως θέλαμε η επίθεση να έχει ASN προορισμού αυτό του ΕΜΠ, το 3323. Καθώς το αρχείο περιέχει μόνο εισερχόμενη δικτυακή κίνηση προς τη διεύθυνση IP του θύματος, αλλάξαμε το πεδίο ASN προορισμού των ροών Netflow, που πήραμε στην έξοδο του nProbe, σε αυτό του ΕΜΠ, το 3323. Το νευρωνικό δίκτυο δε δέχεται κατά την εκπαίδευση τα πεδία των διευθύνσεων IP αλλά μόνο τα πεδία των ASN, οπότε δεν προκαλείται πρόβλημα με την αλλαγή του ASN του δικτύου προορισμού.

Πλημμύρα UDP:

Η δικτυακή κίνηση, η οποία χρησιμοποιήθηκε για την επίθεση αυτής της κατηγορίας, ήταν τεχνητή κίνηση. Η Πλημμύρα UDP που υλοποιήθηκε, διάρκειας 5 λεπτών, κατασκευάστηκε με τη βοήθεια του εργαλείου Scapy, βλέπε [21]. Πιο συγκεκριμένα, στείλαμε δεδομενογράμματα (datagrams) UDP μεταβλητού μεγέθους, από ψευδείς IP διευθύνσεις αποστολής (spoofed IP addresses), προς υπολογιστή του τοπικού μας δικτύου. Επιλέξαμε αυθαίρετα κάποιες θύρες αποστολής και ως θύρες προορισμού των UDP πακέτων θέσαμε τις 23, 25, 53, 161 και 50005.

Για να πετύχουμε όγκο επίθεσης αντίστοιχο με αυτό των επιθέσεων που προέκυψαν από το αρχείο της CAIDA ακολουθήσαμε τα εξής βήματα:

- Αρχικά καταγράψαμε κίνηση UDP παραγόμενη από το εργαλείο Scapy που αποστέλνονταν προς έναν τοπικό υπολογιστή του δικτύου για διάρκεια αρκετά μεγαλύτερη των 5 λεπτών. Αυτή η κίνηση όμως δεν είχε αρκετά μεγάλο ρυθμό καθώς είχαμε περιορισμούς, οι οποίοι πιθανολογούμε ότι προέρχονταν από τον πυρήνα (kernel) του λειτουργικού μας συστήματος (Linux).
- Έπειτα συνδέσαμε δύο υπολογιστές του τοπικού δικτύου μέσω ενός καλωδίου Ethernet.
- Στη συνέχεια χρησιμοποιήσαμε το εργαλείο tcpreplay, βλέπε [22], για την αναμετάδοση της κίνησης μεταξύ των δύο υπολογιστών με πολύ μεγαλύτερη ταχύτητα.
- Τέλος η δικτυακή κίνηση που προκλήθηκε, συλλέχθηκε με τη βοήθεια του εργαλείου Wireshark, βλέπε [23], από το δεύτερο υπολογιστή και το pcap αρχείο που προέκυψε δόθηκε ως είσοδος στο nProbe. Για να έχουμε στην έξοδο του nProbe μόνο ροές της Πλημμύρας UDP που υλοποιήσαμε, χρησιμοποιήσαμε φίλτρο καταγραφής, το οποίο καταγράφει μόνο πακέτα UDP.

Τέλος και σε αυτή την επίθεση αλλάξαμε το πεδίο ASN προορισμού των ροών Netflow, που πήραμε στην έξοδο του nProbe, σε αυτό του ΕΜΠ, όπως ακριβώς κάναμε και για τις επιθέσεις Πλημμύρα ICMP και Πλημμύρα SYN.

Σάρωση Θυρών (Port Scanning):

Κατά την επιλογή των ειδών των επιθέσεων που επιθυμούμε να αναγνωρίζει το νευρωνικό δίκτυο που κατασκευάσαμε, επιλέξαμε να συμπεριλάβουμε και μια επίθεση Port Scanning, με τελείως διαφορετικά χαρακτηριστικά από αυτά των υπολοίπων επιθέσεων Πλημμύρας, βλέπε ενότητα 2.1.1. Για την επίθεση αυτή χρησιμοποιήσαμε τεχνητή κίνηση. Έτσι υλοποιήσαμε μια επίθεση Port Scanning με TCP SYN πακέτα (TCP SYN scan), κάνοντας χρήση του εργαλείου nmap, βλέπε [24]. Πιο συγκεκριμένα εκτελέσαμε μια σάρωση των 1000 πιο γνωστών θυρών σε ένα μηχάνημα του τοπικού μας δικτύου.

Η δικτυακή κίνηση που προκλήθηκε από την επίθεση, συλλέχθηκε με τη βοήθεια του εργαλείου nProbe. Για να έχουμε στην έξοδο του nProbe μόνο ροές της επίθεσης Port Scanning, χρησιμοποιήσαμε φίλτρο καταγραφής, το οποίο καταγράφει μόνο πακέτα TCP, με την επιλογή –f ‘‘ip proto 6’’.

Τέλος και σε αυτή την επίθεση αλλάξαμε το πεδίο ASN προορισμού των ροών Netflow, που πήραμε στην έξοδο του nProbe, σε αυτό του ΕΜΠ, όπως ακριβώς κάναμε και για τις επιθέσεις Πλημμύρας.

Καλόβουλη κίνηση (Legitimate traffic):

Τέλος για την απόκτηση δεδομένων ‘’καλόβουλης κίνησης’’, χωρίς δηλαδή να περιέχονται πακέτα κάποιας επίθεσης, συλλέξαμε πραγματική κίνηση από υποδίκτυο εσωτερικό του ΕΜΠ και πιο συγκεκριμένα το δίκτυο του εργαστηρίου NETMODE. Η κίνηση αυτή περιλαμβάνει ένα μεγάλο εύρος διευθύνσεων αποστολής και προορισμού, οι οποίες ανήκουν εντός αλλά και εκτός του δικτύου του ΕΜΠ.

Το αρχείο καταγραφής pcap που προέκυψε χωρίστηκε σε κομμάτια πεντάλεπτης διάρκειας, με χρήση του προγράμματος ‘’editcap’’, βλέπε [25], με την επιλογή –i 300. Για τη δημιουργία ροών Netflow τροφοδοτήσαμε ένα από αυτά τα κομμάτια της καταγραφής στο εργαλείο nProbe.

4.2 Πεδία των ροών της κίνησης που καταγράφεται

Αναφέραμε σε προηγούμενες ενότητες ότι η δικτυακή κίνηση που χρησιμοποιήσαμε για την εκπαίδευση και την μετέπειτα λειτουργία του Νευρωνικού Δικτύου συλλέγεται με τη μορφή ροών IP κάνοντας χρήση του εργαλείου nProbe. Το nProbe λειτουργεί με το πρωτόκολλο Netflow και παρέχει τη δυνατότητα της επιλογής των πεδίων και των χαρακτηριστικών που επιθυμούμε να περιέχει κάθε ροή IP. Αυτά τα πεδία, έπειτα από κάποια επεξεργασία, είναι και τα πεδία του διανύσματος εισόδου που τροφοδοτείται στο Νευρωνικό Δίκτυο. Παρουσιάζουμε εδώ αναλυτικά όλα τα πεδία μιας ροής IP όπως αυτή προκύπτει στην έξοδο του nProbe και παράλληλα εξηγούμε την επεξεργασία που υφίσταται κάθε ένα από αυτά τα πεδία.

IPV4_SRC_ADDR:
Η IPv4 διεύθυνση αποστολέα των πακέτων που εντάσσονται στη ροή αυτή. Αν και το πεδίο αυτό δε δίνεται ως είσοδος στο Νευρωνικό Δίκτυο, χρησιμοποιείται κατά τη συλλογή των πακέτων από το nProbe και είναι απαραίτητο για το διαχωρισμό τους σε ροές.

IPV4_DST_ADDR:
Η IPv4 διεύθυνση προορισμού των πακέτων που εντάσσονται στη ροή αυτή. Ούτε το πεδίο αυτό δίνεται ως είσοδος στο Νευρωνικό Δίκτυο αλλά χρησιμοποιείται κατά τη συλλογή των πακέτων από το nProbe και είναι απαραίτητο για το διαχωρισμό τους σε ροές.

IN_PKTS:
Το πλήθος των εισερχόμενων πακέτων που ανήκουν στη ροή αυτή. Καθώς κάθε εγγραφή αντιστοιχεί σε ροή πακέτων δύο κατευθύνσεων κάνουμε λόγο για εισερχόμενα και εξερχόμενα πακέτα.

OUT_PKTS:
Το πλήθος των εξερχόμενων πακέτων που ανήκουν στη ροή αυτή.

IN_BYTES:
Ο συνολικός αριθμός των bytes όλων των εισερχόμενων πακέτων της ροής. Στο νευρωνικό δίκτυο δε δίνουμε απευθείας αυτόν τον ακέραιο αριθμό αλλά εκτελούμε μια ακέραια διαίρεση με το πλήθος των εισερχόμενων πακέτων (IN_PKTS), ώστε να βρούμε το μέσο αριθμό των bytes ενός εισερχόμενου πακέτου της ροής αυτής.

OUT_BYTES:
Ο συνολικός αριθμός των bytes όλων των εξερχόμενων πακέτων της ροής. Στο νευρωνικό δίκτυο δε δίνουμε απευθείας αυτόν τον ακέραιο αριθμό αλλά εκτελούμε μια ακέραια διαίρεση με το πλήθος των εξερχόμενων πακέτων (OUT_PKTS), ώστε να βρούμε το μέσο αριθμό των bytes ενός εξερχόμενου πακέτου της ροής αυτής.

PROTOCOL:
Ο κωδικός του πρωτοκόλλου που ενθυλακώνεται στα πακέτα IP της ροής. Καθώς όμως πρόκειται για έναν ακέραιο αριθμό χωρίς ιδιαίτερη σημασία για το νευρωνικό δίκτυο, η πληροφορία δίνεται σε αυτό με τη μορφή σημαιών. Έχουμε δηλαδή τρία πεδία ICMP, TCP και UDP τα οποία παίρνουν τιμές 1 ή 0, ανάλογα με το αν ο κωδικός αντιστοιχεί σε κάποιο από τα τρία πρωτόκολλα ή όχι. Αυτά τα πεδία δίνονται τελικά στο νευρωνικό δίκτυο.

ICMP_TYPE (ICMP Type και Code):
Το πεδίο αυτό αναφέρεται μόνο σε ροές με ICMP μηνύματα και διαφορετικά είναι μηδενικό. Στο πεδίο αυτό περιέχεται το Type αλλά και το Code του IPv4 ICMP μηνύματος. Πιο συγκεκριμένα ο ακέραιος αριθμός ICMP_TYPE που συλλέγουμε προκύπτει ως $(Type * 256) + Code$. Για να απομονώσουμε το κάθε πεδίο εκτελούμε ακέραια διαίρεση του πεδίου που δίνεται με τον αριθμό 256. Το πηλίκο της διαίρεσης είναι το πεδίο Type. Το υπόλοιπο της διαίρεσης μπορεί να πάρει τιμές από 0 έως 15. Οπότε έχουμε 15 πεδία-σημαίες τα οποία παίρνουν τιμές 1 ή 0, ανάλογα με το αν ο κωδικός αντιστοιχεί σε κάποιο από αυτά. Αυτά τα πεδία δίνονται τελικά στο νευρωνικό δίκτυο. Επιλέξαμε να δώσουμε στο δίκτυο μόνο το πεδίο Code και όχι το Type αλλά όπως θα δούμε στο πειραματικό στάδιο πετυχαίνουμε τη βέλτιστη δυνατή ακρίβεια για πακέτα ICMP και επομένως δεν υπάρχει λόγος να αλλάξουμε την επιλογή μας. Ωστόσο με την προσθήκη νέων επιθέσεων και την επέκταση του εργαλείου ίσως κριθεί απαραίτητο να τροφοδοτηθεί και το πεδίο Type στο νευρωνικό δίκτυο.

TCP_FLAGS:
Το πεδίο αυτό αφορά μόνο ροές με TCP πακέτα, διαφορετικά είναι μηδενικό. Αποτελείται από 6 bits τα οποία αντιστοιχούν στις σημαίες URG, ACK, PSH, RST, SYN και FIN του πρωτοκόλλου TCP. Κάθε ένα από αυτά είναι 1 ή 0, αν υπάρχει ή όχι αντίστοιχα κάποιο πακέτο στη ροή με αναμμένη την αντίστοιχη TCP σημαία. Οπότε η πληροφορία αυτή δίνεται ως έξι πεδία με τιμές 1 ή 0 στο νευρωνικό δίκτυο.

L4_SRC_PORT:
Η θύρα προέλευσης του στρώματος μεταφοράς, των πακέτων που ανήκουν στη ροή αυτή. Το πεδίο αυτό αφορά μόνο ροές που κάνουν χρήση κάποιου πρωτοκόλλου του στρώματος μεταφοράς, διαφορετικά είναι μηδενικό. Καθώς όμως πρόκειται για έναν ακέραιο αριθμό χωρίς ιδιαίτερη σημασία για το νευρωνικό δίκτυο, η πληροφορία δίνεται σε αυτό με τη μορφή σημαιών. Έχουμε δηλαδή πεδία τα οποία παίρνουν τιμές 1 ή 0, ανάλογα με το αν η θύρα αντιστοιχεί σε κάποια από τις ακόλουθες θύρες ή όχι. Διαλέξαμε να ελέγχουμε για την ύπαρξη ή όχι των ακόλουθων 19 θυρών που κρίνουμε ως πιο σημαντικές και επομένως είναι συχνά στόχος επιθέσεων: FTP data, FTP control, SSH, TELNET, SMTP, HOST NAME SERVER, DNS, BOOTP server, BOOTP client, TFTP, HTTP, POP3, SFTP, SQL Services, NTP, SQL Service, SNMP, BGP, HTTPS. Επίσης έχουμε και μια τελευταία σημαία για το αν η θύρα είναι μεγαλύτερη της θύρας 1023 και επομένως δεν είναι ευρέως γνωστή θύρα ή θύρα συστήματος. Αυτά τα 20 πεδία-σημαίες δίνονται τελικά στο νευρωνικό δίκτυο. Δηλαδή η

πληροφορία που δίνεται είναι αν τα πακέτα έχουν ως θύρα προέλευσης κάποια από τις παραπάνω, τις οποίες κρίνουμε ως πιο βασικές και συχνά χρησιμοποιούμενες.

L4_DST_PORT:
Η θύρα προορισμού του στρώματος μεταφοράς, των πακέτων που ανήκουν στη ροή αυτή. Το πεδίο αυτό αφορά μόνο ροές που κάνουν χρήση κάποιου πρωτοκόλλου του στρώματος μεταφοράς, διαφορετικά είναι μηδενικό. Κατά αντιστοιχία με το L4_SRC_PORT, η πληροφορία για αυτό το πεδίο παρέχεται στο νευρωνικό δίκτυο με τη μορφή 20 σημαιών ίδιας μορφής για τις ίδιες βασικές θύρες.

SRC_AS:
Το Autonomous System Number (ASN) του AS του αποστολέα των πακέτων, που εντάσσονται στη ροή αυτή, όπως αυτό προκύπτει από την IPv4 διεύθυνση αποστολέα των πακέτων. Η πληροφορία για αυτό το πεδίο δίνεται στο νευρωνικό δίκτυο με τη μορφή μίας σημαίας η οποία παίρνει τιμές 1 ή 0 αντίστοιχα αν το ASN είναι ίδιο με το ASN του δικτύου που παρακολουθούμε τη διερχόμενη κίνηση και εκτελούμε την κατηγοριοποίηση της. Έτσι στην περίπτωση μας η σημαία αυτή δηλώνει αν το SRC_AS είχε ή όχι την τιμή 3323, ASN του ΕΜΠ, όπου και εκτελέσαμε τα πειράματα.

DST_AS:
Το Autonomous System Number (ASN) του AS του παραλήπτη των πακέτων, που εντάσσονται στη ροή αυτή, όπως αυτό προκύπτει από την IPv4 διεύθυνση προορισμού των πακέτων. Σε αντιστοιχία με το SRC_AS η πληροφορία για αυτό το πεδίο παρέχεται στο νευρωνικό δίκτυο με τη μορφή σημαίας η οποία δηλώνει αν το DST_AS είχε ή όχι την τιμή 3323. Με τη χρήση των δύο σημαιών από τα πεδία SRC_AS και DST_AS το νευρωνικό δίκτυο μπορεί να ‘’καταλαβαίνει’’ την κατεύθυνση μιας καταγεγραμμένης ροής.

Για ροές των δεδομένων εκπαίδευσης στις οποίες δε συμμετείχε καθόλου το δίκτυο του ΕΜΠ (πχ κίνηση αντλούμενη από το αρχείο επίθεσης της CAIDA, βλέπε ενότητα 4.1), τέθηκε το DST_AS του θύματος να είναι το 3323 ώστε το νευρωνικό δίκτυο να μπορεί να καταλάβει ότι πρόκειται για ροή κίνησης εισερχόμενης στο διαχειριζόμενο δίκτυο, το οποίο στα πειράματα της εργασίας ανήκει στο AS του πολυτεχνείου.

