# Αναλυτική Περιγραφή Μοντέλων

Σε αυτό το κείμενο θα αναφερθούν κάποια βασικά κομμάτια θεωρίας πάνω στα μοντέλα μηχανικής μάθησης που χρησιμοποιήθηκαν, ώστε να είναι ευκολότερη η κατανόηση της προσέγγισης και της πειραματικής διαδικασίας. Αναλυτικότερα, θα γίνει μία σύντομη περιγραφή και επεξήγηση των δύο διαφορετικών μοντέλων που χρησιμοποιήθηκαν και θα δοθούν συμπληρωματικά αποσπάσματα από των κώδικα όπου χρειάζονται.

## 

# 1. Σειριακά Μοντέλα ([Sequential Models (Tensorflow/Keras))](#_bookmark18)

Τα σειριακά μοντέλα των βιβλιοθηκών TensorFlow/Keras της Python[1] είναι ένας τύπος υπολογιστικού μοντέλου που προορίζεται για το σχεδιασμό μοντέλων βαθιάς μάθησης σε μια γραμμική στοίβα επιπέδων. Παρέχουν μια απλή μέθοδο για τη δημιουργία νευρωνικών δικτύων όπου ένα μοντέλο δημιουργείται στρώμα προς στρώμα.

ο τρόπος λειτουργίας τους είναι ο ακόλουθος:

* **Αρχικοποίηση(Initialization)**: Ένα διαδοχικό μοντέλο αρχικοποιείται αρχικά με τη συνάρτηση Sequential().
* **Προσθήκη στρωμάτων/επιπέδων(Adding Layers)**: Στη συνέχεια μπορούν να προστεθούν στρώματα στο μοντέλο με τη συνάρτηση add(). Αυτά τα στρώματα μπορούν να είναι πυκνά (πλήρως συνδεδεμένα) στρώματα, συνεπτυγμένα στρώματα κ.λπ. Κάθε στρώμα δέχεται εισόδους από προηγούμενα στρώματα (όπου το σχήμα εισόδου του ταιριάζει με το σχήμα εξόδου του προηγούμενου στρώματος) και στέλνει την έξοδό του στο επόμενο στρώμα.
* **Σύνταξη(Compilation)**: Αφού προστεθούν όλα τα επιθυμητά στρώματα, το μοντέλο μεταγλωττίζεται με τη συνάρτηση compile(), η οποία διαμορφώνει το μοντέλο για εκπαίδευση. Αυτό το βήμα καθορίζει τον βελτιστοποιητή (όπως SGD, Adam), τη συνάρτηση απώλειας (όπως categorical\_crossentropy, mse) και τις μετρικές (όπως ακρίβεια/accuracy) που θα χρησιμοποιεί το μοντέλο.
* **Εκπαίδευση(Training)**: Μετά τη μεταγλώττιση, το μοντέλο είναι έτοιμο να εκπαιδευτεί χρησιμοποιώντας δεδομένα. Για το σκοπό αυτό χρησιμοποιείται η συνάρτηση fit(). Δέχεται δεδομένα εισόδου, ετικέτες, αριθμό εποχών και μέγεθος δέσμης.
* **Αξιολόγηση και πρόβλεψη(Evaluation and Prediction)**: Αφού εκπαιδευτεί το μοντέλο, μπορεί να χρησιμοποιηθεί για την αξιολόγηση της απόδοσής του σε δεδομένα δοκιμής χρησιμοποιώντας την evaluate() και για την πραγματοποίηση προβλέψεων σε νέα δεδομένα χρησιμοποιώντας την predict().

