

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ  
Π.Μ.Σ. ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ ΚΑΙ  
ΥΠΗΡΕΣΙΕΣ

Νευρωνικά Δίκτυα και Εφαρμογές αυτών για  
Πρόβλεψη Χρονοσειρών στο Χρηματιστήριο



Βασίλειος Θωμαΐσινας

Επιβλέπων: Μιχαήλ Φιλιππάκης

## Περιεχόμενα

Περίληψη .....	3
Κεφάλαιο 1: Εισαγωγή .....	4
Κεφάλαιο 2: .....	5
2.1 Μετοχές και Αγορές .....	5
2.2 Μηχανική Μάθηση .....	5
2.3 Νευρωνικά Δίκτυα .....	7
2.3.1 Multilayer Perceptron .....	8
2.3.2 Συνελκτικά Νευρωνικά Δίκτυα .....	13
2.3.3 Αναδρομικά Νευρωνικά Δίκτυα .....	16
2.3.4 Long Short-Term Memory (LSTM).....	18
Κεφάλαιο 3: Χρονοσειρές .....	19
3.1 Βασικά χαρακτηριστικά χρονοσειρών .....	20
3.1.1 Τάση:.....	20
3.1.2 Κυκλικότητα:.....	21
3.1.3 Εποχικότητα: .....	21
3.2 Αποσύνθεση χρονοσειρών.....	22
3.3 Ανάλυση Χρονοσειρών .....	24
Κεφάλαιο 4: Πειραματικό Μέρος και Αποτελέσματα .....	26
4.1 Λογισμικό .....	27
4.2 Δεδομένα .....	27
4.3 Μετασχηματισμός των Δεδομένων.....	34
4.4 Πειραματικό σκέλος .....	36
4.4.1 Πρόβλεψη μετοχής επόμενης ημέρας.....	36
4.4.2 Πρόβλεψη τάσης.....	43
4.4.3 Σύγκριση νευρωνικών δικτύων με κλασικές μεθόδους ανάλυσης χρονοσειρών .....	51
Κεφάλαιο 5: Συμπεράσματα .....	56
Βιβλιογραφία .....	58

## Περίληψη

Τα τελευταία χρόνια η επιστημονική και τεχνολογική εξέλιξη έχει οδηγήσει στην καθολικότητα της χρήσης νευρωνικών δικτύων για την επίλυση διάφορων προβλημάτων. Μια απ' τις πολλές εφαρμογές των νευρωνικών δικτύων, αφορά στη χρήση τους για την ανάλυση και πρόβλεψη χρονοσειρών που αφορούν χρηματιστηριακούς δείκτες.

Στην εν λόγω εργασία παρουσιάζονται οι κύριες αρχιτεκτονικές νευρωνικών δικτύων που χρησιμοποιούνται για την ανάλυση χρονοσειρών. Με χρήση των γλωσσών προγραμματισμού R και Python, αναλύουμε 4 χρονοσειρές που αφορούν χρηματιστηριακούς δείκτες εταιρειών που σχετίζονται με την αυτοκινητοβιομηχανία, μοντελοποιούμε κατάλληλα τα προβλήματα της πρόβλεψης τιμών χρονοσειρών και πρόβλεψης μελλοντικής τάσης αυτών, εκπαιδεύουμε κατάλληλα νευρωνικά δίκτυα για την επίλυση αυτών των προβλημάτων κι αξιολογούμε τα αποτελέσματα των πειραμάτων.

## Κεφάλαιο 1: Εισαγωγή

Ζούμε την εποχή των μεγάλων δεδομένων. Κάθε μέρα ολοένα και περισσότερες πηγές παράγουν δεδομένα που μεταφράζονται σε χρήσιμη πληροφορία μέσω της οποίας μπορεί να παραχθεί γνώση. Η γνώση αυτή είναι χρήσιμη σε πολλούς τομείς της ζωής μας και η σωστή αξιοποίησή της μπορεί να αποτελέσει χρήσιμο εργαλείο στη λήψη αποφάσεων. Ως απόρροια της ανάγκης που δημιουργείται για την αποθήκευση, επεξεργασία και ανάλυση του μεγάλου όγκου αυτών των δεδομένων, τα τελευταία χρόνια βρίσκει πρόσφορο έδαφος ανάπτυξης και άνθισης ο τομέας της επιστήμης που καλείται επίσημη των δεδομένων (data science).

Μεγάλο μέρος της επιστήμης των δεδομένων αποτελεί η αξιοποίηση αλγοριθμικών μοντέλων για την ανάλυση των δεδομένων και προσδιορίζεται από τον όρο μηχανική μάθηση (machine learning). Η μηχανική μάθηση αποτελεί το σύνολο των αλγορίθμων που χρησιμοποιούνται για την ανακάλυψη προτύπων μέσα απ' τα δεδομένα και έχει ως στόχο της την παραγωγή γνώσης.

Ένας παράγοντας που συνεισέφερε τα τελευταία χρόνια στην ανάπτυξη της μηχανικής μάθησης και τις εφαρμογές αυτής σε μια πληθώρα προβλημάτων της αγοράς, είναι η διαρκώς αυξανόμενη διάθεση που έχουμε σε υπολογιστικούς πόρους. Την εποχή που διατρέχουμε, η νεφρουπολογιστική (cloud computing) επιτρέπει σε ελάχιστο χρόνο να περατώνουμε διεργασίες μηχανικής μάθησης μεγάλης κλίμακας, σε χρόνο ο οποίος πριν από 10 χρόνια θα φάνταζε απίθανος.

Η αστάθεια λόγω της αβεβαιότητας στο χώρο των αγορών, καθιστά επιβεβλημένη τη χρήση των εργαλείων της επιστήμης των δεδομένων και ιδιαίτερα της μηχανικής μάθησης. Η αβεβαιότητα αυτή αναδεικνύει τη σημαντικότητα χρήσης αλγορίθμων μηχανικής μάθησης για τη διεξαγωγή προβλέψεων, ικανών να τη μειώσουν όσο το δυνατόν περισσότερο. Για το λόγο αυτό γίνεται όλο και πιο διαδεδομένη η χρήση τεχνητών νευρωνικών δικτύων (ΤΝΔ) για τη διεκπεραίωση τέτοιων προβλέψεων, όπως για παράδειγμα είναι η πρόβλεψη τιμών δεικτών.

Σε αυτή την εργασία εξετάζουμε την εφαρμογή τεχνητών νευρωνικών δικτύων για την πρόβλεψη χρονοσειρών στο χρηματιστήριο. Αρχικά στο κεφάλαιο 2 παρουσιάζουμε βασικά στοιχεία μηχανικής μάθησης, καθώς επίσης και βασικές αρχιτεκτονικές νευρωνικών δικτύων. Στο κεφάλαιο 3 γίνεται μια εισαγωγή στη μελέτη των χρονοσειρών και στο κεφάλαιο 4 ακολουθεί το πειραματικό σκέλος της εργασίας. Το κεφάλαιο 5 κλείνει την εργασία με τα συμπεράσματα που προκύπτουν από αυτή, μαζί με προτάσεις για μελλοντική επέκταση της.

## Κεφάλαιο 2:

### 2.1 Μετοχές και Αγορές

Οι μετοχές μιας εταιρίας είναι μερίδια κατοχής του κεφαλαίου της. Κάθε κάτοχος μιας μετοχής, ανάλογα το είδος της μετοχής που έχει και το πλήθος τους, έχει δικαιώματα συμμετοχής στη διοίκηση μιας εταιρίας και στη λήψη αποφάσεων.

Οι μετοχές είναι αντικείμενα διαπραγμάτευσης στην αγορά του χρηματιστηρίου και κάθε επενδυτής (ιδιώτης ή εταιρία) μπορεί να τις αγοράσει από εκεί. Οι αγορές αυτές χαρακτηρίζονται από ιδιαίτερη αβεβαιότητα καθότι υπάρχουν πάρα πολλοί παράγοντες που επηρεάζουν τις τιμές τους.

### 2.2 Μηχανική Μάθηση

Η μηχανική μάθηση είναι ο τομέας της επιστήμης των δεδομένων που ασχολείται με εφαρμογές μαθηματικών αλγορίθμων σε πληροφοριακά συστήματα για την εξόρυξη γνώσης, αποτελώντας το σύνολο των αλγορίθμων οι οποίοι μέσω της διαδικασίας της εκπαίδευσης, είναι ικανοί στο να ξεχωρίζουν μοτίβα μέσα από σύνολα δεδομένων. Εξαιτίας του μεγάλου όγκου αυτών των δεδομένων, θα αποτελούσε μεγάλη δυσκολία για έναν άνθρωπο να υλοποιήσει αυτούς τους αλγορίθμους από μόνος του και να βγάλει κάποια χρήσιμα συμπεράσματα για τα δεδομένα αυτά. Για το λόγο αυτό κατά τη μηχανική μάθηση οι αλγόριθμοι εκτελούνται με τη βοήθεια υπολογιστικών συστημάτων,

τα οποία μειώνουν το χρόνο εκτέλεσης των αλγορίθμων και καθιστούν την εκπαίδευση χρονικά ανεκτή.

Μεγάλο μέρος της μηχανικής μάθησης αποτελεί ο κλάδος της τεχνητής νοημοσύνης (Artificial Intelligence). Η τεχνητή νοημοσύνη ασχολείται με προβλήματα μηχανικής μάθησης όπως είναι η μηχανική όραση (computer vision) κα, τα οποία έχουν μεγάλο βαθμό πολυπλοκότητας, γεγονός που καθιστά πολλές φορές την εκτέλεση των εργασιών εκπαίδευσης των αντίστοιχων μοντέλων χρονοβόρα.

Μπορούμε να θεωρήσουμε ότι η μηχανική μάθηση χωρίζεται σε δύο μεγάλες κατηγορίες, την επιβλεπόμενη μηχανική μάθηση (supervised machine learning) και τη μη επιβλεπόμενη μηχανική μάθηση (unsupervised machine learning) [1].

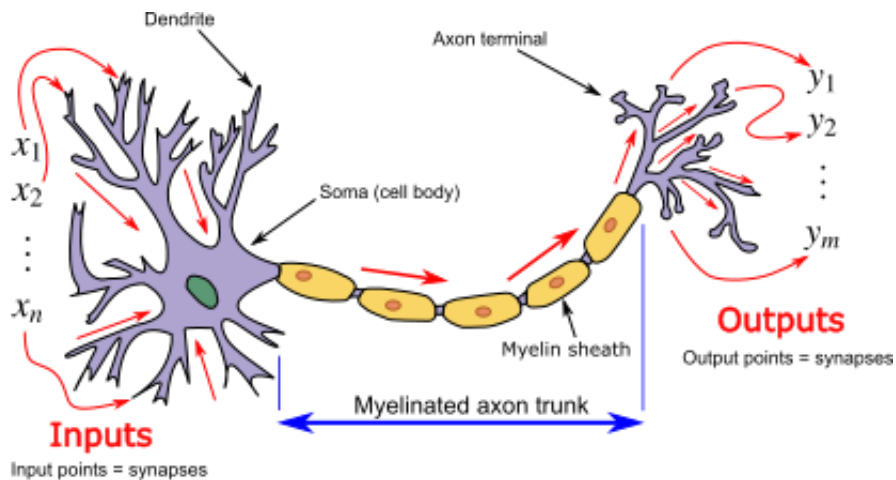
Η επιβλεπόμενη μηχανική μάθηση στοχεύει στην εκπαίδευση μοντέλων συσχετιζόμενα με ήδη υπάρχουσα πληροφορία κατά την εκπαίδευση. Δίνοντας ως είσοδο στον αλγόριθμο κάποια δεδομένα (inputs) προσδοκούμε το μοντέλο μας να παράξει κάποια δεδομένα εξόδου (outputs). Μερικές συχνές εφαρμογές επιβλεπόμενης μηχανικής μάθησης αποτελούν: α) η ταξινόμηση (classification) όπου σκοπός μας είναι να ταξινομήσουμε τα δεδομένα εισόδου σε κάποια συγκεκριμένη κατηγορία, β) η παλινδρόμηση (regression) όπου σκοπός μας είναι ανάλογα με τα δεδομένα εισόδου να επιστρέψουμε μία ή περισσότερες τιμές στόχους ως πρόβλεψη κα. Μερικοί αλγόριθμοι ταξινόμησης είναι η λογιστική παλινδρόμηση (logistic regression) που χρησιμοποιεί τη λογαριθμοπιθανοφάνεια για να κάνει ταξινόμηση σε μια κατηγορία, οι μηχανές διανυσμάτων στήριξης (support vector machines) που χρησιμοποιούν γεωμετρικά χαρακτηριστικά των δεδομένων, λύνοντας ένα πρόβλημα γραμμικού προγραμματισμού (linear programming), ενώ ο πιο διαδεδομένος αλγόριθμος παλινδρόμησης είναι αυτός της γραμμικής παλινδρόμησης (linear regression) που στοχεύει στην ελαχιστοποίηση του μέσου τετραγωνικού σφάλματος [2].

Η μη επιβλεπόμενη μηχανική μάθηση σε αντίθεση με την επιβλεπόμενη, στοχεύει στην εκπαίδευση μοντέλων χωρίς τη χρήση γνώσης για τα δεδομένα εξόδου. Η πιο συνηθισμένη εφαρμογή μη επιβλεπόμενης μηχανικής μάθησης, είναι αυτή της συσταδοποίησης (clustering) κατά την οποία θέλουμε να βρούμε υποομάδες δεδομένων στο σύνολο των δεδομένων μας. Μερικοί αλγόριθμοι

συσταδοποίησης αποτελούν ο αλγόριθμος k-means που εφαρμόζει συσταδοποίηση με βάσει τα κεντροειδή και ο DBSCAN που εφαρμόζει συσταδοποίηση με βάση την πυκνότητα των δεδομένων μας [2].

## 2.3 Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks) είναι υπολογιστικά μοντέλα που χρησιμοποιούνται στη μηχανική μάθηση. Η βασική ιδέα απ' την οποία εμπνέονται είναι οι βιολογικοί νευρώνες [1][2][3].



Βιολογικός νευρώνας [4]

Όπως οι βιολογικοί νευρώνες μεταδίδουν το ηλεκτρικό σήμα με μια σειρά από τους δενδρίτες στις απολήξεις, έτσι και τα νευρωνικά δίκτυα αποτελούνται από συστοιχίες υπολογιστικών μονάδων οι οποίες χρησιμοποιούνται για τη διάδοση της πληροφορίας. Πιο συγκεκριμένα, αναφερόμενοι στα νευρωνικά δίκτυα εμπρόσθιας διάδοσης, έχουμε επίπεδα (layers) από υπολογιστικές μονάδες (units) που συνδέονται μεταξύ τους.

Στα νευρωνικά δίκτυα δίνουμε τα δεδομένα εισόδου στο πρώτο επίπεδο (input layer), τα επεξεργάζεται το νευρωνικό δίκτυο μέσω του στρώματος των κρυφών επιπέδων (hidden layers) και έτσι παράγεται ένα διάνυσμα εξόδου μέσω του επιπέδου εξόδου (output layer).

Για το σχεδιασμό νευρωνικών δικτύων, είναι πολλές φορές σκόπιμο να χρησιμοποιούμε υπολογιστικά γραφήματα (computational graphs). Τα υπολογιστικά γραφήματα είναι γραφήματα που περιγράφουν μια ακολουθία εφαρμογής μαθηματικών πράξεων που εκτελούνται στα νευρωνικά δίκτυα.

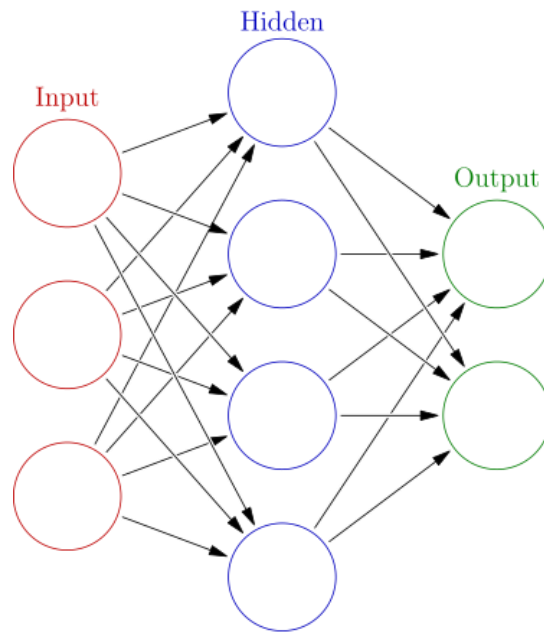
## 2.3.1 Multilayer Perceptron

Ένα νευρωνικό δίκτυο με περισσότερα από ένα επίπεδα στα hidden layers, ονομάζεται πολυστρωματικός perceptron (multilayer perceptron ή MLP). Στον multilayer perceptron τα σήματα δεδομένων εισόδου έχουν μοναδική κατεύθυνση διάδοσης πληροφορίας και καταλήγουν στο διάνυσμα εξόδου. Για την επεξεργασία και την εμπρόσθια διάδοση του σήματος, τα νευρωνικά δίκτυα χρησιμοποιούν συναπτικά βάρη (synaptic weights) που συνδέουν τις μονάδες μεταξύ τους, καθώς επίσης και μια άλλη ποσότητα που καλείται προκατάληψη (bias). Το bias είναι ένα μέτρο για την αβεβαιότητα του σήματος που διαδίδεται πρόσθια απ' τους νευρώνες ενός επιπέδου στους νευρώνες του επόμενου. Πιο συγκεκριμένα, αν με  $x_{ij}$  συμβολίσουμε την τιμή του  $j$ -οστού νευρώνα στο  $i$ -οστό επίπεδο, τότε θεωρούμε το αποτέλεσμα της συνάρτησης

$$\sum_j w_{ij} \cdot x_j + \mu_j$$

η οποία αποτελεί έναν γραμμικό συνδυασμό των τιμών των νευρώνων του  $j$ -οστού επιπέδου και των αντίστοιχων συναπτικών βαρών του  $x_j$  με τον αντίστοιχο νευρώνα που ετοιμάζεται να δεχτεί την πληροφορία, προστιθέμενα με το αντίστοιχο bias  $\mu_j$ . Στη συνέχεια το εν λόγω άθροισμα δίνεται ως όρισμα σε μια συνάρτηση ενεργοποίησης (activation function), το αποτέλεσμα της οποίας τροφοδοτεί τον εκάστοτε νευρώνα του  $i+1$  επιπέδου.





Βασική αρχιτεκτονική ενός Multilayer Perceptron [4]

Οι συναρτήσεις ενεργοποίησης είναι συνήθως μη γραμμικές συναρτήσεις, προκειμένου να προσομοιώσουν τη μη γραμμική συσχέτιση μεταξύ των νευρώνων. Μερικές απ' τις πιο συνήθεις συναρτήσεις ενεργοποίησης δίνονται παρακάτω:

Ταυτοτική συνάρτηση:

$$f(x) = x$$

Σιγμοειδής συνάρτηση:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Υπερβολική εφαπτομένη:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Rectified Linear Unit:

$$f(x) = \max(0, x)$$

Softmax:

$$f(x)_i = \frac{e^{x_i}}{\sum_j e^{x_j}}$$

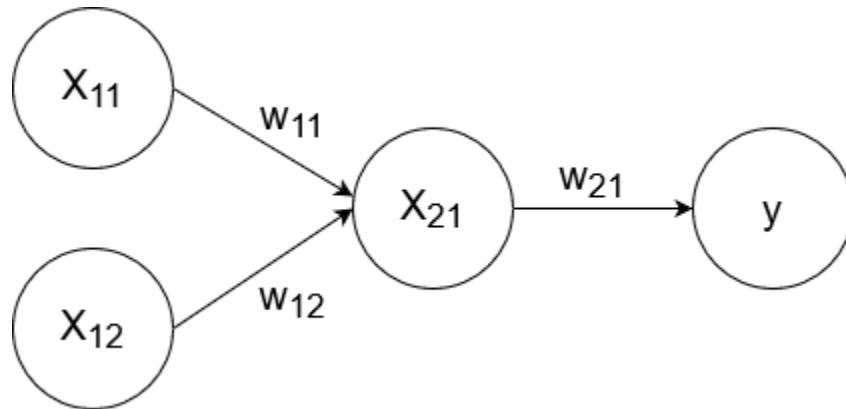
Η softmax συνήθως χρησιμοποιείται μεταξύ του τελευταίου επιπέδου των hidden layers και του output σε προβλήματα ταξινόμησης πολλών κλάσεων για την εξομάλυνση των τιμών και την παραγωγή πιθανοτήτων.