FLOW_START_MILLISECONDS:
Η χρονική σφραγίδα (timestamp), δοσμένη σε δευτερόλεπτα από την έναρξη λειτουργίας του Εξαγωγέα, του πρώτου καταγεγραμμένου πακέτου της ροής. Το πεδίο αυτό δε δίνεται ως είσοδος στο νευρωνικό δίκτυο. Χρησιμοποιείται όμως κατά τον υπολογισμό της διάρκειας της ροής. Η διάρκεια δίνεται στο νευρωνικό δίκτυο με ένα πεδίο και υπολογίζεται ως: $duration = \text{FLOW\_START\_MILLISECONDS} - \text{FLOW\_END\_MILLISECONDS}$

FLOW_END_MILLISECONDS:
Η χρονική σφραγίδα (timestamp), δοσμένη σε δευτερόλεπτα από την έναρξη λειτουργίας του Εξαγωγέα, του τελευταίου καταγεγραμμένου πακέτου της ροής. Ούτε το πεδίο αυτό δίνεται ως είσοδος στο Νευρωνικό Δίκτυο.

Τα πεδία που αναλύσαμε επιλέχτηκαν να περιέχονται στην έξοδο του nProbe με την επιλογή -T "${format}", η οποία ορίζει ένα template από πεδία. Η μεταβλητή ${format} περιέχει τη συμβολοσειρά του template και στην περίπτωσή μας είναι:

format="%IPV4_SRC_ADDR %IPV4_DST_ADDR %IN_PKTS %OUT_PKTS %IN_BYTES %OUT_BYTES %PROTOCOL %ICMP_TYPE %TCP_FLAGS %L4_SRC_PORT %L4_DST_PORT %SRC_AS %DST_AS %FLOW_START_MILLISECONDS %FLOW_END_MILLISECONDS"

4.3 Κατασκευή και εκπαίδευση των Νευρωνικών Δικτύων

Για την υλοποίηση και την εκπαίδευση των Νευρωνικών δικτύων χρησιμοποιήσαμε τη γλώσσα προγραμματισμού Python. Αν και πρόκειται για interpreter γλώσσα, η χρονική καθυστέρηση που εισάγει το χαρακτηριστικό της αυτό δεν κρίθηκε τόσο σημαντική για τα πειράματα που θέλουμε να εκτελέσουμε. Ωστόσο περιμένουμε ότι με τη χρήση διαφορετικής γλώσσας, όπως για παράδειγμα Matlab, θα είχαμε τα ίδια αποτελέσματα όσον αφορά την ακρίβεια των Νευρωνικών Δικτύων.

Κύριος λόγος της επιλογής της Python είναι ότι περιέχει τη βιβλιοθήκη Keras [27], μια από τις πιο εύχρηστες και πλήρεις βιβλιοθήκες μηχανικής μάθησης με τη χρήση νευρωνικών δικτύων που υπάρχουν τη στιγμή της εκπόνησης της παρούσας διπλωματικής εργασίας.

Κώδικας της Διπλωματικής Εργασίας:

Ο κώδικας που υλοποιήθηκε κατά την εκπόνηση της διπλωματικής εργασίας και χρησιμοποιήθηκε για την υλοποίηση του μηχανισμού και των διαφόρων πειραμάτων βρίσκεται αποθηκευμένος στην πλατφόρμα GitHub και ο σχετικός σύνδεσμος είναι ο ακόλουθος:

https://github.com/StamatisKourkoutas/Web-Traffic-Classifier

5 Πειραματικό Στάδιο – Αξιολόγηση Αποτελεσμάτων

5.1 Πρώτη προσέγγιση – Χρήση δικτύων MLP

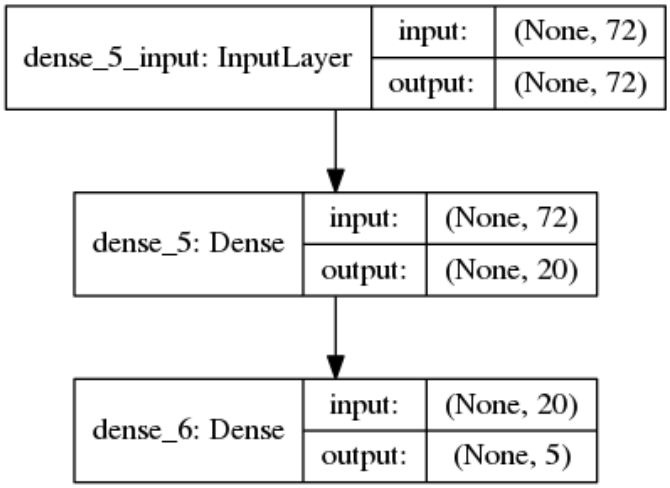
Σε μια πρώτη προσπάθεια για την κατασκευή ενός κατάλληλου νευρωνικού δικτύου για τον όσο το δυνατόν καλύτερο διαχωρισμό της κίνησης των 5 κατηγοριών εκτελούμε το ακόλουθο πείραμα.

Αρχικά κατασκευάζουμε διάφορα νευρωνικά δίκτυα τα οποία διαφέρουν σε μέγεθος τόσο ως προς **το πλήθος των κρυφών επιπέδων** που περιείχαν, ο οποίος κυμάνθηκε από 1 μέχρι 4, αλλά και ως προς **το πλήθος των νευρώνων** που περιείχε κάθε κρυφό επίπεδο, το οποίο κυμάνθηκε από 5 μέχρι 30. Τα νευρωνικά αυτά δίκτυα τα εκπαιδεύουμε με τον πιο ευρέως χρησιμοποιούμενο αλγόριθμο εκπαίδευσης, τη **Στοχαστική Κατάβαση Δυναμικού** (SGD), βλέπε ενότητα 2.2.7, **με ρυθμό μάθησης (learning rate) ίσο με 0,01**.

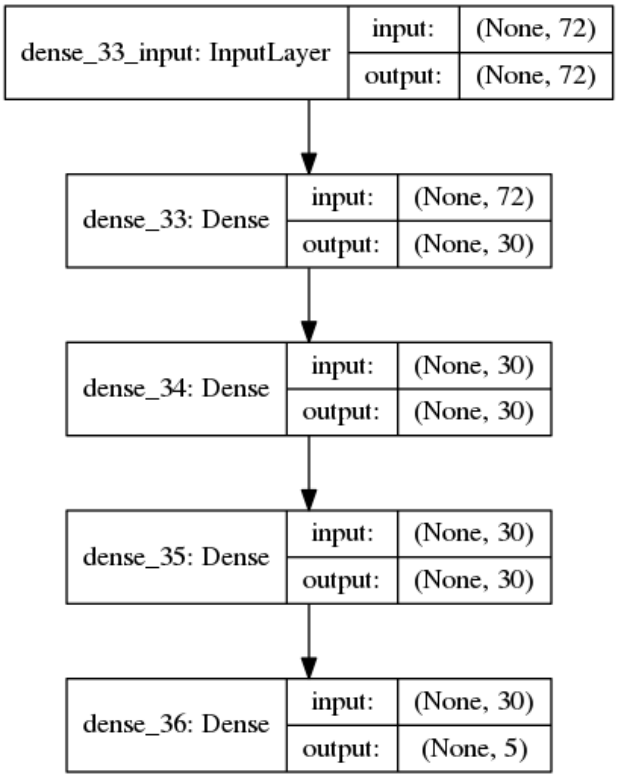
Τα δεδομένα που χρησιμοποιήσαμε κατά την εκπαίδευση είναι αυτά που αναφέρουμε στην ενότητα 4.4.1 και τροφοδοτήθηκαν στα νευρωνικά δίκτυα με τη μορφή που αναφέρονται στην ενότητα 4.4.2. Το πρώτο Dataset που χρησιμοποιήσαμε για την εκπαίδευση και τον έλεγχο των δικτύων περιέχει **50.000 ροές που αντλήθηκαν ισάριθμα από τις 5 κατηγορίες ροών που μελετάμε. Τα 40000 δείγματα χρησιμοποιήθηκαν για την εκπαίδευση ενώ τα 10000 για τη φάση ελέγχου (test) των νευρωνικών δικτύων** και την εξαγωγή αντίστοιχα της μετρικής Validation Accuracy που εμφανίζουμε στα αποτελέσματα.

Σε αυτή την πρώτη προσέγγιση του προβλήματος κάναμε κάποιες απλοποιήσεις για να πάρουμε μια πρώτη εκτίμηση, οι οποίες όμως δεν έγιναν αυθαίρετα αλλά προέκυψαν έπειτα από την εκπαίδευση μικρών νευρωνικών δικτύων. Όσον αφορά τον αριθμό των εποχών που διήρκησε η εκπαίδευση τον θέσαμε ίσο με **10 εποχές** καθώς παρατηρήσαμε ότι τα νευρωνικά δίκτυα δεν συνέκλιναν περαιτέρω παρόλο που πειραματιστήκαμε με μεγαλύτερο αριθμό εποχών. Τα δείγματα των δεδομένων που διαθέτουμε παρουσιάζουν αρκετή ομοιότητα μεταξύ τους οπότε 50000 δείγματα με διάρκεια εκπαίδευσης 10 εποχές κρίνεται αρκετό για τη σύγκλιση των δικτύων. Τέλος θέσαμε το batch size κατά τη διαδικασία της εκπαίδευσης να είναι ίσο με 1.

Ακολουθεί σχηματική αναπαράσταση των επιπέδων δύο τυχαίων νευρωνικών δικτύων MLP από αυτά που κατασκευάστηκαν για το πείραμα. Η αναπαράσταση αυτή προέκυψε κατευθείαν από εντολές της βιβλιοθήκης keras που χρησιμοποιούμε για την κατασκευή των δικτύων.

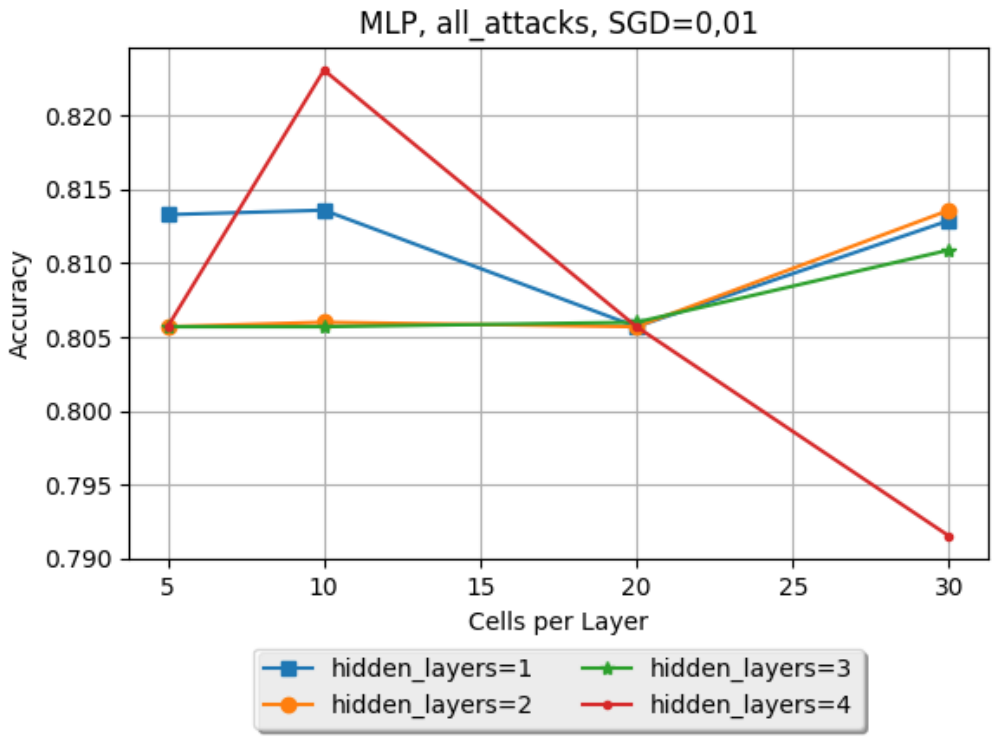


Σχήμα 15: MLP δίκτυο με ένα κρυφό επίπεδο. Στο σχήμα βλέπουμε το επίπεδο εισόδου, το οποίο δέχεται διάνυσμα εισόδου μεγέθους 72 πεδίων, ένα κρυφό επίπεδο με 20 νευρώνες και το επίπεδο εξόδου το οποίο έχει μέγεθος 5 νευρώνες, καθώς εκτελούμε ταξινόμηση μεταξύ 5 διαφορετικών κατηγοριών. Για κάθε διάνυσμα εισόδου παίρνουμε τη μεγαλύτερη τιμή (πιο κοντά στη μονάδα) σε μια από τις 5 εξόδους η οποία αντιστοιχεί στην κατηγορία του δείγματος που τροφοδοτήσαμε. Η τιμή None τυπώνεται στη θέση του πλήθους των δειγμάτων που δίνεται στο νευρωνικό δίκτυο, το οποίο πλήθος μας ενδιαφέρει μόνο στη φάση της εκπαίδευσης αλλά δεν έχει σχέση με τη δομή του δικτύου για να τυπωθεί ως σταθερά αυτού από τη βιβλιοθήκη Keras.



Σχήμα 16: MLP δίκτυο με τρία κρυφά επίπεδα. Στο σχήμα βλέπουμε το επίπεδο εισόδου, το οποίο δέχεται διάνυσμα εισόδου μεγέθους 72 πεδίων, τρία κρυφά επίπεδα με 30 νευρώνες το καθένα και το επίπεδο εξόδου το οποίο έχει μέγεθος 5 νευρώνες, καθώς εκτελούμε ταξινόμηση μεταξύ 5 διαφορετικών κατηγοριών. Για κάθε διάνυσμα εισόδου παίρνουμε τη μεγαλύτερη τιμή (πιο κοντά στη μονάδα) σε μια από τις 5 εξόδους η οποία αντιστοιχεί στην κατηγορία του δείγματος που τροφοδοτήσαμε.

Τα αποτελέσματα που προέκυψαν από τα νευρωνικά που εκπαιδεύσαμε παρουσιάζονται στην ακόλουθη γραφική παράσταση.

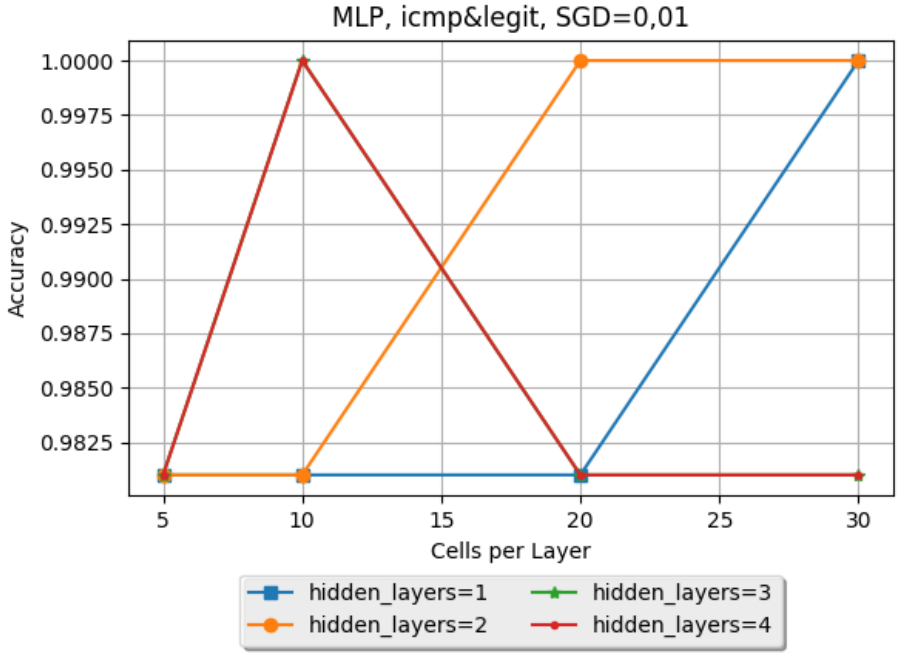


Σχήμα 17: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples και από τις 5 κατηγορίες.

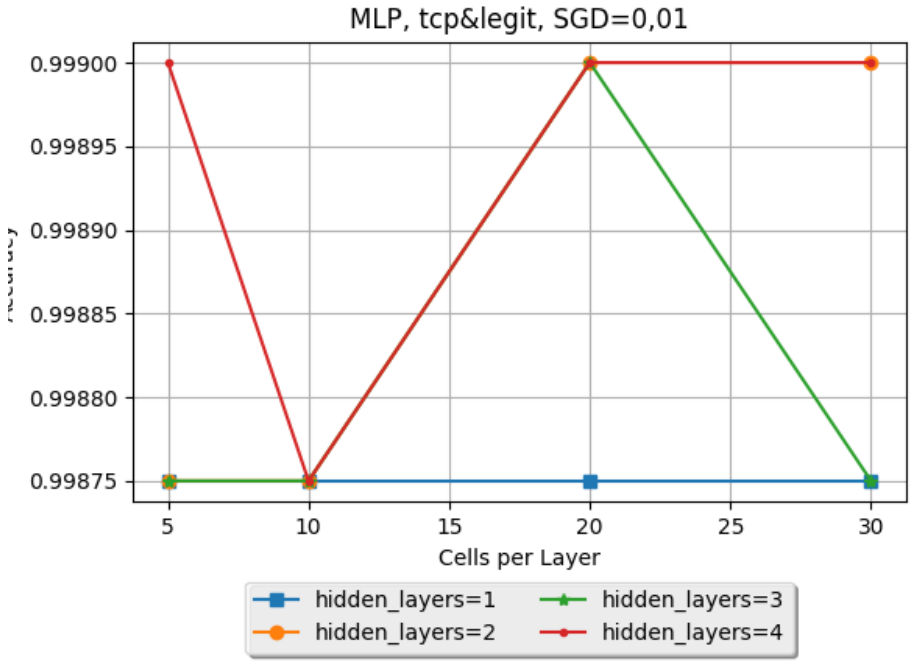
Το βασικό συμπέρασμα που εξάγουμε από την παραπάνω γραφική είναι ότι κανένα από τα νευρωνικά δίκτυα MLP που εκπαιδεύσαμε δεν ήταν σε θέση να ξεχωρίσει πλήρως τα δεδομένα των 5 κατηγοριών. Για το λόγο αυτό η ακρίβεια κυμάνθηκε από 80,57% μέχρι 82,31%.