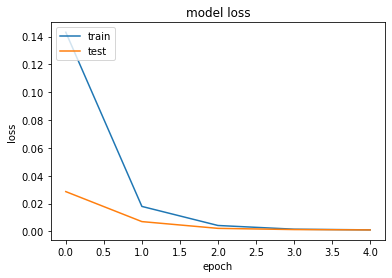
## 1.1 Σειριακό Μοντέλο Πρόβλεψης PDR με χρήση των Ptran,Prec,RSSI

Το πρώτο σειριακό μοντέλο χρησιμοποιεί τις τιμές της Ισχύος μετάδοσης και εκπομπής (Ptran,Prec) αλλά και την τιμή του θορύβου/παρεμβολής (RSSI) για να προβλέψει την αντίστοιχη τιμή αναλογίας παράδοσης Πακέτων (PDR). Ο κώδικας που έχει χρησιμοποιηθεί περιγράφεται παρακάτω (αρχικοποίηση, προσθήκη στρωμάτων, σύνταξη, εκπαίδευση, αξιολόγηση και πρόβλεψη):



**Εικόνα 1. Παράδειγμα Δημιουργίας, Σύνταξης, Εκπαίδευσης 1ου Σειριακού Μοντέλου**

Στο παραπάνω κομμάτι κώδικα γίνεται η αρχικοποίηση του μοντέλου, ύστερα προστίθενται τα απαραίτητα στρώματα, έπειτα γίνεται η σύνταξή του και τέλος εκπαιδεύεται πάνω στα δεδομένα εκπαίδευσης. Το μοντέλο διαθέτει 3 εισόδους και 1 έξοδο ενώ ενδιάμεσα υπάρχουν 2 κρυφά επίπεδα με 32 και 16 νευρώνες αντίστοιχα. Ως συνάρτηση ενεργοποίησης έχει οριστεί η “LeakyRelu”. Ως συνάρτηση απώλειας έχει οριστεί η συνάρτηση μέσου τετραγωνικού σφάλματος(mse) της οποίας οι τιμές κατά την διάρκεια των εποχών περιγράφονται από το παρακάτω διάγραμμα:



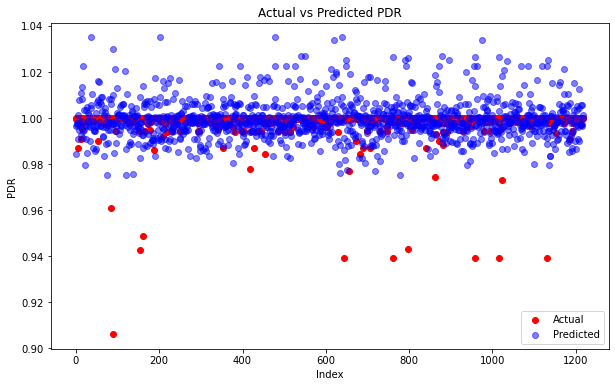
**Εικόνα 2. Γραφική Αναπαράσταση Διακύμανσης Απώλειας 1ου Σειριακού Μοντέλου**

Για τις προβλέψεις του παραπάνω μοντέλου χρησιμοποιήθηκε το κομμάτι δοκιμών του dataset (x\_test,y\_test):



**Εικόνα 3. Παράδειγμα Πρόβλεψης 1ου Σειριακού Μοντέλου**

Τα αποτελέσματα των προβλέψεων περιγράφονται από το παρακάτω γράφημα:



**Εικόνα 4. Γραφική Απεικόνιση Προβλέψεων 1ου Σειριακού Μοντέλου**

Αρκετά ικανοποιητικό αποτέλεσμα με τις περισσότερες προβλέψεις να είναι αρκετά κοντά ή και σχεδόν ίδιες με τις πραγματικές τιμές. Το παραπάνω γράφημα μπορεί να επιβεβαιωθεί και από τις τιμές των ανάλογων συναρτήσεων σφάλματος:

* **Mean Absolute Error**: 0.0067646327650631825
* **Mean Squared Error**: 0.00012134809571202615
* **Root Mean Squared Error**: 0.01101581116904362

Όλες οι τιμές μας δείχνουν, πως παρότι υπάρχει σφάλμα, το παραπάνω μοντέλο θα μπορούσε να προβλέψει αρκετά ικανοποιητικά τις τιμές που επιθυμούμε.

