
# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

*ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ*

*ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ*
*ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ*

*ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ*


# *Κατηγοριοποίηση Δικτυακών Επιθέσεων με μεθόδους Μηχανικής Μάθησης*

**Διπλωματική Εργασία**


*Ορέστης Άλπος*

**Επιβλέπων Καθηγητής:** *Βασίλειος Μάγκλαρης, Καθηγητής Ε.Μ.Π.*


14 Ιουλιου 2017


# ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

*ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ*

*ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ ΗΛΕΚΤΡΟΝΙΚΗΣ*
*ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ*

*ΕΡΓΑΣΤΗΡΙΟ ΔΙΑΧΕΙΡΙΣΗΣ ΚΑΙ ΒΕΛΤΙΣΤΟΥ ΣΧΕΔΙΑΣΜΟΥ ΔΙΚΤΥΩΝ*


# *Κατηγοριοποίηση Δικτυακών Επιθέσεων με μεθόδους Μηχανικής Μάθησης*

**Διπλωματική Εργασία**


*Ορέστης Άλπος*

**Επιβλέπων Καθηγητής:** *Βασίλειος Μάγκλαρης*


**Εγκρίθηκε από την τριμελή επιτροπή:**

| | | |
|---|---|---|
| ..... | ..... | ..... |
| Μάγκλαρης Β. | Συκάς Ε. | Σταφυλοπάτης Α. |
| Καθηγητής Ε.Μ.Π. | Καθηγητής Ε.Μ.Π. | Καθηγητής Ε.Μ.Π. |

14 Ιουλιου 2017

.....

*Ορέστης Άλπος*
Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών ΕΜΠ

# Περίληψη

Τα τελευταία χρόνια παρατηρείται μία έξαρση τόσο στην έρευνα σχετικά με τα Νευρωνικά Δίκτυα και κυρίως τα Βαθιά Νευρωνικά Δίκτυα (Deep Neural Networks) όσο και στις εφαρμογές αυτών. Το πεδίο των εφαρμογών είναι πολύ ευρύ, περιλαμβάνει από επεξεργασία φυσικής γλώσσας μέχρι ανάλυση βίντεο και αναγνώριση μοτίβων σε αυτά.

Από την άλλη πλευρά, το θέμα της Δικτυακής Ασφάλειας εξακολουθεί να είναι εξαιρετικά επίκαιρο, καθώς δικτυακές επιθέσεις, όπως για παράδειγμα επιθέσεις Άρνησης Υπηρεσίας (Denial of Service, DoS) και επιθέσεις κρυπτογράφησης αρχείων (τύπου Ransomware) συμβαίνουν καθημερινά.

Στη συγκεκριμένη διπλωματική εργασία εξετάζεται η δυνατότητα χρήσης των Νευρωνικών Δικτύων στο πρόβλημα της κατηγοριοποίησης της δικτυακής κίνησης σε ομαλή, νόμιμη κίνηση και σε κίνηση που προέρχεται από κάποια κακόβουλη πηγή και αποτελεί μέρος επίθεσης. Εξετάζονται κυρίως Βαθιά Νευρωνικά Δίκτυα, δηλαδή Νευρωνικά Δίκτυα με ένα τουλάχιστον κρυφό επίπεδο, και δίνονται συμπεράσματα σχετικά με τη δομή του Νευρωνικού Δικτύου που εξυπηρετεί καλύτερα το υπό εξέταση πρόβλημα. Εξετάζονται τρεις τύποι Νευρωνικών Δικτύων, τα Δίκτυα Νευρώνων Πολλών Επιπέδων (Multi-Layer Perceptron, MLP), τα Αναδρασιακά Νευρωνικά Δίκτυα (Recurrent Neural Networks, RNN) και τα LSTM (Long Short-Term Memory).

Όσον αφορά στις επιθέσεις, δίνεται βαρύτητα στις επιθέσεις Άρνησης Υπηρεσίας, καθώς αναγνωρίζονται τρεις υποκατηγορίες αυτών. Συγκεκριμένα, εξετάζονται η Πλημμύρα UDP (UDP Flood), η Πλημμύρα ICMP (ICMP Flood) και η επίθεση με TCP SYN πακέτα (SYN Flood). Ακόμη, ασχολούμαστε με την επίθεση Port Scanning.
Τα αρχεία διαδικτυακής κίνησης (pcap καταγραφές) που χρησιμοποιήθηκαν προήλθαν είτε από πραγματικές επιθέσεις, όπως το γνωστό αρχείο επίθεσης από το 2007 της CAIDA[1], είτε κατασκευάστηκαν στα πλαίσια της διπλωματικής, χρησιμοποιώντας εργαλεία όπως το Scapy[2] και το nmap[3].

Φάνηκε ότι τα παραπάνω είδη επίθεσης μπορούν να αναγνωριστούν και να διακριθούν - τόσο σε σχέση με την ομαλή κίνηση όσο και το ένα από το άλλο - με πολύ καλή ακρίβεια. Φάνηκε επίσης ότι η αύξηση του βάθους των δικτύων σε περισσότερα από 3 κρυφά επίπεδα δεν προσφέρουν καμία επιπλέον βελτίωση. Έτσι, προτείνονται κάποια δίκτυα, ένα από κάθε είδος που αναφέρθηκε, που κρίθηκαν βέλτιστα.

**Λέξεις Κλειδιά**: Ανίχνευση Επιθέσεων, Επιθέσεις Άρνησης Υπηρεσίας, Νευρωνικά Δίκτυα, Βαθιά Νευρωνικά Δίκτυα, Ταξινόμηση Δικτυακής Κίνησης

# Abstract

Throughout the last years, we are witnessing a huge increase both in the research related to the Neural Networks - and especially the Deep Neural Networks - as well as in their applications. The field in which Deep Neural Networks are applied is huge, containing Natural Language Processing, Video Analysis and Pattern Recognition.

On the other hand, Network Security is always an important subject, since network attacks - such as Denial Of Service (DoS) or File Encryption (Ransomware) attacks - occur on a daily basis.

In this Diploma Thesis we are examining the usage of Neural Networks on the problem of Network Traffic Classification. We focus on Deep Neural Networks, which means Neural Networks with at least one hidden layer, and we make conclusion concerning the structure of the Networks. Three types of Neural Networks are considered, the Multi-Layer Perceptron (MLP), the Recurrent Neural Network (RNN) and the Long Short-Term Memory (LSTM) Networks.

As for the attacks, we focus on the Denial of Service Attacks and we feed our system with three types of it, the UDP Flood, the ICMP Flood and the TCP SYN Attack. Apart from that, we also examined the Port Scanning Attack.

The attack data we used either originated from real network traffic, such as the CAIDA DDoS Dataset from 2007[1], or were constructed for the purposes of the thesis, using tools like scapy[2] and nmap[3].

It was found out that the aforementioned attacks can be recognized and distinguished - not only from the legitimate traffic but also from each other - with very high precision. It was also shown that increasing the depth of a Neural Network to more than three hidden layers did not incur any improvement in our task. In the end, we suggest some networks, one from each of the aforementioned categories, which we concluded as the most appropriate for our task.



**Keywords**: Attack recognition, Dos, Denial of Service, Neural Networks, Deep Neural Networks, Network traffic classification

# Ευχαριστίες

Ένα μακρύ, όμορφο και ανταποδοτικό ταξίδι φτάνει για εμένα στο τέλος του.

Θέλω πρωτίστως να ευχαριστήσω την οικογένειά μου και ιδιαίτερα τους γονείς μου για όσα έχουν κάνει για εμένα όλα αυτά τα χρόνια και για τη συνεχή στήριξή τους, ακόμα και υπό τις πιο δύσκολες συνθήκες. Χωρίς αυτούς και την ανιδιοτελή τους προσφορά δε θα ήταν δυνατό να φτάσω έως εδώ.

Επίσης, ευχαριστώ όσους καθηγητές έδειξαν, όλα αυτά τα χρόνια, όρεξη και μεράκι για τους φοιτητές τους. Αυτοί αποτέλεσαν για εμένα πηγή γνώσης και έμπνευσης.

Τέλος, θα ήθελα να ευχαριστήσω τον καθηγητή μου κο Βασίλη Μάγκλαρη για την εμπιστοσύνη του και το χρόνο που μου αφιέρωσε, καθώς επίσης τον Αδάμ και τον Κώστα για την πολύτιμη βοήθειά τους.

Αν η σχολή μου δίδαξε ένα πράγμα, αυτό είναι ότι στα δύσκολα δεν πρέπει να το βάζει κανείς κάτω. Στο σημείο ακριβώς που κάτι μοιάζει ακατόρθωτο, εκεί είναι που πρέπει να δείξεις τη δύναμή σου και να συνεχίσεις. Γιατί ο δρόμος μερικές φορές είναι δύσβατος, ο προορισμός όμως σε περιμένει και είναι ανθηρός.

# Κατάλογος Σχημάτων

# Κατάλογος Πινάκων

# Περιεχόμενα

# 1 Εισαγωγή

## 1.1 Το Ερευνητικό Πρόβλημα

Το ερευνητικό θέμα της παρούσας Διπλωματικής Εργασίας είναι η δυνατότητα χρήσης των Νευρωνικών Δικτύων, και ιδιαίτερα των Βαθιών Νευρωνικών Δικτύων (Deep Neural Networks), δηλαδή των Νευρωνικών που έχουν ένα και περισσότερα κρυφά επίπεδα, στην ανίχνευση δικτυακών επιθέσεων και στη διάκριση της κίνησης σε καλόβουλη και κακόβουλη.
Το πρόβλημα εντάσσεται στην ευρύτερη κατηγορία της Ασφάλειας Δικτύων, που εξακολουθεί να γίνεται όλο και πιο σημαντικό και επίκαιρο. Οι επιθέσεις Αρνησης Υπηρεσίας (Denial Of Service, DoS) εξακολουθούν να αποτελούν ένα από τα πλέον συχνά και καταστροφικά είδη επίθεσης. Μεγάλες εταιρίες, όπως η Incapsula[4], η Cloudflare[5] και η Arbor[6] έχουν αναπτύξει συστήματα ανίχνευσης και αντιμετώπισης (mitigation) επιθέσεων άρνησης υπηρεσίας.

## 1.2 Σκοπός της Εργασίας

Για την ανίχνευση και την αντιμετώπιση των επιθέσεων Άρνησης Υπηρεσίας έχουν προταθεί πολλές μέθοδοι. Στην παρούσα διπλωματική εργασία ερευνάται η δυνατότητα των Νευρωνικών Δικτύων (Neural Networks) να κατηγοριοποιήσουν την εισερχόμενη σε ένα σύστημα (υποδίκτυο ή τερματικό σταθμό) δικτυακή κίνηση σε Ομαλή (ή καλόβουλη, legitimate, benign) κίνηση και σε κίνηση που αποτελεί μέρος επίθεσης (malificent).

Απαραίτητη προϋπόθεση για ένα τέτοιο σύστημα κατηγοριοποίησης (classification) της κίνησης είναι να μπορεί να πάρει γρήγορα αποφάσεις, με όσο το δυνατόν μεγαλύτερη ακρίβεια.

Σκοπός είναι, στο τέλος της εργασίας, να έχει διαμορφωθεί μία σαφής εικόνα για το ποια είδη Νευρωνικών ταιριάζουν καλύτερα στο υπό εξέταση πρόβλημα και με ποιες παραμέτρους.

## 1.3 Δομή της Εργασίας

Στην Ενότητα 2 παρουσιάζεται το θεωρητικό υπόβαθρο σχετικά με τη Μηχανική Μάθηση (Machine Learning) και τα Νευρωνικά Δίκτυα (Neural Networks), αλλά και την Ασφάλεια Δικτύων. Στη συνέχεια, στην Ενότητα 3 δίνεται διαγραμματικά η γενική εικόνα του προτεινόμενου συστήματος, ενώ στην Ενότητα 4 παρουσιάζονται όλες οι λεπτομέρειες σχετικα με το προτεινόμενο σύστημα. Τέλος, δίνονται τα αποτελέσματα και τα σχόλια σχετικά με τις επιδόσεις των Νευρωνικών Δικτύων που δοκιμάστηκαν.

# 2 Θεωρητικό Υπόβαθρο

## 2.1 Μηχανική Μάθηση και Νευρωνικά Δίκτυα

Στην ενότητα αυτή παρουσιάζεται όλο το απαραίτητο θεωρητικό υπόβαθρο σχετικά με τον τομέα της Μηχανικής Μάθησης και κυρίως σχετικά με τα Νευρωνικά Δίκτυα. Παρουσιάζονται οι κυριότεροι τύποι Νευρωνικών Δικτύων και στοιχεία σχετικά με την εκπαίδευση και την αξιολόγηση.

### 2.1.1 Το πρόβλημα της Κατηγοριοποίησης (Classification)

Ο τομέας της Μηχανικής Μάθησης (Machine Learning) μπορεί να διακριθεί σε δύο ευρείες κατηγορίες, όπως φαίνεται και στον πίνακα 1.

**Ελεγχόμενη Μάθηση (Supervised Learning)**
Χαρακτηριστικό της Ελεγχόμενης Μάθησης είναι ότι γνωρίζουμε εκ των προτέρων σε πόσες και ποιες κατηγορίες θα κατανεμηθούν τα δείγματα (samples).
Επιπλέον, υπάρχουν κάποια δείγματα για τα οποία γνωρίζουμε ακριβώς σε ποια κατηγορία ανήκουν. Τα δείγματα αυτά χρησιμοποιούνται κατά τη φάση της Εκπαίδευσης του συστήματος (βλ. και Ενότητα 2.1.2).

**Μη ελεγχόμενη Μάθηση (Unsupervised Learning)**
Αντίθετα, στην Μη Ελεγχόμενη Μάθηση δεν υπάρχουν δείγματα για τα οποία γνωρίζουμε την Κατηγορία στην οποία ανήκουν. Συνήθως δεν είναι γνωστές ούτε οι Κατηγορίες εκ των προτέρων. Στόχος του αλγορίθμου είναι να βρει τις ομοιότητες μεταξύ αυτών και να τα κατατάξει.

**Κατηγοριοποίηση**
Η Κατηγοριοποίηση (Classification), το πρόβλημα στο οποίο εντάσσεται η παρούσα διπλωματική, ανήκει στην Ελεγχόμενη Μάθηση. Στόχος είναι να εκπαιδευτεί το σύστημα Μηχανικής Μάθησης στο να αποκρίνεται, δοθέντος ενός δείγματος, σε ποια κατηγορία αυτό ανήκει.

| **Ελεγχόμενη (Supervised) Μάθηση** | **Μη-Ελεγχόμενη (Unsupervised) Μάθηση** |
|---|---|
| Κατηγοριοποίηση (Classification) | Ομαδοποίηση (Clustering) |
| Παλινδρόμηση (Regression) | Μείωση διαστάσεων (Dimesionality Reduction) |

Πίνακας 1: Οι βασικές κατηγορίες Μηχανικής Μάθησης

### 2.1.2 Φάσεις σε ένα Σύστημα Κατηγοριοποίησης

Ανεξάρτητα από την τεχνική Μηχανικής Μάθησης που θα χρησιμοποιηθεί για την επίλυση του προβλήματος της Κατηγοριοποίησης, οι βάσικές φάσεις στις οποίες αυτό θα βρεθεί είναι οι ίδιες. Αυτές είναι η Εκπαίδευση, ο Έλεγχος και η Λειτουργία.
Κατ΄ αρχάς, αφού το πρόβλημα της Κατηγοριοποίησης (Classification) εντάσσεται στην Ελεγχόμενη Μάθηση (Supervised Learning), απαιτείται η ύπαρξη κάποιων δεδομένων με γνωστή την Κατηγορία (Class) στην οποία εντάσσονται. Τα δεδομένα αυτά τα χωρίζουμε συνήθως σε τρία σύνολα, ξένα μεταξύ τους. Αυτά είναι τα **Δεδομένα Εκπαίδευσης (Training Data)**, τα Δεδομένα Επαλήθευσης (**Validation Data**) και τα Δεδομένα Ελέγχου (**Test Data**). Το ποσοστό τους επί των αρχικών διαθέσιμων δεδομένων είναι συνήθως 60%, 20% και 20% αντίστοιχα. Κάθε σύνολο χρησιμοποιείται στην αντίστοιχη φάση.

#### Εκπαίδευση (Training)

Κατά τη φάση της Εκπαίδευσης καθορίζεται η τιμή όλων των βαρών του συστήματος. Εδώ χρησιμοποιούνται τα **Training Data**. Το απαιτούμενο πλήθος εξαρτάται από τη φύση του συστήματος, αλλά η γενική αρχή είναι ότι περισσότερα Δεδομένα Εκπαίδευσης συνεπάγονται και καλύτερη δυνατότητα μάθησης.

#### Έλεγχος (Testing)

Τονίζεται ότι στη φάση της Εκπαίδευσης είναι συχνό να δοκιμάζονται διαφορετικά συστήματα, έστω διαφορετικά Νευρωνικά Δίκτυα, ή Νευρωνικά Δίκτυα με διαφορετικές παραμέτρους. Στο σημείο αυτό, λοιπόν, χρησιμοποιούνται τα **Validation Data** για να συγκριθούν τα διαφορετικά δίκτυα μεταξύ τους. Ο λόγος που χρησιμοποιείται ένα ξεχωριστό σύνολο και όχι τα ίδια τα Training Data έχει να κάνει με την αντικειμενικότητα και τη δυνατότητα γενίκευσης του συστήματος. Τα διαφορετικά δίκτυα πρέπει να ελεγχθούν σε εισόδους που τους είναι άγνωστες, όχι στις εισόδους πάνω στις οποίες εκπαιδεύτηκαν.
Έπειτα, αφού βρεθεί το δίκτυο που δίνει τις βέλτιστες μετρικές πάνω στα δεδομένα Επαλήθευσης, χρησιμοποιούμε και το τρίτο σύνολο, τα Δεδομένα Ελέγχου (**Test Data**). Αυτά θα μας δώσουν μία τελική τιμή για την ακρίβεια την οποία αναμένουμε να έχει το σύστημά μας. Ο λόγος που χρησιμοποιούμαι και εδώ ένα νέο σύνολο έχει να κάνει και πάλι με την αντικειμενικότητα του ελέγχου.

#### Λειτουργία και Επανεκπαίδευση

Τέλος, το δίκτυό μας είναι σε αυτό το σημείο έτοιμο να ταξινομήσει νέα δεδομένα, για τα οποία φυσικά δε γνωρίζουμε εκ των προτέρων την κατηγορία στην οποία ανήκουν.
Σε ορισμένες περιπτώσεις, ειδικά όταν τα μοτίβα στις εισόδους μεταβάλονται με το χρόνο, απαιτείται το σύστημα να επανεκπαιδευτεί, ώστε να προσαρμοστεί σε αυτά. Αυτό, φυσικά, ισοδυναμεί με την ανάγκη για εξεύρεση νέων δεδομένων με γνωστή την Κατηγορία στην οποία ανήκουν (Labeled Data).

### 2.1.3 Ο Νευρώνας (Perceptron)

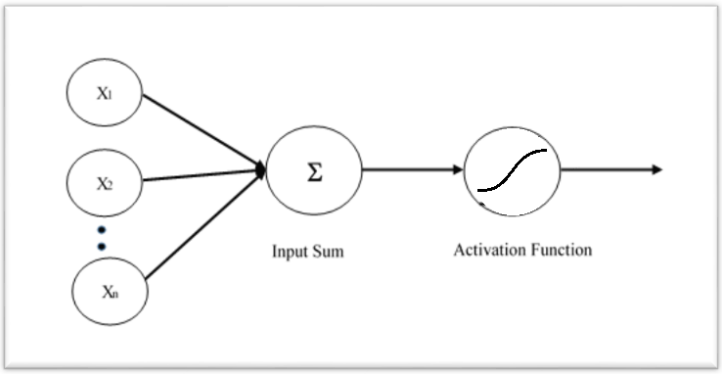
Το βασικό δομικό στοιχείο ενός Νευρωνικού Δικτύου είναι ο Νευρώνας (Perceptron). Στο σχήμα 1 φαίνεται ένας νευρώνας Ν εισόδων. Ο τύπος που δίνει την τιμή της εξόδου (Output) είναι ο ακόλουθος.

$$\sum_{i=1}^{n} w_i X_i + b \tag{1}$$

όπου
$X$ είναι το διάνυσμα εισόδων,
$w_i$ είναι τα βάρη που ο νευρώνας αποδίδει σε κάθες είσοδο,
$b$ είναι ο σταθερός όρος (bias)

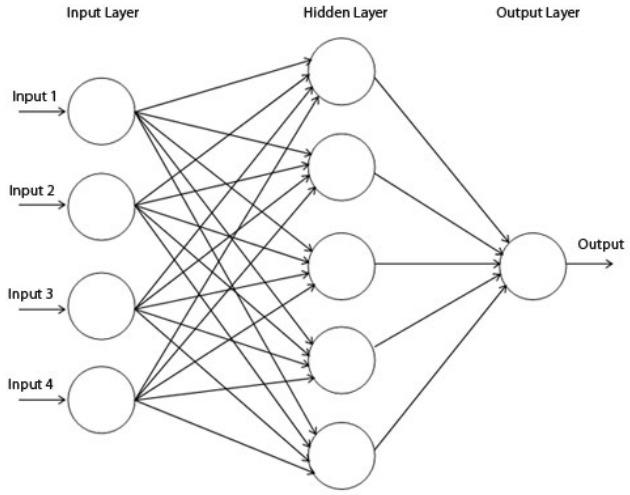
Όπως είναι φανερό, η εξίσωση 1 είναι γραμμική ως προς τις εισόδους $X_i$. Για να μπορεί ένα Νευρωνικό Δίκτυο να χειριστεί και μη-γραμμικές συναρτήσεις, η έξοδος κάθε νευρώνα τροφοδοτείται σε μία συνάρτηση ενεργοποίησης (Activation Function). Οι πιο συνηθισμένες συναρτήσεις ενεργοποίησης παρουσιάζονται στην Ενότητα 2.1.9. Αν $g$ είναι μία συνάρτηση ενεργοποίησης, η εξίσωση 1 γίνεται:

$$g(\sum_{i=1}^{n} w_i X_i + b) \tag{2}$$




Σχήμα 1: Ένας Νευρώνας Ν εισόδων με συνάρτηση ενεργοποίησης
Πηγή [7]

Όπως θα δούμε στην επόμενη Ενότητα, σε κάθε επίπεδο (layer) ενός Νευρωνικού Δικτύου χρησιμοποιούνται πολλοί Νευρώνες, καθένας από τους οποίους έχει ως είσοδο τις εξόδους του προηγουμένου επιπέδου. Ένα χαρακτηριστικό παράδειγμα φαίνεται στο σχήμα 2.

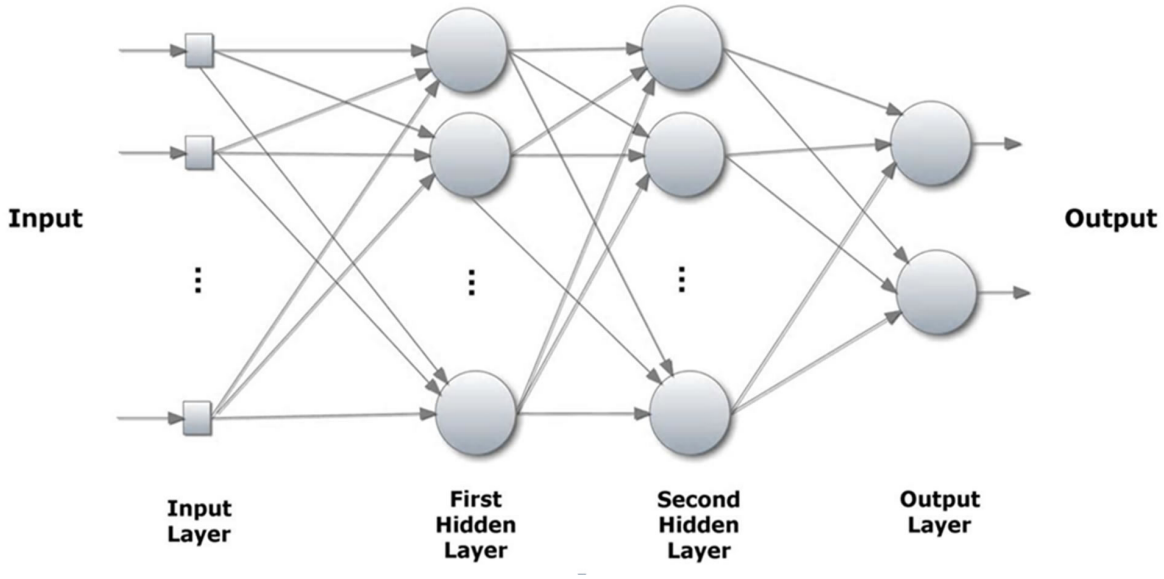



Σχήμα 2: Επίπεδο Νευρωνικού Δικτύου με πολλούς Νευρώνες
Πηγή [8]

### 2.1.4 Νευρώνας Πολλών Επιπέδων (Multilayer Perceptron, MLP)

Μία από τις πιο συνηθισμένες μορφές νευρωνικών δικτύων. Ένα MLP νευρωνικό δίκτυο έχει τις εξής ιδιότητες:

- Feedforward αποκλειστικά, καμία ανάδραση.
- Πλήρως συνδεδεμένα (fully connected) επίπεδα, δηλαδή κάθε νευρώνας ενός επιπέδου συνδέεται με όλους τους νευρώνες του επομένου. Οι νευρώνες του ίδιου επιπέδου δε συνδέονται μεταξύ τους.
- Μη-γραμμική συνάρτηση ενεργοποίησης (activation function).




Σχήμα 3: Ένα τυπικό MLP με δύο Κρυφά Επίπεδα
Πηγή [9]

Χαρακτηριστικά:

- Ένα Multi-Layer Perceptron δίκτυο, ακριβώς εξαιτίας της μη-γραμμικής συνάρτησης ενεργοποίησης, είναι σε θέση να υλοποιήσει γραμμικές και μη-γραμμικές συναρτήσεις.
- Σύμφωνα με το paper[10] των Kurt Hornik, Maxwell Stinchcombe, Halbert White από το 1988, τα Multilayer Feedforward Networks μπορούν να προσεγγίσουν οποιαδήποτε συνάρτηση με οσηδήποτε ακρίβεια.

> [...] standard multilayer feedforward networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable

> function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators.

- Εκπαίδευση με back-propagation.

Δυνατότητα εφαρμογής στην Κατηγοριοποίηση δικτυακής κίνησης:

- Εκ φύσεως δεν έχει «μνήμη», δε μπορεί να θυμάται προηγούμενες εισόδους/ροές και να γνωρίζει την κατάσταση του συστήματος. Αυτό αναμένεται να αποτελέσει μειονέκτημα. Για παράδειγμα, μία επίθεση Port Scanning θα ανιχνευόταν πιο εύκολα αν το δίκτυο είχε την πληροφορία για το μεγάλο εύρος των θυρών προορισμού (destination ports) – τώρα δεν την έχει, βλέπει μόνο τη θύρα προορισμού κάθε ροής. Επίσης, σε μία DDoS, σημαντικό χαρακτηριστικό της οποίας είναι η μεγάλη εντροπία στις διευθύνσεις προέλευσης, το δίκτυο θα αγνοεί την πληροφορία αυτή καθ' αυτή (ωστόσο έχει άλλες όπως πλήθος πακέτων ανά ροή, διάρκεια, μέγεθος πακέτων)
- Ανάγκη για προεπεξεργασία των δεδομένων, όπως κανονικοποίηση. Αυτό βέβαια ισχύει για όλα τα είδη νευρωνικών δικτύων.
- Ανάγκη ύπαρξης labeled data - δηλαδή δεδομένων για τα οποία γνωρίζουμε από πριν την κατηγορία στην οποία ανήκουν - για την εκπαίδευση του δικτύου. Η ανάγκη αυτή υπάρχει προφανώς πάντα όταν κάνουμε Ελεγόμενη Μάθηση (Supervised Learning)
- Αφού ολοκληρωθεί η εκπαίδευση του δικτύου, η κατηγοριοποίηση νέων δειγμάτων (με feedforward) γίνεται γρήγορα.
- Αναμένω μέτρια ακρίβεια με σχετικά χαμηλό recall (βλ. Ενότητα 2.1.14)

Το 2016, οι Oliveira, Barbar και Soares μελέτησαν και συνέκριναν τις επιδόσεις τριών προσεγγίσεων Νευρωνικών Δικτύων – MLP, RNN και SAE (Stacked Autoencoder) – πάνω στην εργασία της πρόβλεψης του όγκου της δικτυακής κίνησης. Το βέλτιστο αποτέλεσμα, τόσο σε χρόνο εκπαίδευσης όσο και ακρίβειας στην πρόβλεψη, επιτεύχθηκε από το RNN δίκτυο. Το MLP έδωσε επίσης ικανοποιητικά αποτελέσματα, με ακρίβεια πρόβλεψης παρόμοια του RNN.

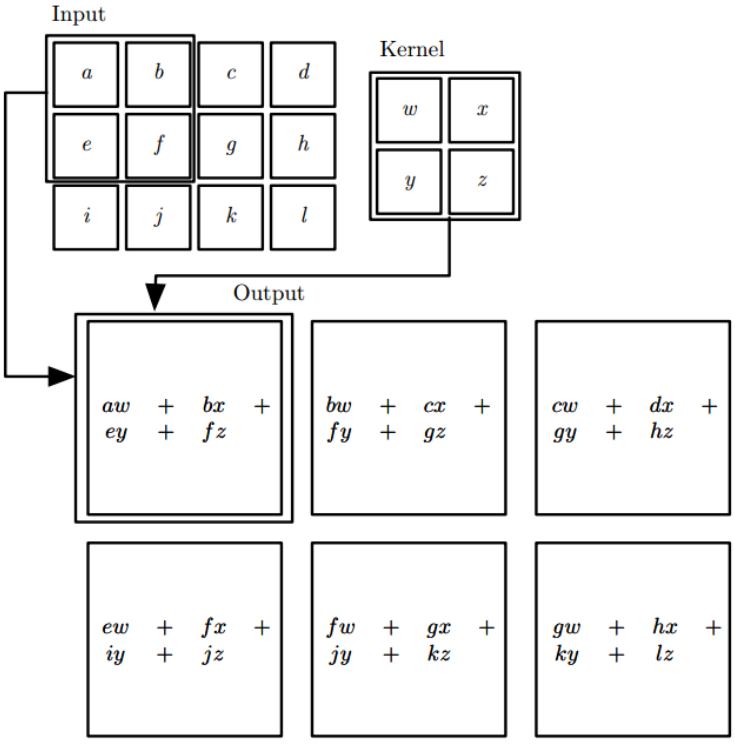
Όλα τα ανωτέρω οδηγούν στο συμπέρασμα ότι το MLP δίκτυο μπορεί να χρησιμοποιηθεί με σκοπό την κατηγοριοποίηση της δικτυακής κίνησης και θα είναι το πρώτο είδος Νευρωνικού Δικτύου που θα δοκιμάσουμε.

### 2.1.5 Συνελλικτικό Νευρωνικό Δίκτυο (Convolutional Neural Network, CNN)

Ένα ερώτημα που έχει προκύψει στον τομέα της μηχανικής μάθησης είναι το εξής. Πώς μπορούμε να επεξεργαστούμε μεγάλες εισόδους, π.χ. μεγάλες εικόνες, για να αποφανθούμε αν ένα συγκεκριμένο αντικείμενο εμφανίζεται σε αυτές· Επίσης, πώς μπορούμε να επεξεργαστούμε μία συνεχή ροή δεδομένων, έστω δεδομένων φωνής, αγνώστου μήκους·

Στα κλασικά Νευρωνικά Δίκτυα, όπως το MLP, κάθε νευρώνας ενός επιπέδου συνδέεται, και άρα μεταφέρει την πληροφορία του, σε όλους τους νευρώνες του επομένου επιπέδου. Αντίστροφα, κάθε νευρώνας αποκτά μία τιμή που προκύπτει και επηρεάζεται από τις τιμές όλων των νευρώνων του προηγουμένου επιπέδου.

Βασική ιδέα ενός Συνελικτικού Νευρωνικού Δικτύου (CNN) είναι η χρήση μίας μάσκας-πυρήνα (kernel) μεγέθους – αρκετά, συνήθως – μικροτέρου από την είσοδο. Όπως φαίνεται και στην εικόνα 4, ο πυρήνας αυτός εφαρμόζεται επανειλημμένα πάνω στην είσοδο. Με αυτή την τεχνική διευκολύνεται η επεξεργασία μεγάλων εισόδων και γίνεται εφικτή, όπως θα δούμε, η επεξεργασία εισόδων μεταβλητού μήκους.




Σχήμα 4: Η ιδέα της Συνέλιξης σε ένα CNN
Πηγή [9]

Επειδή τα Συνελικτικά Νευρωνικά Δίκτυα παρουσιάζουν αρκετά πλεονεκτήματα και επειδή έχουν χρησιμοποιηθεί με επιτυχία στην αναγνώριση προτύπων σε δεδομένα εικόνας και ήχου, εξετάστηκε, στα

πλαίσια της παρούσας Διπλωματικής, και η δυνατότητά χρήσης τους στην αναγνώριση προτύπων στη δικτυακή κίνηση.

Ωστόσο, επειδή τα χαρακτηριστικά της δικτυακής κίνησης (όπως αυτά συλλέγονται από το πρότυπο Netflow) δεν έχουν μεταξύ τους κάποια χωρική συσχέτιση, δε φάνηκε εφικτή η χρήση των CNN.

### 2.1.6 Αναδρασιακό Νευρωνικό Δίκτυο (Recurrent Neural Network, RNN)

Τα RNN είναι, σε αντιστοιχία με τα CNN, νευρωνικά δίκτυα ικανά να επεξεργαστούν ακολουθίες εισόδων. Όπως ακριβώς είδαμε ότι τα CNN ταιριάζουν σε περιπτώσεις όπου τα δεδομένα είναι οργανωμένα σε χωρικές διαστάσεις (π.χ. εικόνες σε 2-διάστατο πλέγμα), έτσι και τα RNN προσφέρουν σημαντικά πλεονεκτήματα για δεδομένα που σχετίζονται με κάποια χρονική εξέλιξη (time series).
Όπως πολύ χαρακτηριστικά αναφέρει ο Alex Graves στο paper [11]:

> An MLP can only map from input to output vectors, whereas an RNN can in principle map from the entire history of previous inputs to each output.

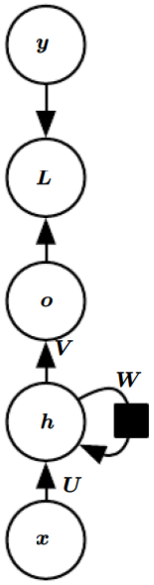
#### Χρήση κοινών παραμέτρων (Parameter Sharing) στα RNN

Εδώ, η τεχνική του Parameter Sharing θα μας βοηθήσει να επεξεργαστούμε εισόδους διαφορετικού και μη-καθορισμένου μήκους και έτσι να γενικευτεί η δυνατότητα του δικτύου να ταξινομεί την κίνηση. Θα εφαρμόζουμε το ίδιο πίνακα πάνω σε όλες τις εισόδους, αλλά τώρα οι είσοδοι μπορούν να είναι κομμάτια μιας μεγαλύτερης ακολουθίας.

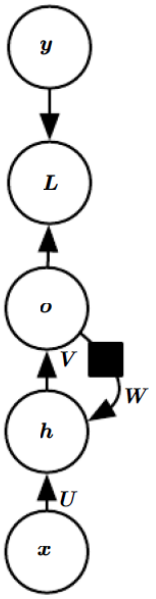
#### Είδος Ανάδρασης

Στα RNN έχουμε δύο βασικές προσεγγίσεις ως προς τον τρόπο με τον οποίο γίνεται η ανάδραση.

1. Από κρυφό (Hidden επίπεδο σε κρυφό επίπεδο. Βλ. σχήμα 5
2. Από το επίπεδο εξόδου (Output layer σε κάποιο κρυφό επίπεδο. Βλ. σχήμα 6
3. Και φυσικά συνδυασμός των δύο.




Σχήμα 5: RNN δίκτυο με ανάδραση τύπου 1, από κρυφό επίπεδο προς κρυφό επίπεδο.
Πηγή [9]




Σχήμα 6: RNN δίκτυο με ανάδραση τύπου 2, από το επίπεδο εξόδου προς κρυφό επίπεδο.
Πηγή [9]

Η πρώτη προσέγγιση είναι πιο ισχυρή αλλά η 2η μπορεί να εκπαιδευτεί πιο εύκολα. Στην περίπτωσή μας χρειαζόμαστε την ανάδραση τύπου 1, η απόκριση του συστήματος πρέπει να εξαρτάται από τις προηγούμενες εισόδους και όχι από τις προηγούμενες εξόδους.

**Βαθιά Αναδρασικά Δίκτυα (Deep RNN)**

Ένα Νευρωνικό Δίκτυο που περιέχει περισσότερα από ένα κρυφά επίπεδα ονομάζεται βαθύ νευρωνικό δίκτυο.

Η ροή της πληροφορίας και ο υπολογισμός (computation) σε ένα RNN μπορεί να χωριστεί στις ακόλουθες τρεις κατηγορίες.

1. Από την είσοδο στη κατάσταση ενός κρυφού επιπέδου(hidden state - αφού έχουμε ανατροφοδότηση στα κρυφά επίπεδα μπορούμε πλέον να μιλάμε και για κατάσταση των κρυφών επιπέδων).
2. Από μία κρυφή κατάσταση στην επόμενη.
3. Από την κρυφή κατάσταση στην έξοδο.

Η προσθήκη βάθους σε καθένα από αυτά τα στάδια μπορεί να οδηγήσει σε βελτίωση [12, 13]. Η βασική ιδέα είναι ότι χρειαζόμαστε ένα αρκετά βαθύ δίκτυο (Deep RNN), έτσι ώστε να είναι σε θέση να εκφραστούν και να αποκωδικοποιηθούν οι εξαρτήσεις των δεδομένων εισόδου.

**Το πρόβλημα της Εξασθένισης Βάρους (Weight Decaying) στα RNN και η αδυναμία έκφρασης μακροχρόνιων εξαρτήσεων (Long-Term Dependencies)**

Η παράγωγος, όταν τροφοδοτείται συνεχόμενα σε πολλά επίπεδα ενός δικτύου, έχει συχνά την τάση να παίρνει είτε πολύ μικρές είτε πολύ μεγάλες τιμές. Το γεγονός αυτό μπορεί να αποτελέσει πρόβλημα στην εκπαίδευση ενός RNN, λόγω των ανατροφοδοτήσεων που υπάρχουν σε αυτό. Μάλιστα, τις περισσότερες φορές η παράγωγος τείνει προς μικρές τιμές (vanishing – gradient problem), που σημαίνει ότι μία κατάσταση ή ένα πρότυπο (pattern) που έχει παρατηρηθεί δε θα μεταδοθεί (propagation) παρά μόνο στις αμέσως επόμενες καταστάσεις.