Εκτός απ' τις προαναφερθείσες συναρτήσεις ενεργοποίησης, υπάρχουν ακόμα πολλές άλλες, καθώς επίσης και ιδιαίτερες παραλλαγές αυτών.

Για την εκπαίδευση των νευρωνικών δικτύων, χρησιμοποιείται η τεχνική της οπίσθιας διάδοσης σήματος (backpropagation) [6]. Κατά το backpropagation συγκρίνουμε τις τιμές εξόδου του νευρωνικού δικτύου με τις αντίστοιχες επιθυμητές τιμές εξόδου μέσω μιας συνάρτησης κόστους (loss function) και χρησιμοποιώντας τον κανόνα της αλυσίδας (chain rule), ανανεώνουμε τα δομικά στοιχεία του δικτύου, δηλαδή τα συναπτικά βάρη και τα biases. Έστω ότι θέλουμε να εφαρμόσουμε backpropagation στο παρακάτω νευρωνικό δίκτυο, το οποίο το χρησιμοποιούμε για να λύσουμε ένα πρόβλημα παλινδρόμησης με τη βοήθεια της συνάρτησης κόστους του Μέσου Τετραγωνικού Σφάλματος (Mean Squared Error), που δίνεται απ' τον τύπο

$$L = \frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2$$

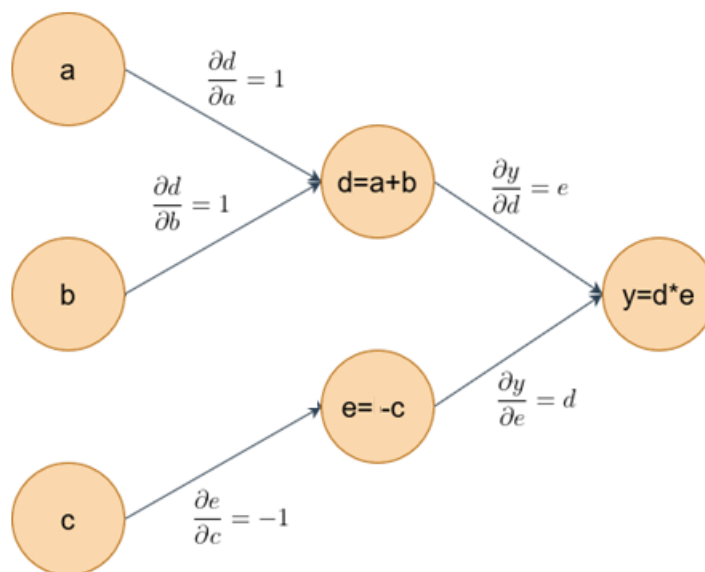
Όπου  $p_i$  είναι η πραγματική τιμή του output και  $y_i$  είναι το output που δίνει το νευρωνικό δίκτυο με αρχιτεκτονική όπως παρακάτω



Ορίζοντας ένα βήμα μάθησης (learning rate)  $\alpha$ , ανανεώνουμε τα συναπτικά βάρη  $w_{ij}$  με τον εξής τρόπο:

$$w_{ij} = w_{ij} - \alpha \frac{\partial L}{\partial w_{ij}}$$

Πιο συγκεκριμένα με χρήση ενός computational graph [5][6], όπως παρακάτω, μπορούμε να δούμε πως λειτουργεί η διαδικασία του backpropagation



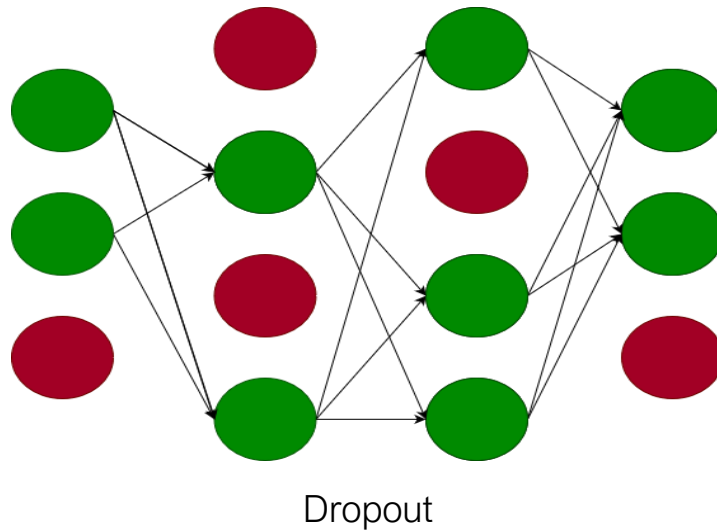
Έτσι, αν θέλουμε, μπορούμε να υπολογίσουμε τις μερικές παραγώγους του  $y$  ως προς τις άλλες παραμέτρους, για να τις ανανεώσουμε, με τον εξής τρόπο:

$$\frac{\partial y}{\partial a} = \frac{\partial y}{\partial d} \cdot \frac{\partial d}{\partial a} = e \cdot 1 = e$$

$$\frac{\partial y}{\partial b} = \frac{\partial y}{\partial d} \cdot \frac{\partial d}{\partial b} = e \cdot 1 = e$$

$$\frac{\partial y}{\partial c} = \frac{\partial y}{\partial e} \cdot \frac{\partial e}{\partial c} = d \cdot -1 = -d$$

Οι Multilayer Perceptrons είναι ικανοί να προσαρμοστούν σε οποιαδήποτε μη γραμμική συνάρτηση θέλουμε. Αυτό μας δείχνει την ικανότητά τους να ανακαλύπτουν και να προσαρμόζονται σε μη γραμμικές σχέσεις των δεδομένων, γεγονός που πολλές φορές μπορεί να ενέχει και κίνδυνο. Ένα νευρωνικό δίκτυο βασίζεται αποκλειστικά και μόνο στα δεδομένα εκπαίδευσης τα οποία του παρέχουμε και ως εκ τούτου, караδοκεί ο κίνδυνος της υπερπροσαρμογής (overfitting) [7]. Η υπερπροσαρμογή των δεδομένων σχετίζεται με την ιδιότητα των μοντέλων μηχανικής μάθησης να «μαθαίνουν» πολύ καλά με βάση τα δεδομένα εκπαίδευσης και την στέρση ικανότητας γενίκευσης. Μπορούμε να την εντοπίσουμε κατά την εκπαίδευση ενός νευρωνικού δικτύου, όταν παρατηρούμε το σφάλμα που έχουμε ορίσει να μικραίνει για τα δεδομένα εκπαίδευσης και να μεγαλώνει (ή να παραμένει σταθερό), χρησιμοποιώντας δεδομένα που δε χρησιμοποιήσαμε για εκπαίδευση. Υπάρχουν διάφορες τεχνικές για την αποφυγή της υπερπροσαρμογής κατά την εκπαίδευση ενός νευρωνικού δικτύου. Μερικές από αυτές είναι ο περαιτέρω εμπλουτισμός του συνόλου δεδομένων εκπαίδευσης (data augmentation), η χρήση κάποιων ποινών στα διάφορα επίπεδα του νευρωνικού δικτύου ( $L_p$  regularization), η εφαρμογή της τεχνικής του dropout κα. Το dropout αφορά στην τυχαία επιλογή ενός συγκεκριμένου συνόλου συναπτικών βαρών κατά την εκπαίδευση, το οποίο μεταβάλλεται ανάλογα με τα δεδομένα εισόδου τυχαία κατά την πρόσθια διάδοση.



Κατά τον τρόπο αυτό, θεωρούμε ότι για ένα συγκεκριμένο στιγμιότυπο του δικτύου, δεν υπάρχει πλήρης σύνδεση μεταξύ των νευρώνων από το ένα επίπεδο στο άλλο.

### 2.3.2 Συνελικτικά Νευρωνικά Δίκτυα

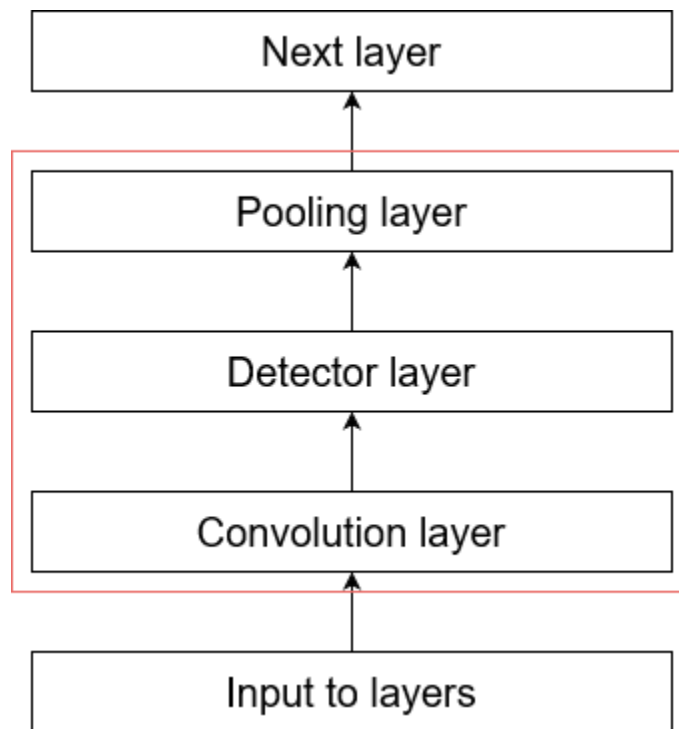
Τα συνελικτικά νευρωνικά δίκτυα (convolutional neural networks) είναι μια ιδιαίτερη αρχιτεκτονική νευρωνικών δικτύων που χρησιμοποιείται συχνά στον τομέα της υπολογιστικής όρασης, καθότι είναι ικανά στο να βρίσκουν χωρικές συσχετίσεις μεταξύ των δεδομένων. Η τελευταία διαπίστωση, οφείλεται στη χρήση ειδικών επιπέδων συγκέντρωσης (pooling layers) στην αρχιτεκτονική τους, γεγονός που καθιστά τη συνάρτηση εκμάθησης αμετάβλητη σε μικρές χωρικές μεταβολές [6]. Πήραν το όνομά τους εξαιτίας της πράξης της συνέλιξης κατά τη διάδοση του σήματος στην αρχιτεκτονική τους, που είναι μια μαθηματική πράξη μεταξύ πινάκων.

Στην αρχιτεκτονική τους, τα συνελικτικά νευρωνικά δίκτυα απαρτίζονται από συνελικτικά επίπεδα (convolutional layers), επίπεδα συγκέντρωσης (pooling layers) και πλήρως συνδεδεμένα επίπεδα (fully connected layers). Τα πλήρως συνδεδεμένα επίπεδα δεν διαφέρουν απ' τους multilayer perceptrons.

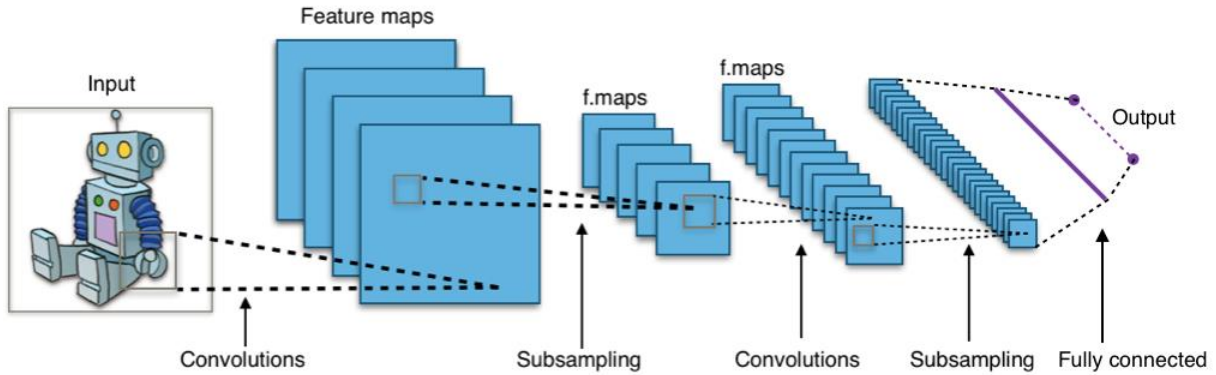
Τα συνελικτικά δίκτυα μπορούν να δεχτούν ως είσοδο πολυδιάστατα αντικείμενα, όπως για παράδειγμα πίνακες, τένσορες κ.α. Τα συνελικτικά

επίπεδα λαμβάνουν μέρος στη διαδικασία εκπαίδευσης, έχοντας βάρη που ανανεώνονται κατά το backpropagation. Για το σκοπό αυτό του μετασχηματισμού και της εξαγωγής χαρακτηριστικών (feature transformation/extraction) τα συνελκτικά επίπεδα χρησιμοποιούν συναρτήσεις πυρήνων (kernel functions). Οι συναρτήσεις πυρήνων παίζουν σημαντικό ρόλο στον ορισμό της δομής του δικτύου. Στα pooling layers γίνεται δειγματοληψία των χαρακτηριστικών κάθε επιπέδου προκειμένου να δημιουργηθεί μια χαρτογράφηση των χαρακτηριστικών (feature mapping) και να μειωθούν οι προς εκπαίδευση παράμετροι.

Παρακάτω δίνονται τα τυπικά βήματα εκτέλεσης ενός συνελκτικού δικτύου μέσω των convolutional και των pooling layers [6]



Στο σχήμα φαίνεται η εφαρμογή κι ενός επιπέδου ανίχνευσης (detector layer) που είναι προαιρετική.



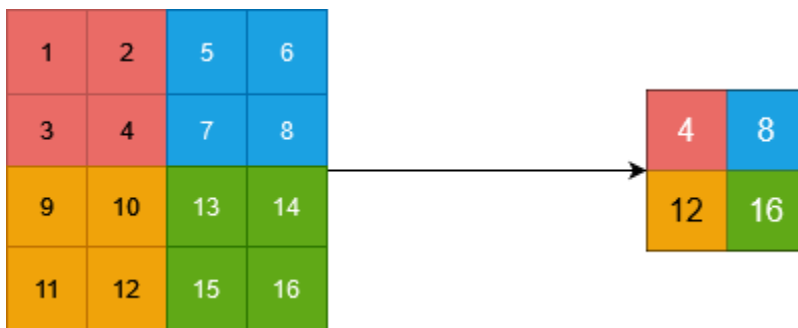
Συνελικτικό νευρωνικό δίκτυο [8]

Χρησιμοποιώντας δισδιάστατα δεδομένα, όπως για παράδειγμα πίνακες, ως δεδομένα εισόδου, μπορούμε να ορίσουμε την πράξη της συνέλιξης ως εξής

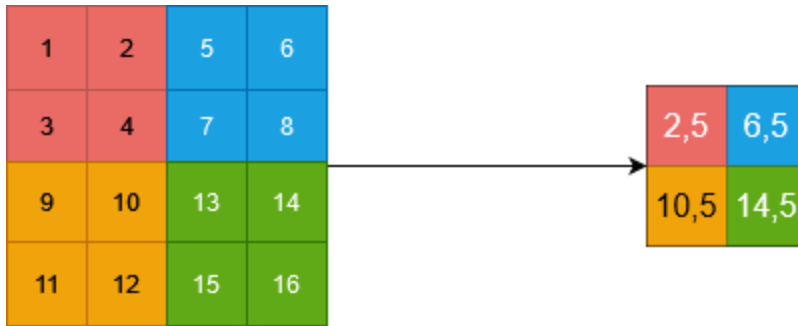
$$S(i, j) = (K \star I)(i, j) = \sum_m \sum_n I(n, m)K(i - m, j - n)$$

χρησιμοποιώντας αντίστοιχες kernel functions K και I.

Για τη διαδικασία του pooling μπορούν να εφαρμοστούν διάφορες μέθοδοι. Μερικές από αυτές είναι η εξαγωγή της μέγιστης τιμής στο πλέγμα που έχει οριστεί (max pooling) ή της μέσης τιμής (average pooling)



Εφαρμογή max pooling



Εφαρμογή average pooling

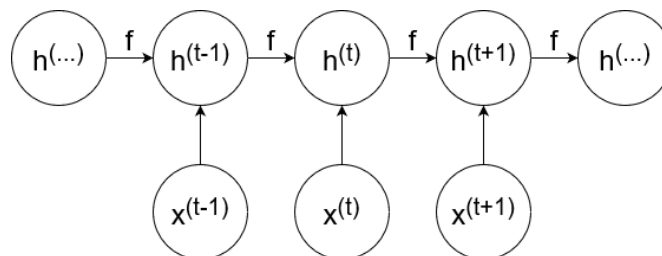
Για την εκπαίδευση ενός συνελκτικού νευρωνικού δικτύου εφαρμόζουμε backpropagation, ενώ για την αποφυγή του overfitting μπορούμε να χρησιμοποιήσουμε τις ίδιες τεχνικές με τους multilayer perceptrons.