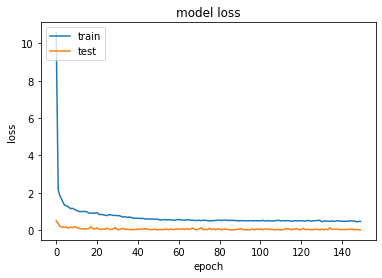
## 1.2 Σειριακό Μοντέλο Πρόβλεψης Ptran με χρήση των Prec,RSSI,PDR

Το δεύτερο σειριακό μοντέλο χρησιμοποιεί την τιμής ισχύος εκπομπής (Prec), την τιμή του θορύβου/παρεμβολής (RSSI) αλλά και την τιμή της αναλογίας παράδοσης Πακέτων (PDR) για να προβλέψει την τιμή της ισχύος μετάδοσης (Ptran) .Ο κώδικας που έχει χρησιμοποιηθεί περιγράφεται παρακάτω (αρχικοποίηση, προσθήκη στρωμάτων, σύνταξη, εκπαίδευση, αξιολόγηση και πρόβλεψη):



**Εικόνα 5. Παράδειγμα Δημιουργίας, Σύνταξης, Εκπαίδευσης 2ου Σειριακού Μοντέλου**

Στο παραπάνω κομμάτι κώδικα γίνεται η αρχικοποίηση του μοντέλου, ύστερα προστίθενται τα απαραίτητα στρώματα, έπειτα γίνεται η σύνταξή του και τέλος εκπαιδεύεται πάνω στα δεδομένα εκπαίδευσης. Το μοντέλο διαθέτει 3 εισόδους και 1 έξοδο ενώ ενδιάμεσα υπάρχουν 3 κρυφά επίπεδα με 64, 32 και 16 νευρώνες αντίστοιχα. Επίσης έχουν προστεθεί και επίπεδα κανονικοποίησης που χρησιμοποιούν τη μέθοδο “Dropout”. Ως συνάρτηση ενεργοποίησης έχει οριστεί η “LeakyRelu”. Επίσης ως μέθοδος βελτιστοποίησης των αποτελεσμάτων έχει προστεθεί και η διαδικασία των σημείων αναφοράς “Checkpoints’’ κατά την διάρκεια της εκπαίδευσης. Ως συνάρτηση απώλειας έχει οριστεί η συνάρτηση μέσου τετραγωνικού σφάλματος(mse) της οποίας οι τιμές κατά την διάρκεια των εποχών περιγράφονται από το παρακάτω διάγραμμα:



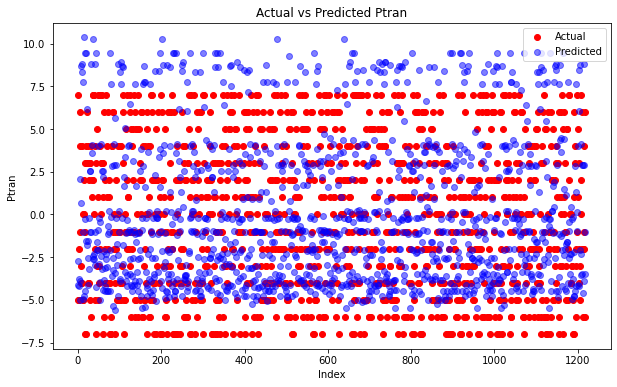
**Εικόνα 6. Γραφική Αναπαράσταση Διακύμανσης Απώλειας 2ου Σειριακού Μοντέλου**

Για τις προβλέψεις του παραπάνω μοντέλου χρησιμοποιήθηκε το κομμάτι δοκιμών του dataset (x\_test,y\_test):



**Εικόνα 6. Παράδειγμα Πρόβλεψης 2ου Σειριακού Μοντέλου**

Τα αποτελέσματα των προβλέψεων περιγράφονται από το παρακάτω γράφημα:



**Εικόνα 7. Γραφική Απεικόνιση Προβλέψεων 2ου Σειριακού Μοντέλου**

Καθόλου ικανοποιητικά αποτελέσματα, με τις περισσότερες προβλέψεις να είναι αρκετά μακριά από τις πραγματικές και με κάποιες ακόμα να ξεφεύγουν από το όριο των επιτρεπτών/υπαρκτών τιμών. Το μοντέλο φαίνεται να μην μπορεί να ανταποκριθεί, ανεξαρτήτως των πολλαπλών στρωμάτων βελτιστοποίησης και κανονικοποίησης που έχουν εφαρμοστεί. Το παραπάνω γράφημα μπορεί επιβεβαιωθεί και από τις τιμές των ανάλογων συναρτήσεων σφάλματος:

* **Mean Absolute Error**: 5.271561640014504
* **Mean Squared Error**: 42.41495150154359
* **Root Mean Squared Error**: 6.512676216544439

Είναι προφανής η αστοχία του μοντέλου μας, καθώς οι συναρτήσεις σφαλμάτων έχουν υπερβολικά μεγάλες τιμές ,κάτι που υποδηλώνει πολύ μεγάλη απόκλιση στις προβλέψεις του μοντέλου.

# 2. Μοντέλα Τυχαίου Δάσους (Random Forest Regressor(Scikit-Learn/sklearn))

Ο Random Forest Regressor είναι ένα δημοφιλές μοντέλο μηχανικής μάθησης από τη βιβλιοθήκη Scikit-Learn. Είναι ένας μετα-εκτιμητής που προσαρμόζει έναν αριθμό δέντρων απόφασης ταξινόμησης σε διάφορα υποδείγματα του συνόλου δεδομένων και χρησιμοποιεί τον μέσο όρο για να βελτιώσει την ακρίβεια πρόβλεψης και να ελέγξει την υπερβολική προσαρμογή(over-fitting).

O τρόπος λειτουργίας τους είναι ο ακόλουθος:

* **Αρχικοποίηση(Initialization)**:: Ξεκινάει με την αρχικοποίηση ενός αντικειμένου τυχαίου δάσους (RandomForestRegressor) καθορίζοντας παραμέτρους όπως ο αριθμός των δέντρων (n\_estimators), η συνάρτηση για τη μέτρηση της ποιότητας ενός διαχωρισμού (criterion, η οποία είναι προεπιλεγμένη σε "mse" για το μέσο τετραγωνικό σφάλμα στην περίπτωση της παλινδρόμησης(regression)) και άλλα.
* **Εκπαίδευση(Τraining):** Αφού αρχικοποιηθεί, προσαρμόζετε το μοντέλο στα δεδομένα εκπαίδευσής σας χρησιμοποιώντας τη μέθοδο fit(). Σε αυτό το σημείο δημιουργείται το δάσος των δέντρων. Για κάθε δέντρο, ο αλγόριθμος θα επιλέξει τυχαία ένα υποσύνολο δειγμάτων από τα δεδομένα σας (με αντικατάσταση) και ένα υποσύνολο χαρακτηριστικών και στη συνέχεια θα χωρίσει αυτά τα δεδομένα για να σχηματίσει κλαδιά και φύλλα που ελαχιστοποιούν το καθορισμένο κριτήριο (όπως το μέσο τετραγωνικό σφάλμα). Αυτή η διαδικασία επαναλαμβάνεται για όλα τα δέντρα.
* **Πρόβλεψη(Prediction):** Μετά την προσαρμογή του μοντέλου, μπορείτε να κάνετε προβλέψεις σε νέα δεδομένα χρησιμοποιώντας τη μέθοδο predict(). Αυτό περιλαμβάνει την εκτέλεση των νέων δεδομένων σε κάθε ένα από τα δέντρα για να καταλήξετε σε μια πρόβλεψη. Για το RandomForestRegressor, η τελική πρόβλεψη είναι ο μέσος όρος των προβλέψεων όλων των δέντρων.
* **Αξιολόγηση(Evaluate)**: H αξιολόγηση της απόδοσης του μοντέλου γίνεται χρησιμοποιώντας κατάλληλες μετρικές όπως το μέσο τετραγωνικό σφάλμα (MSE) ή μέσο απόλυτο σφάλμα (MAE) κ.λπ.