Είναι, λοιπόν, δύσκολο να κρατήσει το δίκτυο πληροφορίες σχετικά με μακροχρόνιες εξαρτήσεις. Το γεγονός έχει να κάνει με το ότι στα RNN ένας πίνακας που συμμετέχει στο βρόχο ανάδρασης πολλαπλασιάζεται πολλές φορές με τον εαυτό του και έτσι οι τιμές του τείνουν προς το 0 (ή σπανιότερα προς το άπειρο) [9]. Από τα δύο προαναφερθέντα φαίνεται ότι η έκφραση των Long-Term Dependencies σε ένα νευρωνικό δίκτυο αποτελεί ένα σημαντικό ζήτημα και αντικείμενο μελέτης. Προς την επίλυσή τους έχουν γίνει διάφορες προτάσεις, όπως η χρήση συναρτήσεων ενεργοποίησης (activation functions) με τιμές που δεν τείνουν στο άπειρο, όπως π.χ. η ReLU ή η softmax (βλέπε Ενότητα 2.1.9. Επίσης, η τεχνική των LSTM, που παρουσιάζονται στο αμέσως επόμενο κεφάλαιο, έχει προταθεί για την επίλυση αυτών των προβλημάτων.

### 2.1.7 Προσθέτοντας Μνήμη στα RNN: Το Δίκτυο LSTM

**Η τεχνική των Leaky Units**

Μία προσέγγιση στο πρόβλημα της εξασθένισης των βαρών (Weight Decaying) είναι να θέσουμε αναδράσεις μόνο από κάθε νευρώνα προς τον εαυτό του και να τους δώσουμε ένα σταθερό βάρος, έστω $\alpha$. Έτσι, κάθε μονάδα θα δίνει τη χρονική στιγμή $t$ στην έξοδό της την τιμή

$$\mu^{(t)} = a\mu^{(t-1)} + (1-a)u^{(t)} \tag{3}$$

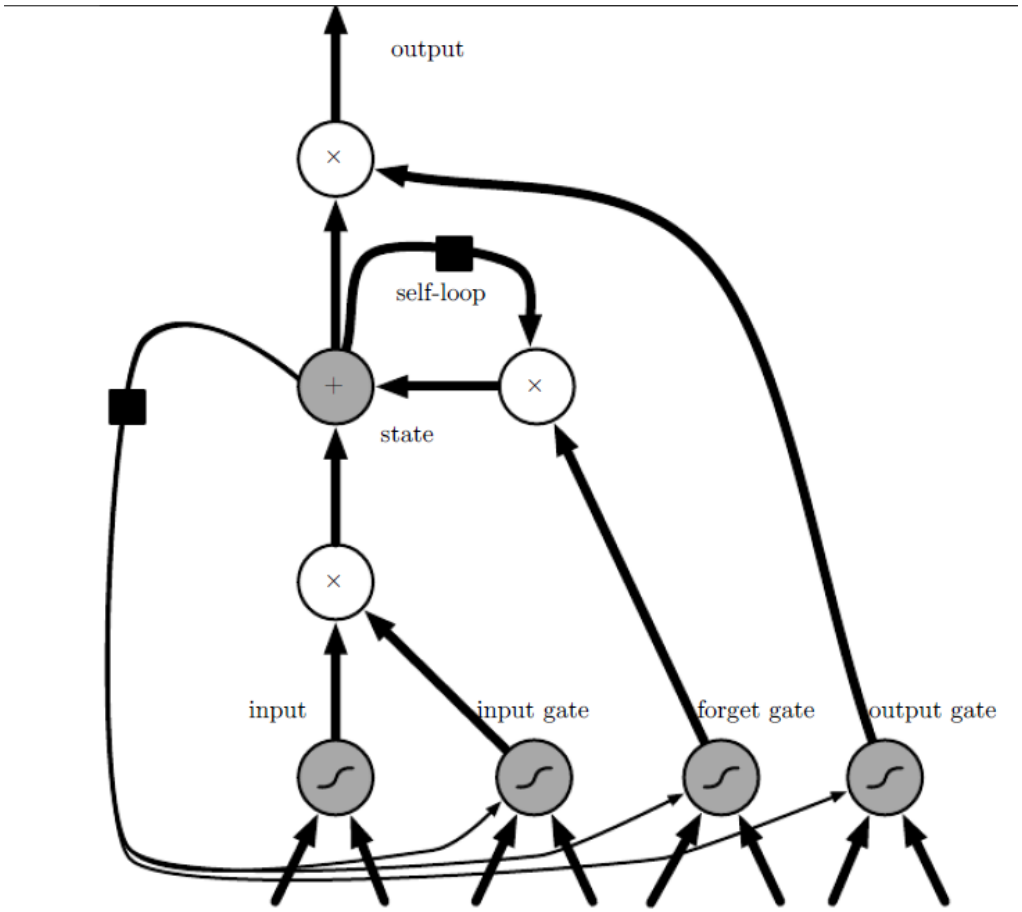
Αν το α μεγάλο, κοντά στο 1, τότε ο νευρώνας θα θυμάται πληροφορίες από μακριά στο παρελθόν, ενώ αν είναι μικρό, κοντά στο 0, οι πληροφορίες του παρελθόντος θα ξεχνιούνται γρήγορα. Η προσέγγιση αυτή αποτελεί την τεχνική των Leaky Units, έχει όμως το μειονέκτημα ότι το $\alpha$ αυτό θα είναι σταθερό (ακόμα και αν δεν το θέσουμε εξ' αρχής και αφήσουμε τα σύστημα να το μάθει).

**Η τεχνική των Skip Connections**

Μία άλλη τεχνική χρησιμοποιεί τις λεγόμενες Skip Connections, οι οποίες προσθέτουν στο σύστημα ανάδραση όχι από τη χρονική στιγμή $t$ στην $t+1$, αλλά από την $t$ στην $t+d$, όπου το $d$ θα είναι και πάλι μία σταθερά. Με $d > 1$ το σύστημα μπορεί να βοηθηθεί στο να θυμάται πληροφορίες από το παρελθόν, αλλά και πάλι, αφού το $d$ είναι σταθερά, το σύστημα δε θα μπορεί να προσαρμοστεί σε ιδιαιτερότητες της εισόδου.

**Τα δίκτυα LSTM**

Την πιο αποτελεσματική λύση αυτή τη στιγμή στην πράξη [9] δίνουν τα λεγόμενα Νευρωνικά Δίκτυα με πύλες (gated RNNs), κατηγορία στην οποία ανήκουν και τα LSTM RNNs (Long Short-Term Memory RNNs). Σε αυτά, το κλασικό κύτταρο (cell)(που χρησιμοποιείται στα Νευρωνικά δίκτυα και εφαρμόζει ουσιαστικά έναν πολλαπλασιασμό των εισόδων με βάρη και μετά έναν μη-γραμμικό μετασχηματισμό) αντικαθίσταται με ένα gated cell. Αυτό εξωτερικά παραμένει ίδιο με το κλασικό, αλλά εσωτερικά χρησιμοποιεί πύλες, με βάρη που μαθαίνει, για ελέγξει την ανάδραση.




Σχήμα 7: Η εσωτερική δομή ενός LSTM κυττάρου
Κάθε LSTM κύτταρο έχει εσωτερικά ένα cell στο οποία διατηρεί την κατάστασή του (το state στο σχήμα και χρησιμοποιεί τρεις πύλες. Και οι τρεις δέχονται την ίδια είσοδο από το εξωτερικό του LSTM κυττάρου και παράγουν έξοδό στο διάστημα 0 έως 1, με βάρη που βάρη που μαθαίνουν. Η πρώτη, η input gate, καθορίζει το συντελεστή με τον οποίο θα πολλαπλασιαστεί η εξωτερική είσοδος του LSTM κυττάρου. Η δεύτερη, η forget gate, καθορίζει το συντελεστή με τον οποίο θα πολλαπλασιαστεί η εσωτερική ανάδραση του LSTM κυττάρου. Η επόμενη κατάσταση καθορίζεται από την τρέχουσα κατάσταση και από την ανάδραση. Τέλος, η τρίτη, η output gate, καθορίζει το συντελεστή με τον οποίο η εσωτερική κατάσταση του κυττάρου θα περάσει στην έξοδο (μπορεί άρα να απενεργοποιήσει συνολικά το LSTM cell). Πηγή [9]

**Εξισώσεις**

Ακολουθούν οι εξισώσεις που διέπουν τη λειτουργία του LSTM κυττάρου.

Με $U$ συμβολίζουμε τον πίνακα με τα βάρη που θα πολλαπλασιάσουν την εξωτερική είσοδο.

Με $W$ συμβολίζουμε τον πίνακα με τα βάρη που θα πολλαπλασιάσουν την εξωτερική ανάδραση. Υπενθυμίζεται εδώ ότι εξωτερικά ένα LSTM δίκτυο δε διαφέρει σε τίποτα από ένα RNN , άρα κάθε νευρώνας μπορεί να δέχεται ανάδραση τύπου 1 ή τύπου 2, όπως αυτές ορίστηκαν στην Ενότητα 2.1.6.

Με $b$ συμβολίζεται η σταθερά (bias).

Τα $b$, $U$ και $W$ θα έχουν δείκτη $g$ όταν αφορούν στην input gate, δείκτη $f$ όταν αφορούν στην forget gate και δείκτη $o$ όταν αφορούν στην output gate.

Έτσι έχουμε, για την input gate:

$$g_i^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + W_{i,j}^g h_j^{(t-1)}) \tag{4}$$

Για τη forget gate:

$$f_i^{(t)} = \sigma(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + W_{i,j}^f h_j^{(t-1)}) \tag{5}$$

Για την output gate:

$$o_i^{(t)} = \sigma(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + W_{i,j}^o h_j^{(t-1)}) \tag{6}$$

Η τιμή $s$ της κατάστασης του κυττάρου τη χρονική στιμγή τ καθορίζεται τόσο από την $s(t-1)$, πολλαπλασιασμένη με την τιμή της forget gate, όσο και από την εξωτερική είσοδο (εξωτερική είσοδος είναι ο όρος μέσα στην παρένθεση), πολλαπλασιασμένη με την τιμή της input gate:

$$s_i(t) = f_i(t)s_i(t-1) + g_i(t)\sigma(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{(i,j)} h_j^{(t-1)}) \tag{7}$$

Σε όλες τις προηγούμενες εξισώσεις, ο δείκτης $i$ υποδεικνύει τον $i$-οστό νευρώνα του επιπέδου στο οποίο βρισκόμαστε και ο δείκτης $j$, πάνω στον οποίο γίνονται όλα τα αθροίσματα, αναφέρεται σε όλους τους νευρώνες του προηγουμένου επιπέδου.

Το κύταρρο LSTM συνολικά χρησιμοποιεί 4-πλάσιες παραμέτρους από τον κλασικό νευρώνα:

- Τα $b$, $U$ και $W$, που είναι το bias, ο πίνακας για τις εξωτερικές εισόδους και ο πίνακας για τις εξωτερικές αναδράσεις.
- Μία τριάδα $b$, $U$ και $W$ για κάθε πύλη

**Τα GRU RNN δίκτυα**

Το κατά πόσο απαιτούνται και οι 3 πύλες είναι αντικείμενο έρευνας. Έχουν προταθεί [14] cells με 2 πύλες – update και ignore. Αυτά είναι τα λεγόμενα GRU RNNs (Gated Recurrent Units).

**Εφαρμογές των LSTM**

Τα τελευταία χρόνια εφαρμογές σε αναγνώριση φυσικής γλώσσας (speech recognition).
Ενδεικτικά, αναφέρουμε από το paper των Graves et. al [13].

> When trained end-to-end with suitable regularisation, we find that deep Long Short-term Memory RNNs achieve a test set error of 17.7% on the TIMIT phoneme recognition benchmark.

Κατέληξαν στα εξής συμπεράσματα:

- Η χρήση του LSTM ως κυττάρου δουλεύει πολύ καλύτερα από τη χρήση «κλασικών» νευρωνικών κυττάρων. (δοκίμασαν την tanh ως συνάρτηση ενεργοποίησης).
- Το βάθος (depth) του δικτύου είναι πιο σημαντικό από το μέγεθος (πλάτος) των επιπέδων.

**Εφαρμογή στην ταξινόμηση κίνησης - Προτεινόμενο δίκτυο**

Δεν έχουμε στατιστική εξάρτηση της εξόδου $y$ από προηγούμενες εξόδους, άρα δεν είναι απαραίτητο να χρησιμοποιήσουμε ανάδραση τύπου 2 (από το επίπεδο εξόδου σε κρυφό επίπεδο). Ωστόσο, μοιάζει απαραίτητο να χρησιμοποιήσουμε ανάδραση τύπου 1 (από κρυφό επίπεδο σε κρυφό επίπεδο). Έχουμε εξάρτηση από την προηγούμενη κατάσταση του συστήματος, όχι από την προηγούμενη έξοδο.
Για την ανίχνευση μια επίθεσης ή μιας ανωμαλίας στο ίντερνετ, αυτό που μας ενδιαφέρει είναι να ελέγξουμε για την ύπαρξη κάποιων συγκεκριμένων μοτίβων στην κίνηση, όπως π.χ. πολλές ροές (flows ) με λίγα πακέτα σε κάθε μία, ροές με διαφορετική IP προέλευσης ή θύρα προορισμού. Αυτά τα μοτίβα δε περιγράφονται σε μία μεμονωμένη είσοδο του συστήματός μας και μπορεί να εμφανιστούν σε διάφορες χρονικές στιγμές, έτσι το RNN δίκτυο φαίνεται να ταιριάζει στο classification που θέλουμε να κάνουμε.

### 2.1.8 Η Διαδικασία της Μάθησης Μέσω Παραγώγων (Gradient Based Learning)

Οι περισσότεροι σύγχρονοι αλγόριθμοι Βαθιάς Μάθησης (Deep Learning ) περιλαμβάνουν κάποιου είδους βελτιστοποίησης (Optimization). Πιο συγκεκριμένα περιλαμβάνουν την ελαχιστοποίηση (Minimization) κάποιας Συνάρτησης Κόστους (Cost Function ή Loss Function ή Error Function ή Objective Function, όλοι οι όροι χρησιμοποιούνται με την ίδια έννοια στα πλαίσια της Μηχανικής Μάθησης).
Με τον όρο ελαχιστοποίηση εννοούμε την εύρεση εκείνου του $x*$ για το οποίο η Συνάρτηση κόστους $f(x)$ παίρνει την ελάχιστη δυνατή τιμή της, είναι δηλαδή $x* = argmin f(x)$.

Η Συνάρτηση Κόστους $f(x)$, όπου το $x$ είναι παράμετρος ή διάνυσμα παραμέτρων, εκφράζει, στα πλαίσια της Μηχανικής Μάθησης, κάποιο σφάλμα και η ελαχιστοποίηση αυτού θα δώσει τις τελικές των παραμέτρων. Όταν, λοιπόν, λέμε ότι το σύστημα «μαθαίνει» τις τιμές του $x$, βρίσκει το $x$ εκείνο για το οποίο κάποια Συνάρτηση Κόστους ελαχιστοποιείται.

Φαίνεται λοιπόν ότι δύο είναι τα βασικά στοιχεία ενός αλγορίθμου μάθησης.
Πρώτον, η Συνάρτηση Κόστους, η συνάρτηση δηλαδή που αξιολογεί τις εξόδους του αλγορίθμου μάθησης και αναθέτει έναν αριθμό - ποινή ή Κόστος (Penalty) - σε κάθε έξοδο. Έχουν προταθεί και χρησιμοποιηθεί πολλές Συναρτήσεις Κόστους. Οι σημαντικότερες και σχετικές με τη μάθηση μέσω Βαθέων Νευρωνικών Δικτύων (Deep Neural Networks) θα παρουσιαστούν στην Ενότητα 2.1.10.
Δεύτερον, απαιτείται μία μέθοδος η οποία, έχοντας ως είσοδο τις παραμέτρους του αλγορίθμου μάθησης και την Ποινή που ανατέθηκε στην έξοδο του αλγορίθμου θα προτείνει νέες τιμές για τις παραμέτρους, με σκοπό να μειωθεί η τιμή της Ποινής. Με άλλα λόγια, απαιτείται μία Μέθοδος Βελτιστοποίησης (Optimizer). Οι πλέον ευρέως χρησιμοποιούμενες μέθοδοι βελτιστοποίησης παρουσιάζονται στην Ενότητα. 2.1.11

### 2.1.9 Συναρτήσεις Ενεργοποίησης(Activation Functions)

Όπως φαίνεται και από τις εξισώσεις που διέπουν τη λειτουργία του, ένας Νευρώνας παράγει έξοδο που είναι γραμμική συνάρτηση των εισόδων του.
Επειδή κάτι τέτοιο θα ήταν περιοριστικό, όσον αφορά στις συναρτήσεις και τις κατανομές που μπορεί να μάθει ένα Νευρωνικό δίκτυο, χρησιμοποιούνται ευρέως οι Συναρτήσεις Ενεργοποίησης (Activation Functions).

Έστω $x$ το διάνυσμα εισόδων ενός Νευρώνα και $W$ το διάνυσμα των βαρών. Τότε, η έξοδός του θα είναι

$$z = W * x + b = W_0 * x_0 + W_1 * x_1 + ... + W_n * x_n + b \tag{8}$$

Χρησιμοποιώντας μία Συνάρτηση ενεργοποίησης $g$, η έξοδος γίνεται:

$$y = g(z) \tag{9}$$

Στη συνέχεια παρουσιάζονται οι πιο διαδεδομένες συναρτήσεις ενεργοποίησης.

1. **Γραμμική**
   Ουσιαστικά δε χρησιμοποιείται Συνάρτηση Ενεργοποίησης και η έξοδος παραμένει γραμμική ως προς τις εισόδους.

2. **Σιγμοειδής (sigmoid)**

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{10}$$

   όπου $z$ όπως περιγράφεται στην 8.
   Χαρακτηριστικά της sigmoid Συνάρτησης Ενεργοποίησης:

   (α′) Κορεσμός (Saturation) για $z >> 0$ και $z << 0$.

   (β′) Χρησιμοποιείται κυρίως σε επίπεδα εξόδου (Output layers) όταν η έξοδος παίρνει τιμές 0 και 1.

   (γ′) Χρησιμοποιείται συχνά με τη Συνάρτηση κόστους crossentropy (βλ. 2.1.10).

3. **Softmax**

   Αποτελεί γενίκευση της sigmoid για Categorical μεταβλητές, δηλαδή μεταβλητές με τιμές σε περισσότερες από 2 κλάσεις.

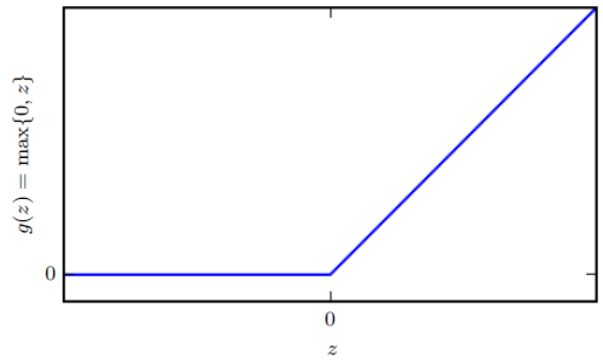
$$softmax(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{k} z_i} \tag{11}$$

   όπου $k$ το πλήθος των νευρώνων στο τρέχον επίπεδο.

4. **Rectified Linear Unit, ReLU**

$$y = max(0, z) \tag{12}$$

   Είναι η συνάρτηση ενεργοποίησης που προτείνεται σήμερα στα περισσότερα δίκτυα εμπρόσθιας τροφοδότησης (Feeforward Networks) (Πηγή [9]). Η εφαρμογή αυτής της συνάρτησης σε έναν γραμμικό μετασχηματισμό (όπως αυτόν της εξίσωσης 8) δημιουργεί έναν μη-γραμμικό μετασχηματισμό, διατηρεί όμως πολλές ιδιότητες των γραμμικών μοντέλων, που βοηθούν κατά την εκπαίδευση. Στην Εικόνα 8 φαίνεται γραφικά η έξοδος της ReLU και η ομοιότητά της με μία γραμμική συνάρτηση.




Σχήμα 8: Η Συνάρτηση Ενεργοποίσης (Activation Function) ReLU (Rectified Linear Unit) Πηγή [9]

5. **Maxout**

   Γενίκευση της ReLU, προτάθηκε από τον Ian Goodfellow, επιστρέφει το μέγιστο μεταξύ δύο γειτονικών στοιχείων.

$$y = max(W_1 x_1, W_2 x_2) \tag{13}$$

### 2.1.10 Συναρτήσεις Κόστους (ή Αντικειμενικές Συναρτήσεις, Loss Functions)

Προκειμένου να γίνει η εκπαίδευση ενός Νευρωνικού Δικτύου, απαιτείται ένας τρόπος αξιολόγησης της εξόδου του σε σχέση με την είσοδο και την αναμενόμενη έξοδο. Για το λόγο αυτό χρησιμοποιούνται οι Συναρτήσεις Κόστους, ή ισοδύναμα Αντικειμενικές Συναρτήσεις, (Loss Functions).
Παρουσιάζονται οι πιο ευρέως χρησιμοποιούμενες Συναρτήσεις κόστους. [9], [15], [16]
Στα ακόλουθα, τα $y_{true}$, $y_{pred}$ είναι τα διανύσματα της πραγματικής, αναμενόμενης τιμής και της εξόδου του δικτύου αντίστοιχα. Όπου χρησιμοποιείται ο δείκτης $j$, δηλώνει το $j$-οστό στοιχείο του αντιστοίχου διανύσματος. Το $\sigma$ δηλώνει εκτίμηση πιθανότητας (probability estimate).

1. **L1 ή MAE (Minimum Absolute Error)**

$$|y_{true} - y_{pred}| \tag{14}$$

2. **L2 ή MSE (Minimum Squared Error)**

$$|y_{true} - y_{pred}|^2 \tag{15}$$

   Είναι μία ειδική περίπτωση Maximum Likelihood Estimator (βλέπε και 4), όπου θεωρούμε ότι τα δεδομένα που έχουμε προς εκπαίδευση (το $y$ ως συνάρτηση της εισόδου $x$) ακολουθούν κανονική κατανομή.
   Συνηθίζεται περισσότερο σε προβλήματα Παλινδρόμησης (Regression). Είναι η προεπιλεγμένη (default) Συνάρτηση Κόστους στη Γραμμική Παλινδρόμηση (Linear Regression), αλλά χρησιμοποιείται πολύ και στην εκπαίδευση νευρωνικών δικτύων.

3. **Hinge Loss**

$$\sum_j \max(0, \frac{1}{2} - y^j_{pred} * y^j_{true}) \tag{16}$$

   όπου το $y_{pred}$ πρέπει να μετατραπεί σε +- 1 μορφή. Χρησιμοποιείται σε ταξινόμηση (classification) με SVM (Support Vector Machines).

4. **Log Loss ή Maximum Likelihood ή Cross-Entropy**

$$-\sum_{j} y^{j}_{true} * \log \sigma(o)^{(j)}) \tag{17}$$

Συνηθίζεται περισσότερο σε προβλήματα Ταξινόμησης (Classification). Σήμερα είναι ευρέως χρησιμοποιούμενη σε Βαθιά Νευρωνικά Δίκτυα, ενώ έχει εκτενώς χρησιμοποιηθεί σε Αναγνώριση Φωνής (Speech Recognition) [17] και Αναγνώριση Γραφής [18].

### 2.1.11 Μέθοδοι Βελτιστοποίησης (ή Μέθοδοι Εκπαίδευσης, Optimizers)

Στην Υποενότητα αυτή παρουσιάζονται οι τεχνικές που χρησιμοποιούνται από το δίκτυο για τον προσδιορισμό της βέλτιστης τιμής των βαρών. Όλες έχουν ως είσοδο τη Συνάρτηση Κόστους (βλ. Ενότητα 2.1.10), όπως αυτή έχει υπολογιστεί πάνω σε ένα ή περισσότερα δείγματα (samples) εισόδου. Στα ακόλουθα, η Συνάρτηση κόστους συμβολίζεται με $L$.

1. **Στοχαστική Κατάβαση Δυναμικού (Stochastic Gradient Descent)**
   Η πλέον συνηθισμένη τεχνική βελτιστοποίησης σε Νευρωνικά Δίκτυα είναι η Στοχαστική Κατάβαση

   Δυναμικού (Stochastic Gradient Descent, SGD), η οποία αποτελεί μία επέκταση της απλούστερης τεχνικής Κατάβασης Δυναμικού (Gradient Descent).

   Ένα βασικό πρόβλημα στον τομέα της Μηχανικής Μάθησης είναι η αργή εκπαίδευση, όταν το πλήθος των δειγμάτων εκπαίδευσης (training samples) είναι μεγάλο, καθώς απαιτείται να υπολογιστεί η συνάρτηση κόστους για όλα αυτά. Η μέθοδος της Στοχαστικής Κατάβασης Δυναμικού επιλύει αυτό το πρόβλημα, με το να υπολογίζει τη συνάρτηση κόστους χρησιμοποιώντας ένα μικρό υποσύνολο (minibatch) των δειγμάτων εκπαίδευσης. Ουσιαστικά, προσπαθεί να προσεγγίσει τη μέθοδο Κατάβασης Δυναμικού χρησιμοποιώντας πολύ λιγότερα σημεία.

**Αλγόριθμος**

Είσοδοι:

Learning Rate $\epsilon$

Αρχικές τιμές βαρών $\theta$

Σε κάθε επανάληψη:

(α′) Υπολογισμός παραγώγων: $g \Leftarrow \frac{1}{m}\nabla \sum_{i=0}^{m} L(y_{true}, y_{pred})$

(β′) Υπολογισμός ενημέρωσης: $\Delta\theta \Leftarrow -\epsilon g$

(γ′) Εφαρμογή ενημέρωσης: $\theta \Leftarrow \theta + \Delta\theta$

Η παράμετρος $\epsilon$ έχει πολύ μεγάλη σημασία για τον αλγόριθμο, καθώς καθορίζει το βαθμό στον οποίο οι παράγωγοι των βαρών θα επηρεάζουν τα βάρη.
Μικρή τιμή του $\epsilon$ θα έχει ως αποτέλεσμα την αργή σύγκλιση του αλγορίθμου προς κάποιο ελάχιστο, καθώς οι ενημερώσεις στο $\theta$ θα το επηρεάζουν λίγο.
Από την άλλη πλευρά, μεγάλη τιμή του $\epsilon$ ενδέχεται να κάνει τη συνάρτηση κόστους να αυξηθεί μετά την εφαρμογή της ενημέρωσης και μπορεί να οδηγήσει ακόμα και στη μη-σύγκλιση του αλγορίθμου.
Στην πράξη, συνηθίζεται να δοκιμάζουμε επαναληπτικά διάφορες τιμές του $\epsilon$, ενώ συχνά χρησιμοποιείται μεταβλητή τιμή, η οποία μειώνεται σε κάθε επανάληψη, σε κάθε ενημέρωση των βαρών δηλαδή. Σε αυτή την περίπτωση, έχουμε:

$$\epsilon_k = (1-\alpha)\epsilon_0 + \alpha\epsilon_\tau, \alpha = \frac{k}{\tau}, k < \tau \tag{18}$$

Ο περιορισμός $k < \tau$ δηλώνει ότι μετά από $\tau$ ενημερώσεις διατηρούμε το $\epsilon$ σταθερό. Εδώ, τα $\epsilon_0$ και $\epsilon_\tau$ και $\tau$ είναι νέες παράμετροι:

$\epsilon_0$ είναι ο αρχικός βαθμός μάθησης (Learning Rate)

$\epsilon_\tau$ είναι το ελάχιστο Learning Rate που επιτρέπουμε στο σύστημα

$\tau$ είναι το πλήθος των ενημερώσεων των βαρών, μετά από το οποίο θέλουμε το $\epsilon$ να παραμείνει σταθερό.

2. **Στοχαστική Κατάβαση Δυναμικού με χρήση Ορμής (SGD με Momentum)**
Μικρή τροποποίησης της μεθόδου SGD, σύμφωνα με την οποία εισάγουμε μία μεταβλητή $u$ , η

οποία παίζει το ρόλο της ορμής στο σύστημά μας. Η $u$ θα είναι ένας κινητός μέσος όρος των βαθμίδων $g$ . Πλέον, κάθε νέο διάνυσμα $g$ των παραγώγων δε θα επηρεάζει την τιμή του $\theta$ τόσο όσο πριν, αλλά θα λαμβάνονται υπόψη και οι προηγούμενες τιμές του $g$ .

**Αλγόριθμος**
Είσοδοι:

Learning Rate $\epsilon$

Αρχικές τιμές βαρών $\theta$

Παράμετρος ορμής $\alpha$

Σε κάθε επανάληψη:

(α′) Υπολογισμός παραγώγων: $g \Leftarrow \frac{1}{m}\nabla \sum_{i=0}^{m} L(y_{true}, y_{pred})$

(β′) Υπολογισμός ενημέρωσης: $u \Leftarrow \alpha u - \epsilon g$

(γ′) Εφαρμογή ενημέρωσης: $\theta \Leftarrow \theta + u$

Η διαφοροποίηση σε σχέση με τη μέθοδο SGD έγκυται στο Βήμα 3. Πλέον δεν προσθέτουμε το διάνυσμα $g$ αλλά το $u$.

3. **RMSprop (Root Mean Square Propagation)**
Επέκταση της Adagrad, αντιμετωπίζει το πρόβλημα της γρήγορης εξασθένισης του βαθμού μάθη-

σης. Αντί να συσσωρεύει τις τιμές των παραγώγων από όλη την εκπαίδευση διατηρεί έναν κινητό μέσο όρο, που δε λαμβάνει υπόψη το μακρινό παρελθόν.

**Αλγόριθμος**
Είσοδοι:

Learning Rate $\epsilon$

Αρχικές τιμές βαρών $\theta$

Ρυθμός μείωσης $\rho$

Μικρή σταθερά $\delta$, συνήθως $10^{-7}$

Αρχικοποίηση:

$\rho = 0$

Σε κάθε επανάληψη:

(α′) Υπολογισμός παραγώγων: $g \Leftarrow \frac{1}{m}\nabla \sum_{i=0}^{m} L(y_{true}, y_{pred})$

(β′) Συνυπολογισμός προηγουμένων τιμών: $r \Leftarrow \rho r + (1-\rho) g * g$

(γ′) Υπολογισμός ενημέρωσης: $\Delta\theta \Leftarrow \frac{-\epsilon}{\sqrt{\delta + r}} g$

(δ′) Εφαρμογή ενημέρωσης: $\theta \Leftarrow \theta + \Delta\theta$

Αυτό που έχει τροποποιηθεί σε σχέση με τη μέθοδο AdaGrad είναι ο τρόπος με τον οποίο λαμβάνονται υπόψη οι προηγούμενες τιμές της βαθμίδας. Εδώ, η προηγούμενη τιμή του $r$ επηρεάζει τη νέα του τιμή μόνο κατά ένα ποσοστό $\rho$, ενώ το υπόλοιπο $(1-\rho)$ εξαρτάται μόνο από την τρέχουσα τιμή της βαθμίδας.
Το $\rho$ αποτελεί βέβαια μία νέα παράμετρο, η τιμή της οποίας συνήθως καθορίζεται πειραματικά.

4. **AdaGrad (Adaptive Gradient)**
   Δουλεύει και αυτή σε minibatch, όπως και η μέθοδος Στοχαστικής Κατάβασης Δυναμικού. Βασικός στόχος της είναι να δημιουργήσει προσαρμοστικό βαθμό μάθησης (Adaptive Learning Rate)

**Αλγόριθμος**

Είσοδοι:

Learning Rate $\epsilon$

Αρχικές τιμές βαρών $\theta$

Μικρή σταθερά $\delta$, συνήθως $10^{-7}$

Σε κάθε επανάληψη:

(α΄) Υπολογισμός παραγώγων: $g \Leftarrow \frac{1}{m}\nabla \sum_{i=0}^{m} L(y_{true}, y_{pred})$

(β΄) Συνυπολογισμός προηγουμένων τιμών: $r \Leftarrow r + g * g$

(γ΄) Υπολογισμός ενημέρωσης: $\Delta\theta \Leftarrow \frac{-\epsilon}{\delta+\sqrt{r}} g$

(δ΄) Εφαρμογή ενημέρωσης: $\theta \Leftarrow \theta + \Delta\theta$

Οι μόνες διαφοροποιήσεις από τη μέθοδο SGD είναι η ύπαρξη του Βήματος 2 και η ύπαρξη του παρονομαστή στο Βήμα 3. Στο $r$ συσσωρεύονται όλες οι προηγούμενες τιμές των παραγώγων. Έτσι, το $\epsilon$ μειώνεται πλέον όχι αυθαίρετα, αλλά ανάλογα με την τιμή (το μέτρο) των παραγώγων. Πλεονέκτημα είναι η επίτευξη προσαρμοστικού βαθμού μάθησης χωρίς να εισάγουμε επιπλέον σταθερές.
Εμπειρικά έχει φανεί [9] ότι η συσσώρευση τιμών από την αρχή της εκπαίδευσης μπορεί να οδηγήσει σε πολύ γρήγορη μείωση του learning rate και άρα σε αδυναμία εκπαίδευσης του δικτύου.

### 2.1.12 Προεπεξεργασία Δεδομένων

**Κανονικοποίηση (Normalization)**
Είναι ένα de facto βήμα προεπεξεργασίας των δεδομένων πριν την έναρξη της εκπαίδευσης. Με την κανονικοποίηση, όλα τα χαρακτηριστικά (features) των δεδομένων εκπαίδευσης (training data) ανάγονται στο ίδιο εύρος, πράγμα που είναι απαραίτητο για τη σωστή εκπαίδευση του δικτύου. Αν δε γινόταν αυτό το βήμα, τότε ένα χαρακτηριστικό που λαμβάνει τιμές σε ένα εύρος έστω $[0, A]$ θα επηρέαζε τις τελικές τιμές των βαρών (learned weights) περισσότερο από ένα χαρακτηριστικό που λαμβάνει τιμές στο διάστημα $[0, B]$, με $B < A$. Έχουν προταθεί και χρησιμοποιηθεί διάφορες τεχνικές κανονικοποίησης. Παρουσιάζουμε τις σημαντικότερες.[19]

1. **Γραμμική Κλιμάκωση (Linear Scaling) σε μοναδιαίο εύρος**

$$x \Leftarrow \frac{x - l}{u - l} \tag{19}$$

όπου $u$ και $l$ είναι το μέγιστο (upper bound) και το ελάχιστο (lower bound) ανά χαρακτηριστικό (feature).
Ως αποτέλεσμα, κάθε χαρακτηριστικό $x$ είναι στο εύρος $[0, 1]$.

2. **Γραμμική Κλιμάκωση (Linear Scaling) σε μοναδιαίο εύρος**
Αυτή η τεχνική κανονικοποίησης έχει ως στόχο την μετατροπή του χαρακτηριστικού $x$ σε $x'$, όπου το $x'$ έχει μηδενική μέση τιμή και μοναδιαία απόκλιση.

$$x \Leftarrow \frac{x - \mu}{\sigma} \tag{20}$$

Όπου $\mu$ είναι η μέση τιμή και $\sigma$ η απόκλιση κάθε χαρακτηριστικού $x$. Αν θεωρήσουμε ότι κάθε χαρακτηριστικό $x$ ακολουθεί Κανονική Κατανομή, τότε ο προηγούμενος τύπος έχει ως αποτέλεσμα κάθε feature να είναι στην κλίμακα $[-1, 1]$ με πιθανότητα 68 %.
Μία άλλη προσέγγιση, που εξασφαλίζει ότι κάθε χαρακτηριστικό θα είναι στην κλίμακα $[-1, 1]$ με πιθανότητα 99% είναι η:

$$x \Leftarrow \frac{\frac{x-\mu}{3\sigma} + 1}{2} \tag{21}$$

### 2.1.13 Δυνατότητα Γενίκευσης (Generalization) και Overfitting – Underfitting (Υπερπροσαρμογή στα Δεδομένα Εκπαίδευσης – Ελλιπής Εκπαίδευση)

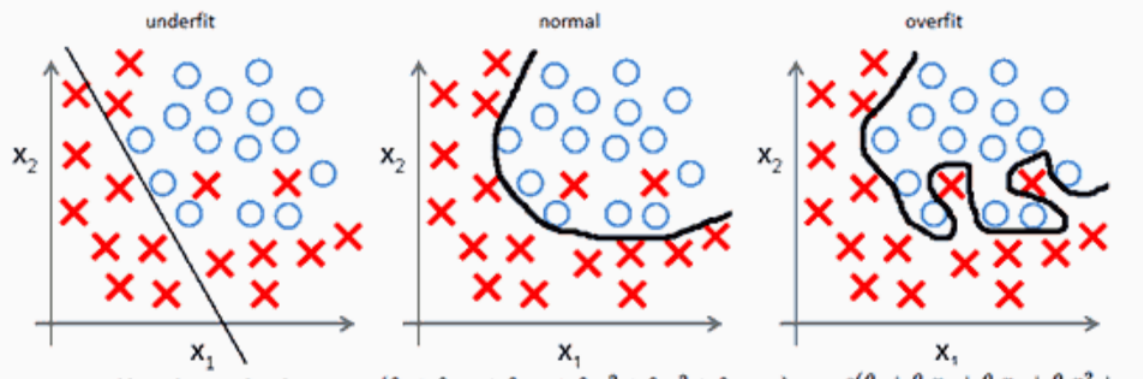
Βασικός στόχος και πρόκληση στη Μηχανική Μάθηση είναι να μπορεί το εκπαιδευμένο δίκτυο να προσαρμοστεί σε νέα δεδομένα, σε δεδομένα που δεν έχει προηγουμένως ξαναδεί, η δυνατότητα, δηλαδή, για γενίκευση (generalization).
Για να μετρήσουμε αυτή τη δυνατότητα γενίκευσης, είναι σύνηθες στη Μηχανική Μάθηση να χωρίζουμε τα διαθέσιμα δεδομένα σε δύο μέρη, το σετ εκπαίδευσης (training set) και το σετ ελέγχου (test set).
Η εκπαίδευση γίνεται χρησιμοποιώντας το training set και αποδίδει τιμές σε όλα τα βάρη του νευρωνικού δικτύου. Μας δίνει, επίσης, ως αποτέλεσμα την τιμή του σφάλματος εκπαίδευσης (training error), που είναι η τιμή της συνάρτησης κόστους (Loss function) μετά το πέρας της εκπαίδευσης, χρησιμοποιώντας δηλαδή τις τελικές τιμές των βαρών (τα learned weights).
Το test set μας δίνει μία επιπλέον τιμή, το σφάλμα ελέγχου (test error), που είναι η τιμή της συνάρτησης

κόστους με τις τελικές τιμές των βαρών που το δίκτυο έχει μάθει, αλλά αυτή τη φορά πάνω στο test set.
Οι δύο στόχοι είναι:

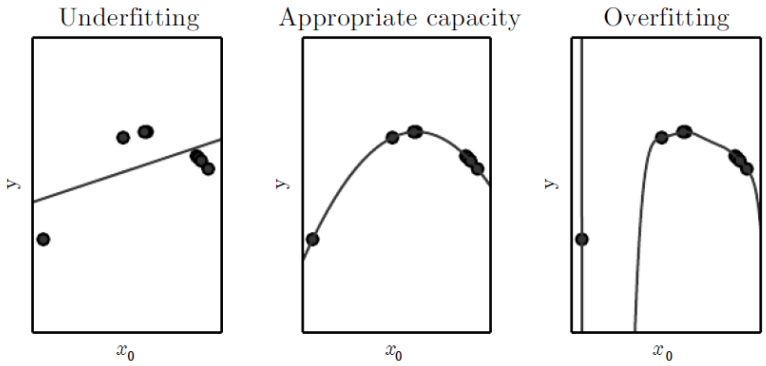
1. Η ελαχιστοποίηση του training error.
2. Η ελαχιστοποίηση της διαφοράς μεταξύ training error και test error.