5.2 Εκτέλεση ίδιου πειράματος αλλάζοντας τα δεδομένα εκπαίδευσης

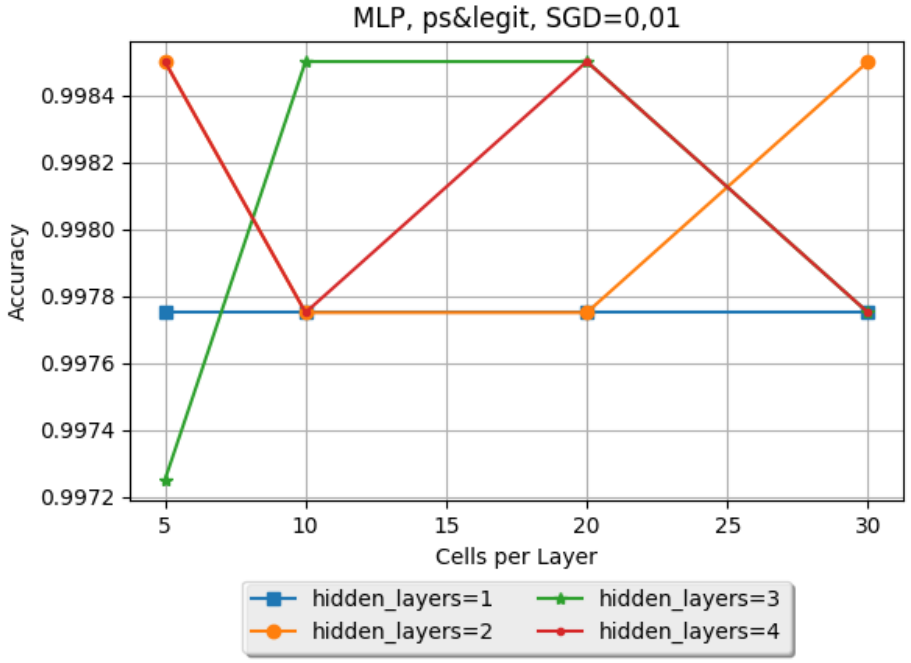
Για να διαπιστώσουμε το πρόβλημα κατά την εκπαίδευση εκτελούμε το ίδιο ακριβώς πείραμα αλλά με διαφορετικά δεδομένα εκπαίδευσης κάθε φορά. Έτσι προκύπτουν οι ακόλουθες γραφικές από τα αντίστοιχα πειράματα.



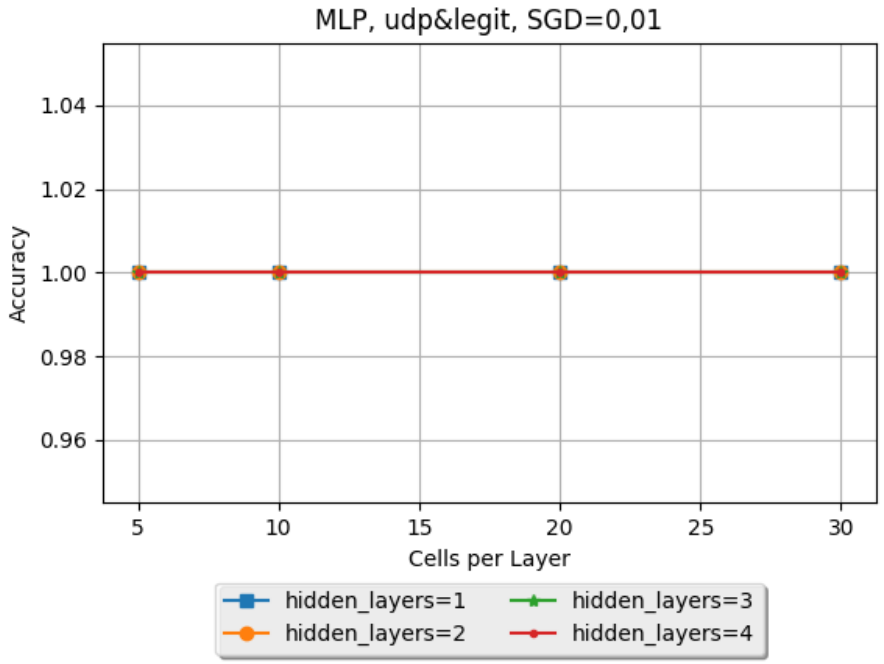
Σχήμα 18: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples και από την επίθεση icmp flood και την καλόβουλη κίνηση.



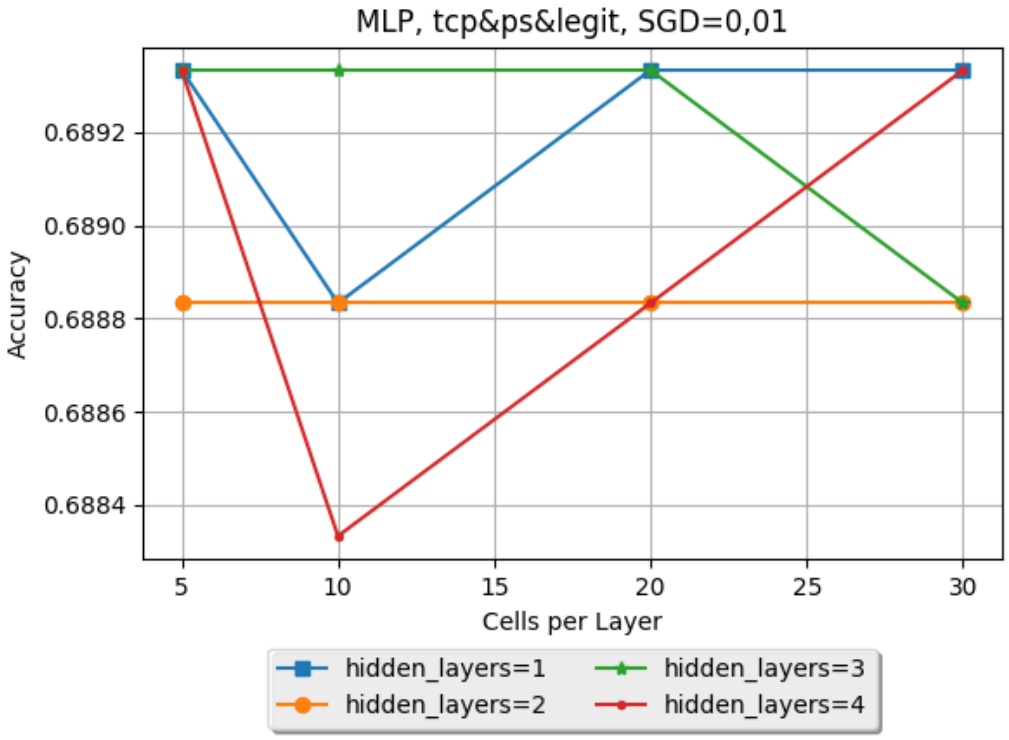
Σχήμα 19: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples από την επίθεση tcp syn flood και την καλόβουλη κίνηση.



Σχήμα 20: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples από την επίθεση port scanning και την καλόβουλη κίνηση.



Σχήμα 21: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples από την επίθεση udp flood και την καλόβουλη κίνηση.



Σχήμα 22: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples από τις επιθέσεις tcp syn flood, port scanning και την καλόβουλη κίνηση.

Από τα αποτελέσματα των πειραμάτων τα οποία συνοψίσαμε στις παραπάνω γραφικές παρατηρούμε ότι οι επιμέρους επιθέσεις μπορούν να ξεχωριστούν εύκολα και με πολύ καλή ακρίβεια (validation accuracy) από την καλόβουλη κίνηση. Όπως διαπιστώνουμε από τις 4 πρώτες γραφικές παραστάσεις ακόμα και τα πιο ρηχά και μικρά σε πλάτος νευρωνικά δίκτυα MLP συγκλίνουν ικανοποιητικά και μάλιστα δίνουν ακρίβεια πολύ κοντά στη μονάδα.

Ωστόσο όπως είδαμε στην ενότητα 5.1 κανένα MLP νευρωνικό δίκτυο δε μπόρεσε να πετύχει τόσο υψηλή ακρίβεια στο dataset με όλα τα είδη κίνησης και από τις 5 κατηγορίες. Το πρόβλημα γίνεται περισσότερο εμφανές από την τελευταία γραφική παράσταση που παρουσιάσαμε, βλέπε σχήμα 22. Η εκπαίδευση των δικτύων εδώ έγινε σε dataset το οποίο περιείχε δείγματα (samples) από τις επιθέσεις tcp syn flood, port scanning και την καλόβουλη κίνηση. Καθώς και οι δύο επιθέσεις αποτελούνται αποκλειστικά από ροές με πακέτα tcp syn, τα δίκτυα δυσκολεύτηκαν να διαχωρίσουν αυτές τις δύο κατηγορίες μεταξύ τους. Ωστόσο τονίζουμε ότι τα δείγματα καλόβουλης κίνησης περιέχουν και ροές με πακέτα tcp syn τα οποία όπως φαίνεται από τα σχήματα 19 και 20 ξεχωρίζονται επιτυχώς από τα νευρωνικά δίκτυα MLP.

5.3 Προσέγγιση του προβλήματος με Νευρωνικά Δίκτυα RNN

Είναι προφανές ότι οι πληροφορίες των ροών που τροφοδοτούνται στα νευρωνικά δίκτυα MLP στις προηγούμενες ενότητες δεν επαρκούν για τον ικανοποιητικό διαχωρισμό των επιθέσεων tcp syn flood και port scanning. Περαιτέρω εξέταση των ροών που προκύπτουν στην έξοδο του εργαλείου nProbe επιβεβαιώνει αυτόν τον ισχυρισμό.

Αναλυτική επεξήγηση του προβλήματος σύγκλισης που παρατηρήσαμε:

Όπως είναι φυσικό τα δείγματα της σάρωσης θυρών (port scanning) είναι κυρίως ροές οι οποίες περιέχουν ένα tcp πακέτο, μικρού μεγέθους, με σημαία syn και κανένα πακέτο ως απάντηση όταν η εξεταζόμενη θύρα είναι κλειστή. Θα περίμενε κανείς τα δείγματα της επίθεσης tcp syn flood να είναι κυρίως ροές με χιλιάδες εισερχόμενα πακέτα tcp με σημαία syn. Ωστόσο παρατηρούμε ότι και σε αυτήν την περίπτωση οι ροές αποτελούνται κυρίως από ένα tcp πακέτο μικρού μεγέθους με σημαία syn, ίδιες δηλαδή με την επίθεση σάρωσης θυρών. Αυτή η ομοιότητα των δειγμάτων των δύο επιθέσεων οφείλεται στο γεγονός ότι ο επιτιθέμενος χρησιμοποιεί μια τεχνική αποφυγής εντοπισμού της επίθεσης.

Πιο συγκεκριμένα, κατά την εξέλιξη της επίθεσης tcp syn flood του αρχείου της Caida, ο επιτιθέμενος φροντίζει ώστε σε κάθε νέο πακέτο ενός υπολογιστή, ο οποίος συμμετέχει στην επίθεση προς το θύμα, να αλλάζει κάθε φορά η θύρα προέλευσης του στρώματος μεταφοράς (L4 source port) των πακέτων και να μη χρησιμοποιείται η ίδια. Έτσι ο θύτης από κάθε υπολογιστή στη διάθεσή του επιτίθεται σε συγκεκριμένη διεύθυνση IPv4 και θύρα προορισμού, σαρώνοντας όμως όλες τις διαθέσιμες θύρες προέλευσης του μολυσμένου υπολογιστή που συμμετέχει στην επίθεση και χρησιμοποιώντας μόνο μία φορά κάθε θύρα. Σαν αποτέλεσμα, εφόσον κάθε tcp πακέτο της επίθεσης διαθέτει διαφορετική θύρα προορισμού, ανήκει και σε διαφορετική ροή IP. Επομένως στην έξοδο του εργαλείου nProbe προκύπτουν και σε αυτή την περίπτωση ροές που αποτελούνται από μόνο ένα μικρό πακέτο σε αντίθεση με τις προσδοκίες μας.

Πρόταση επίλυσης προβλήματος - Προσθήκη μνήμης με την ‘’ιστορία’’ κάθε επίθεσης με τη χρήση δικτύων RNN:

Για την αντιμετώπιση του παραπάνω προβλήματος που περιγράψαμε θέλουμε να αποφύγουμε την προσθήκη επιπλέον χαρακτηριστικών στο διάνυσμα εισόδου των Νευρωνικών Δικτύων. Προσθήκη περισσότερων πεδίων όπως για παράδειγμα η ακριβής θύρα προέλευσης του στρώματος μεταφοράς κατά πάσα πιθανότητα θα βοηθούσε στο διαχωρισμό των δειγμάτων των δύο επιθέσεων. Ωστόσο, τέτοια συγκεκριμένα χαρακτηριστικά θα οδηγούσαν σε υπερβολική εκπαίδευση των νευρωνικών δικτύων στις επιθέσεις, βλέπε ενότητα 4.1, και σαν αποτέλεσμα τα νευρωνικά δίκτυα που θα εκπαιδεύαμε **θα έχαναν τη δυνατότητα της γενίκευσης για επιθέσεις που προέρχονται από έναν διαφορετικό αποστολέα**. Αυτό μπορεί να διαπιστωθεί άμεσα με την εκτέλεση αντίστοιχου πειράματος. Τα χαρακτηριστικά που επιλέξαμε να τροφοδοτήσουμε στα Νευρωνικά δίκτυα και παρουσιάζουμε στην ενότητα 4.2 επιλέχθηκαν με αυτό το κριτήριο και δεν επιθυμούμε να τα αλλάξουμε.

Μια πιθανή λύση στο πρόβλημα είναι να προσπαθήσουμε να **εισάγουμε στα Δίκτυα που κατασκευάζουμε και εκπαιδεύουμε μνήμη με την ιστορία κάθε επίθεσης**, χωρίς όμως

εξειδικευμένα χαρακτηριστικά για τον εκάστοτε επιτιθέμενο για να μη χάσουμε τη δυνατότητα γενίκευσης. Αυτή η μνήμη που επιθυμούμε μπορεί να εκφραστεί με την προσθήκη ανάδρασης στα νευρωνικά δίκτυα, βλέπε ενότητα 2.2.9. Οπότε στο στάδιο αυτό θα πειραματιστούμε με Recurrent Neural Networks (RNN), για την επίλυση του προβλήματος.

Είδος ανάδρασης που χρησιμοποιήθηκε:

Επιθυμούμε να συνδέσουμε κάθε ροή ή δείγμα που τροφοδοτούμε στο νευρωνικό δίκτυο με τις προηγούμενες οπότε χρησιμοποιούμε **ανάδραση από τους νευρώνες κάθε κρυφού επιπέδου στον εαυτό τους** ώστε να κρατηθεί η προηγούμενη κατάσταση τους και να συνυπολογιστεί στον υπολογισμό της εξόδου στην τωρινή ροή. Δηλαδή χρησιμοποιούμε ανάδραση όπως στο σχήμα 10 που παρουσιάσαμε στην ενότητα 2.2.9.

Για να επιτύχουμε αυτό δίνουμε κάθε ροή ξεχωριστά στο νευρωνικό δίκτυο RNN χωρίς να προσδιορίζουμε μήκος ακολουθίας όπως συνήθως προσδιορίζουμε στη βιβλιοθήκη Keras αλλά θέτοντας το μήκος ακολουθίας ίσο με τη μονάδα. Ωστόσο θέτουμε **την επιλογή stateful να είναι αληθής**. Με αυτόν τον τρόπο η εσωτερική κατάσταση όλων των νευρώνων κατά την τροφοδότηση ενός δείγματος κάθε batch κατά την εκπαίδευση και τον έλεγχο του δικτύου χρησιμοποιείται μαζί με το δείγμα που βρίσκεται στην ίδια θέση στο επόμενο batch για τον υπολογισμό της επόμενης εξόδου. Θέτοντας **το μέγεθος των batch να είναι ίσο με τη μονάδα** καταφέρνουμε να χρησιμοποιήσουμε την εσωτερική κατάσταση του δικτύου από την προηγούμενη ροή στην αμέσως επόμενη. Έτσι μπορούμε εμμέσως να συνυπολογίσουμε όλη την ιστορία των προηγούμενων εισερχόμενων ροών για κάθε νέα ροή μέσω των βρόγχων ανάδρασης.