### 2.3.3 Αναδρομικά Νευρωνικά Δίκτυα

Τα αναδρομικά νευρωνικά δίκτυα (recurrent neural networks) είναι μια ιδιαίτερη αρχιτεκτονική νευρωνικών δικτύων, αφού ειδικεύεται στην επεξεργασία ακολουθιακών δεδομένων όπως οι χρονοσειρές [6]. Έστω  $x=(x_1, \dots, x_n)$  ένα διάνυσμα, τότε αν θέλουμε να χρησιμοποιήσουμε πληροφορία απ' τη χρονική περίοδο  $t-1$  για την χρονική περίοδο  $t$ , μπορούμε να χρησιμοποιήσουμε τον αναδρομικό τύπο

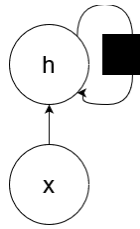
$$h^{(t)}=f(h^{(t-1)}, x^{(t)}; \theta)$$

Για να δούμε πως λειτουργεί στην πράξη αυτό, μπορούμε να σχεδιάσουμε το υπολογιστικό γράφημα

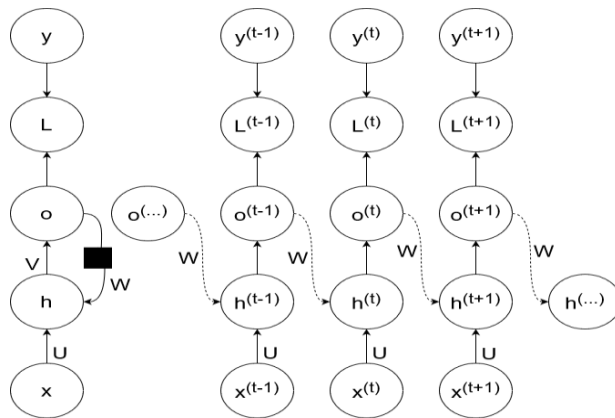


Το συγκεκριμένο υπολογιστικό γράφημα μπορεί να περιγραφεί με τη μορφή ενός κλειστού βρόγχου όπως παρακάτω





Κατ' αυτό τον τρόπο μπορούμε να επεκτείνουμε την έννοια του υπολογιστικού γραφήματος, δημιουργώντας κύκλους. Αυτό μας δίνει τη δυνατότητα να χτίσουμε πιο περίπλοκα δίκτυα, συνδυάζοντας αποτελέσματα υπολογισμών σε προηγούμενες πράξεις



όπου

$$a^{(t)} = b + Wh^{(t-1)} + Ux^{(t)}$$

$$h^{(t)} = \tanh(a^{(t)})$$

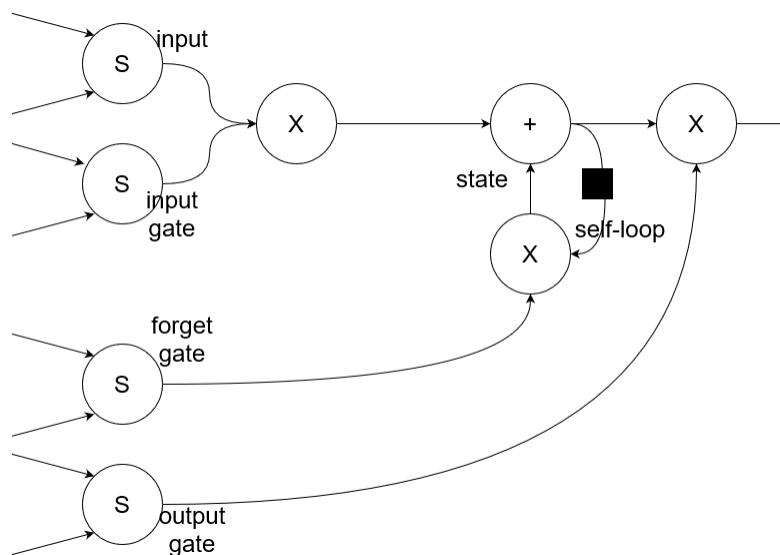
$$o^{(t)} = c + Vh^{(t)}$$

$$y_{\text{hat}} = \text{softmax}(o^{(t)})$$

οι πράξεις που εκτελούνται σε κάθε κόμβο του υπολογιστικού γραφήματος, με  $b$ ,  $c$  να είναι τα διανύσματα μεροληψίας (biases),  $U$ ,  $V$  και  $W$  οι πίνακες των βαρών,  $o$  το output vector που παίρνουμε χρησιμοποιώντας τη συνάρτηση  $h$  και  $y_{\text{hat}}$  η εκτιμωμένη τιμή που μαζί με την πραγματική τιμή  $y$  τροφοδοτούν τη συνάρτηση κόστους  $L$ .

### 2.3.4 Long Short-Term Memory (LSTM)

Τα LSTMs [6][9] είναι μια ειδική μορφή Gated recurrent neural networks. Βασικό χαρακτηριστικό των LSTMs είναι τα LSTM κελιά (LSTM cells), τα οποία έχουν έναν εσωτερικό μηχανισμό αναδρομής. Κάθε LSTM κελί έχει την ίδια δομή με ένα τυπικό αναδρομικό νευρωνικό δίκτυο όσον αφορά τα δεδομένα εισόδου και εξόδου, αλλά έχει περισσότερες παραμέτρους καθώς επίσης και μια μονάδα πύλης (gating unit) που διαχειρίζεται τη ροή της πληροφορίας στο δίκτυο. Η πιο σημαντική παράμετρος ενός LSTM είναι η μονάδα κατάστασης (state unit) που απαρτίζεται από έναν γραμμικό βρόγχο (self-loop) κι ελέγχεται από μια πύλη λήθης (forget gate) της οποίας σκοπός είναι να ορίζει το βάρος στο διάστημα  $[0, 1]$  μέσω μιας σιγμοειδούς συνάρτησης.



LSTM cell

Συμβολίζουμε με  $x^{(t)}$  το input vector, με  $f_i^t$  το forget gate για το χρονικό βήμα  $t$  του  $i$ -οστού cell, με  $h^{(t)}$  το διάνυσμα του hidden layer που περιέχει τα outputs ενός LSTM cell, με  $b^f$ ,  $U^f$  και  $W^f$  τις μεροληψίες, τα βάρη εισόδου και τα αναδρομικά βάρη για τα forget gates αντίστοιχα, με  $s_i^{(t)}$  το state unit για το χρονικό βήμα  $t$  του  $i$ -οστού cell, με  $g_i^{(t)}$  το external input gate για το χρονικό βήμα  $t$  του  $i$ -οστού cell, με  $h_i^{(t)}$  το output του LSTM cell, με  $q_i^{(t)}$  το output gate για το χρονικό βήμα  $t$  του  $i$ -οστού cell και  $b^o$ ,  $U^o$  και  $W^o$  τις μεροληψίες, τα input weights και τα recurrent weights του  $q_i^{(t)}$  αντίστοιχα. Με βάση αυτό το συμβολισμό, σύμφωνα με το [2] οι υπολογισμοί σε ένα LSTM cell γίνονται με βάσει τους παρακάτω τύπους

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

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

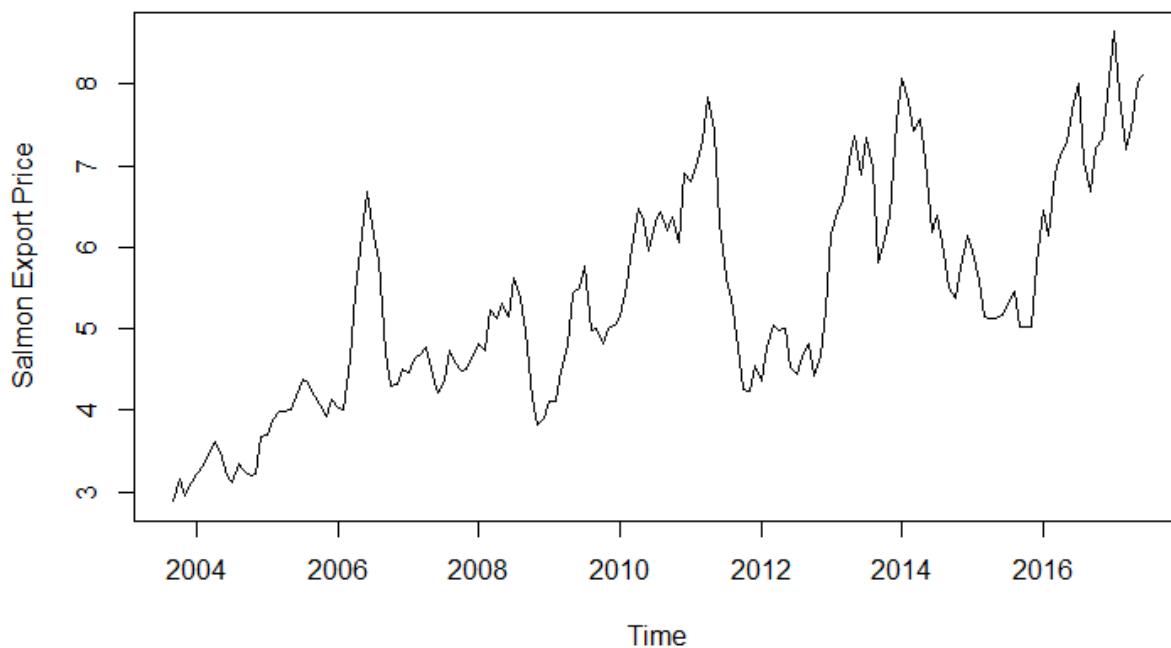
$$g_i^{(t)} = \sigma(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)})$$

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

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

### Κεφάλαιο 3: Χρονοσειρές

Οι χρονοσειρές είναι ακολουθίες δεδομένων οι οποίες σχετίζονται με το χρόνο παρατήρησής τους [10]. Γενικά περιγράφονται μέσω μιας ακολουθίας  $(X_t)$  όπου  $t \in T$ , όπου με  $T$  συμβολίζουμε το πεδίο ορισμού του χρόνου. Το  $T$  μπορεί να αποτελείται από διακριτές τιμές, οπότε έχουμε να κάνουμε με μια χρονοσειρά διακριτού χρόνου, ή μπορεί να αποτελείται από συνεχείς τιμές, οπότε έχουμε να κάνουμε με μια χρονοσειρά συνεχούς χρόνου. Ως μια χρονοσειρά συνεχούς χρόνου μπορούμε να θεωρήσουμε τη μέση ταχύτητα ενός αυτοκινήτου ανά πάσα στιγμή, ενώ ως ένα παράδειγμα διακριτής χρονοσειράς μπορούμε να θεωρήσουμε την μηνιαία τιμή εξαγωγής σολομού των νορβηγικών εκτροφείων από το Σεπτέμβριο του 2003 μέχρι τον Ιούνιο του 2017 (χρονοσειρά salmon του πακέτου `astsa` της R)

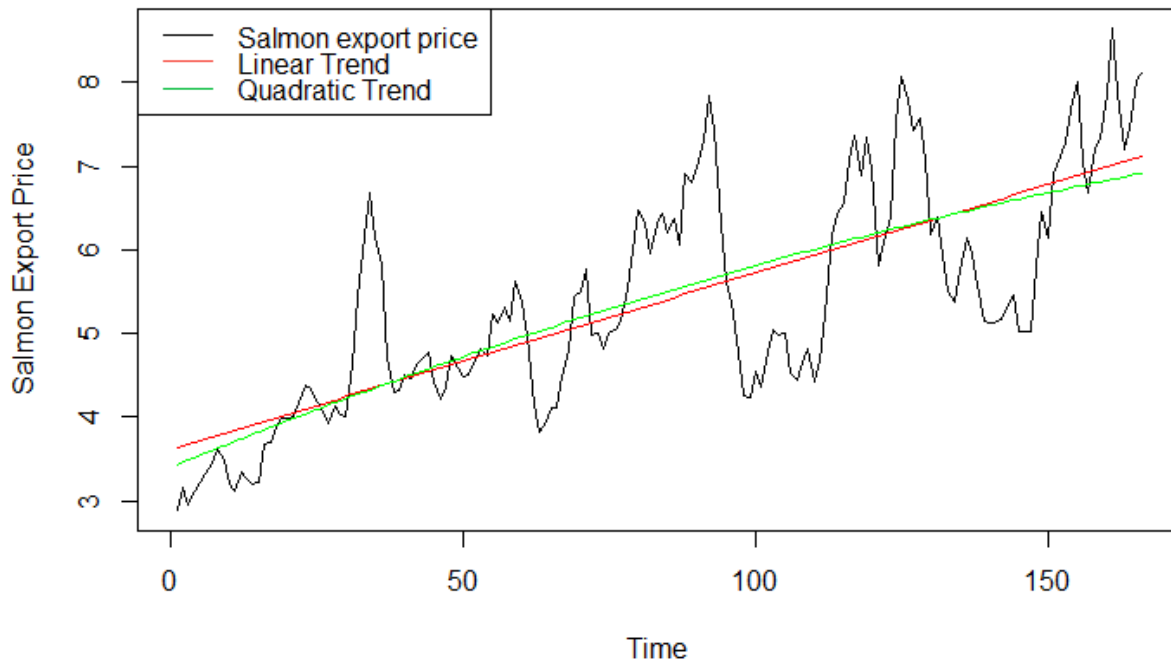


### 3.1 Βασικά χαρακτηριστικά χρονοσειρών

Μια χρονοσειρά απαρτίζεται από 3 κύρια χαρακτηριστικά: α)την τάση (trend), β)την εποχικότητα (seasonality) και γ)την κυκλικότητα (cyclic).

#### 3.1.1 Τάση:

Η τάση εκφράζει τη γενική κατεύθυνση προς την οποία κινείται η χρονοσειρά. Ένας τρόπος για να υπολογίσουμε την τάση, είναι εφαρμόζοντας μοντέλο γραμμικής παλινδρόμησης ή με τη βοήθεια του κινητού μέσου (moving average). Παρακάτω κάνουμε εκτίμηση της τάσης της παραπάνω χρονοσειράς χρησιμοποιώντας γραμμική και πολυωνυμική παλινδρόμηση βαθμού 2.



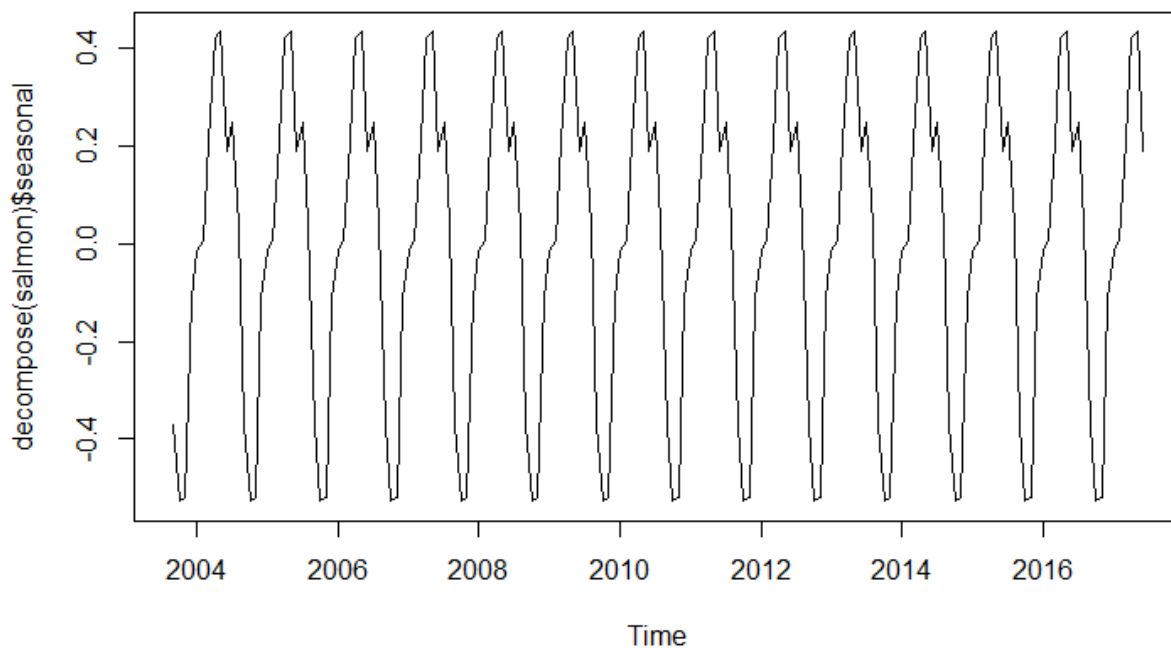
Βλέπουμε ότι χρησιμοποιώντας παλινδρόμηση για την εκτίμηση της τάσης στη συγκεκριμένη χρονοσειρά, αυτή φαίνεται να είναι αυξητική.

### 3.1.2 Κυκλικότητα:

Η κυκλικότητα εκφράζει την τάση μιας χρονοσειράς να αυξομειώνεται σε ακαθόριστες συχνότητες.

### 3.1.3 Εποχικότητα:

Η εποχικότητα μοιάζει με την κυκλικότητα καθώς εκφράζει την τάση μιας χρονοσειράς να αυξομειώνεται. Η διαφορά της με την κυκλικότητα, έγκειται στο γεγονός ότι αυτή η αυξομείωση συμβαίνει σε καθορισμένες συχνότητες. Η εποχικότητα για την παραπάνω χρονοσειρά δίνεται στο παρακάτω γράφημα.



### 3.2 Αποσύνθεση χρονοσειρών

Οι χρονοσειρές μπορούν να “σπάσουν” βάσει των παραπάνω στοιχείων που τις χαρακτηρίζουν. Οι δύο βασικοί τρόποι αποσύνθεσης χρονοσειρών είναι: α) το αθροιστικό μοντέλο (additive) αποσύνθεσης

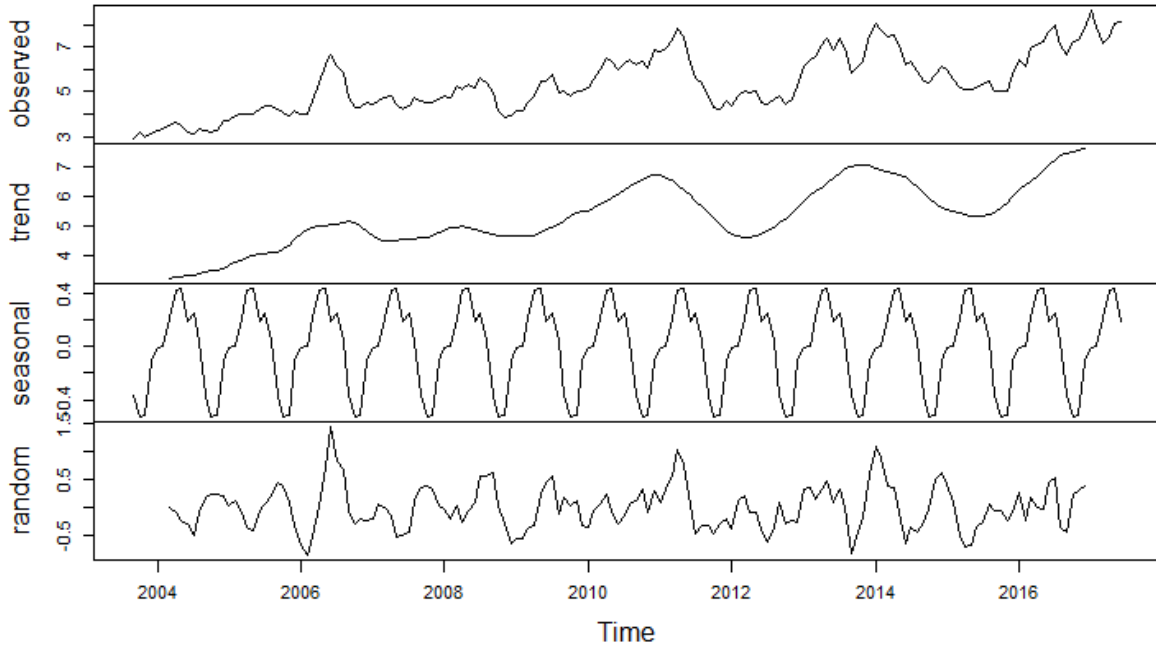
$$X_t = T_t + S_t + R_t$$

και β) το πολλαπλασιαστικό (multiplicative) μοντέλο αποσύνθεσης

$$X_t = T_t \cdot S_t \cdot R_t$$

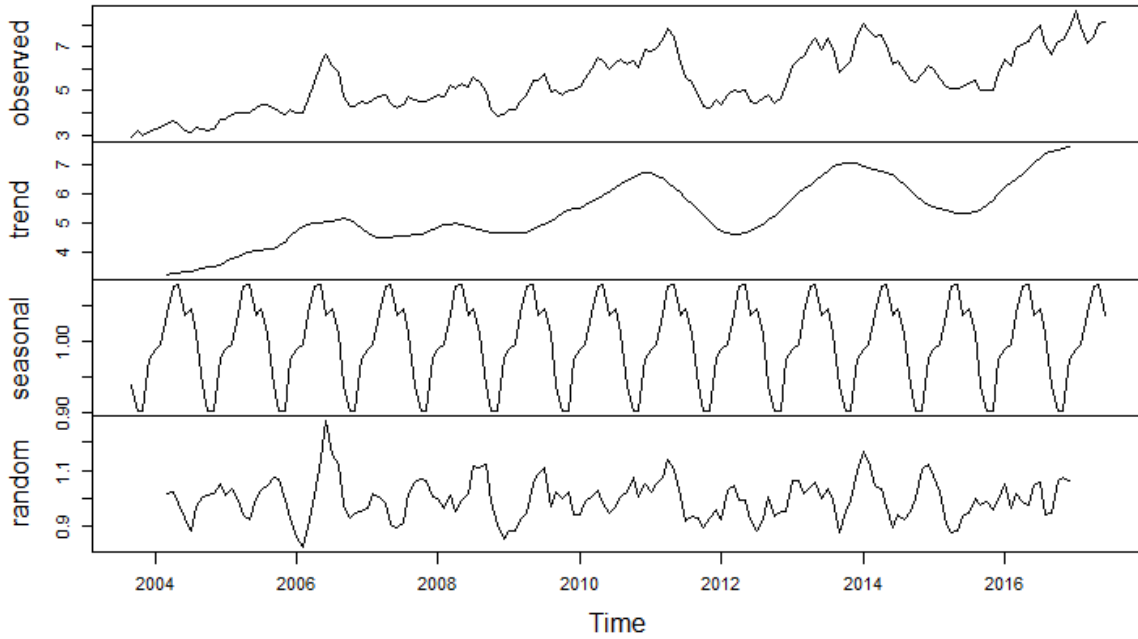
όπου  $R_t$  είναι τα κατάλοιπα (residuals) της χρονοσειράς τα οποία για το αθροιστικό μοντέλο δίνονται ως  $R_t = X_t - T_t - S_t$ , ενώ για το πολλαπλασιαστικό μοντέλο δίνονται ως  $R_t = X_t / (T_t \cdot S_t)$ . Παρακάτω παρουσιάζεται η αποσύνθεση της παραπάνω.