Η αδυναμία εκπλήρωσης του πρώτου στόχου ονομάζεται **Underfitting**. Συμβαίνει όταν το δίκτυο δε μπορεί να προσεγγίσει καθόλου καλά τις αναμενόμενες εξόδους. Αυτό μπορεί να συμβαίνει για διάφορους λόγους, όπως η χαμηλή χωρητικότητα του δικτύου (έχουμε π.χ. χρησιμοποιήσει λίγα επίπεδα ή μικρό αριθμό νευρώνων ανά επίπεδο), η χρήση μόνο γραμμικών συναρτήσεων ενεργοποίησης (activation functions) ενώ τα δεδομένα δεν παρουσιάζουν γραμμική εξάρτηση κ.α.

Η αδυναμία εκπλήρωσης του δεύτερου στόχου ονομάζεται **Overfitting**. Συμβαίνει όταν καταφέρνουμε να ελαχιστοποιήσουμε το σφάλμα πάνω στα δεδομένα εκπαίδευσης αλλά όχι πάνω στα δεδομένα ελέγχου. Και εδώ υπάρχουν πολλοί λόγοι για τους οποίους αυτό μπορεί να συμβαίνει. Ο κυριότερος είναι το μικρό πλήθος δεδομένων εκπαίδευσης, που έχει ως αποτέλεσμα να μάθει το δίκτυο τιμές για τα βάρη που ναι μεν ελαχιστοποιούν - ή και μηδενίζουν – το training error, αλλά δε δίνουν σωστές προβλέψεις στο test set. Μία ακόμη αιτία είναι η πολύ μεγάλη χωρητικότητα του δικτύου ή η υπερβολική εκπαίδευση (δηλαδή εκπαίδευση με πάρα πολλές επαναλήψεις, πάρα πολλές εποχές). Αυτή έχει πάλι ως αποτέλεσμα την πολύ καλή μάθηση του Training Set, χωρίς όμως αυτό να συνεπάγεται δυνατότητα γενίκευσης στο Test Set.




Σχήμα 9: Underfitting και Overfitting
Στην πρώτη εικόνα φαίνεται η αδυναμία του συστήματος να διακρίνει επαρκώς μεταξύ θετικών και αρνητικών δειγμάτων. Στη συγκεκριμένη περίπτωση οφείλεται στη γραμμικότητα του δικτύου, που δε μπορεί να προσομοιώσει τα μη-γραμμικά δεδομένα. Στην Τρίτη εικόνα, αντίθετα, βλέπουμε ένα δίκτυο που έχει «μάθει» έναν πολύ ιδιαίτερο και εξαρτώμενο από τα training data τρόπο διαχωρισμού των δειγμάτων σε θετικά και αρνητικά. Παρότι το σφάλμα εκπαίδευσης εδώ να είναι μηδενικό, το δίκτυο αυτό δε θα μπορεί κατά πάσα πιθανότητα να γενικευτεί, αφού η διαχωριστική γραμμή έχει προσαρμοστεί τέλεια στα συγκεκριμένα δεδομένα. Μια καλύτερη λύση φαίνεται να δίνεται στη δεύτερη εικόνα, όπου υπάρχει μεν κάποιο σφάλμα εκπαίδευσης, αλλά αυτό είναι μικρό και αναμένουμε ότι εξίσου μικρό θα είναι και το σφάλμα πάνω στα test data. Πηγή [20]




Σχήμα 10: Underfitting και Overfitting 2
Αντίστοιχα με την προηγούμενη εικόνα, βλέπουμε και εδώ τρεις καμπύλες που προσπαθούν να προσαρμοστούν στα training data. Στην πρώτη βλέπουμε πάλι το Underfitting, στην δεύτερη μία ομαλή εκπάιδευση και στην Τρίτη το Overrfitting, αφού εκεί η καμπύλη είναι ένα πολυώνυμο μεγάλου βαθμού, μεγαλύτερου από ότι απαιτείται για το σωστό fitting των δεδομένων. Πηγή [9]

Έχουν προταθεί αρκετές μέθοδοι για regularization (Κανονικοποίηση). Οι σημαντικότερες είναι η L2 regularization, η τεχνική του Dropout και η τεχνική Early Stopping.

**L2 Regularization**

Η μέθοδος[9] συνίσταται ουσιαστικά στην προσθήκη ενός επιπλέον όρου στη συνάρτηση κόστους. Έστω

$$J(w; Q, y) \tag{22}$$

η συνάρτηση κόστους, όπου $\theta$ είναι το διάνυσμα των βαρών, $Q$ το διάνυσμα με τα χαρακτηριστικά (features) των εισόδων και οι $y$ το διάνυσμα των (αναμενόμενων) εξόδων.

Η συνάρτηση κόστους μετά την προσθήκη του όρου κανονικοποίησης γίνεται:

$$J'(w; Q, y) = J(w; Q, y) + \lambda\Omega(w) \tag{23}$$

όπου $\lambda$ είναι μία παράμετρος που καθορίζει τη βαρύτητα που θα έχει ο όρος κανονικοποίησης.

Βλέπουμε ότι ο όρος κανονικοποίησης $\Omega$ εξαρτάται μόνο από τις τιμές των βαρών $\theta$ και όχι από τις εσόδους και τις εξόδους.

Στην – πλέον συνήθη – περίπτωση που ο όρος $\Omega$ είναι δευτέρου βαθμού, έχουμε

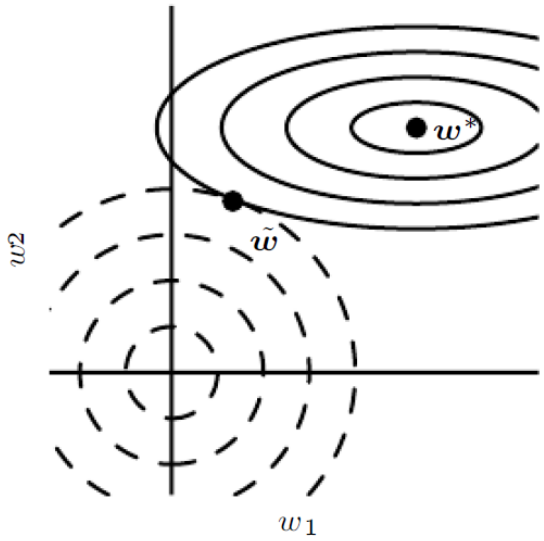
$$\Omega(w) = \frac{1}{2}W^T * W \tag{24}$$

και

$$J'(w; Q, y) = J(w; Q, y) + \frac{\lambda}{2}W^T * W \tag{25}$$

Η επίδραση που έχει ο όρος κανονικοποίησης είναι ότι δεν αφήνει τα βάρη ω να πάρουν την τιμή στην οποία ελαχιστοποιείται η Συνάρτηση Κόστους αλλά μία τιμή αρκετά κοντά – ανάλογα με τη σταθερά $\lambda$ – σε αυτή. Ως αποτέλεσμα, αποφεύγεται σε μεγάλο βαθμό το overfitting.

Μάλιστα, όπως φαίνεται και αναλύεται στην ακόλουθη εικόνα 11, η επίδραση του όρου κανονικοποίησης είναι περισσότερο έντονη στις διαστάσεις στις οποίες οι μεταβολές επηρεάζουν πολύ την τιμή της συνάρτηση κόστους.




Σχήμα 11: Η επίδραση της κανονικοποίησης regularization στην ελαχιστοποίηση της συνάρτησης κόστους.

Γραφική απεικόνιση της επίδρασης της L2 Regularization στη βέλτιστη τιμή των βαρών $w^*$, σε ένα μοντέλο με 2 βάρη $w_1$, $w_2$. Οι ελλείψεις με τη συνεχή γραμμή είναι οι ισοδυναμικές γραμμές της συνάρτησης κόστους – χωρίς να λαμβάνεται υπόψη ο όρος κανονικοποίησης, ενώ οι διακεκομμένες γραμμές είναι οι ισοδυναμικές του όρου $L2$. Στην οριζόντια διάσταση, η τιμή της συνάρτησης κόστους δε μεταβάλλεται έντονα όταν μετακινούμαστε κοντά στο $w^*$. Βλέπουμε ότι στη διάσταση αυτή, η επίδραση της κανονικοποίησης είναι έντονη, καθώς το $w$ απέχει οριζόντια πολύ από το $w^*$. Αντίθετα, στη δεύτερη διάσταση, η συνάρτηση κόστους είναι πολύ πιο ευαίσθητη. Εδώ, η επίδραση της κανονικοποίησης είναι πιο μικρή. Έτσι, η L2 Regularization εμποδίζει μεν το overfitting, αλλά δε στερεί μεγάλη ποσότητα πληροφορίας από το δίκτυο. Πηγή [9]

**Η μέθοδος Early stopping**

Μία απλή αλλά αποτελεσματική τεχνική αποφυγής του Overfitting είναι το Early Stopping.

Κατά το Overfitting, αυτό που παρατηρείται είναι από τη μία πλευρά να μειώνεται συνεχώς το σφάλμα εκπαίδευσης (training error) και από την άλλη να μη μειώνεται και από ένα σημείο και μετά να αυξάνεται το σφάλμα ελέγχου (test error).

Ελέγχοντας την τιμή του test error, η τεχνική του Early stopping μας λέει να σταματήσουμε την εκπαίδευση όταν αυτή αρχίσει να αυξάνεται.

Μία παράμετρος που εισάγεται στο σύστημα είναι ο «χρόνος» εκπαίδευσης – δηλαδή το πλήθος των επαναλήψεων ή οι εποχές - μέχρι να γίνει ο έλεγχος του test error. Αν T αυτό το πλήθος, τότε το υπολογιστικό κόστος της τεχνικής αυτής είναι μικρό, συνίσταται μόνο στο τρέξιμο του test set και την παραγωγή προβλέψεων μία φορά ανά T επαναλήψεις. Αυτό, μάλιστα, μπορεί να γίνει και ταυτόχρονα με την εκπαίδευση για εξοικονόμηση χρόνου.

**Η τεχνική του Dropout**

Τέλος, μία ακόμη υπολογιστικά φθηνή και αποτελεσματική τεχνική είναι το Dropout. Αυτή εισάγει μία σταθερά $p$ στο διάστημα $[0, 1]$, η οποία καθορίζει την πιθανότητα με την οποία η έξοδος κάθε νευρώνα θα τίθεται από το σύστημα σε 0. Ισοδύναμα, καθορίζει την πιθανότητα με την οποία ένας νευρώνας θα αποκόπτεται από το δίκτυο.

Σε μία μαθηματική διατύπωση, μπορούμε να θεωρήσουμε μία μάσκα $\mu$, μήκους ίσου με το πλήθος των βαρών του νευρωνικού δικτύου μας. Έτσι, η συνάρτηση κόστους γίνεται $J(\theta, \mu)$, εξαρτάται δηλαδή από την τιμή των βαρών και από τη μάσκα που θα επιλέξουμε.
Η εκπαίδευση του δικτύου συνίσταται τώρα στην ελαχιστοποίηση της μέσης τιμής $E[J(\theta, \mu)]$. Ο μέσος όρος περιέχει εκθετικά πολλούς όρους, μπορούμε όμως να υπολογίσουμε μία καλή προσέγγισή του δειγματολειπτώντας το $\mu$.
Σημειώνεται ότι στα Νευρωνικά Δίκτυα που περιέχουν ανάδραση (π.χ. RNN ) το Dropout εφαρμόζεται με τον ίδιο ακριβώς με τον ίδιο τρόπο και στις αναδράσεις.

### 2.1.14 Μετρικές και Αξιολόγηση Αποτελεσμάτων στα Νευρωνικά Δίκτυα

Οι μετρικές που χρησιμοποιήθηκαν στη Διπλωματική εργασία είναι η **Ακρίβεια** και το **F-score** (βλ. Εξίσωση 28).

Επίσης, αναφορά γίνεται στα Precision και Recall. Έστω πρόβλημα ταξινόμησης με 2 κλάσεις, $C_+$ και $C_-$. Τότε:

> Το **Precision** δίνει την πιθανότητα, δεδομένου ότι το σύστημα έχει αποκριθεί ότι ένα δείγμα ανήκει στη $C_+$, να ανήκει πράγματι εκεί. Δηλαδή
>
> $$Precision = P = \frac{\#TP}{\#TP + \#FP} \tag{26}$$
>
> Το **Recall** δίνει την πιθανότητα, δεδομένου ενός θετικού δείγματος (δηλ. ενός δείγματος που ανήκει στη $C_+$), το σύστημα να βρει ότι ανήκει σε αυτή. Δηλαδή
>
> $$Recall = R = \frac{\#TP}{\#TP + \#FN} \tag{27}$$

Έτσι, έχουμε ότι

$$Fscore = F = 2\frac{PR}{P + R} \tag{28}$$

## 2.2 Ασφάλεια Δικτύων (Network Security) και Παρακολούθηση Δικτυακής Κίνησης (Monitoring)

### 2.2.1 Είδη Δικτυακών Επιθέσεων

Αναφέρονται εδώ κάποια βασικά είδη δικτυακών επιθέσεων.

**Ογκομετρικές Επιθέσεις (Volumetric Attacks)**
Έχουν ως στόχο την εξάντληση του εύρους ζώνης που είναι διαθέσιμο εντός του δικτύου του θύματος ή του εύρους ζώνης του δικτύου μεταξύ του θύματος και του υπόλοιπου Ίντερνετ, στέλνοντας προς το θύμα έναν πολύ μεγάλο όγκο κίνησης.

**Επιθέσεις Σύνδεσης (TCP Connection Attacks)**
Εδώ γίνεται προσπάθεια να χρησιμοποιηθούν και να εξαντληθούν όλοι οι πόροι μιας συσκευής υποδομής, όπως για παράδειγμα ένας εξυπηρετητής. Η διαφορά με το προηγούμενο είδος επίθσης έγκυται στον ακριβή στόχο. Εδώ η επίθεση δε γίνεται στο εύρος ζώνης του δικτύου αλλά στους φυσικούς πόρους κάποιου μηχανήματος.

**Επιθέσεις στο DNS**
Αποτελούν έναν πολύ συνηθισμένο και πολύ αποτελεσματικό τύπο επιθέσεων, δεδομένης της σημασίας του συστήματος DNS για τη λειτουργία του διαδικτύου. Οι επιθέσεις στο DNS μπορεί να έχουν πολλές μορφές.

- Πρώτον, υπάρχει το λεγόμενο DNS Poisoning, κατά το οποίο ένας εκτεθειμένος (compromised) εξυπηρετητής DNS, ή ακόμα και μία DNS Cache, απαντά λάθος σε ερωτήματα των χρηστών. Ως αποτέλεσμα, η πρόσβαση σε υπηρεσίες και σελίδες μπορεί να καταστεί αδύνατη ή οι χρήστες ενδέχεται να οδηγηθούν σε διαφορετική σελίδα από αυτή που ζήτησαν.
- Δεύτερον, υπάρχει η λεγόμενη Ενίσχυση μέσω DNS, που στην πραγματικότητα δε στοχεύει στο DNS αλλά το εκμεταλεύεται, ώστε να χτυπήσει κάποιο άλλο θύμα. Η βάση της επίθεσης έγκυται στο να στείλουμε μεγάλο πλήθος ερωτημάτων σε εξυπηρετητές DNS, ερωτημάτων που είναι τέτοια ώστε να παράγουν μεγάλου όγκου απαντήσεις. Αν τα ερωτήματα αυτά αποσταλούν στον DNS Server με πλαστή spoofed διεύθυνση προέλευσης, με διεύθυνση προέλευσης ουσιαστικά αυτή του θύματος, τότε το θύμα θα δεχτεί όλον τον όγκο των απαντήσεων.
- Επιθέσεις Άρνησης Υπηρεσίας εναντίον των DNS Server. Πρόκειται για ογκομετρικές επιθέσεις, αλλά με στόχο όλο το DNS σύστημα μια περιοχής. Μία επιτυχημένη επίθεση αυτού του τύπου θα σημαίνει αυτόματα την αδυναμία λειτουργίας για μεγάλο πλήθος υπηρεσιών.
  Ενδεικτικά αναφέρουμε την επίθεση που σημειώθηκε στις 21 Οκτωβρίου 2016 στις δυτικές ΗΠΑ και είχε ως θύμα τη Dyn[21], εταιρία παροχής υποδομών Ίντερνετ και υπηρεσίας DNS. Η επίθεση

προήλθε από περίπου 100,000 εκτεθειμένες συσκευές IoT (Internet of Things) και έφτασε σε όγκο το 1 Tbps, επηρεάζοντας τη λειτουργία πολλών μεγάλων υπηρεσιών, όπως BBC, CNN, Amazon, Twitter, Spotify κλπ. [22, 23, 24, 25]

Πηγές [26, 27]

**Ανίχνευση ανοιχτών θυρών (Port Scanning)**

Όπως φαίνεται και από το όνομά της, ο επιτειθέμενος στέλνει προς το θύμα TCP ή και UDP πακέτα, εξετάζει την απάντηση που λαμβάνει σε κάθε περίπτωση και καταλαβαίνει έτσι ποιες θύρες του θύματος είναι ανοιχτές και πιθανώς και ποιες εφαρμογές ακούνε σε κάθε μία. Αν, για παράδειγμα, σταλεί ένα TCP πακέτο με αναμμένο το flag SYN προς κάποια θύρα και ως απάντηση λάβουμε πακέτο με τα flags RST, ACK να έχουν τεθεί, αυτό αποτελεί ένδειξη κλειστής θύρας. Αν δε λάβουμε καθόλου απάντηση ή λάβουμε ως απάντηση μήνυμα ICMP Destination Unreachable, τότε μπορούμε να συμπεράνουμε ότι πρόκειται για φιλτραρισμένη θύρα. Αποτελεί έναν κλασικό τύπο επίθεσης που δεν είναι ιδιαίτερα δύσκολο να αναγνωριστεί.

### 2.2.2 Επιθέσεις Άρνησης Υπηρεσίας (DoS Attacks, Denial of Service)

Οι επιθέσεις Άρνησης Υπηρεσίας (DoS, Denial of Service) εντάσσονται κυρίως στις Ογκομετρικές Επιθέσεις. Συνίστανται στην προσπάθεια να βγάλουμε μία υπηρεσία εκτός λειτουργίας κατακλύζοντάς την με κίνηση. Η κίνηση αυτή μοιάζει με ομαλή κίνηση, δηλαδή κίνηση καλόβουλων χρηστών, ωστόσο ο συνολικός της όγκος καθιστά τον εξυπηρετητή αδύνατο να την επεξεργαστεί και να απαντήσει. Έτσι, ο εξυπηρετητής δεν αποκρίνεται στην κακόβουλη αλλά ούτε και στην καλόβουλη κίνηση.
Μία ειδική περίπτωση επίθεσης Αρνησης Υπηρεσίας είναι η Κατανεμημένη Αρνησης Υπηρεσίας (Distributed Denial Of Service, DDoS). Σε αυτή την περίπτωση, η μεγάλου όγκου κίνηση προς τον εξυπηρετητή στέλνεται από κακόβουλο λογισμικό που έχει εγκατασταθεί σε πολλά κατανεμημένα μηχανήματα. Το πλήθος των εκτεθειμένων μηχανημάτων είναι συνήθως μεγάλο - αρκετές χιλιάδες, πράγμα που την καθιστά ακόμη πιο δύσκολο να ανιχνευτεί και να αντιμετωπιστεί.
Κάποιες ειδικές περιπτώσεις DDos επιθέσεων είναι οι εξής.

**Πλημμύρα UDP (UDP Flood)**

Αυτή η περίπτωση χρησιμοποιεί το πρωτόκολλο UDP , που είναι πρωτόκολλο Χωρίς-Σύνδεση (Sessionless). Ένα τεράστιο πλήθος από από πακέτα στέλνονται προς τυχαίες θύρες του θύματος, αναγκάζοντάς το να απαντά σε αυτά με μηνύματα ICMP . Έτσι, εκτός από το εύρος ζώνης, έχει ως αποτέλεσμα και την εξάντληση πόρων του θύματος.
Η αντιμετώπιση γίνεται συνήθως σε επίπεδο Firewall, αλλά απέναντι σε έναν τεράστιο όγκο πακέτων η εξάντληση τους εύρους ζώνης δε μπορεί να αποφευχθεί

**Πλημμύρα ICMP (ICMP Flood)**

Παρόμοια είναι και η λογική της πλημμύρας ICMP . Πακέτα ICMP κατακλύζουν το θύμα, συνήθως σε πολύ γρήγορο ρυθμό, χωρίς να περιμένουμε να απαντήσει. Αν το θύμα απαντάει σε αυτά τότε καταναλώνεται ακόμα περισσότερο εύρος ζώνης, τόσο εισερχόμενο όσο εξερχόμενο.

Η αντιμετώπιση αυτής της κατηγορίας επίθεσης είναι πιο εύκολη. Μία συνήθης λύση είναι να μην επιτρέπονται καθόλου τα ICMP μηνύματα.

**Πλημμύρα SYN (SYN Flood)**

Η επίθεση αυτή εκμεταλλεύεται ένα βασικό χαρακτηριστικό του πρωτοκόλλου TCP , την Τριπλή Χειραψία (3-Way Handshake). Σύμφωνα με αυτή, για τη δημιουργία μίας TCP σύνδεσης, ο πελάτης (client) στέλνει προς τον εξυπηρετητή ένα TCP πακέτο με ενεργή την σημαία SYN. Ο εξυπηρετητής πρέπει να απαντήσει είτε με ACK, RST, στην περίπτωση που δε θέλει ή δε μπορεί να δημιουργήσει τη σύνδεση, ή με SYN, ACK, που είναι και το σύνηθες σενάριο. Τέλος, ο Client στέλνει ένα τρίτο πακέτο με ενεργό το Flag ACK. Η τριπλή αυτή χειραψία φαίνεται και στο Σχήμα 12. Στο σημείο αυτό η σύνδεση έχει δημιουργηθεί και αρχίζει η ανταλλαγή των πακέτων.

Ο εξυπηρετητής πρέπει να περιμένει κάποιο χρονικό διάστημα μέχρι να λάβει το τρίτο και τελευταίο πακέτο, αυτό με το Flag ACK, άρα πρέπει να δεσμεύεσει χώρο στη μνήμη τους για την πληροφορία αυτή. Αν το τρίτο πακέτο δε φτάσει ποτέ, τότε μιλάμε για μία Ημι-ανοιχτή σύνδεση (Half-Open Connection). Δημιουργώντας πολλές ημι-ανοιχτές συνδέσεις, ο επιτιθέμενος μπορεί να εξαντλήσει τους πόρους του θύματος.

**Επίθεση στο NTP**

Σε αυτό το είδος DDoS ο επιτειθέμενος χρησιμοποιεί το πρωτόκολλο NTP (Network Time Protocol) για να κατακλείσει το θύμα με κίνηση UDP . Το NTP είναι ένα από τα παλιότερα δικτυακά πρωτόκολλα, που χρησιμοποιείται για συγχρονισμό των ρολογιών μεταξύ υπολογιστών, ενώ προσφέρει επίσης και δυνατότητες monitoring, μέσω π.χ. της εντολής *monlist*, που επιστρέφει μία λίστα με τους τελευταίους 600 υπολογιστές που συνδέθηκαν στον εξυπηρετητή. Στέλνοντας αλλεπάλληλα ΄γετ μονλιστ΄ αιτήματα και χρησιμοποιώντας ως διεύθυνση αποστολέα τη διεύθυνση του θύματος (IP Spoofing) στα αιτήματα αυτά, το θύμα θα δεχτεί μία τεράστια ανακλόμενη κίνηση.

Όπως και οι υπόλοιπες DDoS επιθέσεις, η επίθεση στο NTP είναι δύσκολο να ανιμετωπιστεί, γιατί τα πακέτα που λαμβάνονται μιάζουν με νόμιμα και καλόβουλα πακέτα. Μία κίνηση που έγινε για την αντιμετώπισή της ήταν η επαναθεώρηση της εντολής ΄μονλιστ΄.

**Πλημμύρα HTTP (HTTP Flood)**

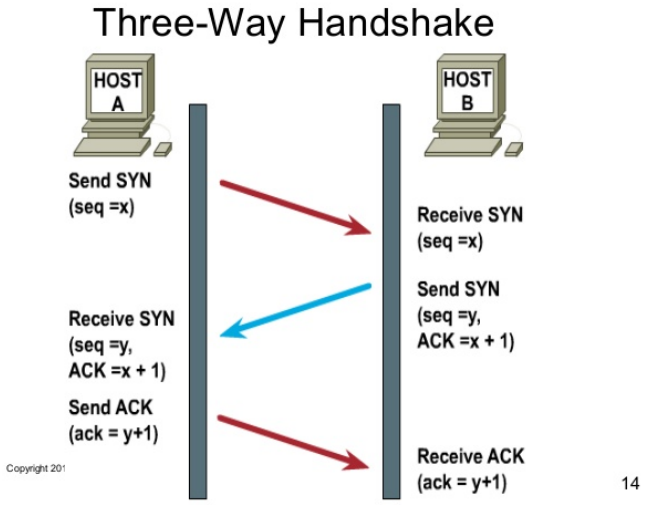
Εδώ ο χρήστης χρησιμοιποιεί αιτήματα HTTP GET ή POST, πακέτα που μοιάζουν νόμιμα, χωρίς πλαστές IP διευθύνσεις. Εκμεταλλεύται τα χαρακτηριστικά της συγκεκριμένης εφαρμογής στην οποία επιτείθεται, άρα απαιτεί καλή γνώση της εφαρμογής και του τρόπου λειτουργίας του εξυπηρετητή.

Αυτό το είδος είναι μάλλον το δυσκολότερο ως προς την αντιμετώπισή, αφού η κίνηση έχει εξ΄ ολοκλήρου

τη μορφή καλοπροαίρετης κίνησης. Κάποιες μέθοδοι που χρησιμοποιούνται είναι η φήμη (reputation) των διευθύνσεων IP και η αποκοποή των διευθύνσεων που έχουν κακή φήμη, λύσεις μέσω JavaScript κλπ.

**Ενίσχυση μέσω DNS (DNS Amplification)**
Η ενίσχυση μέσω DNS, που αναφέραμε στην προηγούμενη Ενότητα, αποτελεί μία συνήθη και καταστροφική DDoS επίθεση.

**Zero-Day επιθέσεις**
Τέλος, οι Zero-Day επιθέσεις εκμεταλλεύονται αδυναμίες ενός συστήματος, οι οποίες έχουν προκύψει συνήθως μετά από κάποια αναβάθμιση λογισμικού και δεν έχουν προλάβει να αναγνωριστούν από τους διαχειριστές του και να διορθωθούν.
Πηγές [28, 4]




Σχήμα 12: Η τριπλή χειραψία (3-Way Handshake) του πρωτοκόλλου TCP .
Πηγή [29]

### 2.2.3 Το Πρωτόκολλο Netflow

Ορισμός Flow : Ένα σύνολο IP πακέτων που διέρχονται από ένα σημείο παρατήρησης σε μία ορισμένη χρονική περίοδο, τέτοιο ώστε όλα τα πακέτα που ανήκουν σε μία συγκεκριμένη Flow να έχουν ένα σύνολο κοινών ιδιοτήτων.

Βασικά flow-based πρωτόκολλα για monitoring δικτύων είναι τα NetFlow [30] και IPFIX (IP Flow Information eXport) [31].

Το πλεονέκτημα του flow-based μονιτορινγ έχει να κάνει με την κλιμακωσιμότητα, καθώς δε γίνεται ανάλυση κάθε πακέτου ξεχωριστά αλλά αναλύεται συνολικά και αθροιστικά όλη η Flow . Ως αποτέλεσμα, πολύ μεγαλύτερος όγκος κίνησης μπορεί να ελεγχθεί, ενώ μειώνεται σημαντικά και ο απαιτούμενος χώρος αποθήκευσης, σε σχέση με την περίπτωση αποθήκευσης και ελέγχου ολόκληρων των IP πακέτων. Επίσης, η παρακολούθηση της κίνησης με χρήση flow-based πρωτοκόλλων είναι σχεδόν πραγματικού χρόνου με την έννοια ότι μία ροή (flow) στέλνεται στο συλλέκτη (collector) αμέσως μόλις λήξει [32].

Στη συνέχεια παρουσιάζονται και αναλύονται οι βασικοί όροι και ο τρόπος λειτουργίας του NetFlow .

**Σημείο Παρατήρησης (Observation Point)**
Το σημείο στο ΙΠ δίκτυο όπου τα διερχόμενα πακέτα παρατηρούνται.

**Ροή IP (IP Flow ή απλά Flow )**
Σύνολο IP πακέτων που περνούν από ένα Observation Point κατά τη διάρκεια ενός ορισμένου διαστήματος. Όλα τα πακέτα που ανήκουν σε μία συγκεκριμένη Flow έχουν ένα κοινό σύνολο ιδιοτήτων, τις οποίες έχουμε συμπεράνει από τα δεδομένα των διερχομένων από το Observation Point πακέτων καθώς επίσης και από τον τρόπο διαχείρισης των πακέτων στο Observation Point.

**Flow Record**
Μία Flow Record παρέχει πληροφορίες για μία Ροή IP , όπως αυτή παρατηρήθηκε σε ένα Observation Point. Στο NetFlow v.9 ορίζονται 4 είδη Εγγραφών:

1. **Template Record**, που ορίζει τη δομή και την ερμηνεία των πεδίων σε μία Data Record. Βλ. εικόνα 14.
2. **Data Record**, που περιέχει τις τιμές για όλες τις παραμέτρους που έχουν οριστεί στην αντίστοιχη Template Record. Βλ. εικόνα 15.
3. **Options Template Record**, που ορίζει τη δομή και την ερμηνεία κάποιων έξτρα πεδίων που περιλαμβάνονται στις Options Data Record, καθώς και την εμβέλεια μέσα στην οποία η Options Data Record ορίζεται. Βλ. εικόνα 16.

4. **Options Data Record**, που περιέχει τιμές για όλα τα πεδία που έχουν οριστεί στην αντίστοιχη Options Template Record. Βλ. εικόνα 15.

**Εξαγωγέας NetFlow (NetFlow Exporter)**
Μία συσκευή (π.χ. Router) με ενεργοποιημένες τις υπηρεσίες NetFlow , που καταγράφει τα πακέτα που εισέρχονται στο Observation Point και δημιουργεί τις Flows. Όλη η πληροφορία που δημιουργείται στον Exporter στέλνεται με τη μορφή Flow Records στον Collector.

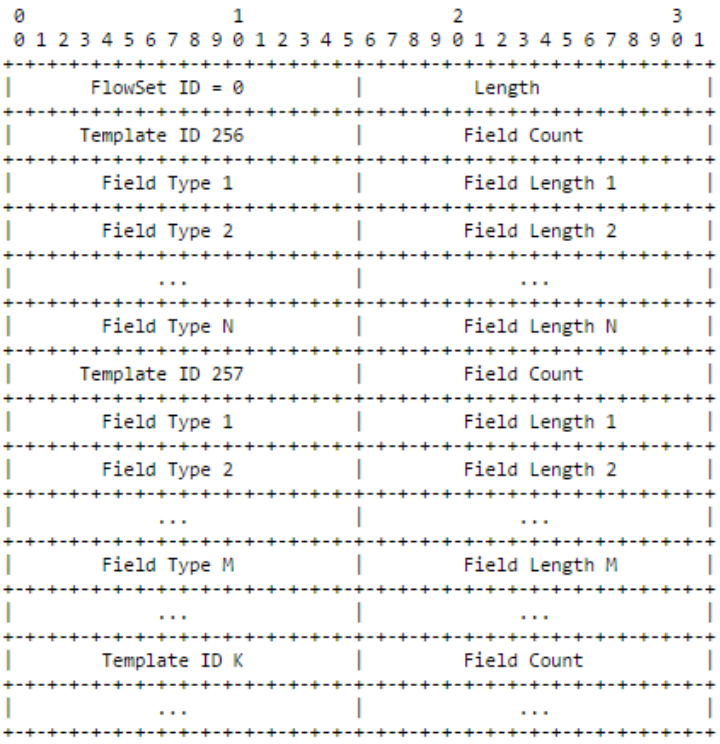
**Συλλέκτης NetFlow (NetFlow Collector)**
Ο Συλλέκτης λαμβάνει τις Flow Records από έναν ή περισσότερους Exporters και τις επεξεργάζεται με κάποιον τρόπο.

**Export Packet**
Το πακέτο που στέλνεται από τον Εξαγωγέα (Exporter) προς το Συλλέκτη (Collector) και μεταφέρει τις Εγγραφές NetFlow (Flow Records) που έχουν καταγραφεί από τον Εξαγωγέα. Περιέχει την **Packet Header**, που αποτελεί πάντα το πρώτο μέρος του Export Packet και περιέχει βασικές πληροφορίες, όπως την έκδοση του NetFlow και το πλήθος των Flow Records στο πακέτο. Βλ. και εικόνα 13. Περιέχει επίσης ένα πλήθος από FlowSets.

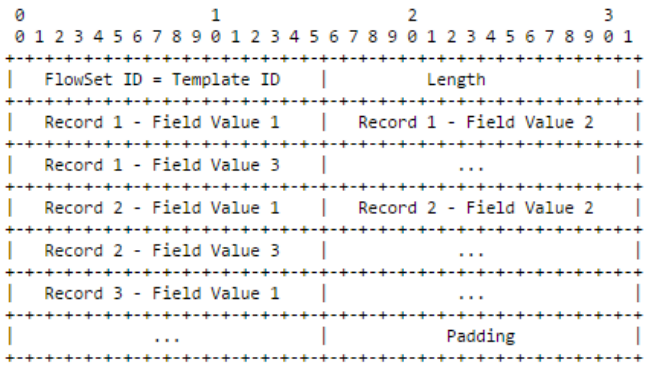
**FlowSet**
Ο όρος FlowSet περιγράφει ένα σύνολο από Flow Records που έχουν παρόμοια δομή. Για λόγους ταχύτητας και εξοικονόμησης εύρους ζώνης, ένα σύνολο από Flow Records μπορούν να σχηματίσουν μία FlowSet και να αποσταλούν όλες μαζί στον Collector. Υπάρχουν τρις τύποι FlowSet:

1. **Template FlowSet**, που είναι ένα σύνολο από Template Records. Βλ. εικόνα 14
2. **Options Template FlowSet**, που είναι ένα σύνολο από Options Template Records. Βλ. εικόνα 16
3. **Data FlowSet**, που είναι ένα σύνολο από Data Records ή Options Data Records, που όμως σχετίζονται με την ίδια Template FlowSet, η οποία έχει ήδη οριστεί και αποσταλεί με Export Packet στον Collector. Βλ. εικόνα 15

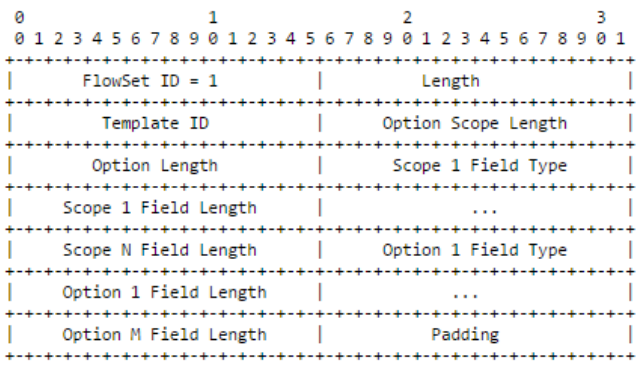

Σχήμα 13: Ένα πακέτο Export
Είναι το πακέτο που στέλνεται από τον Εξαγωγέα NetFlow (NetFlow Exporter) προς το Συλλέκτη NetFlow (NetFlow Collector). Στο σχήμα φαίνεται με σαφήνεια η απαίτηση να στέλνονται πρώτα τα Templates και μετά οι εγγραφές NetFlow που σχετίζονται με το Template αυτό. Πηγή [33]




Σχήμα 14: Ένα σύνολο από Template Records σχηματίζουν ένα Template Flowset.
Στο σχήμα διακρίνονται K Template Records, καθεμία από τις οποίες περιέχει ένα διαφορετικό πλήθος Πεδίων (ή Χαρακτηριστικών, Fields ή Attributes). Η πρώτη, για παράδειγμα, περιέχει N Πεδία και η δεύτερη M. Αυτό που καθορίζεται για κάθε πεδίο είναι ο τύπος του (αριθμητικό, αλφαριθμητικό, κ.λ.π) και το μήκος του. Πηγή [33]




Σχήμα 15: Ένα σύνολο από Data Records σχηματίζουν ένα Data Flowset.
Στην αρχή ενός Data FlowSet υπάρχει πάντα το FlowSet ID, που συσχετίζει το Data FlowSet με μία Template Flowset, η οποία βέβαια πρέπει να έχει προηγουμένως σταλεί στο συλλέκτη. Στη συνέχεια περιέχει τις τιμές όλων των πεδίων για όλες τις Template Records που έχουν οριστεί μέσα στην αντίστοιχη Template FlowSet. Τα πεδία αποκωδικοποιούνται με βάση τον ορισμό τους (τύπος και μήκος). Πηγή [30]




Σχήμα 16: Ένα σύνολο από Options Template Records σχηματίζουν ένα Options Template Flowset. Το Options Template FlowSet χρησιμοποιείται σε περιπτώσεις που απατείται συλλογή δεδομένων που δε δχετίζονται άμεσα με τις IP Flows, αλλά με τη διεργασία που συλλέγει αυτές τις Flows, όπως π.χ. ο ρυθμός και η μέθοδος δειγματοληψίας που χρησιμοποιούνται (αν χρησιμοποιούνται). Όπως και οι Template Records, έτσι και οι Options Template Records πρέπει να αποσταλούν στο Συλλέκτη πριν από τις αντίστοιχες Data Records. Πηγή [33]

### Λήξη Ροής (Flow Expiration) και Χρονικά όρια απόστολής των Ροών

Μία Ροή θεωρείται ότι έχει ολοκληρωθεί ή λήξει όταν ο Exporter αντιληφθεί το τέλος της (π.χ. FIN ή RST σε μία TCP σύνδεση). Στην περίπτωση αυτή είναι διαθέσιμη για απόστολή προς το Συλλέκτη με το επόμενο Export Packet.

Ωστόσο, μία Ροή μπορεί - και πρέπει - να σταλεί προς το Συλλέκτη και στις ακόλουθες περιπτώσεις.

