

Ατομική Διπλωματική Εργασία

**ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΣΗΜΑΤΩΝ ΕΕΓ ΜΕ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΤΥΠΟΥ  
ECHO STATE**

Άντρη Κώστα

Επιβλέπων Καθηγητής: Δρ. Χρίστος Χριστοδούλου

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**



**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

Μάιος 2017

# **ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

## **ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

### **ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗ ΣΗΜΑΤΩΝ ΕΕΓ ΜΕ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ ΤΥΠΟΥ ΕCHO STATE**

**Άντρη Κώστα**

Επιβλέπων Καθηγητής

Δρ. Χρίστος Χριστοδούλου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου  
Κύπρου

Μάιος 2017

## Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου Δρ. Χρίστο Χριστοδούλου για την ευκαιρία που μου έδωσε για να εργαστώ γύρω από το συγκεκριμένο θέμα, αλλά και για τη βοήθεια και καθοδήγησή του στην επίτευξη της παρούσας διπλωματικής εργασίας.

Επίσης, θα ήθελα να ευχαριστήσω το διδακτορικό φοιτητή Μιχάλη Αγαθοκλέους για την πολύτιμη βοήθειά του, την υποστήριξη και καθοδήγησή του καθ' όλη τη διάρκεια της εκπόνησης της διπλωματικής μου εργασίας.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου για τη συνεχή υποστήριξη καθ' όλη τη διάρκεια των σπουδών μου.

## Περίληψη

Υπάρχουν άνθρωποι οι οποίοι δεν μπορούν να κουνήσουν τα άκρα τους όπως οι ανάπηροι, άνθρωποι με παραπληγικά προβλήματα. Οι άνθρωποι αυτοί δεν μπορούν να κουνήσουν τα άκρα τους λόγω μυϊκής αδυναμίας και όχι αδυναμίας του εγκεφάλου. Το πρόβλημα είναι πώς μπορούμε να χρησιμοποιήσουμε τα σήματα του εγκεφάλου έτσι ώστε να κινηθούν τα άκρα των ανθρώπων που αντιμετωπίζουν κινησιολογικά προβλήματα. Είναι αντιληπτή η σημαντικότητα του προβλήματος λόγω της μεγάλης ανάγκης των ανθρώπων αυτών να μπορούν να κουνήσουν τα άκρα τους.

Ο κύριος σκοπός της παρούσας πτυχιακής εργασίας είναι με δεδομένα κάποια ηλεκτροεγκεφαλογραφήματα από κίνηση δεξιού χεριού ή κίνηση αριστερού χεριού να κατηγοριοποιηθούν αναλόγως από ποιο χέρι προήλθαν, χρησιμοποιώντας νευρωνικά δίκτυα με ανάδραση τύπου Echo State.

Για το σκοπό αυτό χρησιμοποιήθηκε ο αλγόριθμος που έγραψε ο Mantas Lukoševičius (2012), στον οποίο γίνεται η κατηγοριοποίηση με βάση τα δεδομένα σε κίνηση δεξιού χεριού ή κίνηση αριστερού χεριού. Τα ηλεκτροεγκεφαλογραφήματα βρίσκονται σε έναν πίνακα ο οποίος είναι η είσοδος στο πρόβλημα και σε έναν άλλο πίνακα περιέχεται η επιθυμητή έξοδος. Στην παρούσα διπλωματική εργασία εκπαιδεύουμε ένα μοντέλο και το συγκρίνουμε με την επιθυμητή έξοδο. Η σύγκριση γίνεται μέσω των γραφικών παραστάσεων που εξαγάγει το πρόγραμμα. Η μέγιστη ακρίβεια που εξήγαγε το πρόγραμμα είναι 69%.

# Περιεχόμενα

|                   |                                       |                  |
|-------------------|---------------------------------------|------------------|
| <b>Κεφάλαιο 1</b> | <b>Εισαγωγή.....</b>                  | <b><u>8</u></b>  |
| 1.1               | Περιγραφή του προβλήματος και στόχος  | <u>9</u>         |
| 1.2               | Κίνηση                                | <u>9</u>         |
| 1.3               | Κινητικός φλοιός (motor cortex)       | <u>10</u>        |
| 1.4               | Brain Computer Interface (BCI)        | <u>12</u>        |
| 1.4.1             | Συστατικά του BCI συστήματος          | <u>13</u>        |
| 1.4.2             | Εφαρμογές του BCI                     | <u>14</u>        |
| 1.5               | Electroencephalogram (EEG)            | <u>16</u>        |
| 1.6               | Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)        | <u>18</u>        |
| 1.6.1             | Τι είναι;                             | <u>18</u>        |
| 1.6.2             | Σύγκριση με υπολογιστές               | <u>18</u>        |
| 1.6.3             | Ιστορική Αναδρομή                     | <u>18</u>        |
| 1.6.4             | Κατηγορίες μάθησης                    | <u>19</u>        |
| 1.6.5             | Το μοντέλο Mc Culloch and Pitts (MCP) | <u>20</u>        |
| 1.6.6             | Perceptrons                           | <u>21</u>        |
| <b>Κεφάλαιο 2</b> | <b>Γνωσιολογικό Υπόβραθο.....</b>     | <b><u>23</u></b> |
| 2.1               | Recurrent Neural Networks (RNN)       | <u>24</u>        |
| 2.2               | Echo State Networks (ESN)             | <u>24</u>        |
| 2.3               | Παραγωγή του DR                       | <u>26</u>        |
| 2.3.1             | Ανάλυση global παραμέτρων             | <u>26</u>        |

|            |                                                                   |           |
|------------|-------------------------------------------------------------------|-----------|
| 2.3.2      | Ρύθμιση για την επιλογή παραμέτρων                                | <u>28</u> |
| 2.3.3      | Εκπαίδευση του Readout                                            | <u>29</u> |
| 2.3.4      | Ridge Regression                                                  | <u>29</u> |
| 2.3.5      | Regularization                                                    | <u>30</u> |
| 2.4        | Γενική ανάλυση των γραφικών παραστάσεων που εξαγάγει το πρόγραμμα | <u>30</u> |
| 2.4.1      | Receiver Operating Characteristic – ROC Curve                     | <u>30</u> |
| 2.4.2      | Cohen’s Kappa                                                     | <u>33</u> |
| 2.4.3      | Accuracy                                                          | <u>33</u> |
| 2.4.4      | Mean Square Error (MSE)                                           | <u>34</u> |
| 2.5        | Ανάλυση Δεδομένων                                                 | <u>34</u> |
| Κεφάλαιο 3 | Σχεδιασμός και Υλοποίηση.....                                     | <u>36</u> |
| 3.1        | Ανάλυση παραμέτρων του προγράμματος                               | <u>36</u> |
| 3.2        | Διάγραμμα Ροής του προγράμματος                                   | <u>38</u> |
| Κεφάλαιο 4 | Αποτελέσματα και Συζήτηση.....                                    | <u>47</u> |
| 4.1        | Ενεργοποίηση του DR                                               | <u>48</u> |
| 4.2        | Έλεγχος του δικτύου                                               | <u>51</u> |
| 4.3        | Spectral Radius                                                   | <u>51</u> |
| 4.4        | Αραιότητα του DR                                                  | <u>51</u> |
| 4.5        | Χρόνος εκτέλεσης του προγράμματος                                 | <u>53</u> |
| 4.6        | Ρυθμός διαρροής ( $\alpha$ )                                      | <u>54</u> |
| 4.7        | Regularization Coefficient ( $\beta$ )                            | <u>56</u> |
| 4.8        | Μέγεθος παραθύρου                                                 | <u>59</u> |

|                                       |                    |
|---------------------------------------|--------------------|
| 4.9 Μέγεθος DR                        | <a href="#">63</a> |
| 4.10 Τελικές παραμέτροι               | <a href="#">65</a> |
| 4.11 Σύγκριση αποτελεσμάτων           | <a href="#">69</a> |
| Κεφάλαιο 5 Συμπεράσματα.....          | <a href="#">71</a> |
| Βιβλιογραφία .....                    | <a href="#">72</a> |
| Παράρτημα Α: ESNBCITrainTest.py ..... | <a href="#">75</a> |
| Παράρτημα Β: demo.py.....             | <a href="#">83</a> |

## **Συντομογραφίες**

BCI Brain Computer Interface

EEG ElectroEncephaloGraphy

MCP Mc Culloch and Pitts

MLP Multilayer Perceptron

TNΔ Τεχνητά Νευρωνικά Δίκτυα

RNN Recurrent Neural Network

ESN Echo State Networks

LSM Liquid State Machines

DR Dynamic Reservoir

RC Reservoir Computing

MSE Mean Square Error

RMSE Root Mean Square Error

ROC Receiver Operating Characteristic

TPR True Positive Rate

FPR False Positive Rate

AUC Area Under a ROC Curve



# Κεφάλαιο 1

## Εισαγωγή

---

|                                             |           |
|---------------------------------------------|-----------|
| 1.1 Περιγραφή του προβλήματος και στόχος    | <u>9</u>  |
| 1.2 Κίνηση                                  | <u>9</u>  |
| 1.3 Κινητικός φλοιός (motor cortex)         | <u>10</u> |
| 1.4 Brain Computer Interface (BCI)          | <u>12</u> |
| 1.4.1 Συστατικά του BCI συστήματος          | <u>13</u> |
| 1.4.2 Εφαρμογές του BCI                     | <u>14</u> |
| 1.5 Electroencephalogram (EEG)              | <u>16</u> |
| 1.6 Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)          | <u>18</u> |
| 1.6.1 Τι είναι;                             | <u>18</u> |
| 1.6.2 Σύγκριση με υπολογιστές               | <u>18</u> |
| 1.6.3 Ιστορική Αναδρομή                     | <u>18</u> |
| 1.6.4 Κατηγορίες μάθησης                    | <u>19</u> |
| 1.6.5 Το μοντέλο Mc Culloch and Pitts (MCP) | <u>20</u> |
| 1.6.6 Perceptrons                           | <u>21</u> |

---

## 1.1 Περιγραφή του προβλήματος και στόχος

Υπάρχουν άνθρωποι οι οποίοι δεν μπορούν να κουνήσουν τα άκρα τους όπως οι ανάπηροι, άνθρωποι με παραπληγικά προβλήματα. Ο εγκέφαλος αυτών των ανθρώπων λειτουργεί κανονικά αλλά οι μύες των άκρων είναι αδύναμοι. Συνεπώς δεν μπορούν να κουνήσουν τα άκρα τους λόγω μυϊκής αδυναμίας και όχι αδυναμίας του εγκεφάλου. Τα σήματα στον εγκέφαλο στέλνονται κανονικά. Το πρόβλημα είναι πώς μπορούμε να χρησιμοποιήσουμε τα σήματα του εγκεφάλου έτσι ώστε να κινηθούν τα άκρα των ανθρώπων που αντιμετωπίζουν κινησιολογικά προβλήματα. Είναι αντιληπτή η σημαντικότητα του προβλήματος λόγω της μεγάλης ανάγκης των ανθρώπων αυτών να μπορούν να κουνήσουν τα άκρα τους.

Στόχος της παρούσας διπλωματικής εργασίας είναι με δεδομένα κάποια ηλεκτροεγκεφαλογραφήματα να κατηγοριοποιηθούν αυτά αναλόγως της κίνησης τους δηλαδή αν προήλθαν από κίνηση αριστερού χεριού ή κίνηση δεξιού χεριού. Πιο συγκεκριμένα με δεδομένα έναν πίνακα  $u(n)$  και την επιθυμητή έξοδο  $y^{target}(n)$  όπου  $n=1,2,...,T$  και  $T$  ο αριθμός των δεδομένων στο training dataset, το ζητούμενο είναι να μάθουμε ένα μοντέλο  $y(n)$  όπου το  $y(n)$  αντιστοιχεί στο  $y^{target}(n)$  όσο το δυνατό πιο πιθανό μειώνοντας το σφάλμα και το πιο σημαντικό γενικεύοντας καλά τα αόρατα δεδομένα.

## 1.2 Κίνηση

Σύμφωνα με τις Σ. Γιακουμάκη και Ζ.Παπαδοπούλου [18], υπάρχουν πολλές περιοχές του εγκεφάλου που εμπλέκονται στην εκτέλεση μιας κίνησης από τον άνθρωπο. Η νευρωνική δραστηριότητα σε αυτές τις περιοχές συνδυάζεται έτσι ώστε να δημιουργηθεί μια χαλαρή αλυσίδα εντολών. Μια κινητική ιεραρχία από τον εγκεφαλικό φλοιό και τα βασικά γάγγλια προς την περεγκεφαλίδα και το νωτιαίο μυελό. (Σχήμα 1.1)

Η πληροφορία που κωδικοποιείται στον κινητικό φλοιό έχει αποτελέσει το θέμα μακροχρόνιας διαμάχης το ερώτημα ήταν εάν τα κύτταρα του φλοιού επεξεργάζονται την πληροφορία που αφορά τις κινήσεις που θέλει να κάνει κάποιος ή τους μυς που πρέπει να συσπαστούν, προκειμένου να διεκπεραιωθεί η κίνηση; Η απάντηση είναι ότι μεμονωμένοι νευρώνες δεν κωδικοποιούν ούτε το ένα ούτε το άλλο. Αντίθετα, χρησιμοποιείται μία πληθυσμιακή κωδικοποίηση, κατά την οποία οι κινήσεις καθορίζονται από την πυροδότηση ενός συνόλου νευρώνων. Ακριβώς μπροστά από τον κινητικό φλοιό βρίσκονται σημαντικές προκινητικές περιοχές, που εμπλέκονται στο σχεδιασμό των κινήσεων, στην προετοιμασία των νωτιαίων κυκλωμάτων της κίνησης και σε διαδικασίες που συνδέουν το πώς βλέπουμε τις κινήσεις και το πως κατανοούμε τις χειρονομίες.



Σχήμα 1.1: Οι διάφορες περιοχές του εγκεφάλου που εμπλέκονται στην κίνηση [18]

### 1.3 Κινητικός φλοιός (motor cortex)

Με βάση τον Αγαθοκλέους (2013) [1], Graziano (2009) [7] και Hammond (2008) [9], ο κινητικός φλοιός είναι μια περιοχή του εγκεφάλου που είναι υπεύνη για το σχεδιασμό, τον έλεγχο και την εκτέλεση των εκούσιων κινήσεων. Αυτή η περιοχή του εγκεφάλου βρίσκεται στον εγκεφαλικό φλοιό και φαίνεται στο πιο κάτω σχήμα (Σχήμα 1.2).

Ο κινητικός φλοιός αποτελείται από τρεις διαφορετικές περιοχές του μετωπιαίου λοβού. Οι περιοχές αυτές είναι ο πρωτοταγής κινητικός φλοιός, ο προκινητικός φλοιός και ο συμπληρωματικός κινητικός φλοιός. (Σχήμα 1.2) Κάθε μία από αυτές τις περιοχές είναι υπεύνη για ειδικές εργασίες του κινητικού φλοιού.

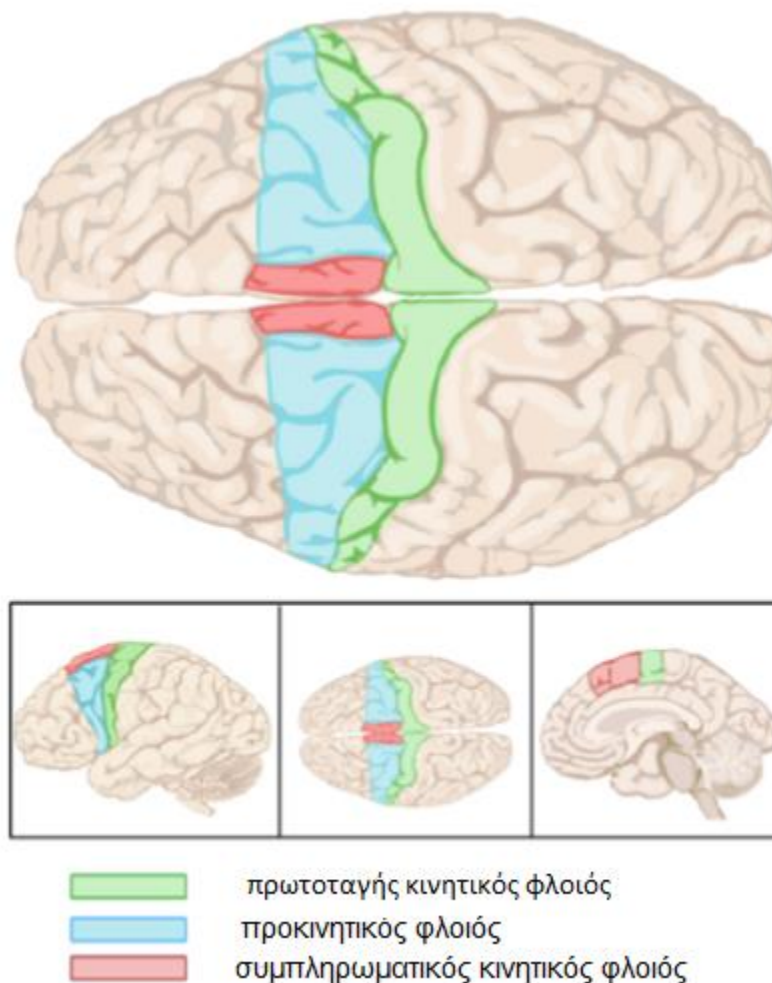
Ο πρωτοταγής κινητικός φλοιός ελέγχει μεμονωμένες κινήσεις ή σειρά κινήσεων που απαιτούν τη δραστηριότητα πολλαπλών μυών. Οι νευρώνες του πρωτοταγούς κινητικού φλοιού ενεργοποιούνται 5-100msec πριν την έναρξη μιας κίνησης. Συνεπώς κωδικοποιούν τις πληροφορίες που αναστέλλουν τις κινήσεις των μυών. Επιπλέον, αυτό το μέρος του κινητικού φλοιού είναι υπεύθυνο για την κωδικοποίηση της έντασης, της κατεύθυνσης, της έκτασης και της ταχύτητας της κίνησης.

Ο προκινητικός φλοιός είναι μια περιοχή του κινητικού φλοιού η οποία δρα πριν από τον πρωτοταγή κινητικό φλοιό για την εκτέλεση πολύπλοκων εργασιών. Στέλνει απ' ευθείας σήματα στον πρωτοταγή κινητικό φλοιό και στο νωτιαίο μυελό. Αυτό το μέρος του κινητικού φλοιού εκτελεί πιο σύνθετες κινήσεις από τον πρωτοταγή κινητικό φλοιό όπως πολύπλοκες στάσεις του σώματος. Συμμετέχει στην επιλογή των εκούσιων κινήσεων οι οποίες θα εκτελεστούν από τον πρωτοταγή κινητικό φλοιό. Πιο συγκεκριμένα, οι νευρώνες του προκινητικού φλοιού είναι υπεύθυνοι για την προετοιμασία των κινήσεων. Επιπλέον, αυτή η

περιοχή του κινητικού φλοιού είναι ευαίσθητη στη συμπεριφορά μιας συγκεκριμένης κίνησης. Τέλος, αυτό το μέρος του εγκεφάλου κωδικοποιεί αν μια κίνηση εκτελέστηκε σωστά ή όχι.

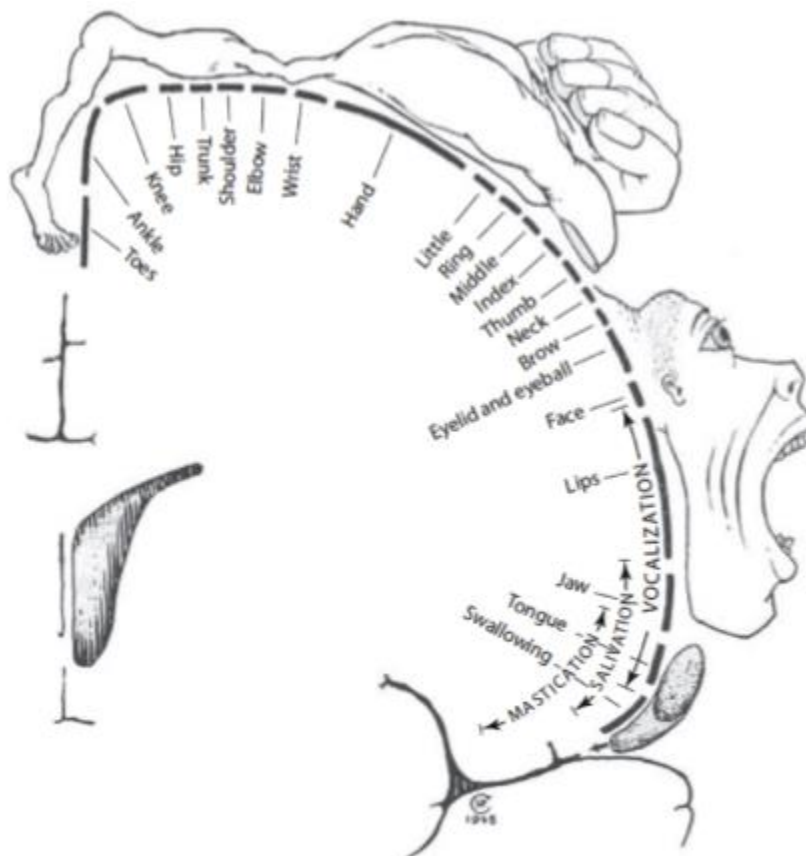
Η τρίτη περιοχή του κινητικού φλοιού είναι ο συμπληρωματικός κινητικός φλοιός ο οποίος εμπλέκεται στον προγραμματισμό πολύπλοκων ακολουθιών κινήσεων και στο συντονισμό των διμερών κινήσεων.

Σε αντίθεση με τον προκινητικό φλοιό ο οποίος εμπλέκεται στην επιλογή προγραμμάτων που βασίζονται σε οπτικά ερεθίσματα και αφηρημένες ενώσεις, ο συμπληρωματικός κινητικός φλοιός εμπλέκεται στην επιλογή κινήσεων με βάση προηγούμενες εμπειρίες και ανταποκρίνεται σε σειρές κινήσεων. Σε αντίθεση με τις άλλες δύο περιοχές του κινητικού φλοιού, η περιοχή αυτή ανταποκρίνεται κατά την εκτέλεση της ακολουθίας των κινήσεων ή κατά τη διάρκεια της φαντασίας της αλληλουχίας των κινήσεων. Επίσης, εμπλέκεται στη μετατροπή της κινηματικής σε δυναμικές πληροφορίες κατά τη διάρκεια μιας κίνησης.



Σχήμα 1.2: Περιοχές κινητικού φλοιού (Hammond(2008) [9])

Με βάση τον Αγαθοκλέους (2013) [1], μια παραδοσιακή άποψη του κινητικού φλοιού έχει σχεδιαστεί από τον Penfield and Rasmussen (1950) [13]. Η άποψη αυτή περιέχει ένα χάρτη του σώματος. Χρησιμοποιώντας ένα ηλεκτρόδιο διέγερσης μπορούμε να εντοπίσουμε σταδιακά τους νευρώνες που είναι υπεύθυνοι για την κίνηση ποδιών, γονάτων, βραχιόνων, χεριών, προσώπου και γλώσσας. Ο αναλυτικός χάρτης του κινητικού φλοιού φαίνεται στο Σχήμα 1.2. Συγκεκριμένα, μπορούμε να δούμε την αναπαράσταση των μερών του σώματος που εκτελούν συγκεκριμένες κινήσεις όπως τα χέρια και το πρόσωπο τα οποία είναι συγκριτικά μεγαλύτερα από την αναπαράσταση άλλων μερών του σώματος όπως τα πόδια.



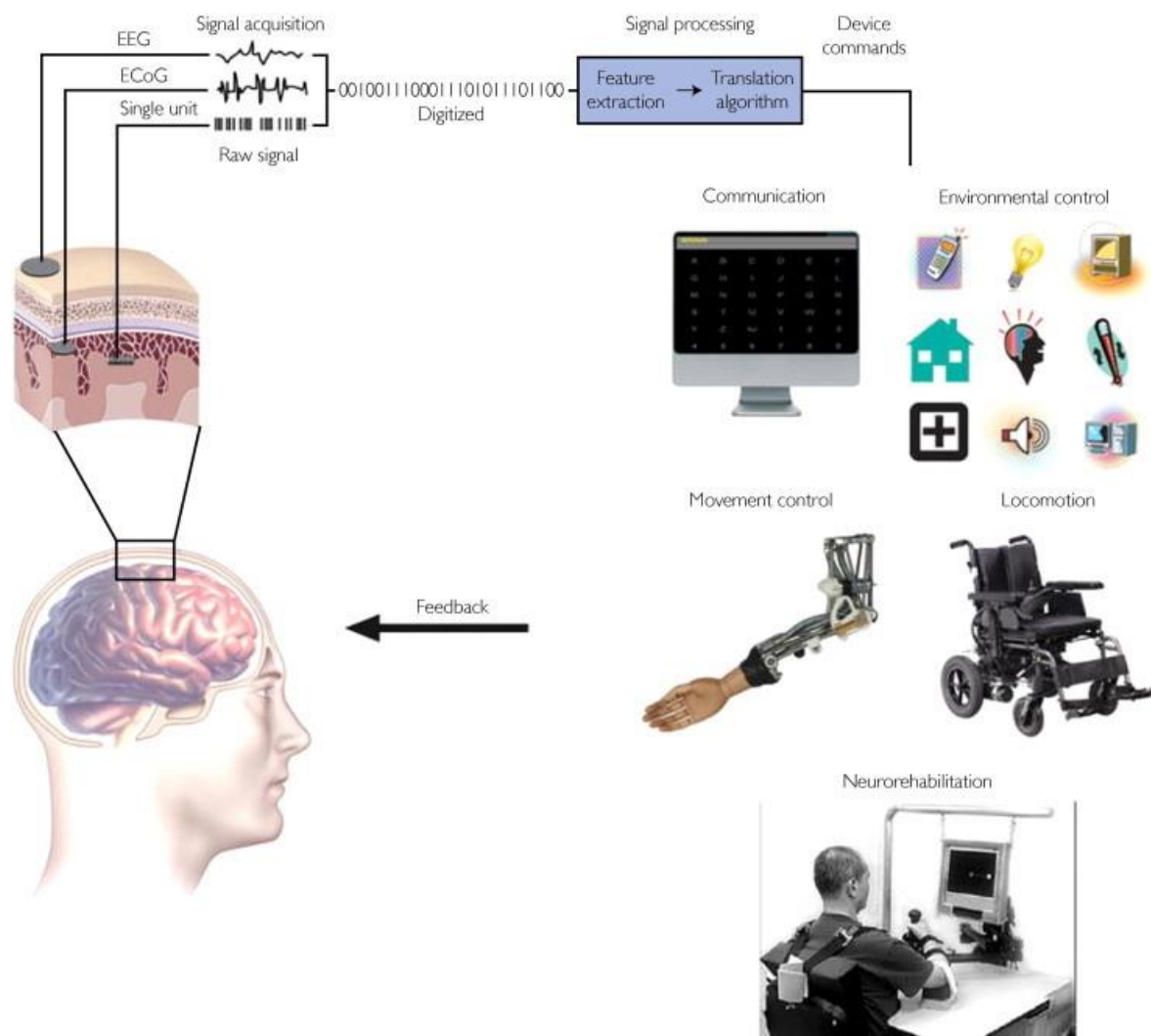
Σχήμα 1.3: Ένα μετωπιαίο μέρος του κινητικού φλοιού το οποίο δείχνει τον κινητήρα του ανθρώπινου εγκεφάλου (Graziano (2009), [7])

#### 1.4 Brain Computer Interface (BCI):

Με βάση τον Αγαθοκλέους (2013) [1], Brain Computer Interface(BCI) είναι ένα σύστημα επικοινωνίας που δεν απαιτεί καμία περιφεριακή μυϊκή δραστηριότητα, αλλά επιτρέπει σε ένα αντικείμενο να στείλει εντολές σε μια ηλεκτρονική συσκευή μόνο μέσω της δραστηριότητας του εγκεφάλου. BCIs έχουν αναπτυχθεί κυρίως ως συσκευές επικοινωνίας για τους ανθρώπους που δεν μπορούν να χρησιμοποιήσουν τους μύες τους, λόγω διαφόρων ασθενειών.

## Συστατικά του BCI συστήματος

Σύμφωνα με Jerry J. Shih, Dean J. Krusienski και Jonathan R. Wolpaw [15], ο σκοπός ενός BCI είναι να ανιχνεύει και να ποσοτικοποιεί χαρακτηριστικά των εγκεφαλικών σημάτων που υποδηλώνουν τις προθέσεις του χρήστη και να μεταφράζουν αυτά τα χαρακτηριστικά σε πραγματικό χρόνο σε εντολές συσκευών που επιτυγχάνουν την πρόθεση του χρήστη (Σχήμα 1.4). Για να επιτευχθεί αυτό, ένα σύστημα BCI αποτελείται από 4 διαδοχικά συστατικά: (1) απόκτηση σήματος, (2) εξαγωγή χαρακτηριστικών, (3) μετάφραση χαρακτηριστικών και (4) έξοδος συσκευής. Αυτά τα 4 συστατικά ελέγχονται από ένα πρωτόκολλο λειτουργίας που καθορίζει την έναρξη και το χρόνο λειτουργίας, τις λεπτομέρειες της επεξεργασίας σήματος, τη φύση των εντολών της συσκευής και την επίβλεψη της απόδοσης. Ένα αποτελεσματικό πρωτόκολλο λειτουργίας επιτρέπει σε ένα σύστημα BCI να είναι ευέλικτο και να εξυπηρετεί τις συγκεκριμένες ανάγκες κάθε χρήστη.