### Decomposition of additive time series



Αθροιστικό μοντέλο αποσύνθεσης

### Decomposition of multiplicative time series



Πολλαπλασιαστικό μοντέλο αποσύνθεσης

Ένα χαρακτηριστικό των χρονοσειρών, αποτελεί η ιδιότητα της στασιμότητας. Μία στάσιμη χρονοσειρά είναι μια χρονοσειρά η οποία δεν εξαρτάται απόλυτα από το χρόνο. Αυτό σημαίνει ότι μια χρονοσειρά η οποία έχει τάση ή εποχικότητα δεν μπορεί να χαρακτηριστεί ως στάσιμη. Για να κάνουμε μια χρονοσειρά στάσιμη, μπορούμε να αφαιρέσουμε την τάση ή την εποχικότητά της μέσω της διαδικασίας της αποσύνθεσης. Ένας άλλος τρόπος είναι εφαρμόζοντας τον τελεστή της διαφοράς στη χρονοσειρά μας

$$\Delta x_t = x_{t+1} - x_t$$

Μπορούμε επίσης να εφαρμόσουμε διαδοχικά τον τελεστή της διαφοράς στη χρονοσειρά μας για να την απαλλάξουμε από την τάση και την εποχικότητα:  $\Delta^2 x_t = \Delta(\Delta(x_t)) = x_t - 2x_{t-1} + x_{t-2}$ . Εφαρμόζοντας αναδρομικά τον τελεστή της διαφοράς παίρνουμε και τα αντίστοιχα αποτελέσματα.

### 3.3 Ανάλυση Χρονοσειρών

Έχοντας ως τελικό μας σκοπό την ανάλυση των χρονοσειρών, να κάνουμε προβλέψεις για τη μελλοντική συμπεριφορά τους και την πρόβλεψη μελλοντικών τιμών, χρησιμοποιούμε διάφορα στατιστικά μοντέλα. Τα πιο συνήθη μοντέλα που χρησιμοποιούνται είναι τα: Moving Average (MA) μοντέλα, τα Auto Regressive (AR) μοντέλα, τα Auto Regressive Moving Average (ARMA) μοντέλα και τα Auto Regressive Integrated Moving Average (ARIMA) μοντέλα [12]. Παρακάτω δίνονται οι βασικές περιγραφές των παραπάνω μοντέλων:

#### **Moving Average:**

Δεδομένης μιας χρονοσειράς ( $X_t$ ), ένα AR(p) μοντέλο δίνεται ως

$$X_t = \sum_{i=1}^p \phi_i X_{t-i} + \varepsilon_t$$

όπου οι  $\phi_1, \dots, \phi_p$  είναι παράμετροι του μοντέλου και  $\varepsilon_t$  είναι τυχαία μεταβλητή που ονομάζεται λευκός θόρυβος.

#### **Auto Regressive:**

Δεδομένης μιας χρονοσειράς ( $X_t$ ), ένα MA(q) μοντέλο δίνεται ως



$$X_t = \mu + \varepsilon_t + \sum_{i=1}^p \theta_i \varepsilon_{t-i}$$

όπου οι  $\theta_1, \dots, \theta_q$  είναι παράμετροι του μοντέλου,  $\mu$  είναι η μέση τιμή της χρονοσειράς και  $\varepsilon_t$  είναι τυχαία μεταβλητή που ονομάζεται λευκός θόρυβος.

#### Auto Regressive Moving Average:

Δεδομένης μιας χρονοσειράς ( $X_t$ ), ένα ARMA( $p$ ) μοντέλο δίνεται ως

$$X_t = \varepsilon_t + \sum_{i=1}^p \phi_i X_{t-i} + \sum_{i=1}^q \theta_i \varepsilon_{t-i}$$

όπου οι παράμετροι είναι συνδυαστικά αυτές που ορίστηκαν παραπάνω για τα MA και AR μοντέλα.

#### Auto Regressive Integrated Moving Average:

Δεδομένης μιας χρονοσειράς ( $X_t$ ), ένα ARIMA( $p, i, q$ ) μοντέλο δίνεται ως

$$X_t - a_1 X_{t-1} - \dots - a_p X_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}$$

όπου το  $i$  ορίζει την τάξη εφαρμογής του τελεστή διαφοράς,  $a_j$  είναι οι παράμετροι του αυτοπαλινδρομούμενου μέρους του μοντέλου,  $\theta_j$  είναι οι παράμετροι του μέρους του μοντέλου που αφορά τον κινητό μέσο και  $\varepsilon_j$  είναι όροι σφάλματος.

#### Τρόποι αξιολόγησης

Για την αξιολόγηση ενός μοντέλου που χρησιμοποιείται για την ανάλυση χρονοσειρών, χρησιμοποιούμε διάφορες μετρικές. Μερικές από αυτές δίνονται παρακάτω

#### Μέσο Απόλυτο Σφάλμα (Mean Absolute Error, MAE):

Το MAE είναι η μέση τιμή της απόλυτης διαφοράς των τιμών που παράγει το μοντέλο από τις πραγματικές τιμές της χρονοσειράς, κάθε χρονική στιγμή

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - a_i|$$

όπου  $y_i$  είναι οι πραγματική τιμή της χρονοσειράς τη χρονική στιγμή  $i$  και  $a_i$  η τιμή που παράγει το μοντέλο τη χρονική στιγμή  $i$ .

### Μέσο Τετραγωνικό Σφάλμα (Mean Squared Error, MSE):

Το MSE είναι η μέση τιμή του τετραγώνου της διαφοράς των τιμών που παράγει το μοντέλο από τις πραγματικές τιμές της χρονοσειράς, κάθε χρονική στιγμή

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - a_i)^2$$

όπου  $y_i$  είναι οι πραγματική τιμή της χρονοσειράς τη χρονική στιγμή  $i$  και  $a_i$  η τιμή που παράγει το μοντέλο τη χρονική στιγμή  $i$ .

Συνήθως χρησιμοποιείται και η τετραγωνική ρίζα του μέσου τετραγωνικού σφάλματος (Root Mean Squared Error, RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - a_i)^2}$$

Εξαιτίας του μεγάλου όγκου δεδομένων που παράγονται κάθε μέρα, αποτελεί πλέον αναγκαία η χρήση νευρωνικών δικτύων για την ανάλυση των χρονοσειρών και την δημιουργία προβλέψεων. Αυτό οφείλεται στη δυνατότητα των νευρωνικών δικτύων να ανακαλύπτουν μοτίβα μέσα σε μεγάλα σύνολα δεδομένων, χωρίς να μας απασχολεί ο χαρακτηρισμός τους ως στάσιμες ή όχι.

## Κεφάλαιο 4: Πειραματικό Μέρος και Αποτελέσματα

Σε αυτό το κεφάλαιο περιγράφουμε το πλαίσιο του πειραματισμού και των αντίστοιχων αποτελεσμάτων της παρούσας εργασίας. Αρχικά παρουσιάζουμε τα δεδομένα που θα χρησιμοποιήσουμε για τον πειραματισμό τα οποία προέρχονται από το χώρο του χρηματιστηρίου, τις αρχιτεκτονικές των νευρωνικών δικτύων και τα αποτελέσματά μας.

## 4.1 Λογισμικό

Καθ' όλη τη διάρκεια του πειραματισμού, χρησιμοποιήσαμε τη γλώσσα προγραμματισμού Python [13]. Η Python είναι μια δυναμική γλώσσα αντικειμενοστραφούς προγραμματισμού. Η ευρεία χρήση της στις μέρες μας την καθιστά ιδιαίτερα δημοφιλή, με αποτέλεσμα να έχουν δημιουργηθεί γύρω της διάφορα APIs και βιβλιοθήκες, που στόχο έχουν τη περάτωση διάφορων εφαρμογών. Η ευκολία στη χρήση της την καθιστά ιδιαίτερα χρήσιμη στην Επιστήμη των Δεδομένων, τόσο αναφορικά με τη διαχείρισή τους, όσο κι αναφορικά με τη δημιουργία μοντέλων Μηχανικής Μάθησης.

Πιο συγκεκριμένα για τη διαχείριση/οπτικοποίηση των δεδομένων μας χρησιμοποιούμε τις βιβλιοθήκες NumPy, Pandas και matplotlib, ενώ για τη δημιουργία νευρωνικών δικτύων και την αξιολόγησή τους το Keras και την TensorFlow [14]. Η TensorFlow κάνει χρήση του Keras για την εύκολη δημιουργία ακόμα και των πιο σύνθετων αρχιτεκτονικών νευρωνικών δικτύων.

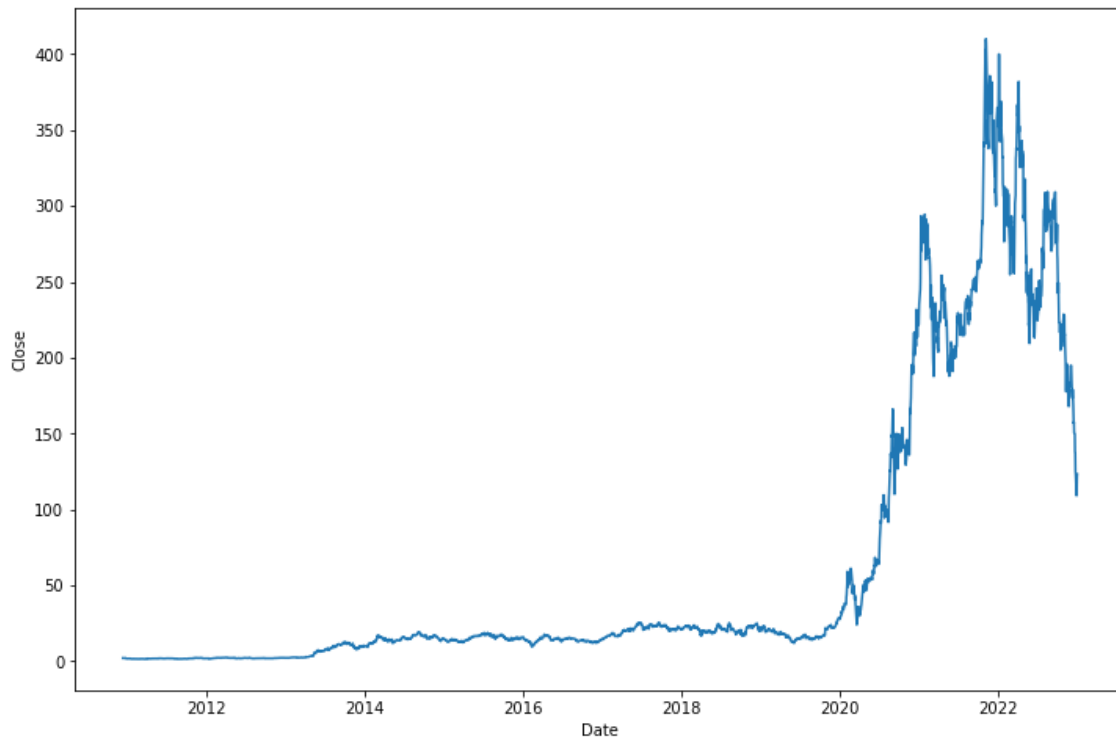
## 4.2 Δεδομένα

Τα δεδομένα που θα χρησιμοποιήσουμε τα αντλούμε απ' την ιστοσελίδα [www.finance.yahoo.com](http://www.finance.yahoo.com). Οι μετοχές με τις οποίες θα ασχοληθούμε αφορούν τις εταιρείες Tesla , Toyota , Mercedes και την BMW στο χρονικό διάστημα 10/12/2010-30/12/2022. Για κάθε σύνολο δεδομένων μετοχών, έχουμε πληροφορία για την ημερομηνία (Date), την αξία της μετοχής στο άνοιγμα των αγορών (Open), τη μέγιστη (High) αξία της μετοχής εντός της ημέρας, την ελάχιστη (Low) αξία της μετοχής εντός της ημέρας, την αξία της μετοχής στο κλείσιμο της ημέρας (Close), την τιμή κλεισίματος μετά τις προσαρμογές για όλα τα ισχύοντα splits και τις διανομές μερισμάτων (Adjusted Close) και την τιμή του πλήθους των assets (περιουσιακών στοιχείων) ή ασφαλειών που άλλαξαν χέρια εντός της ημέρας (Volume). Για τη μετοχή της Tesla έχουμε στη διάθεσή μας 3.035 εγγραφές, τη μετοχή της Toyota 3.035 εγγραφές, για τη μετοχή της Mercedes 3.060 εγγραφές και τη μετοχή της BMW 3.059 εγγραφές, οι οποίες αφορούν τις ημέρες κατά τις οποίες οι αντίστοιχες αγορές ήταν σε λειτουργία (καθημερινά πλην Σαββατοκύριακων κι επίσημων αργιών). Από όλα τα διαθέσιμα δεδομένα που έχουμε, θα ασχοληθούμε με την τιμή κλεισίματος

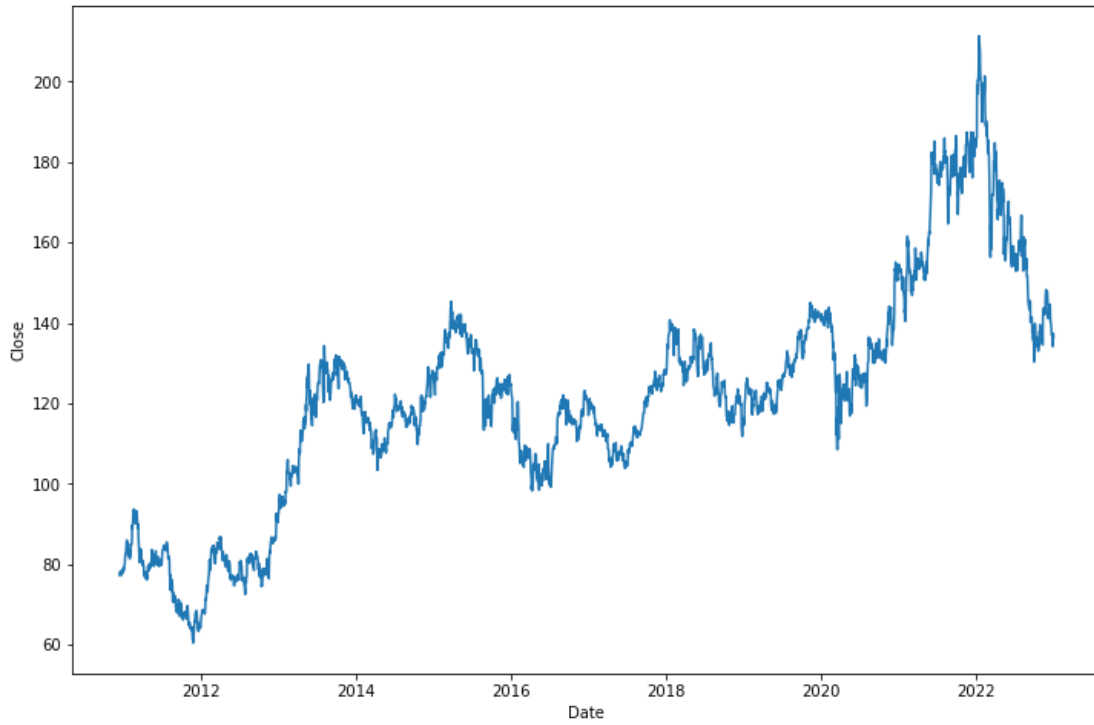
της μέρας (Close), για όλες τις διαθέσιμες μετοχές.

Αρχικά ελέγχουμε αν υπάρχουν τιμές οι οποίες λείπουν απ' τα δεδομένα που μας ενδιαφέρουν, δηλαδή τις ημερομηνίες και τις τιμές των μετοχών, παρατηρώντας ότι για καμία μετοχή δεν έχουμε missing values.

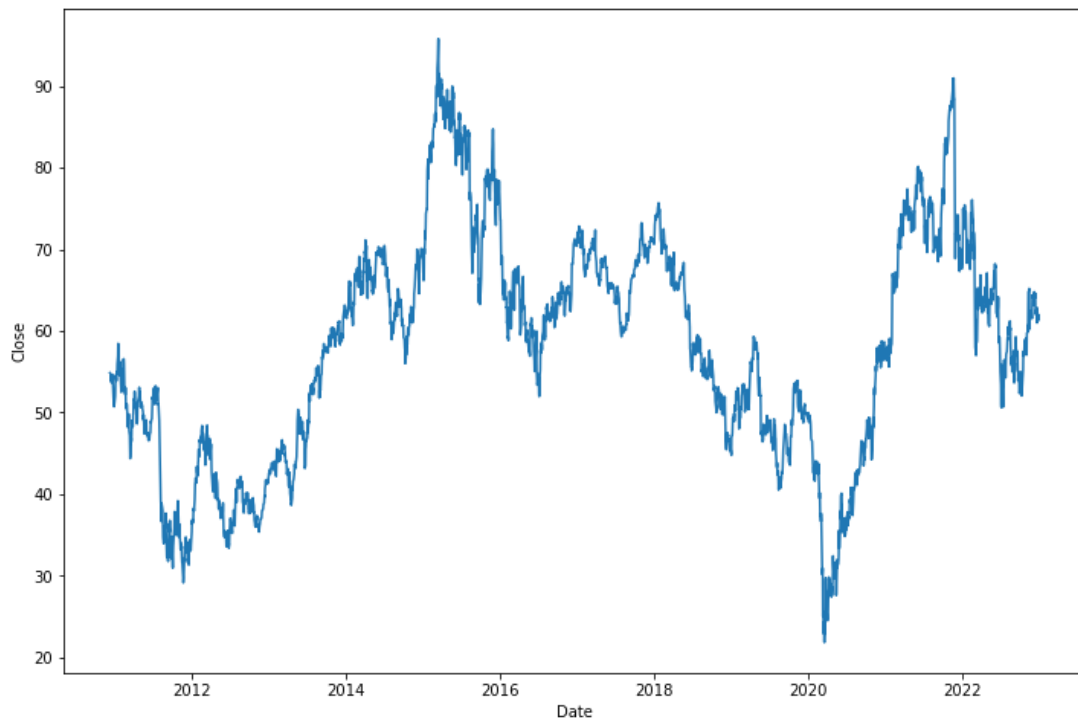
Παρακάτω δίνεται η οπτικοποίηση των καθαρισμένων χρονοσειρών των δεδομένων μας, ανά ημερήσια τιμή της μετοχής.