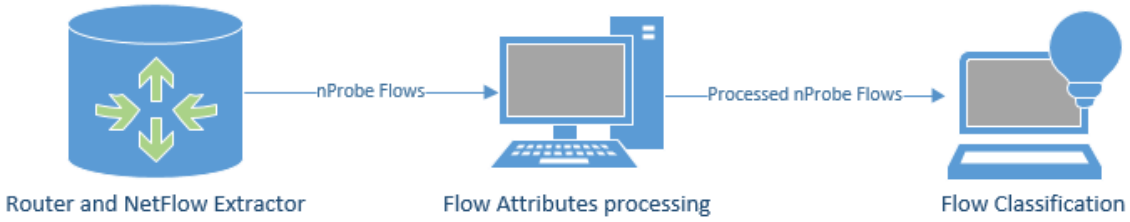
1. Αν είναι ανενεργή (δηλαδή δεν έχει ληφθεί κανένα πακέτο που ανήκει σε αυτή) για ένα ορισμένο χρονικό διάστημα. Σύμφωνα με το RFC 3954 [30] του NetFlow , αυτό το διάστημα, που συνήθως ονομάζεται **Idle Timeout ή Active Timeout**, πρέπει να είναι ρυθμίσιμο από το χρήστη με ελάχιστη δυνατή τιμή 0, για άμεση λήξη. Στο nProbe , που αποτελεί υπολοποίηση του NetFlow , όπως φαίνεται και στον Οδηγό Χρήσης [34], το Idle Timeout είναι ρυθμίσιμο με Default τιμή τα 30sec.

2. Ακόμα και οι ενεργές ροές πρέπει να στέλνονται σε τακτά χρονικά διαστήματα στο Συλλέκτη. Για το λόγο αυτό, το RFC 3954 [30] ορίζει και ένα δεύτερο χρονόμετρο, που συνήθως ονομάζεται **Lifetime Timeout ή Hard Timeout** και πρέπει επίσης να είναι διαθέσιμο προς ρύθμιση από το χρήστη. Στο nProbe είναι προκαθορισμένο στα 120sec.

3. Τέλος, σε περίπτωση που ο Εξαγωγέας Ροών αντιμετωπίσει πρόβλημα λειτουργίας, οι Ροέςς ενδέχεται να λήξουν πρώιμα και να αποσταλούν άμεσα στο Συλλέκτη.

# 3 Αρχιτεκτονική του Συστήματος

Στο ακόλουθο σχήμα φαίνεται παραστατικά ο τρόπος συλλογής των δεδομένων από το Δρομολογητή, η εξαγωγή των Χαρακτηριστικών Κίνησης, η επεξεργασία τους και η τροφοδότηση τους Νευρωνικού δικτύου.




Σχήμα 17: Μία γενική εικόνα του τρόπου λειτουργίας του προτεινόμενου συστήματος.
Όλη η πληροφορία σχετικά με την κίνηση συλλέγεται με τη βοήθεια του nProbe [34], όπως αυτό έχει υλοποιηθεί στο πρωτόκολλο NetFlow v9[30][32] της Cisco. Ο τρόπος λειτουργίας του NetFlow καθώς και τα χαρακτηριστικά της κίνησης που εξάγονται στο σημείο αυτό περιγράφονται στην Ενότητα 2.2.3. Στη συνέχεια, τα χαρακτηριστικά των Ροών που έχουν συλλεχθεί από το NetFlow υφίστανται μία τροποποίηση, ώστε να γίνει δυνατή η τροφοδότησή τους στο Νευρωνικό Δίκτυο. Αυτή η διαδικασία περιγράφεται στην Ενότητα 4.3.
Τέλος, το Flow Classification, το τελευταίο στάδιο στο σχήμα, υλοποιήθηκε με διάφορα είδη Νευρωνικών, όπως αναλύεται στην επόμενη Ενότητα.

# 4 Υλοποίηση

## 4.1 Είδη Επίθεσης Που Αναγνωρίστηκαν

Στην Υποενότητα αυτή περιγράφονται τα είδη επίθεσης (μεταξύ αυτών που αναφέρθηκαν στην Ενότητα 2.2.1 που που ελέγχθηκαν και αναγνωρίστηκαν στα πλαίσια της παρούσας Διπλωματικής εργασίας, καθώς επίσης και η πηγή κάθε επίθεσης (αν ήταν πραγματική ή τεχνητή).
Για τον πειραματισμό και τον έλεγχο της επίδοσης των διαφόρων δικτύων χρησιμοποιήθηκε κίνηση προερχόμενη από διάφορες πηγές. Ως θύμα θεωρήθηκε η διεύθυνση 147.102.222.210, που βρίσκεται εντός του υποδισκτύου 147.102.0.0\16. Όπως θα περιγραφεί στα αμέσως επόμενα, η συλλεχθείσα κίνηση προσαρμόστηκε ώστε το θύμα να έχει πάντα αυτή την IP .

**Ομαλή (Legitimate) κίνηση**
Πραγματική κίνηση από το εσωτερικό δίκτυο του ΕΜΠ. Καταγράφηκε με Port Mirroring πάνω σε έναν Μεταγωγέα (Switch). Η κίνηση προέρχεται από και προορίζεται για ένα μεγάλο εύρος διευθύνσεων, εντός και εκτός του υποδικτύου του ΕΜΠ. Ένα σημαντικό μέρος της κίνηση κατευθύνεται προς την 147.102.222.210, ωστόσο διατηρήθηκε και η κίνηση προς όλες τις άλλες διευθύνσεις.

**Επίθεση πλημμύρας UDP (UDP Flood)**
Για τη δημιουργία των αρχείων καταγραφής (traces) της UDP Flood επίθεσης χρησιμοποιήθηκε το εργαλείο Scapy[2].
Με τη βοήθεια του εργαλείου VirtualBox ςιτειρτυαλβοξ, στήθηκε ένας Ubuntu Server. Σε αυτόν τοποθετήθηκαν κατάλληλες εγγραφές στο τοίχος προστασίας (Firewall) iptables[35] καθώς επίσης ανοίχτηκαν κάποιες πόρτες μέσω του ufw[36]. Στη συνέχεια, από ένα δεύτερο VM στάλθηκαν πακέτα UDP προς διάφορες θύρες του Server, τόσο ανοιχτές, όσο κλειστές και φιλτραρισμένες. Όλη αυτή η κίνηση καταγράφηκε σε πςαπ αρχείο από το Wireshark[37] που έτρεχε στο φιλοξενούν μηχάνημα. Σημειώνεται εδώ ότι, όπως αναφέρεται στην Ενότητα 4.3 η διευθύνσεις IP προέλευσης και προορισμού δε δίνονται ως είσοδοι στο νευρωνικό δίκτυο. Ωστόσο, δίνονται τα AS (Autonomous System) προέλευσης και προορισμού. Για το λόγο αυτό, στο το αρχείο καταγραφής της προηγούμενης παραγράφου αλλάξαμε τις διευθύνσεις χρησιμοποιώντας το εργαλείο editcap[38], ώστε η IP του θύματος να είναι η 147.102.222.210. Τέλος, το τελικό αρχείο καταγραφής με τις σωστές IP δόθηκε ως είσοδος στο nProbe για να παραχθούν οι ροές (flows) που δόθηκαν ως είσοδος στα νευρωνικά δίκτυα.

**Επίθεση πλημμύρας TCP (TCP Syn Flood) και πλημμύρας ICMP (ICMP Flood)**
Η κίνηση που αποτέλεσε τις κλάσεις της TCP Syn Flood και της ICMP Flood προήλθε από το γνωστό αρχείο επίθεσης από το 2007 της CAIDA[1]. Στα αρχικά αρχεία καταγραφής μετασχηματίστηκε η IP προορισμού, με τον τρόπο που περιγράφεται στην προηγούμενη παράγραφο. Στη συνέχεια παράχθηκαν οι Flows με χρήση του nProbe .

**Port Scanning**
Τέλος, η Port Scanning επίθεση δημιουργήθηκε με το εργαλείο nmap[3].
Η επίθεση έγινε προς host σε εικονικό περιβάλλον χρησιμοποιώντας την εντολή
*nmap 192.168.1.254 -sS -p 19-25,35,42,43,50-58,80-88,101,109,110-117,123,137-139,156,161,179,443,8008,8080,8888*

## 4.2 Τα Τελικά Δεδομένα (Datasets)

Τα δεδομένα καταγραφής που περιγράφονται στην Ενότητα 4.1 συνδυάστηκαν σε 5 διαφορετικά Datasets, όπως περιγράφεται παρακάτω.

**Dataset 1** Περιέχει Ομαλή Κίνηση και κίνηση από την επίθεση Ping Flood . Το πλήθος των δειγμάτων (samples) που αφορούν σε Ομαλή κίνηση είναι 390,000 και των δειγμάτων που αφορούν σε επίθεση είναι 390,000.

**Dataset 2** Περιέχει Ομαλή Κίνηση και κίνηση από την επίθεση SYN Flood . Το πλήθος των δειγμάτων που αφορούν σε Ομαλή κίνηση είναι 390,000 και των δειγμάτων που αφορούν σε επίθεση είναι 390,000.

**Dataset 3** Περιέχει Ομαλή Κίνηση και κίνηση από την επίθεση UDP Flood . Το πλήθος των δειγμάτων που αφορούν σε Ομαλή κίνηση είναι 390,000 και των δειγμάτων που αφορούν σε επίθεση είναι 390,000.

**Dataset 4** Περιέχει Ομαλή Κίνηση και κίνηση από την επίθεση Port Scanning . Το πλήθος των δειγμάτων που αφορούν σε Ομαλή κίνηση είναι 362,800 και των δειγμάτων που αφορούν σε επίθεση είναι 68,000.

**Dataset 5** Περιέχει Ομαλή Κίνηση και κίνηση από τις επιθέσεις Port Scanning και SYN Flood . Το πλήθος των δειγμάτων που αφορούν σε Ομαλή κίνηση είναι 362,800, των δειγμάτων που αφορούν σε επίθεση SYN Flood είναι 118,000 και των δειγμάτων που αφορούν σε επίθεση Port Scanning είναι 88,000.

**Dataset 0** Περιέχει Ομαλή Κίνηση και κίνηση όλα τα είδη επιθέσεων Ping Flood , SYN Flood , UDP Flood και Port Scanning , περιέχει το σύνολο δηλαδή των δεδομένων που καταγράφηκαν. Το πλήθος αυτών είναι για την Ομαλή Κίνηση περίπου 450,000 Ροές, για την επίθεση Ping Flood περίπου 400,000 Ροές, για την επίθεση SYN Flood περίπου 400,000 Ροές, για την UDP Flood επίσης 400,000 και για την Port Scanning περίπου 90,000.

**Dataset 6** Αποτελεί μικρή παραλλαγή του Dataset 2 , με τη διαφορά ότι τα δεδομένα της Ομαλής κίνησης και της επίθεσης έχουν ανακατευτεί με διαφορετκό τρόπο.

## 4.3 Χαρακτηριστικά (Attributes) της Κίνησης που Συλλέχθηκαν (με το Πρωτόκολλο NetFlow) και Δόθηκαν ως Είσοδος στο Νευρωνικό

Όπως αναφέρθηκε ήδη, στη παρούσα Διπλωματική χρησιμοποιήθηκε το nProbe [34] για τη συλλογή των Ρωών και των επιθυμητών χαρακτηριστικών της κίνησης.
Τα Χαρακτηριστικά για τα οποία θέλουμε ο Εξαγωγέας να μας ενημερώνει ορίζονται μέσω Command Line (και τελικά δημιουργούν την κατάλληλη Template Record που στέλνεται στον Εξαγωγέα). Η IANA[39] έχει ορίσει ένα σύνολο Στοιχείων Πληροφορίας (**Information Elements, IEs**), τα οποία μπορούν να υλοποιηθούν από τους Exporters. Καθένα από αυτά έχει καθορισμένο τύπο και μέγεθος και αφορά σε μία πολύ συγκεκριμένη πληροφορία που ο Εξαγωγέας θα συγκεντρώνει και θα αποστέλει για κάθε Ροή. Τα Information Elements μπορεί κανείς να βρει στο και στο RFC 7012 [40] και στη σελίδα της IANA[41].

Στον Πίνακα 2 παρουσιάζονται τα Information Elements που χρησιμοποιήθηκαν στην παρούσα Διπλωματική. Αναγράφεται το ID που έχουν πάρει από την IANA, ο τύπος τους, η περιγραφή τους και το όνομά τους στην υπλοποίηση του nProbe .

**Σημείωση σχετικά με τα Χαρακτηριστικά (IEs) που παρουσιάζονται στον Πίν. 2**
Τα Χαρακτηριστικά για τα οποία ο NetFlow Exporter μας έστελνε πληροφορίες είναι αυτά που παρουσιάζονται στον Πίνακα. Ωστόσο, αυτά δε δίνονται αυτούσια ως είσοδο στο Νευρωνικό Δίκτυο. Για κάθε Flow που φτάνει στο Συλλέκτη γίνεται μία επεξεργασία που αφορά σε όσα Χαρακτηριστικά (IEs) έχουν διακριτές τιμές. Αυτά είναι τα PROTOCOL, L4_SRC_PORT, L4_DST_PORT, TCP_FLAGS, ICMP_Type, SRC_AS, SRC_AS. Πιο συγκεκριμένα,

- Το PROTOCOL δίνεται στο Νευρωνικό ως 3 χαρακτηριστικα: IsTCP, IsUDP και IsICMP. Καθένα από αυτά έχει την τιμή 0 ή 1. Το ποια πρωτόκολλα θα αναγνωρίζουμε αποφασίστηκε με βάση τα είδη των επιθέσεων που επιθυμούμε να αναγνωρίσουμε.
- Το TCP_FLAGS διασπάστηκε αντίστοιχα σε 5 χαρακτηριστικά, που αφοφούν στα Flags ACK, PSH, RST, SYN, FIN.
- Από το ICMP_Type κρατήθηκε με αντίστοιχη λογική πληροφορία μόνο για τους Τύπους ICMP 0 (Echo Reply), 3(Destination Unreachable), 5(Redirect), 11(Time Exceeded).
- Από το L4_SRC_PORT κρατήθηκε με αντίστοιχη λογική πληροφορία σχετικά με τις πόρτες 17, 18, FTP data (20), FTP control (21), SSH (22), Telnet (23), SMTP (25), 42, DNS(53), DHCP (67,68), HTTP(80), Kerberos Auth (88), 101, 109, 110, NTP (123), 156, SNMP (161), BGP (179). Οι θύρες αυτές επιλέχθηκαν επειδή είναι από τις πλέον συχνές.
- Το L4_DST_PORT αναλύεται στις ίδιες θύρες με το L4_SRC_PORT.
- Τέλος, από τα SRC_AS μας ενδιέφεραν μόνο οι Ροές που είχαν την τιμή 3323 (AS Κωδικός του ΕΜΠ), αφού στο ΕΜΠ θεωρήσαμε ότι βρισκόταν το θύμα.

Τέλος, σημειώνεται ότι οι IP διευθύνσεις προέλευσης και προορισμού δε δόθηκαν ως είσοδοι στο Νευρωνικό δίκτυο.

| **ID** | **Τύπος** | **Περιγραφή** | **Όνομα nProbe** |
|---|---|---|---|
| 24 | Unsigned64 | Πλήθος εξερχόμενων πακέτων από την προηγούμενη αναφορά (αν υπήρξε) για τη συγκεκριμένη Flow | OUT_PKTS |
| 2 | Unsigned64 | Πλήθος εισερχόμενων πακέτων από την προηγούμενη αναφορά (αν υπήρξε) για τη συγκεκριμένη Flow | IN _PKTS |
| 4 | Unsigned8 | Ο κωδικός του πρωτοκόλλου που ενθυαλακώνεται στο IP Payload, όπως έχει δηλωθεί στο IANA Protocol Numbers registry. Σε IPv4 πακέτα, ο κωδικός αυτός υπάρχει στο πεδίο Protocol της επικεφαλίδας. | PROTOCOL |
| 8 | ipv4Address | Η διεύθυνση προέλευσης του πακέτου. | IPV4_SRC_ADDR |
| 12 | ipv4Address | Η διεύθυνση προορισμού του πακέτου. | IPV4_DST_ADDR |
| 7 | Unsigned16 | Η θύρα προέλευσης στο πρωτόκολλο του στρώματος μεταφοράς (TCP , UDP ή SCTP) | L4_SRC_PORT |
| 11 | Unsigned16 | Η θύρα προορισμού στο πρωτόκολλο του στρώματος μεταφοράς (TCP , UDP ή SCTP) | L4_DST_PORT |
| 6 | Unsigned16 | Συσσωρευτική απεικόνιση των TCP Flags ως εξής. Για κάθε bit των TCP Flags υπάρχει ένα bit στο πεδίο αυτό. Αν το bit των TCP Flags έχει τεθεί σε κάποιο από τα πακέτα της Flow, τότε το αντίστοιχο bit του πεδίου είναι 1, αλλιώς είναι 0. | TCP_FLAGS |
| 252 | Unsigned8 | Η ελάχιστη τιμή του πεδίου TTL που παρατηρήθηκε στη Flow . | MIN_TTL |
| 53 | Unsigned8 | Η μέγιστη τιμή του πεδίου ΤΤΛ που παρατηρήθηκε στη Φλοω. | MAX_TTL |
| 32 | Unsigned16 | Type και Code του IPv4 ICMP μηνύματος (ICMP type * 256) + ICMP code) | ICMP_Type |
| - | Unsigned64 | Πλήθος πακέτων με μέγεθος μικρότερο ή ίσο από 128 Bytes. | NUM_PKTS_UP_TO_128_BYTES |
| 57581 | Unsigned64 | Πλήθος πακέτων που ήρθαν ως retransmission (από προέλευση προς προορισμό) | RETRANSMITTED_IN_PKTS |
| 57582 | Unsigned64 | Πλήθος πακέτων που ήρθαν ως retransmission (από προορισμό προς προέλευση) | RETRANSMITTED_OUT_PKTS |
| 152 | Unsigned32 | Η σχετική (ως προς την έναρξη λειτουργίας του Exporter) Timestamp της λήψης του πρώτου πακέτου της Flow . | FLOW_START_MILLISECONDS |
| 153 | Unsigned32 | Η σχετική (ως προς την έναρξη λειτουργίας του Exporter) Timestamp της λήψης του τελευταίου πακέτου της Flow . | FLOW_END_MILLISECONDS |
| 16 | Unsigned32 | Το AS (Autonomous System) προέλευσης βάσει της IP διεύθυνσης. | SRC_AS |
| 17 | Unsigned32 | Το AS (Autonomous System) προορισμού βάσει της IP διεύθυνσης. | DST_AS |

Πίνακας 2: Χαρακτηριστικά (Attributes) της κίνησης που συλλέχθηκαν από τον Εξαγωγέα NetFlow (NetFlow Exporter)

## 4.4 Υλοποίηση και Εκπαίδευση των Νευρωνικών Δικτύων

Τέλος, για την υλοποίηση, την εκπαίδευση των νευρωνικών, την αξιολόγησή τους και την αποθήκευση των εκπαιδευμένων δικτύων χρησιμοποιήθηκε η βιβλιοθήκη keras[42] σε Python.

Όλος ο κώδικας βρίσκεται στο github.com/OrestisALpos/WebTrafficClassifier

# 5 Αξιολόγηση

## 5.1 Σύγκριση Διαφορετικών Δομών MLP

Αρχικά έγινε εκπαίδευση και επαλήθευση (validation) πολλών διαφορετικών συνδυασμών MLP δικτύων. Οι διαφορετικοί συνδυασμοί αφορούσαν στο βάθος του δικτύου (πλήθος κρυφών επιπέδων), στον αριθμό των νευρώνω ανά επίπεδο και στο ποσοστό εφαρμογής του Dropout (Dropout Rate). Συγκεκριμένα, δοκιμάστηκαν δίκτυα με πλήθος κρυφών επιπέδων από 0 (είχε μόνο επίπεδα Εισόδου και Εξόδου) έως 4 (4 κρυφά επίπεδα, συνολικά 6 μαζί με το Εισόδου και το Εξόδου). Το πλήθος νευρώνων ανά επίπεδο έπαιρνε τις τιμές 20, 30, 40, 50 και 60 και αφορούσε μόνο στα κρυφά επίπεδα, αφού το μέγεθος του επιπέδου εισόδου (Input Layer) ήταν πάντα όσα και τα χαρακτηριστικά (Features) εισόδου και το μέγεθος του επιπέδου εξόδου εξαρτόταν από το πλήθος των κλάσεων προς αναγνώριση. Τέλος, το Dropout Rate έπαιρνε τιμές μεταξύ των 0, 0.2, 0.4, 0.5, 0.6, 0.8. Έγινε εκπαίδευση όλων των συνδυασμών δικτύων πάνω στα Dataset 1 έως 4 (βλέπε Ενότητα 4.2 για περιγραφή των ειδών επίθεσης και των Datasets) και στη συνέχεια και στα Dataset 0 , Dataset 5 και Dataset 6.

**Επιλογή πλήθους εποχών εκπαίδευσης**

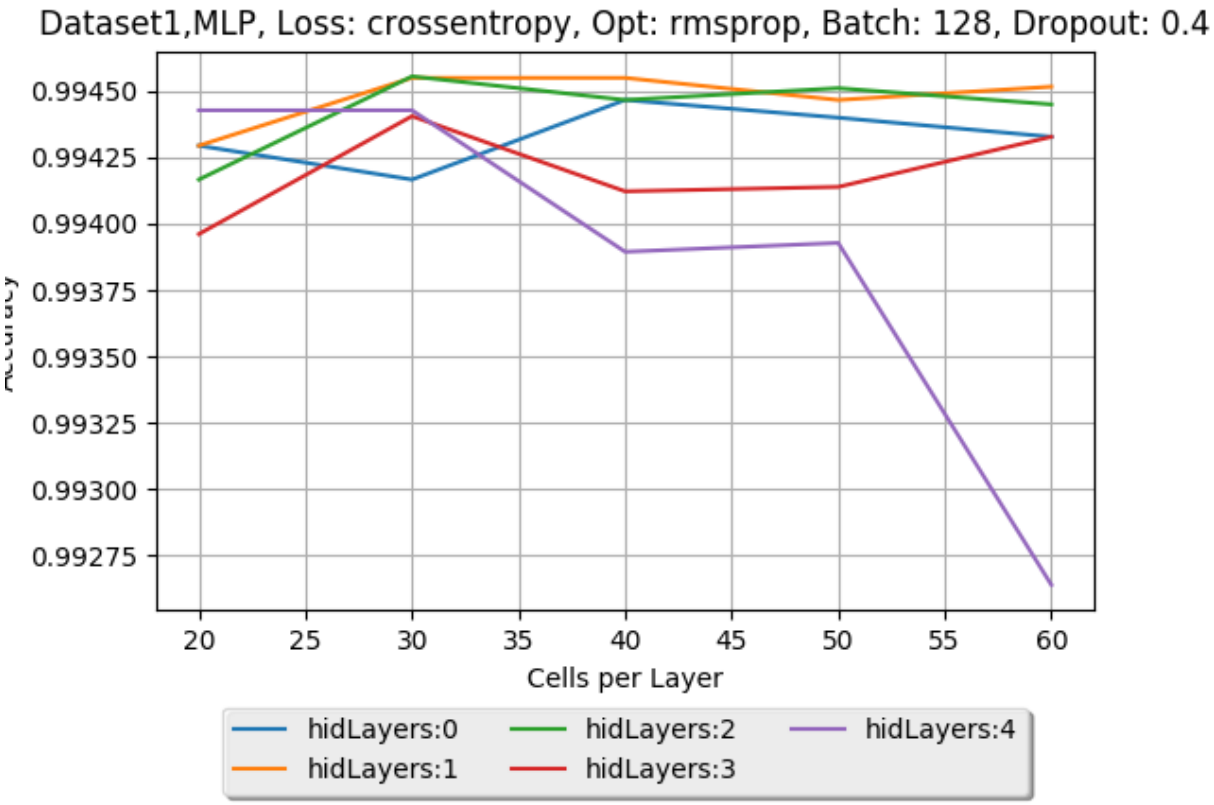
Επειδή αυτό που αρχικά μας ενδιέφερε ήταν να πάρουμε μία πρώτη εικόνα για τη συμπεριφορά των MLP δικτύων όταν εκπαιδευτούν πάνω στα διάφορα Dataset, δε δόθηκε έμφαση στο να εκπαιδευτεί κάθε δίκτυο με το βέλτιστο δυνατό τρόπο (ή το βέλτιστο τρόπο που είχα υπ' όψη μου). Έτσι, έγιναν κάποιες απλοποιήσεις. Μία από αυτές ήταν να τεθεί το πλήθος των εποχών (number of epochs) στην τιμή 10 για όλες τις εκπαιδεύσεις. Ο αριθμός αυτός βέβαια δεν επιλέχθηκε τυχαία, αλλά προέκυψε εκπαιδεύοντας και ελέγχοντας την ακρίβεια ενός μικρού MLP δικτύου. Το μικρό αυτό δίκτυο έδειξε ότι σε περισσότερες από 10 εποχές η ακρίβεια (Accuracy) των προβλέψεων σταματά να βελτιώνεται. Έτσι, επιλέχθηκε, για λόγους αυτοματοποίησης και απλούστευσης της διαδικασίας, όλες οι εκπαιδεύσεις να γίνονται για 10 εποχές.

Σημειώνεται εδώ, σχετικά με το πλήθος των εποχών, ο αριθμός 10 είναι γενικά αρκετά μικρός για πλήθος εποχών σε εκπαίδευση επιβλεπόμενου (supervised) νευρωνικού δικτύου. Ωστόσο, εδώ δικαιολογείται η τιμή του επειδή τα Dataset ήταν αρκετά μεγάλα και περιείχαν δείγματα (training samples) που έμοιαζαν σε πολύ μεγάλο βαθμό μεταξύ τους (π.χ. τα δεδομένα που χρησιμοποιήθηκαν για τη UDP επίθεση προήλθαν από το εργαλείο Scapy[2] και, εκτός από διευθύνσεις IP και θύρες, ήταν ίδια).
Οι γραφικές παραστάσεις με τα σημαντικότερα αποτελέσματα ακολουθούν στην τρέχουσα Ενότητα, ενώ κάποιες Γραφικές που δεν απαιτούν σχολιασμό υπάρχουν στο στο Παράστημα Α΄.

Για τα Dataset 1 (Καλόβουλη κίνηση vs ICMP Ping Flood επίθεση), 2 (Καλόβουλη κίνηση vs SYN Flood επίθεση) και 4 (Καλόβουλη κίνηση vs Port Scanning επίθεση) υπήρξε νευρωνικό δίκτυο που έδωσε ακρίβεια μεγαλύτερη από 99 %.

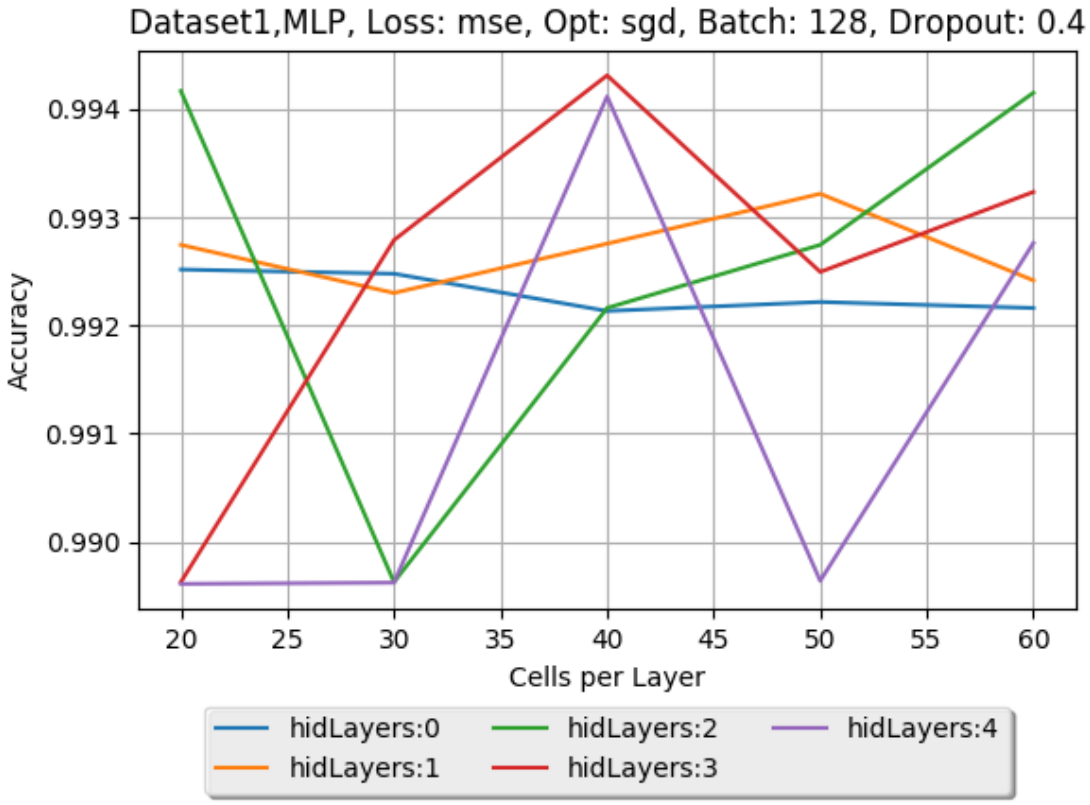
Εξαίρεση αποτέλεσε το Dataset 3 (Καλόβουλη κίνηση vs UDP Flood επίθεση). Εδώ, η καλύτερη ακρίβεια που επιτεύχθηκε ήταν 71%. Για το λόγο αυτό, έγιναν οι κινήσεις που περιγράφονται στην Ε-

νότητα 5.3. Το βασικό συμπέρασμα που προέκυψε, όπως θα δούμε και παρακάτω, είναι ότι η απόδοση του 71% οφειλόταν κυρίως σε Overfitting.
Τέλος, στο Dataset 0 (όλα τα είδη κίνησης, η καλόβουλη μαζί με τις 4 επιθέσεις) είχαμε τα εξής αποτελέσματα. Πολλοί συνδυασμοί δικτύων έδωσαν ακρίβεια μεταξύ 90% και 94%, όπως φαίνεται και στις ακόλουθες Γραφικές Παραστάσεις, ωστόσο υπήρξαν κάποια δίκτυα που έδωσαν ακόμα μεγαλύτερη ακρίβεια. Το βέλτιστο MLP περιγράφεται στην αμέσως επόμενη Ενότητα.



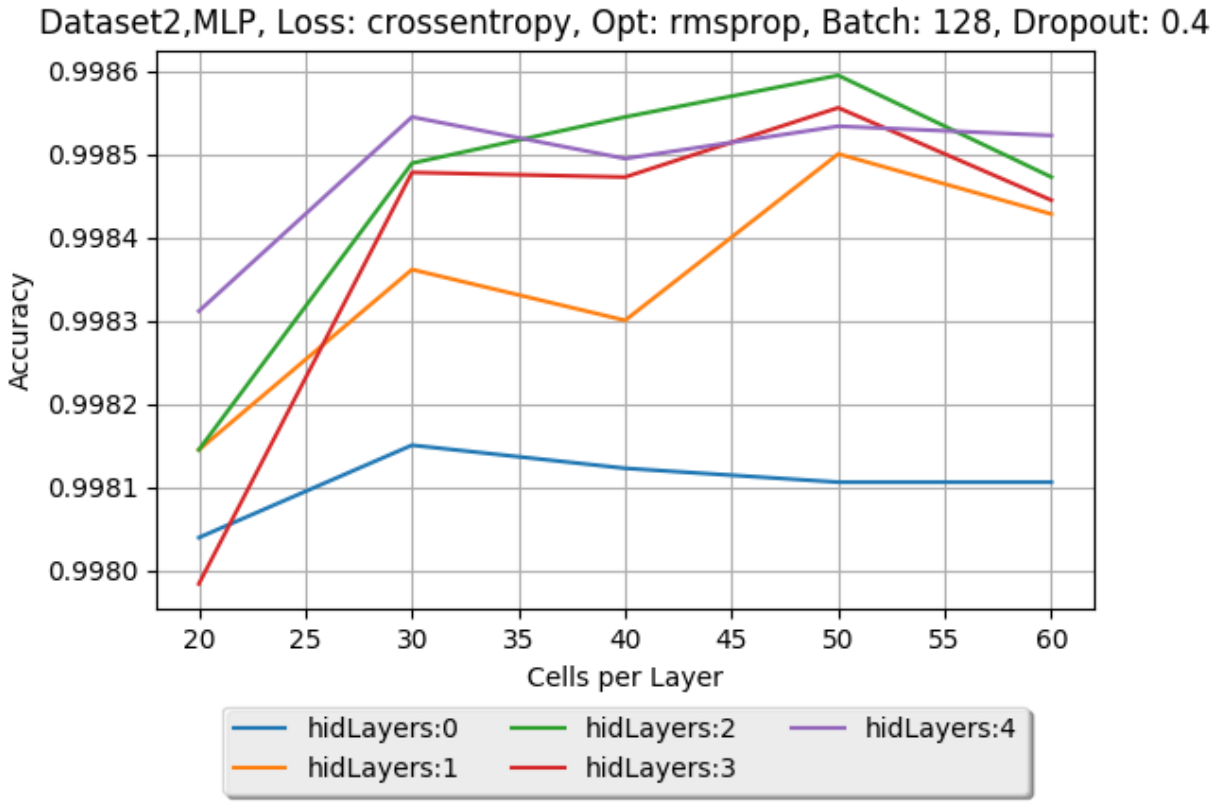

Σχήμα 18: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 1.

Στη γραφική παράσταση βλέπουμε την Ακρίβεια που έδωσαν τα MLP πάνω στο Dataset 1. Οι παράμετροι που εξετάστηκαν ήταν το πλήθος των κρυφών επιπέδων (Hidden Layers και το πλήθος των Νευρώνων ανά επίπεδο. Σημειώνεται ότι όλα τα επίπεδα είχαν ίδιο πλήθος Νευρώνων, πλην φυσικά του Επιπέδου Εισόδου (Input Layer) και του Επιπέδου Εξόδου (Output Layer). Το Dataset 1 περιέχει, όπως περιγράφεται στην Ενότητα 4.2, ομαλή κίνηση και κίνηση από την επίθεση SYN Flood . Η ακρίβεια είναι σε όλες τις περιπτώσεις πάνω από 99%, κάτι που σημαίνει ότι τα εν λόγω είδη κίνησης είναι εύκολο να διακριθούν το ένα από το άλλο, ακόμα και από Δίκτυα χωρίς κρυφό επίπεδο.



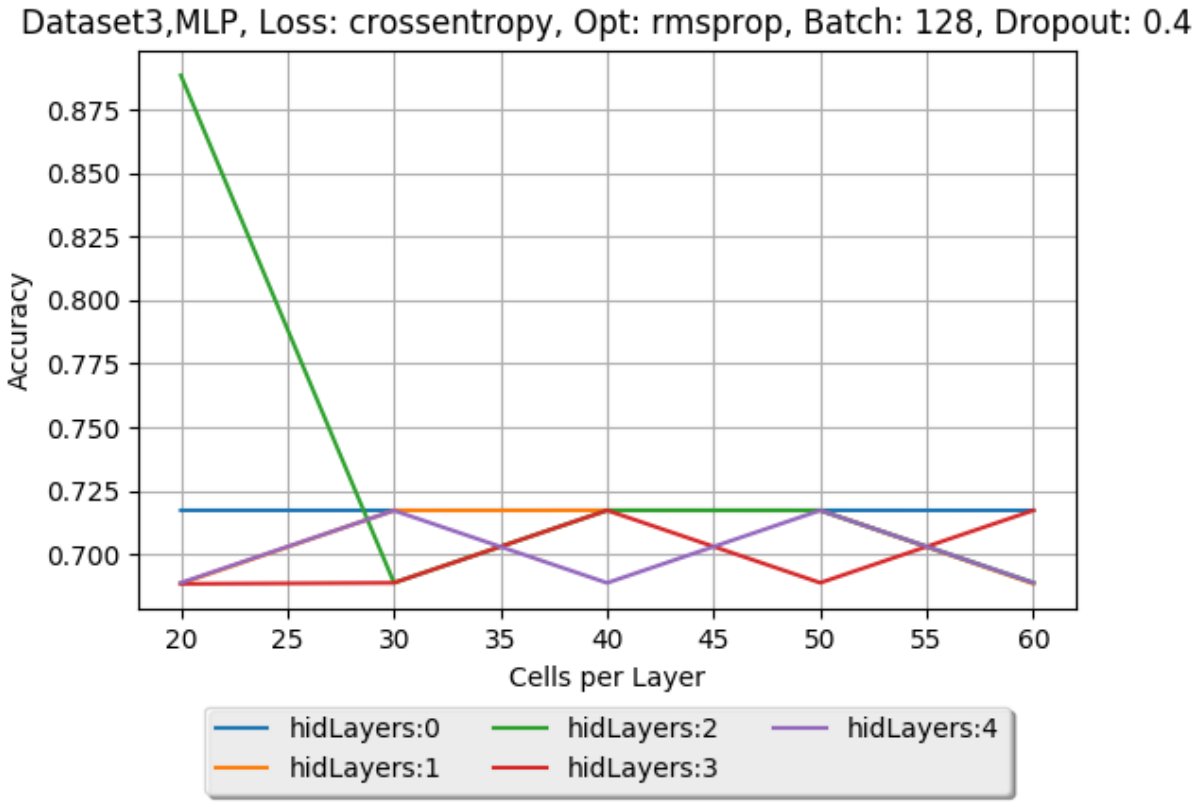

Σχήμα 19: MLP Νευρωνικά, με Συνάρτηση Κόστους MSE, Μέθοδο Εκπαίδευσης SGD και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 1.

Εντελώς αντίστοιχα με την προηγούμενη Γραφική, εδώ φαίνεται η Ακρίβεια που έδωσαν τα MLP πάνω στο Dataset 1, αλλά τώρα είχαν ως Συνάρτηση Κόστους την MSE και Μέθοδο Εκπαίδευσης την SGD. Η ακρίβεια των προβλέψεων κινήθηκε και εδώ πάνω από το 99%, ελάχιστα χαμηλότερα από την προηγούμενη περίπτωση. Παρόμοια ήταν και τα αποτελέσματα σε όλα τα άλλα MLP δίκτυα που δοκιμάστηκαν στο Dataset 1, γι΄ αυτό και δεν παρουσιάζονται άλλες γραφικές παραστάσεις από το Dataset αυτό.