Σχήμα 1.4: Συστατικά ενός BCI συστήματος [15]

## **Εφαρμογές του BCI:**

Σύμφωνα με τους S. N. Abdulkater, A. Atia και M. M. Mostafa [2], οι εφαρμογές του BCI εμπλέκονται στους εξής τομείς:

- *Ιατρική*  
Στην ιατρική η λειτουργία παρακολούθησης ψυχικών καταστάσεων των συστημάτων BCI συνέβαλε επίσης στην πρόβλεψη και την ανίχνευση προβλημάτων υγείας όπως η ανώμαλη δομή του εγκεφάλου (όπως ο όγκος στον εγκέφαλο), η διαταραχή κατάσχεσης (όπως η επιληψία), η διαταραχή του ύπνου (όπως η ναρκοληψία) και η διόγκωση του εγκεφάλου (εγκεφαλίτιδα). Η δυσλεξία, μία από τις διαταραχές του εγκεφάλου, μπορεί να διαγνωστεί με τη μέτρηση της συμπεριφοράς του εγκεφάλου. Επίσης, και οι διαταραχές ύπνου μπορούν να διαγνωστούν μέσω του BCI. Το BCI συμβάλλει επίσης στην αποκατάσταση της κινητικότητας για ασθενείς με κινητικά προβλήματα έτσι ώστε να αποκαταστήσουν τις χαμένες λειτουργίες τους και να ανακτήσουν προηγούμενα επίπεδα λειτουργικότητας ή να τους βοηθήσει να προσαρμοστούν στις αναπηρίες τους. Τα κινητά ρομποτόι μπορούν να χρησιμοποιηθούν για τη διεκπαιρέωση των καθημερινών δραστηριοτήτων των ανθρώπων με κινητικά προβλήματα, αλλά και προσθετικά άκρα με βάση το BCI μπορούν να χρησιμοποιηθούν για την κανονική λειτουργικότητα.
- *Νευροεργονομία και έξυπνο περιβάλλον*  
Τα έξυπνα περιβάλλοντα μπορούν να εκμεταλλευτούν τα BCI προσφέροντας περισσότερη ασφάλεια, πολυτέλεια και φυσιολογικό έλεγχο στην καθημερινή ζωή των ανθρώπων. Για παράδειγμα, η έξυπνη μεταφορά έχει επωφεληθεί από το BCI, αφού με τη χρήση των σημάτων EEG μπορεί να ανιχνευθεί η κόπωση και ο δείκτης φόρτου εργασίας όπου μέσω αυτών μπορεί να αξιολογηθεί η ψυχική κατάσταση του οδηγού. Επίσης, σε ένα έξυπνο σύστημα οδήγησης μπορούν να προβλεφθούν η συγκέντρωση και το άγχος αναλύοντας τα σήματα ECG (electrocardiography) και EEG και τον έλεγχο της ταχύτητας του αυτοκινήτου μέσω της τιμής συγκέντρωσης των σημάτων του εγκεφάλου. Επίσης, οι αλκοολικοί οδηγοί, ως συντελεστές τροχαίων ατυχημάτων, θα μπορούσαν επίσης να χαρακτηριστούν με τη χρήση σημάτων EEG.
- *Neuromarketing και διαφήμιση*  
Ο τομέας μάρκετινγκ υπήρξε επίσης ενδιαφέρον για τις έρευνες του BCI. Υπάρχουν πολλά οφέλη από τη χρήση της αξιολόγησης EEG για τηλεοπτικές διαφημίσεις που σχετίζονται τόσο με εμπορικούς όσο και με πολιτικούς τομείς.
- *Εκπαίδευση και αυτορρύθμιση*  
Το neurofeedback είναι μια προσέγγιση για την ενίσχυση της απόδοσης του εγκεφάλου μέσω της στοχοθέτησης της διαφοροποίησης της ανθρώπινης εγκεφαλικής

δραστηριότητας. Εισβάλλει στα εκπαιδευτικά συστήματα, τα οποία χρησιμοποιούν τα ηλεκτρικά σήματα του εγκεφάλου για να καθορίσουν το βαθμό της σαφήνειας των μελετημένων πληροφοριών. Επίσης, έχει μελετηθεί η αυτορρύθμιση μέσω του BCI. Παρέχει ένα μέσο για τη βελτίωση των γνωστικών θεραπευτικών προσεγγίσεων. Επιπλέον, η συναισθηματική νοημοσύνη με βάση το EEG έχει εφαρμοστεί σε αθλητικούς αγώνες για τον έλεγχο του συνοδευτικού άγχους.

- *Παιχνίδια και ψυχαγωγία*

Εφαρμογές παιχνιδιών και ψυχαγωγίας έχουν επίσης αναπτυχθεί για μη ιατρικά BCI. Για παράδειγμα, ελικοπτερα τα οποία μπορούν να πετάξουν σε οποιοδήποτε σημείο σε έναν εικονικό κόσμο 2D ή 3D.

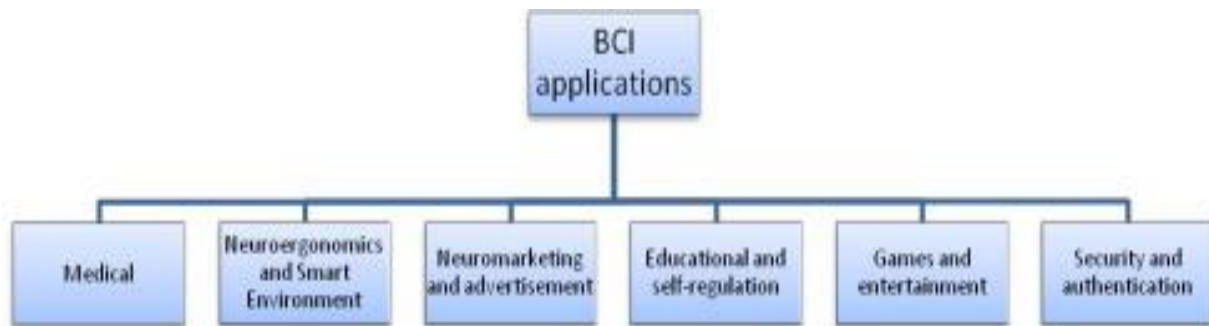
Ο συνδυασμός των χαρακτηριστικών των υπάρχοντων παιχνιδιών με δυνατότητες ελέγχου του εγκεφάλου έχει αποτελέσει αντικείμενο πολλών ερευνών, οι οποίες τείνουν να προσφέρουν εμπειρία πολυεγκεφαλικής ψυχαγωγίας. Το βιντεοπαιχνίδι ονομάζεται BrainArena. Οι παίκτες μπορούν να συμμετάσχουν σε ένα συνεργατικό ή ανταγωνιστικό ποδοσφαιρικό παιχνίδι μέσω δύο BCI. Μπορούν να βάλουν γκολ με το να φαντάζονται κινήσεις αριστερού ή δεξιού χεριού.

Έχουν αναπτυχθεί επίσης ορισμένα EEG σοβαρά εκπαιδευτικά παιχνίδια τα οποία έχουν χρησιμοποιηθεί για το συναισθηματικό έλεγχο και τη νευροπροστατική αποκατάσταση. Για παράδειγμα, το παιχνίδι brainball το οποίο σκοπεύει να μειώσει το επίπεδο άγχους. Οι χρήστες μπορούν μόνο να μετακινήσουν την μπάλα χαλαρώνοντας. Έτσι, ο πιο ήρεμος παίκτης είναι πιο πιθανό να είναι ο νικητής και έτσι θα μάθουν να ελέγχουν το άγχος τους ενώ είναι διασκεδασμένοι.

- *Ασφάλεια και πιστοποίηση*

Τα συστήματα ασφαλείας περιλαμβάνουν την εξακρίβωση γνησιότητας βασισμένη στη γνώση, βάσει αντικειμένων. Η γνωστική βιομετρία ή ηλεκτροφυσιολογία, όπου μόνο οι μέθοδοι που χρησιμοποιούν βιοστοιχεία (όπως τα σήματα του εγκεφάλου) χρησιμοποιούνται ως πηγές πληροφοριών ταυτότητας, παρέχουν μια λύση σε προβλήματα όπως ανασφαλές κωδικοί πρόσβασης, έγκλημα κλοπής και άκυρα βιομετρικά στοιχεία.





Σχήμα 1.5: Πεδία Εφαρμογών του BCI [2]

### 1.5 Electroencephalogram (EEG):

Σύμφωνα με τον M.Terlan [16], ηλεκτροεγκεφαλογράφημα είναι μια ιατρική μέθοδος η οποία διαβάζει το κρανίο και απεικονίζει την ηλεκτρική δραστηριότητα του εγκεφάλου. Η καταγραφή γίνεται από το τριχωτό της κεφαλής μέσω μεταλλικών ηλεκτροδίων και αγωγίμων μέσων. Είναι μια μη επεμβατική διαδικασία η οποία μπορεί να εφαρμοστεί επανειλημμένα σε ασθενείς, ενήλικες και παιδιά χωρίς κανένα κίνδυνο ή περιορισμό.

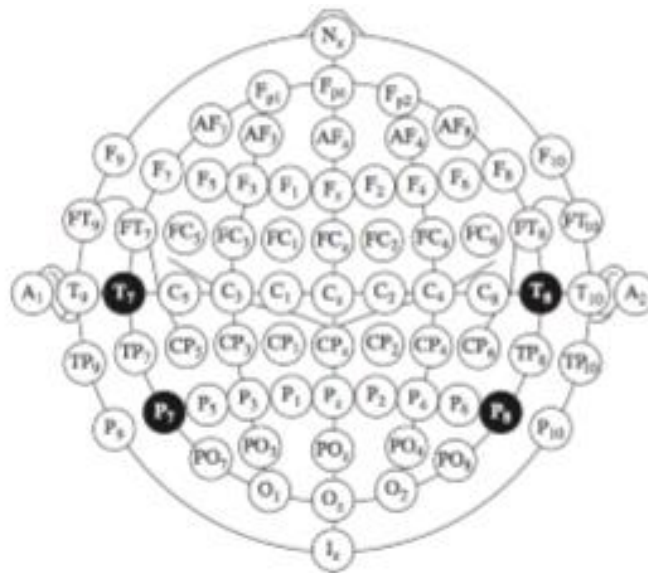
Ο εγκέφαλος αποτελείται από δισεκατομμύρια κύτταρα. Το καθένα από αυτά παράγει και μεταδίδει απειροελάχιστα ηλεκτρικά ρεύματα τα οποία αθροιζόμενα με εκείνα των άλλων νευρικών κυττάρων του εγκεφάλου δίνουν μεγαλύτερα σήματα τα οποία μπορούμε να καταγράψουμε. Η καταγραφή των ηλεκτρικών σημάτων του εγκεφάλου γίνεται με ηλεκτρόδια που τοποθετούνται στην επιφάνεια του κρανίου. Επομένως, αυτό που καταγράφουμε στο EEG είναι ηλεκτρικά σήματα από τον φλοιό του εγκεφάλου (αφού αυτό είναι πιο κοντά στο κρανίο σε σχέση με άλλες δομές του φλοιού που είναι στο εσωτερικό του εγκεφάλου).

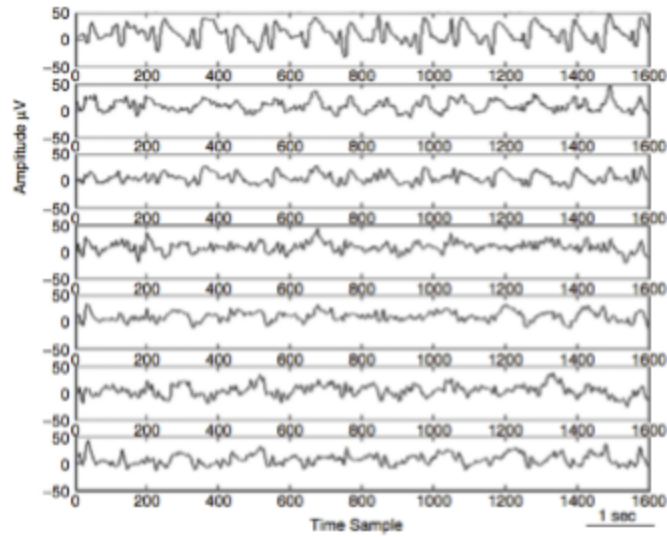
Στη μεμβράνη ενός νευρικού κυττάρου υπάρχει μια διαφορά δυναμικού μεταξύ του εσωτερικού και του εξωτερικού της. Η μεμβράνη αυτή δεν είναι πάντοτε ίδια, αφού όταν ενεργοποιείται ένα κύτταρο αλλάζει από το δυναμικό ηρεμίας σε ένα νέο δυναμικό (εκπόλωση) και αν είναι αρκετά μεγάλη η αλλαγή το δυναμικό της μεμβράνης του κυττάρου αλλάζει από το δυναμικό ηρεμίας στο δυναμικό ενεργείας. Όταν λέμε αλλάζει δυναμικό εννοούμε ότι κινήθηκε ρεύμα δηλ. φορτία. Η διαφορά δυναμικού στις δυο πλευρές της μεμβράνης οφείλεται σε διαφορετικό αριθμό και είδος φορτίων (θετικών, αρνητικών) στις δυο πλευρές της μεμβράνης. Άρα η διαφορά δυναμικού δηλ. η διαφορετική κατανομή φορτίων, μπορεί να αλλάξει μονάχα όταν κινηθούν φορτία από την μία πλευρά της μεμβράνης στην άλλη μέσω διαμεμβρανικών πρωτεϊνών (πόρων της μεμβράνης) που ονομάζονται δίαυλοι ιόντων. Και ιόντα (φορτία) μπορούν να κινηθούν είτε από το εσωτερικό του κυττάρου προς τον εξωκυττάριο χώρο, είτε αντιθέτως από τον εξωκυττάριο χώρο προς τον ενδοκυττάριο χώρο.

Για την απόκτηση βασικών μοτίβων του εγκεφάλου ζητείται από τα άτομα να κλείσουν τα μάτια τους και να χαλαρώσουν. Τα μοτίβα του εγκεφάλου σχηματίζουν μορφές κύματος που είναι συνήθως ημιτονοειδή. Συνήθως μετριοούνται από μια άκρια σε μια άλλη με εύρος από 0.5

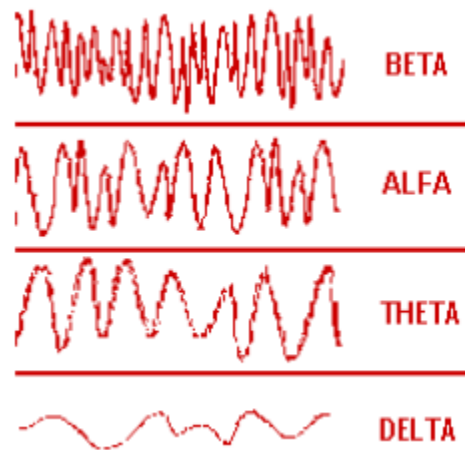
μέχρι 100  $\mu V$ . Μέσω του φάσματος ισχύος μετασχηματισμού Fourier προέρχεται το ακατέργαστο σήμα EEG. Επίσης στο φάσμα είναι ορατή η συμβολή των ημιτονοειδών κυμάτων με διαφορετικές συχνότητες. Το φάσμα είναι συνεχές, κυμαινόμενο από 0 Hz έως και το ήμισυ της συχνότητας δειγματοληψίας.

Τα EEG σήματα είναι περίπλοκα δεδομένα τα οποία παρουσιάζονται με γραφικές των millivolts έναντι του χρόνου. (Σχήμα 1.7). Τα κύματα του εγκεφάλου κατηγοριοποιούνται σε τέσσερις ομάδες. (Σχήμα 1.8). Τα βήτα κύματα ( $>13$  Hz), στα οποία το άτομο βρίσκεται σε άγρυπνη εγρήγορση, στα άλφα κύματα (8-13 Hz) στα οποία το άτομο βρίσκεται σε ήρεμη εγρήγορση, στα θήτα κύματα (4-8 Hz) όπου το άτομο αρχίζει να νυστάζει και στα δέλτα κύματα (0.5-4 Hz) όπου το άτομο κοιμάται. Επίσης, υπάρχουν και τα βήτα κύματα όπου το άτομο κοιμάται και βλέπει και όνειρο, ύπνος REM.





Σχήμα 1.7: Παράδειγμα EEG σημάτων (Sanei and Chambers (2008) [14] )



Σχήμα 1.8: Δείγματα κύματος του εγκεφάλου τα οποία χωρίζονται σε θήτα, άλφα, θήτα και δέλτα. [15]

## 1.6 Τεχνητά Νευρωνικά Δίκτυα (ΤΝΔ)

### 1.6.1 Τι είναι;

Σύμφωνα με τον Αγαθοκλέους (2013) [1], ένα τεχνητό νευρωνικό δίκτυο είναι ένας βιολογικά εμπνευσμένος Machine Learning αλγόριθμος ο οποίος αποτελείται από κόμβους ή 'νευρώνες' (τους βιολογικούς νευρώνες). Κάθε νευρώνας του αλγορίθμου αναπαριστάται από το McCulloch and Pitts μοντέλο. (McCulloch and Pitts, 1943, [12])

Τα τεχνητά νευρωνικά δίκτυα προέκυψαν από την ανάγκη να φτιάξουμε τεχνητά συστήματα που να μιμούνται τον τρόπο λειτουργίας του βιολογικού εγκεφάλου. Έχουν αποδειχθεί μια επιτυχημένη τεχνολογία για την ανάπτυξη συστημάτων με καλή γενικευτική ικανότητα χρησιμοποιώντας ένα σύνολο από αντιπροσωπευτικά παραδείγματα εκπαίδευσης.

### 1.6.2 Σύγκριση με υπολογιστές:

Σε αντίθεση με τους υπολογιστές που λειτουργούν σειριακά τα νευρωνικά δίκτυα υποστηρίζουν τον παραλληλισμό. Επίσης, σε αντίθεση με τους υπολογιστές τα ΤΝΔ είναι ανθεκτικά στα λάθη και σε βλάβες, έχουν την ικανότητα να κατηγοριοποιούν και να μαθαίνουν από τα υπάρχοντα παραδείγματα ενώ οι υπολογιστές προγραμματίζονται και ακόμη, σε αντίθεση με τους υπολογιστές δεν έχουν μεγάλο υπολογιστικό κόστος.

### 1.6.3 Ιστορική Αναδρομή

Με βάση τις σημειώσεις του μαθήματος ΕΠΛ442 [17], πιο κάτω περιγράφεται η ιστορική αναδρομή των νευρωνικών δικτύων.

- Μοντέλα νευρωνικών δικτύων:  
Το 1943 δημοσιεύεται από τον McCulloch & Pitts το άρθρο «A Logical Calculus of the Ideas Immanent in Nervous Activity». Το 1949 δημοσιεύεται από τον Hebb στο βιβλίο του «The organization of Behavior» στο οποίο περιγράφει αυτό που γνωρίζουμε σήμερα ως «Hebbian cell assemblies».
- Εκπαίδευση των νευρωνικών δικτύων:  
Το 1950 ο Hebbian learning rule: Όταν δύο νευρώνες είναι ταυτόχρονα ενεργοί, τότε η δύναμη της σύνδεσης ανάμεσα τους πρέπει να αυξηθεί.  
Το 1951 ο Minisky δημοσίευσε τη δουλειά του για την ενισχυτική μάθηση.
- Η περίοδος του Perceptron:  
Το 1960 ο Rosenblatt δημοσίευσε τις αρχές της νευροδυναμικής, για τις οποίες περιέγραψε τη διαδικασία του Perceptron learning.
- Περιορισμοί του Perceptron:  
Το 1969 δημοσιεύεται το βιβλίο «Perceptrons» από τους Minisky και Papert, το οποίο έδειξε ότι η διαδικασία Perceptron είναι ικανή να μαθαίνει μια περιορισμένη κατηγορία λειτουργιών. Επίσης πρόβλεψαν ότι δεν θα υπάρξουν ενδιαφέρουσες επεκτάσεις του Perceptron ακόμα και αν οι πολυεπίπεδες διαδικασίες βελτιωθούν. Η ροή της χρηματοδότησης των νευρωνικών δικτύων παύει προσωρινά. Το 1974 ανακαλύφθηκε από τον Werbos ο αλγόριθμος μάθησης Backpropagation το οποίο όμως δεν απέκτησε πολλή δημοσιότητα.
- Η νέα ανακάλυψη του backpropagation

Το 1986 ο backpropagation και άλλες διαδικασίες μάθησης για πολυεπίπεδα νευρωνικά δίκτυα ανακαλύφθηκαν ξανά. Επίσης, το 1982 ανακαλύφθηκε το δίκτυο Hopfield το οποίο εισήγαγε μια λειτουργία (energy function) η οποία αντιπροσωπεύει τις συνεχείς καταστάσεις ενός δικτύου. Το 1972, 1982 ανακαλύφθηκε ο αλγόριθμος αυτοοργάνωσης Kohonen στον οποίο οι νευρώνες οργανώνουν τον εαυτό τους. Βασικά συστατικά του είναι η ανταγωνιστική μάθηση και η αυτοοργάνωση.

#### 1.6.4 Κατηγορίες μάθησης:

- Επιβλεπόμενη μάθηση: Σ' αυτή την μάθηση έχουμε ένα 'δάσκαλο' ο οποίος λέει στο σύστημα τι να κάνει. Υπάρχει το ζεύγος εισόδου και η επιθυμητή έξοδος. Τα βάρη προσαρμόζονται έτσι ώστε να ελαχιστοποιείται η διαφορά μεταξύ των πραγματικών και των επιθυμητών αποτελεσμάτων για κάθε μορφή εισόδου. Στην παρούσα διπλωματική εργασία θα ασχοληθούμε μόνο με επιβλεπόμενη μάθηση. Ένα βασικό ζήτημα στην επιβλεπόμενη μάθηση είναι το **overfitting**. Σύμφωνα με τον Jaeger (2013 5<sup>th</sup> edition) [10], παρουσιάζεται το overfitting όταν το μοντέλο ταιριάζει με τα δεδομένα εκπαίδευσης πάρα πολύ καλά (στην ακραία περίπτωση το μοντέλο αναπαράγει τα δεδομένα εκπαίδευσης ακριβώς), δηλ. «έμαθε τα δεδομένα εκπαίδευσης από καρδιά» κι έτσι το μοντέλο δε θα γενικεύσει καλά. Το πρόβλημα είναι ιδιαίτερα σημαντικό για μικρά δείγματα εκπαίδευσης.
- Ενισχυτική μάθηση: Είναι η πιο βιολογικά ρεαλιστική μάθηση. Σ' αυτή την μάθηση έχουμε έναν 'κριτή' ο οποίος κρίνει εάν η πράξη είναι σωστή ή λάθος και επιβραβεύει ή τιμωρεί αναλόγως. Δεν γνωρίζουμε το επιθυμητό αποτέλεσμα.
- Μη επιβλεπόμενη μάθηση: Σ' αυτή την μάθηση δεν έχουμε ούτε κριτή ούτε δάσκαλο έχουμε μόνο δεδομένα εισόδου. Ομαδοποιεί δεδομένα τα οποία έχουν κοινά χαρακτηριστικά μεταξύ τους.

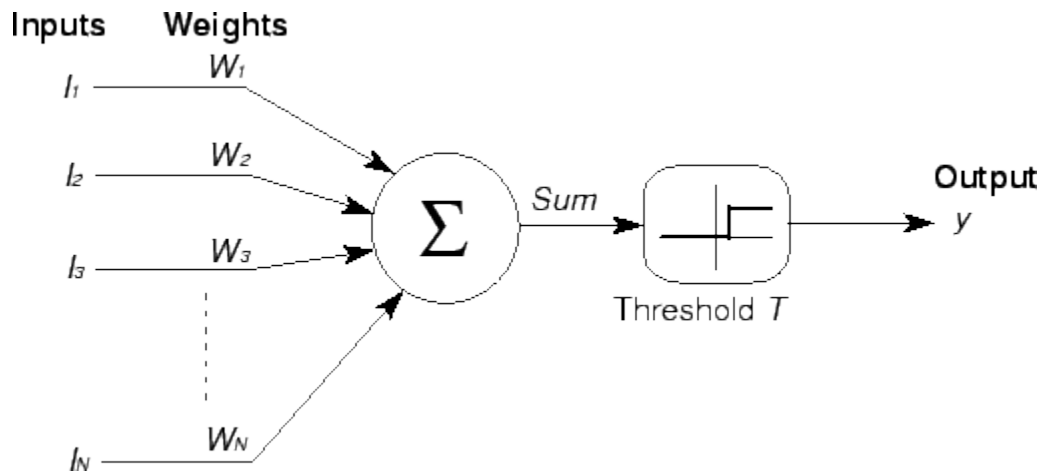
#### 1.6.5 Το μοντέλο Mc Culloch and Pitts (MCP)

Το πρώτο μοντέλο ενός τεχνητού νευρώνα εισάγεται από τους Warren McCulloch και Walter Pitts το 1943 (McCulloch and Pitts, 1943, [12]). Το MCP είναι ένας νευρώνας με εισόδους  $I_1, I_2, \dots, I_n$ , βάρη  $W_1, W_2, \dots, W_n$  και μια έξοδο  $y$ . (Σχήμα 1.9). Το γραμμικό κατώφλι κατηγοριοποιεί τις εισόδους σε δύο κλάσεις. Γι' αυτό και η έξοδος  $y$  παίρνει τιμή 0 ή 1. Η συνολική είσοδος υπολογίζεται πιο κάτω:

$$\text{Sum} = \sum_{i=1}^n W_i I_i \quad (1)$$

Οι τιμές των βαρών κυμαίνονται από (0,1) ή (-1,1). Sum είναι το άθροισμα των εισόδων και  $T$  το κατώφλι. Η έξοδος του τεχνητού νευρώνα προκύπτει από την εφαρμογή της συνάρτησης ενεργοποίησης  $f$  (activation function) στη συνολική του είσοδο Sum.

$$y = f(\text{Sum}) \quad (2)$$

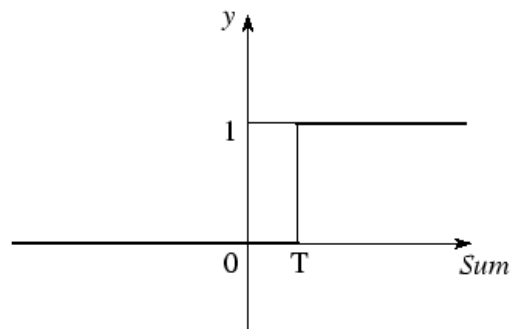


Σχήμα 1.9: Νευρώνας MCP

Υπάρχουν διάφορες περιπτώσεις συναρτήσεων ενεργοποίησης όπως η βηματική συνάρτηση (step function) (Σχήμα 1.10) ή συνάρτηση κατωφλίου (threshold function), η σιγμοειδής συνάρτηση (sigmoid ή logistic function) (Σχήμα 1.11) και άλλες.

Βηματική Συνάρτηση:

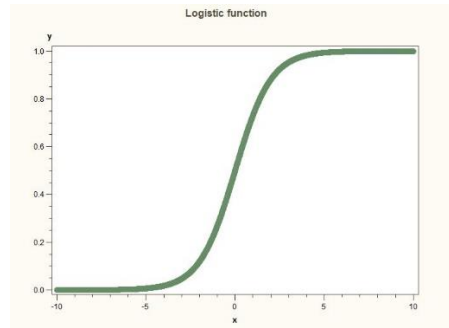
$$y(s) = \begin{cases} 0 & \text{αν } s > 0 \\ 1 & \text{αν } s \leq 0 \end{cases}$$



Σχήμα 1.10: Γραφική παράσταση βηματικής συνάρτησης

Σιγμοειδής συνάρτηση:

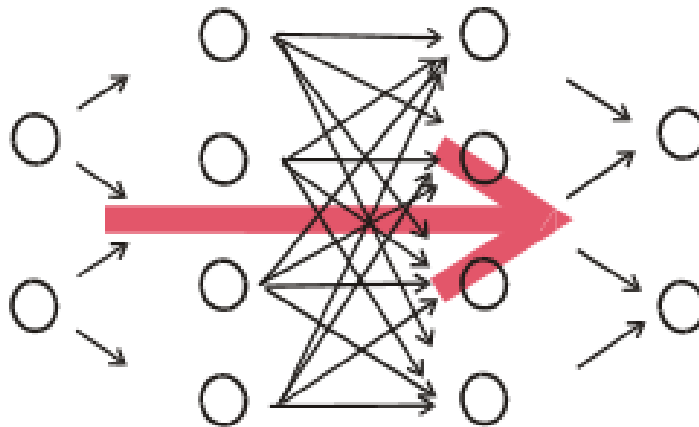
$$f(x) = \frac{1}{1 + e^{-ax}} \quad (3) \text{ όπου } x = \text{sum και } a \text{ η κλίση παραμέτρου } [-\infty, \infty]$$



Σχήμα 1.11: Γραφική Παράσταση σιγμοειδής συνάρτησης

### 1.6.6 Perceptrons

Τα Perceptrons είναι τα πιο απλά τεχνητά νευρωνικά δίκτυα. Αποτελούνται από μόνο ένα επίπεδο νευρώνων όπου η είσοδος του δικτύου είναι και η έξοδος, ή από πολλά επίπεδα νευρώνων (Multilayer Perceptron - MLP). (Σχήμα 1.12) Η αρχιτεκτονική των MLP αποτελείται από το επίπεδο εισόδου, τα κρυφά επίπεδα και το επίπεδο εξόδου. Η πληροφορία σε αυτή την αρχιτεκτονική ρέει εμπρόσθια δηλαδή από το επίπεδο εισόδου στο επίπεδο εξόδου. Σε αυτή την αρχιτεκτονική δεν υπάρχουν μονοπάτια – κύκλοι.