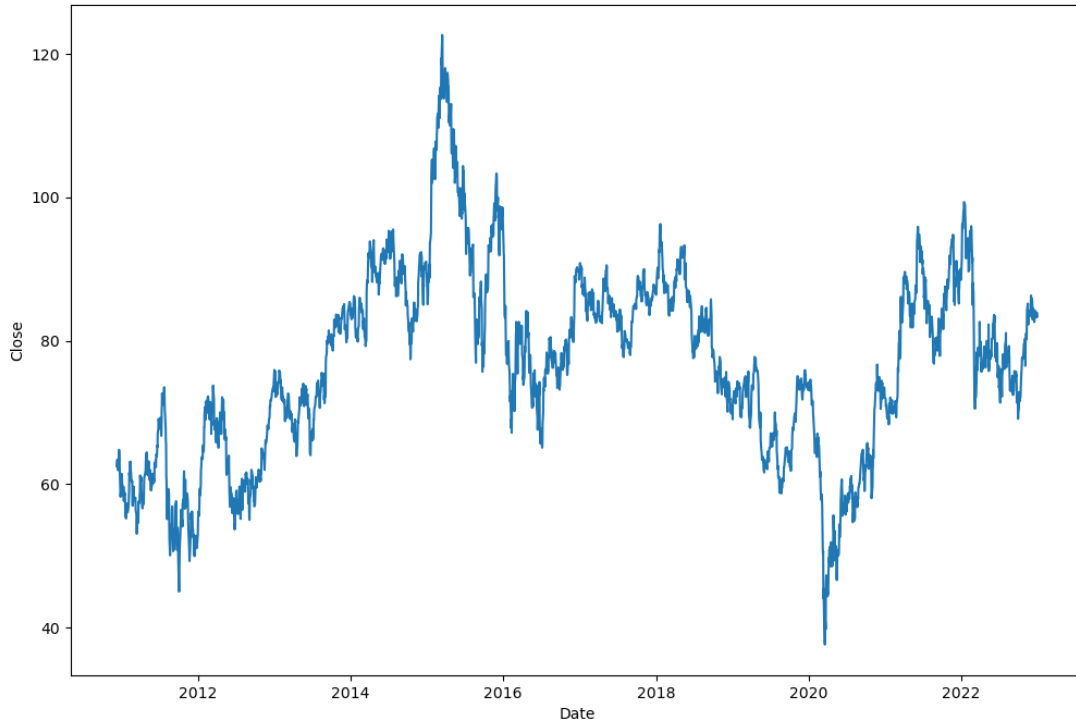
Μετοχή της Tesla



Μετοχή της Toyota



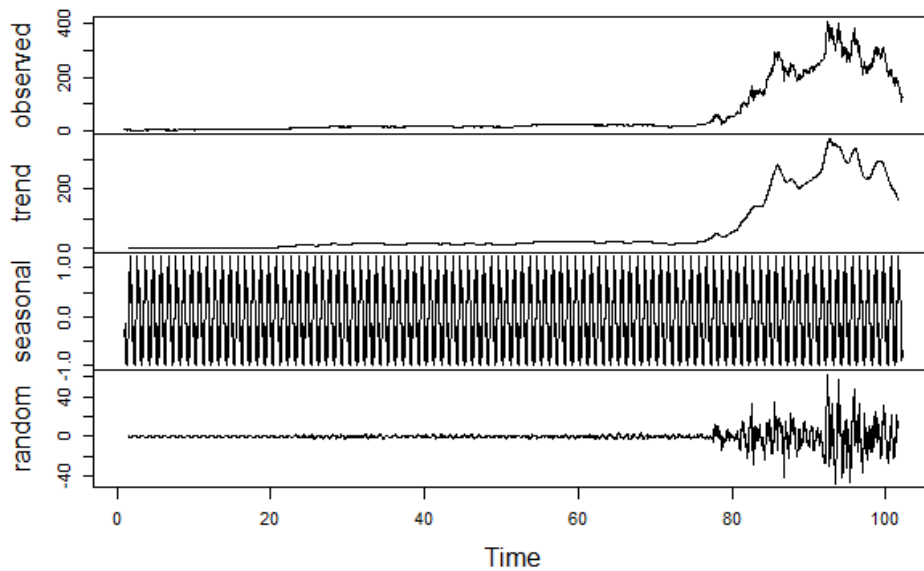
Μετοχή της Mercedes



Μετοχή της BMW

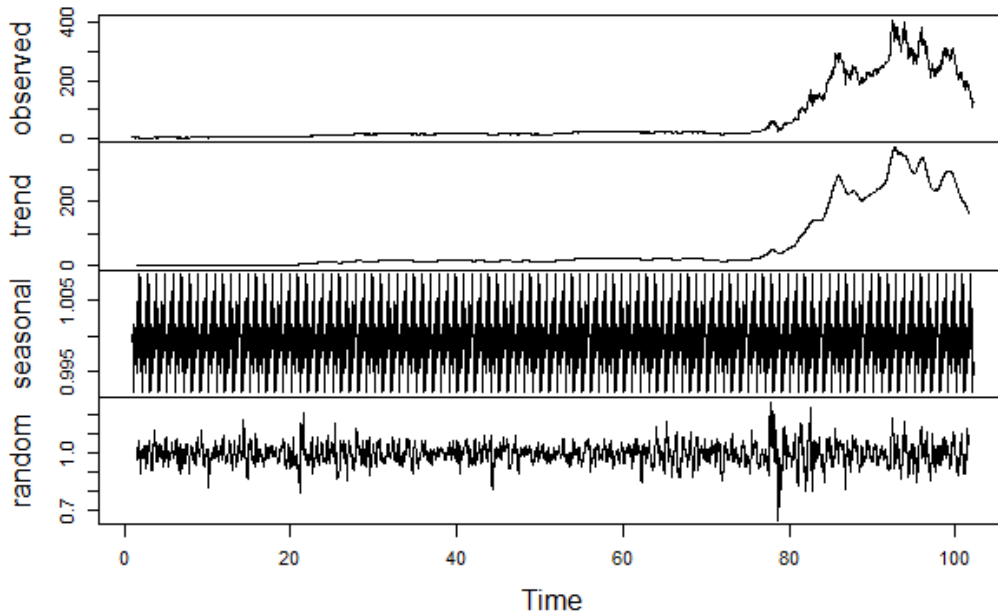
Παρακάτω παρουσιάζουμε τις αποσυνθέσεις των παραπάνω χρονοσειρών

**Decomposition of additive time series**



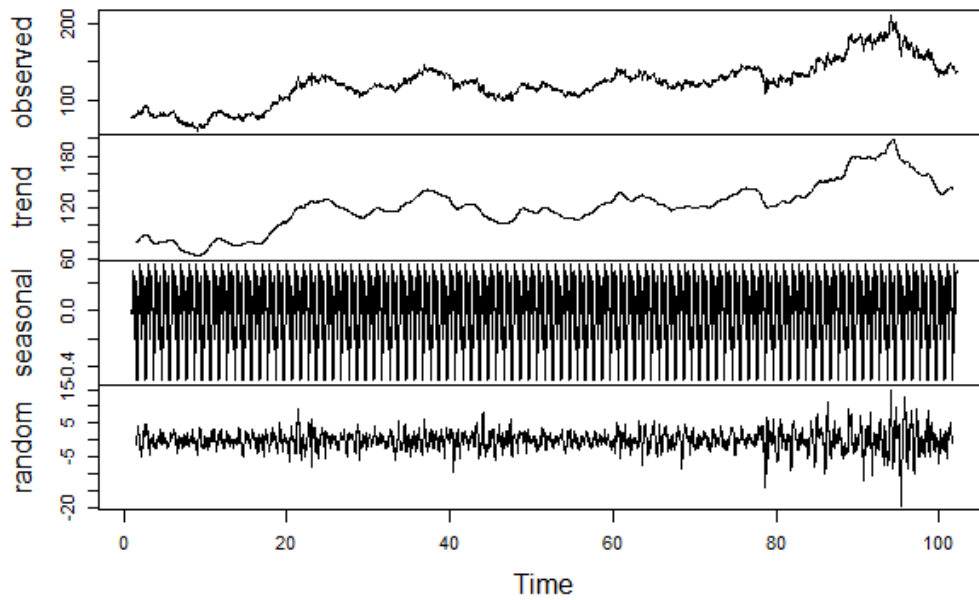
Αθροιστική αποσύνθεση της χρονοσειράς της Tesla

### Decomposition of multiplicative time series



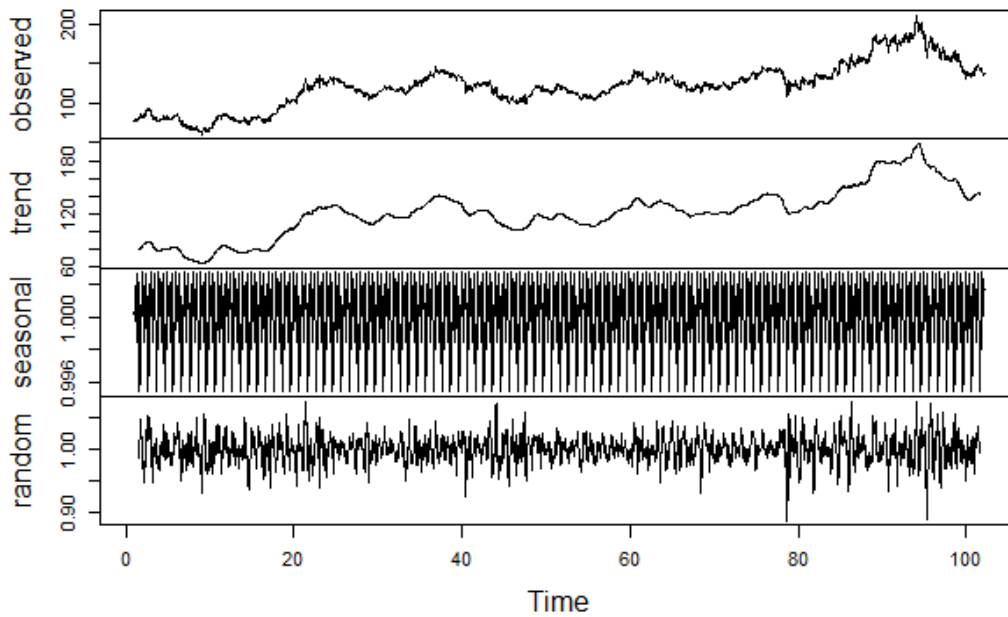
Πολλαπλασιαστική αποσύνθεση της χρονοσειράς της Tesla

### Decomposition of additive time series



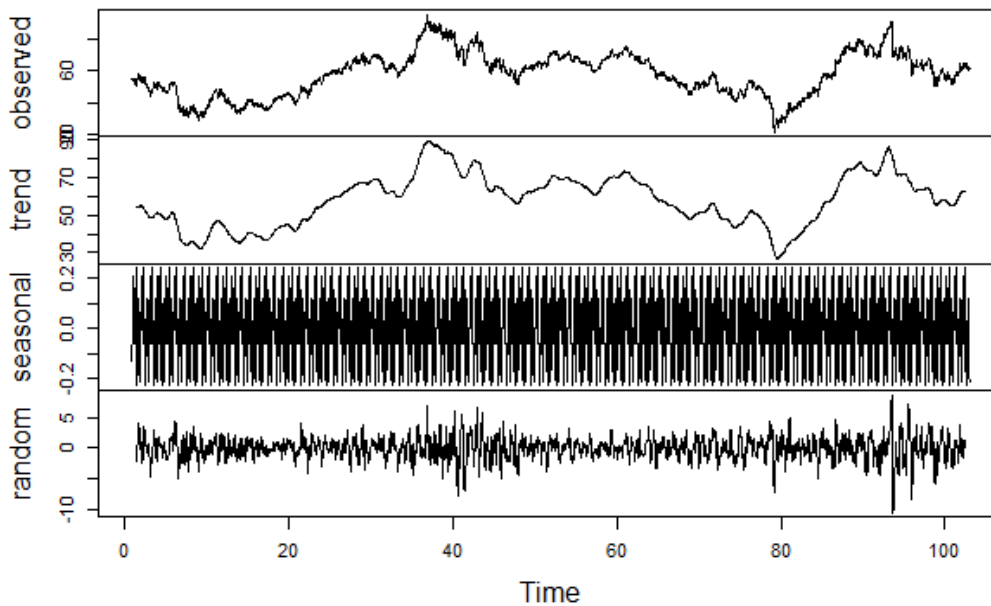
Αθροιστική αποσύνθεση της χρονοσειράς της Toyota

### Decomposition of multiplicative time series



Πολλαπλασιαστική αποσύνθεση της χρονοσειράς της Toyota

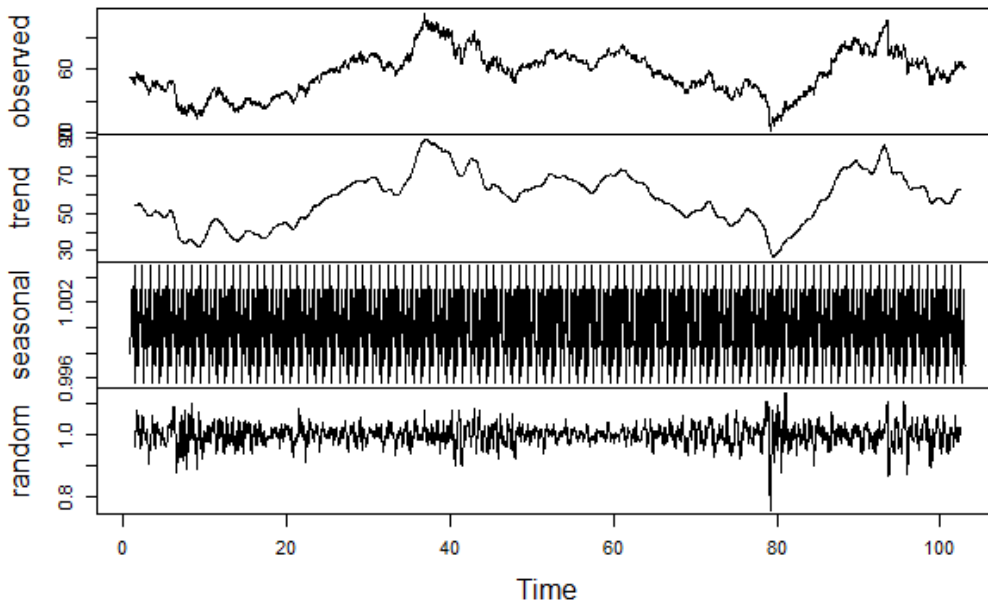
### Decomposition of additive time series



Αθροιστική αποσύνθεση της χρονοσειράς της Mercedes

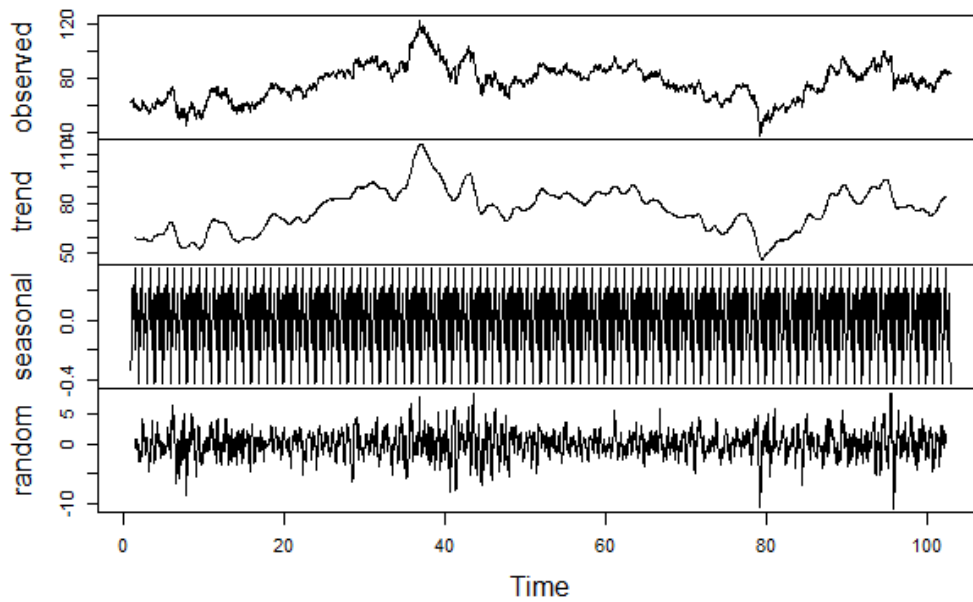


### Decomposition of multiplicative time series

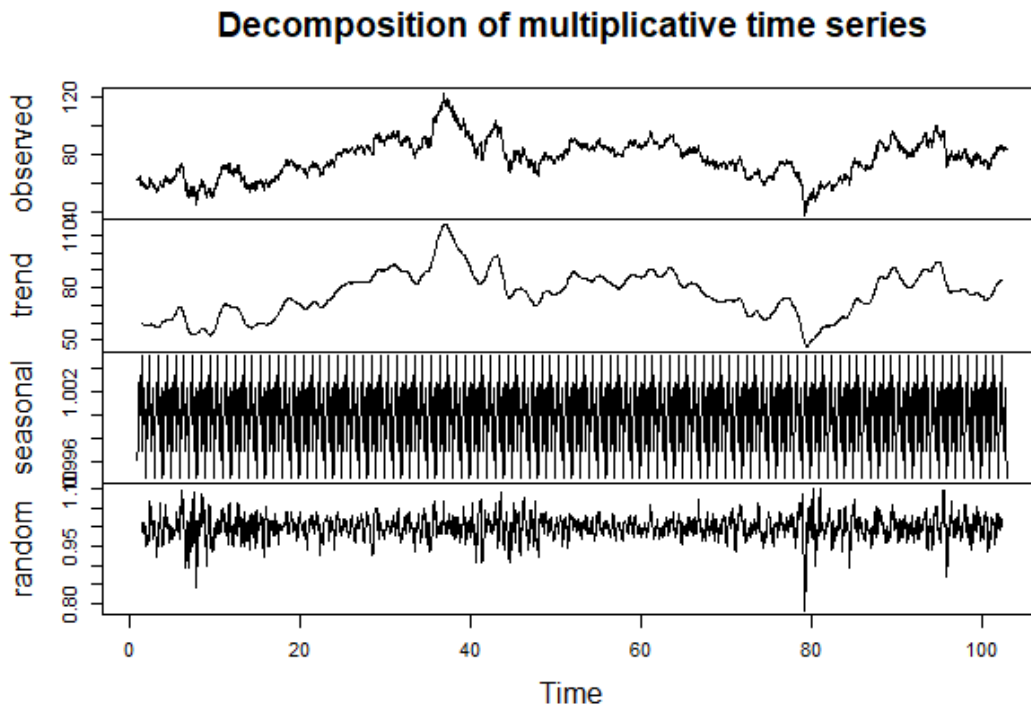


Πολλαπλασιαστική αποσύνθεση της χρονοσειράς της Mercedes

### Decomposition of additive time series



Αθροιστική αποσύνθεση της χρονοσειράς της BMW



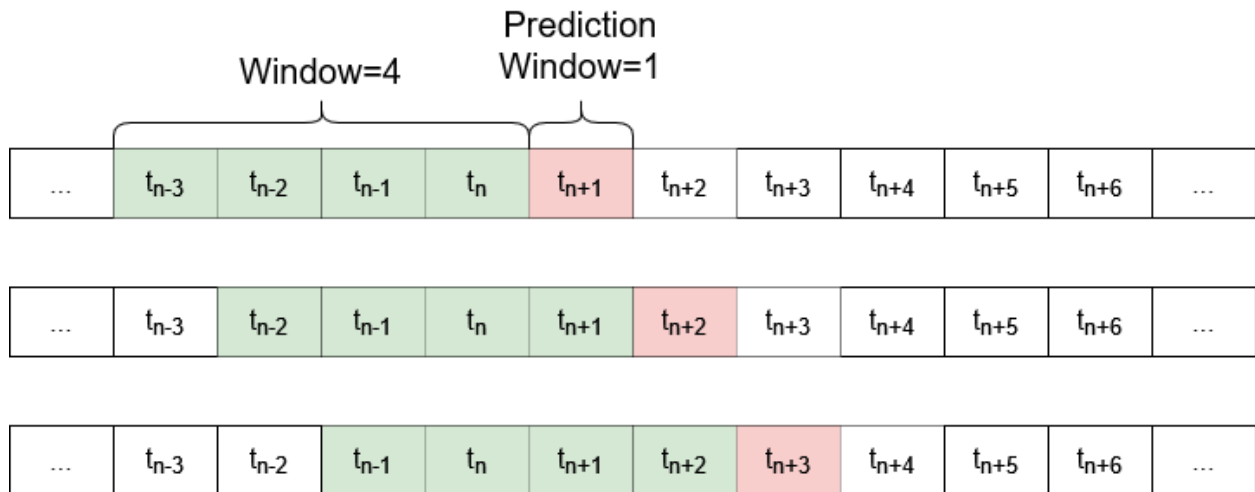
Πολλαπλασιαστική αποσύνθεση της χρονοσειράς της BMW