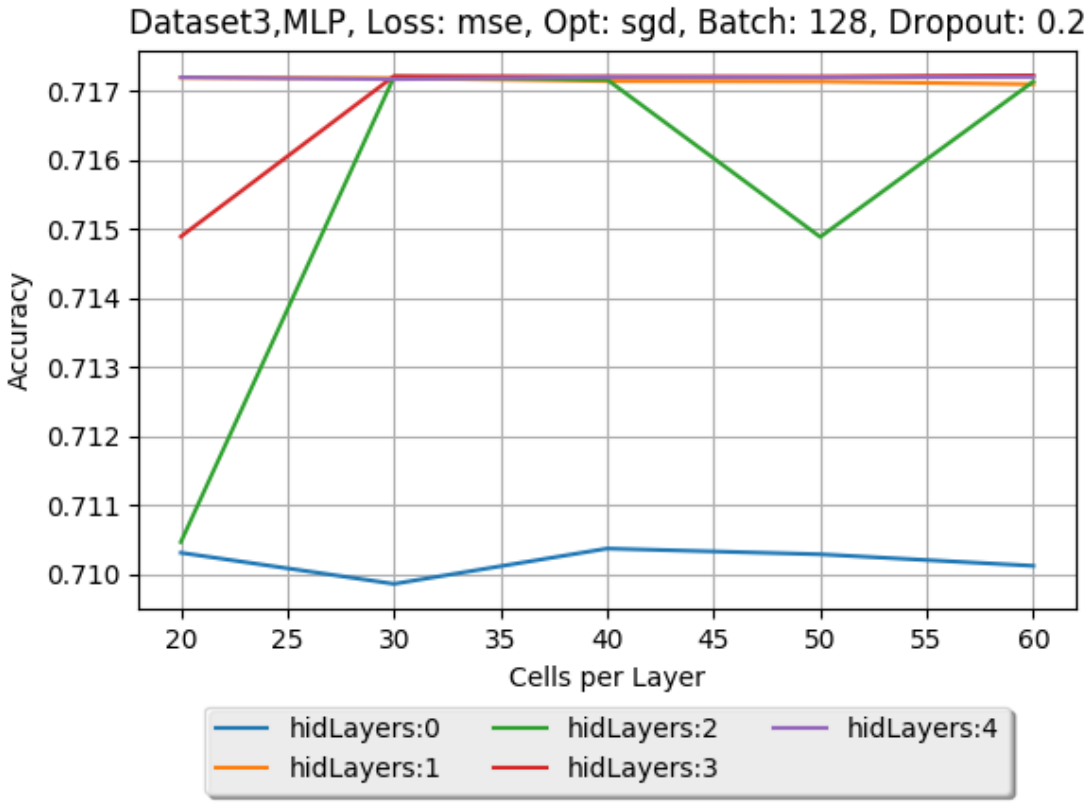

Σχήμα 20: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 2.

Στη γραφική παράσταση βλέπουμε την Ακρίβεια που έδωσαν τα MLP πάνω στο Dataset 2. Οι παράμετροι που εξετάστηκαν ήταν και εδώ το πλήθος των κρυφών επιπέδων και το πλήθος των Νευρώνων ανά επίπεδο. Το Dataset 2 περιέχει, όπως περιγράφεται στην Ενότητα 4.2, ομαλή κίνηση και κίνηση από την επίθεση UDP Flood . Η ακρίβεια είναι σε όλες τις περιπτώσεις πάνω από 99%, κάτι που οδηγεί στο συμπέρασμα ότι τα εν λόγω είδη κίνησης είναι εύκολο να διακριθούν το ένα από το άλλο. Επίσης, η διαφορά είναι πολύ μικρή, αλλά βλέπουμε τα δίκτυα με 2 και 3 κρυφά επίπεδα να δίνουν καλύτερη ακρίβεια. Και για το Dataset 2 δοκιμάστηκαν διάφοροι συνδυασμοί Συνάρτησης Κόστους, Μεθόδου Εκπαίδευσης και ρυθμού Dropout. Οι διαφορές ήταν αμελητέες, ωστόσο την καλύτερη Ακρίβεια έδωσε ο συνδυασμός που παρουσιάζεται εδώ, Συνάρτηση Κόστους Crossentropy και Μέθοδος Εκπαίδευσης RMSProp.



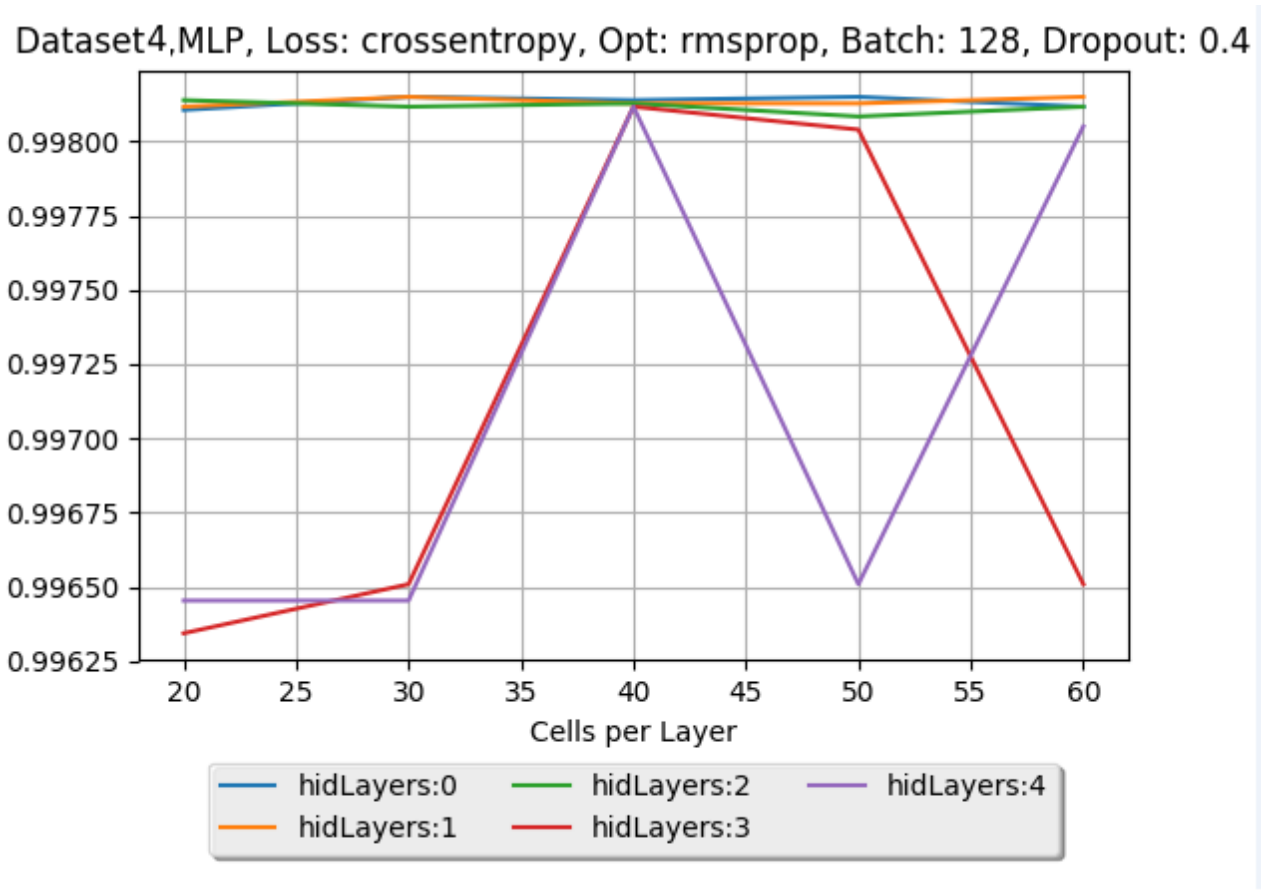

Σχήμα 21: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 3.

Το Dataset 3 ήταν το μόνο Dataset στο οποίο τα MLP δεν απέδωσαν καλή ακρίβεια. Το Dataset 3 περιέχει, όπως περιγράφεται στην Ενότητα 4.2, ομαλή κίνηση και κίνηση από την επίθεση UDP Flood . Ουσιαστικά, βλέπουμε ότι σε κάθε συνδυασμό η ακρίβεια των προβλέψεων είναι 70% με 72%, κάτι που φανερώνει ότι η έξοδος έπαψε να εξαρτάται από την είσοδο. Για το Dataset 3 έγινε περαιτέρω διερεύνηση (βλ. Ενότητα ;;) και διαπιστώθηκε το Overfitting που είχε συμβεί.



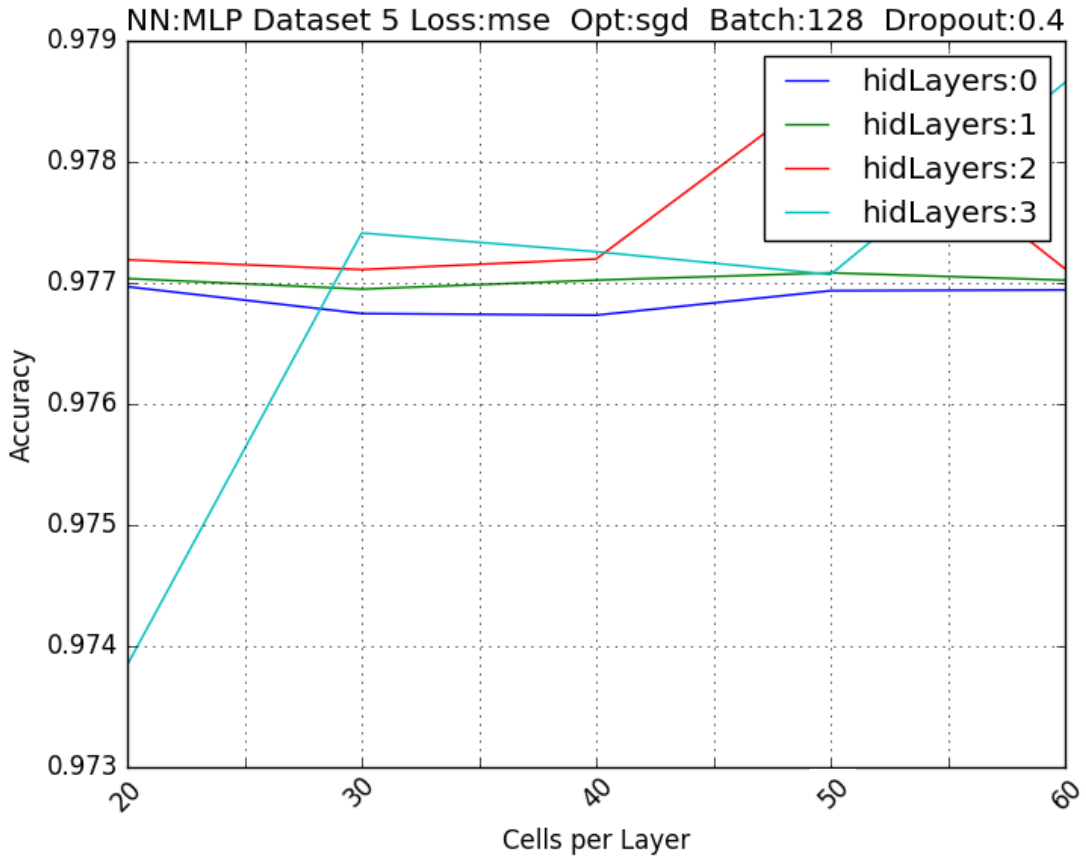

Σχήμα 22: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 3.

Και σε αυτή, όπως και στην προηγούμενη Γραφική, φαίνεται η αποτυχία των Νευρωνικών που δοκιμάστηκαν να εκπαιδευτούν στο Dataset 3, κάτι που, όπως φαίνεται στην Ενότητα **;;**, συνέβη λόγω Overfitting.




Σχήμα 23: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 4.

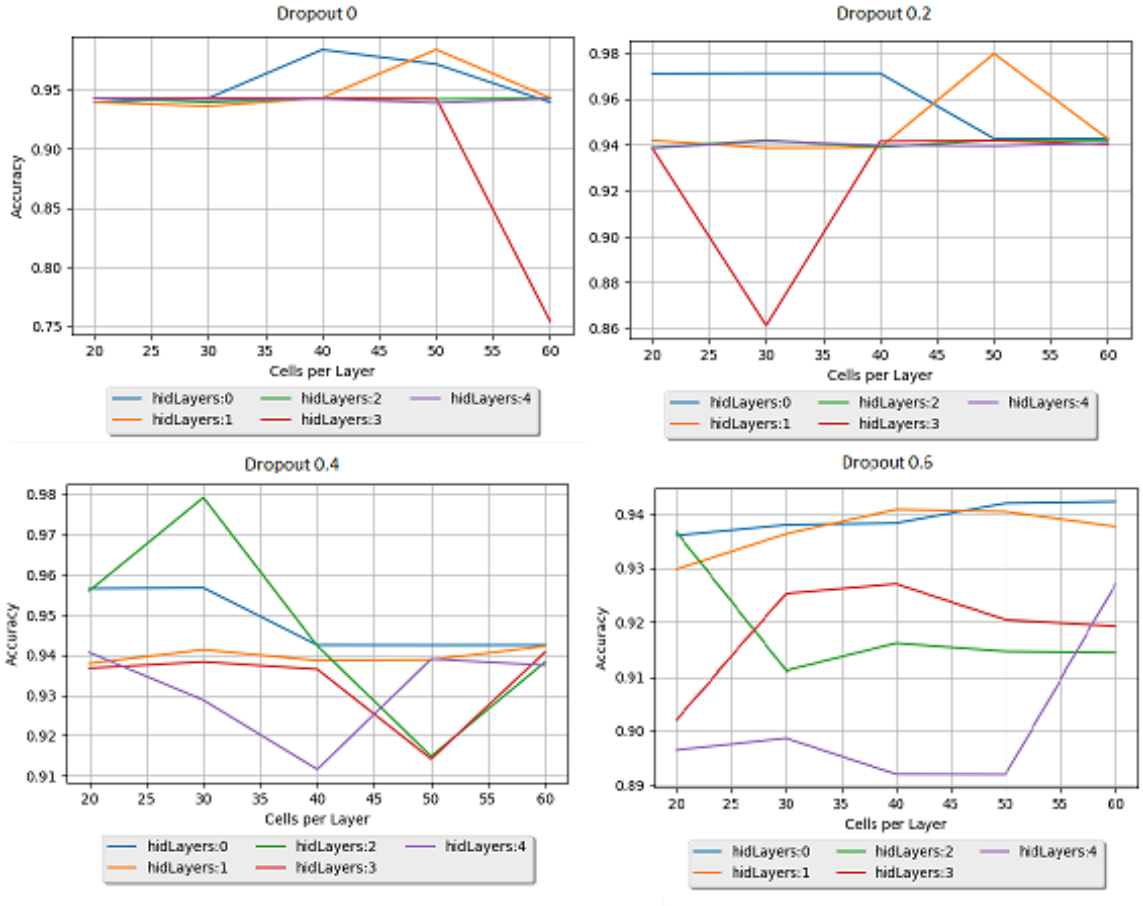
Στη γραφική παράσταση βλέπουμε την Ακρίβεια που έδωσαν τα MLP πάνω στο Dataset 4. Οι παράμετροι που εξετάστηκαν ήταν και εδώ το πλήθος των κρυφών επιπέδων και το πλήθος των Νευρώνων ανά επίπεδο. Το Dataset 4 περιέχει, όπως περιγράφεται στην Ενότητα 4.2, ομαλή κίνηση και κίνηση από την επίθεση Port Scanning . Η ακρίβεια είναι σε όλες τις περιπτώσεις πάνω από 99%, κάτι που οδηγεί στο συμπέρασμα ότι τα εν λόγω είδη κίνησης είναι εύκολο να διακριθούν το ένα από το άλλο. Και για το Dataset 4 δοκιμάστηκαν διάφοροι συνδυασμοί Συνάρτησης Κόστους, Μεθόδου Εκπαίδευσης και ρυθμού Dropout και η ακρίβεια ήταν παρόμοια σε όλες τις περιπτώσεις.




Σχήμα 24: MLP Νευρωνικά, με Συνάρτηση Κόστους MSE, Μέθοδο Εκπαίδευσης SGD και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 5.

Στη γραφική παράσταση βλέπουμε την Ακρίβεια που έδωσαν τα MLP πάνω στο Dataset 5. Οι παράμετροι που εξετάστηκαν ήταν και εδώ το πλήθος των κρυφών επιπέδων και το πλήθος των Νευρώνων ανά επίπεδο. Το Dataset 5 περιέχει, όπως περιγράφεται στην Ενότητα 4.2, ομαλή κίνηση και κίνηση από τις επιθέσεις Port Scanning και SYN Flood , είναι δηλαδή το πρώτο Dataset που δοκιμάστηκε και περιείχε 3 διαφορετικά είδη κίνησης. Αυτό που άλλαξε όσον αφορά στην δομή των Νευρωνικών Δικτύων σε σχέση με τα προηγούμενα Dataset είναι το Επίπεδο Εξόδου (Output Layer), που τώρα περιείχε 3 Νευρώνες.

Το αποτέλεσμα ήταν ακρίβεια χαμηλότερη μεν από τις περιπτώσεις που είχαμε αναγνώριση σε 2 κλάσεις, αλλά και πάλι παρέμεινε σε υψηλό επίπεδο, περίπου 97.7%. Και εδώ βλέπουμε τα δίκτυα με 2 ή 3 κρυφά επίπεδα να αποδίδουν ελαφρώς καλύτερα από τα υπόλοιπα.

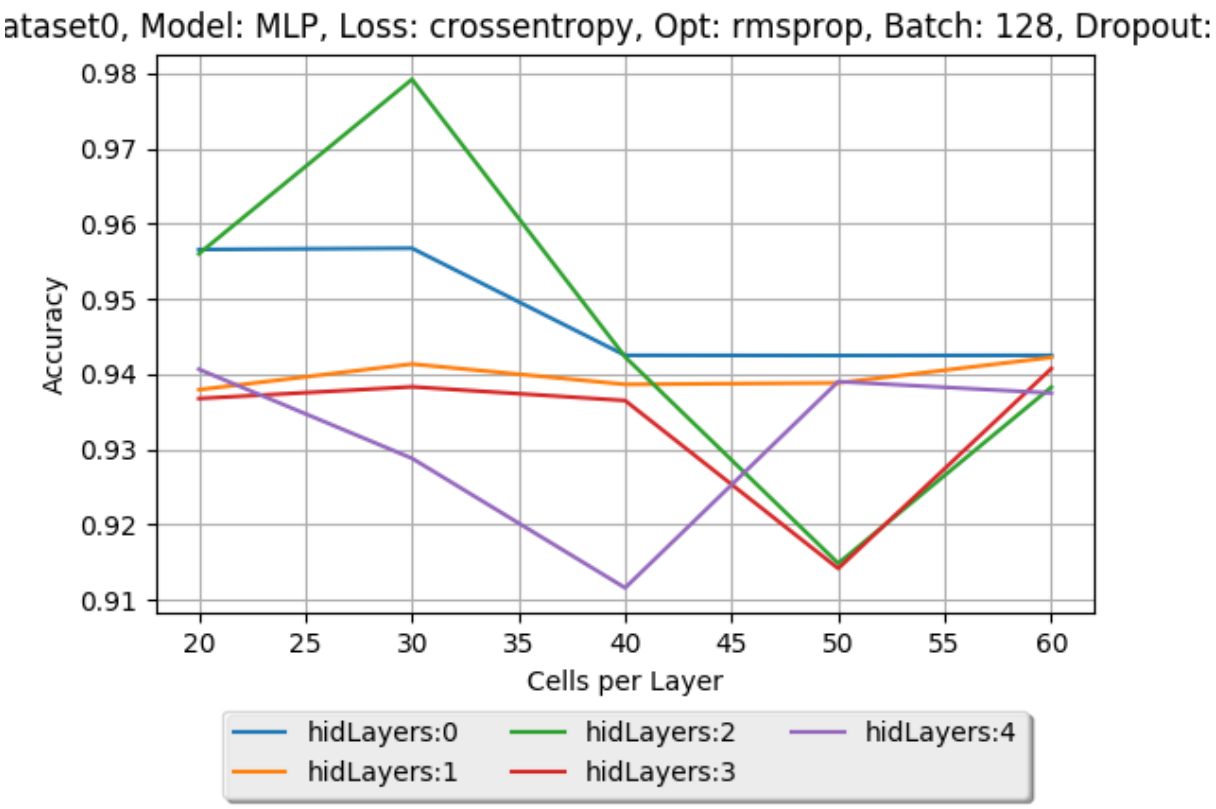



Σχήμα 25: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0, 0.2, 0.4 και 0.6, εκπαιδευμένα πάνω στο Dataset 0.

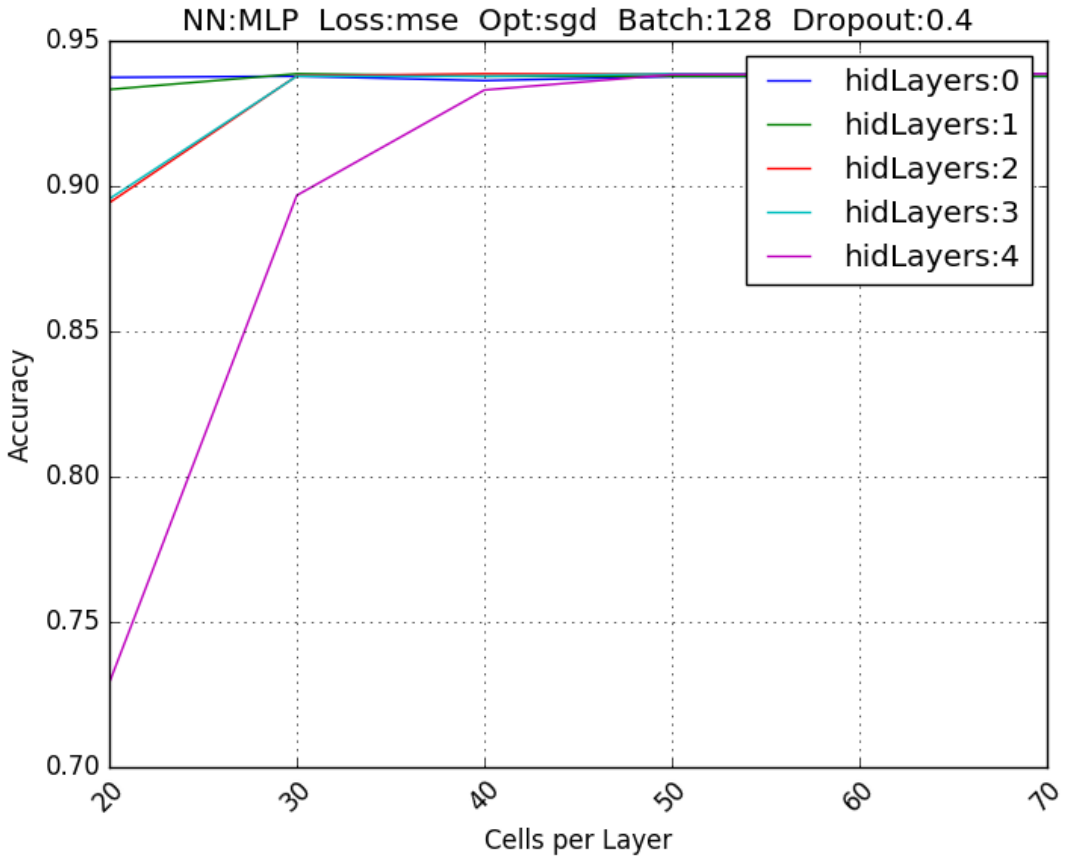
Στις 4 αυτές παραστάσεις βλέπουμε την ακρίβεια για τα MLP που δοκιμάστηκαν πάνω στο Dataset 0, το Dataset που περιέχει, όπως περιγράφεται στην Ενότητα 4.2, και τα 5 είδη κινήσεων. Οι παράμετροι που εξετάστηκαν ήταν και εδώ το πλήθος των κρυφών επιπέδων και το πλήθος των Νευρώνων ανά επίπεδο. Κάθε Γραφική αντιστοιχεί σε ένα Dropout Rate, όπως αναγράφεται στο Σχήμα.

Βλέπουμε ότι τα βέλτιστα αποτελέσματα προέκυψαν για ρυθμό Dropout 0.4. Συμπαιρένουμε ότι υπάρχει ανάγκη για κανονικοποίηση και ότι η χρήση του Dropout επιδρά θετικά. Το ποσοστό 0.6 όμως, εξ' αιτίας και του βάθους που είχαν οι περισσότερες αρχιτεκτονικές, οδηγούσε σε πολύ μεγάλη απώλεια πληροφορίας και άρα χειρότερη ακρίβεια. Όσον αφορά στο πλήθος των κρυφών επιπέδων, φαίνεται ότι τα 4 είναι πάρα πολλά και δε δίνουν καλό αποτέλεσμα, ενώ τις καλύτερες ακρίβειες έδιναν συνήθως οι αρχιτεκτονικές με 1 ή 2 κρυφά επίπεδα.

Η γραφική που αφορά στο Dropout 0.4, που έδωσε το βέλτιστο αποτέλεσμα, φαίνεται ξεχωριστά στο επόμενο Σχήμα.



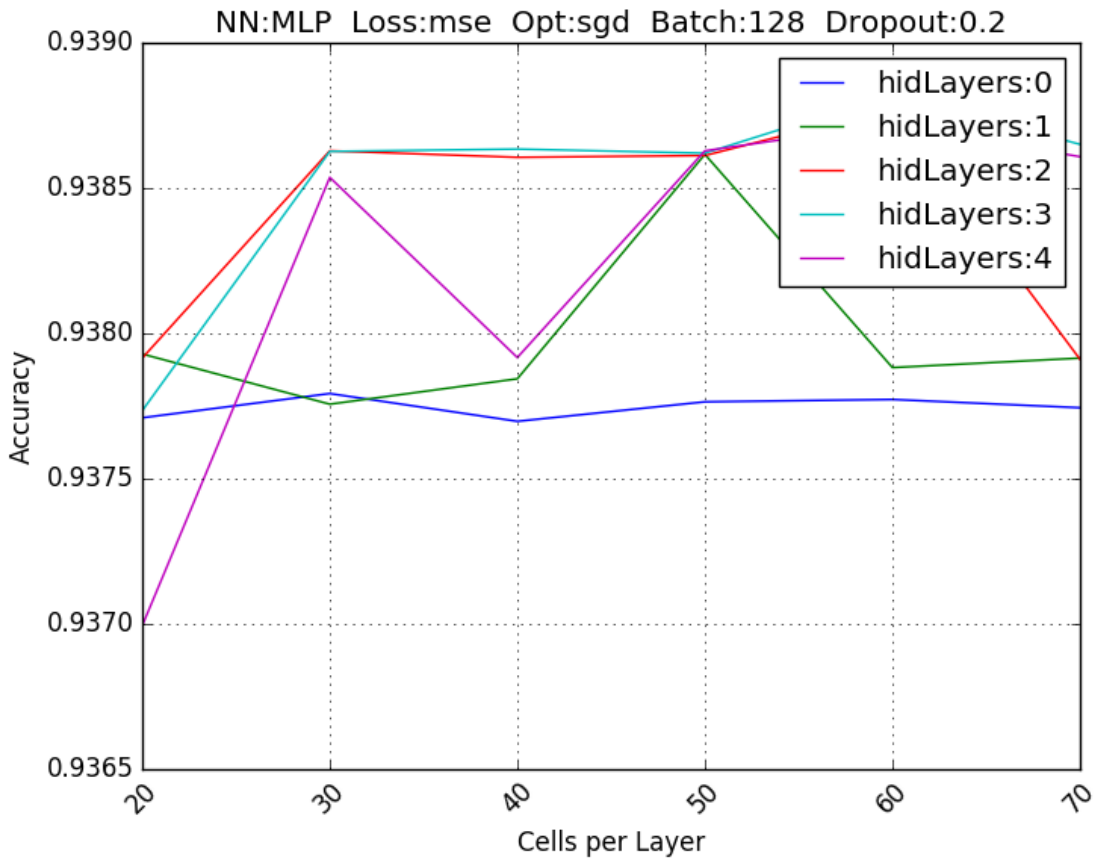

Σχήμα 26: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 0.
Τμήμα του προηγουμένου Σχήματος, όπου βλέπουμε την Ακρίβεια που έδωσαν τα MLP με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης RMSProp και ρυθμό Dropout 0.4 πάνω στο Dataset 0. Μεταξύ διάφορων τιμών Dropout Rate, το 0.4 έδωσε τα καλύτερα αποτελέσματα.
Μάλιστα, εδώ διακρίνουμε το MLP δίκτυο με 2 κρυφά επίπεδα, 30 Νευρώνες σε καθένα από αυτά και Ρυθμό Dropout 0.4, το οποίο κρίθηκε ως το **βέλτιστο MLP** Δίκτυο.




Σχήμα 27: MLP Νευρωνικά, με Συνάρτηση Κόστους MSE, Μέθοδο Εκπαίδευσης SGD και ρυθμό Dropout 0, 0.4, 0.5 και 0.8, εκπαιδευμένα πάνω στο Dataset 0.

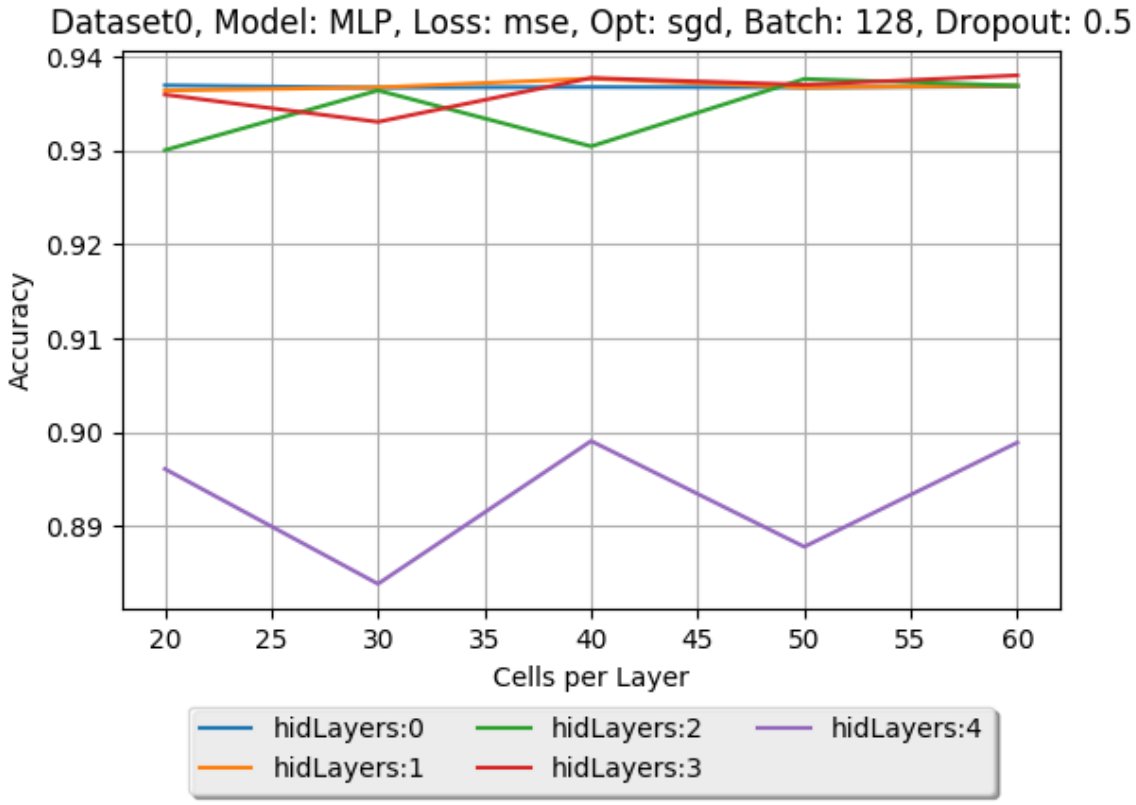
Στις 4 αυτές παραστάσεις βλέπουμε την ακρίβεια για τα MLP που δοκιμάστηκαν πάνω στο Dataset 0, το Dataset που περιέχει, όπως περιγράφεται στην Ενότητα 4.2, και τα 5 είδη κινήσεων. Οι παράμετροι που εξετάστηκαν ήταν και εδώ το πλήθος των κρυφών επιπέδων και το πλήθος των Νευρώνων ανά επίπεδο. Κάθε Γραφική αντιστοιχεί σε ένα Dropout Rate, όπως αναγράφεται στο Σχήμα.

Όπως και στην προηγούμενη Γραφική Παράσταση με τετράδα έτσι και εδώ, τα βέλτιστα αποτελέσματα προέκυψαν για ποσοστο Dropout μεταξύ του 0 και του 0.4, και συγκεκριμένα ήταν για 0.2, όπως φαίνεται στην ακόλουθη Γραφική.



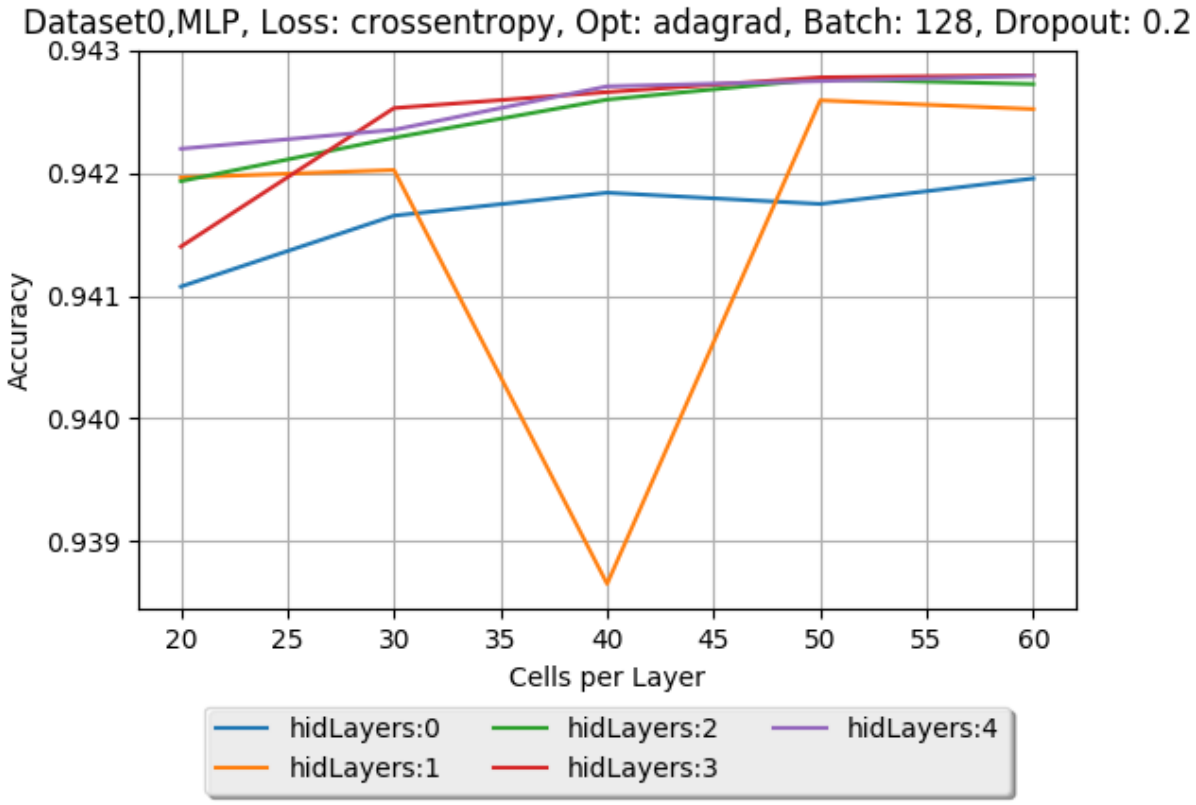

Σχήμα 28: MLP Νευρωνικά, με Συνάρτηση Κόστους MSE, Μέθοδο Εκπαίδευσης SGD και ρυθμό Dropout 0.2, εκπαιδευμένα πάνω στο Dataset 0.

Το Dropout Rate 0.2 έδωσε τα βέλτιστα αποτελέσματα, για Συνάρτηση Κόστους MSE και Μέθοδο Εκπαίδευσης SGD.



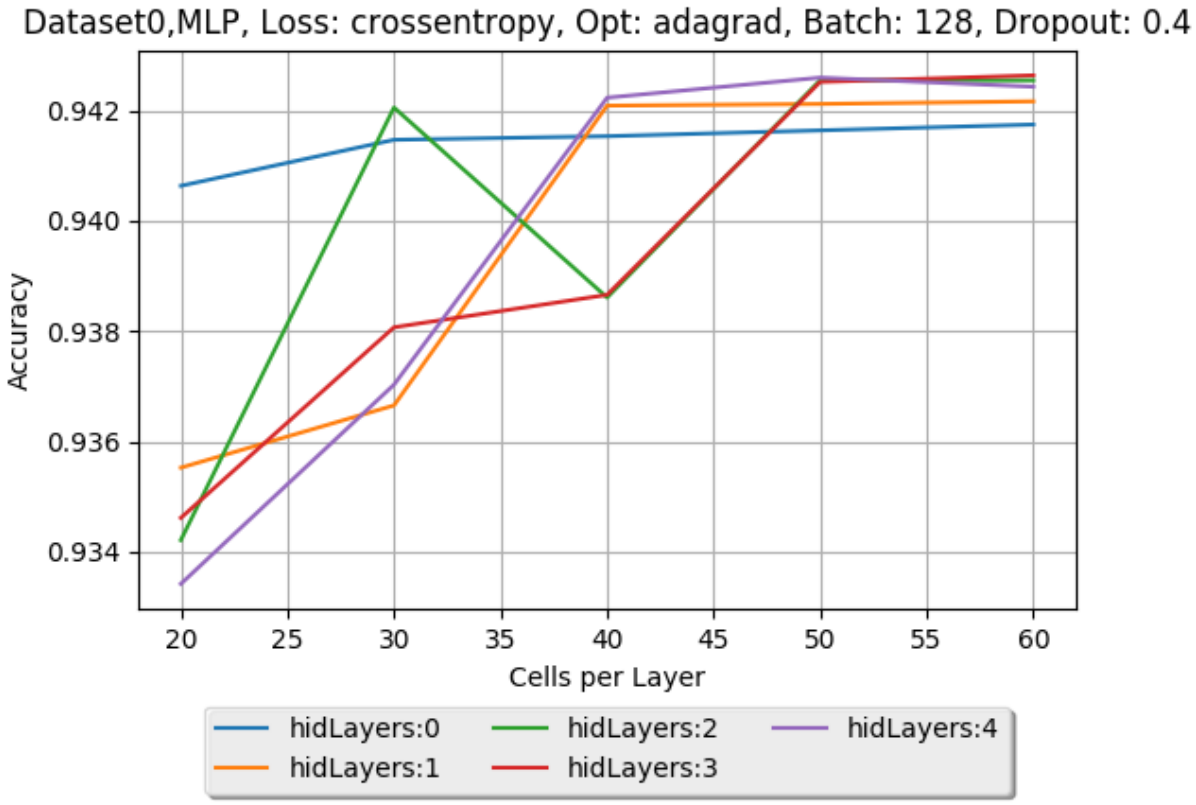

Σχήμα 29: MLP Νευρωνικά, με Συνάρτηση Κόστους MSE, Μέθοδο Εκπαίδευσης SGD και ρυθμό Dropout 0.8, εκπαιδευμένα πάνω στο Dataset 0.