Λεπτομέρειες του πειράματος:

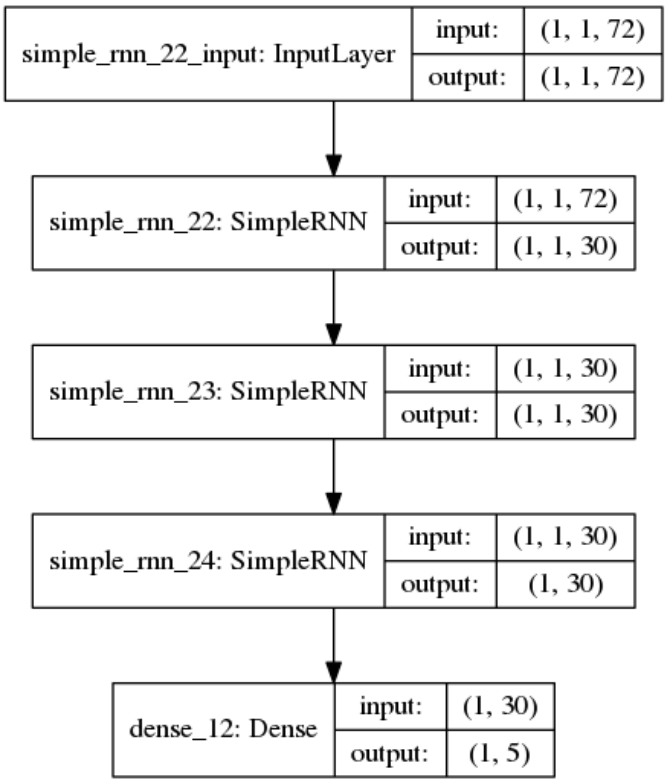
Κατά αντιστοιχία με τα προηγούμενα πειράματα κατασκευάζουμε διάφορα νευρωνικά δίκτυα RNN τα οποία διαφέρουν σε μέγεθος τόσο ως προς **το πλήθος των κρυφών επιπέδων** που περιείχαν, ο οποίος κυμάνθηκε από 1 μέχρι 3, αλλά και ως προς **το πλήθος των νευρώνων** που περιείχε κάθε κρυφό επίπεδο, το οποίο κυμάνθηκε από 5 μέχρι 30. Χρησιμοποιούμε και αυτή τη φορά ως μέθοδο εκπαίδευσης, τη **Στοχαστική Κατάβαση Δυναμικού** (SGD), βλέπε ενότητα 2.2.7, **με ρυθμό μάθησης (learning rate) ίσο με 0,01**. Τα δεδομένα που χρησιμοποιήσαμε κατά την εκπαίδευση είναι τα ίδια με προηγουμένως. Όσον αφορά τον αριθμό των εποχών που διήρκησε η εκπαίδευση τον θέσαμε ίσο με **10 εποχές**. Τέλος ως **συναρτήσεις ενεργοποίησης στους νευρώνες των επιπέδων χρησιμοποιήσαμε τη συνάρτηση tanh αντί της συνάρτησης relu** που χρησιμοποιήσαμε προηγουμένως καθώς για RNN δίκτυα με συνάρτηση σφάλματος την categorical cross-entropy δεν ενδείκνυται η χρήση της relu στη βιβλιοθήκη Keras.

Τέλος το Dataset που χρησιμοποιήσαμε για την εκπαίδευση και τον έλεγχο των δικτύων είναι το ίδιο ακριβώς με προηγουμένως και περιέχει **50.000 ροές που αντλήθηκαν ισάριθμα από τις 5 κατηγορίες ροών που μελετάμε. Τα 40000 δείγματα χρησιμοποιήθηκαν για την εκπαίδευση ενώ τα 10000 για τη φάση ελέγχου (test) των νευρωνικών δικτύων** και την εξαγωγή αντίστοιχα της μετρικής Validation Accuracy που εμφανίζουμε στα αποτελέσματα.

Ωστόσο αυτή τη φορά **παίζει μεγάλο ρόλο η σειρά με την οποία παρουσιάζουμε τις ροές κατά την εκπαίδευση.** Αρχικά εκτελούμε το πείραμα αφήνοντας τη βιβλιοθήκη του

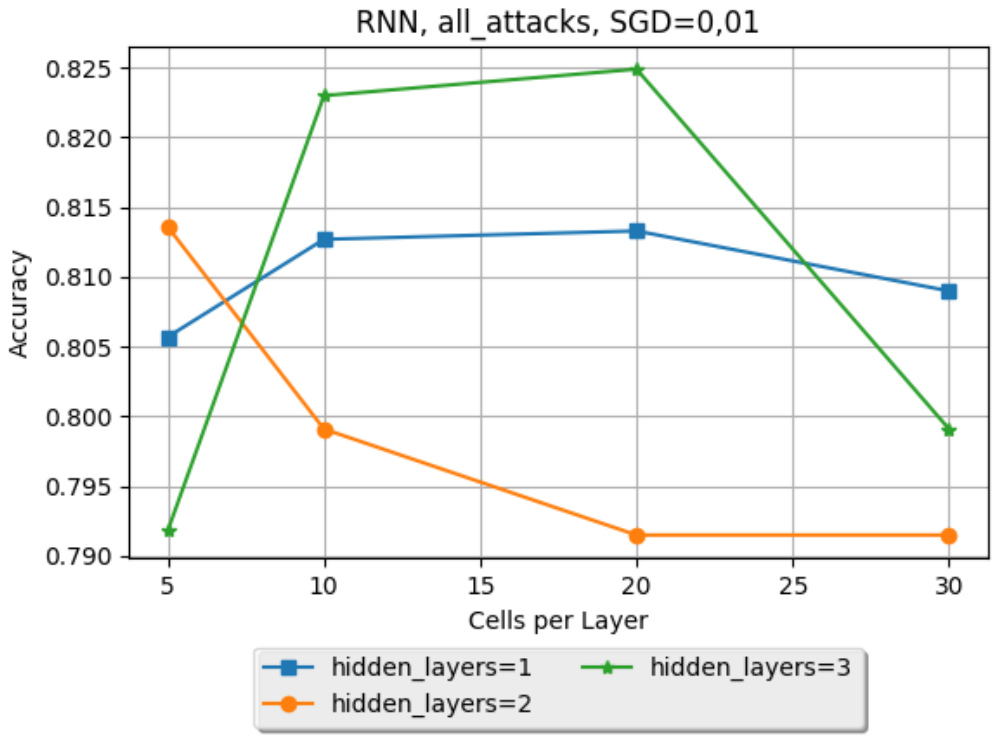
Keras να ανακατεύσει τα δεδομένα του dataset με τυχαία σειρά όπως και στα MLP και έτσι χάνεται το ιστορικό των δεδομένων.

Ακολουθεί σχηματική αναπαράσταση των επιπέδων ενός τυχαίου νευρωνικού δικτύου RNN από αυτά που κατασκευάστηκαν για το πείραμα. Η αναπαράσταση αυτή προέκυψε κατευθείαν από εντολές της βιβλιοθήκης Keras που χρησιμοποιούμε για την κατασκευή των δικτύων.



Σχήμα 23: RNN δίκτυο με τρία κρυφά επίπεδα. Στο σχήμα βλέπουμε το επίπεδο εισόδου, το οποίο δέχεται διάνυσμα εισόδου μεγέθους 72 πεδίων, τρία κρυφά επίπεδα με 30 νευρώνες το καθένα και το επίπεδο εξόδου το οποίο έχει μέγεθος 5 νευρώνες, καθώς εκτελούμε ταξινόμηση μεταξύ 5 διαφορετικών κατηγοριών. Τα διανύσματα που φαίνονται στο σχήμα σε κάθε επίπεδο περιέχουν (batch size, sequence length, input characteristics). Το μέγεθος των batch είναι ίσο με 1, το μέγεθος της ακολουθίας που δίνεται στο νευρωνικό δίκτυο για κάθε δείγμα (sample) είναι 1 αλλά κάνουμε χρήση της επιλογής stateful, βλέπε είδος ανάδρασης που χρησιμοποιήθηκε σε αυτή την ενότητα.

Τα αποτελέσματα που προέκυψαν από τα νευρωνικά που εκπαιδεύσαμε παρουσιάζονται στην ακόλουθη γραφική παράσταση.



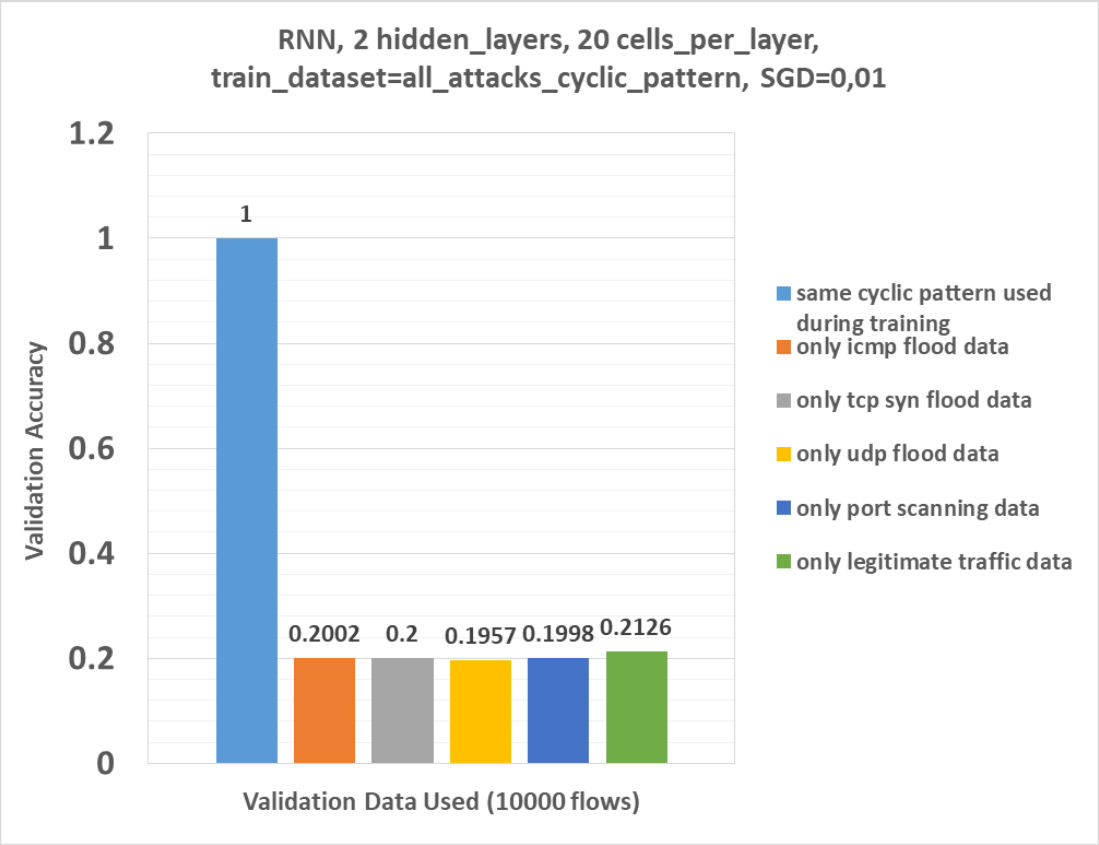
Σχήμα 24: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός RNN νευρωνικού δικτύου, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples και από τις 5 κατηγορίες ανακατεμένα με τυχαίο τρόπο.

Όπως παρατηρούμε τα RNN δίκτυα που προέκυψαν έχουν όλα περίπου ίδια χαμηλή ακρίβεια όπως τα MLP δίκτυα και το πρόβλημα δεν έχει λυθεί. Κατά την εκπαίδευση παρατηρήσαμε ότι παίζει τεράστιο ρόλο η σειρά με την οποία παρουσιάζονται τα δεδομένα του dataset. Στο πείραμα αφήσαμε τη βιβλιοθήκη του Keras να ανακατεύσει τα δεδομένα του dataset με τυχαία σειρά και έτσι χάθηκε το ιστορικό των δεδομένων. Δηλαδή αντιμετωπίζουμε κάθε ροή ως ανεξάρτητη από τις προηγούμενες. Οπότε **τα δίκτυα RNN δε βρήκαν κατάλληλες χρονικές εξαρτήσεις μεταξύ των ροών και έτσι εκφυλίζονται σε ακρίβεια ίδια με αυτή των MLP. Η ανάδραση δε λήφθηκε σημαντικά υπόψη και αντιμετωπίστηκε ως θόρυβος κατά την εκπαίδευση.**

Ωστόσο όπως βλέπουμε στη συνέχεια **τα παραπάνω αποτελέσματα δεν διαφέρουν ποιοτικά από όταν φροντίζουμε να διατηρηθεί η χρονική σειρά των δεδομένων και να ανακατευτούν όμως οι επιμέρους κατηγορίες μεταξύ τους με τυχαίο τρόπο κατά την εκπαίδευση**.

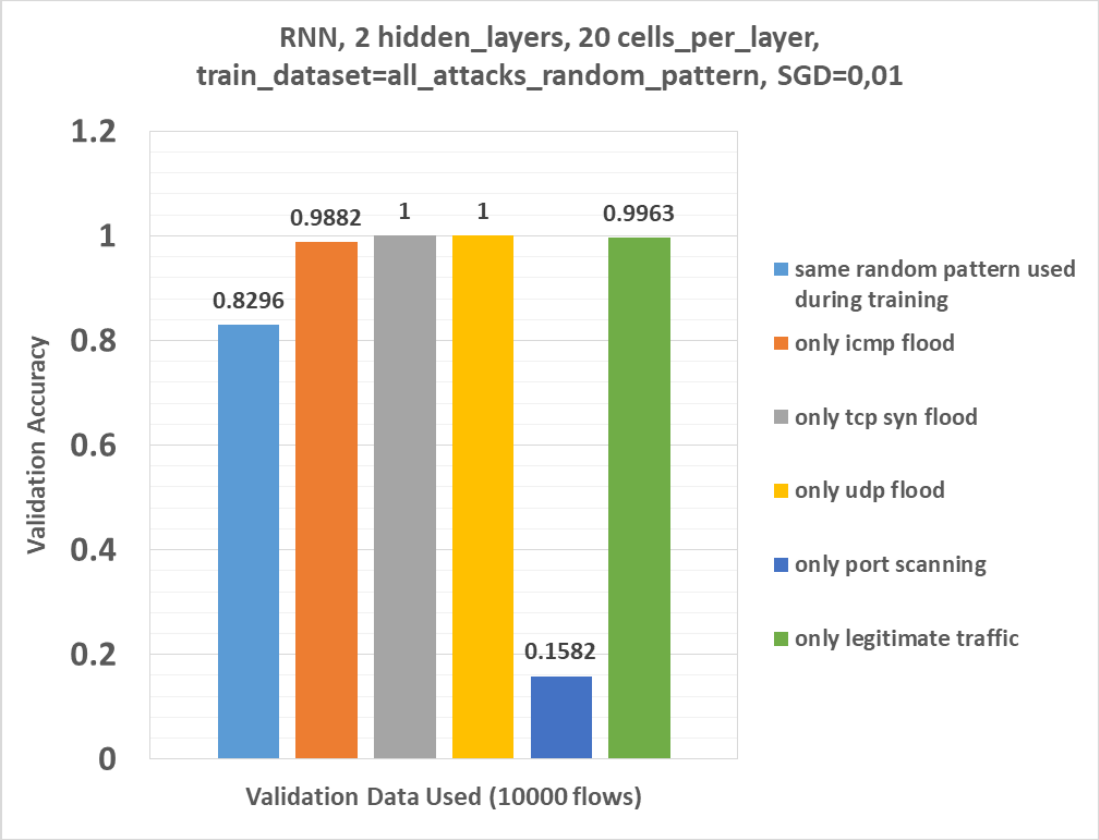
5.4 Μελέτη της σημαντικότητας της σειράς τροφοδότησης και του συνδυασμού των ροών διαφορετικών κατηγοριών κατά την εκπαίδευση και τον έλεγχο δικτύων RNN.

Εκπαιδεύουμε ένα RNN (τυχαία επιλέγουμε το πλήθος των κρυφών επιπέδων να είναι 2 και το πλήθος των νευρώνων που περιέχει κάθε κρυφό επίπεδο να είναι 20) με ίδιες παραμέτρους με προηγουμένως και με τα ίδια δεδομένα που όμως παρουσιάζονται αυτή τη φορά με ένα συγκεκριμένο pattern κατά την εκπαίδευση του δικτύου. Έτσι έχουμε τα ακόλουθα.



Σχήμα 25: Γραφική αναπαράσταση της ακρίβειας πάνω σε διάφορα δεδομένα ελέγχου (validation accuracy) ενός RNN νευρωνικού δικτύου με 2 κρυφά επίπεδα και 20 νευρώνες ανά κρυφό. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) περιείχε δεδομένα από όλες τις επιμέρους κατηγορίες οι οποίες τροφοδοτήθηκαν στο δίκτυο με ένα συγκεκριμένο κυκλικό σχήμα. Πιο συγκεκριμένα δίναμε μία ροή από κάθε κατηγορία ακολουθώντας τη σειρά ‘’κατ. 1, κατ. 2, κατ. 3, κατ. 4, κατ. 5, κατ. 1, κατ. 2, …’’.