### 4.3 Μετασχηματισμός των Δεδομένων

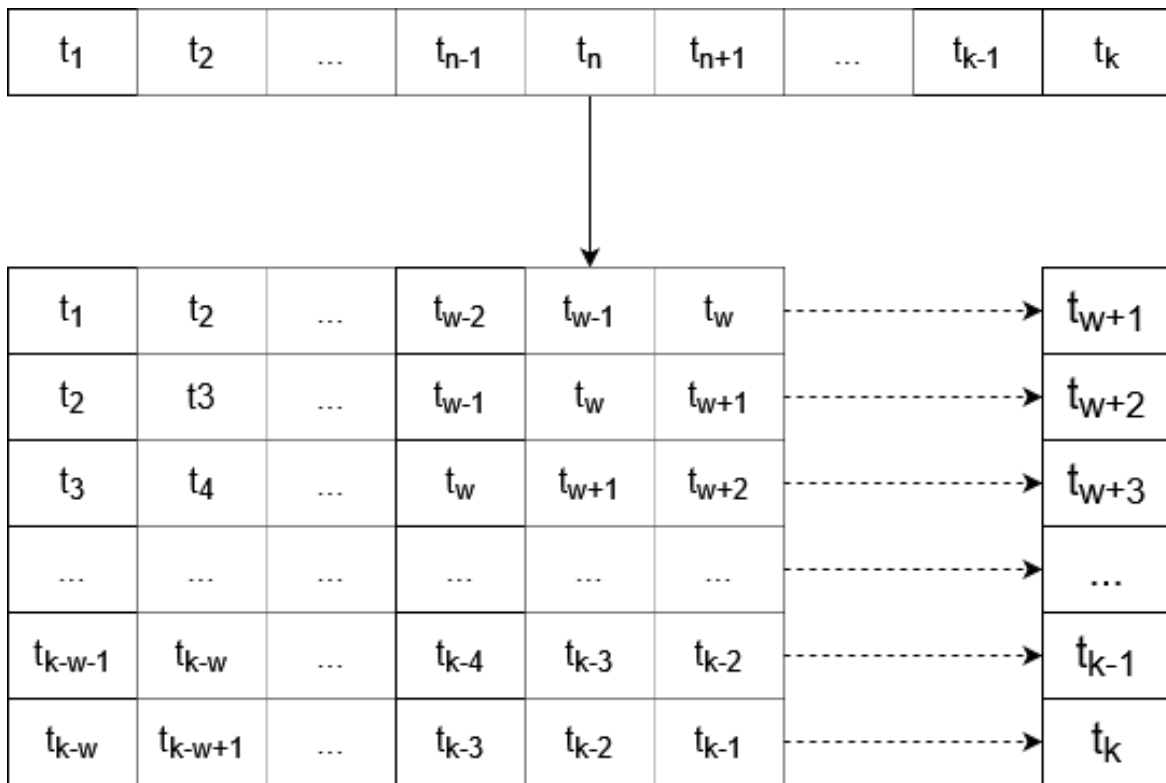
Για να μπορέσουμε να χρησιμοποιήσουμε τις χρονοσειρές μας ούτως ώστε να εκπαιδεύσουμε τα νευρωνικά μας δίκτυα, πρέπει να τις μετασχηματίσουμε κατάλληλα. Αυτό σημαίνει ότι τα δεδομένα σε μορφή ακολουθίας, θα πρέπει να τα μετατρέψουμε σε κατάλληλα inputs και outputs για να εκπαιδεύσουμε τα νευρωνικά δίκτυα.

Αρχικά ορίζουμε έναν αριθμό, τον *window*, ο οποίος ορίζει το πλήθος των προηγούμενων δεδομένων για να προβλέψουμε τις επόμενες τιμές και έναν δεύτερο, τον *prediction window*, που ορίζει το πλήθος των επόμενων ημερών τις οποίες θέλουμε να προβλέψουμε.

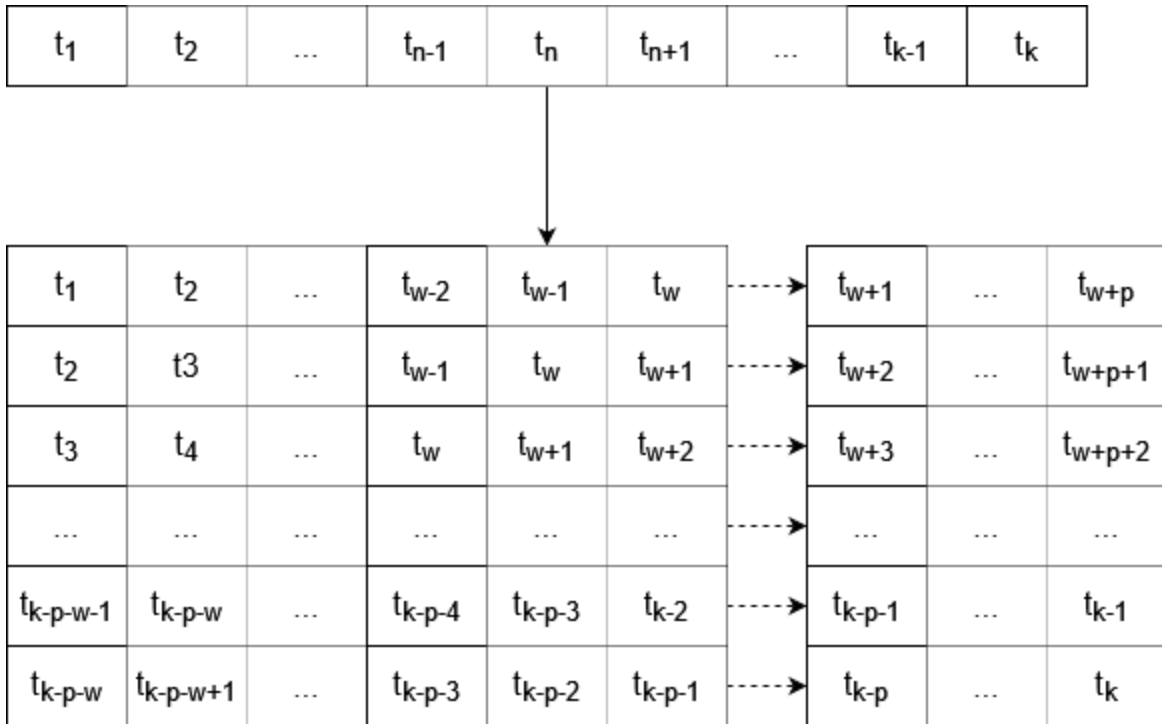
Για  $window=4$  και  $prediction\ window=1$ , έχουμε τον παρακάτω σχεδιασμό



Γενικότερα για window=w και prediction window=1, έχουμε



ενώ για window=w και prediction window=p, έχουμε



Τα window και prediction window που χρησιμοποιούμε, είναι πολύ σημαντικά για τον ορισμό των αρχιτεκτονικών.

Για να εκπαιδύσουμε σωστά τα δίκτυά μας, εφαρμόζουμε scaling στο διάστημα  $[0, 1]$  στα δεδομένα μας.

Για τη σωστή αξιολόγηση της απόδοσης των μοντέλων που θα χρησιμοποιήσουμε, χωρίζουμε για κάθε μετοχή ημερολογιακά σε δύο σύνολα: ένα για την εκπαίδευση των μοντέλων (train data) κρατώντας το 90% των πρώτων παρατηρήσεων κι ένα για την αξιολόγηση των παραχθέντων μοντέλων (test data), κρατώντας το τελευταίο 10% των παρατηρήσεων.

## 4.4 Πειραματικό σκέλος

### 4.4.1 Πρόβλεψη μετοχής επόμενης ημέρας

Για να προβλέψουμε τις τιμές των μετοχών θα χρησιμοποιήσουμε 3 είδη νευρωνικών δικτύων MLPs, LSTMs κι ένα CNNs.

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	704
dense_1 (Dense)	(None, 64)	4160
dense_2 (Dense)	(None, 64)	4160
dense_3 (Dense)	(None, 64)	4160
dense_4 (Dense)	(None, 1)	65

=====  
Total params: 13,249  
Trainable params: 13,249  
Non-trainable params: 0

### MLP

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 64)	16896
dense_40 (Dense)	(None, 64)	4160
dense_41 (Dense)	(None, 64)	4160
dense_42 (Dense)	(None, 64)	4160
dense_43 (Dense)	(None, 64)	4160
dense_44 (Dense)	(None, 1)	65

=====  
Total params: 33,601  
Trainable params: 33,601  
Non-trainable params: 0

### LSTM

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 8, 64)	256
conv1d_1 (Conv1D)	(None, 6, 64)	12352
average_pooling1d (AveragePooling1D)	(None, 3, 64)	0
flatten (Flatten)	(None, 192)	0
dense_80 (Dense)	(None, 64)	12352
dense_81 (Dense)	(None, 64)	4160
dense_82 (Dense)	(None, 64)	4160
dense_83 (Dense)	(None, 64)	4160
dense_84 (Dense)	(None, 1)	65

=====  
 Total params: 37,505  
 Trainable params: 37,505  
 Non-trainable params: 0

### CNN

Την εκπαίδευση των δικτύων την κάνουμε χρησιμοποιώντας ως loss function την MSE, 100 epochs και ως optimizer τον Adam.

Στη συνέχεια δίνουμε τα αποτελέσματα των RMSE και MAE στα δεδομένα εκτίμησης, χρησιμοποιώντας ως window τις τιμές 10 και 20, αντίστοιχα και ως prediction window την τιμή 1 για να κάνουμε πρόβλεψη για την τιμή κλεισίματος της μετοχής την επόμενη μέρα

Tesla	MLP	LSTM	CNN
Window=10	13,2247	13,5196	13,8596
Window=20	14,0483	12,3336	15,1014

### RMSE

Tesla	MLP	LSTM	CNN
Window=10	10,2557	10,4635	10,9938
Window=20	10,9295	9,2883	12,0876

### MAE

Toyota	MLP	LSTM	CNN
Window=10	2,8657	2,8883	3,0088
Window=20	2,9172	2,8139	3,0671

RMSE

Toyota	MLP	LSTM	CNN
Window=10	2,1618	2,2032	2,2913
Window=20	2,1966	2,1301	2,3253

MAE

Mercedes	MLP	LSTM	CNN
Window=10	1,7800	1,7701	1,8713
Window=20	1,7753	1,6774	1,7747

RMSE

Mercedes	MLP	LSTM	CNN
Window=10	1,2874	1,2823	1,2892
Window=20	1,2191	1,1355	1,1747

MAE

BMW	MLP	LSTM	CNN
Window=10	1,6409	1,6331	1,7588
Window=20	2,2247	1,9324	2,4925

RMSE

BMW	MLP	LSTM	CNN
Window=10	1,1994	1,1963	1,2992
Window=20	1,8699	1,5600	2,1328

MAE

Για όλες τις μετοχές παρατηρούμε ότι έχουμε καλύτερα αποτελέσματα χρησιμοποιώντας την LSTM αρχιτεκτονική με window=20, εκτός από την BMW που έχουμε καλύτερα αποτελέσματα για το LSTM με window=10. Αντίστοιχα, όλα τα CNN δίκτυα φαίνεται ότι δεν αποδίδουν καλά. Οι μεγάλες διαφορές στα σφάλματα οφείλονται στις διαφορετικές διακυμάνσεις των χρονοσειρών.

	Tesla	Toyota	Mercedes	BMW
Τυπική Απόκλιση	96,6539	27,6962	13,8273	13,1648

Παρακάτω δίνονται τα διαγράμματα πρόβλεψης επόμενης ημέρας για όλες τις μετοχές χρησιμοποιώντας το καλύτερο μοντέλο:

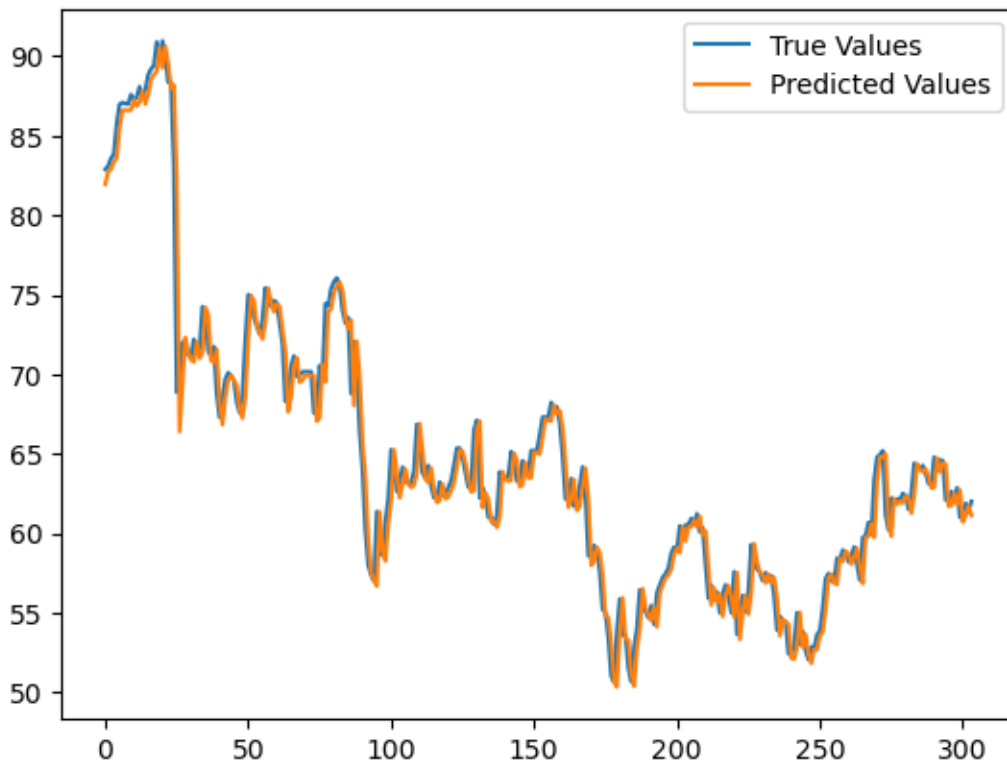


Μετοχή Tesla – LSTM (window=20)

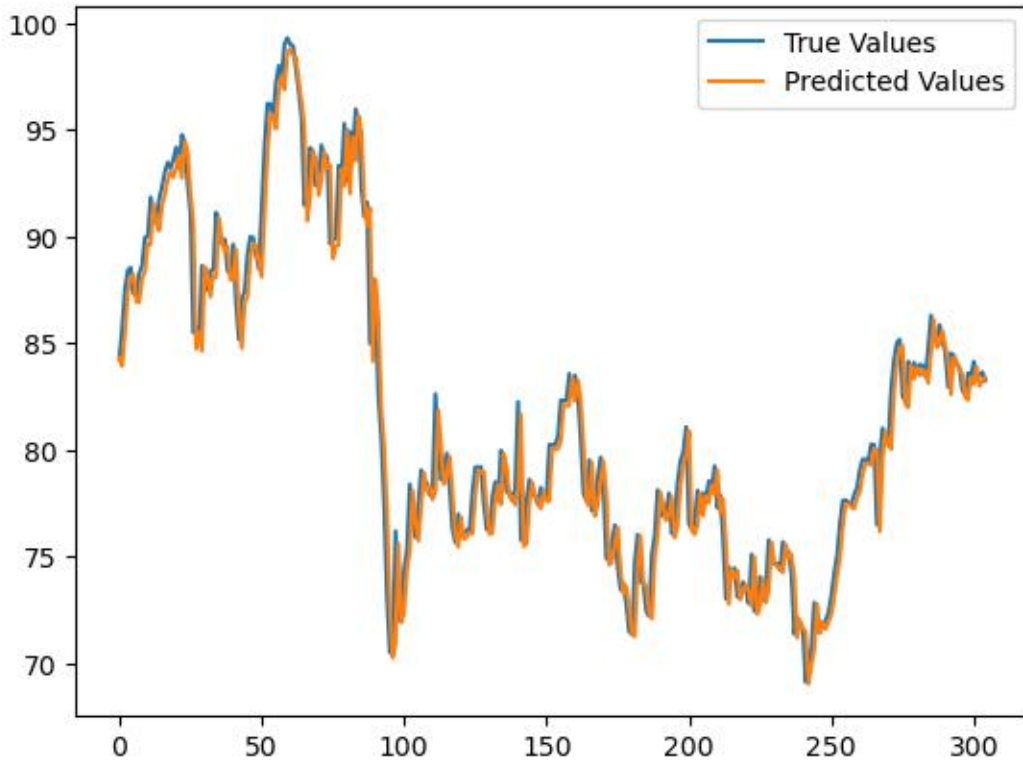




Μετοχή Toyota – LSTM (window=20)



Μετοχή Mercedes – LSTM (window=20)



Μετοχή BMW – LSTM (window=10)

Τέλος, κάνουμε πρόβλεψη για τις τιμές των μετοχών για την ημέρα 3/1/2023 για τις μετοχές των Tesla και Toyota και 2/1/2023 για τις μετοχές των Mercedes και BMW, που είναι οι πρώτες διαδοχικά ημέρες για τις οποίες δεν έχουμε χρησιμοποιήσει ούτε στα δεδομένα εκπαίδευσης, ούτε στα δεδομένα αξιολόγησης των μοντέλων.

Tesla	MLP	LSTM	CNN
Window=10	115,6963	118,5344	120,2200
Window=20	120,5374	116,3349	125,0966

Πραγματική τιμή Tesla 3/1/2023: 108,10

Toyota	MLP	LSTM	CNN
Window=10	134,4853	134,0766	134,1775
Window=20	134,8790	134,5364	135,2845

Πραγματική τιμή Toyota 3/1/2023: 138,28

Mercedes	MLP	LSTM	CNN
Window=10	60,7298	60,7466	60,7513
Window=20	61,1731	61,1061	61,2809

Πραγματική τιμή Mercedes 2/1/2023: 63,78

BMW	MLP	LSTM	CNN
Window=10	83,4641	83,3969	84,2035
Window=20	82,0226	82,5517	82,0094

Πραγματική τιμή BMW 2/1/2023: 85,80

Παρατηρούμε ότι κανένα από τα μοντέλα που προέκυψαν ως βέλτιστα για κάθε μετοχή δεν δίνει κοντινότερη τιμή στην πραγματική. Αυτό μπορεί να οφείλεται σε πολλούς παράγοντες, όπως για παράδειγμα στο ότι τα μοντέλα δεν εκπαιδεύτηκαν με δεδομένα ημερολογιακά κοντά σε αυτά της προβλεπόμενης τιμής, οι αγορές είναι ασταθείς και απρόβλεπτες κα.