Τμήμα της προηγούμενης τετράδας γραφικών, όπου βλέπουμε την ακρίβεια Δικτύων με Συνάρτηση Κόστους MSE, Μέθοδο Εκπαίδευσης SGD και ποσοστό Dropout 0.8. Το 0.8 είναι φυσικά πολύ μεγάλο για Dropout Rate, ωστόσο επιλέξαμε να το δοκιμάσουμε και πήραμε τα αναμενόμενα αποτελέσματα. Όσο μεγαλύτερο το πλήθος των επιπέδων, τόσο περισσότερη πληροφορία χάνεται κατά τη Εμπρόσθια μετάδοση (Forward Propagation) και άρα τόσο μικρότερη η ακρίβεια. Ποσοστό Dropout πάνω από 0.5 δεν έχει νόημα να εφαρμοστεί.




Σχήμα 30: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης Adagrad και ρυθμό Dropout 0.2, εκπαιδευμένα πάνω στο Dataset 0.

Στη γραφική παράσταση βλέπουμε την Ακρίβεια που έδωσαν τα MLP πάνω στο Dataset 0. Αυτό ήταν το πρώτο πείραμα που έγινε χρησιμοποιώντας τη Μέθοδο Εκπαίδευσης Adagrad και η Ακρίβεια ήταν αρκετά καλή, στο 94%. Δεν πλησίασε βέβαια το 98% που πετύχαμε με τη Μέθοδο Εκπαίδευσης RMSProp, αλλά κρίθηκε ικανοποιητική.




Σχήμα 31: MLP Νευρωνικά, με Συνάρτηση Κόστους Crossentropy, Μέθοδο Εκπαίδευσης Adagrad και ρυθμό Dropout 0.4, εκπαιδευμένα πάνω στο Dataset 0.
Άλλη μία Γραφική Παράσταση που αφορά σε Νευρωνικό εκπαιδευμένο με την τεχνική Adagrad, αυτή τη φορά με ρυθμό Dropout 0.4. Τα αποτελέσματα είναι παρόμοια με την προηγούμενη περίπτωση, ελαφρώς χαμηλότερα. Βλέπουμε εδώ, λοιπόν, το Dropout Rate 0.2 να αποδίδει ελαφρώς καλύτερα από το 0.4.

Οι προηγούμενες Γραφικές Παραστάσεις ήταν οι σημαντικότερες από όσες αφορούσαν στα MLP . Όλες οι υπόλοιπες υπάρχουν στο Παράρτημα Α΄.

## 5.2 Συμπεράσματα Από MLP - Βέλτιστο Δίκτυο

1. Όσον αφορά στο **πλήθος των επιπέδων**: Τα δίκτυα με περισσότερα κρυμμένα επίπεδα, παρά τη μεγαλύτερη «χωρητικότητα» και τις μεγαλύτερες δυνατότητες μάθησης, δεν έδωσαν εδώ καλύτερη ακρίβεια από τα μικρότερα δίκτυα. Δεδομένου ότι και μικρότερα δίκτυα (όπως εδώ το MLP με 2 κρυφά επίπεδα των 30 νευρώνων) μπορούν να φτάσουν στα ίδια επίπεδα ακρίβειας προβλέψεων, δεν υπάρχει λόγος να χρησιμοποιηθούν πολύ βαθιές αρχιτεκτονικές (με 4 και πάνω επίπεδα).

2. Όσον αφορά στο **μέγεθος κάθε επιπέδου** (πλήθος νευρώνων στο επίπεδο): Όταν είχαμε 20 νευρώνες ανά επίπεδο, η ακρίβεια ήταν σχεδόν πάντα χαμηλή. Αυξάνοντας τους σε 30, παρατηρήθηκε σε όλα σχεδόν τα μοντέλα αύξηση και στην ακρίβεια. Αυξάνοντάς τους σε πάνω από 30, σε κάποιες περιπτώσεις συνέχιζε να αυξάνεται η ακρίβεια, αλλά η μεταβολή ήταν πλέον πολύ μικρότερη. Επομένως, οποιοδήποτε πλήθος από 30 και πάνω φαίνεται να συμπεριφέρεται καλά. Εφόσον ένα δίκτυο με πιο «στενά» επίπεδα έχει και πολύ λιγότερα βάρη προς μάθηση, προτείνουμε ένα δίκτυο με 30 ή 40 νευρώνες ανά επίπεδο.

3. Όσον αφορά στο **Dropout**: Ένα ποσοστό Dropout 0.2 ή 0.4 έδωσε γενικά τα καλύτερα αποτελέσματα. Η εξήγηση γι' αυτό είναι ότι μικρότερο ποσοστό Dropout οδηγούσε σε Overfitting κατά την εκπαίδευση ενώ μεγαλύτερο, εξ' αιτίας και του βάθους που είχαν οι περισσότερες αρχιτεκτονικές, οδηγούσε σε πολύ μεγάλη απώλεια πληροφορίας.

**Βέλτιστο MLP δίκτυο**

Με 2 κρυφά επίπεδα με 30 νευρώνες σε καθένα και Dropout Rate 0.4, εκπαίδευση με RMSprop [βλ. Ενότητα 2.1.11]. Στην Ενότητα 5.4 φαίνονται αναλυτικά οι τιμές της Συνάρτησης Κόστους (Cost Function) και της ακρίβειας κατά τη διάρκεια της εκπαίδευσης.

| **Κρυφά επίπεδα** | **Νευρώνες ανά επίπεδο** | **Dropout Rate** | **Συναρτηση Κόστους - Μέθοδος εκπαίδευσης** |
|---|---|---|---|
| 2 | 30 | 0.4 | Crossentropy - RMSprop |

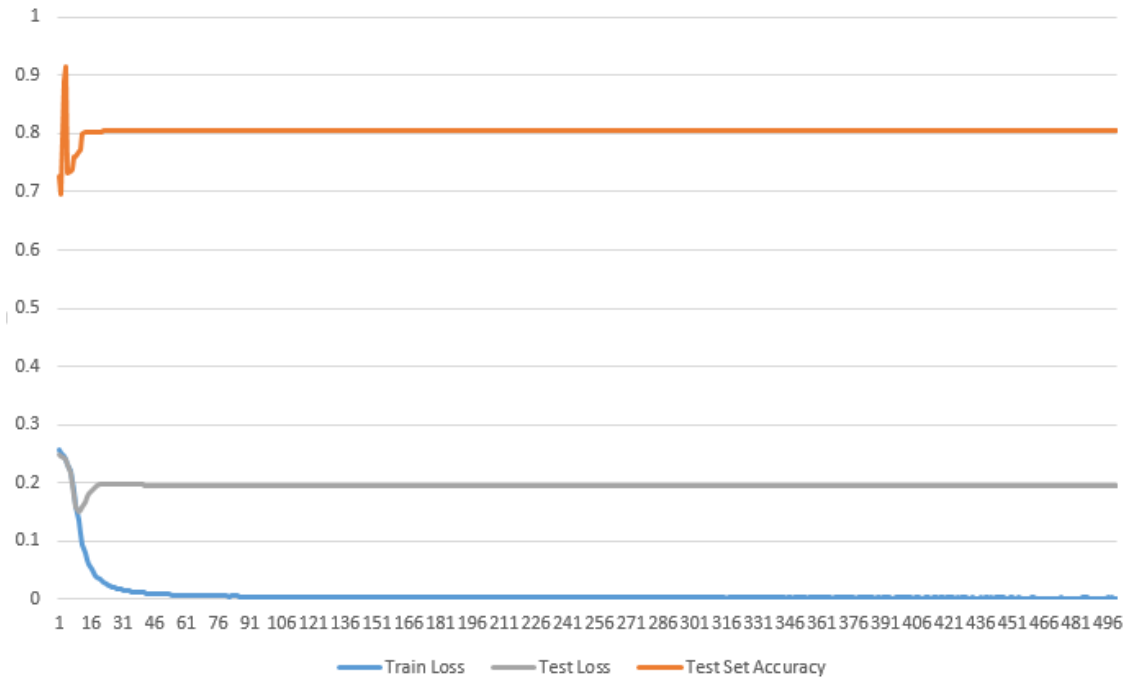
Πίνακας 3: Το MLP δίκτυο που επιλέχθηκε ως βέλτιστο

## 5.3 Διερεύνηση του Dataset 3

Στη συνέχεια παράχθηκαν γραφικές παραστάσεις που απεικονίζουν το σφάλμα (τιμή της Loss Function) τόσο πάνω στα δεδομένα εκπαίδευσης (training set) όσο και στα δεδομένα ελέγχου (validation set). Αυτό έγινε με σκοπό να διερευνηθεί περισσότερο το Dataset 3, η δυνατότητα του MLP δικτύου να το «μάθει» και να βγάλουμε κάποια συμπέρασμα όσον αφορά το overfitting, όπως το πότε συμβαίνει, για πόσα επίπεδα κλπ.
Η ακόλουθη γραφική παράσταση απεικονίζει τιμές για το σφάλμα εκπαίδευσης και ελέγχου σε κάθε εποχή εκπαίδευσης του MLP που βρέθηκε ως βέλτιστο στην προηγούμενη Ενότητα.

### Συμπεράσματα

Η χαμηλή απόδοση στο Dataset 3 οφείλεται σε Overfitting και θα πρέπει να γίνει εκπαίδευση χρησιμοποιώντας τη μέθοδο του Early Stopping για να αντιμετωπιστεί αυτό.

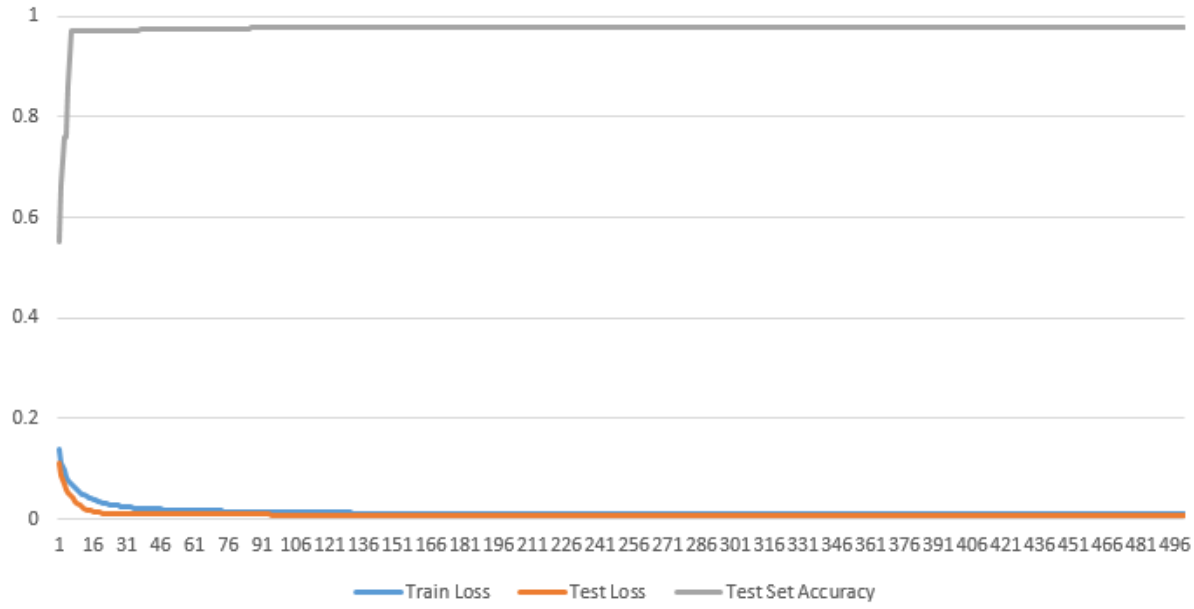



Σχήμα 32: Σφάλμα εκπαίδευσης (Train error) και σφάλμα ελέγχου (Test error) του βέλτιστου MLP πάνω στο Dataset 3.

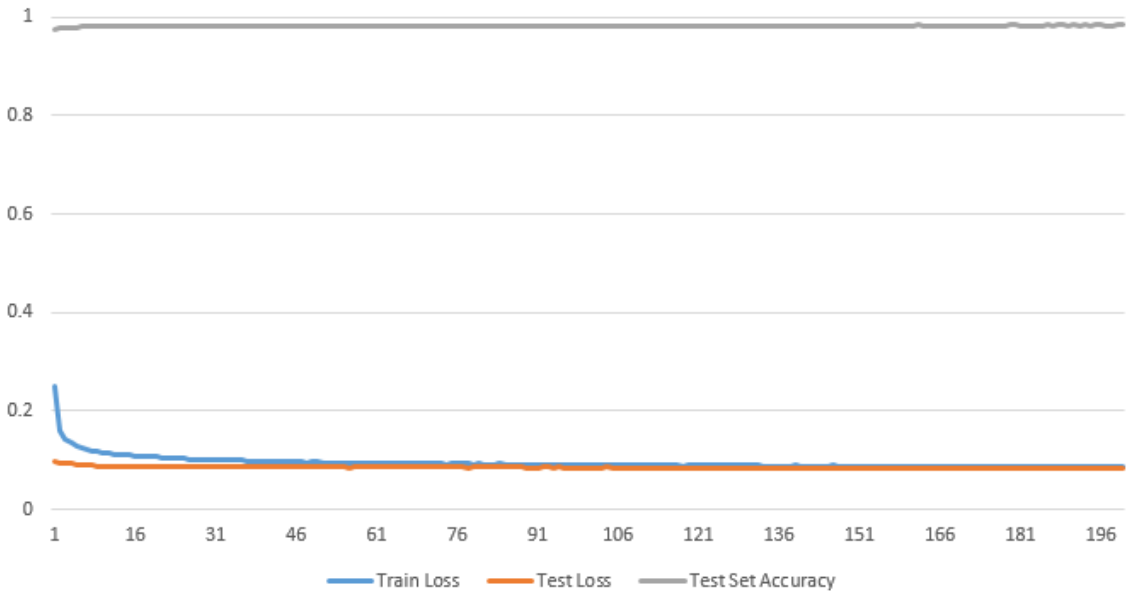
Το Νευρωνικό το οποίο έδωσε το αποτέλεσμα αυτό είχε 2 κρυφά επίπεδα και 30 Νευρώνες σε καθένα από αυτά. Στο σχήμα φαίνεται επίσης η ακρίβεια πάνω στο Test set. Φαίνεται το Overfitting που ξεκινάει από την $11^{η}$ εποχή και μετά. Μέχρι εκείνο το σημείο η Συνάρτηση Κόστους (Loss Function) τόσο πάνω στο Training Set όσο και στο Test Set μειώνονται και η ακρίβεια αυξάνεται. Από τη $11^{η}$ εποχή όμως το δίκτυο αλλάζει συμπεριφορά και ουσιαστικά όσο περισσότερο εκπαιδεύεται τόσο χαλάει η απόδοσή του. Αυτό οδήγησε στο συμπέρασμα πως ο λόγος για τον οποίον δεν πετύχαμε καλή ακρίβεια στο Dataset 3 είναι το Overfitting. Φαίνεται ότι πρέπει να γίνει πιο προσεκτική εκπαίδευση.

## 5.4 Σύγκριση Διαφορετικών Μεθόδων Εκπαίδευσης και Ρυθμών Μάθησης (Learning Rate) Πάνω στο Βέλτιστο MLP

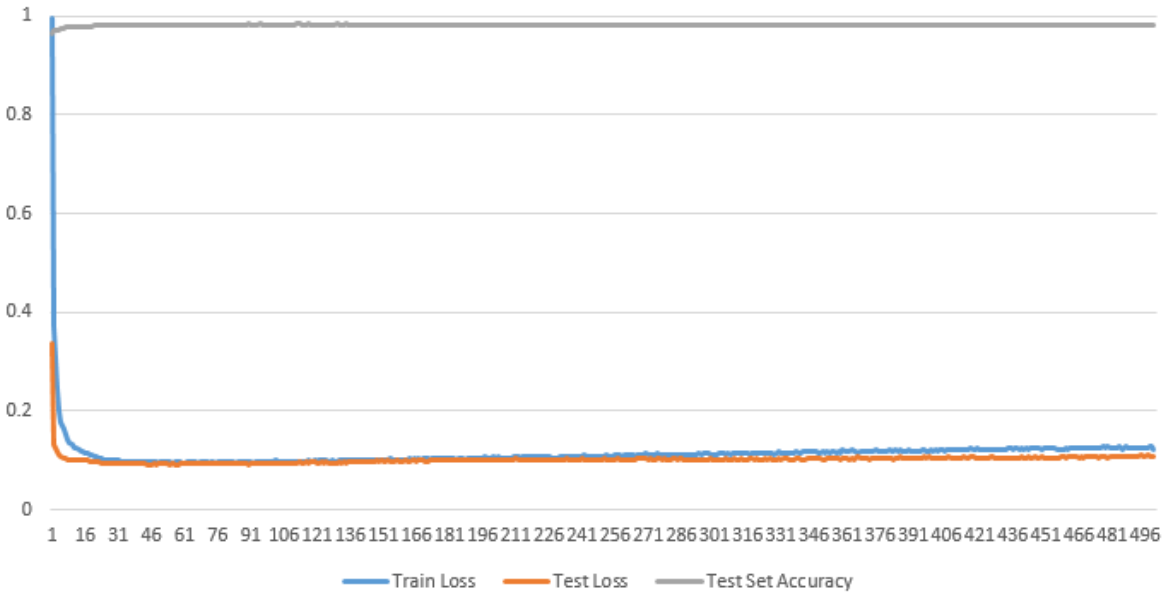
Στη συνέχεια παρουσιάζονται κάποιες γραφικές παραστάσεις του σφάλματος εκπαίδευσης και του σφάλματος ελέγχου συγκεκριμένα για το MLP δίκτυο με 2 κρυφά επίπεδα και 30 νευρώνες σε καθένα από αυτά, πάνω στο Dataset 0.
Δοκιμάστηκαν οι συναρτήσεις κόστους [βλ. Ενότητα 2.1.10] MSE και crossentropy, οι μέθοδοι εκπαίδευσης [βλ. Ενότητα 2.1.11] SGD, Adagrad και RMSprop.




Σχήμα 33: Σφάλμα εκπαίδευσης (Train error) και σφάλμα ελέγχου (Test error) του βέλτιστου MLP πάνω στο Dataset 0, Συνάρτηση Κόστους MSE, Μέθοδος εκπαίδευσης SGD, Dropout 0.4.
Πλεονέκτημα της εκπαίδευσης με SGD είναι, όπως φαίνεται, η ομαλή εκπαίδευση και η αποφυγή του Overfittng. Η ακρίβεια έφτασε στο 0.979%, λίγο χαμηλότερα από την εκπαίδευση με RMSprop, όπως βλέπουμε στην Εικόνα 35

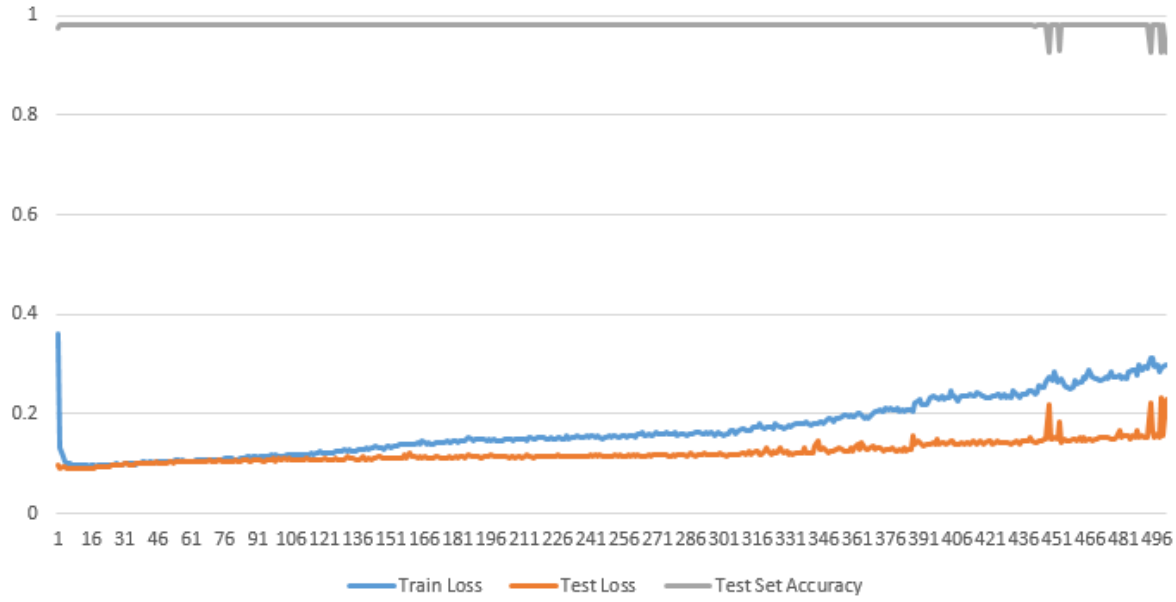



Σχήμα 34: Σφάλμα εκπαίδευσης (Train error) και σφάλμα ελέγχου (Test error) του βέλτιστου MLP πάνω στο Dataset 0, Συνάρτηση Κόστους Crossentropy, Μέθοδος εκπαίδευσης Adagrad, Dropout 0.4. Η εκπαίδευση με Adagrad έδωσε εξαιρετικά αποτελέσματα, παρόμοια με την RMSprop . Η ακρίβεια σταθεροποιήθηκε στο 0.983%.



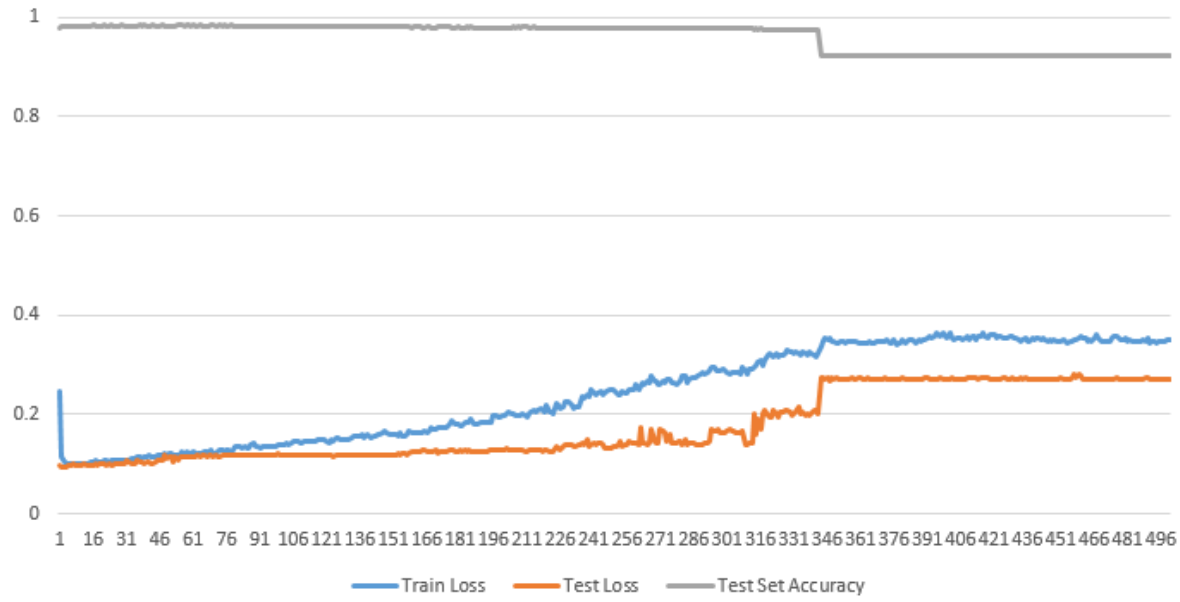

Σχήμα 35: Σφάλμα εκπαίδευσης (Train error) και σφάλμα ελέγχου (Test error) του βέλτιστου MLP πάνω στο Dataset 0, Συνάρτηση Κόστους Crossentropy, Μέθοδος εκπαίδευσης RMSprop με ρυθμό μάθησης 0.0001, Dropout 0.4.

Σε αυτή την Εικόνα, καθώς και στις δύο επόμενες, φαίνεται η επίδραση του Learning Rate0 στην εκπαίδευση του δικτύου. Σε αυτή την Εικόνα, όπου ο Ρυθμός Μάθησης είναι σχετικά μικρός, δεν υπάρχει Overfitting και η ακρίβεια φτάνει στο 0.983%.




Σχήμα 36: Σφάλμα εκπαίδευσης (Train error) και σφάλμα ελέγχου (Test error) του βέλτιστου MLP πάνω στο Dataset 0, Συνάρτηση Κόστους Crossentropy, Μέθοδος εκπαίδευσης RMSprop με ρυθμό μάθησης 0.0005, Dropout 0.4.0

Σε σύγκριση με την προηγούμενη Εικόνα 35, ο Ρυθμός Μάθησης έχει αυξηθεί. Ως αποτέλεσμα, η εκπαίδευση είναι λιγότερο ομαλή και η Accuracy δε παρουσιάσει ξαφνικές μεταβολές.




Σχήμα 37: Σφάλμα εκπαίδευσης (Train error) και σφάλμα ελέγχου (Test error) του βέλτιστου MLP πάνω στο Dataset 0, Συνάρτηση Κόστους Crossentropy, Μέθοδος εκπαίδευσης RMSprop με ρυθμό μάθησης 0.001, Dropout 0.4.

Σε συνέχεια των δύο προηγουμένων εικόνων, ο Learning Rate έχει αυξηθεί ακόμα περισσότερο και η αστάθεια στη συνάρτηση Κόστους και την Ακρίβεια είναι μεγαλύτερη.

### Συμπέρασμα όσον αφορά στη σύγκριση των Μεθόδων Εκπαίδευσης

Η εκπαίδευση με RMSProp οδήγησε γενικά σε λίγο καλύτερη ακρίβεια, αλλά το έκανε πολύ πιο γρήγορα. Όπως είναι φανερό από τις γραφικές παραστάσεις, από τις πρώτες κιόλας εποχές η τιμή της συνάρτησης κόστους φτάνει πολύ κοντά στην ελάχιστη τιμή της. Έτσι, η χρήση της κρίνεται προτιμητέα. Ωστόσο, φάνηκε ότι η εκπαίδευση με RMSProp είναι πολύ πιο ευαίσθητη στην τιμή του Learning Rate και χρειάστηκε να γίνει αρκετός πειραματισμός μέχρι να βρεθεί η βέλτιστη τιμή. Η εκπαίδευση με Adagrad οδήγησε σε εξίσου καλά με την RMSProp αποτελέσματα.

### Συμπέρασμα όσον αφορά το ρυθμό μάθησης (Learning rate)

Η τεχνική RMSProp οδηγεί μεν πολύ γρήγορη ελαχιστοποίηση της Cost Function, απαιτείται δε μία αρκετά μικρή τιμή στο ρυθμό μάθησης. Αυτό οφείλεται στη φύση των δεδομένων μας. Αυτά δεν έχουν συνεχείς τιμές και το πλήθος των features είναι σχετικά μεγάλο (79 για την ακρίβεια). Γι' αυτούς τους λόγους, η μάθηση με μικρά «βήματα» κρίνεται απαραίτητη.

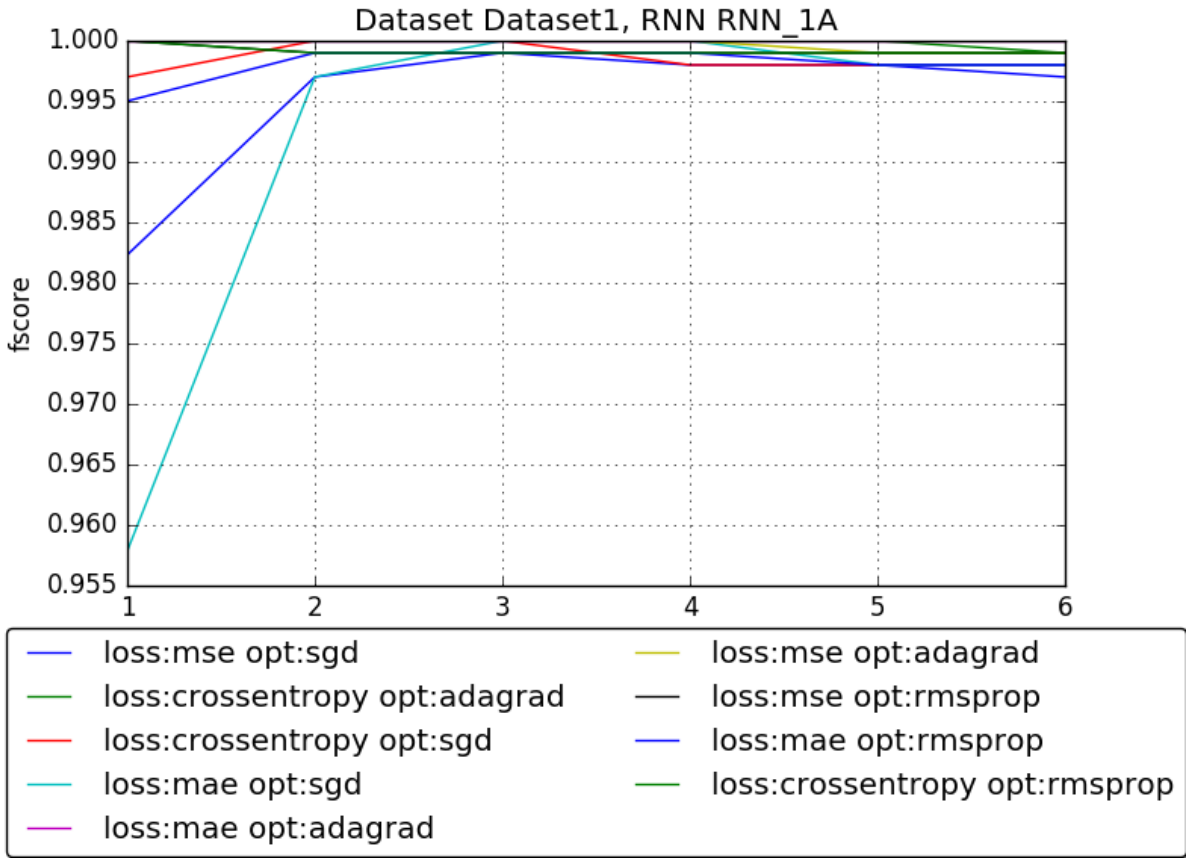
## 5.5 Σύγκριση Διαφορετικών Δομών Αναδρασιακών Νευρωνικών Δικτύων (RNN)

Στη συνέχεια έγινε χρήση των Αναδρασιακών Νευρωνικών δικτύων (RNN ), κυρίως για να λύσουμε το πρόβλημα χαμηλής ακρίβειας του Dataset 3, αλλά και για να εξεταστούν οι δυνατότητες των RNN .
Από τα είδη ανάδρασης που περιγράφονται στην Ενότητα 2.1.6 χρησιμοποιήθηκε η ανάδραση τύπου 1, από κρυφό επίπεδο προς κρυφό επίπεδο.
Δοκιμάστηκαν συγκεκριμένα 3 βασικές αρχιτεκτονικές δικτύων, με 2, 3 και 4 επίπεδα, σύμφωνα με τον επόμενο πίνακα.

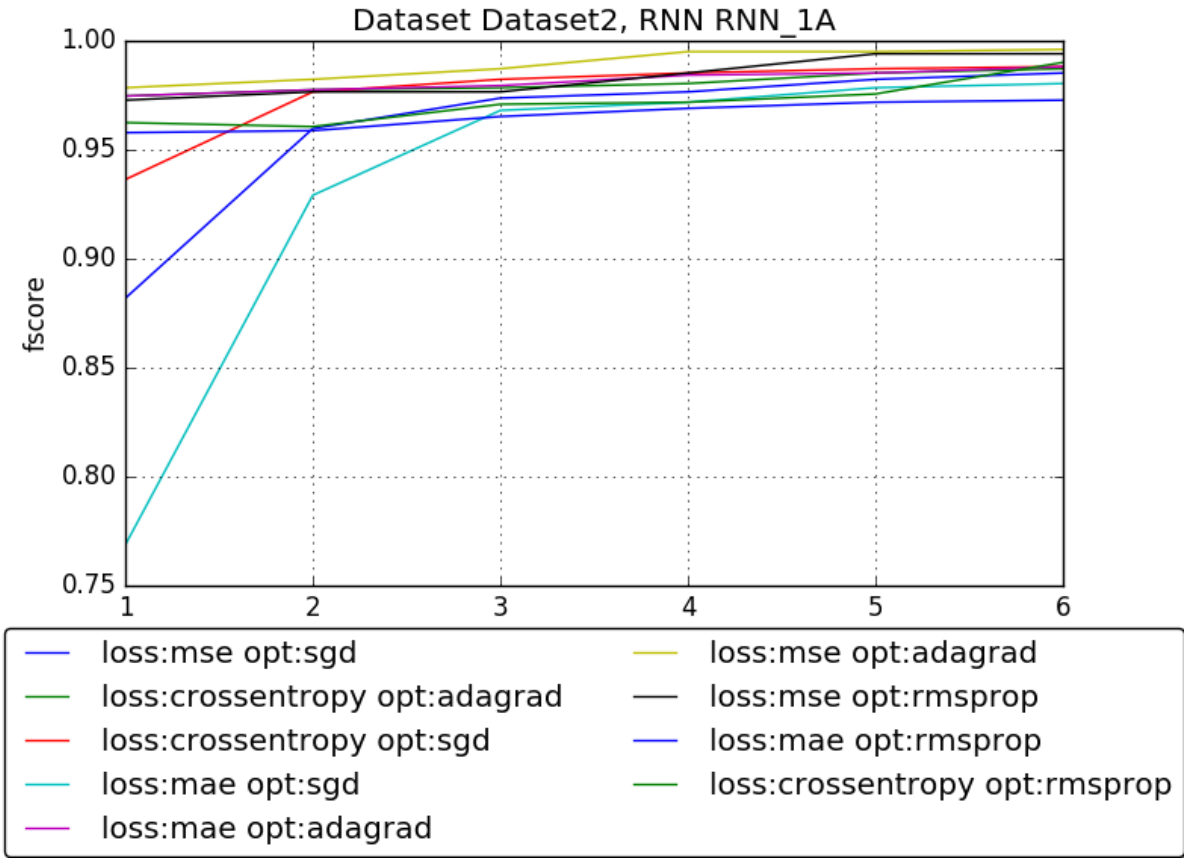
| **Όνομα δικτύου** | **Συνολικό πλήθος επιπέδων** | **Νευρώνες σε κάθε επίπεδο** | **Συναρτηση Ενεργοποίησης** |
|---|---|---|---|
| RNN 1.1 | 1 | 50 | relu |
| RNN 1.2 | 1 | 50 | sigmoid |
| RNN 2.1 | 2 | 50,50 | relu |
| RNN 2.2 | 2 | 50,50 | sigmoid |
| RNN 3.1 | 3 | 50,40,30 | relu |
| RNN 3.2 | 3 | 50,40,30 | sigmoid |

Πίνακας 4: Τα δίκτυα RNN που υπολοιήθηκαν και δοκιμάστηκαν.
Σημειώνεται ότι σε όλα υπήρχε ένα ακόμα επίπεδο, το επίπεδο εξόδου, που περιείχε 1 ή 5 νευρώνες, ανάλογα με το πλήθος των κλάσεων που υπάρχουν στο Dataset , και συνάρτηση ενεργοποίησης Σιγμοειδή ή softmax, για binary και multiclass Ταξινόμηση αντίστοιχα.

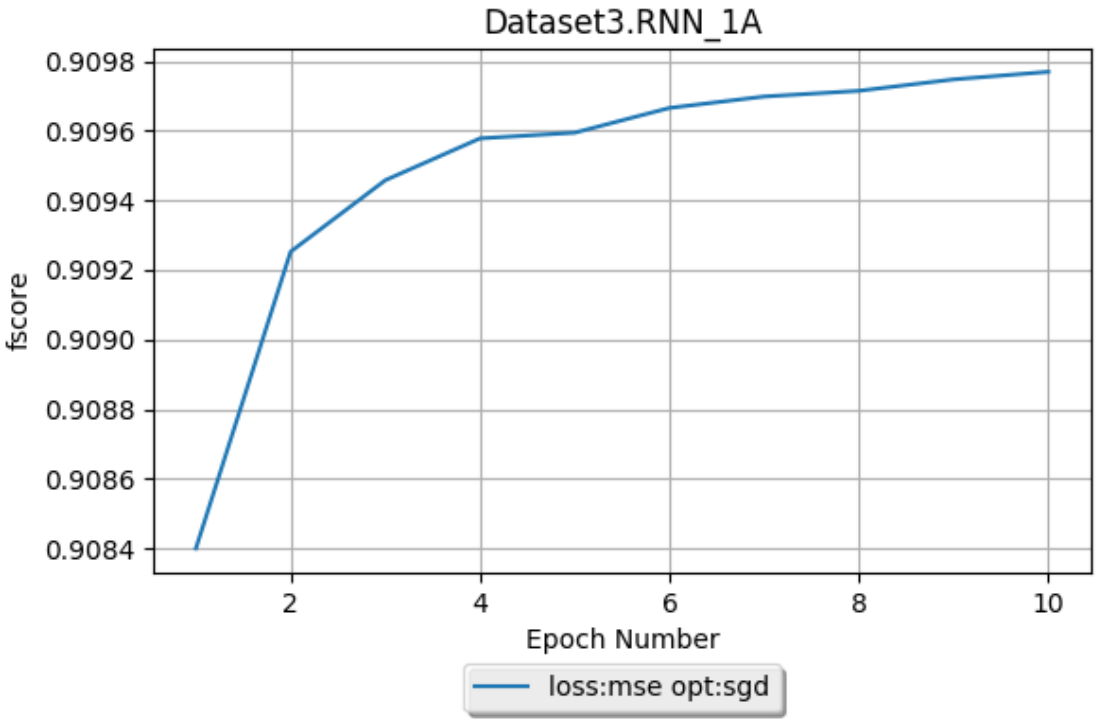
Για τα ανωτέρω δίκτυα δοκιμάστηκαν πολλοί διαφορετικοί συνδυασμοί Συναρτήσεων Κόστους και μεθόδων Εκπαίδευσης. Όλα τα αποτελέσματα φαίνονται στις ακόλουθες Γραφικές Παραστάσεις.




Σχήμα 38: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 1.1 πάνω στο Dataset 1 Όπως φαίνεται στη Γραφική, δοκιμάστηκαν διαφορετικοί συνδυασμοί Συνάρτησης Κόστους και Μεθόδου Εκπαίδευσης. Όλες συνέκλιναν σε ακρίβεια πάνω από 99% από τη 2η ήδη εποχή. Όπως συνέβη και με τα MLP , έτσι και τα RNN έμαθαν με πολύ καλή ακρίβεια το Dataset 1 (που περιέχει, όπως περιγράφεται στην Ενότητα 4.2, ομαλή κίνηση και κίνηση από την επίθεση SYN Flood ). Μάλιστα, φαίνεται ότι ένα μόνο επίπεδο με ανάδραση αρκεί για να διακρίνει το Δίκτυο μεταξύ των δύο αυτών ειδών κίνησης.