Εκπαίδευση της παραπάνω μορφής έχει σαν αποτέλεσμα το RNN δίκτυο να **υπερεκπαιδεύεται σε μια συγκεκριμένη σειρά εμφάνισης των δεδομένων και να εκπαιδεύεται ελλιπώς πάνω στα χαρακτηριστικά της κάθε κατηγορίας**. Έτσι παρατηρούμε τέλεια ακρίβεια (validation accuracy) κατά τον έλεγχο σε dataset της ίδιας μορφής, με το ίδιο σχήμα, αλλά πολύ χαμηλή ακρίβεια σε οποιοδήποτε άλλο dataset όπως για παράδειγμα αν εξετάσουμε μόνο ροές μιας κατηγορίας στον έλεγχο. Αυτό υποδηλώνει **αστοχία των δομών RNN αν ο συνδυασμός των κατηγοριών κατά την εκπαίδευση δεν εμπεριέχει γενικά και τυχαία χαρακτηριστικά.**

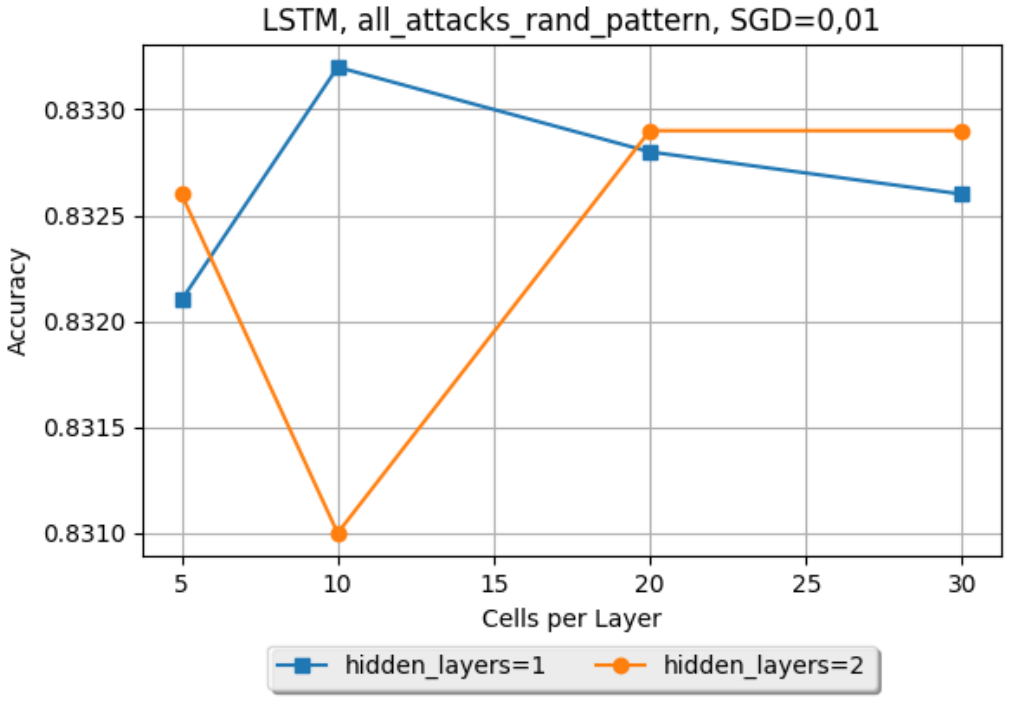


Σχήμα 26: Γραφική αναπαράσταση της ακρίβειας πάνω σε διάφορα δεδομένα ελέγχου (validation accuracy) ενός RNN νευρωνικού δικτύου με 2 κρυφά επίπεδα και 20 νευρώνες ανά κρυφό επίπεδο. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) περιείχε ροές από όλες τις επιμέρους κατηγορίες, οι οποίες τροφοδοτήθηκαν στο δίκτυο με σχήμα το οποίο διατηρούσε τη χρονική συσχέτιση των ροών μιας κατηγορίας αλλά συνδύαζε δεδομένα διαφορετικών κατηγοριών με τυχαίο τρόπο.

Όπως προαναφέραμε και διαπιστώνουμε εδώ **τα αποτελέσματα των RNN, όταν φροντίζουμε να διατηρηθεί η χρονική σειρά των δεδομένων και να ανακατευτούν οι επιμέρους κατηγορίες μεταξύ τους με ''αρκετά'' τυχαίο τρόπο κατά την εκπαίδευση, είναι περίπου ίδια με αυτά των MLP**. Το πρόβλημα του διαχωρισμού των ροών της επίθεσης tcp syn flood με ροές της επίθεσης port scanning παραμένει και βλάπτει σημαντικά την ακρίβεια των δικτύων. Περισσότερη μελέτη προς την κατεύθυνση αυτή, τον κατάλληλο δηλαδή και αντιπροσωπευτικό της πραγματικότητας συνδυασμό των δεδομένων κατά την εκπαίδευση πιθανόν να οδηγούσε σε κάπως καλύτερη ακρίβεια για δίκτυα RNN, όχι όμως αρκετά ικανοποιητική.

5.5 Χρήση δικτύων LSTM

Για λόγους πληρότητας της ανάλυσης εκτελούμε το ίδιο πείραμα με την ενότητα 5.3, κάνοντας όμως χρήση δικτύων τύπου LSTM, βλέπε ενότητα 2.2.11. Τα δεδομένα εκπαίδευσης που χρησιμοποιούμε περιέχουν ροές από όλες τις κατηγορίες επιθέσεων και καλόβουλη κίνηση και φροντίζουμε να διατηρηθεί η χρονική σειρά των δεδομένων και να ανακατευτούν οι επιμέρους κατηγορίες ροών μεταξύ τους με ‘’αρκετά’’ τυχαίο τρόπο κατά την εκπαίδευση, όπως στην ενότητα 5.4.



Σχήμα 27: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός LSTM αναδρασιακού νευρωνικού δικτύου, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε ροές από όλες τις επιμέρους κατηγορίες, οι οποίες τροφοδοτήθηκαν στο δίκτυο με σχήμα το οποίο διατηρούσε τη χρονική συσχέτιση των ροών μιας κατηγορίας αλλά συνδύαζε δεδομένα διαφορετικών κατηγοριών με τυχαίο τρόπο.

Διαπιστώνουμε ότι τα δίκτυα LSTM που εκπαιδεύσαμε, έχουν όλα **περίπου ίδια ακρίβεια όπως τα RNN δίκτυα και το πρόβλημα δεν έχει λυθεί**. Αυτό ήταν αναμενόμενο καθώς τα LSTM είναι δίκτυα RNN με την επιπλέον δυνατότητα να ανακαλύπτουν και να εκφράζουν πιο μακροπρόθεσμες χρονικές εξαρτήσεις. Όμως οι μακροπρόθεσμες χρονικές εξαρτήσεις των δεδομένων μας που μπορούν να εκφράσουν τα νευρωνικά δίκτυα τύπου LSTM δεν είναι σε θέση να επιτρέψουν τον καλύτερο διαχωρισμό των πέντε κατηγοριών που εξετάζουμε.

Ένα άλλο αξιοσημείωτο συμπέρασμα είναι ότι **τα δίκτυα της μορφής LSTM απαιτούσαν περισσότερο χρόνο εκπαίδευσης από τα δίκτυα RNN** καθώς η εκπαίδευση αυτών περιλαμβάνει αρκετά μεγαλύτερο πλήθος μαθηματικών πράξεων. Αυτός ήταν και ο λόγος που δεν εκπαιδεύσαμε LSTM δίκτυα με περισσότερα κρυφά επίπεδα καθώς η εκπαίδευση

ήταν πολύ χρονοβόρα και μεγαλύτερες σε βάθος αρχιτεκτονικές δεν ήταν σε θέση να συγκλίνουν.

5.6 Διαφορετική προσέγγιση του προβλήματος - Προσθήκη επιπλέον αθροιστικών πεδίων στα χαρακτηριστικά που τροφοδοτούνται στο νευρωνικό δίκτυο.

Αντιλαμβανόμαστε ότι το πρόβλημα μπορεί να επιλυθεί με την προσθήκη της ‘’ιστορίας’’ κάθε επίθεσης. Όμως τα δίκτυα RNN και LSTM που προσπαθούν να βρουν αυτή την ιστορία μέσω της ανάδρασης δεν μας οδήγησαν σε ικανοποιητικά αποτελέσματα. Μια διαφορετική πρόταση είναι να εισάγουμε εμείς αυτή την ‘’ιστορία’’ με την προσθήκη στα χαρακτηριστικά που τροφοδοτούνται στο νευρωνικό δίκτυο δύο επιπλέων αθροιστικών πεδίων.

Πιο συγκεκριμένα επεκτείνουμε το διάνυσμα εισόδου των νευρωνικών δικτύων, το οποίο αντιστοιχούσε μέχρι στιγμής σε χαρακτηριστικά μόνο μίας ροής IP, με ένα πεδίο στο οποίο περιέχεται ο συνολικός αριθμός των πακέτων που έχουν ληφθεί, σε ένα συγκεκριμένο χρονικό παράθυρο, από τη διεύθυνση αποστολής της ροής IP, προς την ίδια διεύθυνση προορισμού και την ίδια θύρα προορισμού του στρώματος μεταφοράς. Πρόκειται δηλαδή για πεδία που περιλαμβάνουν το άθροισμα του πλήθους των πακέτων που ανήκουν στην ίδια ή σε διαφορετικές ροές, οι οποίες έχουν κοινά τα παραπάνω χαρακτηριστικά, αλλά διαφέρουν στη θύρα προέλευσης του στρώματος μεταφοράς. Έτσι μπορούμε να ξεχωρίσουμε ροές που ανήκουν στην επίθεση tcp syn flood από ροές της επίθεσης port scanning καθώς διαφέρουν σημαντικά σε αυτό το νέο πεδίο. Βλέπε ενότητα 5.3 όπου περιγράφεται αναλυτικά το πρόβλημα σύγκλισης των δικτύων.

Επίσης δημιουργούμε ένα ακόμα πεδίο στο οποίο περιέχεται ο συνολικός αριθμός των πακέτων που έχουν σταλθεί ως απάντηση και ανήκουν σε αυτές τις ροές διπλής κατεύθυνσης των οποίων αθροίσαμε τα εισερχόμενα πακέτα στο πρώτο αθροιστικό πεδίο που αναφέρθηκε.

Χρονικό Παράθυρο που εκτελούμε το aggregation:

Τα νέα αθροιστικά πεδία που εισάγουμε δημιουργούνται από επεξεργασία των ροών όπως αυτές προκύπτουν στην έξοδο του εργαλείου nProbe. Στην υλοποίηση μας θέσαμε ως χρονικό παράθυρο για τη συλλογή δεδομένων τα 5 λεπτά. Δηλαδή αφού συλλέξουμε δεδομένα στην έξοδο του nProbe για διάρκεια 5 λεπτών, επεξεργαζόμαστε το αρχείο ώστε μαζί με τα χαρακτηριστικά κάθε ροής να τροφοδοτηθούν και τα δύο νέα αθροιστικά πεδία κάθε ροής στα νευρωνικά δίκτυα. Η άθροιση για το νέο πεδίο κάθε ροής γίνεται επί του συνολικού πλήθους ροών που ανήκουν στο ίδιο χρονικό παράθυρο 5 λεπτών.

Ωστόσο αυτή η τακτική επίλυσης του προβλήματος εμπεριέχει ένα σημαντικό μειονέκτημα. Πλέον είμαστε αναγκασμένοι να καθυστερούμε την τροφοδότηση των ροών στο νευρωνικό δίκτυο μέχρι να συλλέξουμε δεδομένα διάρκειας ίδιας με αυτής του χρονικού παραθύρου ώστε να μπορούμε να δημιουργήσουμε το νέο αθροιστικό πεδίο για κάθε ροή. Τροφοδοτούμε δηλαδή τις ροές στο νευρωνικό δίκτυο για αξιολόγηση ανά 5 λεπτά και όχι κατευθείαν όταν αυτές δημιουργούνται στην έξοδο του nProbe. Έτσι το εργαλείο μας δεν είναι σε θέση να αξιολογεί ροές άμεσα όπως αυτές καταγράφονται.

Υλοποίηση του aggregation:

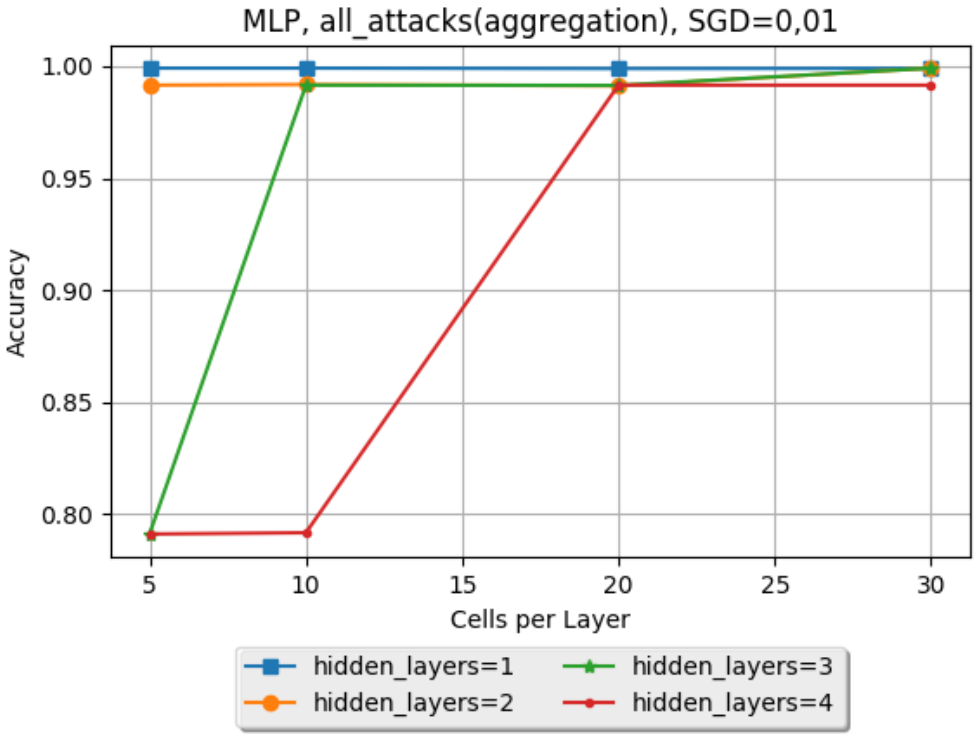
Τα νέα αθροιστικά πεδία δημιουργούνται από κώδικα python την ίδια στιγμή όπου προσαρμόζουμε τα δεδομένα του nProbe για τροφοδότηση στο νευρωνικό δίκτυο. Ο τρόπος άθροισης γίνεται με τη χρήση λεξικού(dictionary) που ανανεώνεται καθώς διαβάζονται οι ροές του εκάστοτε χρονικού παραθύρου πέντε λεπτών και το οποίο αποθηκεύει πληροφορίες για την εύκολη και άμεση άθροιση του πλήθους πακέτων διαφορετικών ροών. Βλέπε το αρχείο **data_creator.py** για περαιτέρω κατανόηση της διαδικασίας άθροισης.

Εκτέλεση του πειράματος:

Εκτελούμε το ίδιο πείραμα με την ενότητα 5.1. Οπότε θέτουμε τις ίδιες παραμέτρους εκπαίδευσης και κάνουμε χρήση των ίδιων δεδομένων εκπαίδευσης. Όμως **οι ροές IP έχουν επεκταθεί με την προσθήκη των δύο νέων αθροιστικών πεδίων που περιγράψαμε**. Τα αποτελέσματα που προέκυψαν από τα νευρωνικά που εκπαιδεύσαμε παρουσιάζονται στον πίνακα και στην ακόλουθη γραφική παράσταση.

Hidden Layers	Cells Per Layer	Validation Accuracy
1	5	0.9991
1	10	0.9991
1	20	0.9989
1	30	0.999
2	5	0.9915
2	10	0.9918
2	20	0.9912
2	30	0.9991
3	5	0.7914
3	10	0.9915
3	20	0.9914
3	30	0.9991
4	5	0.7912
4	10	0.7918
4	20	0.9915
4	30	0.9915

Πίνακας 6: Αποτελέσματα εκπαίδευσης διαφορετικών νευρωνικών δικτύων MLP, αλλάζοντας το πλήθος των νευρώνων που περιέχει το δίκτυο σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples και από τις 5 κατηγορίες τα οποία περιλάμβαναν και τα δύο νέα αθροιστικά πεδία.



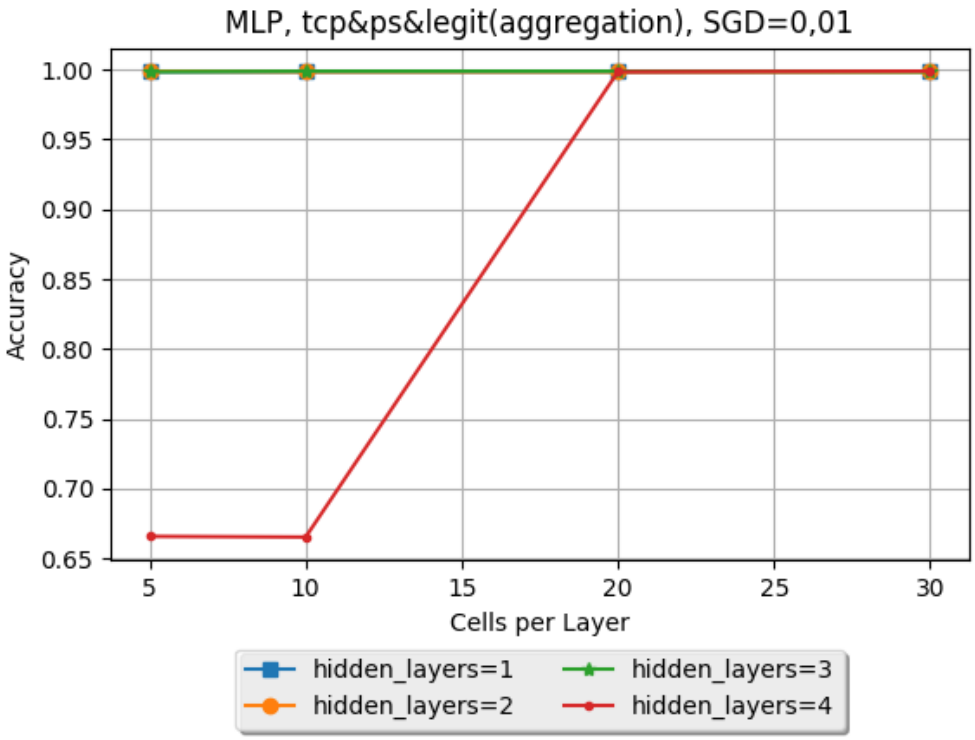
Σχήμα 28: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples και από τις 5 κατηγορίες τα οποία περιλάμβαναν και τα δύο νέα αθροιστικά πεδία.