Σχήμα 1.12: Τυπική δομή ενός εμπρόσθιου δικτύου (Jaeger (2013)[10] )

## Κεφάλαιο 2

### Γνωσιολογικό Υπόβαθρο

---

|                                                                       |           |
|-----------------------------------------------------------------------|-----------|
| 2.1 Recurrent Neural Networks (RNN)                                   | <u>24</u> |
| 2.2 Echo State Networks (ESN)                                         | <u>24</u> |
| 2.3 Παραγωγή του DR                                                   | <u>26</u> |
| 2.3.1 Ανάλυση global παραμέτρων                                       | <u>26</u> |
| 2.3.2 Ρύθμιση για την επιλογή παραμέτρων                              | <u>28</u> |
| 2.3.3 Εκπαίδευση του Readout                                          | <u>29</u> |
| 2.3.4 Ridge Regression                                                | <u>29</u> |
| 2.3.5 Regularization                                                  | <u>30</u> |
| 2.4 Γενική ανάλυση των γραφικών παραστάσεων που εξαγάγει το πρόγραμμα | <u>30</u> |
| 2.4.1 Receiver Operating Characteristic – ROC Curve                   | <u>30</u> |
| 2.4.2 Cohen’s Kappa                                                   | <u>33</u> |
| 2.4.3 Accuracy                                                        | <u>33</u> |
| 2.4.4 Mean Square Error (MSE)                                         | <u>34</u> |
| 2.5 Ανάλυση Δεδομένων                                                 | <u>34</u> |

---



## 2.1 Recurrent Neural Networks

Η αρχιτεκτονική των ΤΝΔ με ανάδραση (Recurrent Neural Networks) αποτελείται και αυτή από τρία επίπεδα: το επίπεδο εισόδου, το κρυφό επίπεδο και το επίπεδο εξόδου. (Σχήμα 2.1). Σε αυτή την αρχιτεκτονική η έξοδος ενώνεται με την είσοδο του δικτύου ή με το κρυφό επίπεδο, δηλαδή δημιουργούνται κύκλοι. Με αυτό τον τρόπο το δίκτυο δημιουργεί ένα είδος βραχυπρόθεσμης μνήμης. Επεξεργάζεται την τρέχων είσοδο μαζί με την προηγούμενη έξοδο.

Σύμφωνα με τον τον Μ. Lukosevicius (2012) [11], η εκπαίδευση των RNN είναι εκ φύσεως δύσκολη. Ωστόσο, αντιπροσωπεύουν ένα πολύ ισχυρό εργαλείο, ενσωματώνοντας μεγάλη δυναμική μνήμη και μεγάλη ευελιξία προσαρμογής υπολογιστικών δυνατοτήτων. Είναι το μοντέλο της μηχανικής μάθησης το οποίο μοιάζει πολύ με το βιολογικό εγκέφαλο, το υπόστρωμα της φυσικής νοημοσύνης. Μια σημαντική ανακάλυψη των RNN είναι τα Echo State Networks (ESN) και τα Liquid State Machines (LSM) στην υπολογιστική νευροεπιστήμη. Η RNN προσέγγιση των ESN η οποία ονομάζεται Dynamic Reservoir (DR) δημιουργείται τυχαία και μόνο η έξοδος του DR (readout) εκπαιδεύεται. Η τάση ξεκίνησε από τα ESN και LSM όπου έγινε πρόσφατα γνωστή ως **Reservoir Computing** (RC). RC είναι σήμερα μεγάλος τομέας της έρευνας ο οποίος δίνει σημαντικές πληροφορίες για RNN, πρακτικά εργαλεία μηχανικής μάθησης.



Σχήμα 2.1: Τυπική δομή ενός RNN δικτύου (Jaeger (2013) [10])

## 2.2 Echo State Networks

Τα Echo State Network(ESN) επινοήθηκαν από τον Herbert Jaeger το 2002 και ανήκουν στην κατηγορία των RNN όπως αναφέρθηκε και προηγουμένως. Για να διατηρηθεί υψηλή ικανότητα μοντελοποίησης των RNN καθορίζεται μεγάλος αριθμός νευρώνων στο DR δηλ. το κρυφό επίπεδο των ESN. Σύμφωνα με τους J. Boedecker, O. Obst, J. T. Lizier, N. Michael Mayer και M. Asada [3], οι κύριες διαφορές των ESN με τα RNN είναι η ρύθμιση των συνδεδεμένων βαρών και η διαδικασία εκπαίδευσης. Για την κατασκευή ενός ESN οι μονάδες στο επίπεδο εισόδου και στο DR συνδέονται τυχαία. Μόνο οι συνδέσεις μεταξύ του DR και του επιπέδου εξόδου εκπαιδεύονται δηλ. το readout. Η εκπαίδευση γίνεται χρησιμοποιώντας επιβλεπόμενη μάθηση και χρησιμοποιώντας την προσέγγιση linear regression.

Στην παρούσα διπλωματική εργασία θα χρησιμοποιηθούν τα ESN για να κατηγοριοποιήσουμε τα EEG του BCI. Επιλέξαμε τα ESN διότι είναι δίκτυα με ανάδραση τα οποία ανήκουν στην επιβλεπόμενη μάθηση και τα οποία έχουν αποδειχτεί μια πολύ πρακτική προσέγγιση στην εκπαίδευση RNN. Επίσης είναι εννοιολογικά απλά δίκτυα και υπολογιστικά ανέξοδα, έτσι μπορούμε χρησιμοποιώντας τα να κατηγοριοποιήσουμε τα EEG. Μπορεί να είναι απλά δίκτυα, ωστόσο χρειάζεται κάποια εμπειρία στην προσαρμογή των παραμέτρων. Τα βάρη προσαρμόζονται έτσι ώστε να ελαχιστοποιείται η διαφορά μεταξύ πραγματικών και επιθυμητών αποτελεσμάτων για κάθε μορφή εισόδου.

Σύμφωνα με τον M. Lukosevicius (2012) [11], τα ESN εφαρμόζονται στην επιβλεπόμενη μάθηση όπου για ένα δεδομένο εισόδου  $u(n) \in \mathbb{R}^{N_u}$  η επιθυμητή έξοδος  $y^{target}(n) \in \mathbb{R}^{N_y}$  είναι γνωστή. (Σχήμα 2.2). Για  $n=1, \dots, T$  όπου  $T$  ο διακριτός χρόνος και  $T$  ο αριθμός των σημείων δεδομένων στο σύνολο δεδομένων εκπαίδευσης. Στόχος των ESN είναι να μάθουν ένα μοντέλο με έξοδο  $y(n) \in \mathbb{R}^{N_y}$  όπου  $y(n)$  αντιστοιχεί στο  $y^{target}(n)$  όσο το δυνατόν καλύτερα ελαχιστοποιώντας ένα μέτρο σφάλματος  $E(y, y^{target})$  και το πιο σημαντικό να γενικεύει καλά τα αόρατα δεδομένα. Το μέτρο σφάλματος  $E$  είναι το Mean Square Error (MSE), για παράδειγμα το Root Mean Square Error (RMSE) το οποίο είναι ο μέσος όρος για πάνω από  $N_y$  διαστάσεις  $i$  της εξόδου. **Όσο πιο μικρό είναι το  $E$  τόσο πιο κοντινή είναι η αντιστοίχιση στα δεδομένα.**

$$E(y, y^{target}) = \frac{1}{N_y} \sum_{i=1}^{N_y} \sqrt{\frac{1}{T} \sum_{n=1}^T (y_i(n) - y_i^{target}(n))^2} \quad (4)$$

Η εξίσωση που χρησιμοποιείται για την εκπαίδευση του DR είναι:

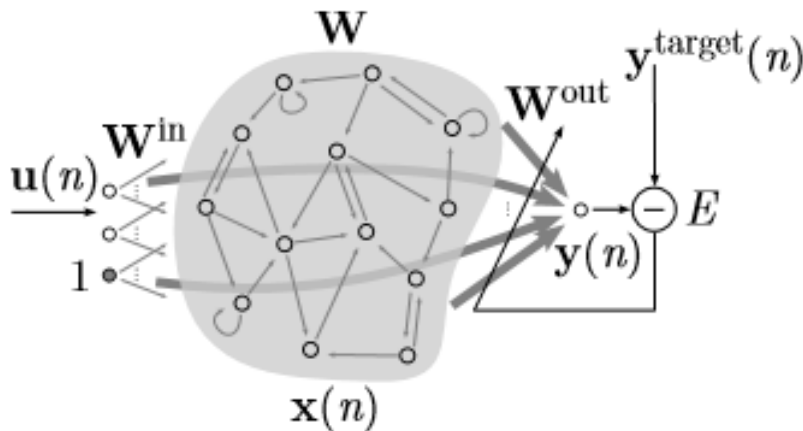
$$\tilde{x}(n) = \tanh(\mathbf{W}^{in}[1; \mathbf{u}(n)] + \mathbf{W}\mathbf{x}(n-1)) \quad (5)$$

$$\mathbf{x}(n) = (1 - \alpha)\mathbf{x}(n-1) + \alpha\tilde{x}(n) \quad (6)$$

όπου  $\mathbf{x}(n) \in \mathbb{R}^{N_x}$  είναι ένα διάνυσμα των ενεργοποιήσεων του DR και  $\tilde{x}(n) \in \mathbb{R}^{N_x}$  είναι η αναβάθμιση του, όλα στο χρονικό βήμα  $n$ .  $\mathbf{W}^{in} \in \mathbb{R}^{N_x \times (1+N_u)}$  και  $\mathbf{W} \in \mathbb{R}^{N_x \times N_x}$  είναι η είσοδος και τα βάρη των πινάκων αντίστοιχα, και  $\alpha \in (0,1]$  είναι ο ρυθμός διαρροής. Ο ορισμός του readout είναι ο εξής:

$$\mathbf{y}(n) = \mathbf{W}^{out}[1; \mathbf{u}(n); \mathbf{x}(n)] \quad (7)$$

όπου  $y(n) \in \mathbb{R}^{N_y}$  είναι η έξοδος του δικτύου,  $\mathbf{W}^{out} \in \mathbb{R}^{N_y \times (1+N_u+N_x)}$  ο πίνακας εξόδου.



Σχήμα 2.2: Ένα ESN (Lukosevicius (2012) [11])

Σύμφωνα με τον M. Lukosevicius (2012) [11], η αρχική μέθοδος RC που εισήχθη με τα ESN ήταν:

1. Η δημιουργία ενός μεγάλου τυχαίου DR RNN ( $W^{in}, W, a$ ).
2. Η εκτέλεση του χρησιμοποιώντας την είσοδο εκτέλεσης  $u(n)$  και τη συλλογή των αντίστοιχων καταστάσεων ενεργοποίησης του DR  $x(n)$ .
3. Ο υπολογισμός της εξόδου του  $W^{out}$ ,  $y(n)$ , (readout) γίνεται χρησιμοποιώντας linear regression και ελαχιστοποιώντας το MSE μεταξύ του  $y(n)$  και  $y^{target}(n)$ .
4. Χρησιμοποιώντας το εκπαιδευμένο δίκτυο στα νέα δεδομένα εισόδου  $u(n)$  γίνεται ο υπολογισμός του  $y(n)$  χρησιμοποιώντας τα εκπαιδευμένα βάρη εξόδου  $W^{out}$ .

## 2.3 Παραγωγή του DR

### 2.3.1 Ανάλυση global παραμέτρων

Σύμφωνα με τον M. Lukosevicius (2012) [11], είναι σημαντικό ότι το DR δρα ως μια μη γραμμική επέκταση και ως μνήμη εισόδου  $u(n)$  την ίδια στιγμή.

Το DR ορίζεται ως μια πλειάδα  $(W^{in}, W, a)$ . Η είσοδος και οι επαναλαμβανόμενες συνδέσεις των πινάκων  $W^{in}$  και  $W$  δημιουργούνται τυχαία σύμφωνα με κάποιες παραμέτρους και ο ρυθμός διαρροής (leaking rate)  $a$ , επιλέγεται ως ελεύθερη παράμετρος.

Στη συνέχεια θα αναλύσουμε τις global παραμέτρους του DR οι οποίες είναι το μέγεθος του DR  $N_x$ , η αραιότητα (sparsity), η διανομή των μη μηδενικών στοιχείων, spectral radius of  $W$ , scaling of  $W^{in}$  και ο ρυθμός διαρροής  $a$ .

- **Το μέγεθος του DR Nx:** Ο αριθμός των μονάδων του DR. Όσο μεγαλύτερο είναι το DR τόσο καλύτερη θα είναι η απόδοση υπό την προϋπόθεση ότι λαμβάνονται τα κατάλληλα μέτρα για να αποφευχθεί το overfitting. Για δύσκολες εργασίες είναι προτεινόμενο να χρησιμοποιηθεί μεγάλο μέγεθος DR όσο διατίθενται υπολογισμοί. Επίσης, είναι προτεινόμενο από τον M. Lukosevicius (2012) [11], να χρησιμοποιηθεί αρχικά μικρό DR για τη συμβατότητα των αποτελεσμάτων και στη συνέχεια να επεκταθεί σε μεγαλύτερο για την καλύτερη δυνατή απόδοση.
- **Η αραιότητα του DR (sparsity):** Είναι προτεινόμενο από τον M. Lukosevicius (2012) [11], το DR να είναι αραιό, δηλ. τα περισσότερα στοιχεία του Win να είναι 0. Οι αραιές συνδέσεις τείνουν να δίνουν μια ελαφρώς καλύτερη απόδοση. Σε γενικές γραμμές αυτή η παράμετρος δεν επηρεάζει και πολύ την απόδοση και έχει χαμηλή προτεραιότητα που πρέπει να βελτιστοποιηθεί. Είναι προτεινόμενο από τον M. Lukosevicius (2012) [11], να συνδέσουμε κάθε κόμβο του DR με ένα μικρό σταθερό αριθμό κόμβων (πχ. 10) κατά μέσο όρο, ανεξάρτητα από το μέγεθος του DR. Είναι επίσης προτεινόμενο η εκμετάλλευση αυτής της παραμέτρου για γρήγορους υπολογισμούς.
- **Διανομή μη μηδενικών στοιχείων:** Ο πίνακας W παράγεται τυπικά αραιός, με τα μη μηδενικά στοιχεία να είναι συμμετρικά ομοιόμορφα ή να επικεντρώνεται στην κανονική κατανομή γύρω από το 0. Διαφορετικοί συγγραφείς προτιμούν διαφορετικές κατανομές. Οι Gaussian κατανομές είναι επίσης δημοφιλείς.
- **Spectral Radius:** Μία από τις πιο σημαντικές παραμέτρους για τις συνδέσεις του DR στον πίνακα W, δηλ. η μέγιστη απόλυτη ιδιοτιμή αυτού του πίνακα. Σύμφωνα με τον F.Grezes(2014) [8], η ιδιότητα echo state δηλώνει ότι το αποτέλεσμα της προηγούμενης κατάστασης και η προηγούμενη εισαγωγή σε μια μελλοντική κατάσταση θα πρέπει να εξαφανίζεται σταδιακά με το πέρασμα του χρόνου, και δεν εξακολουθεί να υπάρχει ή, ακόμη χειρότερα, να ενισχύεται. Διαισθητικά, το spectral radius είναι ένα ακατέργαστο μέτρο της ποσότητας της μνήμης όπου το DR μπορεί να κρατήσει. Οι μικρές τιμές αντιπροσωπεύουν μια σύντομη μνήμη, και οι μεγάλες μεγαλύτερη μνήμη, μέχρι το σημείο της υπερ-ενίσχυσης όταν η echo state ιδιότητα δεν ισχύει πλέον. Σύμφωνα με τον M. Lukosevicius (2012) [11], τυπικά παράγεται ένα τυχαία αραιό W, το spectral radius  $\rho(W)$  υπολογίζεται, έτσι το W μοιράζεται σε  $\rho(W)$  για να δώσει έναν πίνακα με μία μονάδα spectral radius. Μεγάλες τιμές  $\rho(W)$  μπορεί να οδηγήσουν σε DR με πολλά σταθερά σημεία, περιοδικά ή ακόμη και χαοτικά. Είναι προτεινόμενο από τον M. Lukosevicius (2012) [11], να δώσουμε τιμές  $\rho(W) < 1$  επειδή τις περισσότερες φορές εξασφαλίζει ποιότητα στα ESN. Στην πράξη  $\rho(W)$  μπορεί να επιλεγεί για να μεγιστοποιηθεί η απόδοση, με τιμή 1 που χρησιμεύει ως αρχικό σημείο αναφοράς. Είναι επίσης προτεινόμενο από τον M. Lukosevicius (2012) [11], ότι το spectral radius πρέπει να είναι μεγαλύτερο σε εργασίες που απαιτούν περισσότερη μνήμη από της εισόδου.
- **Κλιμάκωση του πίνακα εισόδου:** Η κλιμάκωση του πίνακα Win είναι μια άλλη βασική παράμετρος για την βελτιστοποίηση του ESN. Για ομοιόμορφα κατανομημένο Win ορίζουμε την κλιμάκωση εισόδου a ως το εύρος του διαστήματος  $[-a, a]$  τιμές οι οποίες είναι δείγμα των τιμών του Win. Είναι προτεινόμενο από τον M. Lukosevicius (2012) [11], να κλιμακωθεί ολόκληρος ο πίνακας Win έτσι ώστε να έχουμε λίγες καθολικές παραμέτρους στο ESN. Επίσης, για να αυξηθεί η απόδοση προτείνει να κλιμακωθεί

πρώτα η πρώτη στήλη του Win δηλ. η είσοδος του κατωφλίου ξεχωριστά και στη συνέχεια να κλιμακωθούν οι υπόλοιπες στήλες του Win. Επίσης προτείνει να ομαλοποιηθούν τα δεδομένα έτσι ώστε να κρατηθεί η είσοδος  $u(n)$  σε όρια και να αποφευχθούν οι ακραίες τιμές, προτείνει να χρησιμοποιηθεί η συνάρτηση  $\tanh()$  στην είσοδο  $u(n)$  αν είναι απεριόριστη. Η κλιμάκωση του πίνακα εισόδου ρυθμίζει το ποσό της μη γραμμικότητας της αναπαράστασης του DR  $x(n)$  (το οποίο αυξάνεται επίσης με το  $\rho(W)$ ) κι επίσης ρυθμίζει το σχετικό αποτέλεσμα της τρέχωντος εισόδου  $x(n)$ ).

- **Ρυθμός διαρροής:** Ο ρυθμός διαρροής  $\alpha$  των κόμβων του DR μπορεί να θεωρηθεί ως η ταχύτητα της δυναμικής αναβάθμισης του DR διακριτή στο χρόνο. Είναι προτεινόμενο από τον M. Lukosevicius (2012) [11], να καθορίζουμε τον ρυθμό διαρροής έτσι ώστε να ταιριάζει με την ταχύτητα της δυναμικής του  $u(n)$  και/ή του  $y^{\text{target}}(n)$ . Αυτό μπορεί να είναι δύσκολο και υποκειμενικό να προσδιοριστεί σε κάποιες περιπτώσεις. Ειδικά όταν το χρονοδιάγραμμα του  $u(n)$  και  $y^{\text{target}}(n)$  είναι πολύ διαφορετικό. Είναι μια ακόμη καθολική παράμετρος που πρέπει να προσδιοριστεί με δοκιμή και σφάλμα. Όταν η εργασία απαιτεί μοντελοποίηση της χρονοσειράς που παράγει δυναμικό σύστημα σε πολλαπλές χρονικές κλίμακες, θα ήταν ίσως χρήσιμο να καθοριστούν διαφορετικοί ρυθμοί διαρροής σε διαφορετικές μονάδες (κάνοντας  $\alpha$  ένα διάνυσμα  $\alpha \in \mathbb{R}^{N_x}$ ) με ένα πιθανό μειονέκτημα ότι θα έχουμε περισσότερες παραμέτρους για βελτιστοποίηση. Σε ορισμένες περιπτώσεις ο καθορισμός ενός μικρού  $\alpha$ , οδηγεί στην αργή δυναμική του  $x(n)$ , μπορεί να αυξήσει δραματικά την διάρκεια της βραχυπρόθεσμης μνήμης σε ESN.

Οι τρεις βασικές παραμέτροι για να βελτιστοποιηθεί το ESN DR είναι η κλιμάκωση του πίνακα εισόδου, το spectral radius και ο ρυθμός διαρροής. Αυτές οι παράμετροι είναι πολύ σημαντικές για καλή απόδοση. Το μέγεθος του DR  $N_x$  μπορεί να προσδιοριστεί ως εξωτερικός περιορισμός και οι υπόλοιπες παραμέτροι μπορεί να ρυθμιστούν σε λογικές προεπιλεγμένες τιμές όπως η αραιότητα του DR (2) και η διανομή μη μηδενικών στοιχείων (3) οι οποίες έχουν χαμηλή προτεραιότητα.

Η απόδοση μπορεί επίσης να βελτιστοποιηθεί σε πολλές περιπτώσεις διαχωρίζοντας μια παράμετρο σε πολλές. Ρυθμίζοντας τις στήλες του Win, το οποίο αντιστοιχεί με το bias input. Επίσης, ρυθμίζοντας το ρυθμό διαρροής  $\alpha$ , σε διαφορετικές μονάδες μπορεί να βοηθήσει σε πολυπροθεσμιακές εργασίες.

### 2.3.2 Ρύθμιση για την επιλογή παραμέτρων

Ένα από τα βασικά πλεονεκτήματα του ESN είναι ότι η εκπαίδευση της εξόδου είναι γρήγορη. Αυτό πρέπει να αξιοποιηθεί για την αξιολόγηση του πόσο καλό είναι ένα DR που παράγεται από ένα συγκεκριμένο σύνολο παραμέτρων. Ο πιο ρεαλιστικός τρόπος για να αξιολογήσεις το DR είναι το να εκπαιδεύσεις την έξοδο και να μετρήσεις το λάθος.

Είναι σημαντικό να έχουμε υπόψην μας ότι τα DR που δημιουργούνται τυχαία ακόμα και με τις ίδιες παραμέτρους διαφέρουν ελαφρώς από τις επιδόσεις τους. Αυτή η διακύμανση είναι πάντοτε παρόν αλλά τυπικά είναι πιο έντονη με μικρότερο DR παρά με μεγαλύτερο. Είναι

προτεινόμενο από τον M. Lukosevicius (2012) [11], ότι για να εξαλειφθεί η τυχαία διακύμανση της απόδοσης, να κρατάμε τους τυχαίους αριθμούς σταθερούς ή/και κατά μέσο όρο κατά τη διάρκεια αρκετών δειγμάτων του DR. Επίσης προτείνει να αλλάζουμε μια παράμετρο κάθε φορά για να έχουμε καλή απόδοση αφού αν αλλάξουμε πολλές παραμέτρους ταυτόχρονα είναι αδύνατο να γνωρίζουμε ποια παράμετρος συνέβαλε σε αυτό. Μια λογική προσέγγιση είναι να ορίσουμε μια μόνο παράμετρο σε μια αρκετά καλή τιμή πριν να ξεκινήσουμε ν'αλλάζουμε άλλη παράμετρο, και αυτό επαναλαμβάνεται μέχρι η απόδοση να είναι ικανοποιητική. Είναι επίσης σκόπιμο το να κρατάς σημειώσεις για εκτεταμένες βελτιώσεις έτσι ώστε να μην κάνεις κύκλους επαναλαμβάνοντας τις ίδιες τιμές παραμέτρων. Παράλληλα ο M. Lukosevicius (2012) [11], προτείνει πάντα να βγάζεις γραφικές παραστάσεις της ενεργοποίησης του DR  $x(n)$  έτσι ώστε να καταλαμβαίνεις τι συμβαίνει σε αυτό.

### 2.3.3 Εκπαίδευση των ESN

Σύμφωνα με τον V.Ceperie και A.Baric (2014) [4], η εκπαίδευση των ESN γίνεται με τα ακόλουθα βήματα:

1. Δημιούργησε ένα τυχαίο DR RNN με  $N$  νευρώνες, χρησιμοποιώντας κάποιο νευρωνικό μοντέλο. Στην περίπτωση μας θα χρησιμοποιηθεί η  $\tanh$ . Σύνδεσε τις μονάδες εισόδου του DR δημιουργώντας τυχαίες συνδέσεις. Αν η εργασία απαιτεί οπισθοδρόμηση εξόδου δημιούργησε τυχαία συνδέσεις από την έξοδο στο DR.
2. Οδήγησε το DR με τα δεδομένα εκπαίδευσης στο διακριτό χρονικό βήμα  $n$ . Αυτό έχει ως αποτέλεσμα μια ακολουθία  $x(n)$  με  $n$  καταστάσεις του DR. Κάθε σήμα  $x(n)$  είναι ένας μη γραμμικός μετασχηματισμός της εισόδου.
3. Υπολόγισε τα βάρη εξόδου από το DR στην έξοδο ως γραμμική παλινδρόμηση (linear regression) των βαρών των εκπαιδευμένων εξόδων  $y^{\text{target}}(n)$  στις καταστάσεις  $x(n)$  του DR. Χρησιμοποίησε αυτά τα βάρη για να δημιουργήσεις τις συνδέσεις από το DR στην έξοδο.

### 2.3.4 Ridge Regression:

Αφού το readout του ESN είναι τυπικά γραμμικό και εμπρόσθιο η εξίσωση 4 μπορεί να γραφτεί ως:

$$Y = W^{\text{out}}X \quad (8)$$

Όπου  $Y \in \mathbb{R}^{N_y \times T}$  είναι το  $y(n)$  και  $X \in \mathbb{R}^{(1+N_u+N_x) \times T}$  είναι το  $[1; u(n); x(n)]$ .

Η εύρεση των βέλτιστων βαρών  $W_{\text{out}}$  οι οποίες ελαχιστοποιούν το τετραγωνικό σφάλμα μεταξύ  $y(n)$  και  $y^{\text{target}}(n)$  ανέρχεται στην επίλυση ενός τυπικού υπερπροσδιορισμένου συστήματος γραμμικών εξισώσεων.

$$Y^{\text{target}} = W^{\text{out}}X \quad (9)$$

Όπου  $\mathbf{Y}^{\text{target}} \in \mathbb{R}^{N \times T}$ . Υπάρχουν πολλοί τρόποι για να λυθεί το (9) στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε ridge regression που ορίζεται από τον τύπο (10).

$$\mathbf{W}^{\text{out}} = \mathbf{Y}^{\text{target}} \mathbf{X}^T (\mathbf{X} \mathbf{X}^T + \beta \mathbf{I})^{-1} \quad (10)$$

Όπου  $\beta$  είναι το regularization coefficient (εξηγείται στην επόμενη ενότητα) και  $\mathbf{I}$ , identity matrix.

Ο M. Lukosevicius (2012) [11], συνιστά ότι ο τρόπος για να εκπαιδεύσουμε τη γραμμική έξοδο των βαρών του ESN είναι με το ridge regression.

### 2.3.5 Regularization

Για να εκτιμηθεί η ποιότητα της λύσης που παράγεται από την εκπαίδευση, είναι σκόπιμο να παρακολουθούνται τα βάρη εξόδου  $\mathbf{W}^{\text{out}}$ . Μεγάλα βάρη δείχνουν ότι το  $\mathbf{W}^{\text{out}}$  εκμεταλλεύεται και ενισχύει μικρές διαφορές μεταξύ του  $x(n)$  και μπορεί να είναι πολύ ευαίσθητο σε αποκλίσεις από τις ακριβείς συνθήκες υπό τις οποίες έχει εκπαιδευτεί το δίκτυο. Αυτό είναι ένα μεγάλο πρόβλημα στις ρυθμίσεις όπου το δίκτυο λαμβάνει την έξοδο ως την επόμενη είσοδο. Εξαιρετικά μεγάλες τιμές  $\mathbf{W}^{\text{out}}$  μπορεί να αποτελέσουν ένδειξη μιας πολύ ευαίσθητης και ασταθής λύσης.

Ο M. Lukosevicius (2012) [11], προτείνει να κάνουμε regularization όταν υπάρχει ο κίνδυνος του overfitting ή ασταθής ανατροφοδότηση. Επίσης, προτείνει να επιλέξουμε το  $\beta$  για ένα συγκεκριμένο ESN χρησιμοποιώντας επαλήθευση χωρίς να εκτελείται το DR μέσω των δεδομένων εκπαίδευσης. Οι βέλτιστες τιμές του  $\beta$  μπορεί να ποικίλουν από πολλά μεγέθη, ανάλογα με το ακριβές παράδειγμα του DR και το μήκος των δεδομένων εκπαίδευσης.

## 2.4 Γενική ανάλυση των γραφικών παραστάσεων που εξαγάγει το πρόγραμμα

### 2.4.1 Receiver Operating Characteristic – ROC Curve

Με βάση τον T.Fawcett(2005) [6], μια ROC γραφική παράσταση είναι μια τεχνική για την απεικόνιση, οργάνωση και την επιλογή κατηγοριοποιητών με βάση την απόδοσή τους.

#### Απόδοση Κατηγοριοποιητή

Στην παρούσα διπλωματική εργασία θα μελετηθεί η κατηγοριοποίηση σε δύο κατηγορίες (κίνηση αριστερού χεριού (1) και κίνηση δεξιού χεριού (-1)). Κάθε περίπτωση  $i$ , αντιστοιχεί σε ένα στοιχείο του συνόλου  $\{p, n\}$  που είναι η θετική και αρνητική κατηγορία. Ένας κατηγοριοποιητής είναι η αντιστοίχιση των περιπτώσεων στις προβλεπόμενες κατηγορίες. Για

να διακρίνουμε μεταξύ της πραγματικής και προβλεπόμενης κατηγορίας χρησιμοποιούμε ετικέτες {Y,N} για τις προβλέψεις της κατηγορίας.