## 4.4.2 Πρόβλεψη τάσης

Στη συνέχεια ασχολούμαστε με την πρόβλεψη τάσης.

Ορίζοντας ως βάθος πρόβλεψης για την τάση τις  $K$  παρατηρήσεις, χρησιμοποιώντας τα νευρωνικά δίκτυα που χρησιμοποιήθηκαν προηγουμένως, προσπαθούμε να λύσουμε το πρόβλημα της ταξινόμησης αναφορικά με την τάση. Αυτό σημαίνει ότι ορίζουμε την τάση ως εξής

$$\text{Τάση} = \text{Τιμή κλεισίματος σε } K \text{ ημέρες} - \text{Τιμή κλεισίματος σήμερα}$$

Για κάθε διάνυσμα εισόδου, σύμφωνα με την τάση, μπορούμε να κατηγοριοποιήσουμε το αντίστοιχο διάνυσμα σε 2 κλάσεις, μία για τη θετική τιμή της (αυξητική τάση) και μία για την αρνητική τιμή της (πτωτική τάση).

Τα αποτελέσματα που παίρνουμε δίνονται στους παρακάτω πίνακες:

Tesla	MLP	LSTM	CNN
Window=10	49,50%	49,50%	50,50%
Window=20	49,67%	50,66%	49,34%

Toyota	MLP	LSTM	CNN
Window=10	47,52%	52,48%	47,52%
Window=20	47,35%	52,65%	47,35%

Mercedes	MLP	LSTM	CNN
Window=10	53,44%	52,46%	52,46%
Window=20	50,00%	52,63%	47,04%

BMW	MLP	LSTM	CNN
Window=10	50,82%	50,49%	52,13%
Window=20	50,33%	50,33%	50,33%

Πρόβλεψη Τάσης για K=1

Tesla	MLP	LSTM	CNN
Window=10	50,00%	50,00%	50,00%
Window=20	51,50%	50,17%	49,83%

Toyota	MLP	LSTM	CNN
Window=10	47,35%	52,65%	47,35%
Window=20	47,51%	52,49%	47,51%

Mercedes	MLP	LSTM	CNN
Window=10	52,13%	53,11%	53,11%
Window=20	53,95%	52,96%	52,96%

BMW	MLP	LSTM	CNN
Window=10	53,77%	50,82%	51,48%
Window=20	52,96%	50,99%	50,99%

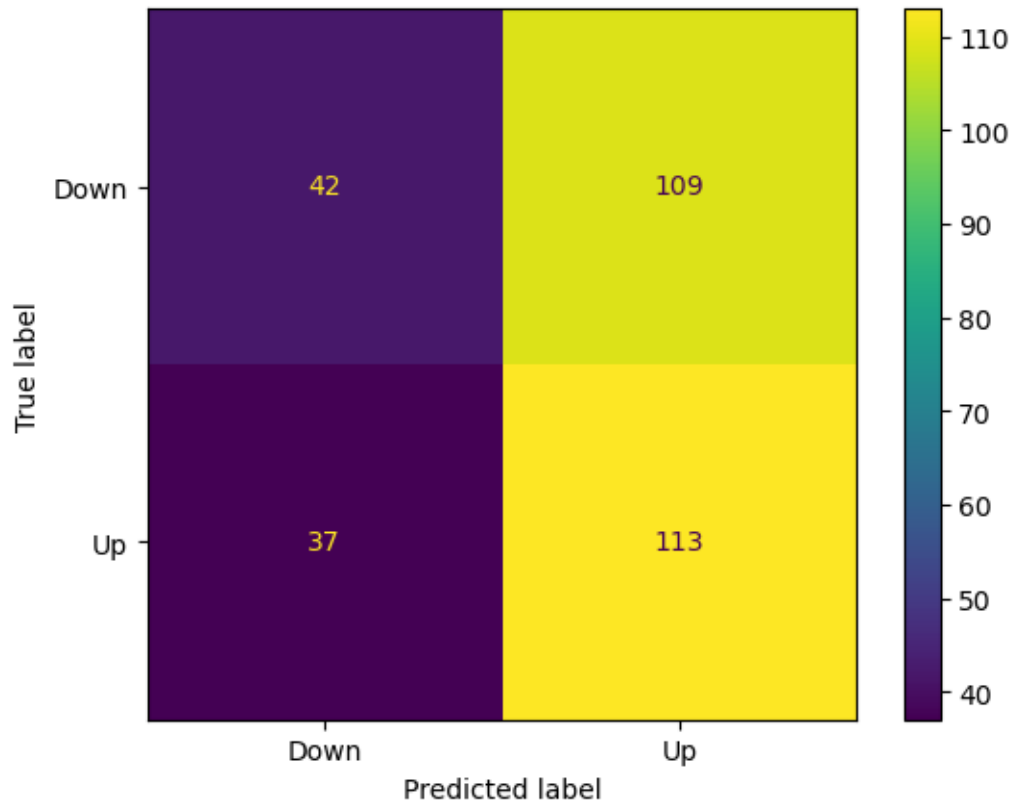
### Πρόβλεψη Τάσης για K=5

Για K=1 καλύτερα αποτελέσματα για την Tesla και την Toyota έχουμε για τον LSTM με window=20, για την Mercedes για τον MLP με window=10 και για την BMW για τον CNN με window=10.

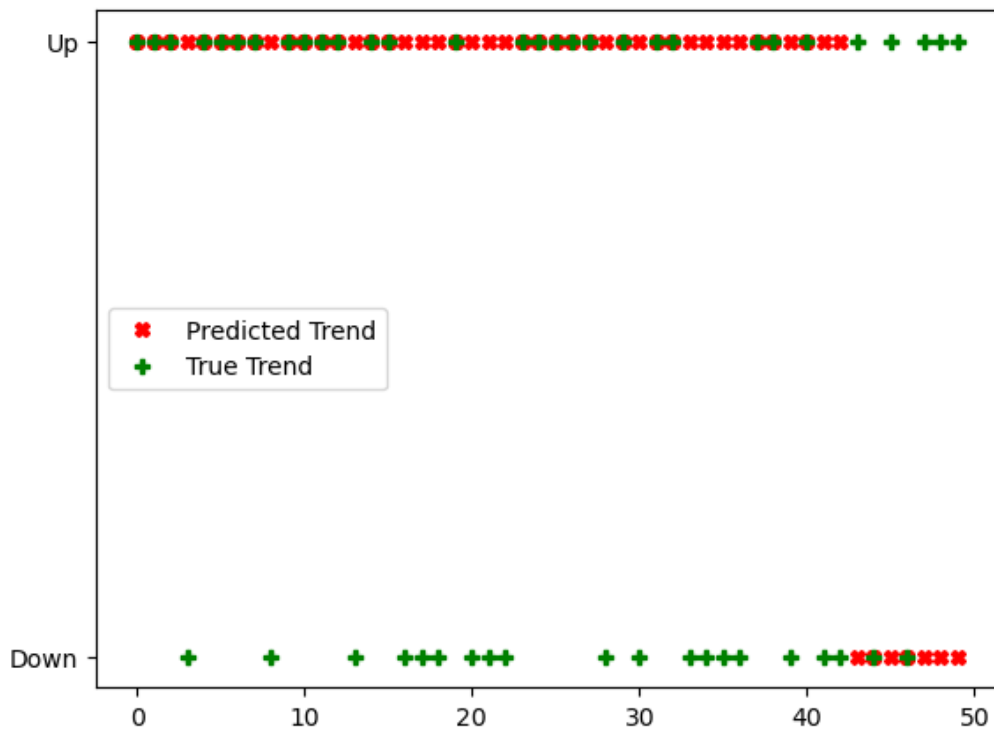
Για K=5 καλύτερα αποτελέσματα για την Tesla έχουμε για τον MLP με window=20, για την Toyota για το LSTM με window=10, για την Mercedes για τον MLP με window=20 και για την BMW για τον MLP με window=10.

Τα ποσοστά ακρίβειας που βρίσκουμε δεν θεωρούνται ικανοποιητικά για κανένα από τα δύο προβλήματα πρόβλεψης τάσης που εξετάσαμε, αλλά αυτό είναι το αποτέλεσμα της δυσκολίας του προβλήματος.

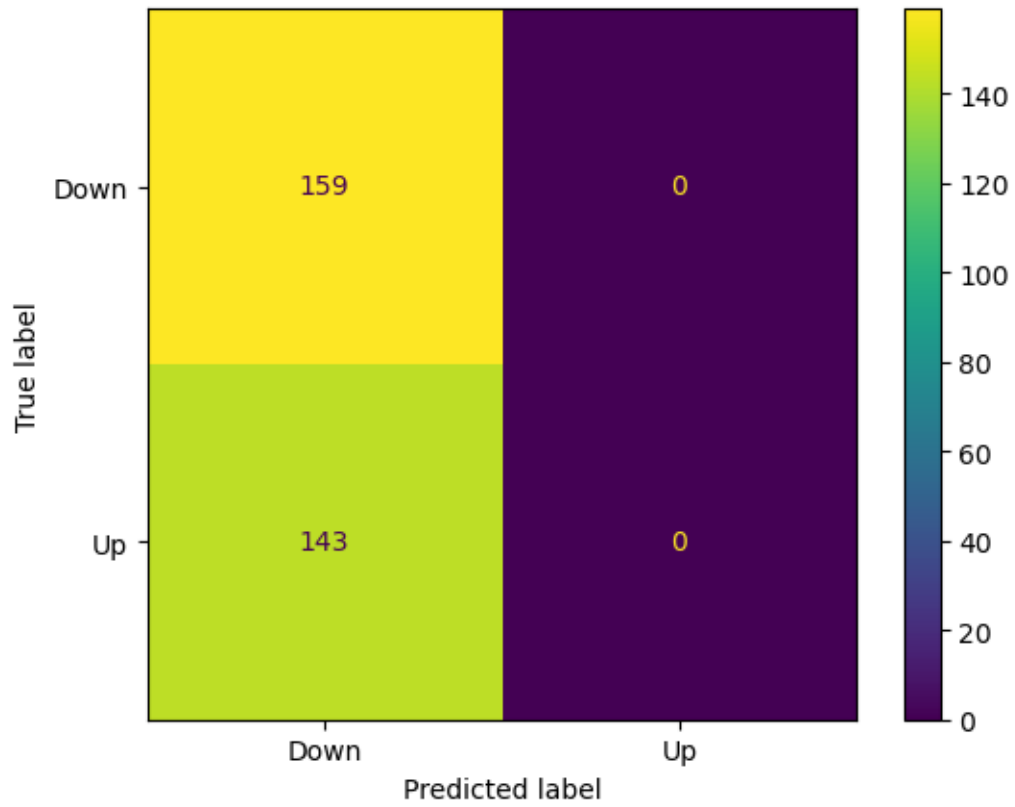
Παρακάτω δίνονται τα confusion matrix των αποτελεσμάτων ταξινόμησης για το μοντέλο καλύτερης επίδοσης κάθε μετοχής, με τις αντίστοιχες προβλέψεις των πρώτων 50 διανυσμάτων εισόδου στο testing σύνολο δεδομένων για K=5:



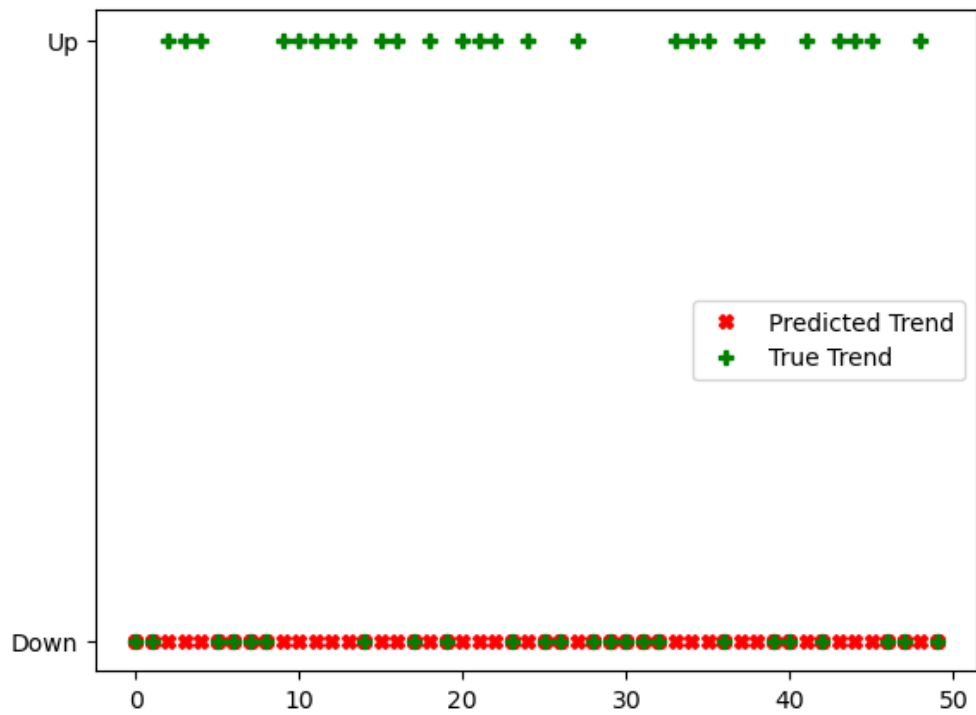
Tesla – MLP (window=20)



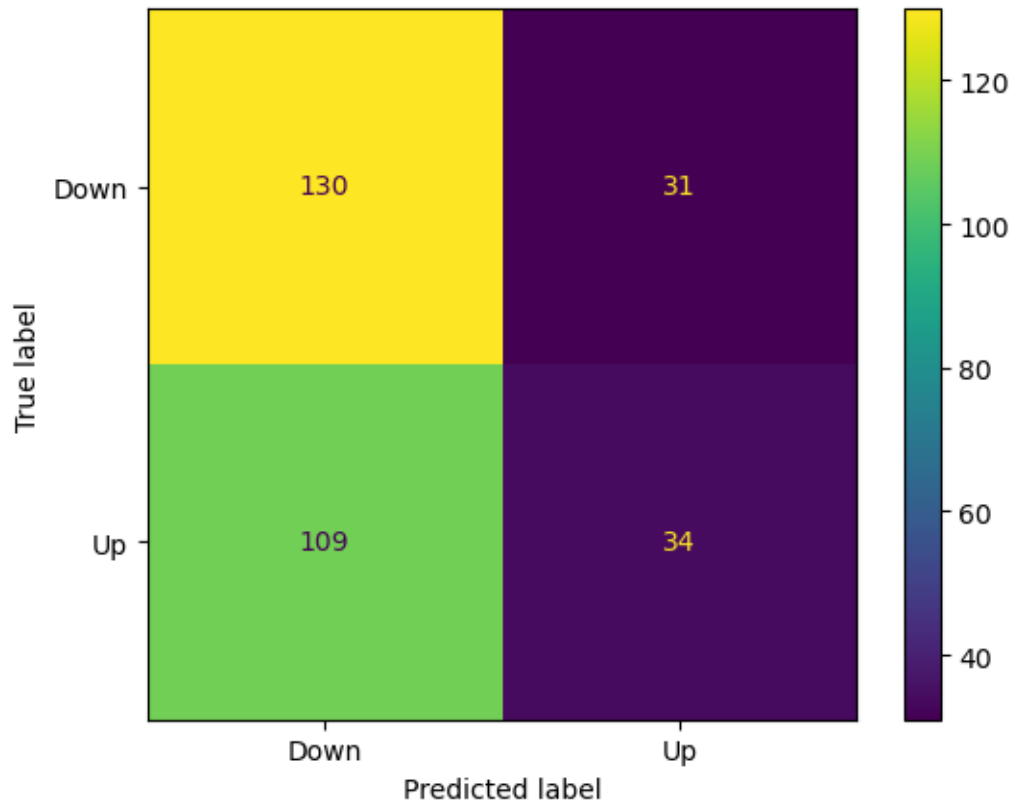
Tesla – MLP (window=20), πρόβλεψη 50 πρώτων ημερών



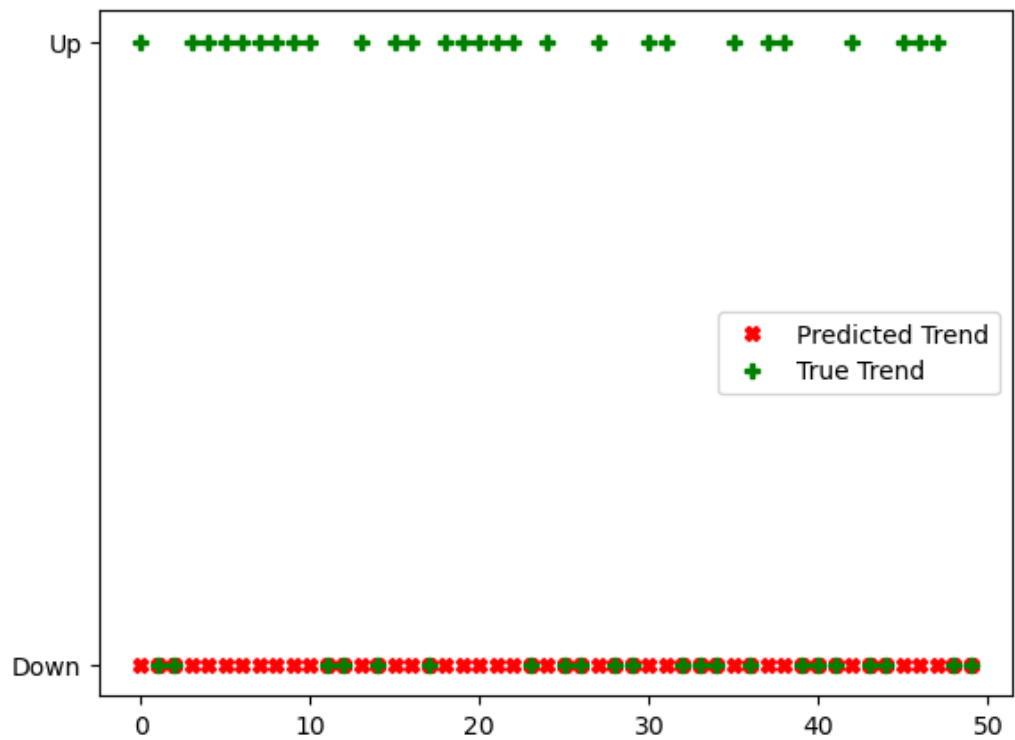
Toyota – LSTM (window=10)