Παρατηρούμε ότι σχεδόν όλα τα δίκτυα που εκπαιδεύσαμε ήταν σε θέση να επιλύσουν το πρόβλημα και οι 5 κατηγορίες διαχωρίστηκαν ικανοποιητικά. Η **μέγιστη ακρίβεια (validation accuracy)** που πετύχαμε είναι **0.9991** δηλαδή **99,91%.** Όπως φαίνεται από τη γραφική παράσταση **MLP με περισσότερα κρυφά επίπεδα (τρία ή τέσσερα) αλλά λίγους νευρώνες σε κάθε επίπεδο (πέντε ή δέκα) εξακολουθούν να δυσκολεύονται να διαχωρίσουν επαρκώς τις 5 κατηγορίες.**

Για να εξακριβώσουμε ότι τα νέα αθροιστικά πεδία που τροφοδοτούμε στα νευρωνικά δίκτυα MLP επιλύουν το πρόβλημα διαχωρισμού των επιθέσεων tcp syn flood και port scanning εκτελούμε το παραπάνω πείραμα εκπαιδεύοντας όμως τα MLP με δεδομένα μόνο από αυτές τις δύο επιθέσεις και καλόβουλη κίνηση. Έτσι προκύπτουν τα ακόλουθα αποτελέσματα.

Hidden Layers	Cells Per Layer	Validation Accuracy
1	5	0.9985
1	10	0.9985
1	20	0.9985
1	30	0.999
2	5	0.999
2	10	0.9985
2	20	0.9985
2	30	0.9985
3	5	0.9985
3	10	0.999
3	20	0.999
3	30	0.9985
4	5	0.665666667
4	10	0.665166667
4	20	0.9985
4	30	0.999

Πίνακας 7: Αποτελέσματα εκπαίδευσης διαφορετικών νευρωνικών δικτύων MLP, αλλάζοντας το πλήθος των νευρώνων που περιέχει το δίκτυο σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples από τις επιθέσεις tcp syn flood, port scanning και την καλόβουλη κίνηση τα οποία περιλάμβαναν και τα δύο νέα αθροιστικά πεδία.

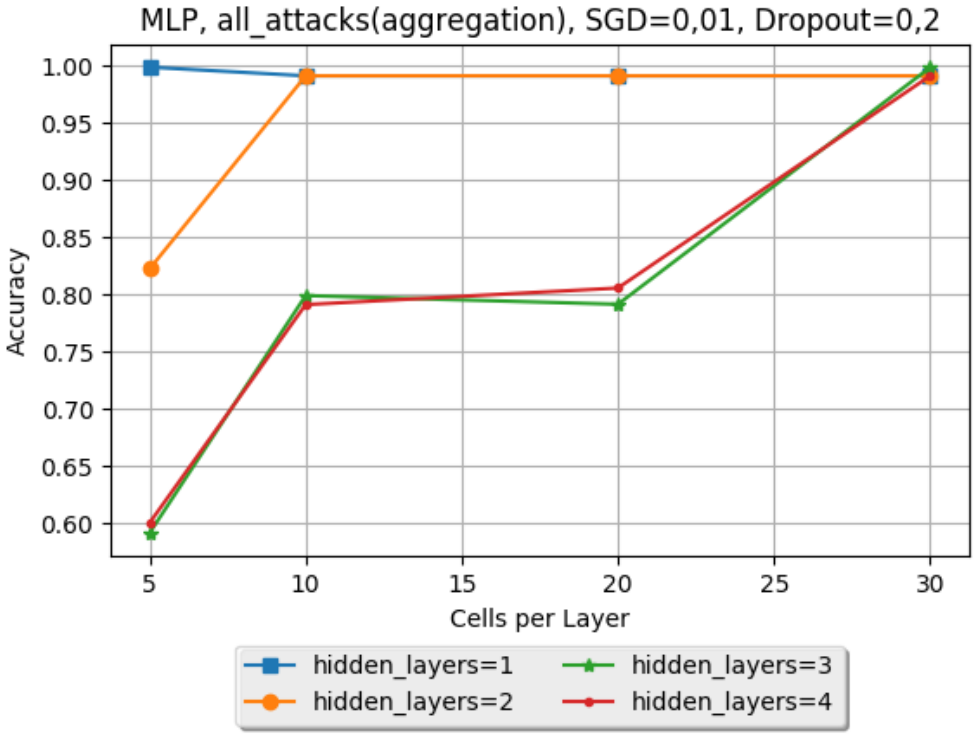


Σχήμα 29: Γραφική παράσταση της ακρίβειας πάνω στα δεδομένα ελέγχου (validation accuracy) ενός νευρωνικού δικτύου MLP, σε συνάρτηση με το πλήθος των νευρώνων που περιέχει σε κάθε κρυφό επίπεδο και το πλήθος των κρυφών επιπέδων. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples από τις επιθέσεις tcp syn flood, port scanning και την καλόβουλη κίνηση τα οποία περιλάμβαναν και τα δύο νέα αθροιστικά πεδία.

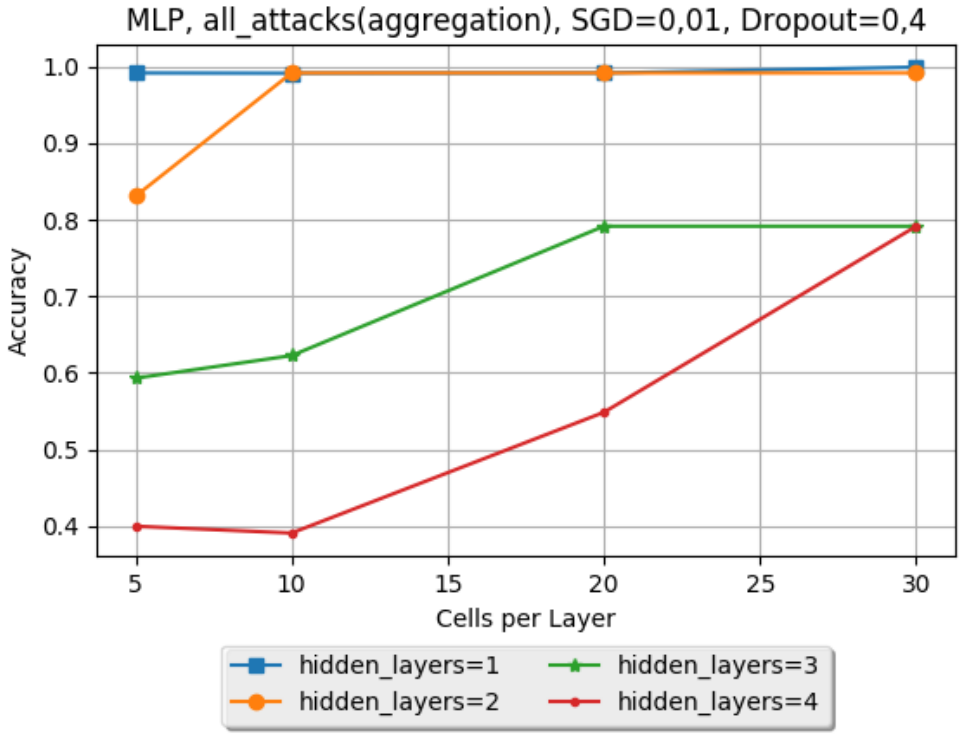
Πράγματι, όπως ήταν αναμενόμενο, σχεδόν όλα τα δίκτυα εκτός από αυτά με 4 κρυφά επίπεδα ήταν σε θέση να επιλύσουν το πρόβλημα με εξαιρετικά καλή ακρίβεια (validation accuracy) και επιβεβαιώνεται ο ισχυρισμός μας. Επομένως **επιλέγουμε δίκτυο MLP με δεδομένα εκπαίδευσης που περιέχουν επιπλέον τα αθροιστικά πεδία που περιγράψαμε για τη βέλτιστη λύση του προβλήματος** και στη συνέχεια προσπαθούμε να ανακαλύψουμε τα συγκεκριμένα χαρακτηριστικά του MLP που αποτελεί τη βέλτιστη λύση.

5.7 Πειραματισμός με διαφορετικούς αλγόριθμους εκπαίδευσης και χρήση Dropout

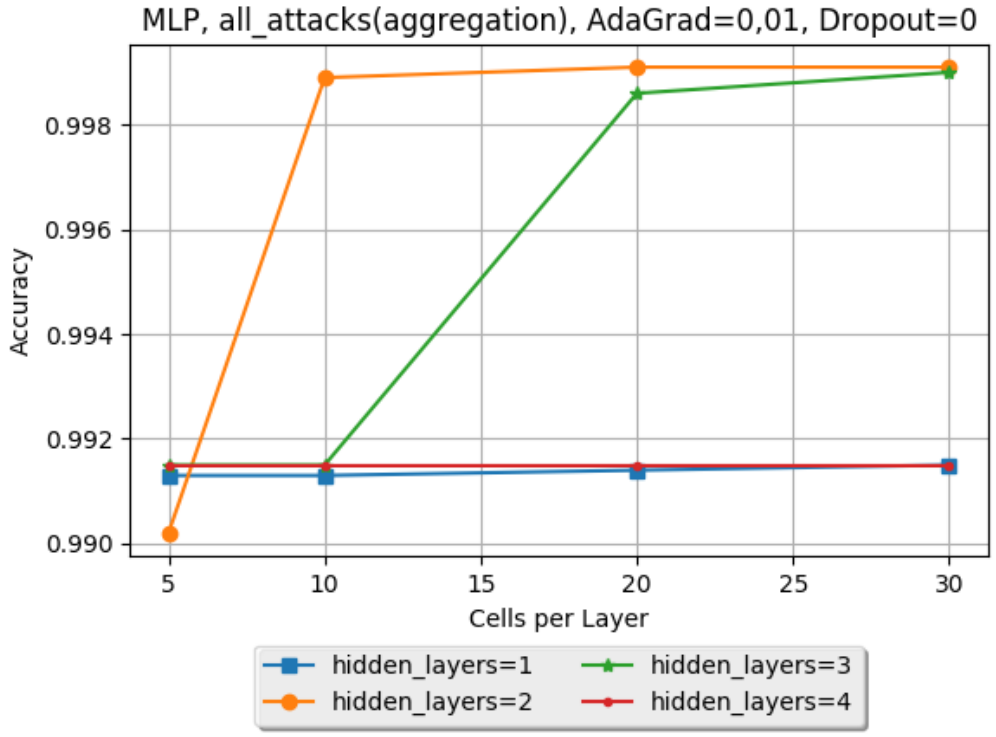
Έχοντας προσδιορίσει μια καλή λύση που διαχωρίζει ικανοποιητικά τις 5 κατηγορίες (MLP νευρωνικό δίκτυο κατά την εκπαίδευση του οποίου τα δεδομένα εισόδου περιέχουν και αθροιστικά πεδία) πειραματιζόμαστε με τρεις διαφορετικές μεθόδους εκπαίδευσης, **την SGD, την AdaGrad και την RMSProp** καθώς και τη **χρήση dropout σε κάθε κρυφό επίπεδο**. Εκτελούμε πολλές φορές το πείραμα της ενότητας 5.6 αλλάζοντας κάθε φορά τη μέθοδο εκπαίδευσης και το dropout μεταξύ των επιπέδων. Έτσι προκύπτουν οι ακόλουθες γραφικές παραστάσεις.



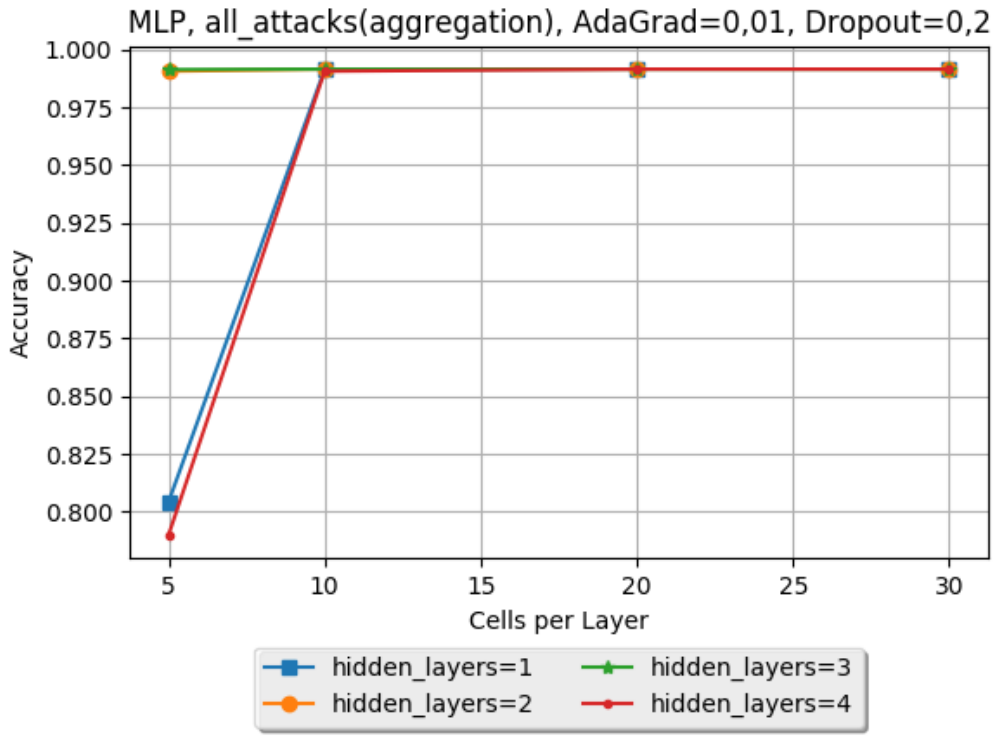
Σχήμα 30: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο SGD με ρυθμό μάθησης 0,01 και Dropout 0,2.



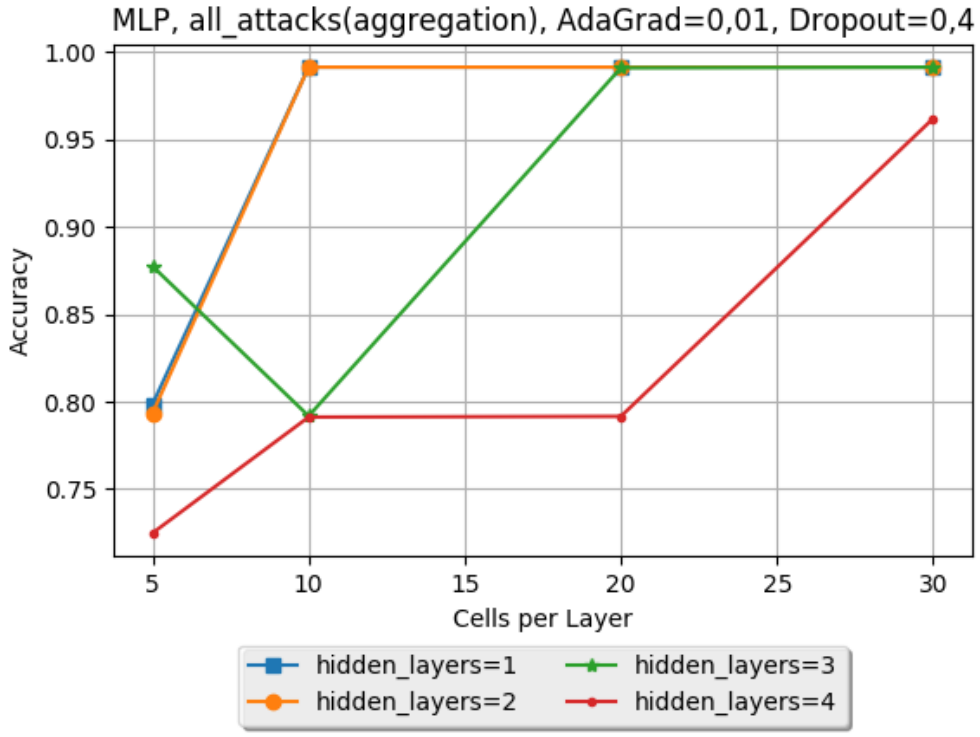
Σχήμα 31: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο SGD με ρυθμό μάθησης 0,01 και Dropout 0,4.



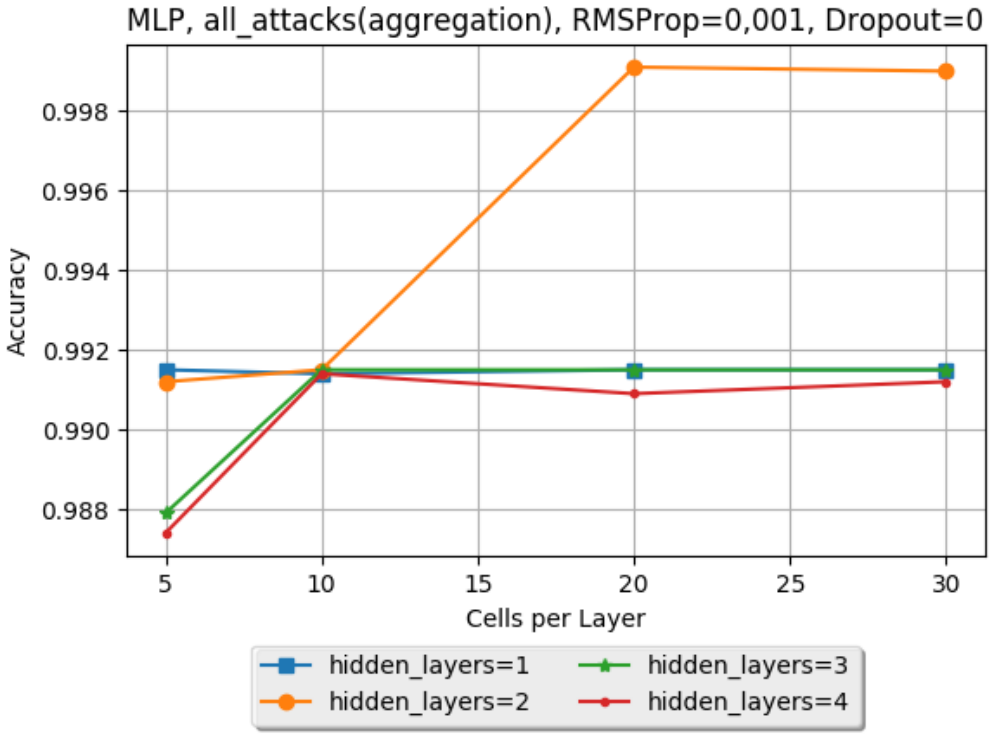
Σχήμα 32: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο AdaGrad με ρυθμό μάθησης 0,01 και Dropout 0.