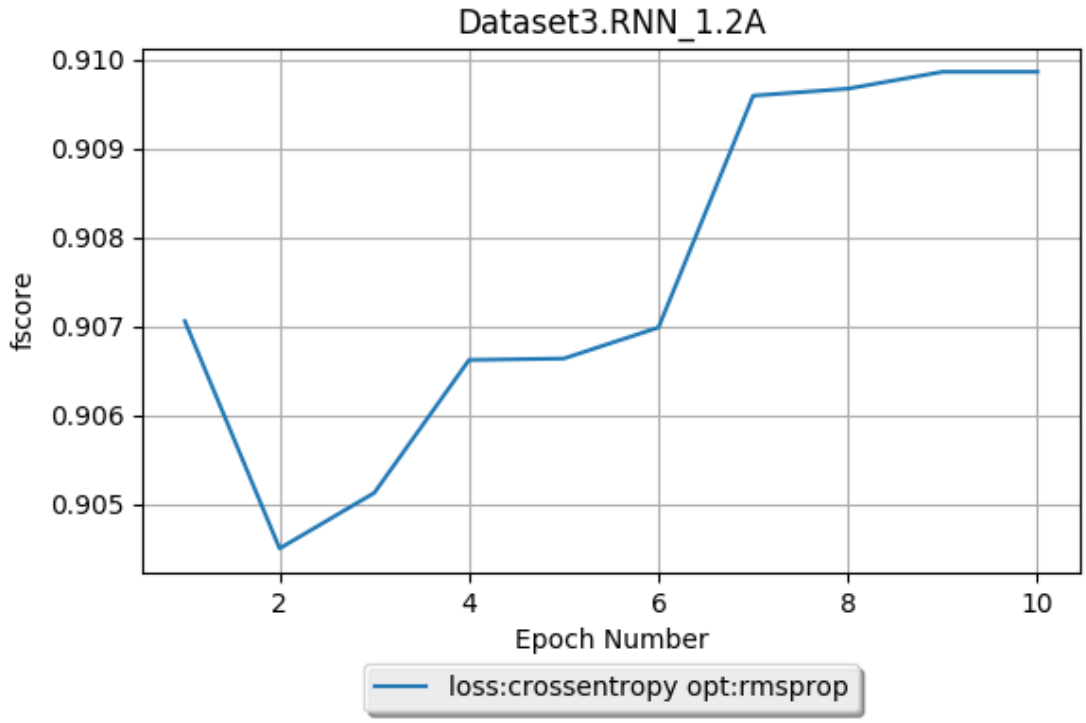



Σχήμα 39: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 1.1 πάνω στο Dataset 2 Όπως φαίνεται στη Γραφική, δοκιμάστηκαν διαφορετικοί συνδυασμοί Συνάρτησης Κόστους και Μεθόδου Εκπαίδευσης. Και πάλι πήραμε πολύ καλή ακρίβεια.



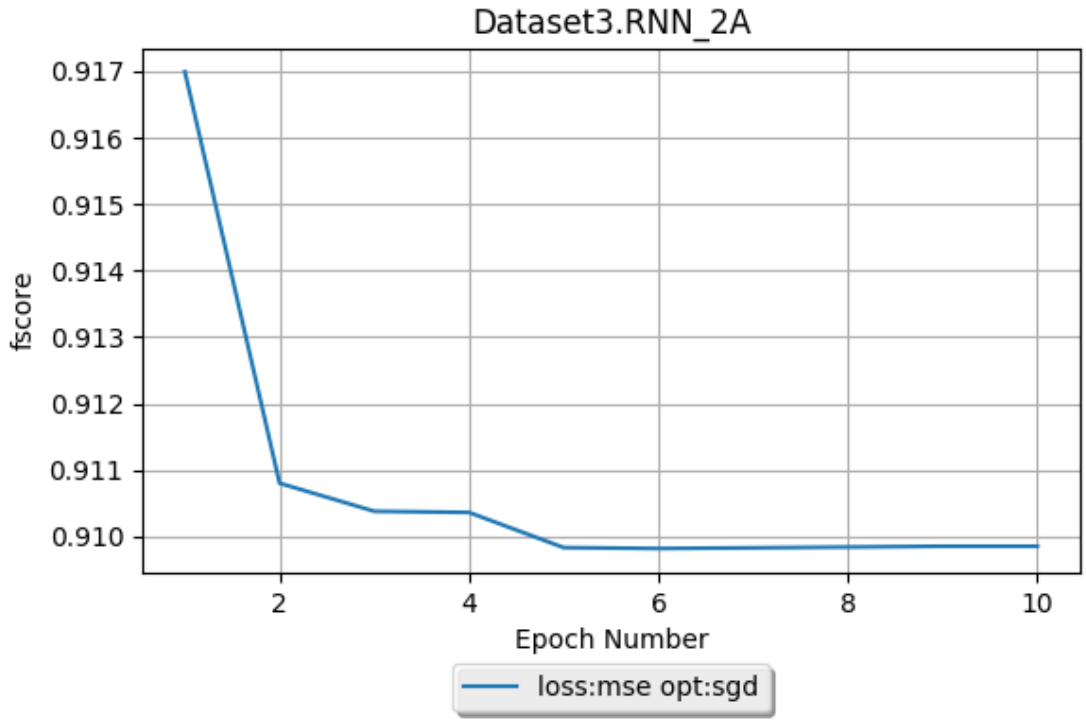

Σχήμα 40: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 1.1 πάνω στο Dataset 3, με Συνάρτηση Κόστους MSE και Μέθοδο Εκπαίδευσης SGD

Όπως περιγράφηκε στον Πίνακα 4, το RNN 1.1 είχε ως συνάρτηση ενεργοποίησης στο επίπεδο εξόδου τη ReLU. Βλέπουμε ότι το RNN 1.1 δίκτυο έφτασε το 91% ακρίβεια, σαφώς καλύτερη από την επίδοση που είχαμε με τα MLP δίκτυα. Ένα συμπέρασμα που προκύπτει από εδώ είναι ότι η Συνάρτηση Ενεργοποίησης ReLU αποδίδει με Συνάρτηση Κόστους MSE και Μέθοδο Εκπαίδευσης SGD.



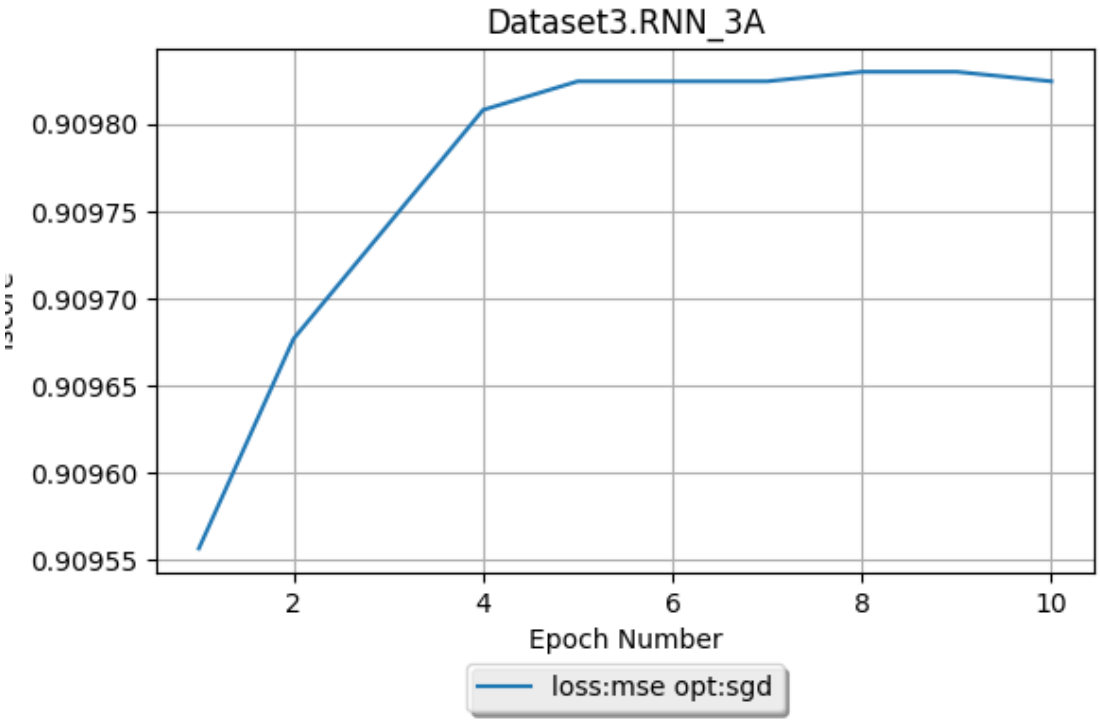

Σχήμα 41: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 1.2 πάνω στο Dataset 3, με Συνάρτηση Κόστους Crossentropy και Μέθοδο Εκπαίδευσης RMSProp

Όπως περιγράφηκε στον Πίνακα 4, το RNN 1.2 είχε ως συνάρτηση ενεργοποίησης στο επίπεδο εξόδου τη Sigmoid. Βλέπουμε ότι και το RNN 1.2 δίκτυο έφτασε το 91% ακρίβεια. Ένα συμπέρασμα που προκύπτει από εδώ είναι ότι η Συνάρτηση Ενεργοποίησης Sigmoid αποδίδει με Συνάρτηση Κόστους Crossentropy και Μέθοδο Εκπαίδευσης RMSProp.



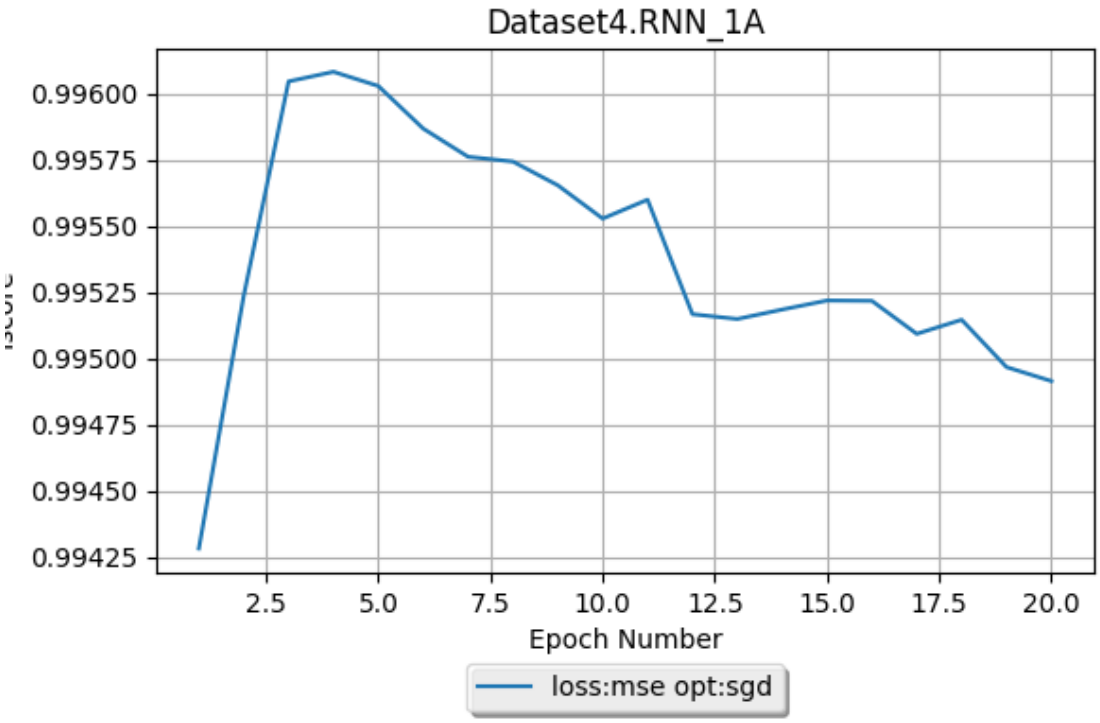

Σχήμα 42: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 2.1 πάνω στο Dataset 3, με Συνάρτηση Κόστους MSE και Μέθοδο Εκπαίδευσης SGD

Όπως περιγράφηκε στον Πίνακα 4, το RNN 2.1 είχε 2 αναδρασιακά επίπεδα και ως συνάρτηση ενεργοποίησης στο επίπεδο εξόδου τη ReLU. Η ακρίβεια στην οποία συνέκλινε πάντως δεν παρουσίασε βελτίωση σε σχέση με το RNN 1.1. Φαίνεται πάντως και εδώ να υπάρχει ένα μικρό Overfitting στο Dataset 3, αφού στις 2 πρώτες εποχές εκπαίδευσης η ακρίβεια στο Test Set είναι λίγο μεγαλύτερη απ΄ ότι στις επόμενες.

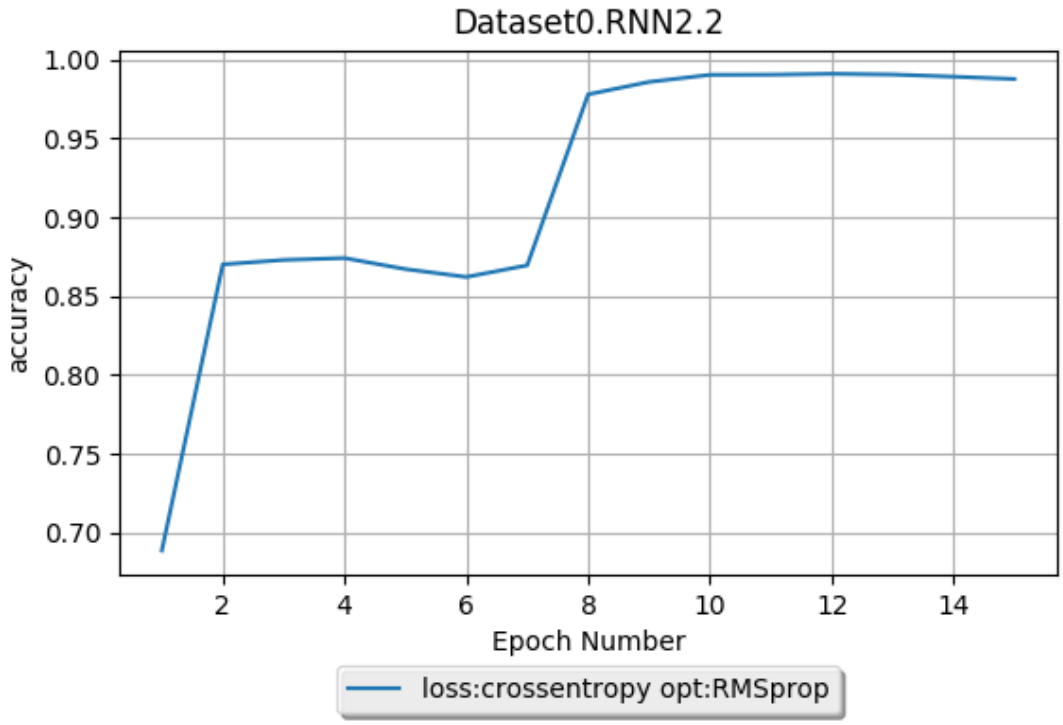



Σχήμα 43: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 3.1 πάνω στο Dataset 3, με Συνάρτηση Κόστους MSE και Μέθοδο Εκπαίδευσης SGD

Όπως και στις δύο προηγούμενες Γραφικές, η ακρίβεια που φτάσαμε στο Dataset 3 ήταν 91%, αλλά όχι περισσότερο.




Σχήμα 44: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 1.1 πάνω στο Dataset 4 Όπως συνέβη και με τα Dataset 1 και 2, έτσι και το 4 αναγωρίστηκε με πολύ καλή ακρίβεια από τα RNN δίκτυα.




Σχήμα 45: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του RNN 2.2 πάνω στο Dataset 0 Στο Σχήμα φαίνεται η Ακρίβεια των προβλέψεων πάνω στο Συνόλου Ελέγχου (Test Set) ανά εποχή εκπαίδευσης. Η εκπαίδευση έγινε για 15 εποχές. Κατά την 7η με 8η εποχή βλέπουμε μία σημαντική αύξηση στην Ακρίβεια, που τελικά φτάνει στο 99%.

## 5.6 Συμπεράσματα από RNN - Βέλτιστο Δίκτυο

### Συμπεράσματα

Βλέπουμε ότι στα Dataset 1, 2, 4, και 5 το F-Score ήταν ιδιαίτερα υψηλό, πάνω από 99%. Όπως και τα MLP δίκτυα, τα τρία RNN που δοκιμάστηκαν είναι σε θέση να «μάθουν» και να κατηγοριοποιήσουν (Classify) με εξαιρετική ακρίβεια τα εν λόγω Dataset . Όσον αφορά στο Dataset 3, βλέπουμε και τα τρία Δίκτυα που υλοποιήθηκαν να φτάνουν σε ακρίβεια περίπου 91%.

Αξίζει εδώ να σημειωθεί ότι, στα εν λόγω πειράματα, έγινε υπολογισμός όχι μόνο της ακρίβειας (Accuracy) αλλά και του F-Score.

### Βέλτιστο RNN δίκτυο

Το F-Score ήταν ιδιαίτερα υψηλό για όλα τα δίκτυα. Βέλτιστο όμως ήταν το RNN 2.1, με συνάρτηση ενεργοποίησης ReLU, συνάρτηση Κόστους Crossentropy και Βελτιστοποιητή RMSprop.

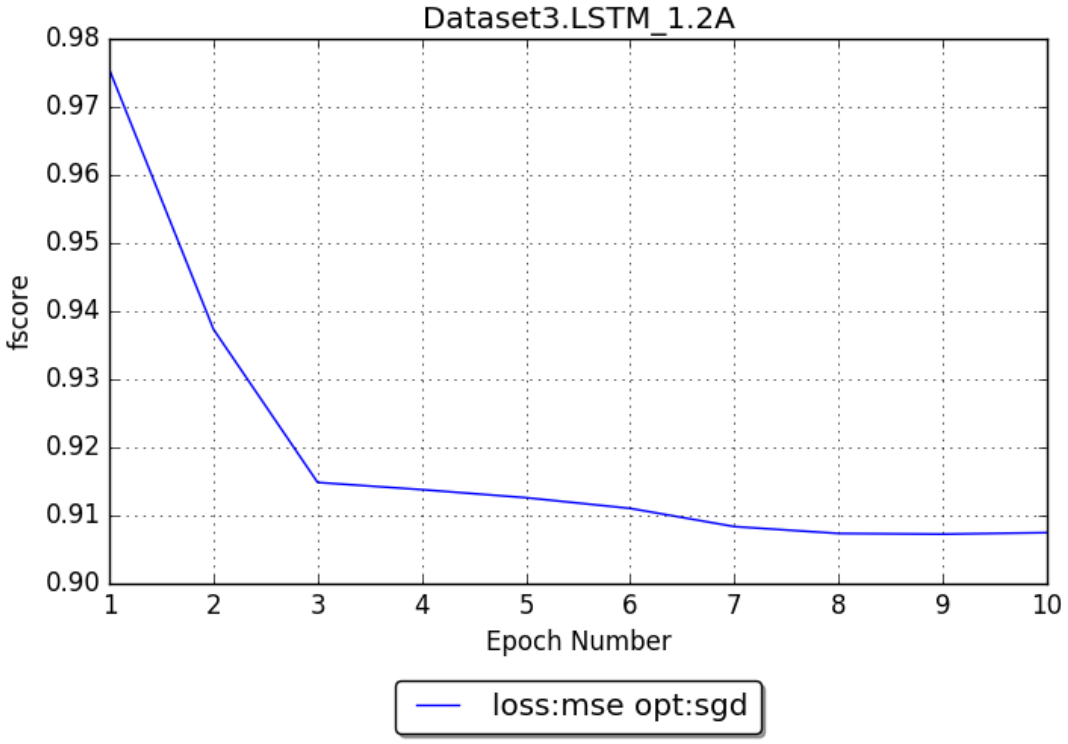
| **Κρυφά επίπεδα** | **Νευρώνες ανά επίπεδο** | **Συνάρτηση ενεργοποίησης** | **Συναρτηση Κόστους - Μέθοδος εκπαίδευσης** |
|---|---|---|---|
| 2 | 50 | ReLU | Crossentropy - RMSprop |

Πίνακας 5: Το RNN δίκτυο που επιλέχθηκε ως βέλτιστο

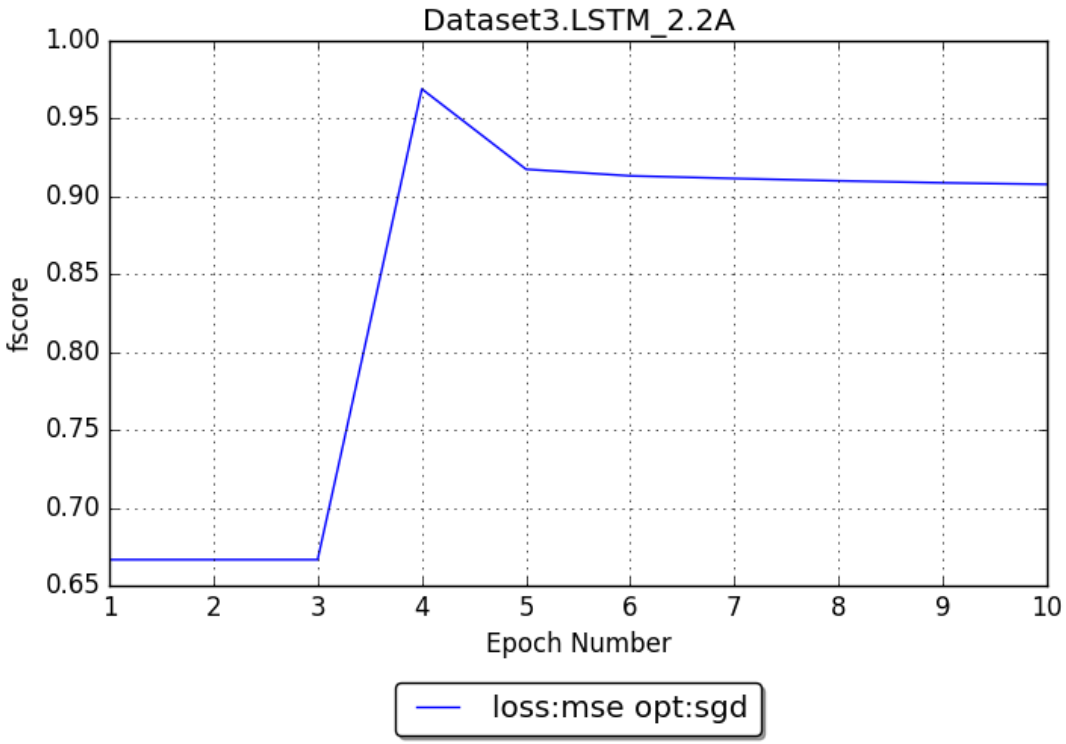
## 5.7 Σύγκριση Διαφορετικών Δικτύων LSTM - Συμπεράσματα και Βέλτιστο Δίκτυο

Τέλος, ακριβώς τα ίδια δίκτυα που περιγράφονται στον Πίνακα 4 υλοποιήθηκαν με δομικές μονάδες (νευρώνες) LSTM . Και πάλι έγινε εκπαίδευση και έλεγχος για διάφορες συναρτήσεις Κόστους και μεθόδους Βελτιστοποίησης. Επικεντρωθήκαμε κυρίως στα Dataset 3 (που παρουσίαζε χαμηλή απόδοση σε MLP και LSTM ) και 0.

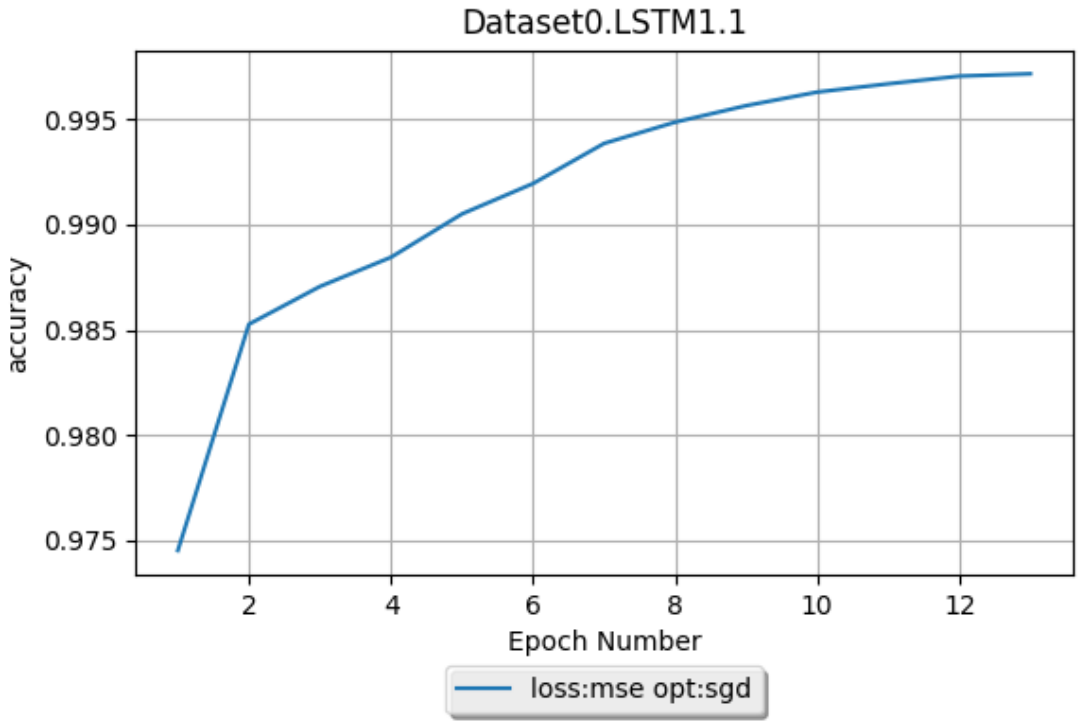
Ακολουθούν οι γραφικές παραστάσεις.



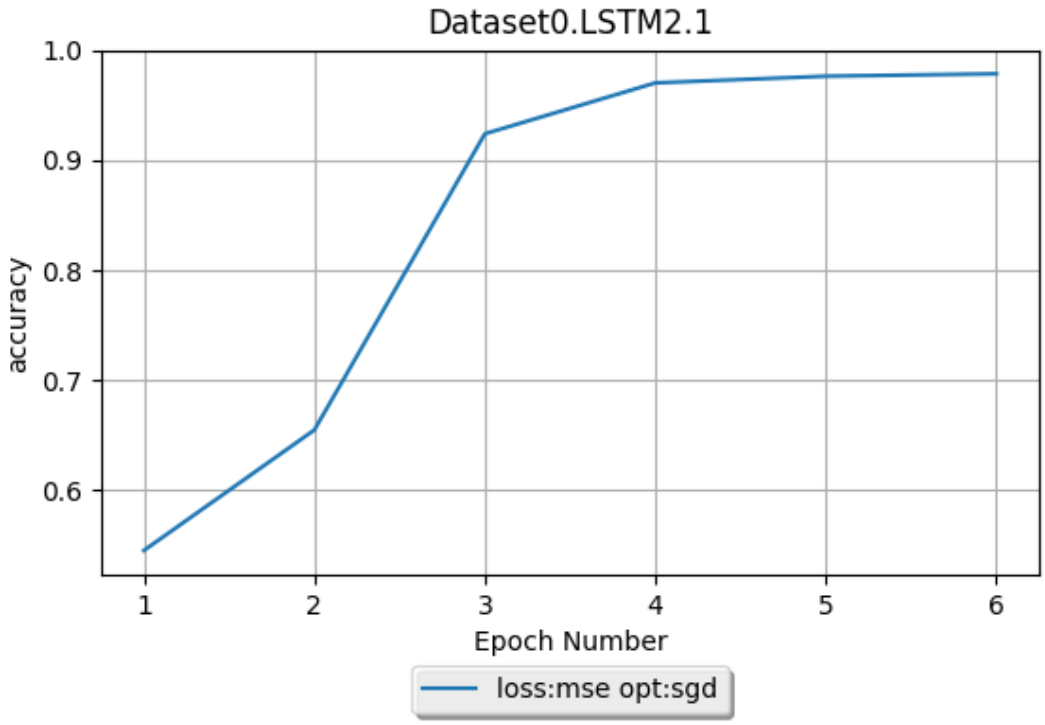

Σχήμα 46: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του LSTM 1.2 πάνω στο Dataset 3 Το LSTM 1.2 δεν έδωσε την αναμενόμενη βελτίωση στο Dataset 3. Ενώ στην 1η εποχή βλέπουμε να έχει πιάσει μία ακρίβεια της τάξης του 97.5%, τελικά συνεκλινε και αυτό στο 91%, όπως συνέβη και με τα RNN . Φαίνεται ότι απαιτείται να εφαρμοστεί η τεχνική του Early Stopping για να πετύχουμε μία καλή ακρίβεια στο Dataset 3.




Σχήμα 47: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του LSTM 2.2 πάνω στο Dataset 3 Και από εδώ φαίνεται ότι το Overfitting στο Dataset 3 συνεχίζει να υφίσταται. Η μέγιστη ακρίβεια την οποία θα μπορούσαμε να πετύχουμε είναι 96%.




Σχήμα 48: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του LSTM 1.1 πάνω στο Dataset 0 Στο Dataset 0 (που περιείχε όλων των ειδών κίνηση) το LSTM 1.1 έφτασε σε ακρίβεια το 99.5%. Η εκπαίδευση εδώ είναι ομαλή, με την ακρίβεια να αυξάνεται σε κάθε εποχή και να φτάνει στο 99.7%. Ένα σημαντικό μειονέκτημα των LSTM είναι, ωστόσο, οι απαιτήσεις σε μνήμη και χρόνο κατά την εκπαίδευση, που είναι σημαντικά μεγαλύτερες από τα RNN δίκτυα. Υπενθυμίζεται εδώ από την Ενότητα 2.1.7 ότι τα LSTM έχουν συνολικά 4πλάσιο πλήθος παραμέτρων σε σχέση με τα RNN .




Σχήμα 49: Γραφική παράσταση Ακρίβειας ανά εποχή εκπαίδευσης του LSTM 2.1 πάνω στο Dataset 0 Όπως φαίνεται στη γραφική, και το LSTM 2.1 πέτυχε ακρίβεια 98% από την 7η ήδη εποχή. Οι απαιτήσεις σε μνήμη και χρόνο εκπαίδευσης κατέστησαν, όμως, δυσκολη την εκπαίδευση για πολλές εποχές εποχές ή τον πειραματισμό με διάφορες τιμές παραμέτρων.

**Συμπεράσματα**

Ο χρόνος εκπαίδευσης ήταν στα LSTM φανερά μεγαλύτερος από τα δύο προηγούμενα είδη. Επίσης, τα αποτελέσματα ήταν συγκρίσιμα με την περίπτωση των RNN Δικτύων. Για το λόγο αυτό, κρίνεται ότι τα RNN μας καλύπτουν. Ωστόσο, μιλάμε πάντα για τα συγκεκριμένα Dataset . Η καλύτερη δυνατότητα γενίκευσης που έχουν τα LSTM σε σχέση με τα RNN θα μπορούσε να αξιοποιηθεί σε τυχόν πιο σύνθετο Dataset .

**Βέλτιστο LSTM δίκτυο**

Το LSTM που κρίνεται ως βέλτιστο έχει την ίδια δομή με το βέλτιστο RNN , δηλαδή το LSTM 2.1 με συνάρτηση ενεργοποίησης ReLU, συνάρτηση Κόστους Crossentropy και Βελτιστοποιητή RMSprop.

| **Κρυφά επίπεδα** | **Νευρώνες ανά επίπεδο** | **Συνάρτηση ενεργοποίησης** | **Συναρτηση Κόστους - Μέθοδος εκπαίδευσης** |
|---|---|---|---|
| 2 | 50 | ReLU | Crossentropy - RMSprop |

Πίνακας 6: Το LSTM δίκτυο που επιλέχθηκε ως βέλτιστο
Δομικά είναι ακριβώς ίδιο με το Βέλτιστο RNN (βλ. Σχήμα 5). Μόνη διαφορά είναι τα δομικά του στοιχεία, που δεν είναι οι κλασικοί νευρώνες αλλά LSTM κύτταρα.

## 5.8 Έλεγχος των Προηγουμένων Αποτελεσμάτων Σε Νέο Αρχείο Καταγραφής

Για περαιτέρω έλεγχο, χρησιμοποιήθηκε ένα έξτρα αρχείο καταγραφής, το οποίο ήταν τελείως άγνωστο στο δίκτυό μας. Προήλθε και αυτό από Port Mirroring πάνω σε έναν Μεταγωγέα του ΕΜΠ.
Επισημαίνεται ότι το αρχείο καταγραφής αυτό περιείχε μόνο Ομαλή, Legitimate κίνηση. Τα Νευρωνικά δίκτυα που δοκιμάστηκαν ήταν ένα MLP , ένα RNN και ένα LSTM και είχαν ήδη εκπαιδευτεί πάνω στο Dataset 0. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 7. Το νέο αυτό αρχείο αποτελεί ένα καλό κριτήριο για τη δυνατότητα γενίκευσης του νευρωνικού μας.

| **Νευρωνικό Δίκτυο** | **Ακρίβεια προβλέψεων** |
|---|---|
| MLP με 2 κρυφά επίπεδα, 30 νευρώνες ανά επίπεδο, Συνάρτηση κόστους crossentropy, Βελτιστοποίηση rmsprop | 0.983 |
| RNN με 2 επίπεδα των 50 νευρώνων, Συνάρτηση κόστους MSE, Βελτιστοποίηση SGD | 0.997 |
| LSTM με 2 επίπεδα των 50 νευρώνων, Συνάρτηση κόστους MSE, Βελτιστοποίηση SGD | 0.977 |

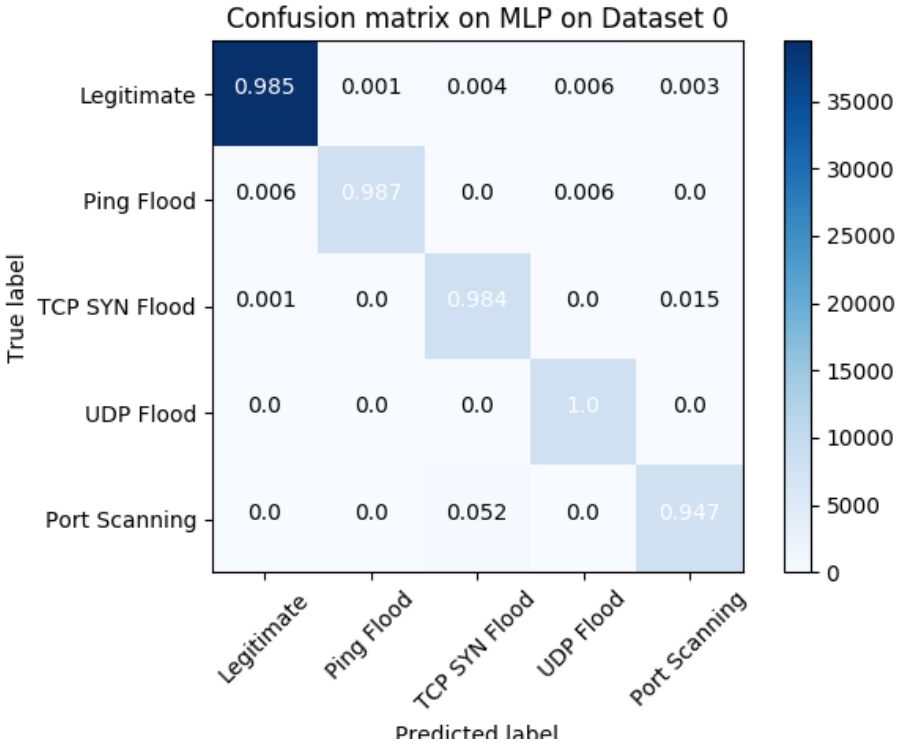
Πίνακας 7: Αποτελέσματα (Ακρίβεια) για τα 3 βέλτιστα δίκτυα (ένα MLP , ένα RNN και ένα LSTM ) πάνω σε ένα νέο αρχείο καταγραφής με Ομαλή Κίνηση.

**Συμπέρασμα**

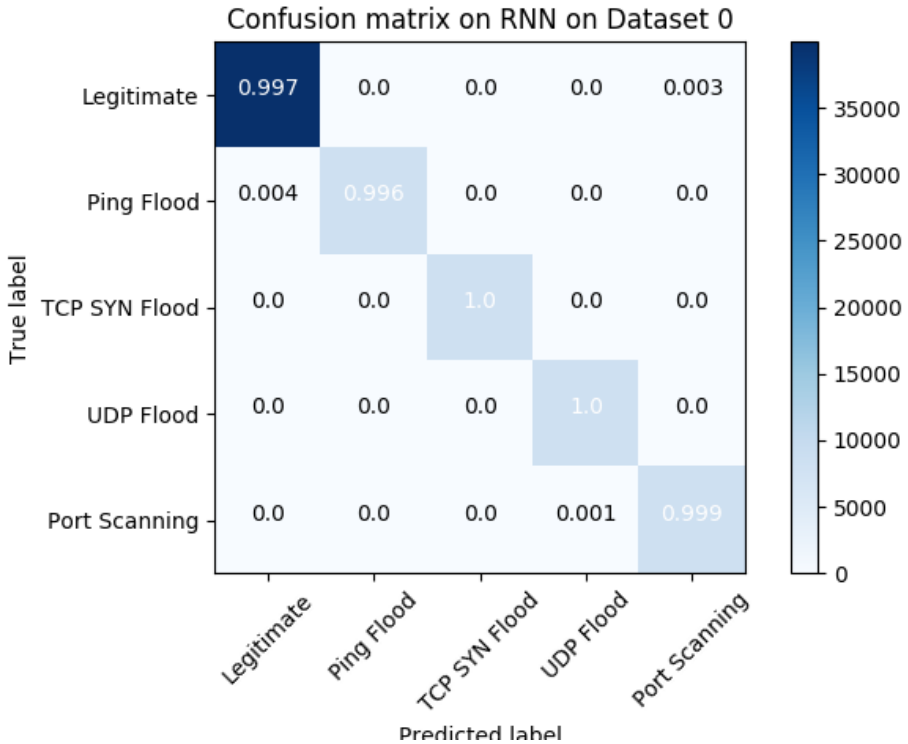
Από αυτό το νέο Dataset , που περιείχε βέβαια μόνο καλόβουλη κίνηση, αποδείχθηκε ότι τα δίκτυά μας είχαν εκπαιδευεί σωστα, χωρίς Overfitting και έχει καλή δυνατότητα γενίκευσης.