Δεδομένου ενός κατηγοριοποιητή και μιας περίπτωσης υπάρχουν τέσσερα πιθανά αποτελέσματα. Αν το παράδειγμα είναι θετικό και κατηγοριοποιείται ως θετικό, υπολογίζεται ως true positive. Αν είναι θετικό και κατηγοριοποιείται ως αρνητικό υπολογίζεται ως false negative. Αν το παράδειγμα είναι αρνητικό και κατηγοριοποιείται ως αρνητικό υπολογίζεται ως true negative. Αν είναι αρνητικό και κατηγοριοποιείται ως θετικό υπολογίζεται ως false positive. Δεδομένου ενός κατηγοριοποιητή και ενός συνόλου περιπτώσεων (test set), ένας 2x2 confusion matrix μπορεί να αναπαραστήσει τις διατάξεις του συνόλου των περιπτώσεων. (Πίνακας 2.1)

Οι αριθμοί της κύριας διαγωνίου αντιπροσωπεύουν τις σωστές αποφάσεις και οι αριθμοί της άλλης διαγωνίου τα σφάλματα.

|          | <b>p</b>        | <b>n</b>        |
|----------|-----------------|-----------------|
| <b>Y</b> | True Positives  | False Positives |
| <b>N</b> | False Negatives | True Negatives  |

*Πίνακας 2.1: Confusion matrix*

Το True Positive Rate (TPR) (ονομάζεται και hit rate και recall) ενός κατηγοριοποιητή και εκτιμάται ως:

$$TPR = \frac{\text{θετικοί που κατηγοριοποιήθηκαν σωστά}}{\text{σύνολο θετικών}} \quad (11)$$

False Positive Rate (FPR) (ονομάζεται και false alarm rate) ενός κατηγοριοποιητή και εκτιμάται ως:

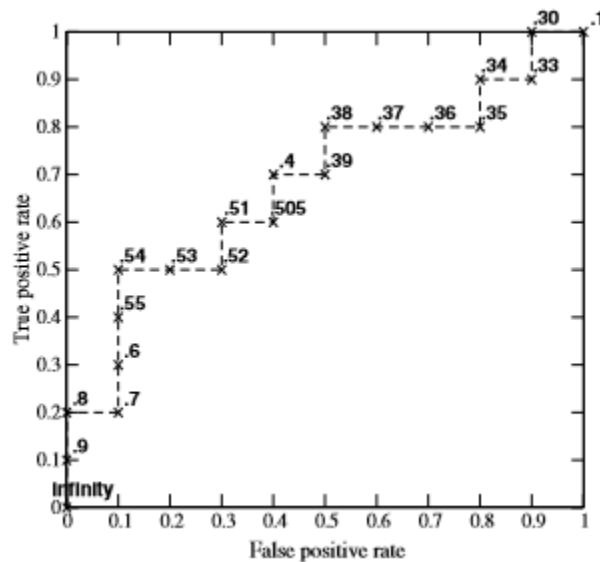
$$FPR = \frac{\text{αρνητικοί που κατηγοριοποιήθηκαν σωστά}}{\text{σύνολο αρνητικών}} \quad (12)$$

Στον άξονα Y μιας ROC γραφικής παράστασης απεικονίζεται το TPR και στον X το FPR.



| Inst# | Class | Score | Inst# | Class | Score |
|-------|-------|-------|-------|-------|-------|
| 1     | p     | .9    | 11    | p     | .4    |
| 2     | p     | .8    | 12    | n     | .39   |
| 3     | n     | .7    | 13    | p     | .38   |
| 4     | p     | .6    | 14    | n     | .37   |
| 5     | p     | .55   | 15    | n     | .36   |
| 6     | p     | .54   | 16    | n     | .35   |
| 7     | n     | .53   | 17    | p     | .34   |
| 8     | n     | .52   | 18    | n     | .33   |
| 9     | p     | .51   | 19    | p     | .30   |
| 10    | n     | .505  | 20    | n     | .1    |

Σχήμα 2.3: Ο πίνακας παρουσιάζει 20 δεδομένα και το σκορ που ανατέθηκε για κάθε κατηγοριοποιητή. (Fawcett (2005) [6])



Σχήμα 2.4: Η αντίστοιχη καμπύλη ROC των δεδομένων του σχήματος 2.3. (Fawcett (2005) [6])

Ένα παράδειγμα μιας ROC curve γραφικής παράστασης με 20 περιπτώσεις, 10 θετικές και 10 αρνητικές απεικονίζεται στα σχήματα 2.3 και 2.4.

Κάθε καμπύλη ROC που παράγεται από ένα σύνολο πεπερασμένων περιπτώσεων είναι στην πραγματικότητα μια βηματική συνάρτηση (step function) η οποία προσεγγίζει μια αληθής καμπύλη ως τον αριθμό των περιπτώσεων που προσεγγίζουν το άπειρο.

### Area Under a ROC Curve (AUC)

Μια ROC καμπύλη είναι μια απεικόνιση δύο διαστάσεων της απόδοσης του κατηγοριοποιητή. Για να συγκρίνουμε κατηγοριοποιητές χρειάζεται να γνωρίζουμε μια τιμή που να αντιπροσωπεύει την απόδοση του κατηγοριοποιητή. Η τιμή της AUC είναι πάντοτε από 0-1. Η τιμή 1 αντιπροσωπεύει την τέλεια αντιστοίχιση ενώ το 0 τη μηδενική αντιστοίχιση και το 0.5 τη φθηνή αντιστοίχιση.

#### **2.4.2 Cohen's Kappa**

Σύμφωνα με τον J.Cohen (1960) [5], Cohen's Kappa είναι μια τιμή που εκφράζει το επίπεδο συμφωνίας μεταξύ δύο παρατηρητών σε ένα πρόβλημα κατηγοριοποίησης. Αντιπροσωπεύει το βαθμό στον οποίο τα δεδομένα που συλλέγονται στη μελέτη είναι σωστές αναπαραστάσεις των μεταβλητών που μετρήθηκαν. Ορίζεται ως:

$$\kappa = (p_o - p_e) / (1 - p_e) \quad (13)$$

όπου  $p_o$  είναι η εμπειρική πιθανότητα συμφωνίας στην ετικέτα που εκχωρήθηκε σε οποιοδήποτε δείγμα (ο παρατηρούμενος δείκτης συμφωνίας), και  $p_e$  είναι η αναμενόμενη συμφωνία όταν και οι δύο παρατηρητές εκχωρήσουν ετικέτες τυχαία, και 1 η πλήρης συμφωνία. Η τιμή  $\kappa$  παίρνει τιμές  $[-1,1]$  οι οποίες αντιπροσωπεύουν το μέγεθος της συμφωνίας όπως φαίνεται στον πίνακα 2.2.

| Τιμή $\kappa$ | Μέγεθος συμφωνίας |
|---------------|-------------------|
| $\leq 0$      | Μηδενική          |
| 0.01-0.2      | Φτωχή             |
| 0.21-0.4      | Μικρή             |
| 0.41-0.6      | Μέτρια            |
| 0.61-0.8      | Σημαντική         |
| 0.81-0.99     | Σχεδόν Πλήρης     |
| 1             | Πλήρης            |

*Πίνακας 2.2: Για κάποια τιμή  $\kappa$  το αντίστοιχο μέγεθος της συμφωνίας της*

#### **2.4.3 Accuracy**

Η ακρίβεια της κατηγοριοποίησης  $A_i$  ενός μεμονωμένου προγράμματος  $i$  εξαρτάται από τον αριθμό των δειγμάτων που ταξινομούνται σωστά και αξιολογείται από τον πιο κάτω τύπο:

$$A_i = \left(\frac{t}{n}\right) \cdot 100 \quad (14)$$

Όπου  $t$  ο αριθμός του δείγματος που ταξινομείται σωστά και  $n$  ο συνολικός αριθμός του δείγματος.

Παίρνει τιμή από 0-1 η οποία αντιπροσωπεύει το ακριβές ποσοστό της πρόβλεψης. Όσο πιο μεγάλο τόσο καλύτερη πρόβλεψη επιτεύχθηκε.

#### 2.4.4 Mean Square Error - MSE

Όπως αναφέραμε και στην ενότητα 2.2 ένα μέτρο σφάλματος  $E(y, y^{target})$  είναι το MSE που ορίζεται από τον πιο κάτω τύπο:

$$\frac{1}{n} \sum_{i=0}^n (y^{target} - y)^2 \quad (15)$$

Όσο πιο μικρό είναι το E τόσο πιο κοντινή είναι η αντιστοίχιση στα δεδομένα.

#### 2.5 Ανάλυση Δεδομένων

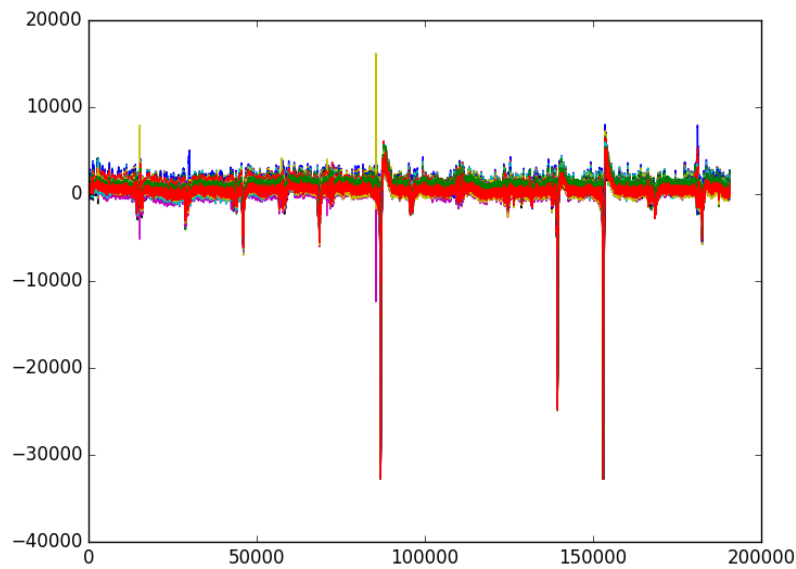
Τα δεδομένα του προβλήματος είναι δύο MATLAB αρχεία τα οποία είναι δύο πίνακες. Ο πίνακας στο BClinputdata100Hz.mat είναι η είσοδος του προβλήματος και ο πίνακας στο BClinputdata100Hz.mat είναι η επιθυμητή έξοδος.

Ο πίνακας εισόδου έχει διαστάσεις 190594x59 ο οποίος περιέχει τιμές από -32768 μέχρι 16188. Κάθε γραμμή του πίνακα αντιπροσωπεύει έναν αισθητήρα ο οποίος καταγράφει την ενεργοποίηση του εγκεφάλου από το BCI.

Ο πίνακας εξόδου έχει διαστάσεις 190594x1 ο οποίος περιέχει τιμές 0,1,-1. Το 0 αντιπροσωπεύει ότι δεν υπάρχει κίνηση, το 1 αντιπροσωπεύει την κίνηση αριστερού χεριού και το -1 την κίνηση δεξιού χεριού.

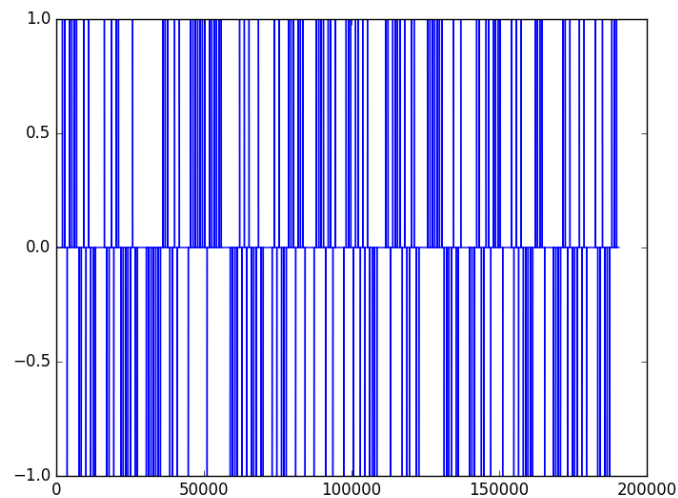
Κάθε γραμμή του πίνακα εισόδου αντιστοιχεί με μια γραμμή από τον πίνακα εξόδου όπου φαίνεται για κάθε καταγραφή του εγκεφάλου ποια είναι η επιθυμητή έξοδος. Δηλαδή, οι 59 τιμές της πρώτης γραμμής του πίνακα εισόδου αντιπροσωπεύουν μια καταγραφή του εγκεφάλου η οποία αντιστοιχεί στην πρώτη γραμμή του πίνακα εξόδου η οποία είναι 0 δηλ. δεν έχω κίνηση.

Η είσοδος στο ESN είναι μια χρονοσειρά (window) δηλ. είναι ένα μέρος μιας από τις γραμμές του πίνακα εισόδου.



*Σχήμα 2.5: Γραφική παράσταση των δεδομένων εισόδου*

Η πιο πάνω γραφική παράσταση αναπαριστά τα δεδομένα εισόδου. Ο άξονας  $y$  αναπαριστά το εύρος αριθμών των δεδομένων το οποίο είναι  $[-32768, 16188]$ . Στον άξονα  $x$  είναι κάποιοι αριθμοί που εξαγάγει η ρύθση εξ' ορισμού οι οποίοι ξεκινούν από το 0.



*Σχήμα 2.6: Γραφική παράσταση των δεδομένων εξόδου*

Η πιο πάνω γραφική παράσταση αναπαριστά τα δεδομένα εξόδου. Ο άξονας  $y$  αναπαριστά το εύρος αριθμών των δεδομένων το οποίο είναι  $0,1,-1$ . Στον άξονα  $x$  είναι κάποιοι αριθμοί που εξαγάγει η ρύθση εξ' ορισμού οι οποίοι ξεκινούν από το 0.

## Κεφάλαιο 3

### Σχεδιασμός και Υλοποίηση

---

|                                         |           |
|-----------------------------------------|-----------|
| 3.1 Ανάλυση παραμέτρων του προγράμματος | <u>36</u> |
| 3.2 Διάγραμμα Ροής του προγράμματος     | <u>38</u> |

---

Στην παρούσα διπλωματική εργασία για την κατηγοριοποίηση των ηλεκτροεγκεφαλογραφημάτων σε κίνηση δεξιού χεριού ή κίνηση αριστερού χεριού βασίστηκε πάνω στον αλγόριθμο που έγραψε ο Mantas Lukoševičius (2012) [11], <http://minds.jacobs-university.de/mantas>. Ο κώδικας βρίσκεται στο Παράρτημα 1.

#### 3.1 Ανάλυση παραμέτρων του προγράμματος

Πιο κάτω αναλύονται οι παράμετροι του προγράμματος.

##### ***windowSize:***

Σύμφωνα με την Ε.Παπακώστα (2016) [19], για να αυξηθεί η έννοια της αλληλεξάρτησης των δεδομένων εισόδου στο δίκτυο χρησιμοποιήθηκε ένα χρονικό κινητό παράθυρο το οποίο ορίζει το χρονικό διάστημα με βάση το οποίο θα μεταβάλλονται τα βάρη του δικτύου. Το χρονικό κινητό παράθυρο είναι μια τεχνική που χρησιμοποιείται για την πρόβλεψη χρονοσειρών και συνηθίζεται σε αρχιτεκτονικές όπως νευρωνικά δίκτυα με ανάδραση RNN. Με βάση το χρονικό κινητό παράθυρο τα δεδομένα εισόδου συσχετίζονται με άλλα δεδομένα εισόδου τα οποία βρίσκονται μεταξύ του χρονικού περιθωρίου που ορίζει η παράμετρος *windowSize*.

##### ***numberOfChannels:***

Ο αριθμός των καναλιών, στην περίπτωση μας είναι 59 όσες και οι εισόδους στο δίκτυο. Ο αριθμός αυτός παραμένει σταθερός.

##### ***numberOfEpochs:***

Εποχή είναι η παρουσίαση μία φορά όλων των δεδομένων που πρόκειται να εκπαιδευτούν. Στην παρούσα διπλωματική εργασία χρησιμοποιήθηκε μια εποχή λόγω του μεγάλου μεγέθους του DR.

***numOfInputRecords:***

Χρησιμοποιήθηκε για να προσδιοριστεί το εύρος τιμών του διανύσματος αντιστοίχισης εισόδου. (inputMapping)

***inSize:***

Καθορίζει τον αριθμό των νευρώνων στο επίπεδο εισόδου. Ο αριθμός αυτός παραμένει σταθερός 59. (Κεφάλαιο 2.5 Ανάλυση Δεδομένων)

***outsize:***

Καθορίζει τον αριθμό των νευρώνων στο επίπεδο εξόδου. Ο αριθμός αυτός παραμένει σταθερός 1. (Κεφάλαιο 2.5 Ανάλυση Δεδομένων)

***resSize:***

Καθορίζει τον αριθμό των νευρώνων στο DR. Μετά από πειράματα καθορίστηκε ο αριθμός του. (Κεφάλαιο 2.3.1 Ανάλυση global παραμέτρων: Το μέγεθος του DR Nx)

***a:***

Μετά από πειράματα καθορίστηκε ο αριθμός του. (Κεφάλαιο 2.3.1 Ανάλυση global παραμέτρων: Ρυθμός διαρροής)

***SpParam:***

Καθορίζει πόσο αραιό θα είναι το DR. Πιο συγκεκριμένα καθορίζει τον αριθμό των συναπτικών ενώσεων που θα υπάρχουν στο DR (Κεφάλαιο 2.3.1 Ανάλυση global παραμέτρων: Η αραιότητα του DR (sparsity))

***regCoef:***

Κανονικοποίηση coefficient. (Τύπος 9, Κεφάλαιο 2.3.5)

***numOfExp:***

Ο αριθμός πειραμάτων δηλ. πόσες φορές το πρόγραμμα θα μπει στο βρόχο.

***resPercForinitWMethod2:***

Η πυκνότητα του πίνακα W.

***initWMethod:***

Χρησιμοποιείται σαν σημαία για το πώς θα αρχικοποιηθεί ο πίνακας W. Αν είναι 1 θα αρχικοποιηθεί με τυχαίες τιμές με εύρος [-0.5, 0.5] και δεδομένου των διαστάσεων του

πίνακα, διαφορετικά θα αρχικοποιηθεί δεδομένου των διαστάσεων του πίνακα και της πυκνότητας της παραμέτρου resPercForinitWMethod2.

#### ***modifyWMethjod:***

Χρησιμοποιείται σαν σημαία για το πώς θα αρχικοποιηθεί ο πίνακας W χρησιμοποιώντας την τιμή του spectral radius (Υποκεφάλαιο 2.3.1). Αν είναι 1 θα χρησιμοποιηθεί ο τύπος

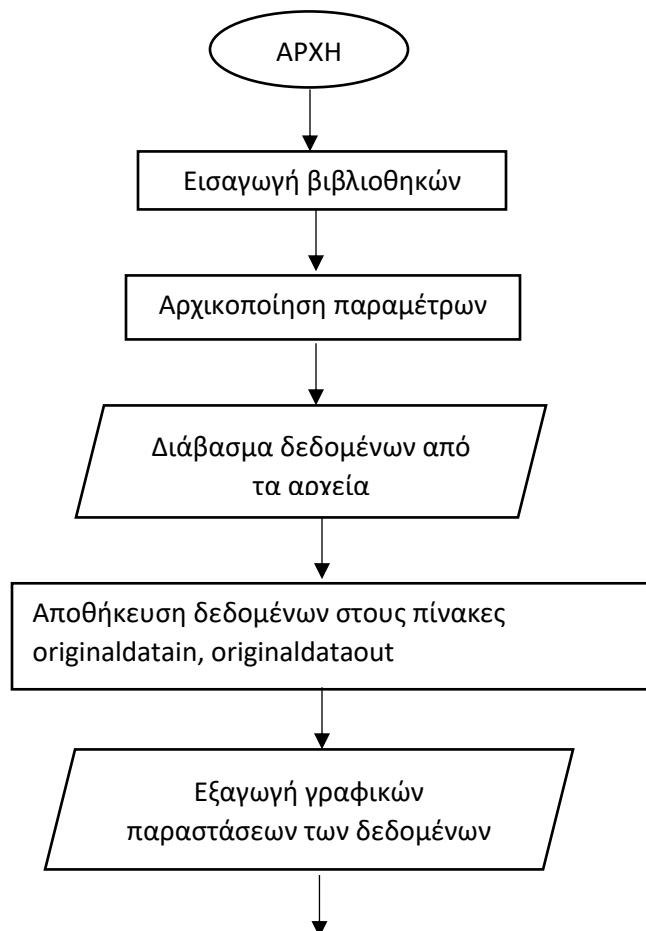
$$W * = SpParam / \text{spectral radius}$$

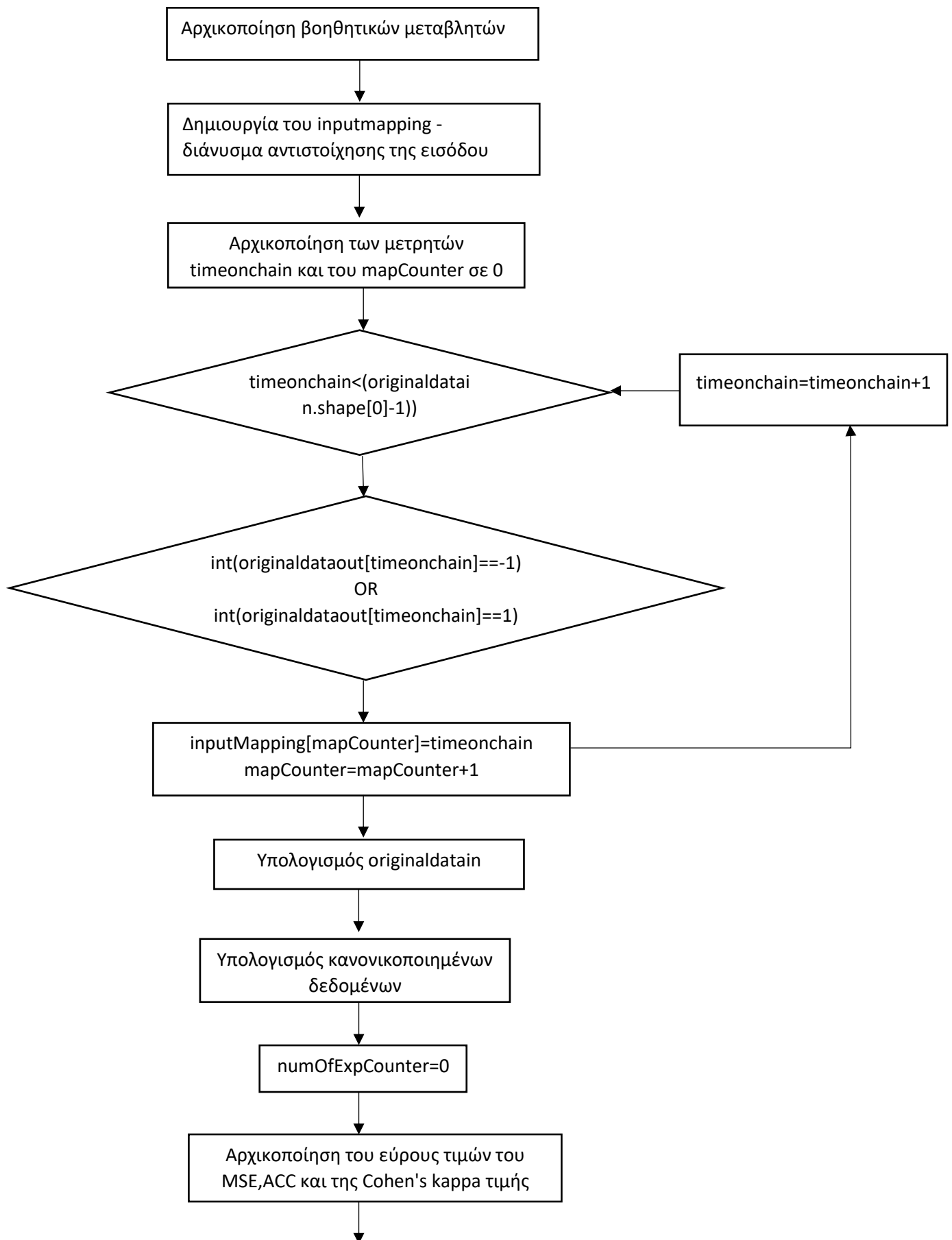
διαφορετικά θα χρησιμοποιηθεί ο τύπος

$$W = W / \text{spectral radius}.$$

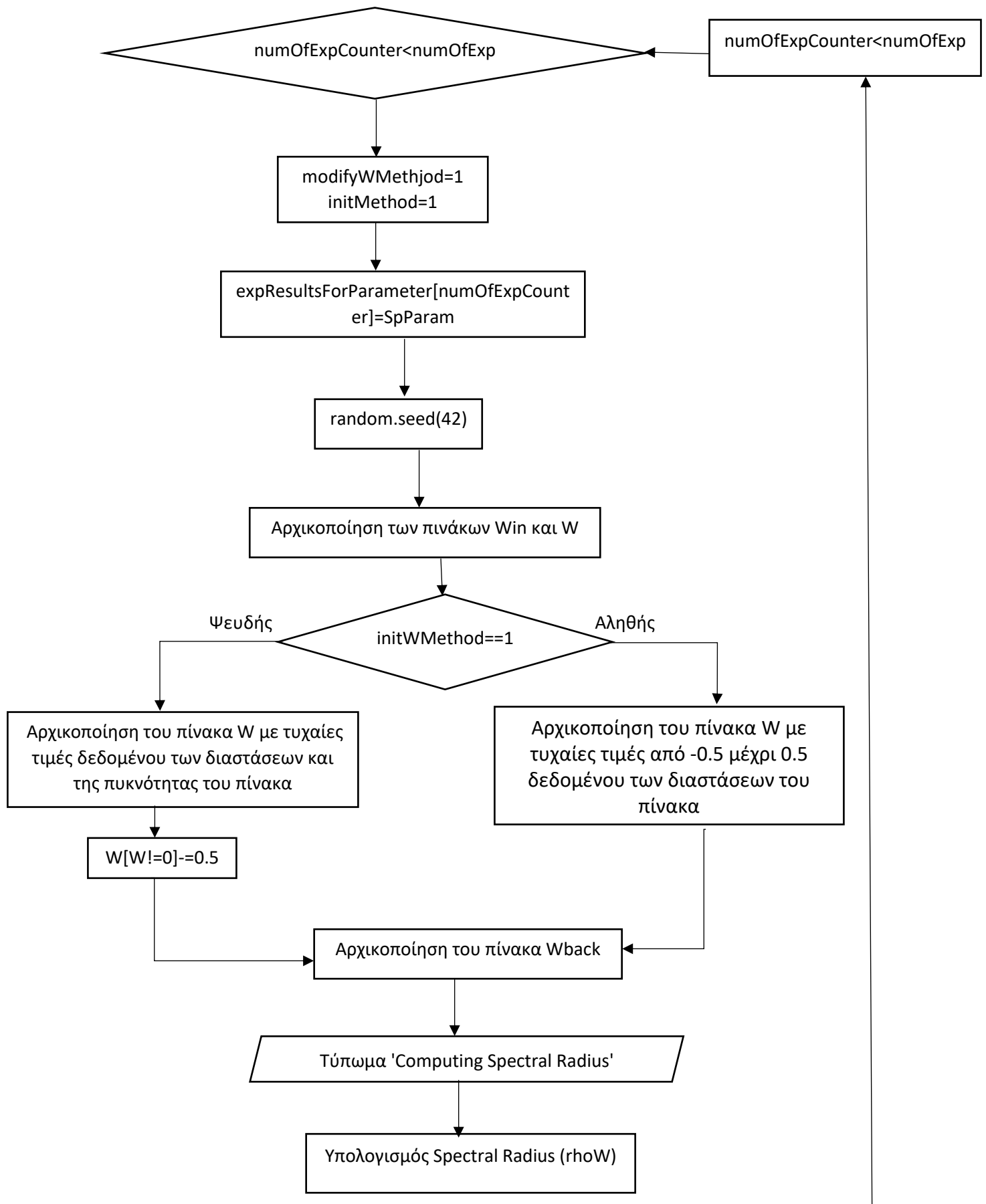
### **3.2 Διάγραμμα Ροής του προγράμματος**

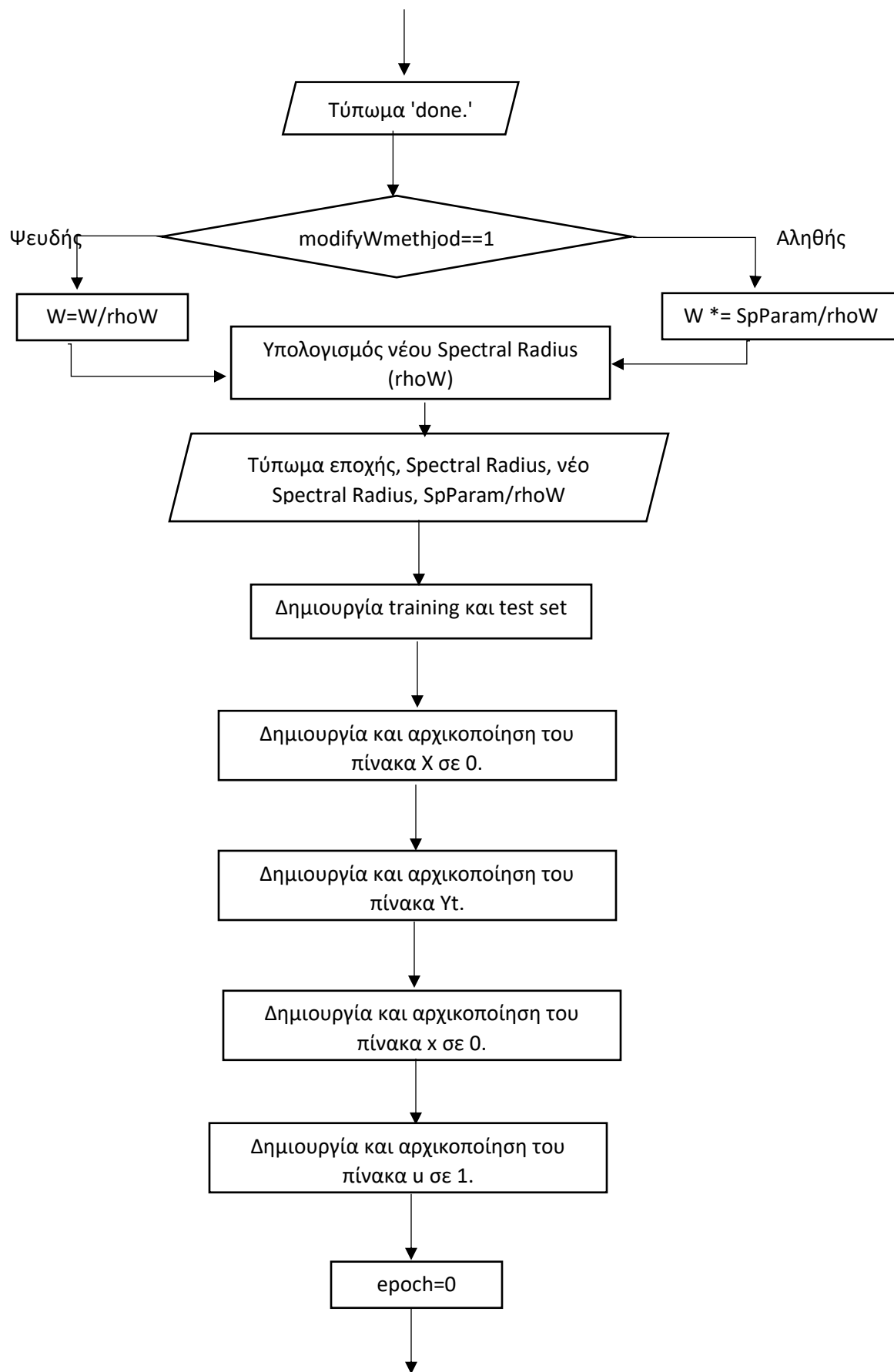
Πιο κάτω επεξηγώ τον κώδικα με διάγραμμα ροής.