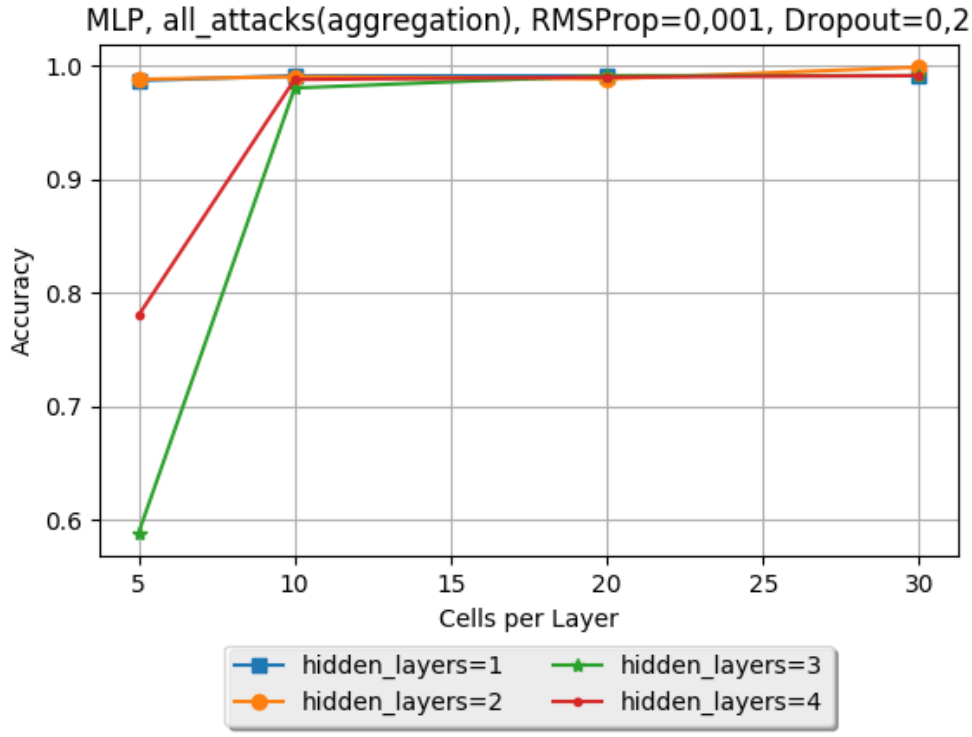
Σχήμα 33: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο AdaGrad με ρυθμό μάθησης 0,01 και Dropout 0,2.



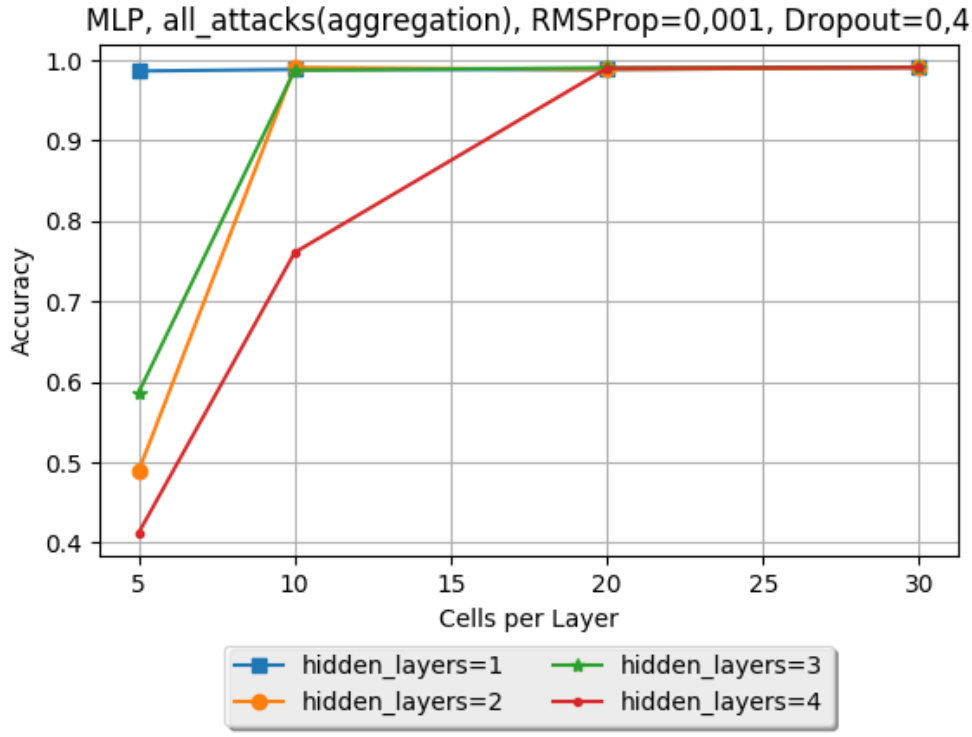
Σχήμα 34: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο AdaGrad με ρυθμό μάθησης 0,01 και Dropout 0,4.



Σχήμα 35: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο RMSProp με ρυθμό μάθησης 0,001 και Dropout 0.



Σχήμα 36: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο RMSProp με ρυθμό μάθησης 0,001 και Dropout 0,2.



Σχήμα 37: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο RMSProp με ρυθμό μάθησης 0,001 και Dropout 0,4.

Συμπεράσματα:

- Η **μέγιστη ακρίβεια (validation accuracy) που πετύχαμε ήταν 0,9991 ή 99,91%** και την πέτυχαν αρκετά δίκτυα.
- **Βαθύτερες αρχιτεκτονικές δικτύων MLP πετύχαιναν μικρότερη ακρίβεια (validation accuracy), ειδικότερα για μικρό πλήθος νευρώνων ανά επίπεδο (5 ή 10 νευρώνες)**. **Αύξηση του βάθους απαιτούσε και αύξηση των νευρώνων κάθε επιπέδου** για να έχουμε υψηλές ακρίβειες. Πιο συγκεκριμένα κανένα δίκτυο με 3 ή 4 κρυφά επίπεδα δεν κατάφερε να πετύχει τη μέγιστη ακρίβεια 99,91% εκτός ενός δικτύου εκπαιδευμένο με τον αλγόριθμο SGD, με 3 κρυφά επίπεδα, 30 νευρώνες ανά κρυφό επίπεδο και μηδενικό Dropout. **Οπότε προτείνουμε τη χρήση δικτύου με 1 ή 2 κρυφά επίπεδα ως βέλτιστη λύση.**
- Ο αλγόριθμος εκπαίδευσης **AdaGrad ήταν αυτός με τη μεγαλύτερη σταθερότητα όπου είχαμε τη μικρότερη απόκλιση μεταξύ νευρωνικών δικτύων με τη βέλτιστη και τη χειρότερη ακρίβεια**, για όλες τις τιμές του Dropout. Αντίθετα **ο αλγόριθμος SGD**, καθώς είναι και ο πιο βασικός αλγόριθμος με τις λιγότερο προχωρημένες τεχνικές, βλέπε ενότητα 2.2.7, **αποδείχθηκε ο χειρότερος όσον αφορά αυτό το κριτήριο ειδικότερα για δίκτυα με περισσότερα κρυφά επίπεδα (3 ή 4).**
- **Η χρήση Dropout φαίνεται να βλάπτει τα δίκτυα MLP, ειδικότερα όταν έχουμε Dropout=0,4.** Μια σημαντική παρατήρηση είναι ότι **το Dropout επηρεάζει περισσότερο δίκτυα τα οποία έχουν μικρότερο πλήθος νευρώνων ανά επίπεδο**. Μάλιστα αυτό **το φαινόμενο εντείνεται για MLP με πολλά κρυφά επίπεδα** όπου έχουμε εφαρμογή Dropout σε κάθε κρυφό επίπεδο. Αυτό είναι πολύ **φυσικό καθώς σε δίκτυα με λίγους νευρώνες ανά επίπεδο η σημασία κάθε νευρώνα είναι μεγαλύτερη οπότε η απουσία του γίνεται περισσότερο αισθητή**. Επίσης πολλά κρυφά επίπεδα σημαίνει ότι έχουμε απουσία περισσότερων νευρώνων αφού εφαρμόζουμε την τεχνική Dropout σε κάθε κρυφό επίπεδο. Οπότε **στη βέλτιστη λύση για το πρόβλημα κατηγοριοποίησης της δικτυακής κίνησης δεν προτείνουμε τη χρήση Dropout** ή τουλάχιστον μπορούμε να κάνουμε χρήση Dropout=0,2 όπου δεν έχει σημαντική επίπτωση στην ακρίβεια αν θέλουμε να είμαστε σίγουροι ότι δεν έχουμε overfitting.

5.8 Βέλτιστο Νευρωνικό Δίκτυο – Πίνακας Σύγχυσης

Αν και έχουμε περισσότερα από ένα βέλτιστα δίκτυα MLP τα οποία πέτυχαν τη μέγιστη ακρίβεια και πληρούν τις προαναφερθείσες βέλτιστες επιλογές θα προτείνουμε ως βέλτιστο δίκτυο MLP, αυτό με 2 κρυφά επίπεδα, 20 νευρώνες ανά κρυφό επίπεδο, χωρίς τη χρήση Dropout και εκπαιδευμένο με τον αλγόριθμο AdaGrad. Τα δεδομένα που τροφοδοτούνται εκτός από τις πληροφορίες της εκάστοτε ροής IP περιέχουν και τα δύο αθροιστικά πεδία που προαναφέραμε.

Η **ακρίβεια (validation accuracy)** ενός τέτοιου δικτύου σε dataset με 10.000 δείγματα όπου περιέχει ισάριθμο πλήθος ροών-δειγμάτων από όλες τις κατηγορίες είναι **0,9991 ή 99,91%** ενώ η ακρίβεια (validation accuracy) στις πέντε επιμέρους κατηγορίες ταξινόμησης, όπως προέκυψε από ελέγχους σε datasets που περιέχουν 10.000 δείγματα μόνο μίας κατηγορίας το κάθε ένα, φαίνεται στον ακόλουθο πίνακα σύγχυσης.

		Predicted Category				
		ICMP Flood	***TCP SYN Flood***	***UDP Flood***	***Port Scanning***	***Legitimate***
Actual Category	***ICMP Flood***	**0.9962**	0	0	0	0.0038
	TCP SYN Flood	0	**1**	0	0	0
	UDP Flood	0	0	**1**	0	0
	Port Scanning	0	0	0	**0.9999**	0.0001
	Legitimate	0.0004	0.0001	0.0001	0.0041	**0.9953**

Πίνακας 8: Πίνακας Σύγχυσης (Confusion Matrix) του βέλτιστου δικτύου MLP που προτείνεται.

5.9 Χρόνος εκπαίδευσης και ταχύτητα σύγκλισης

Στο σημείο αυτό σχολιάζουμε το χρόνο της εκπαίδευσης και την ταχύτητα σύγκλισης ενός δικτύου με βάση τις παρατηρήσεις μας από την εκπαίδευση πολλών διαφορετικών νευρωνικών δικτύων. Τα περισσότερα δίκτυα απαιτούσαν εκπαίδευση λίγων λεπτών. Ωστόσο παρατηρήσαμε τα ακόλουθα σε γενικές γραμμές:

- Τα δίκτυα **MLP απαιτούν αρκετά λιγότερο χρόνο εκπαίδευσης από ότι τα δίκτυα με ανάδραση**. Επίσης τα δίκτυα **RNN απαιτούν περισσότερο χρόνο εκπαίδευσης από τα MLP αλλά αρκετά λιγότερο χρόνο από τα LSTM.** Αυτό είναι φυσικό καθώς πιο πολύπλοκες αρχιτεκτονικές απαιτούν περισσότερες πράξεις κατά τη διαδικασία της εκπαίδευσης και επομένως περισσότερο χρόνο.

- **Δίκτυα με περισσότερα κρυφά επίπεδα και περισσότερους νευρώνες ανά κρυφό επίπεδο απαιτούν περισσότερο χρόνο εκπαίδευσης από ότι ρηχότερα νευρωνικά δίκτυα με λιγότερους νευρώνες**. **Το βάθος** ενός νευρωνικού δικτύου είναι παράμετρος που **αυξάνει σημαντικά το χρόνο εκπαίδευσης** καθώς υπολογισμός των μερικών παραγώγων των βαρών και η διάδοσή τους προς τα πίσω επίπεδα κατά τον αλγόριθμο backpropagation περιλαμβάνει περισσότερες πράξεις για πιο βαθιές αρχιτεκτονικές δικτύων.

- Ο ρυθμός μάθησης επιλέχθηκε τέτοιος σε κάθε αλγόριθμο εκπαίδευσης εκ των SGD, AdaGrad και RMSProp ώστε να συγκλίνει ικανοποιητικά εντός **10 εποχών** το νευρωνικό δίκτυο. **Αλλαγή του ρυθμού μάθησης σημαίνει και αλλαγή του πλήθους των εποχών που απαιτούνται για τη σύγκλιση των νευρωνικών δικτύων. Αύξηση ή μείωση του πλήθους των εποχών συνεπάγεται και αύξηση ή μείωση του χρόνου εκπαίδευσης αντίστοιχα.** Οπότε επιλέγουμε ρυθμό μάθησης ο οποίος μας επιτρέπει τη σύγκλιση των νευρωνικών δικτύων σε ‘’λογικό’’ πλήθος εποχών.

- Όσον αφορά το batch size το θέσαμε ίσο με τη μονάδα. Μεγαλύτερα batch size οδηγούν σε γρηγορότερη εκπαίδευση καθώς τα δεδομένα παρουσιάζονται ομαδοποιημένα στο νευρωνικό δίκτυο κατά την εκπαίδευση όμως απαιτούν μεγαλύτερο πλήθος δειγμάτων εκπαίδευσης. Καθώς πετυχαίναμε ικανοποιητικά αποτελέσματα με μέγεθος batch ίσο με τη μονάδα δεν πειραματιστήκαμε περισσότερο με αυτή την παράμετρο.

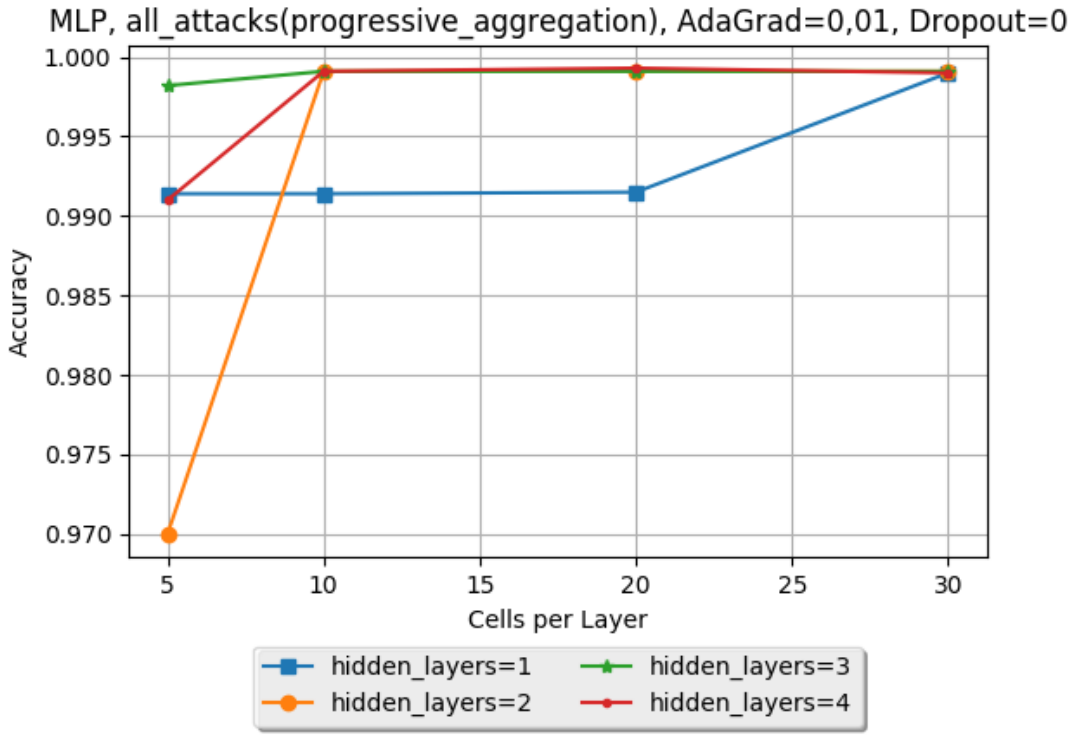
- Τέλος η χρήση της τεχνικής PCA κατά την επεξεργασία των δεδομένων εκπαίδευσης, σύμφωνα με τις δοκιμές μας πράγματι μείωνε σημαντικά το χρόνο εκπαίδευσης των νευρωνικών δικτύων καθώς μείωνε στο μισό περίπου το πλήθος των χαρακτηριστικών των δειγμάτων που τροφοδοτούνται στο νευρωνικό δίκτυο. Ωστόσο κάνοντας PCA ανάλυση στα δεδομένα δικτυακής κίνησης που θέλουμε να αξιολογήσουμε όταν το δίκτυο έχει κατασκευαστεί και χρησιμοποιείται για ταξινόμηση σε ένα δίκτυο, κατά πάσα πιθανότητα θα δώσει δεδομένα με διαφορετικά χαρακτηριστικά εισόδου που δε θα μπορούν να αξιολογηθούν από το νευρωνικό μας δίκτυο, γεγονός που την καθιστά μη εφαρμόσιμη στην περίπτωσή μας.