Ένα από τα κύρια πλεονεκτήματα των Random Forests είναι ότι μπορούν να μοντελοποιήσουν πολύπλοκες αλληλεπιδράσεις μεταξύ των χαρακτηριστικών και δεν απαιτούν κλιμάκωση των χαρακτηριστικών εισόδου.

## 2.1 Μοντέλο Πρόβλεψης PDR με χρήση των Ptran,Prec,RSSI

Το πρώτο μοντέλο χρησιμοποιεί τις τιμές της Ισχύος μετάδοσης και εκπομπής (Ptran,Prec) αλλά και την τιμή του θορύβου/παρεμβολής (RSSI) για να προβλέψει την αντίστοιχη τιμή αναλογίας παράδοσης Πακέτων (PDR). Ο κώδικας που έχει χρησιμοποιηθεί περιγράφεται παρακάτω (αρχικοποίηση, εκπαίδευση, πρόβλεψη και αξιολόγηση):



**Εικόνα 8. Παράδειγμα Δημιουργίας, Εκπαίδευσης 1ου Μοντέλου Τυχαίου Δάσους**

Στο παραπάνω κομμάτι κώδικα αρχικοποιούμε το μοντέλο. Στο μοντέλο προσθέτουμε 100 ‘’δέντρα’’ με χρήση της παραμέτρου ‘’n\_estimators’’ ύστερα το εκπαιδεύουμε στα δεδομένα εκπαίδευσης του συνόλου δεδομένων.

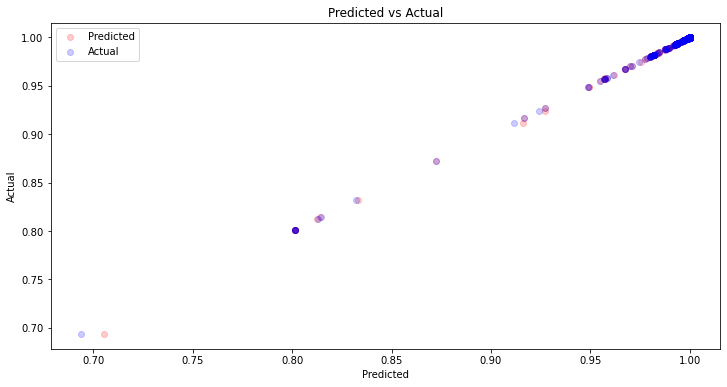
Στο συγκεκριμένο είδος μοντέλου δεν χρειάζεται η διαδικασία της σύνταξης(Compiling). Αυτό οφείλεται στο γεγονός ότι αυτά τα μοντέλα δεν βασίζονται στη βελτιστοποίηση με βάση τη διαβάθμιση, η οποία απαιτεί τον προσδιορισμό μιας συνάρτησης απώλειας και ενός βελτιστοποιητή(όπως στα μοντέλα του Tensorflow/keras). Αντ' αυτού, μοντέλα όπως αυτό του τυχαίου δάσους (RandomForestRegressor) έχουν συγκεκριμένους αλγορίθμους που ακολουθούν για να μάθουν από τα δεδομένα, οι οποίοι δεν απαιτούν τέτοιες ρυθμίσεις.

Για τις προβλέψεις του παραπάνω μοντέλου χρησιμοποιήθηκε το κομμάτι δοκιμών του dataset (x\_test,y\_test):



**Εικόνα 9. Παράδειγμα Πρόβλεψης 1ου Μοντέλου Τυχαίου Δάσους**

Τα αποτελέσματα των προβλέψεων περιγράφονται από το παρακάτω γράφημα:



**Εικόνα 10. Γραφική Απεικόνιση Προβλέψεων 1ου Μοντέλου Τυχαίου Δάσους**

Εξαιρετικό αποτέλεσμα με τις περισσότερες προβλέψεις να είναι ακριβώς ίδιες με τις πραγματικές τιμές των δεδομένων. Παρατηρούνται ελάχιστα σφάλματα, καθώς το μοντέλο κατάφερε να εκπαιδευτεί σχεδόν άψογα πάνω στα δεδομένα εκπαίδευσης. Το παραπάνω γράφημα μπορεί επιβεβαιωθεί και από τη τιμή της ανάλογης συνάρτησης σφάλματος:

* **Mean Squared Error**: 0.000378

Το μέσω τετραγωνικό σφάλμα έχει τιμή που τείνει στο μηδέν κάτι που μας υποδηλώνει ελάχιστη απόκλιση μεταξύ των πραγματικών και των τιμών που προέβλεψε το μοντέλο.