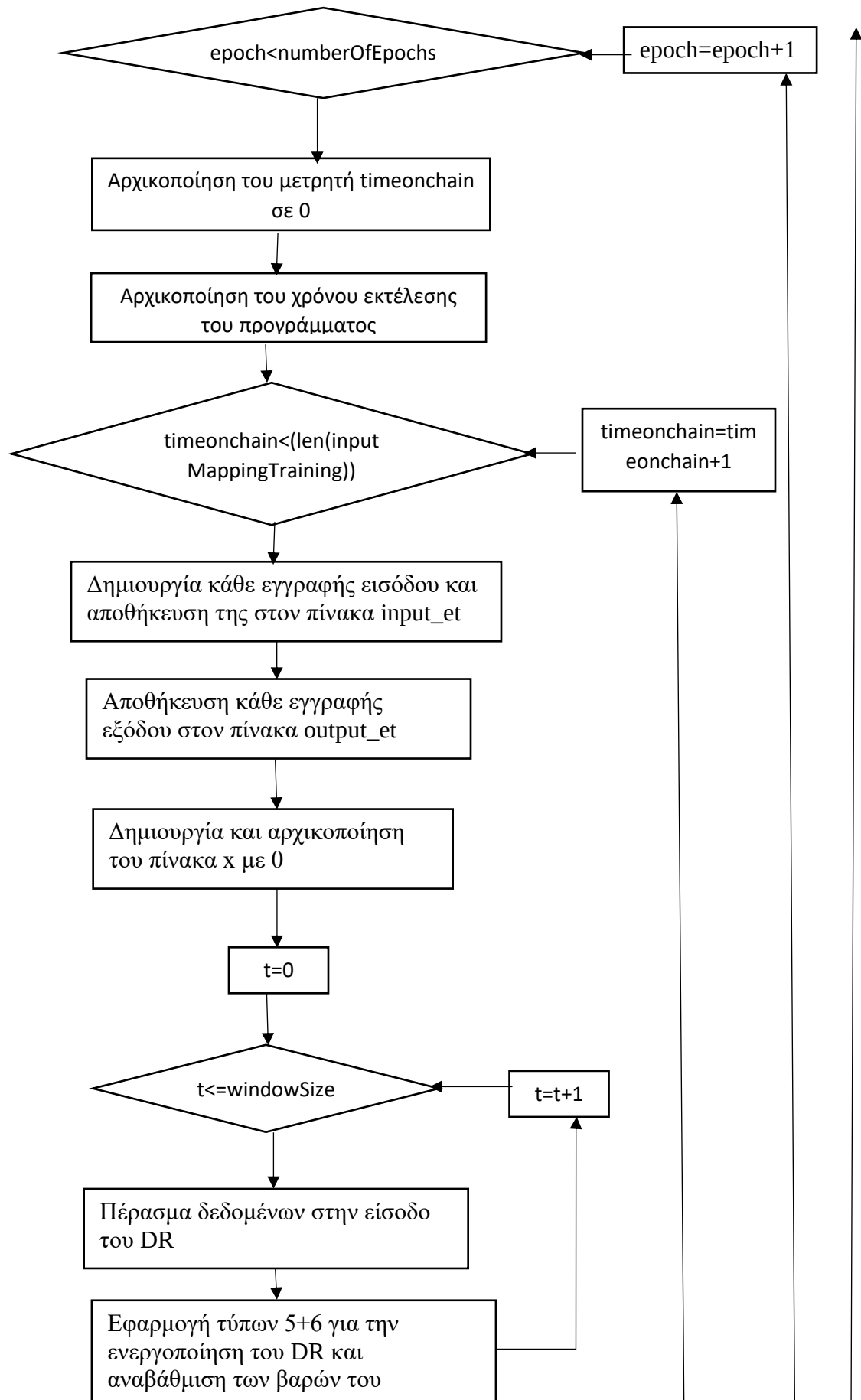


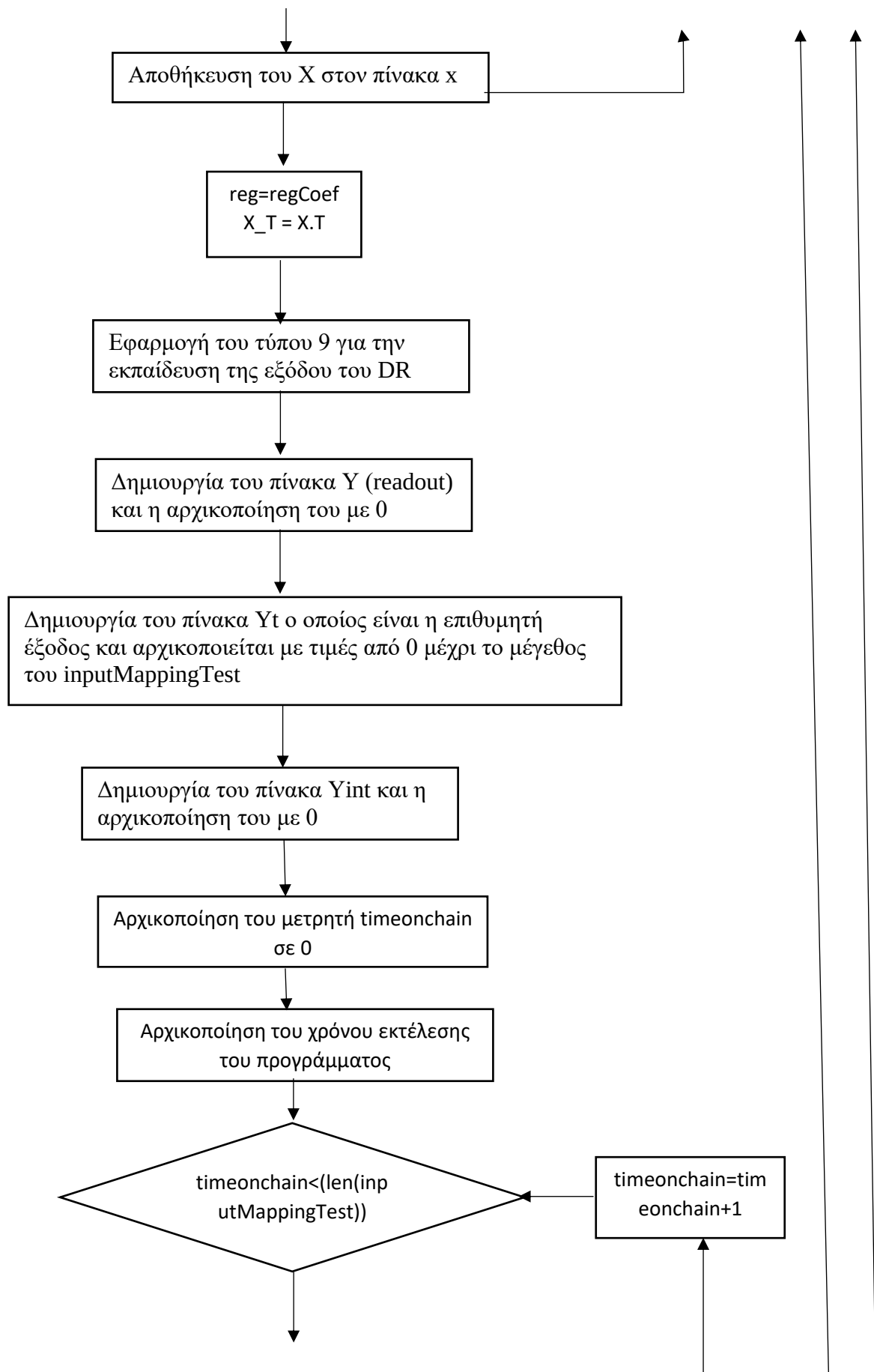


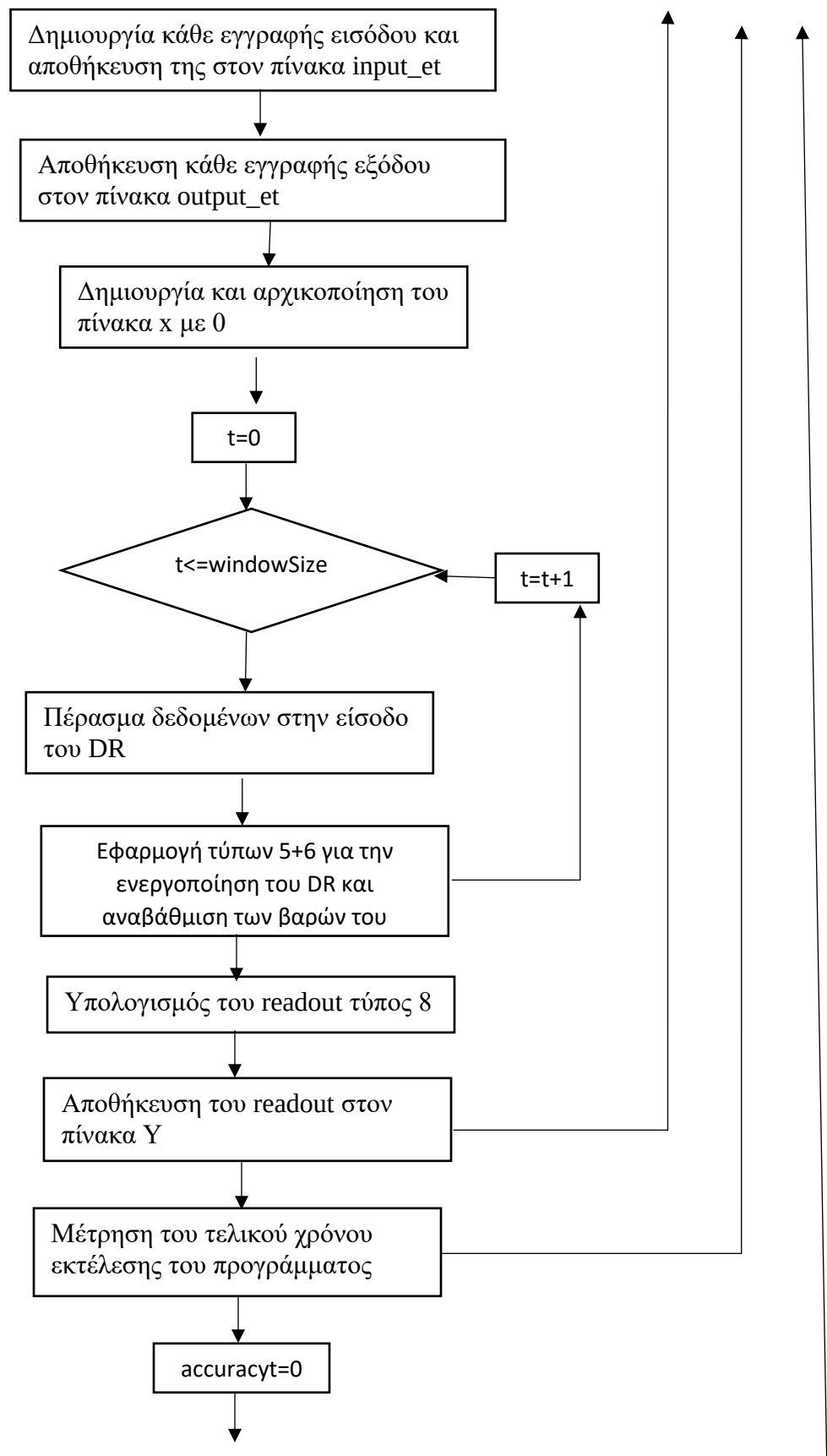


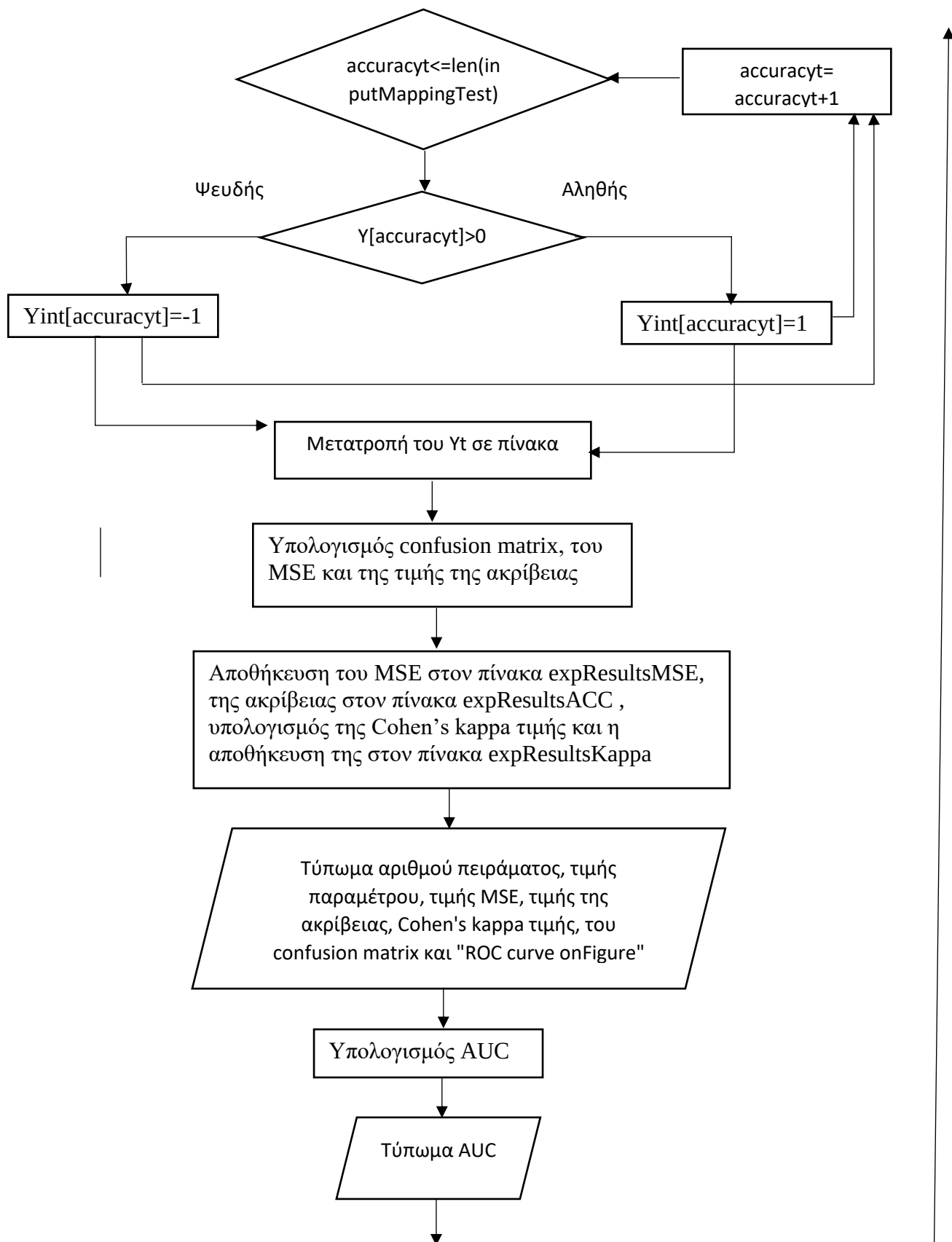


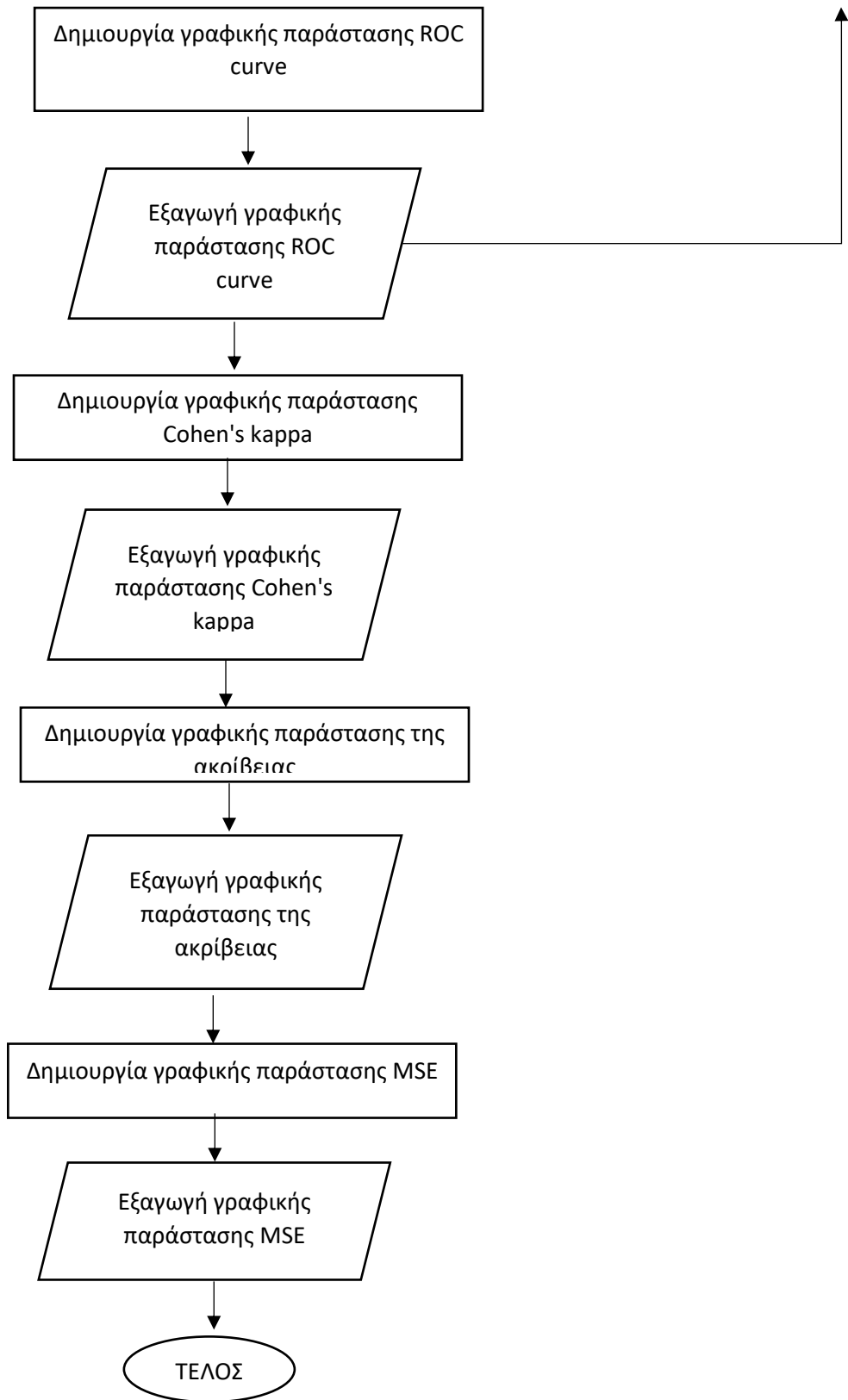












## Κεφάλαιο 4

### Αποτελέσματα και Συζήτηση

---

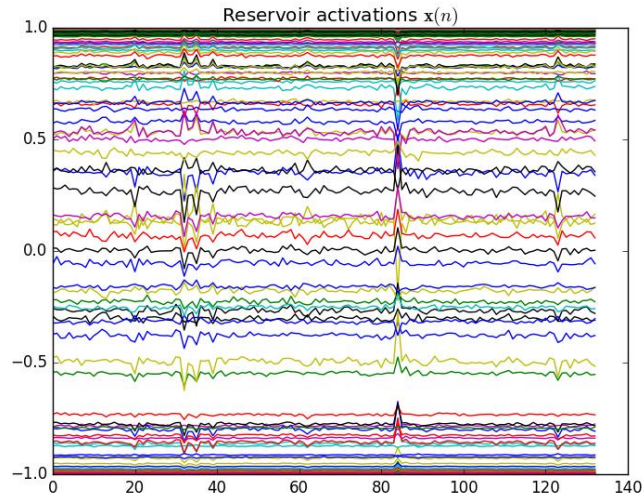
|      |                                        |                    |
|------|----------------------------------------|--------------------|
| 4.1  | Ενεργοποίηση του DR                    | <a href="#">48</a> |
| 4.2  | Έλεγχος του δικτύου                    | <a href="#">51</a> |
| 4.3  | Spectral Radius                        | <a href="#">51</a> |
| 4.4  | Αραιότητα του DR                       | <a href="#">51</a> |
| 4.5  | Χρόνος εκτέλεσης του προγράμματος      | <a href="#">53</a> |
| 4.6  | Ρυθμός διαρροής ( $\alpha$ )           | <a href="#">54</a> |
| 4.7  | Regularization Coefficient ( $\beta$ ) | <a href="#">56</a> |
| 4.8  | Μέγεθος παραθύρου                      | <a href="#">59</a> |
| 4.9  | Μέγεθος DR                             | <a href="#">63</a> |
| 4.10 | Τελικές παραμέτροι                     | <a href="#">65</a> |
| 4.11 | Σύγκριση αποτελεσμάτων                 | <a href="#">69</a> |

---

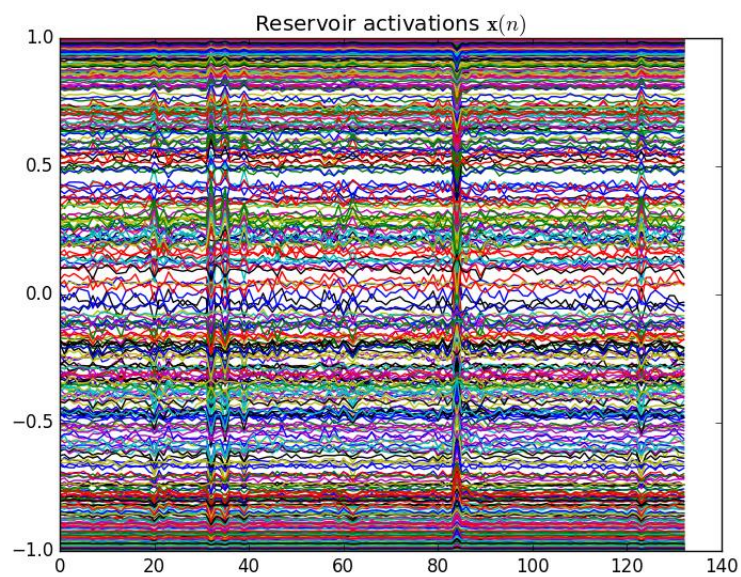


#### 4.1 Ενεργοποίηση του DR

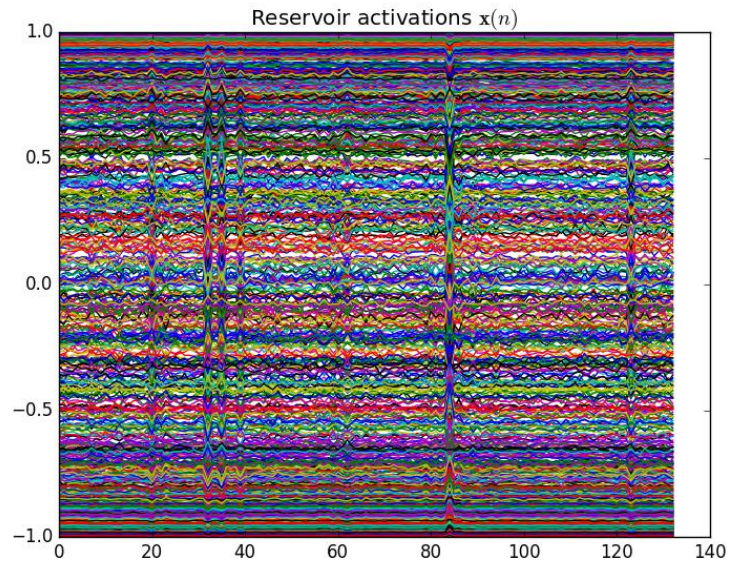
Πιο κάτω παρουσιάζονται οι γραφικές παραστάσεις του διανύσματος  $x(n)$  των ενεργοποιήσεων του DR. Στον άξονα Y παρουσιάζονται οι τιμές του  $x$ , ενώ στον άξονα X παρουσιάζεται η τιμή των γραμμών του πίνακα  $X$  η οποία είναι 133. Παρατηρείται ότι κάθε γραμμή είναι και ένας νευρώνας στο DR γι' αυτό και καθένας απεικονίζεται με διαφορετικό χρώμα.



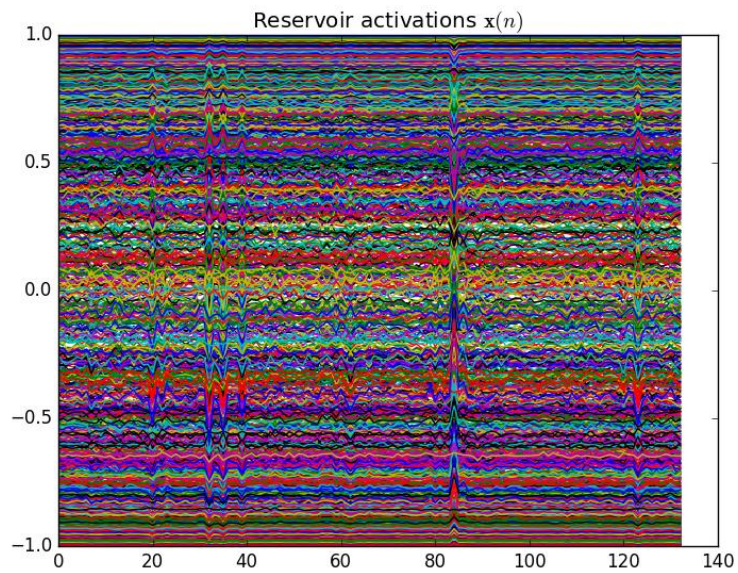
Σχήμα 4.1: Γραφική Παράσταση της ενεργοποίησης του DR με μέγεθος 100