## 5.9 Αξιολόγηση του Συστήματος με Βάση τον Πίνακα Σύγχυσης (Confusion Matrix) και τα False Positives.

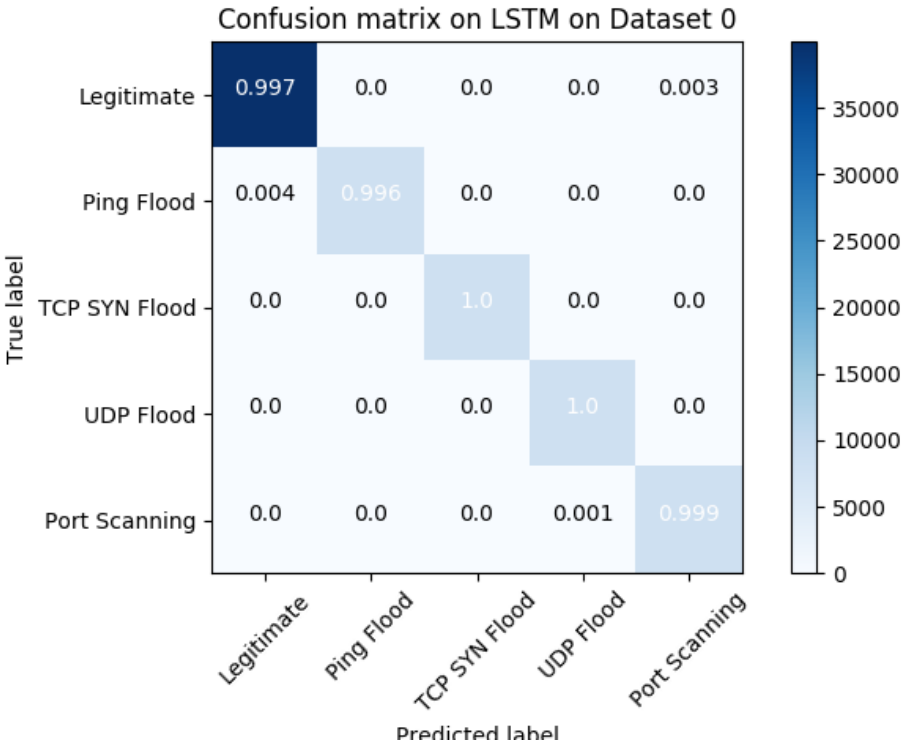
Ακολουθούν τρεις γραφικές παραστάσεις που αφορούν στον Confusion Matrix των 3 βέλτιστων δικτύων (όπως αναφέρονται στην Ενότητα 5.10) πάνω στο Dataset 0 (βλ. Ενότητα 4.2).




Σχήμα 50: Πίνακας Σύγχυσης (Confusion Matrix) για το βέλτιστο MLP πάνω στο Dataset 0.




Σχήμα 51: Πίνακας Σύγχυσης (Confusion Matrix) για το βέλτιστο RNN πάνω στο Dataset 0.




Σχήμα 52: Πίνακας Σύγχυσης (Confusion Matrix) για το βέλτιστο LSTM πάνω στο Dataset 0. Από εδώ επιβεβαιώνεται αυτό που αναφέρθηκε και στην Ενότητα 5.10, ότι το LSTM δίκτυο δεν οδήγησε σε περαιτέρω βελτίωση, σε σύγκριση με το RNN. Σημειώνεται, βέβαια, ότι όλα τα συμπεράσματα αφορούν σε εκπαίδευση πάνω στα συγκεκριμένα Dataset και σε αναγνώριση των συγκεκριμένων επιθέσεων.

## 5.10 Συνολικά Συμπεράσματα και Βέλτιστα Δίκτυα

### Συνολικά Συμπεράσματα

1. Αρχικά, φάνηκε ότι δε χρειάζονται εξεζητημένες μετρικές για να ανιχνευτούν οι συγκεκριμένες επιθέσεις. Είδαμε ότι τα χαρακτηριστικά που μπορεί να παράξει ένας (Flow Extractor) είναι αρκετά για να γίνει η Κατηγοριοποίηση (Classification) της κίνησης.

2. Όλα τα δίκτυα που κρίθηκαν ως βέλτιστα είχαν 2 κρυφά επίπεδα. ένα δίκτυο με 2 κρυφά επίπεδα μπορεί (για τα συγκεκριμένα Dataset τουλάχιστον) να μάθει πολύ καλά τα Δεδομένα Εκπαίδευσης, άρα δε φαίνεται να υπάρχει λόγος να χρησιμοποιηθούν πολύ βαθιές αρχιτεκτονικές (με 4 και πάνω επίπεδα).

3. Η μετάβαση από MLP σε RNN είχε ως συνέπεια αύξηση στο χρόνο εκπαίδευσης αλλά και βελτίωση της ακρίβειας πρόβλεψης. Αντίθετα, η μετάβαση σε LSTM δεν είχε ως συνέπεια αντίστοιχη αύξηση στην ακρίβεια των προβλέψεων.

4. Όσον αφορά στο **χρόνο εκπαίδευσης**, δεν ήταν σε καμία περίπτωση τόσο μεγάλος που να κάνει στο σύστημά μας μη-εφαρμόσιμο στην πράξη. Χρονικά είναι δυνατό ακόμα και να γίνει επανεκπαίδευση με βάση νέα δεδομένα κίνησης. Σε περίπτωση εφαρμογής ενός συστήματος σαν αυτό που δοκιμάστηκε σε κάποιον host ή σε Δρομολογητή, η εκπαίδευση μπορεί να γίνει παράλληλα, χωρίς να χρειαστεί να διακοπεί η λειτουργία του συστήματος.

5. Όσον αφορά τους υπολογιστικούς πόρους, επαληθεύτηκε αυτό που αναφέρθηκε στην Ενότητα **;;**, ότι απαιτείται υπολογισμός 4-πλάσιου πλήθους παραμέτρων, άρα και αισθητά μεγαλύτερη μνήμη κατά την εκπαίδευση. Αυτό αποτελεί σηματνικό πρόβλημα, αν επιθυμούμε τη χρήση του συστήματος σε ρεαλιστικά σενάρια.

Στον επόμενο πίνακα παρουσιάζονται συνολικά τα τρία βέλτιστα δίκτυα (βέλτιστο MLP , βέλτιστο RNN και βέλτιστο LSTM ).

| **Είδος δικτύου** | **Κρυφά επίπεδα** | **Νευρώνες ανά επίπεδο** | **Συνάρτηση ενεργοποίησης** | **Dropout Rate** | **Συναρτηση Κόστους - Μέθοδος εκπαίδευσης** |
|---|---|---|---|---|---|
| MLP | 2 | 30 | ReLU [1] | 0.4 | Crossentropy - RMSprop |
| RNN | 2 | 50 | ReLU | 0 [2] | Crossentropy - RMSprop |
| LSTM | 2 | 50 | ReLU | 0 | Crossentropy - RMSprop |

Πίνακας 8: **Τα δίκτυα (μεταξύ όλων των MLP , RNN και LSTM ) που επιλέχθηκαν ως βέλτιστα.**

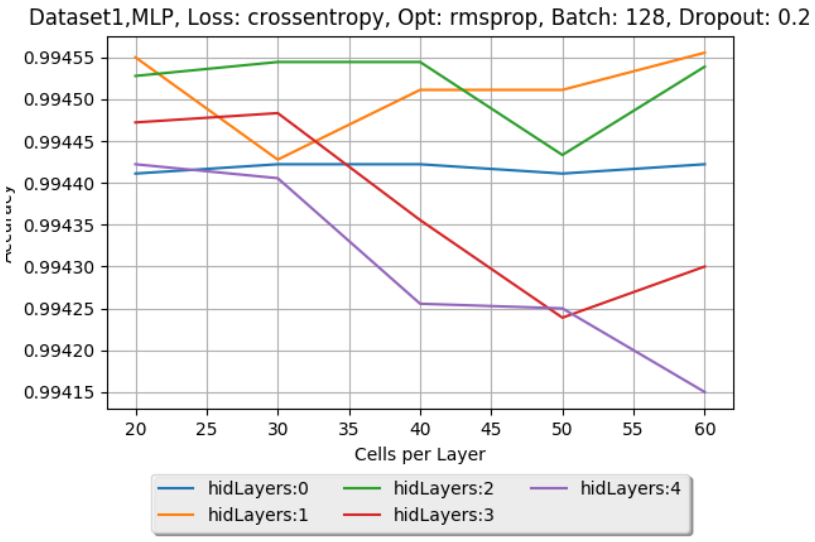
# 6 Μελλοντική Δουλειά

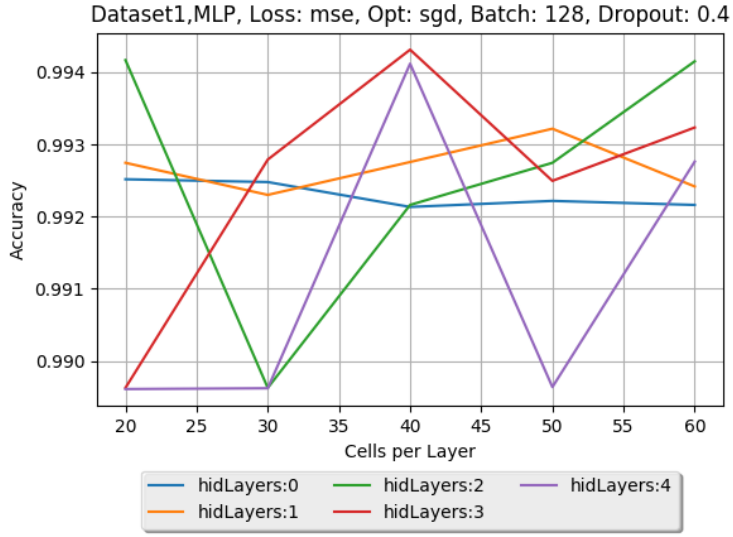
Ως μελλοντική δουλειά, θα θέλαμε να δοκιμάσουμε την προσθήκη ενός ακόμη χαρακτηριστικού (Attribute) στη είσοδο των Νευρωνικών Δικτύων, ενός χαρακτηριστικού που θα σχετίζεται τις διευθύνσεις IP προέλευσης και προορισμού των πακέτων.
Όπως γράψαμε στην Ενότητα 4.3, οι διευθύνσεις προέλευσης και προορισμού των πακέτων δε δόθηκαν ως είσοδος στα Νευρωνικά Δίκτυα που εξετάστηκαν. Όμως, αυτές είναι αρκετά σημαντικές για την ανίχνευση επιθέσεων όπως η SYN Flood. Για παράδειγμα, η ύπαρξη ενός μεγάλου πλήθους Ροών που κατευθύνονται προς μία συκεκριμένη IP διεύθυνση αποτελεί μία ένδειξη επίθεσης.
Έτσι, αξιοποιώντας το μηχανισμό για Συσταδοποίηση Διευθύνσεων (Clustering των IP διευθύνσεων) που προτείνει ο Κ. Γιώτης στο [43], σκοπεύουμε να κάνουμε χρήση ενός αλγορίθμου μέγιστου κοινού προθέματος (Longest Common Prefix), ώστε για κάθε ζεύγος IP προέλευσης και προορισμού να γνωρίζουμε πόσες Ροές (Flows) υπήρξαν με προέλευση και προορισμό αυτές τις διευθύνσεις (ή κάποιο πρόθεμα αυτών των διευθύνσεων).
Με την προσθήκη αυτή αναμένουμε να ανιχνεύσουμε περιπτώσεις στις οποίες ο επιτιθέμενος χρησιμοποιεί μεν IP Spoofing αλλά στέλνει πολλά πακέτα από κάθε διεύθυνση προς το θύμα.
Ως προς την υλοποίηση, αυτή η προσθήκη μπορεί να γίνει με έναν γρήγορο υπολογισμό κατά την Εξαγωγή των Ροών από τον Εξαγωγέα Ροών (Flow Extractor) - βλ. Ενότητα 2.2.3 - χωρίς να προσθέσει σημαντικό υπολογιστικό κόστος.

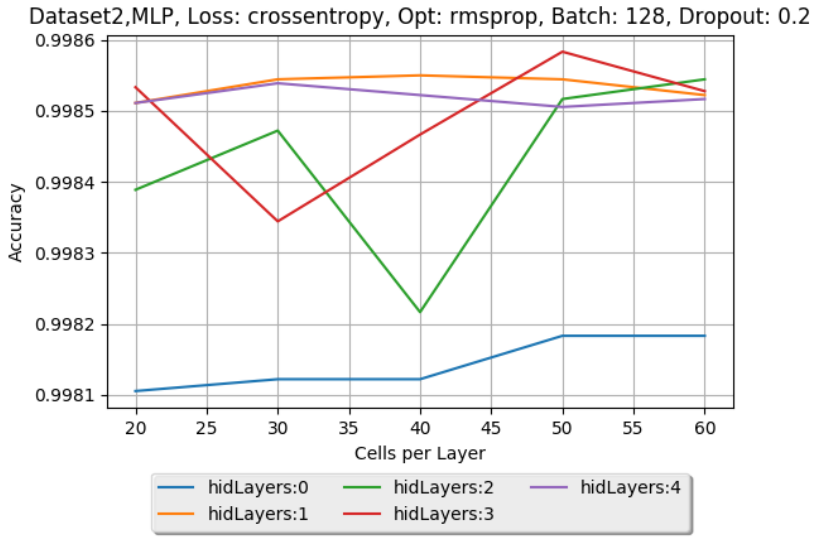
Τέλος, ένα ζήτημα προς σκέψη είναι το πώς μπορεί να γίνει επανεκπαίδευση του συστήματος από τη στιγμή που θα τεθεί σε λειτουργία. Με βάση, δηλαδή, την πραγματική κίνηση, σε ποιο χρονικό σημείο θα μπορούσαμε να επαναεκπαιδεύσουμε το δίκτυο ώστε να προσαρμοστεί στα πραγματικά δεδομένα της κίνησης.

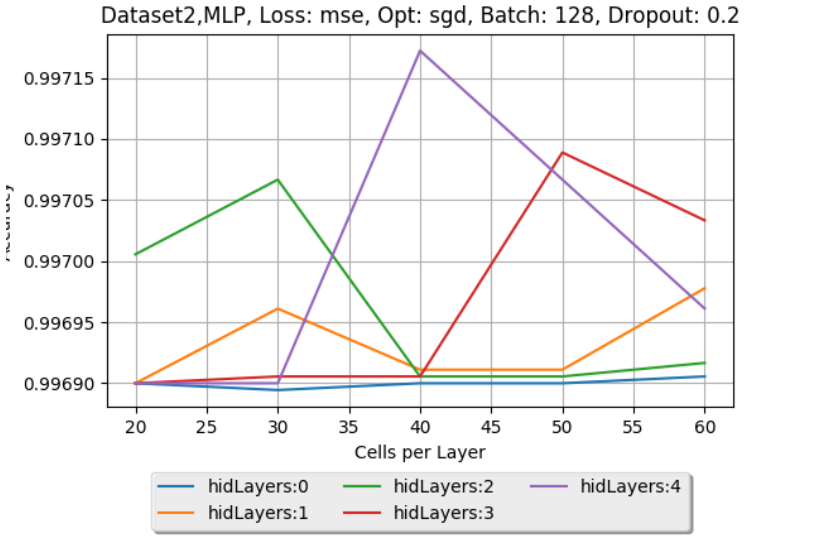
## ΠΑΡΑΡΤΗΜΑ

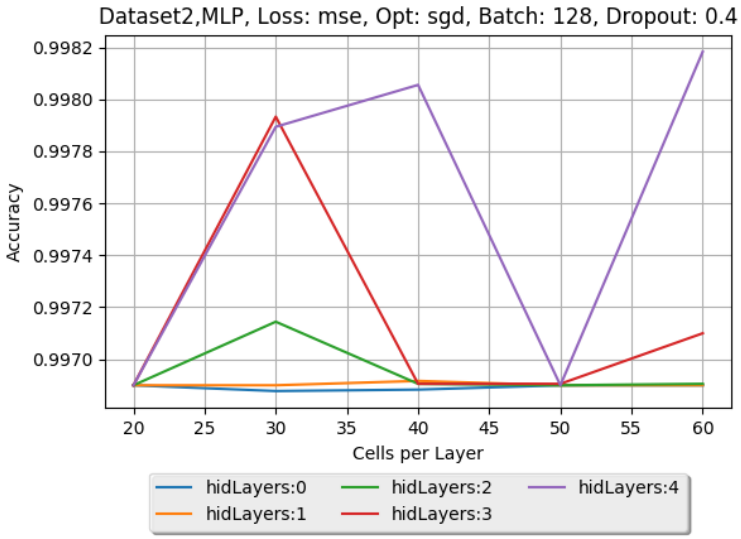
# Α΄ Επιπλέον Γραφικές Παραστάσεις Αποτελεσμάτων MLP Δικύων

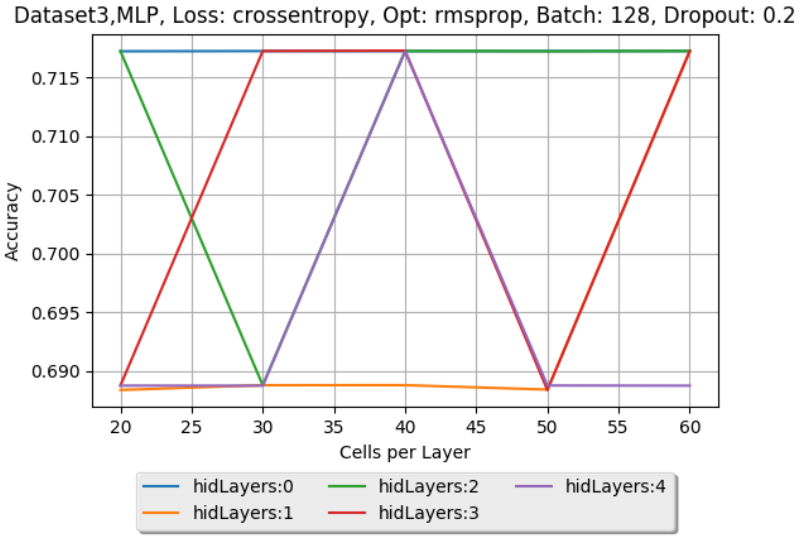


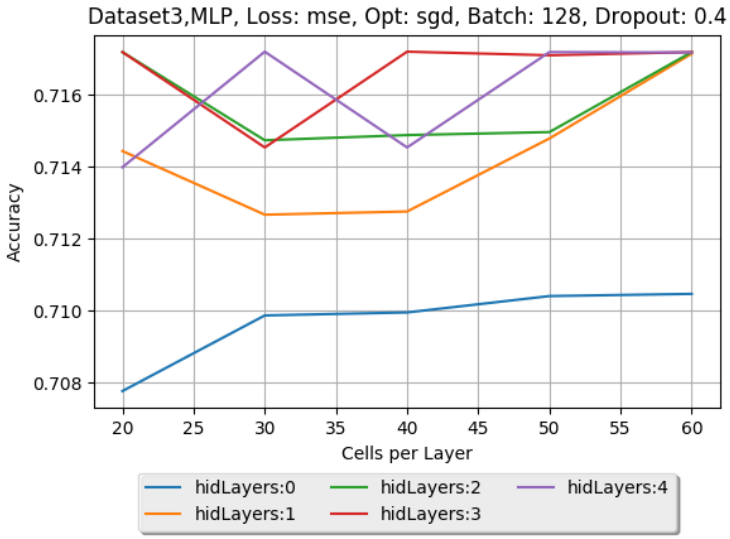



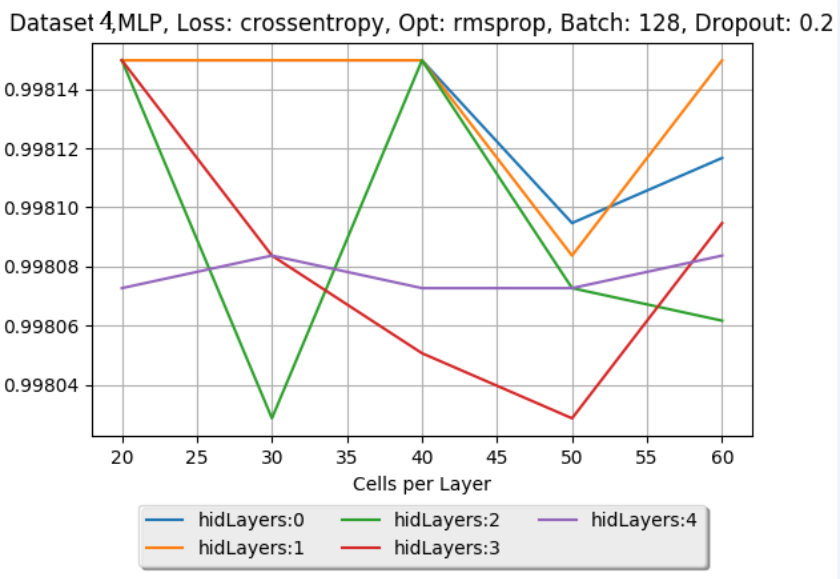

Dataset2,MLP, Loss: crossentropy, Opt: rmsprop, Batch: 128, Dropout: 0.2
Accuracy
0.9986
0.9985
0.9984
0.9983
0.9982
0.9981
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

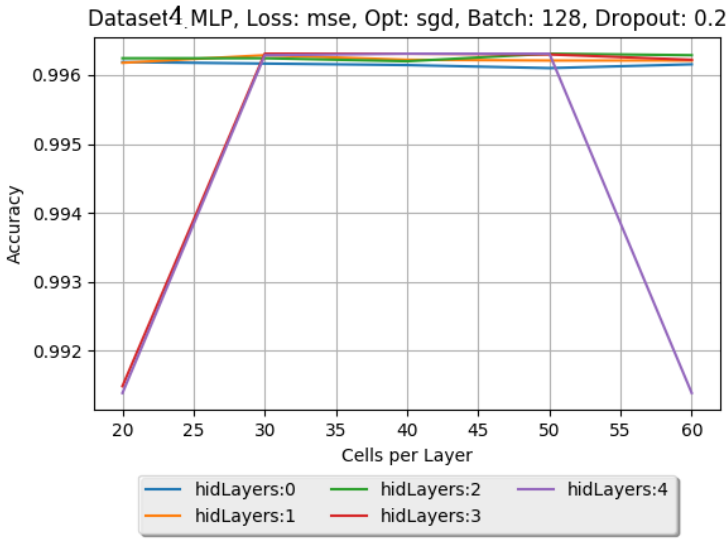

Dataset2,MLP, Loss: mse, Opt: sgd, Batch: 128, Dropout: 0.2
0.99715
0.99710
0.99705
0.99700
0.99695
0.99690
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

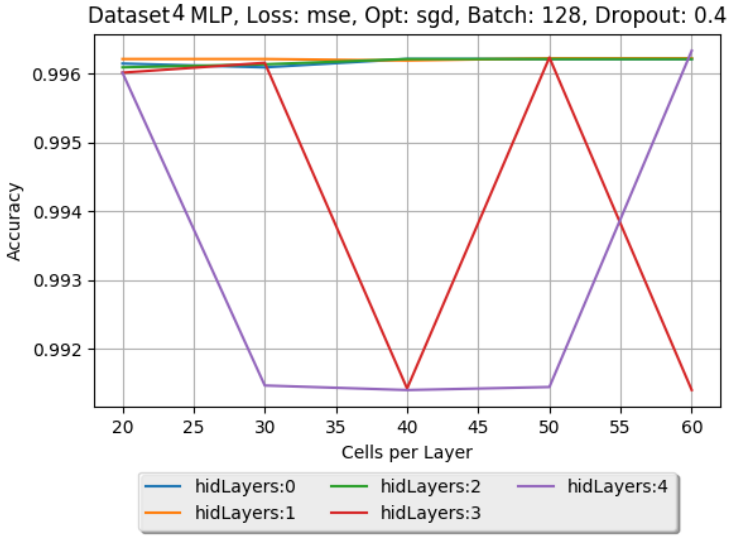

Dataset2,MLP, Loss: mse, Opt: sgd, Batch: 128, Dropout: 0.4
Accuracy
0.9982
0.9980
0.9978
0.9976
0.9974
0.9972
0.9970
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

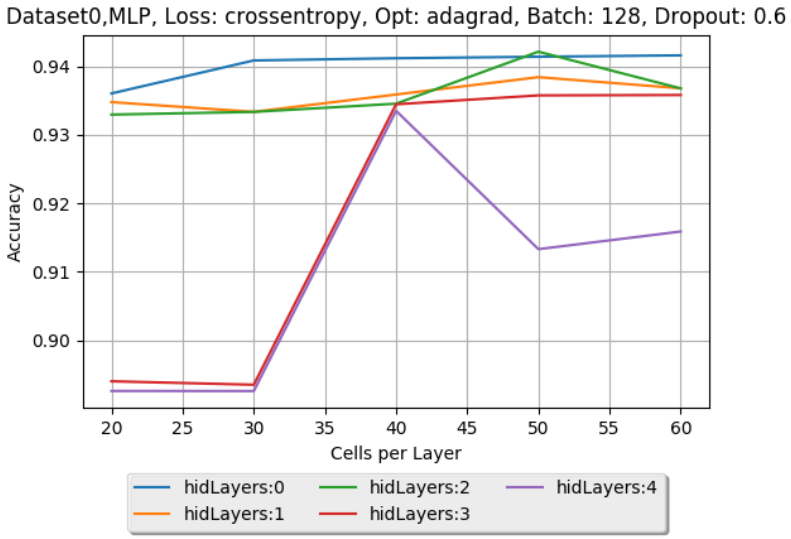

Dataset3,MLP, Loss: crossentropy, Opt: rmsprop, Batch: 128, Dropout: 0.2
Accuracy
0.715
0.710
0.705
0.700
0.695
0.690
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

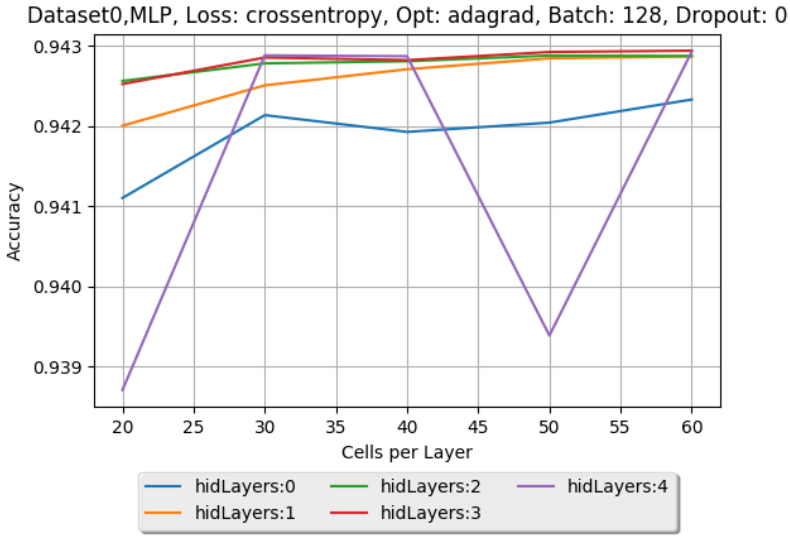

Dataset3,MLP, Loss: mse, Opt: sgd, Batch: 128, Dropout: 0.4
Accuracy
0.716
0.714
0.712
0.710
0.708
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

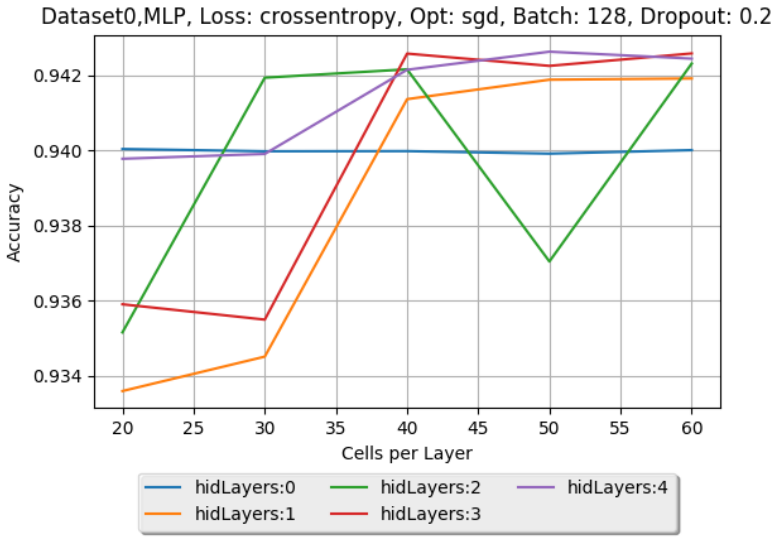

Dataset 4,MLP, Loss: crossentropy, Opt: rmsprop, Batch: 128, Dropout: 0.2
0.99814
0.99812
0.99810
0.99808
0.99806
0.99804
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

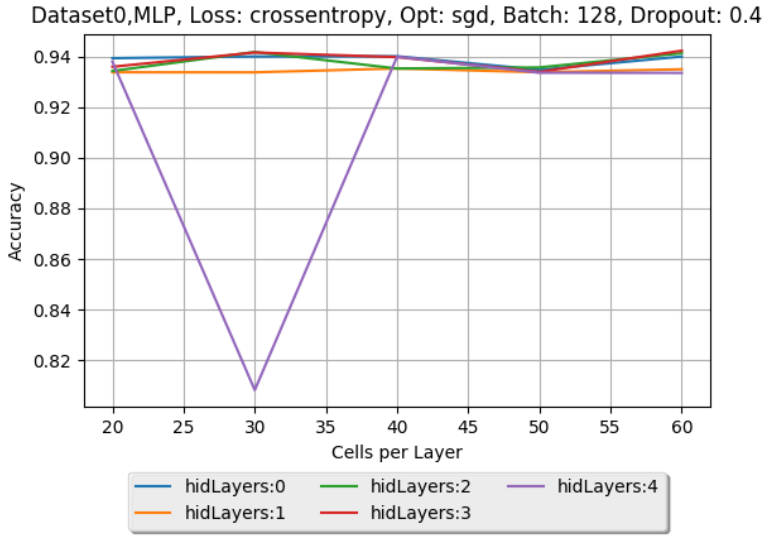

Dataset 4 MLP, Loss: mse, Opt: sgd, Batch: 128, Dropout: 0.2
Accuracy
0.996
0.995
0.994
0.993
0.992
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

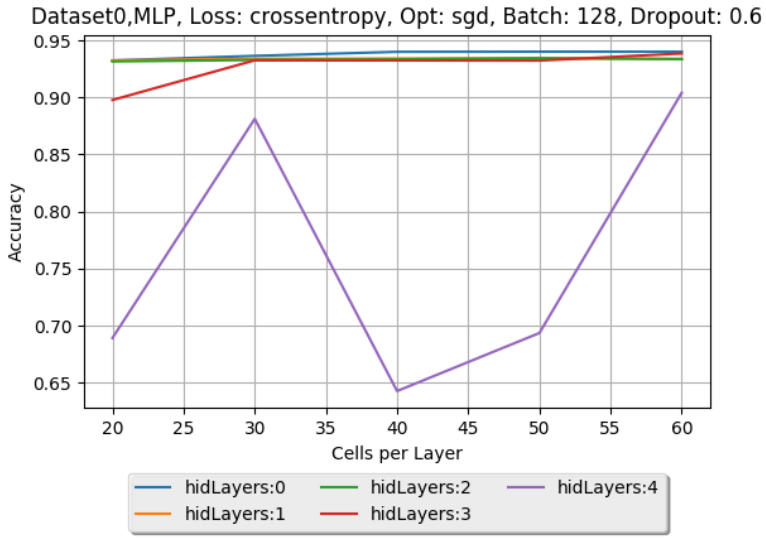

Dataset 4 MLP, Loss: mse, Opt: sgd, Batch: 128, Dropout: 0.4
Accuracy
0.996
0.995
0.994
0.993
0.992
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

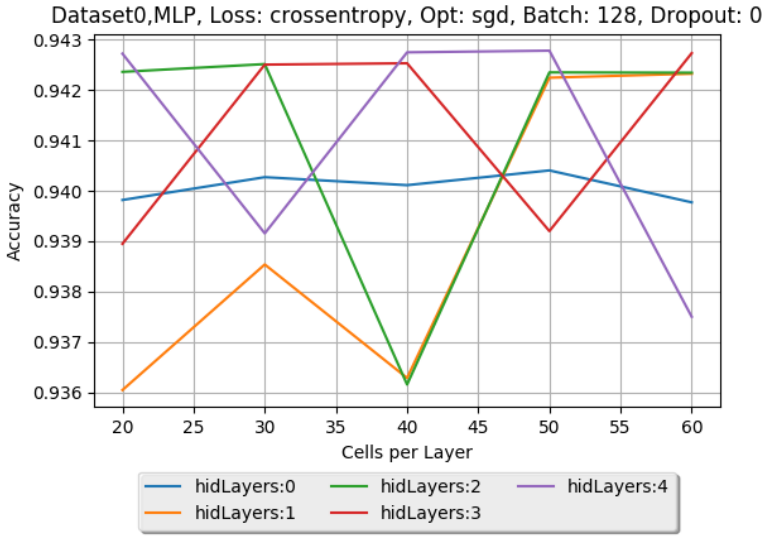

Dataset0,MLP, Loss: crossentropy, Opt: adagrad, Batch: 128, Dropout: 0.6
Accuracy
0.94
0.93
0.92
0.91
0.90
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4


Dataset0,MLP, Loss: crossentropy, Opt: adagrad, Batch: 128, Dropout: 0
Accuracy
0.943
0.942
0.941
0.940
0.939
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4


Dataset0,MLP, Loss: crossentropy, Opt: sgd, Batch: 128, Dropout: 0.2
Accuracy
0.942
0.940
0.938
0.936
0.934
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4


Dataset0,MLP, Loss: crossentropy, Opt: sgd, Batch: 128, Dropout: 0.4
Accuracy
0.94
0.92
0.90
0.88
0.86
0.84
0.82
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4


Dataset0,MLP, Loss: crossentropy, Opt: sgd, Batch: 128, Dropout: 0.6
Accuracy
0.95
0.90
0.85
0.80
0.75
0.70
0.65
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4


Dataset0,MLP, Loss: crossentropy, Opt: sgd, Batch: 128, Dropout: 0
Accuracy
0.943
0.942
0.941
0.940
0.939
0.938
0.937
0.936
20
25
30
35
40
45
50
55
60
Cells per Layer
hidLayers:0
hidLayers:1
hidLayers:2
hidLayers:3
hidLayers:4

# Αναφορές


[1] "The CAIDA UCSD "DDoS Attack 2007" Dataset." [Online]. Available: http://www.caida.org/data/passive/ddos-20070804_dataset.xml

[2] "Scapy." [Online]. Available: http://www.secdev.org/projects/scapy/

[3] "nmap." [Online]. Available: https://nmap.org/

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

[5] "Cloudflare." [Online]. Available: https://www.cloudflare.com/ddos/

[6] "Arbor Networks." [Online]. Available: https://www.arbornetworks.com/

[7] "packtpub." [Online]. Available: https://www.packtpub.com/books/content/implementing-artificial-neural-networks-tensorflow

[8] "codeproject." [Online]. Available: https://www.codeproject.com/Articles/175777/Financial-predictor-via-neural-network

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

[10] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[11] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks," vol. 385, 2012. [Online]. Available: http://link.springer.com/10.1007/978-3-642-24797-2

[12] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," 2013. [Online]. Available: http://arxiv.org/abs/1312.6026

[13] A. Graves, A. r. Mohamed, and G. E. Hinton, "Speech recognition with deep recurrent neural networks," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, no. 6, pp. 6645–6649, 2013.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated Feedback Recurrent Neural Networks," vol. 37, 2015. [Online]. Available: http://arxiv.org/abs/1502.02367

[15] K. Janocha and W. M. Czarnecki, "On Loss Functions for Deep Neural Networks in Classification," pp. 1–10, 2017. [Online]. Available: http://arxiv.org/abs/1702.05659

[16] Y. Lecun, S. Chopra, R. Hadsell, M. A. Ranzato, and F. J. Huang, "A Tutorial on Energy-Based Learning 1 Introduction : Energy-Based Models," pp. 1–59, 2006.

[17] Y. Bengio, *Neural Networks for Speech and Sequence Recognition*. International Thomson Computer Press, 1996. [Online]. Available: https://books.google.gr/books?id=mLVQAAAAMAAJ

[18] Y. Bengio, Yann LeCun, and H. Y, "Globally trained handwritten word recognizer using spatial representation, space displacement neural networks and hidden markov models," *Advances in Neural Information Processing Systems*, vol. 6, 1993.

[19] S. Aksoy and R. M. Haralick, "Feature normalization and likelihood-based similarity measures for image retrieval," *Pattern Recognition Letters*, vol. 22, no. 5, pp. 563–582, 2001.

[20] A. Ng, "Coursera." [Online]. Available: https://www.coursera.org/learn/machine-learning

[21] "Dyn." [Online]. Available: https://dyn.com/

[22] Guardian, "DDoS attack that disrupted internet was largest of its kind in history, experts say," 2016. [Online]. Available: https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet

[23] "Wired." [Online]. Available: https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/

[24] "Adweek." [Online]. Available: http://www.adweek.com/digital/major-cyber-attack-hurting-twitter-spotify-etsy-shopify-and-other-sites-174214/

[25] "kinja." [Online]. Available: http://fusion.kinja.com/here-are-the-sites-you-cant-access-because-someone-took-1793863079

[26] "Ellak." [Online]. Available: https://privacy.ellak.gr/2017/04/24/ti-ine-mia-epithesi-ddos-mia-isagogi-stis-epithesis-distributed-denial-of-service-ddos/

[27] "Calyptix." [Online]. Available: https://www.calyptix.com/top-threats/3-common-dns-attacks-and-how-to-fight-them/

[28] "Digital Attack Map." [Online]. Available: http://www.digitalattackmap.com/understanding-ddos/

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

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

[31] "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information," 2013. [Online]. Available: https://tools.ietf.org/html/rfc7011

[32] "Cisco NetFlow." [Online]. Available: http://www.cisco.com/go/netflow

[33] "Cisco." [Online]. Available: http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186

[34] V. D, "User Guide," *Optimization*, no. June, 2010.

[35] "iptables." [Online]. Available: https://help.ubuntu.com/community/IptablesHowTo

[36] "ufw." [Online]. Available: https://wiki.ubuntu.com/UncomplicatedFirewall

[37] "wireshark." [Online]. Available: https://www.wireshark.org/

[38] "editcap." [Online]. Available: https://www.wireshark.org/docs/man-pages/editcap.html

[39] "IANA." [Online]. Available: https://www.iana.org/

[40] "RFC 7012." [Online]. Available: https://tools.ietf.org/html/rfc7012

[41] "IPFIX IE Assigned Numbers." [Online]. Available: https://www.iana.org/assignments/ipfix/ipfix.xhtml

[42] "Keras." [Online]. Available: https://keras.io/

[43] K. Giotis, G. Androulidakis, and V. Maglaris, "A scalable anomaly detection and mitigation architecture for legacy networks via an OpenFlow middlebox," *SECURITY AND COMMUNICATION NETWORKS*, vol. 9, no. 13, pp. 1958–1970, 2015. [Online]. Available: http://dx.doi.org/10.1002/sec.1368