5.10 Προοδευτική άθροιση ροών

Έχοντας παρουσιάσει νευρωνικά δίκτυα που καταφέρνουν να επιλύσουν ικανοποιητικά το πρόβλημα επιθυμούμε να πειραματιστούμε περαιτέρω με τα νέα αθροιστικά πεδία που προσθέσαμε σε κάθε ροή και τροφοδοτούμε στο Νευρωνικό δίκτυο. Όπως αναφέραμε στην ενότητα 5.6 η λύση με την προσθήκη αθροιστικών πεδίων σημαίνει ότι είμαστε αναγκασμένοι να καθυστερούμε την τροφοδότηση των ροών στο νευρωνικό δίκτυο για αξιολόγηση, μέχρι να συλλέξουμε δεδομένα διάρκειας ίδιας με αυτής του χρονικού παραθύρου που ορίσαμε για την συλλογή ροών ώστε να μπορούμε να δημιουργήσουμε τα νέα αθροιστικά πεδία από τις συλλεγμένες ροές. Τροφοδοτούμε δηλαδή τις ροές στο νευρωνικό δίκτυο για αξιολόγηση ανά 5 λεπτά και όχι κατευθείαν όταν αυτές δημιουργούνται στην έξοδο του nProbe. Έτσι το εργαλείο μας δεν είναι σε θέση να αξιολογεί ροές άμεσα όπως αυτές καταγράφονται.

Μια πρόταση για την αντιμετώπιση αυτού του προβλήματος είναι η υλοποίηση προοδευτικής άθροισης των ροών όπως αυτές παράγονται στην έξοδο του nProbe. Δηλαδή υπολογίζουμε και πάλι το άθροισμα του πλήθους των πακέτων που ανήκουν στην ίδια ή σε διαφορετικές ροές, οι οποίες έχουν κοινά χαρακτηριστικά όπως στην ενότητα 5.6 όμως **η άθροιση γίνεται μόνο επί των ήδη συλλεγμένων ροών και όχι αυτών που θα συλλέξουμε εντός 5 λεπτών. Έτσι εξαλείφουμε την ανάγκη να καθυστερούμε την τροφοδότηση των ροών στο νευρωνικό δίκτυο για αξιολόγηση, μέχρι να συλλέξουμε δεδομένα διάρκειας ίδιας με αυτής του χρονικού παραθύρου που ορίσαμε**. Οπότε αυτό σημαίνει ότι η πρώτη ροή που συλλέγουμε στα αθροιστικά πεδία με τα οποία την επεκτείνουμε θα έχει προσθέσει μόνο το δικό της πλήθος πακέτων. Όσο όμως συλλέγουμε περισσότερες ροές με κοινά χαρακτηριστικά οι νέες ροές θα επεκταθούν με μεγαλύτερα αθροιστικά πεδία που υποδεικνύουν τελικά την ύπαρξη ογκομετρικών επιθέσεων.

Στη συνέχεια εκτελούμε πείραμα αντίστοιχο με αυτό των προηγούμενων ενοτήτων για να ελέγξουμε την απόδοση του να τροφοδοτούμε τα δεδομένα τοιουτοτρόπως στο νευρωνικό δίκτυο.



Σχήμα 38: Γραφική παράσταση της ακρίβειας (validation accuracy) MLP δικτύων, εκπαιδευμένων με τον αλγόριθμο AdaGrad με ρυθμό μάθησης 0,01 και Dropout 0,2. Το dataset που χρησιμοποιήθηκε κατά την φάση εκπαίδευσης (train) και τη φάση ελέγχου (test) περιείχε samples από τις 5 κατηγορίες τα οποία περιλάμβαναν και τα δύο νέα προοδευτικά αθροιστικά πεδία.

Διαπιστώνουμε ότι η εκπαίδευση με δεδομένα στα οποία τα αθροιστικά πεδία προκύπτουν από άθροιση των ροών που ήδη διαθέτουμε **οδηγεί σε εξίσου καλά αποτελέσματα** και όλως περιέργως κάποια δίκτυα εμφάνισαν ακόμα καλύτερα αποτελέσματα. Η βέλτιστη ακρίβεια (validation accuracy) που πετύχαμε είναι **0,9993 ή 99,93%.** Η πιο πιθανή εξήγηση για το φαινόμενο αυτό είναι ότι καθώς κατά την άθροιση μόνο ροών που έχουμε ήδη συλλέξει και όχι ροών ήδη συλλεγμένων σε κάποιο χρονικό παράθυρο, τα αθροιστικά πεδία των ροών πέρασαν από πολλές ενδιάμεσες τιμές ανάλογα με το πόσες ροές είχαμε συλλεγμένες την κάθε χρονική στιγμή και αυτό οδήγησε σε εκπαίδευση όπου τα νευρωνικά δίκτυα βασιζόταν με διαφορετικό τρόπο στα νέα αθροιστικά πεδία και οι πέντε κατηγορίες ροών διαχωρίστηκαν με μεγαλύτερη ακρίβεια.

Στο σημείο αυτό **αναθεωρούμε την επιλογή μας για το βέλτιστο δίκτυο καθώς δίκτυο με 4 κρυφά επίπεδα, 20 νευρώνες ανά κρυφό επίπεδο, εκπαιδευμένο με τον αλγόριθμο AdaGrad με ρυθμό μάθησης 0,01 πέτυχε ακρίβεια (validation accuracy) 99,93%**. Ωστόσο **το δίκτυο που είχαμε προτείνει ως βέλτιστο στην ενότητα 5.8 πέτυχε και πάλι ακρίβεια (validation accuracy) 99,91% όπως και προηγουμένως, διαφορά που είναι αμελητέα**.

Ανάγκη για ύπαρξη παρελθοντικών ροών:

Τονίζουμε ότι το validation set στο πείραμα μας περιείχε 10.000 ροές οι οποίες όμως δεν ήταν οι πρώτες που είχαν συλλεχτεί. Έχουν προηγηθεί 40.000 ροές οι οποίες

χρησιμοποιήθηκαν στην εκπαίδευση (training set) και λαμβάνονται υπόψη στον υπολογισμό του ‘’προοδευτικού’’ αθροιστικού πεδίου. Δηλαδή ο έλεγχος γίνεται πάνω σε ροές όπου τα ‘’προοδευτικά’’ αθροιστικά πεδία έχουν ήδη προλάβει να συλλέξουν αποτελέσματα από παρελθοντικές ροές της κάθε κατηγορίας. Όταν αυτό δε συμβαίνει όπως για παράδειγμα **όταν ενεργοποιούμε το μηχανισμό για πρώτη φορά ή όταν έχουμε μια νέα επίθεση στο δίκτυο η ορθότητα των προβλέψεων του νευρωνικού δικτύου επηρεάζεται μέχρις ότου συλλεχθούν κάποιες ροές**. Ωστόσο **το βασικό μας μέλημα ο προτεινόμενος μηχανισμός να λειτουργεί πρακτικά σε πραγματικό χρόνο έχει ικανοποιηθεί**.

Ακολούθως παρουσιάζουμε τον πίνακα σύγχυσης του νέου προτεινόμενου νευρωνικού δικτύου πάνω σε δύο διαφορετικά datasets για να κάνουμε πιο εμφανή τον παραπάνω ισχυρισμό.

- Dataset με 10000 δείγματα (samples), ισομοιρασμένα στις πέντε κατηγορίες ροών, το οποίο περιλαμβάνει ροές με ‘’προοδευτικά’’ αθροιστικά πεδία χωρίς ύπαρξη παρελθοντικών ροών.

		Predicted Category				
		ICMP Flood	***TCP SYN Flood***	***UDP Flood***	***Port Scanning***	***Legitimate***
Actual Category	***ICMP Flood***	**0.9775**	0	0	0	0.0225
	TCP SYN Flood	0	**0.95**	0	0.05	0
	UDP Flood	0	0	**1**	0	0
	Port Scanning	0	0	0	**0.9995**	0.0005
	Legitimate	0	0	0	0.005	**0.995**

Πίνακας 9: Πίνακας Σύγχυσης (Confusion Matrix) του βέλτιστου δικτύου MLP που προτείνεται όπως προκύπτει από validation dataset με 10000 δείγματα (samples), χωρίς ύπαρξη παρελθοντικών ροών.

- Dataset με 10000 δείγματα (samples), ισομοιρασμένα στις πέντε κατηγορίες ροών, το οποίο περιλαμβάνει ροές όπου τα ‘’προοδευτικά’’ αθροιστικά πεδία έχουν ήδη συλλέξει αποτελέσματα από 40.000 παρελθοντικές ροές όπως και στο πρώτο πείραμα της ενότητας 5.9.

		Predicted Category				
Actual Category		***ICMP Flood***	***TCP SYN Flood***	***UDP Flood***	***Port Scanning***	***Legitimate***
	ICMP Flood	**0.9995**	0	0.0005	0	0
	TCP SYN Flood	0	**1**	0	0	0
	UDP Flood	0	0	**1**	0	0
	Port Scanning	0	0	0	**0.9995**	0.0005
	Legitimate	0	0	0	0.0025	**0.9975**

Πίνακας 10: Πίνακας Σύγχυσης (Confusion Matrix) του βέλτιστου δικτύου MLP που προτείνεται όπως προκύπτει από validation dataset με 10000 δείγματα (samples), στο οποίο τα ‘’προοδευτικά’’ αθροιστικά πεδία έχουν ήδη συλλέξει αποτελέσματα από 40.000 παρελθοντικές ροές.

Τελικό Συμπέρασμα:

Όπως καθίσταται εμφανές από τους δύο πίνακες **όλες οι κατηγορίες, εκτός από επιθέσεις port scanning, αναγνωρίζονται καλύτερα όταν έχουν καταγραφεί περισσότερες ροές οπότε το προοδευτικό αθροιστικό πεδίο έχει ‘’μεγαλώσει’’ περισσότερο**. Αυτό είναι φυσικό καθώς μια ογκομετρική επίθεση δε μπορεί να γίνει εμφανής από τις πρώτες ροές. Οπότε **η ακρίβεια ταξινόμησης (accuracy) της πρώτης παραδείγματος χάριν καταγεγραμμένης ροής μιας νέας επίθεσης εκφυλίζεται σε αυτή των δικτύων MLP χωρίς τη χρήση αθροιστικών πεδίων** όπως στις ενότητες 5.1 και 5.2 όμως **με τη συλλογή περισσότερων ροών τελικά η ακρίβεια ταξινόμησης (accuracy) με τη χρήση προοδευτικών αθροιστικών πεδίων γίνεται πολύ υψηλή**.

Οπότε μπορούμε να έχουμε **μηχανισμό ο οποίος λειτουργεί πρακτικά σε πραγματικό χρόνο όμως η ακρίβεια ταξινόμησης των πρώτων ροών μιας επίθεσης επηρεάζεται όπως παραπάνω.**

6 Μελλοντική Δουλειά και Επεκτάσεις

Ως μελλοντική δουλειά μπορεί να μελετηθεί το μέγεθος του χρονικού παραθύρου κατά το οποίο συλλέγουμε ροές Netflow, τις οποίες στη συνέχεια επεξεργαζόμαστε για να προκύψουν τα αθροιστικά πεδία με τα οποία επεκτείνουμε κάθε ροή και τροφοδοτούμε στο νευρωνικό δίκτυο μαζί με κάθε ροή. Στη δικιά μας υλοποίηση θέσαμε αυτό το χρονικό παράθυρο 5 λεπτά καθώς το κρίναμε επαρκές για την κατηγοριοποίηση των ροών. Περαιτέρω μελέτη σχετικά με το πώς το χρονικό παράθυρο επηρεάζει την ακρίβεια της κατηγοριοποίησης μπορεί να μειώσει αυτό το χρονικό παράθυρο. Αντίστοιχα μπορεί να μελετηθεί το μέγεθος του παρελθοντικού παραθύρου στο οποίο αρκεί να εξετάζεται η άθροιση ροών κατά τη δημιουργία προοδευτικών αθροιστικών πεδίων που προτείνουμε για τη διατήρηση του.

Μια άλλη πιθανή επέκταση της παρούσας διπλωματικής εργασίας είναι η αναζήτηση ενός τρόπου να συμπεριληφθεί στο διάνυσμα εισόδου των νευρωνικών δικτύων πληροφορία σχετικά με τις διευθύνσεις IP αποστολής και προορισμού της κάθε ροής ή συναθροίσεις πολλών διευθύνσεων IPv4 σε υποδίκτυα. Αυτά είναι πεδία που αν και χρησιμοποιήθηκαν κατά το διαχωρισμό των πακέτων σε ροές Netflow και τον υπολογισμό των αθροιστικών και των προοδευτικών αθροιστικών πεδίων δεν τροφοδοτήθηκαν άμεσα στο νευρωνικό δίκτυο, όπως εξηγούμε στην ενότητα 4.2. Αν και τις περισσότερες φορές οι IP διευθύνσεις αποστολής είναι spoofed, αυτά τα πεδία μιας ροής Netflow, όταν συσχετίζονται με τα αντίστοιχα πεδία άλλων ροών μπορούν να αποδώσουν χρήσιμες πληροφορίες, οι οποίες ίσως δώσουν και αυξημένες δυνατότητες στις RNN και τις LSTM αρχιτεκτονικές.

Τέλος ενδιαφέρον έχει η μελέτη της επανεκπαίδευσης του νευρωνικού δικτύου αφού αυτό τεθεί σε εφαρμογή σε κάποιο δίκτυο για αναγνώριση επιθέσεων. Έτσι θα μπορούσαμε για παράδειγμα να λάβουμε υπόψη μας κατά την επανεκπαίδευση πληροφορία από τα πιο πρόσφατα δεδομένα της δικτυακής κίνησης. Μια τέτοια μελέτη, εκτός των άλλων, προϋποθέτει και την εύρεση κατάλληλων χρονικών στιγμών για τη διαδικασία της επανεκπαίδευσης.

Βιβλιογραφία

[1] "The Zettabyte Era: Trends and Analysis" http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/vni-hyperconnectivity-wp.pdf

[2] W. Stallings, (2013). ''Cryptography and Network Security: Principles and Practice'', Prentice Hall, 2013.

[3] W. Stallings, L. Brown, (2014). "Computer Security: Principles and Practice", Pearson, 2014.

[4] ''Incapsula'', [Online]. Available: https://www.incapsula.com/ddos/ddos-attacks/

[5] ''Arbor Networks'', [Online]. Available: https://www.arbornetworks.com/research/what-is-ddos

[6] ''Cloudflare'', [Online]. Available: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/

[7] G. Fyodor Lyon, (2009). ''Nmap Network Scanning", [Online]. Available: https://nmap.org/book/man-port-scanning-techniques.html

[8] "Slideshare'', [Online]. Available: https://www.slideshare.net/welcometofacebook/m05-35513854

[9] B. Claise, "Cisco Systems NetFlow Services Export Version 9", [Online]. Available: https://www.ietf.org/rfc/rfc3954.txt

[10] "Wikipedia", [Online]. Available: https://en.wikipedia.org/wiki/NetFlow

[11] Y. LeCun, MA Ranzato, ''Deep Learning, Tutorial'', ICML, Atlanta, 2013-06-16. [Online]. Available: http://www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf

[12] K. Hornik, M. Stinchcombe, and H. White, (1989). "Multilayer feedforward networks are universal approximators", Neural Networks, Vol. 2, pp. 359-366, 1989.

[13] I. Goodfellow, Y. Bengio, and A. Courville. ''Deep Learning'', MIT Press, 2016.

[14] J. Duchi, E. Hazan, and Y. Singer, (2011). ''Adaptive Subgradient Methods for Online Learning and Stochastic Optimization'', Journal of Machine Learning Research 12 (2011) 2121-2159.

[15] D. Kingma, J. Ba, (2014). "Adam: A Method for Stochastic Optimization", arXiv preprint arXiv:1412.6980.

[16] Y. Bengio, P. Simard, and P. Frasconi, (1994). ''Learning long-term dependencies with gradient descent is difficult'', IEEE Transactions on Neural Networks, Vol. 5, No 2, March 1994.

[17] S. Hochreiter, and J. Schmidhuber, (1997). ''Long short-term memory'', Neural Computation, 9(8):1735–1780, 1997.

[18] A. Graves, (2012). “Supervised Sequence Labelling with Recurrent Neural Networks”, Studies in Computational Intelligence, Springer.

[19] A. Graves, (2013). “Generating sequences with recurrent neural networks”, Technical report, arXiv:1308.0850.

[20] “nProbe”, [Online]. Available: http://www.ntop.org/products/netflow/nprobe/

[21] “Scapy”, [Online]. Available: http://www.secdev.org/projects/scapy/

[22] “Tcpreplay”, [Online]. Available: http://tcpreplay.synfin.net/

[23] “Wireshark”, [Online]. Available: https://www.wireshark.org

[24] “Nmap”, [Online]. Available: https://nmap.org/

[25] “editcap”, [Online]. Available: https://www.wireshark.org/docs/man-pages/editcap.html

[26] “Tcpdump”, [Online]. Available: http://www.tcpdump.org

[27] “Keras”, [Online]. Available: https://keras.io