Σχήμα 4.2: Γραφική Παράσταση της ενεργοποίησης του DR με μέγεθος 500

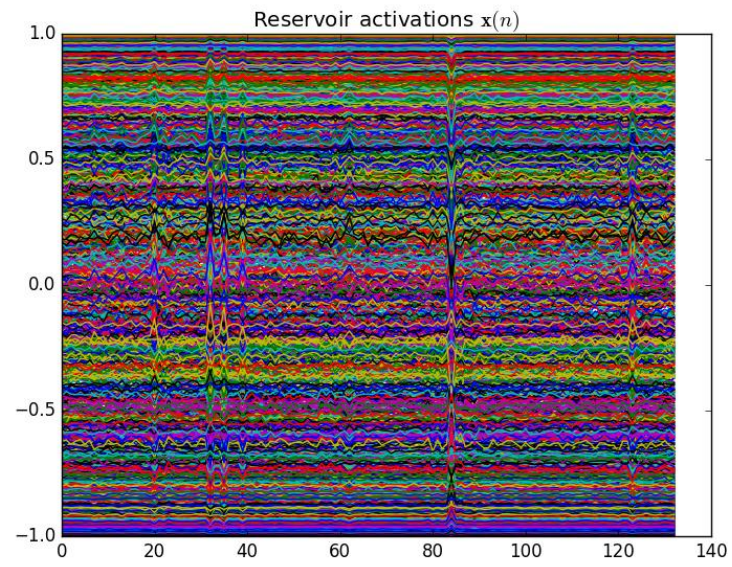


Σχήμα 4.3: Γραφική Παράσταση της ενεργοποίησης του DR με μέγεθος 1000

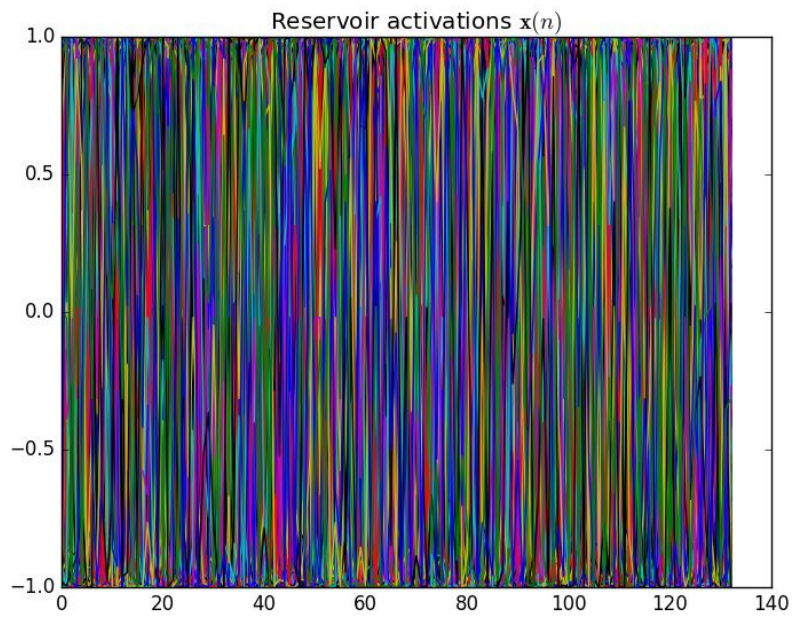


Σχήμα 4.4: Γραφική Παράσταση της ενεργοποίησης του DR με μέγεθος 2000



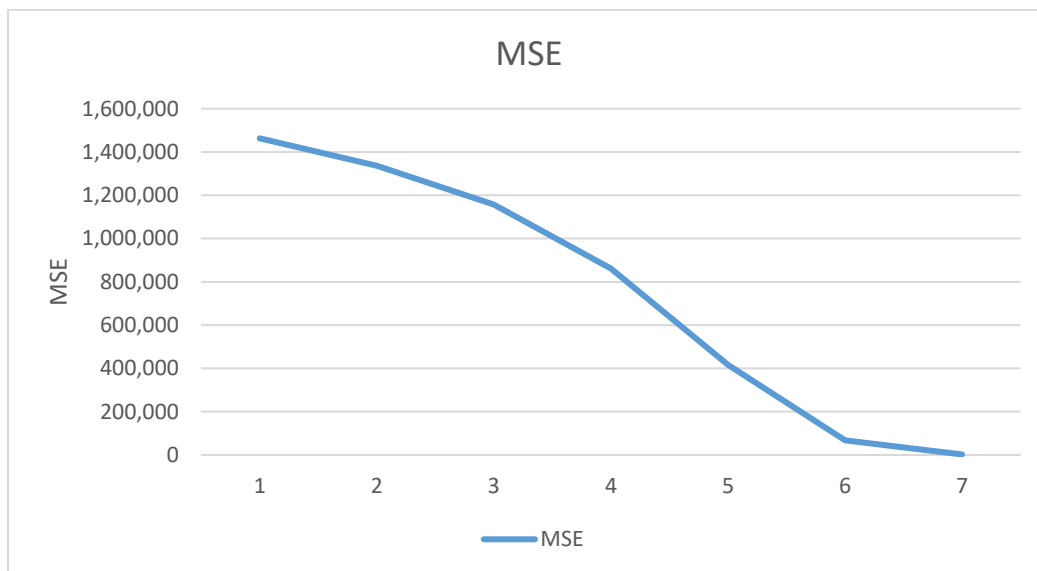


Σχήμα 4.5: Γραφική Παράσταση της ενεργοποίησης του DR με μέγεθος 3000



Σχήμα 4.6: Γραφική Παράσταση της ενεργοποίησης του DR με μέγεθος 5000

## 4.2 Έλεγχος του δικτύου



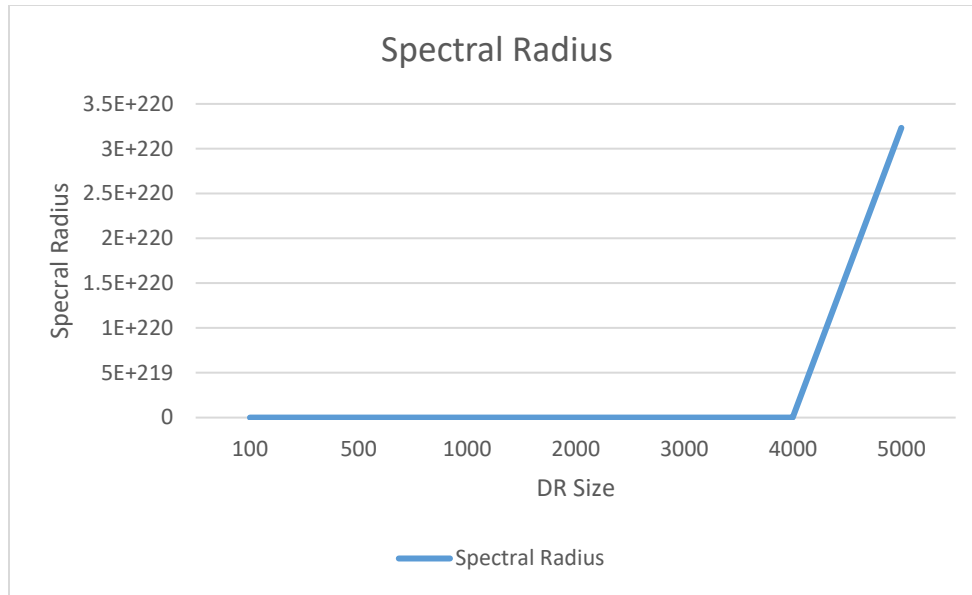
Σχήμα 4.7: Γραφική Παράσταση του MSE κατά τη φάση ελέγχου

Στην πιο πάνω γραφική παράσταση στον άξονα Y παρουσιάζονται οι τιμές του MSE κατά τη φάση ελέγχου. Παρατηρείται ότι το MSE μειώνεται κάθε φορά άρα το δίκτυο μας εκπαιδεύτηκε σωστά.

## 4.3 Spectral Radius

| DR Size | Spectral Radius   |
|---------|-------------------|
| 100     | 3.099             |
| 500     | 6.790             |
| 1000    | 9.286             |
| 2000    | 13.155            |
| 3000    | 16.173            |
| 4000    | 18.556            |
| 5000    | 3.23265535437e209 |

Πίνακας 4.1: Για κάθε μέγεθος του DR παρουσιάζεται η αντίστοιχη τιμή του spectral radius



Σχήμα 4.8: Η αντίστοιχη γραφική παράσταση του πίνακα 4.1

Στο σχήμα 4.8 στον άξονα Y παρουσιάζονται οι τιμές του spectral radius, ενώ στον άξονα X το μέγεθος του DR. Παρατηρείται ότι για μέγεθος 5000 το spectral radius έχει πολύ μεγάλη τιμή. Αυτό είναι αναμενόμενο αφού όπως προαναφέραμε από τον M. Lukosevicius (2012) [11], το spectral radius πρέπει να είναι μεγαλύτερο σε εργασίες που απαιτούν περισσότερη μνήμη από της εισόδου.

#### 4.4 Αραιότητα του DR

| Sparsity      | 0.5   | 1     | 2      | 100     | 1000  | 10000  |
|---------------|-------|-------|--------|---------|-------|--------|
| MSE           | 3.756 | 5.176 | 28.847 | 1.584   | 1.4   | 1.668  |
| Accuracy      | 0.537 | 0.582 | 0.582  | 0.388   | 0.522 | 0.447  |
| Cohen's kappa | 0.074 | 0.165 | 0.162  | -0.0225 | 0.045 | -0.106 |
| AUC           | 0.547 | 0.561 | 0.542  | 0.338   | 0.512 | 0.418  |

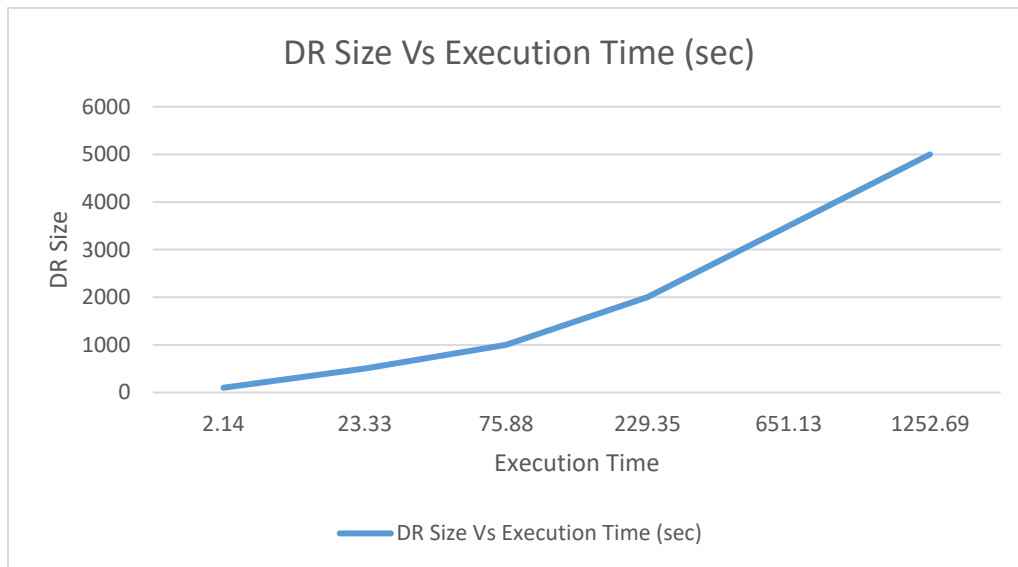
Πίνακας 4.2: Για μέγεθος του DR 1000 και για κάθε μέγεθος του sparsity παρουσιάζονται οι αντίστοιχες τιμές των MSE, Accuracy, Cohen's kappa τιμές και AUC.

Όπως προαναφέραμε στην ενότητα 2.3.1 είναι προτεινόμενο από τον M. Lukosevicius (2012) [11], το DR να είναι αραιό αφού οι αραιές συνδέσεις τείνουν να δίνουν μια ελαφρώς καλύτερη απόδοση. Γι' αυτό και επιλέξαμε sparsity 0.5 όπως φαίνεται και στον πίνακα 4.2. Αλλά, σε γενικές γραμμές αυτή η παράμετρος δεν επηρεάζει και πολύ την απόδοση.

#### 4.5 Χρόνος εκτέλεσης του προγράμματος

| DR Size | Execution Time (sec) |
|---------|----------------------|
| 100     | 2.14                 |
| 500     | 23.33                |
| 1000    | 75.88                |
| 2000    | 229.35               |
| 3500    | 651.13               |
| 5000    | 1252.69              |

Πίνακας 4.3: Για κάθε μέγεθος του DR παρουσιάζεται ο αντίστοιχος χρόνος εκτέλεσης του προγράμματος



Σχήμα 4.9: Η αντίστοιχη γραφική παράσταση του πίνακα 4.3

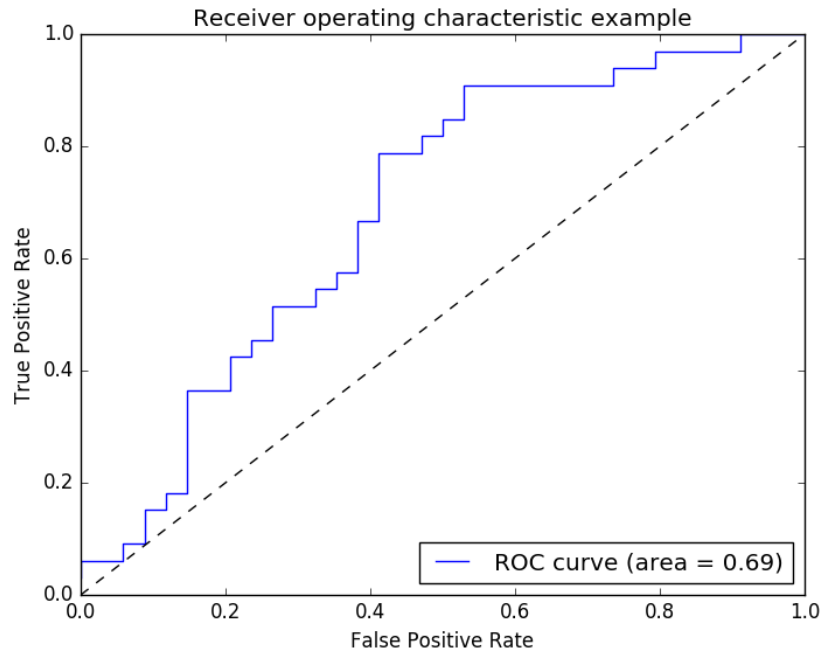
Στην πιο πάνω γραφική παράσταση αναπαριστάται ο χρόνος εκτέλεσης του προγράμματος για διαφορετικά μεγέθη του DR. Ο άξονας Y αντιπροσωπεύει το μέγεθος του DR και ο άξονας X το χρόνο εκτέλεσης. Παρατηρείται ότι η γραφική παράσταση έχει εκθετική μορφή αφού για κάθε μέγεθος του DR ο χρόνος αυξάνεται.

#### 4.6 Ρυθμός διαρροής (a)

| a             | 0.1   | 0.2   | 0.3   | 0.5   | 0.8   | 1     |
|---------------|-------|-------|-------|-------|-------|-------|
| MSE           | 0.992 | 0.985 | 0.987 | 0.986 | 0.994 | 0.987 |
| Accuracy      | 0.552 | 0.671 | 0.552 | 0.537 | 0.582 | 0.582 |
| Cohen's kappa | 0.112 | 0.347 | 0.110 | 0.077 | 0.169 | 0.171 |
| AUC           | 0.523 | 0.686 | 0.636 | 0.581 | 0.566 | 0.617 |

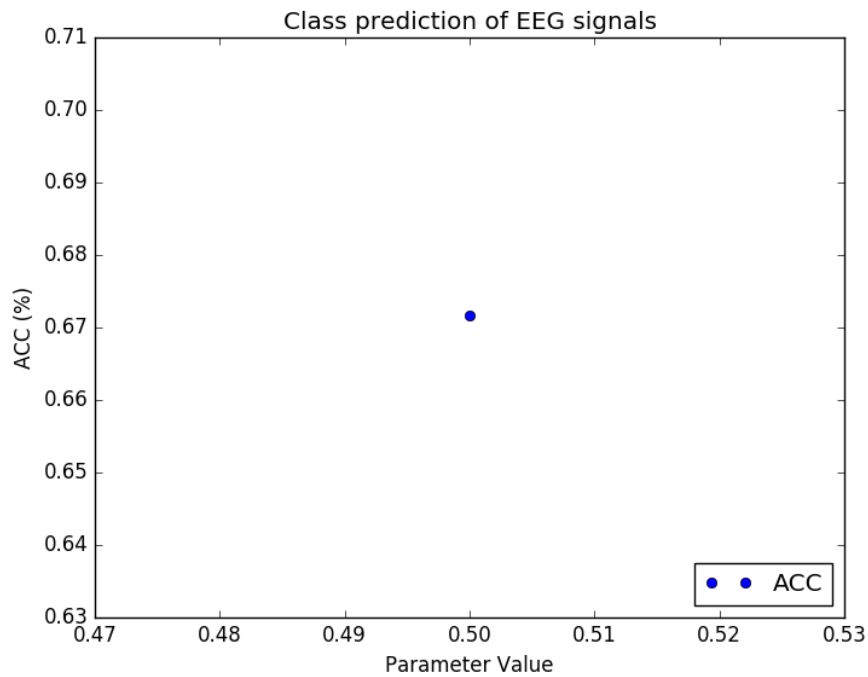
Πίνακας 4.4: Αποτελέσματα για μέγεθος DR 100.

Από τον πίνακα 4.4 παρατηρείται ότι για τιμή του  $\alpha$  ίση με 0.2 έχουμε μικρότερο MSE, μεγαλύτερη ακρίβεια, μεγαλύτερη Cohen's kappa τιμή και μεγαλύτερη AUC τιμή. Συνεπώς, καλύτερα αποτελέσματα έχουμε για τιμή του  $\alpha$  ίση με 0.2.



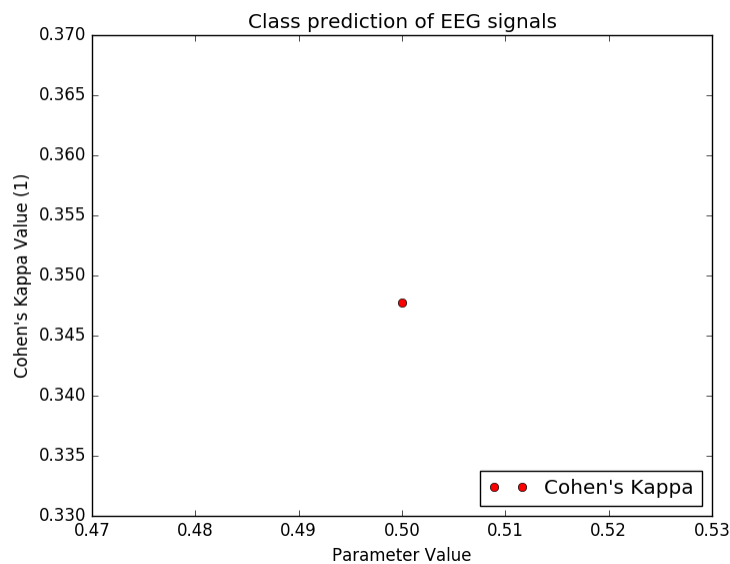
Σχήμα 4.10: Γραφική παράσταση AUC για μέγεθος του DR 100 και  $\alpha$  ίσο με 0.2.

Στην πιο πάνω γραφική παράσταση απεικονίζεται η παράσταση ROC curve με τιμή AUC ίση με 0.69. Η τιμή αυτή όπως προαναφέραμε αντιπροσωπεύει την απόδοση του κατηγοριοποιητή. Όπως προαναφέραμε ο άξονας Y αντιπροσωπεύει το TPR και ο άξονας X το FPR.



Σχήμα 4.11: Γραφική παράσταση της ακρίβειας για μέγεθος του DR 100 και  $\alpha$  ίσο με 0.2.

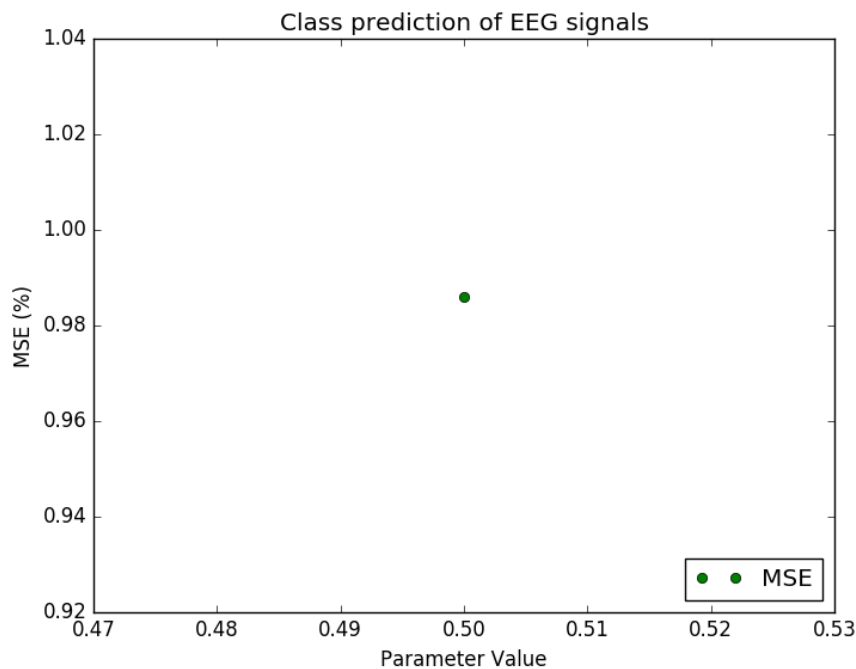
Στην πιο πάνω γραφική παράσταση απεικονίζεται η τιμή της ακρίβειας του δικτύου. Ο άξονας Υ αντιπροσωπεύει το ποσοστό ακρίβειας του δικτύου, ενώ ο άξονας Χ αντιπροσωπεύει την παράμετρο SrParam δηλ. το πόσο αραιό είναι το DR. Στη συγκεκριμένη περίπτωση η ακρίβεια του δικτύου έχει τιμή 0.671 το οποίο είναι ένα καλό ποσοστό.



Σχήμα 4.12: Γραφική παράσταση της Cohen's kappa τιμής για μέγεθος του DR 100 και  $\alpha$  ίσο με 0.2.



Στην πιο πάνω γραφική παράσταση απεικονίζεται η Cohen's kappa τιμή του δικτύου. Ο άξονας Y αντιπροσωπεύει την Cohen's kappa τιμή του δικτύου, ενώ ο άξονας X αντιπροσωπεύει την παράμετρο SpParam δηλ. το πόσο αραιό είναι το DR. Στη συγκεκριμένη περίπτωση η Cohen's kappa τιμή του δικτύου έχει τιμή 0.347 δηλ. το δίκτυο έχει μικρή συμφωνία μεταξύ των δεδομένων και των επιθυμητών δεδομένων.



Σχήμα 4.13: Γραφική παράσταση της MSE τιμής για μέγεθος του DR 100 και  $\alpha$  ίσο με 0.2.

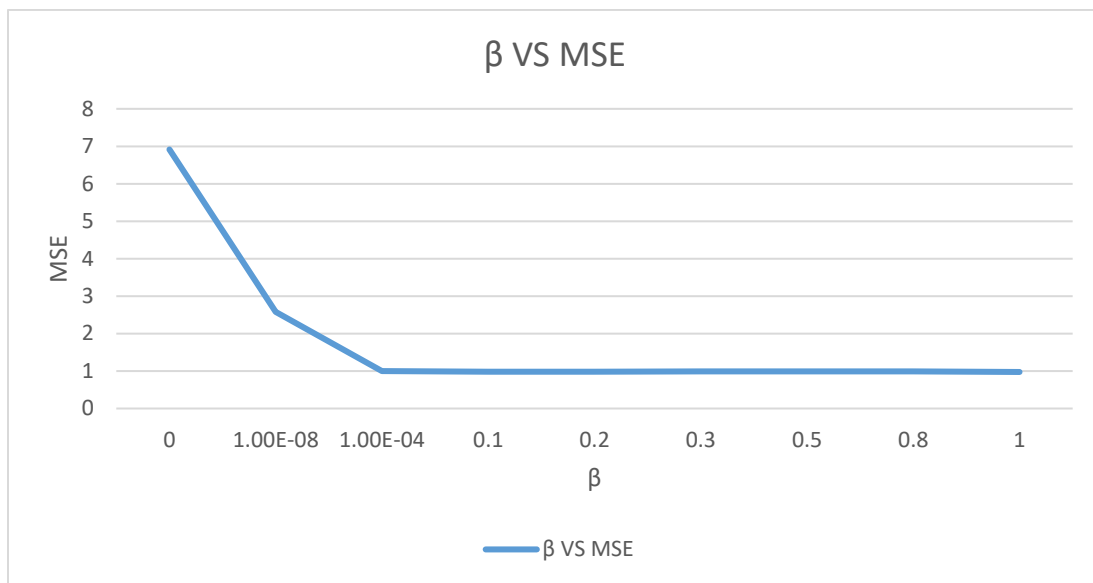
Στην πιο πάνω γραφική παράσταση απεικονίζεται η τιμή του MSE του δικτύου. Ο άξονας Y αντιπροσωπεύει την τιμή του MSE του δικτύου, ενώ ο άξονας X αντιπροσωπεύει την παράμετρο SpParam δηλ. το πόσο αραιό είναι το DR. Στη συγκεκριμένη περίπτωση η τιμή του MSE είναι 0.985 δηλ. το σφάλμα είναι πολύ μικρό κι έτσι έχουμε καλή αντιστοίχιση στα δεδομένα.

#### 4.7 Regularization Coefficient ( $\beta$ )

| $\beta$       | 0     | 1e-8  | 1e-4  | 0.1   | 0.2   | 0.3   | 0.5   | 0.8   | 1     |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| MSE           | 6.915 | 2.577 | 1     | 0.981 | 0.985 | 0.991 | 0.995 | 0.995 | 0.975 |
| Accuracy      | 0.507 | 0.522 | 0.492 | 0.537 | 0.626 | 0.597 | 0.522 | 0.552 | 0.582 |
| Cohen's kappa | 0.012 | 0.043 | 0     | 0.078 | 0.255 | 0.199 | 0.052 | 0.114 | 0.168 |
| AUC           | 0.56  | 0.532 | 0.461 | 0.634 | 0.686 | 0.598 | 0.549 | 0.536 | 0.549 |

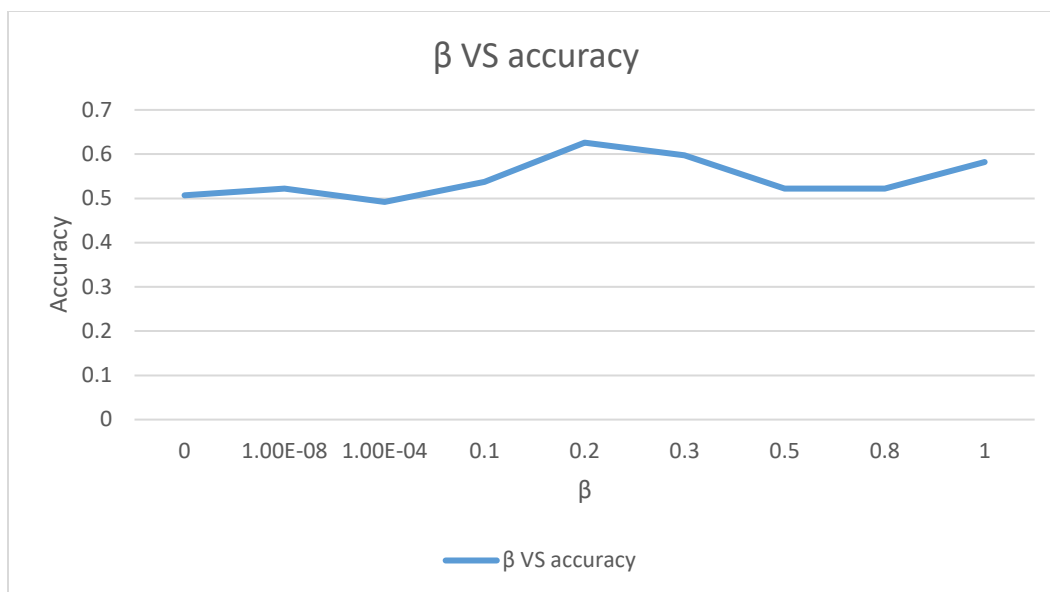
Πίνακας 4.5: Για κάθε τιμή regularization coefficient παρουσιάζονται οι αντίστοιχες τιμές των MSE, Accuracy, Cohen's kappa τιμής και AUC με μέγεθος του DR 100.

Στον πιο πάνω πίνακα παρατηρείται ότι για μέγεθος του regularization coefficient 0.2 η τιμή του MSE είναι η δεύτερη μικρότερη στον πίνακα ενώ η τιμή της ακρίβειας και η Cohen's kappa τιμή και η AUC τιμή είναι οι μεγαλύτερες. Συνεπώς, επιλέγουμε την τιμή 0.2 για την καλύτερη απόδοση του δικτύου.



Σχήμα 4.14: Γραφική παράσταση  $\beta$  με MSE

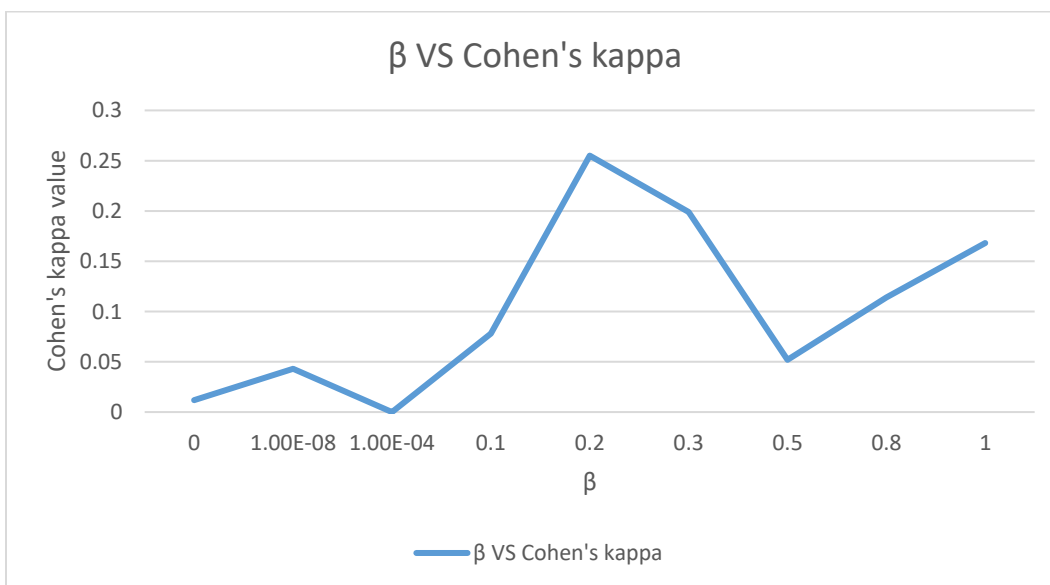
Στην πιο πάνω γραφική παράσταση αναπαριστάται η τιμή του MSE για διαφορετικά  $\beta$ . Ο άξονας Y αντιπροσωπεύει το MSE ενώ ο άξονας X το  $\beta$ . Παρατηρείται ότι η τιμή του MSE για μικρά μεγέθη του  $\beta$  είναι πιο μεγάλη από ότι για τα μεγάλα μεγέθη του  $\beta$ . Συνεπώς, για μεγαλύτερο μέγεθος του  $\beta$  το MSE μειώνεται, άρα δίκτυα με μεγάλο  $\beta$  θα έχουν καλύτερη απόδοση.



Σχήμα 4.15: Γραφική παράσταση  $\beta$  με ακρίβεια

Στην πιο πάνω γραφική παράσταση αναπαριστάται η τιμή της ακρίβειας για διαφορετικά μεγέθη του  $\beta$ . Ο άξονας Y αντιπροσωπεύει την ακρίβεια ενώ ο άξονας X το  $\beta$ .

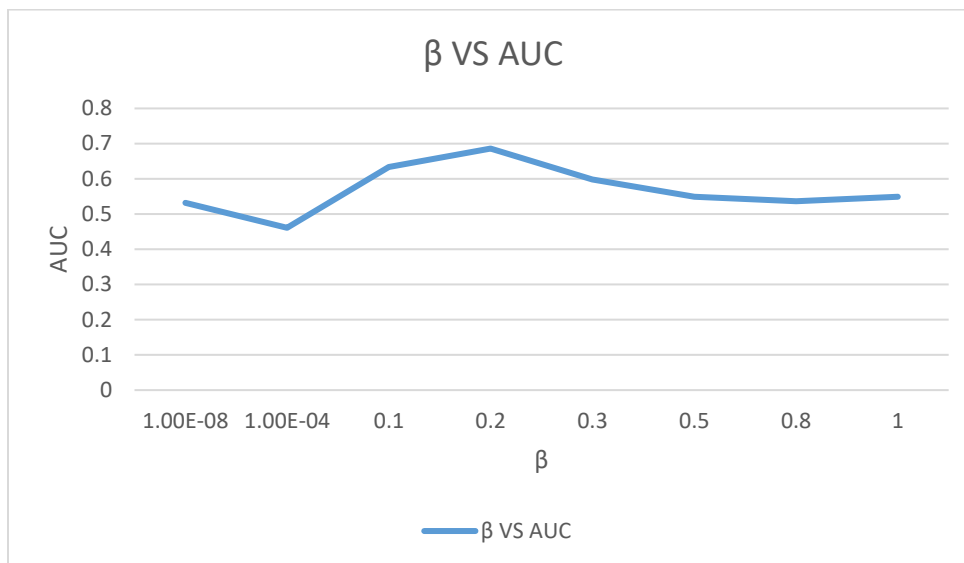
Παρατηρείται ότι για  $\beta$  με τιμή 0.2, η τιμή της ακρίβειας είναι η μεγαλύτερη. Συνεπώς θα επιλέξουμε για  $\beta$  με τιμή 0.2.