## 2.2 Μοντέλο Πρόβλεψης Ptran με χρήση των Prec,RSSI,PDR

Το δεύτερο σειριακό μοντέλο χρησιμοποιεί την τιμή ισχύος εκπομπής (Prec), την τιμή του θορύβου/παρεμβολής (RSSI) αλλά και την τιμή της αναλογίας παράδοσης Πακέτων (PDR) για να προβλέψει την τιμή της ισχύος μετάδοσης (Ptran) .Ο κώδικας που έχει χρησιμοποιηθεί περιγράφεται παρακάτω (αρχικοποίηση, εκπαίδευση, πρόβλεψη, αξιολόγηση):



**Εικόνα 10. Παράδειγμα Δημιουργίας, Εκπαίδευσης 2ου Μοντέλου Τυχαίου Δάσους**

Στο παραπάνω κομμάτι κώδικα αρχικοποιούμε το μοντέλο. Στο μοντέλο προσθέτουμε 100 ‘’δέντρα’’ με χρήση της παραμέτρου ‘’n\_estimators’’ ύστερα το εκπαιδεύουμε στα δεδομένα εκπαίδευσης του συνόλου δεδομένων.

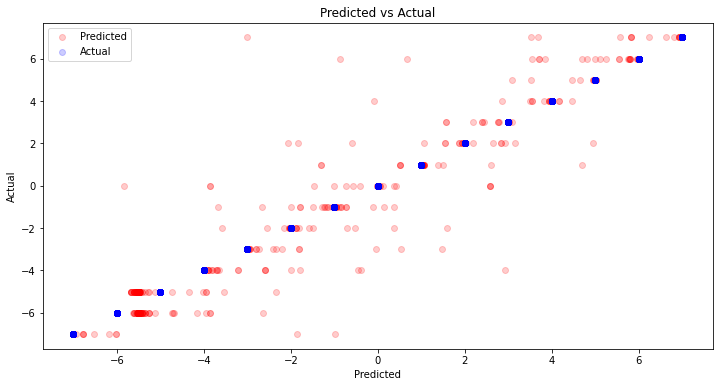
Όπως και στο προηγούμενο μοντέλο η διαδικασία της σύνταξης δεν χρησιμοποιείται.

Για τις προβλέψεις του παραπάνω μοντέλου χρησιμοποιήθηκε το κομμάτι δοκιμών του dataset (x\_test,y\_test):



**Εικόνα 11. Παράδειγμα Πρόβλεψης 2ου Μοντέλου Τυχαίου Δάσους**

Τα αποτελέσματα των προβλέψεων περιγράφονται από το παρακάτω γράφημα:



**Εικόνα 12. Γραφική Απεικόνιση Προβλέψεων 2ου Μοντέλου Τυχαίου Δάσους**

Αρκετά ικανοποιητικό αποτέλεσμα με τις περισσότερες προβλέψεις να είναι αρκετά κοντά ή και σχεδόν ίδιες με τις πραγματικές τιμές. Το παραπάνω γράφημα μπορεί επιβεβαιωθεί και από τη τιμή της ανάλογης συνάρτησης σφάλματος:

* **Mean Squared Error**: 0.792601

Η τιμή του μέσου τετραγωνικού σφάλματος μας δείχνει πως παρ’ όλο που υπάρχει σφάλμα το παραπάνω μοντέλο θα μπορούσε να προβλέψει αρκετά ικανοποιητικά τις τιμές που επιθυμούμε.

# 3.Βιβλιογραφία

[1] N. K. Manaswi, “Understanding and Working with Keras,” *Deep Learning with Applications Using Python*, pp. 31–43, 2018, doi: 10.1007/978-1-4842-3516-4\_2.