Σχήμα 4.16: Γραφική παράσταση  $\beta$  με Cohen's kappa τιμή

Στην πιο πάνω γραφική παράσταση αναπαριστάται η Cohen's kappa τιμή για διαφορετικά μεγέθη του  $\beta$ . Ο άξονας Y αντιπροσωπεύει την Cohen's kappa τιμή ενώ ο άξονας X το  $\beta$ . Παρατηρείται ότι για τις μικρότερες και τις μεγαλύτερες τιμές του  $\beta$  το μέγεθος της συμφωνίας

είναι είτε μηδενικό είτε φτωχό. Για τιμή του  $\beta$  ίση με 0.2 η Cohen's kappa τιμή έχει μικρό μέγεθος συμφωνίας. Άρα, μεγαλύτερη Cohen's kappa τιμή έχει το δίκτυο με  $\beta$  0.2. Συνεπώς, επιλέγουμε μια ενδιάμεση τιμή του  $\beta$ .



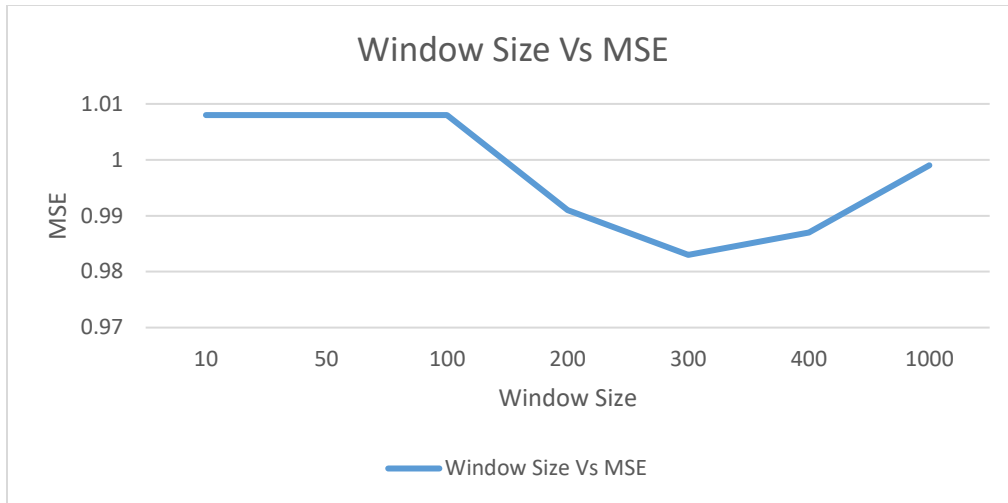
Σχήμα 4.17: Γραφική παράσταση  $\beta$  με AUC

Στην πιο πάνω γραφική παράσταση αναπαριστάται η AUC τιμή για διαφορετικά μεγέθη του  $\beta$ . Ο άξονας Y αντιπροσωπεύει την AUC τιμή ενώ ο άξονας X το  $\beta$ . Παρατηρείται ότι για μικρά μεγέθη του  $\beta$  είναι πιο μικρή παρά για μεγάλα μεγέθη του  $\beta$ . Επίσης, μεγαλύτερη τιμή AUC έχει το δίκτυο με μέγεθος  $\beta$  0.2. Συνεπώς, όσο πιο μεγάλο είναι το  $\beta$  τόσο μεγαλύτερη θα είναι και η τιμή της AUC, άρα δίκτυα με μεγάλη τιμή του  $\beta$  θα έχουν καλύτερη απόδοση.

#### 4.8 Μέγεθος παραθύρου

| Window Size | MSE   | Accuracy | Cohen's kappa | AUC   | Execution Time (sec) |
|-------------|-------|----------|---------------|-------|----------------------|
| 10          | 1.008 | 0.552    | 0.109         | 0.529 | 0.585                |
| 50          | 1.008 | 0.537    | 0.071         | 0.516 | 0.746                |
| 100         | 1.008 | 0.522    | 0.047         | 0.458 | 0.967                |
| 200         | 0.991 | 0.552    | 0.111         | 0.644 | 1.333                |
| 300         | 0.983 | 0.626    | 0.257         | 0.688 | 1.704                |
| 400         | 0.987 | 0.552    | 0.108         | 0.590 | 2.083                |
| 1000        | 0.999 | 0.507    | 0.022         | 0.503 | 7.355                |

Πίνακας 4.6: Για μέγεθος του DR 100, και για κάθε μέγεθος του window size παρουσιάζονται οι αντίστοιχες τιμές των MSE, Accuracy, Cohen's kappa τιμής, AUC και execution time.

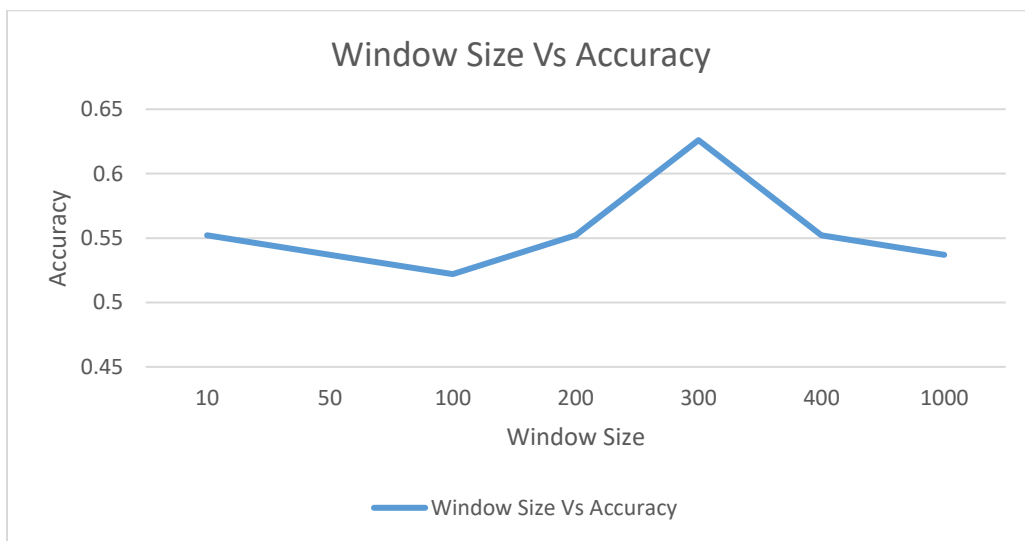


Σχήμα 4.18: Γραφική Παράσταση του Window Size με MSE και μέγεθος του DR 100

Στην πιο πάνω γραφική παράσταση αναπαριστάται η τιμή του MSE για διαφορετικά μεγέθη του παραθύρου. Ο άξονας Y αντιπροσωπεύει το MSE ενώ ο άξονας X το μέγεθος του παραθύρου. Παρατηρείται ότι η τιμή του MSE για μικρά μεγέθη του παραθύρου είναι πιο μεγάλη παρά για μεγάλα μεγέθη του παραθύρου.

Επίσης, παρατηρείται ότι για μέγεθος του παραθύρου με τιμή 300, η τιμή του MSE είναι η πιο μικρή.

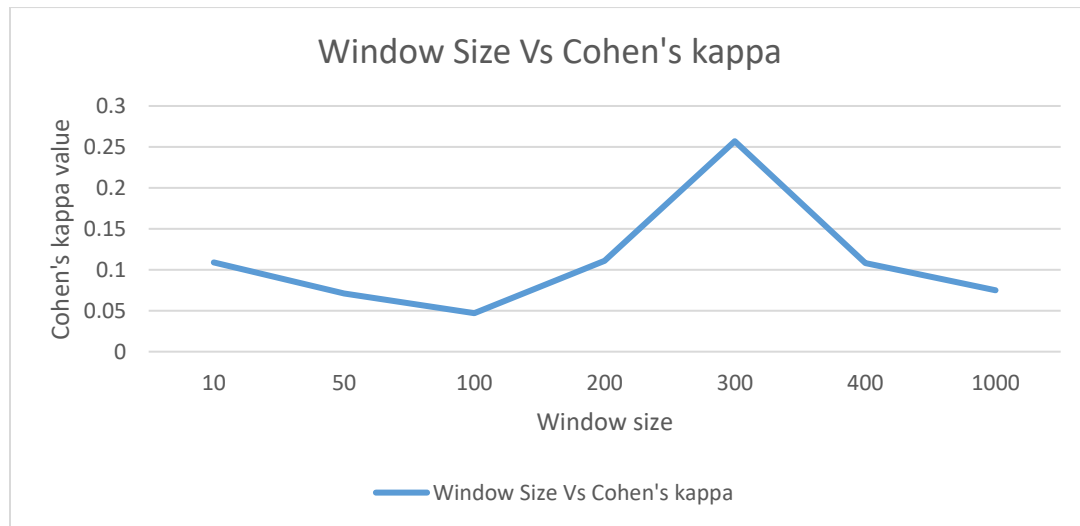
Συνεπώς, για μεγαλύτερο μέγεθος του παραθύρου το MSE μειώνεται, άρα δίκτυα με μεγάλο μέγεθος του παραθύρου θα έχουν καλύτερη απόδοση.



Σχήμα 4.19: Γραφική Παράσταση του Window Size με Accuracy και μέγεθος του DR 100

Στην πιο πάνω γραφική παράσταση αναπαριστάται η τιμή της ακρίβειας για διαφορετικά μεγέθη του παραθύρου. Ο άξονας Y αντιπροσωπεύει την ακρίβεια ενώ ο άξονας X το μέγεθος του παραθύρου. Παρατηρείται ότι η τιμή της ακρίβειας για μικρά μεγέθη του παραθύρου είναι πιο μικρή παρά για μεγάλα μεγέθη του παραθύρου. Επίσης, μεγαλύτερη ακρίβεια έχει το δίκτυο με μέγεθος παραθύρου 300.

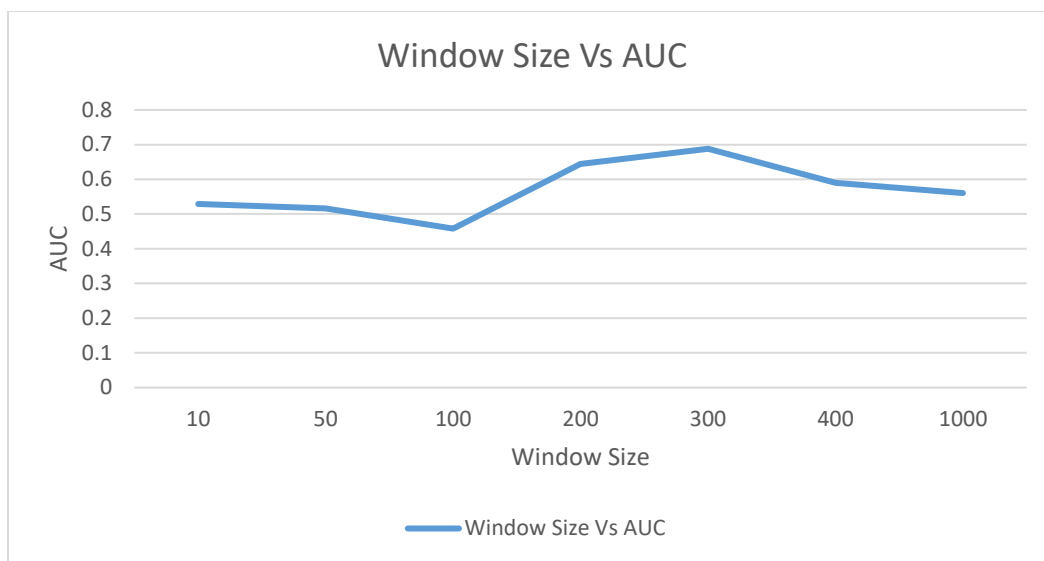
Συνεπώς, όσο πιο μεγάλο είναι το μέγεθος του παραθύρου τόσο μεγαλύτερη θα είναι και η ακρίβεια, άρα δίκτυα με μεγάλο μέγεθος του παραθύρου θα έχουν καλύτερη απόδοση.



Σχήμα 4.20: Γραφική Παράσταση του Window Size με Cohen's kappa τιμής και μέγεθος του DR 100

Στην πιο πάνω γραφική παράσταση αναπαριστάται η Cohen's kappa τιμή για διαφορετικά μεγέθη του παραθύρου. Ο άξονας Y αντιπροσωπεύει την Cohen's kappa τιμή ενώ ο άξονας X το μέγεθος του παραθύρου. Παρατηρείται ότι το μέγεθος της συμφωνίας είναι φτωχό εκτός από το μέγεθος παραθύρου ίσο με 300 το οποίο είναι μικρό. Άρα, μεγαλύτερη Cohen's kappa τιμή έχει το δίκτυο με μέγεθος παραθύρου 300.

Συνεπώς, το μέγεθος του παραθύρου δεν επηρεάζει την Cohen's kappa τιμή.

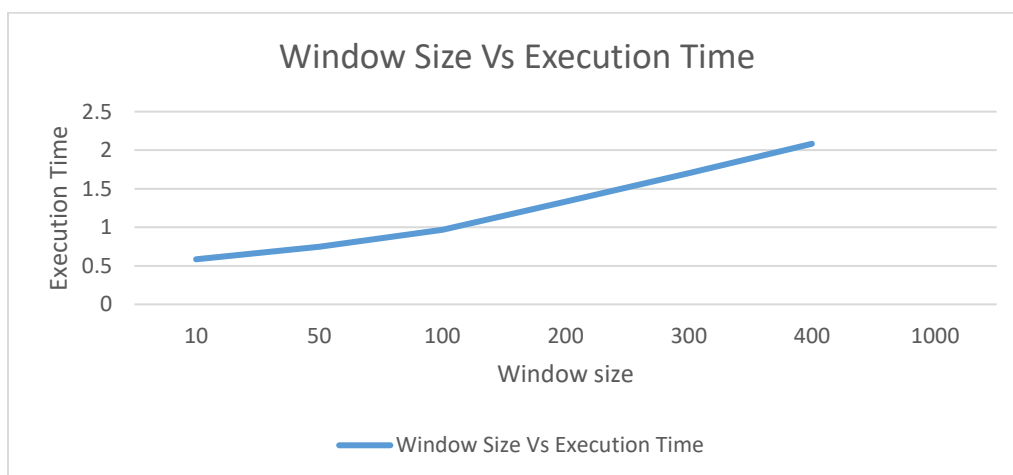


Σχήμα 4.21: Γραφική Παράσταση του Window Size με AUC και μέγεθος του DR 100

Στην πιο πάνω γραφική παράσταση αναπαριστάται η AUC τιμή για διαφορετικά μεγέθη του παραθύρου. Ο άξονας Y αντιπροσωπεύει την AUC ενώ ο άξονας X το μέγεθος του παραθύρου. Παρατηρείται ότι η τιμή της AUC για μικρά μεγέθη του παραθύρου είναι πιο μικρή παρά για μεγάλα μεγέθη του παραθύρου. Επίσης, μεγαλύτερη τιμή AUC έχει το δίκτυο με μέγεθος παραθύρου 300.

Συνεπώς, όσο πιο μεγάλο είναι το μέγεθος του παραθύρου τόσο μεγαλύτερη θα είναι και η τιμή της AUC, άρα δίκτυα με μεγάλο μέγεθος του παραθύρου θα έχουν καλύτερη απόδοση.

Συνεπώς, με βάση τα πιο πάνω καλύτερη απόδοση θα έχει το δίκτυο με μέγεθος παραθύρου 300.



Σχήμα 4.22: Γραφική Παράσταση του Window Size με το χρόνο εκτέλεσης του προγράμματος και μέγεθος του DR 100.

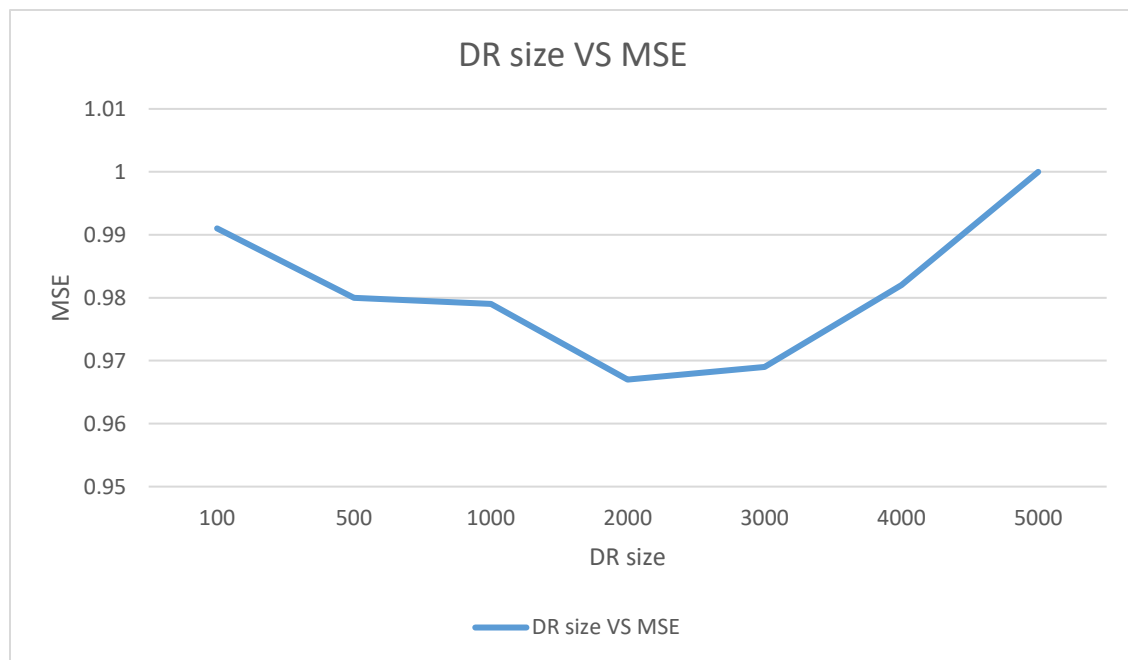
Στην πιο πάνω γραφική παράσταση αναπαριστάται ο χρόνος εκτέλεσης του προγράμματος για διαφορετικά μεγέθη του παραθύρου. Ο άξονας Y αντιπροσωπεύει το χρόνο εκτέλεσης ενώ ο άξονας X το μέγεθος του παραθύρου. Παρατηρείται ότι όσο πιο μικρό είναι το μέγεθος του παραθύρου τόσο πιο μεγάλος ο χρόνος εκτέλεσης του προγράμματος.

#### 4.9 Μέγεθος DR

Μετά από πολλές εκτελέσεις του προγράμματος αλλάζοντας μόνο το μέγεθος του DR είχαμε τα εξής αποτελέσματα:

| DR size              | 100   | 500   | 1000  | 2000  | 3000  | 4000  | 5000  |
|----------------------|-------|-------|-------|-------|-------|-------|-------|
| <b>MSE</b>           | 0.991 | 0.980 | 0.979 | 0.967 | 0.969 | 0.982 | 1     |
| <b>Accuracy</b>      | 0.567 | 0.597 | 0.597 | 0.597 | 0.582 | 0.597 | 0.507 |
| <b>Cohen's kappa</b> | 0.139 | 0.197 | 0.197 | 0.197 | 0.168 | 0.196 | 0     |
| <b>AUC</b>           | 0.629 | 0.6   | 0.599 | 0.593 | 0.604 | 0.581 | 0.5   |

Πίνακας 4.7: Για κάθε μέγεθος του DR παρουσιάζονται οι αντίστοιχες τιμές των MSE, Accuracy, Cohen's kappa τιμή και AUC.

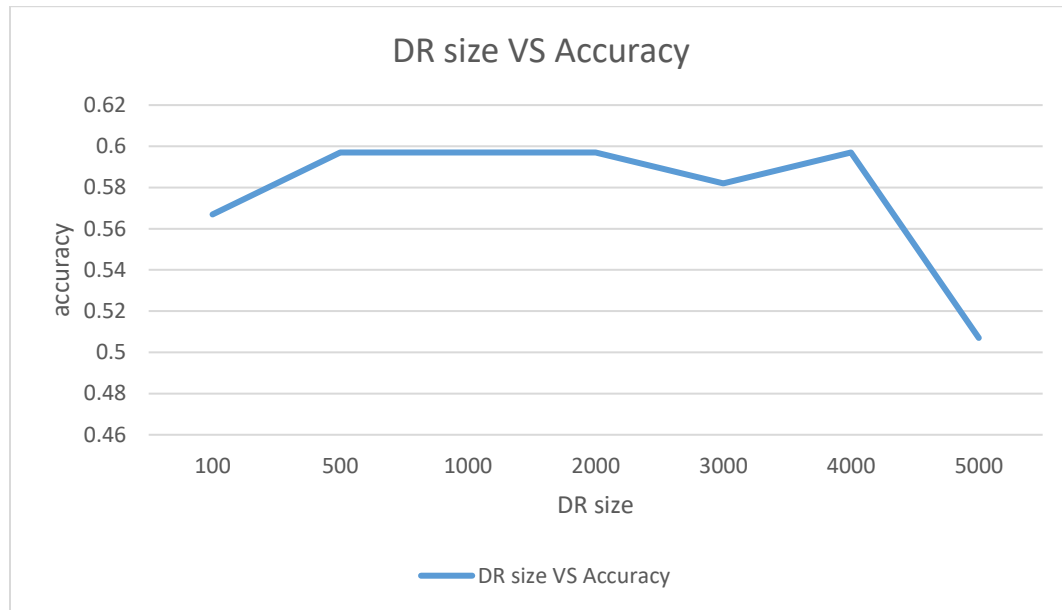


Σχήμα 4.23: Γραφική Παράσταση του DR size με το MSE

Στην πιο πάνω γραφική παράσταση αναπαριστάται η τιμή του MSE για διαφορετικά μεγέθη του DR. Ο άξονας Y αντιπροσωπεύει το MSE ενώ ο άξονας X το μέγεθος του DR.

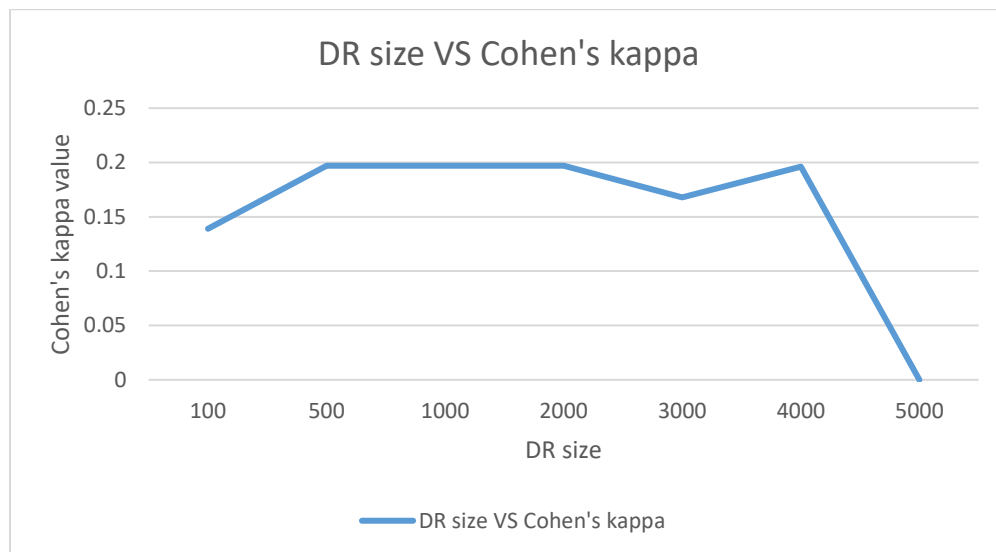


Παρατηρείται ότι μεγαλύτερη τιμή του MSE έχει το δίκτυο με μέγεθος του DR 5000 ενώ μικρότερη τιμή του MSE έχει το δίκτυο με μέγεθος του DR 2000.



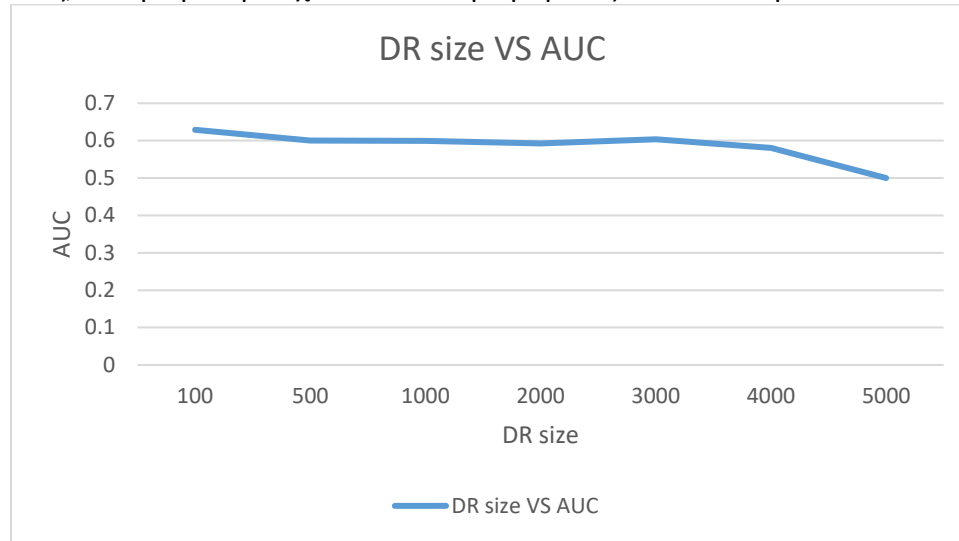
Σχήμα 4.24: Γραφική Παράσταση του DR size με την ακρίβεια (accuracy)

Στην πιο πάνω γραφική παράσταση αναπαριστάται η τιμή της ακρίβειας για διαφορετικά μεγέθη του DR. Ο άξονας Y αντιπροσωπεύει την τιμή της ακρίβειας ενώ ο άξονας X το μέγεθος του DR. Παρατηρείται ότι η ακρίβεια για τα μεγέθη του DR 500, 1000, 2000 και 4000 έχουν την ίδια τιμή ακρίβειας η οποία είναι και η μεγαλύτερη. Μικρότερη ακρίβεια έχει το δίκτυο με μέγεθος του DR 5000.



Σχήμα 4.25: Γραφική Παράσταση του DR size με την Cohen's kappa τιμή

Στην πιο πάνω γραφική παράσταση αναπαριστάται η Cohen's kappa τιμή για διαφορετικά μεγέθη του DR. Ο άξονας Y αντιπροσωπεύει την Cohen's kappa τιμή ενώ ο άξονας X το μέγεθος του DR. Παρατηρείται ότι η Cohen's kappa τιμή έχει μηδενικό μέγεθος συμφωνίας για μέγεθος του DR ίσο με 5000. Τα υπόλοιπα δίκτυα έχουν φτωχό μέγεθος συμφωνίας. Τα δίκτυα με μέγεθος DR 500,1000,2000 έχουν το ίδιο μέγεθος συμφωνίας, ενώ μικρότερο έχει το δίκτυο με μέγεθος DR του ίσο με 5000.



Σχήμα 4.26: Γραφική Παράσταση του DR size με AUC

Στην πιο πάνω γραφική παράσταση αναπαριστάται η AUC τιμή για διαφορετικά μεγέθη του DR. Ο άξονας Y αντιπροσωπεύει την AUC τιμή ενώ ο άξονας X το μέγεθος DR. Παρατηρείται ότι μεγαλύτερη τιμή AUC έχει το δίκτυο με μέγεθος DR 100, ενώ μικρότερη έχει το δίκτυο με μέγεθος του DR 5000.

Συνεπώς, από τα σχήματα 4.23, 4.24, 4.25, 4.26 καλύτερη απόδοση έχει το δίκτυο με μέγεθος του DR 2000.

#### 4.10 Τελικές παραμέτροι

Για την καλύτερη απόδοση του ESN οι τελικές παραμέτροι θα έχουν τις πιο κάτω τιμές:

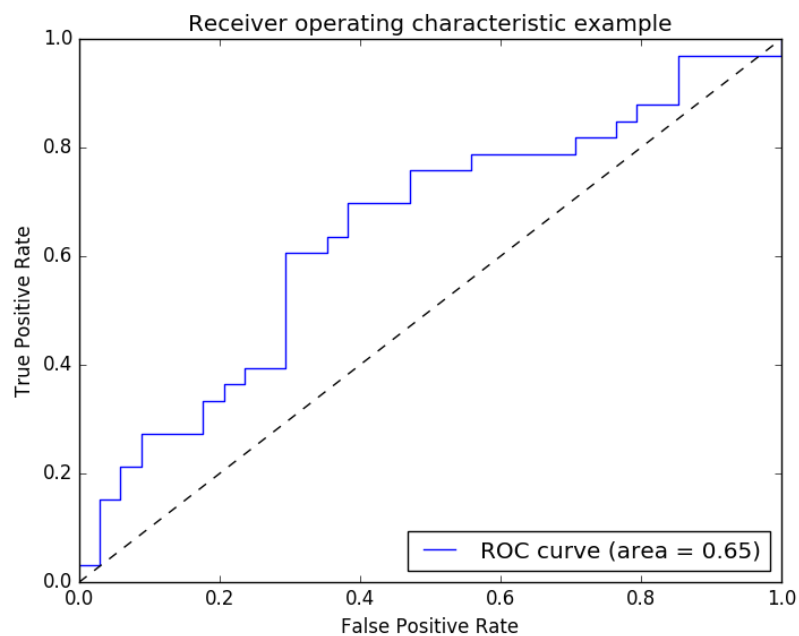
| Name of Parameter      | Value |
|------------------------|-------|
| windowSize             | 300   |
| numberOfChanel         | 59    |
| numberOfEpochs         | 1     |
| numOfInputRecords      | 200   |
| inSize                 | 59    |
| outSize                | 1     |
| resSize                | 1000  |
| a                      | 0.2   |
| SpParam                | 0.5   |
| regCoef                | 0.2   |
| numOfExp               | 1     |
| initWMethod            | 1     |
| modifyWMethjod         | 1     |
| resPercForinitWMethod2 | 0.05  |

*Πίνακας 4.8: Τιμές των τελικών παραμέτρων*

|                      |              |
|----------------------|--------------|
| <b>MSE</b>           | <b>0.968</b> |
| <b>Accuracy</b>      | 0.611        |
| <b>Cohen's kappa</b> | 0.227        |
| <b>AUC</b>           | 0.645        |

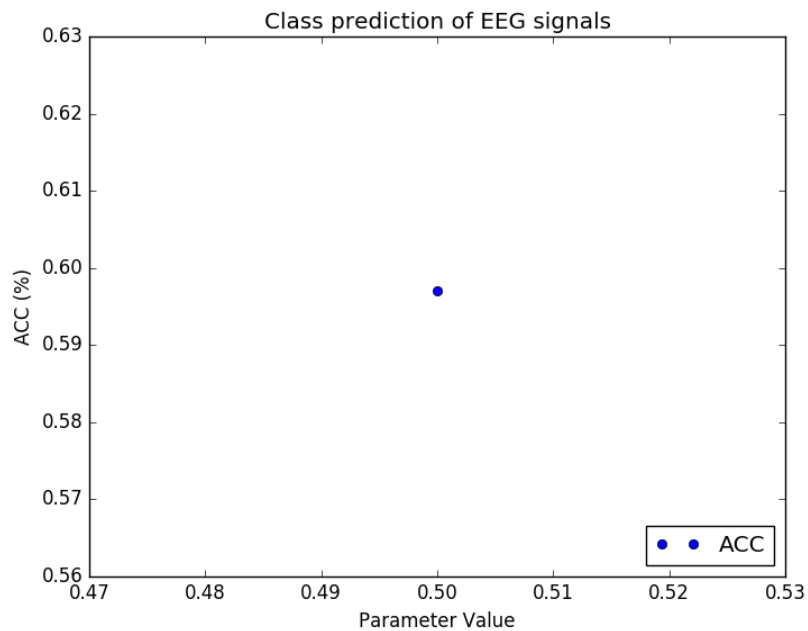
*Πίνακας 4.9: Αποτελέσματα του δικτύου με τις τελικές παραμέτρους*

Πιο κάτω φαίνονται οι γραφικές παραστάσεις που εξαγάγει το πρόγραμμα με βάση τις πιο πάνω παραμέτρους.



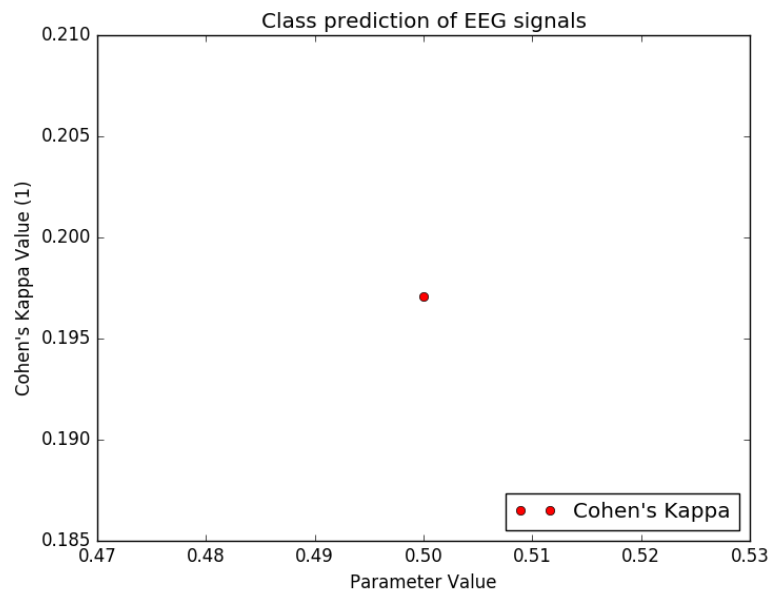
Σχήμα 4.27: Γραφική Παράσταση ROC curve

Στην πιο πάνω γραφική παράσταση απεικονίζεται η παράσταση ROC curve με τιμή AUC ίση με 0.59. Η τιμή αυτή όπως προαναφέραμε αντιπροσωπεύει την απόδοση του κατηγοριοποιητή. Όπως προαναφέραμε ο άξονας Y αντιπροσωπεύει το TPR και ο άξονας X το FPR.



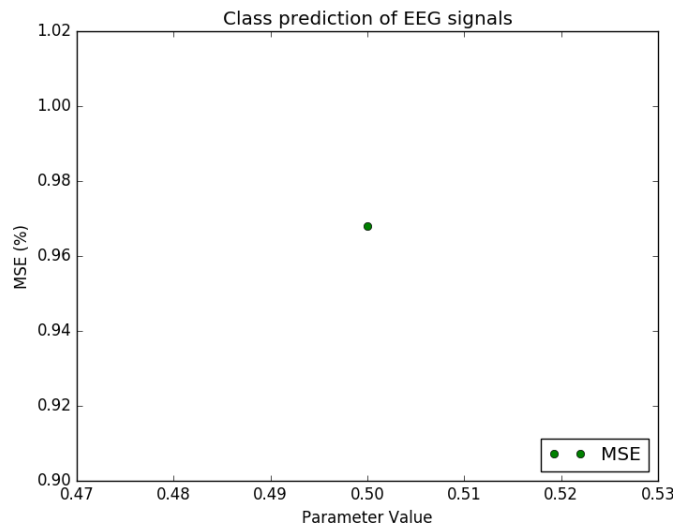
Σχήμα 4.28: Γραφική Παράσταση ακρίβειας (accuracy)

Στην πιο πάνω γραφική παράσταση απεικονίζεται η τιμή της ακρίβειας του δικτύου. Ο άξονας Υ αντιπροσωπεύει το ποσοστό ακρίβειας του δικτύου, ενώ ο άξονας Χ αντιπροσωπεύει την παράμετρο SpParam δηλ. το πόσο αραιό είναι το DR. Στη συγκεκριμένη περίπτωση η ακρίβεια του δικτύου έχει τιμή 0.597 δηλ. το δίκτυο προβλέπει περισσότερα από τα μισά επιθυμητά δεδομένα.



*Σχήμα 4.29: Γραφική Παράσταση Cohen's kappa τιμής*

Στην πιο πάνω γραφική παράσταση απεικονίζεται η Cohen's kappa τιμή του δικτύου. Ο άξονας Υ αντιπροσωπεύει την Cohen's kappa τιμή του δικτύου, ενώ ο άξονας Χ αντιπροσωπεύει την παράμετρο SpParam δηλ. το πόσο αραιό είναι το DR. Στη συγκεκριμένη περίπτωση η Cohen's kappa τιμή του δικτύου έχει τιμή 0.197 δηλ. το δίκτυο έχει φτωχή συμφωνία μεταξύ των δεδομένων και των επιθυμητών δεδομένων.



Σχήμα 4.30: Γραφική Παράσταση MSE

Στην πιο πάνω γραφική παράσταση απεικονίζεται η τιμή του MSE του δικτύου. Ο άξονας Y αντιπροσωπεύει την τιμή του MSE του δικτύου, ενώ ο άξονας X αντιπροσωπεύει την παράμετρο SpParam δηλ. το πόσο αραιό είναι το DR. Στη συγκεκριμένη περίπτωση η τιμή του MSE είναι 0.967 δηλ. το σφάλμα είναι πολύ μικρό κι έτσι έχουμε καλή αντιστοίχιση στα δεδομένα.

#### **4.11 Σύγκριση αποτελεσμάτων**

Για τη σύγκριση των αποτελεσμάτων μου με άλλα αποτελέσματα βασίστηκα πάνω στα αποτελέσματα που εξήλθαν από τον «BCI Competition IV» (<http://www.bbc.de/competition/iv/index.html#dataset1>)[20].

Στόχος του «BCI Competition IV» είναι η επικύρωση επεξεργασίας σημάτων και η κατηγοριοποίηση μεθόδων για BCI. Στην πρώτη θέση ήταν ο Zhang Haihong, από το Institute for Infocomm Research στη Σιγκαπούρη, στον οποίο ο αλγόριθμος αποτελείτο από δύο μέρη. Το πρώτο μέρος ήταν η μηχανική εκμάθησης ενός εύρωστου εκτιμητή της κατάστασης χρήστη από EEG και το δεύτερο μέρος ήταν η επεξεργασία δεδομένων αξιολόγησης με τη χρήση του εκπαιδευμένου εκτιμητή. Στη δεύτερη θέση ήταν ο Dieter Devlaminck, από το Πανεπιστήμιο της Γάνδης όπου η μέθοδος που χρησιμοποίησε αποτελείτο από 4 βήματα το φιλτράρισμα ανά θέμα, η κατηγοριοποίηση μέσω της μεθόδου Common Spatial Patterns και της μεθόδου Support Vector Machines με κανονική παλλινδρόμηση και η εξομάλυνση. Στη δωδέκατη θέση ήταν ο Jing Jin από το Institute for Knowledge Discovery από το Graz University of Technology, όπου η μέθοδος που χρησιμοποίησε αποτελείτο από 4 βήματα 16 ηλεκτρόδια, Short Time Fourier Transform, Power Spectrum Density Features (PSDF) και δύο κατηγοριοποιητές Support Vector Machine. Στην 23η θέση ήταν ο Li Ke, από το Shenyang University of Technology της

Κίνας ο οποίος χρησιμοποίησε τη μέθοδο του μετασχηματισμού κυμάτων και τη μέθοδο Support Vector Machines. Στην τελευταία θέση ήταν ο Yang Banghua από το Shanghai University

της Κίνας. Τα αποτελέσματα φαίνονται με την τιμή του MSE όπως φαίνονται στον πιο κάτω πίνακα:

| Θέση | Όνομα             | MSE   |
|------|-------------------|-------|
| 1    | Zhang Haihong     | 0.382 |
| 2    | Dieter Devlaminck | 0.383 |
| 12   | Jing Jin          | 0.559 |
| 23   | Li Ke             | 1.156 |
| 24   | Yang Banghua      | 1.312 |

*Πίνακας 4.10: Αποτελέσματα του MSE από τον «BCI Competition IV»*

## Κεφάλαιο 6

### Συμπεράσματα

---

Με βάση τα πιο πάνω αποτελέσματα παράγουμε τα πιο κάτω συμπεράσματα.

Αρχικά παρατηρήθηκε ότι κατά τη φάση ελέγχου το MSE μειωνόταν κάθε φορά, άρα το δίκτυο εκπαιδευόταν σωστά. Επίσης, συμπεραίνουμε ότι όσο πιο μεγάλο είναι το μέγεθος του DR τόσο μεγαλύτερη θα είναι και η τιμή του spectral radius. Επιπρόσθετα, συμπεραίνουμε ότι οι αραιές συνδέσεις τείνουν να δίνουν μια ελαφρώς καλύτερη απόδοση και ότι η αραιότητα δεν επηρεάζει και πολύ την απόδοση και έχει χαμηλή προτεραιότητα, γι' αυτό καταλήξαμε στην τιμή του sparsity 0.5. Ακόμη, συμπεραίνουμε ότι όσο πιο μεγάλο είναι το μέγεθος του DR τόσο μεγαλύτερος θα είναι ο χρόνος εκτέλεσης του προγράμματος και μέσω της πειραματικής μεθόδου καταλήξαμε στην τιμή του ρυθμού διαρροής  $\alpha$  ίση με 0.2, αφού το δίκτυο έχει καλύτερη απόδοση. Επίσης, συμπεραίνουμε ότι για μεγάλες τιμές του  $\beta$  το MSE μειώνεται, άρα προτιμάμε μεγάλες τιμές του  $\beta$ . Στη συγκεκριμένη περίπτωση επιλέξαμε τιμή του  $\beta$  ίση με 0.2 αφού η τιμή της ακρίβειας είναι η μεγαλύτερη. Επίσης από την Cohen's kappa τιμή συμπεραίνουμε ότι καλύτερη τιμή του  $\beta$  είναι το 0.2 αφού γι' αυτή την τιμή το μέγεθος συμφωνίας είναι μικρό ενώ για τις υπόλοιπες τιμές το μέγεθος συμφωνίας είναι είτε μηδενικό είτε φτωχό. Επίσης, για μεγάλες τιμές του  $\beta$  η τιμή της AUC είναι μεγαλύτερη. Ακόμη, συμπεραίνουμε ότι η τιμή του MSE για μικρό μέγεθος παραθύρου είναι μεγαλύτερη και ότι όσο πιο μεγάλο είναι το μέγεθος του παραθύρου τόσο πιο μεγάλη είναι η ακρίβεια. Επιπρόσθετα, το μέγεθος του παραθύρου δεν επηρεάζει την Cohen's kappa τιμή και δίκτυα με μεγάλο μέγεθος παραθύρου θα έχουν καλύτερη απόδοση. Παράλληλα, όσο πιο μεγάλο είναι το μέγεθος του παραθύρου τόσο πιο μεγάλος θα είναι ο χρόνος εκτέλεσης του προγράμματος. Τέλος, συμπεραίνουμε ότι καλύτερη απόδοση έχει το δίκτυο με μέγεθος του DR ίσο με 1000.



## Βιβλιογραφία

- [1] M.Agathocleous (2013), Decoding the Neural Code with Machine Learning Techniques: A Brain-Computer Interface Approach, A report submitted for the PhD Qualification Examination.
- [2] S. N. Abdulkater, A. Atia and M. M. Mostafa, Brain Computer Interfacing: Applications and challenges, HCI-LAB, Department of Computer Science, Faculty of Computers and Information, Helwan University, Cairo, Egypt, 2015
- [3] J. Boedecker, O. Obst, J. T. Lizier, N. Michael Mayer and M. Asada, “Information Processing in Echo State Networks at the Edge of Chaos”
- [4] V.Ceperie and A.Baric (2014), “Reducing Complexity of Echo State Networks with Sparse Linear Regression Algorithms”, Faculty of Electrical Engineering and Computing University of Zagreb, Croatia.
- [5] J. Cohen (1960). “A coefficient of agreement for nominal scales”. Educational and Psychological Measurement 20(1):37-46.
- [6] T.Fawcett(2005), “An introduction to ROC analysis”, Institute for the Study of Learning and Expertise, 2164 Staunton Court, Palo Alto, CA 94306, USA
- [7] M. S. A. Graziano, The Intelligent Movement Machine: An Ethological Perspective on the Primate Motor System. New York, NY, USA: Oxford University Press, 2009
- [8] F.Grezes(2014), “Reservoir Computing”, A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York.

- [9] C. Hammond, Cellular and Molecular Neurophysiology, 3rd ed. London, UK: Elsevier Press, 2008
  
- [10] Herbert Jaeger, A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach, Fraunhofer Institute for Autonomous Intelligent Systems (AIS) since 2003: International University Bremen, 5<sup>th</sup> revision 2013
  
- [11] Mantas Lukosevicius (2012), A Practical Guide to Applying Echo State Networks. Jacobs University Bremen, Campus Ring 1, 28759 Bremen, Germany
  
- [12] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," The Bulletin of Mathematical Biophysics, vol. 5, no. 4, pp. 115–133, 1943.
  
- [13] W. Penfield and T. Rasmussen, The cerebral cortex of man: A clinical study of localization of function. New York, NY, USA: Macmillan Press, 1950.
  
- [14] S.Sanei and J.A.Chambers, EEG Signal Processing. Southern Gate, Chichester, UK:John Wiley&Sons, 2008
  
- [15] Jerry J. Shih, Dean J. Krusienski and Jonathan R. Wolpaw, Brain-Computer Interfaces in Medicine, Department of Neurology, Mayo Clinic, Jacksonville, FL, Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA, Laboratory of Neural Injury and Repair, Wadsworth Center, New York State Department of Health and State University of New York, Albany
  
- [16] M.Teplan, FUNDAMENTALS OF EEG MEASUREMENT, Institute of Measurement Science, Slovak Academy of Sciences, Dúbravská cesta 9, 841 04 Bratislava, Slovakia, 2002
  
- [17] Σημειώσεις μαθήματος ΕΠΛ442: Υπολογιστικά Συστήματα Μάθησης.

- [18] Σ. Γιακουμάκη και Ζ. Παπαδοπούλου, Νευροεπιστήμες – Οι επιστήμες του εγκεφάλου εισαγωγικό κείμενο για την εκπαίδευση του κοινού, British Neuroscience Association European Dana Alliance for the Brain
- [19] Ε. Παπακώστα, Πρόβλεψη της Δευτεροταγούς Δομής των Πρωτεϊνών με την βοήθεια Echo State Network και δημιουργία διαδικτυακής εφαρμογής για Αναπαράσταση του προβλήματος, ατομική διπλωματική εργασία, Πανεπιστήμιο Κύπρου, 2016.
- [20] *BCI Competition IV*: <http://www.bbc.de/competition/iv/index.html#dataset1>

## Παράρτημα Α

Αλγόριθμος που έγραψε ο Mantas Lukoševičius (2012) [12], <http://minds.jacobs-university.de/mantas>.

ESNBCITrainTest.py

```
import numpy
import sklearn
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import roc_curve
from sklearn.metrics import auc
from sklearn.metrics import roc_auc_score
from sklearn.decomposition import FastICA
import sklearn.cross_decomposition
from matplotlib.pyplot import *
import scipy.linalg
import sys
import scipy.io
import time
import scipy.sparse
import random
```

```
def kappa(a,b,c,d):
    tot = a + b + c + d
    Pa = float(a + d)/tot
    PA1 = float(a + b)/tot
    PA2 = 1.0- PA1
    PB1 = float(a + c) /tot
    PB2 = 1.0 -PB1
    Pe = PA1 *PB1 + PA2*PB2
    print Pa, PA1, PB1, PA2, PB2
    return (Pa -Pe)/ (1.0 -Pe)
```

```
windowSize=400
numberOfChanel=59
numberOfEpochs=1
numOfInputRecords=200
inSize = 59
outSize = 1
resSize = 100
a = 0.3
```

```

SpParam = 1.25
regCoef = 1e-8
numOfExp = 1
initWMethod = 2
modifyWMethjod = 2
resPercForinitWMethod2=0.3

originaldatain=scipy.io.loadmat('BClinputdata100Hz.mat')
originaldataout=scipy.io.loadmat('BCIoutputdata100Hz.mat')
originaldatain=originaldatain['arr']
originaldataout=originaldataout['arr']

plot(originaldataout)
show()
plot(originaldatain)
show()

input_et=[0]*windowSize*numberOfChanel
output_et = 0
epoch=0
startTime=0
finishTime=0

inputMapping = range(0,numOfInputRecords)

timeonchain=0
mapCounter=0
while timeonchain<(originaldatain.shape[0]-1):
    if int(originaldataout[timeonchain])!=-1 or int(originaldataout[timeonchain])==1:
        inputMapping[mapCounter]=timeonchain
        mapCounter=mapCounter+1
        timeonchain=timeonchain+1

originaldatain=0.1*originaldatain

imin,jmin=(originaldatain.argmin(),originaldatain.shape)
imax,jmax=np.unravel_index(originaldatain.argmax(),originaldatain.shape)
originaldatain=(originaldatain-originaldatain[imin][jmin])/(originaldatain[imax][jmax]-
originaldatain[imin][jmin])

resSize = 5000
SpParam = 2.0

numOfExpCounter=0;

```

```

expResultsForParameter = range(0,numOfExp)
expResultsMSE = range(0,numOfExp)
expResultsASE = range(0,numOfExp)
expResultsACC = range(0,numOfExp)
expResultsKappa = range(0,numOfExp)

windowSize=400
resPercForinitWMethod2 = 0.05

#loop1
while(numOfExpCounter<numOfExp):

    modifyWMethjod = 1
    initWMethod = 1

    expResultsForParameter[numOfExpCounter]=SpParam

    random.seed(42)

    Win = (numpy.random.rand(resSize,1+inSize)-0.5) * 1

    W = numpy.random.rand(resSize,resSize)-0.5

    if(initWMethod==1):
        W = numpy.random.rand(resSize,resSize)-0.5
    else:
        W=scipy.sparse.rand(resSize,resSize,resPercForinitWMethod2).toarray()
        W[W!=0]-=0.5

    Wback = numpy.random.rand(resSize,outSize)-0.5

    print 'Computing spectral radius...',
    rhoW = max(abs(linalg.eig(W)[0]))
    print 'done.'

    if(modifyWMethjod==1):
        W *= SpParam / rhoW
    else:
        W = W / rhoW

    newrhoW = max(abs(linalg.eig(W)[0]))

```

```

print 'epoch:',epoch,'rhoW:',rhoW,'newrhoW:',newrhoW,'SpParam / rhoW:',SpParam / rhoW

inputMappingTraining=random.sample(inputMapping,int(len(inputMapping)*2/3))
inputMappingTest=set(inputMapping)-set(inputMappingTraining)
inputMappingTest=list(inputMappingTest)

X = numpy.zeros((resSize,len(inputMappingTraining)))

Yt = range(0,len(inputMappingTraining))

x=numpy.empty(resSize)
x.fill(0)
ones=(numpy.zeros(50)+1).T
u = numpy.empty(inSize+1)
u.fill(1)

epoch=0

#loop2

while epoch<numberOfEpochs:

timeonchain=0

startTime=time.time()

#loop3
while timeonchain<(len(inputMappingTraining)):

fr=inputMappingTraining[timeonchain]
toin=inputMappingTraining[timeonchain]+windowSize
input_et=originaldatain[fr:toin][:]

output_et=originaldataout[inputMappingTraining[timeonchain]]
Yt[timeonchain]=output_et

x=empty(resSize)
x.fill(0)

#loop4

for t in range(windowSize):

u[1:(inSize+1)] = input_et[t,:]

```

```

x = (1-a)*x + a*tanh(dot( Win, u ) + dot( W, x ))

#end of loop 4

X[:,timeonchain] = x

timeonchain=timeonchain+1

#end of loop 3

reg = regCoef

X_T = X.T

Wout = dot( dot(vstack(Yt).T,X_T), linalg.inv( dot(X,X_T) + reg*eye(resSize) ) )

Y = empty(len(inputMappingTest))
Y.fill(0)

Yt = range(0,len(inputMappingTest))

Yint = empty(len(inputMappingTest))
Yint.fill(0)

timeonchain=0

startTime=time.time()

#loop5
while timeonchain<(len(inputMappingTest)):

    fr=inputMappingTest[timeonchain]-windowSize
    toin=inputMappingTest[timeonchain]
    input_et=originaldatain[fr:toin][:]

    output_et=originaldataout[inputMappingTest[timeonchain]]
    Yt[timeonchain]=output_et

    x=empty(resSize)
    x.fill(0)

#loop6

```



```

for t in range(windowSize):

    u[1:(inSize+1)] = input_et[t,:]

    x = (1-a)*x + a*tanh( dot( Win, u ) + dot( W, x ))

#end of loop 6

y = dot( Wout, x )

Y[timeonchain] = y

timeonchain=timeonchain+1

#end of loop 5

finishTime=time.time()

epoch=epoch+1

#end of loop 2

#loop7
for accuracyt in range(len(inputMappingTest)):
    if(Y[accuracyt]>0):
        Yint[accuracyt]=1
    else:
        Yint[accuracyt]=-1

#end of loop 7
Yt=numpy.asarray(Yt)

confusionMatrix = confusion_matrix(Yt , Yint)
MSEValue = mean_squared_error(Yt , Y)
ACCValue = accuracy_score(Yt , Yint, normalize=True)

expResultsMSE[numOfExpCounter] = MSEValue
expResultsACC[numOfExpCounter] = ACCValue
expResultsKappa[numOfExpCounter] =
kappa(confusionMatrix[0][0],confusionMatrix[0][1],confusionMatrix[1][0],confusionMatrix[1][1]
])

print "Experiment:",numOfExpCounter
print "Parameter Value:",expResultsForParameter[numOfExpCounter]
print "MSE:",expResultsMSE[numOfExpCounter]

```

```

print "ACC:",expResultsACC[numOfExpCounter]
print "Cohen's Kappa:",expResultsKappa[numOfExpCounter]
print "Confussion Matrix\n",confusionMatrix
print "ROC curve on Figure"

fpr = dict()
tpr = dict()
roc_auc_score = dict()
fpr, tpr, thresholds = roc_curve(Yt , Y)
roc_auc_score = auc(fpr, tpr)
print("Area under the ROC curve : %f" % roc_auc_score)

figure(numOfExpCounter).clf()
plot(fpr, tpr, label='ROC curve (area = %0.2f)' % roc_auc_score)
plot([0, 1], [0, 1], 'k--')
xlim([0.0, 1.0])
ylim([0.0, 1.0])
xlabel('False Positive Rate')
ylabel('True Positive Rate')
title('Receiver operating characteristic example')
legend(loc="lower right")
show()

numOfExpCounter=numOfExpCounter+1;

#end of loop 1

figure("Cohen's kappa").clear()
plot( expResultsForParameter, expResultsKappa, 'ro',label='Cohen\'s Kappa' )
title('Class prediction of EEG signals')
legend(loc="lower right")
xlabel('Parameter Value')
ylabel('Cohen\'s Kappa Value (1)')
show()

figure("ACC").clear()
plot( expResultsForParameter, expResultsACC,'bo' ,label='ACC' )
title('Class prediction of EEG signals')
legend(loc="lower right")
xlabel('Parameter Value')
ylabel('ACC (%)')
show()

figure("MSE").clear()

```

```
plot( expResultsForParameter, expResultsMSE, 'go' ,label='MSE' )  
title('Class prediction of EEG signals')  
legend(loc="lower right")  
xlabel('Parameter Value')  
ylabel('MSE (%)')
```

## Παράρτημα Β

demo.py

Αλγόριθμος που έγραψε ο Mantas Lukoševičius (2012) [12], <http://minds.jacobs-university.de/mantas>.

```
# -*- coding: utf-8 -*-
```

```
"""
```

A minimalistic Echo State Networks demo with Mackey-Glass (delay 17) data  
in "plain" scientific Python.

by Mantas Lukoševičius 2012

<http://minds.jacobs-university.de/mantas>

```
"""
```

```
from numpy import *
```

```
from matplotlib.pyplot import *
```

```
import scipy.linalg
```

```
# load the data
```

```
trainLen = 2000
```

```
testLen = 2000
```

```
initLen = 100
```

```
data = loadtxt('MackeyGlass_t17.txt')
```

```
# plot some of it
```

```
figure(10).clear()
```

```
plot(data[0:1000])
```

```
title('A sample of data')
```

```

# generate the ESN reservoir

inSize = outSize = 1

resSize = 1000

a = 0.3 # leaking rate


random.seed(42)

Win = (random.rand(resSize,1+inSize)-0.5) * 1

print'Win', Win

W = random.rand(resSize,resSize)-0.5

print'W', W

# Option 1 - direct scaling (quick&dirty, reservoir-specific):

#W *= 0.135

# Option 2 - normalizing and setting spectral radius (correct, slow):

print 'Computing spectral radius...',

rhoW = max(abs(linalg.eig(W)[0]))

print 'done.'

print'rhoW', rhoW

W *= 1.25 / rhoW

print'W', W

# allocated memory for the design (collected states) matrix

X = zeros((1+inSize+resSize,trainLen-initLen))

print'X', X

# set the corresponding target matrix directly

Yt = data[None,initLen+1:trainLen+1]

print'Yt', Yt

# run the reservoir with the data and collect X

```

```

x = zeros((resSize,1))
print'x', x
for t in range(trainLen):
    u = data[t]
    x = (1-a)*x + a*tanh( dot( Win, vstack((1,u)) ) + dot( W, x ) )
    print'x', x
    if t >= initLen:
        X[:,t-initLen] = vstack((1,u,x))[:,0]

# train the output
reg = 1e-8 # regularization coefficient
X_T = X.T
print'X.T',X.T
Wout = dot( dot(Yt,X_T), linalg.inv( dot(X,X_T) + \
    reg*eye(1+inSize+resSize) ) )
#Wout = dot( Yt, linalg.pinv(X) )
print'Wout', Wout

# run the trained ESN in a generative mode. no need to initialize here,
# because x is initialized with training data and we continue from there.
Y = zeros((outSize,testLen))
print'Y', Y
u = data[trainLen]
for t in range(testLen):
    x = (1-a)*x + a*tanh( dot( Win, vstack((1,u)) ) + dot( W, x ) )
    y = dot( Wout, vstack((1,u,x)) )
    Y[:,t] = y
    # generative mode:

```

```

u = y

## this would be a predictive mode:
#u = data[trainLen+t+1]

# compute MSE for the first errorLen time steps
errorLen = 500
mse = sum( square( data[trainLen+1:trainLen+errorLen+1] - Y[0,0:errorLen] ) ) / errorLen
print 'MSE = ' + str( mse )

# plot some signals
figure(1).clear()
plot( data[trainLen+1:trainLen+testLen+1], 'g' )
plot( Y.T, 'b' )
title('Target and generated signals  $y(n)$  starting at  $n=0$ ')
legend(['Target signal', 'Free-running predicted signal'])

figure(2).clear()
plot( X[0:20,0:200].T )
title('Some reservoir activations  $\mathbf{x}(n)$ ')

figure(3).clear()
bar( range(1+inSize+resSize), Wout.T )
title('Output weights  $\mathbf{W}^{\text{out}}$ ')

show()

```