Toyota – LSTM (window=10), πρόβλεψη 50 πρώτων ημερών

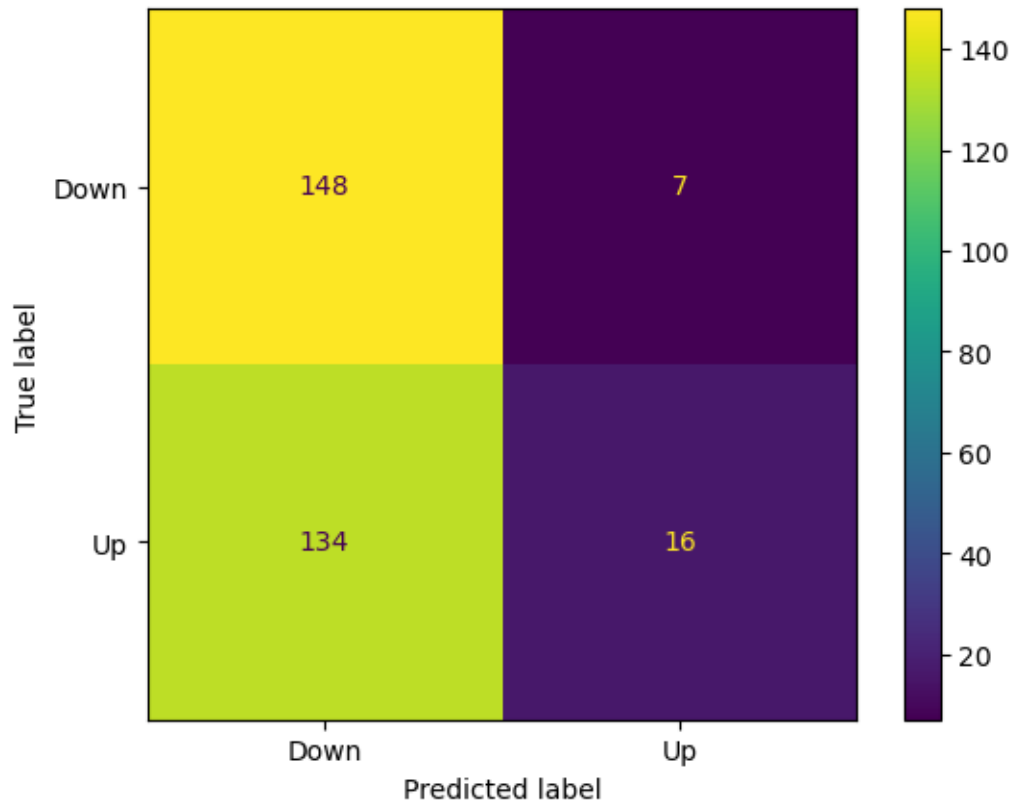


Mercedes – MLP (window=20)

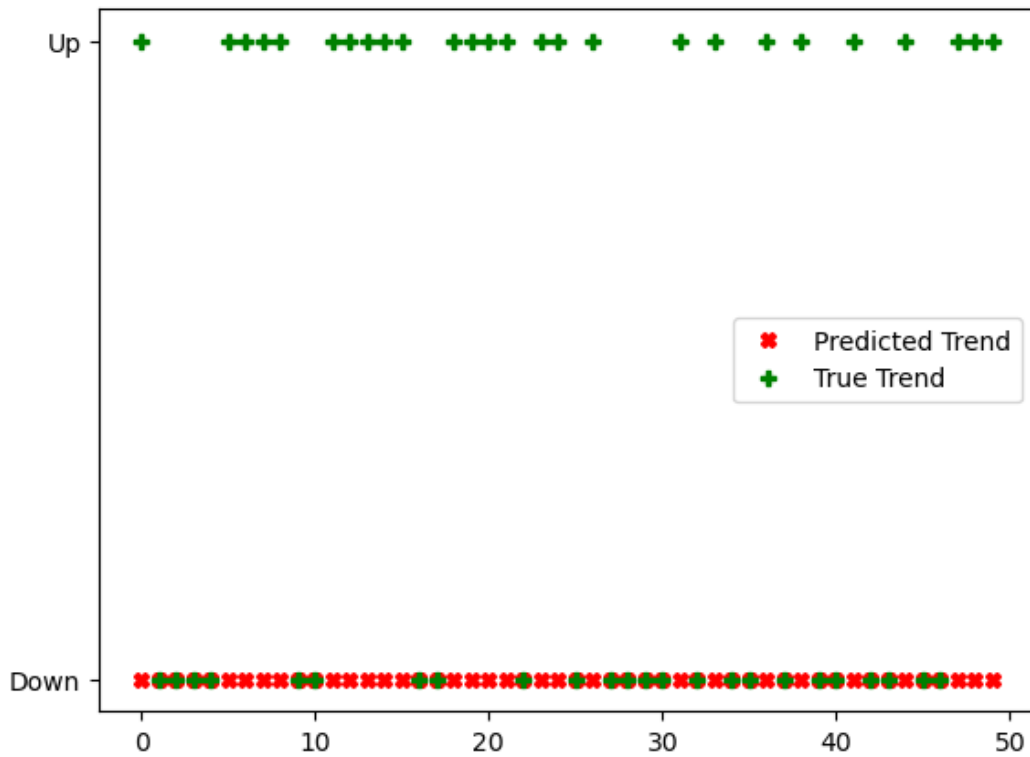


Mercedes – LSTM (window=10), πρόβλεψη 50 πρώτων ημερών



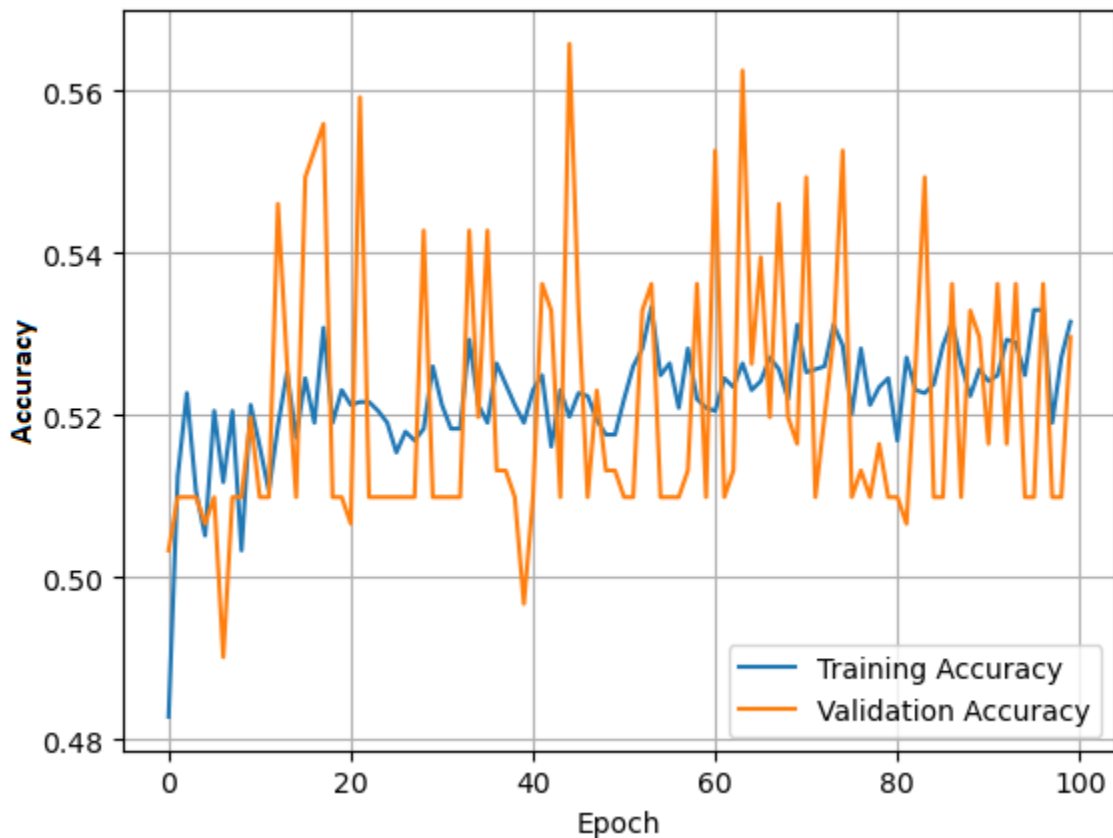


BMW – MLP (window=20)



BMW – MLP (window=20), πρόβλεψη 50 πρώτων ημερών

Κατά τον πειραματισμό παρατηρήθηκε ότι στις περισσότερες περιπτώσεις ο ταξινομητής μας αντιστοιχεί κάθε input σε μια συγκεκριμένη κλάση. Αυτό καθιστά τα μοντέλα άνευ ουσίας. Τα αποτελέσματα των μοντέλων που δίνονται παραπάνω, είναι τα καλύτερα κι αυτό οφείλεται στη δυνατότητά τους να μην παράγουν σταθερή κλάση ως σύσταση (με εξαίρεση αυτό της Toyota και της Mercedes). Αναλύοντας τα ιστορικά ακρίβειας ταξινόμησης ανά epoch, παρατηρούμε ότι δεν υπάρχει αισθητή βελτίωση σε κανένα από αυτά. Χαρακτηριστικό παράδειγμα είναι το παρακάτω που αναφέρεται στην ακρίβεια ανά epoch για τη μετοχή της BMW εκπαιδύοντας τον MLP με window=20.



Απ' το διάγραμμα γίνεται σαφές ότι δεν υπάρχει τάση σύγκλισης σε κάποια βέλτιστη τιμή για την ακρίβεια ταξινόμησης κι ότι οι τιμές που παίρνουμε έχουν μεγάλη διακύμανση με μέση τιμή γύρω απ' το 52%, υποδεικνύοντάς μας ότι υπάρχει τυχαιότητα στις μετρήσεις.

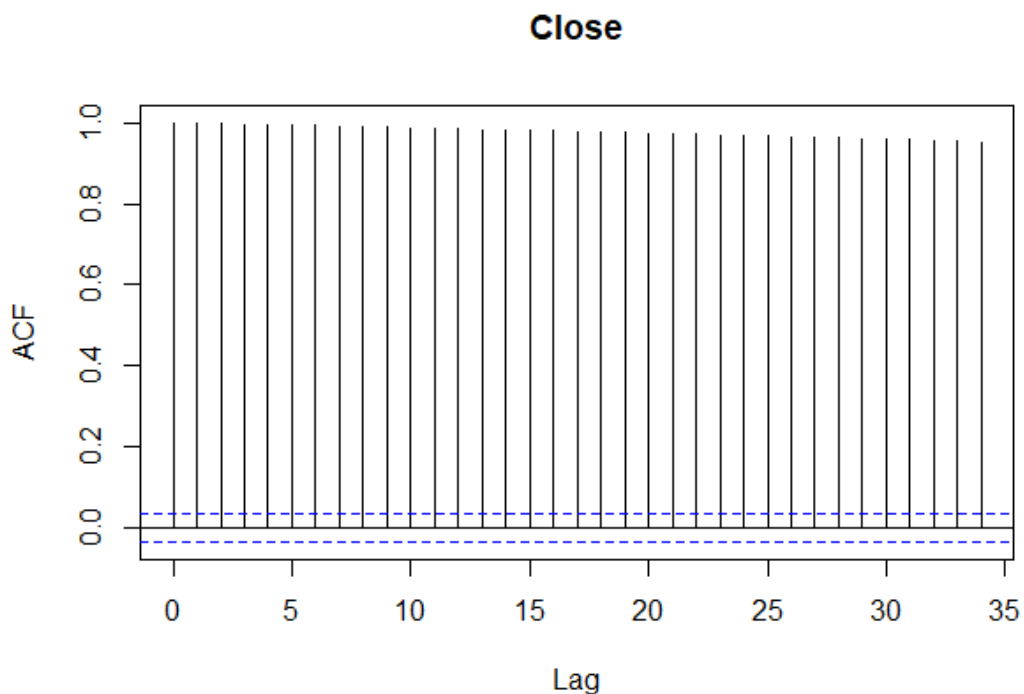
Αυτό το γεγονός αποτελεί και επισήμανση της δυσκολίας του προβλήματος που προσπαθούμε να λύσουμε.

### 4.4.3 Σύγκριση νευρωνικών δικτύων με κλασικές μεθόδους ανάλυσης χρονοσειρών

Τέλος ασχολούμαστε με τη σύγκριση των νευρωνικών δικτύων που σχεδιάσαμε με κλασικές μεθόδους ανάλυσης χρονοσειρών.

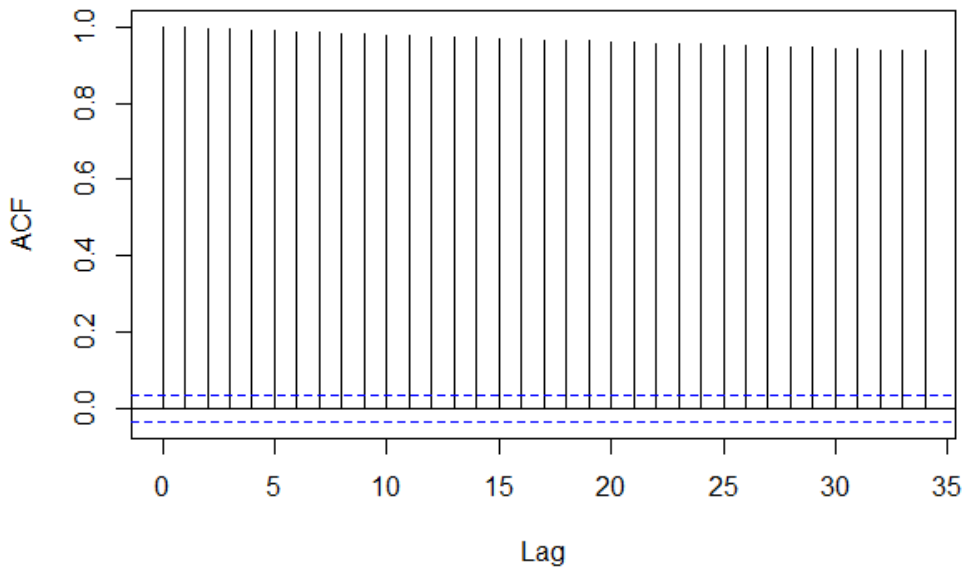
Για το λόγο αυτό χρησιμοποιούμε τη γλώσσα προγραμματισμού R, που είναι ιδανική για ανάλυση χρονοσειρών. Πιο συγκεκριμένα χρησιμοποιώντας κάποια πακέτα της R, εφαρμόζουμε ARIMA για να εκπαιδύσουμε τα αντίστοιχα μοντέλα και να κάνουμε πρόβλεψη για τις 10 τελευταίες παρατηρήσεις των χρονοσειρών μας.

Παρότι είναι προφανές ότι οι χρονοσειρές που εξετάζουμε δεν είναι στάσιμες επειδή έχουν τάση και εποχικότητα, εξετάζουμε επίσης και τη συνάρτηση αυτοσυσχέτισης (autocorrelation function) τους.



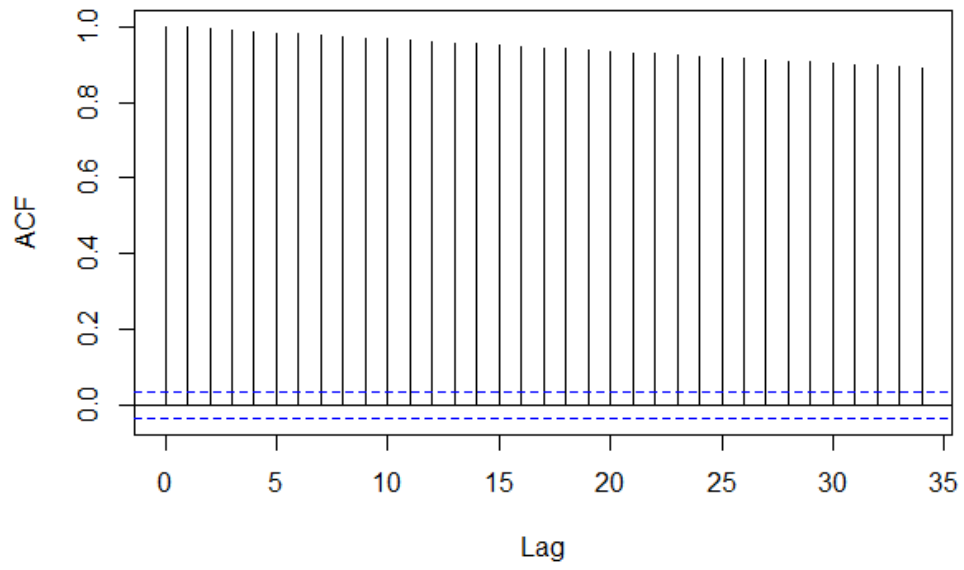
Tesla ACF

### Close

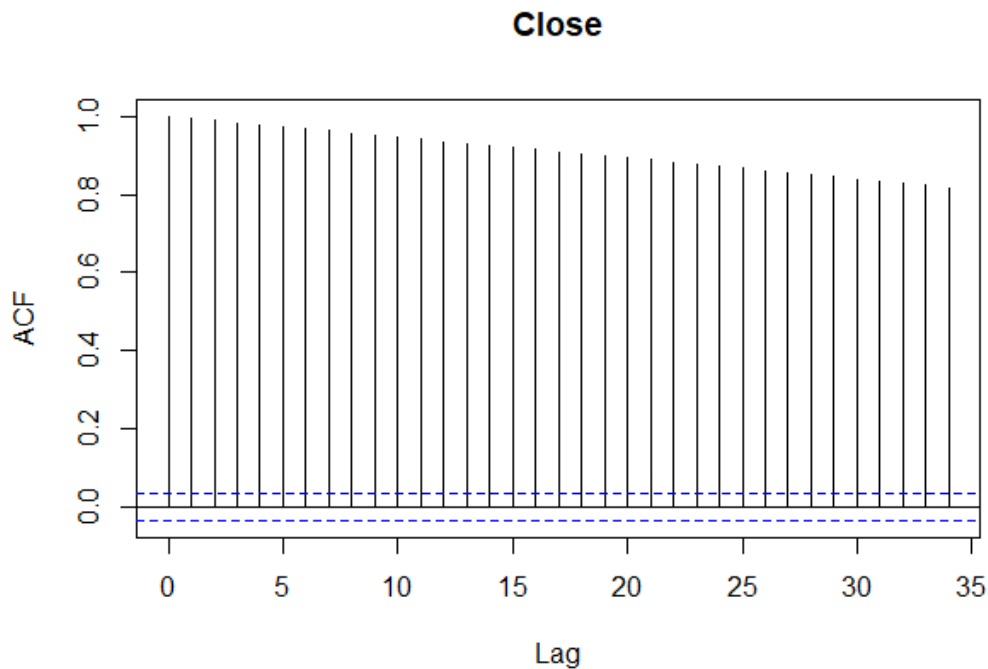


### Toyota ACF

### Close



### Mercedes ACF



### BMW ACF

Για την επιλογή των παραμέτρων των ARIMA μοντέλων χρησιμοποιήθηκε η εντολή `auto.arima` προκειμένου να πάρουμε τις παραμέτρους των μοντέλων. Για τη μετοχή της Toyota η εντολή μας έδωσε το  $ARIMA(0,1,0)$ , ενώ για τις μετοχές των Tesla, Mercedes και BMW το  $ARIMA(0,1,1)$ . Η επιλογή του αριθμού 1 για την παράμετρο I στα ARIMA μοντέλα, υποδεικνύει την έλλειψη στασιμότητας των χρονοσειρών.

Για να κάνουμε σωστή σύγκριση με τα ARIMA μοντέλα, εκπαιδεύσαμε τα MLPs, LSTMs και CNNs με `window=10` και `20` και `prediction window=10`, μετρώντας την πρόβλεψη για τις 10 τελευταίες ημέρες σε κάθε περίπτωση.

	MLP-10	MLP-20	LSTM-10	LSTM-20	CNN-10	CNN-20	ARIMA
Tesla	30,33	29,21	27,31	33,43	29,49	29,03	25,34
Toyota	9,03	7,66	8,55	6,75	9,34	7,51	3,48
Mercedes	3,43	2,79	3,57	3,30	3,77	3,22	0,59
BMW	1,71	0,47	1,63	1,20	1,36	0,58	0,93

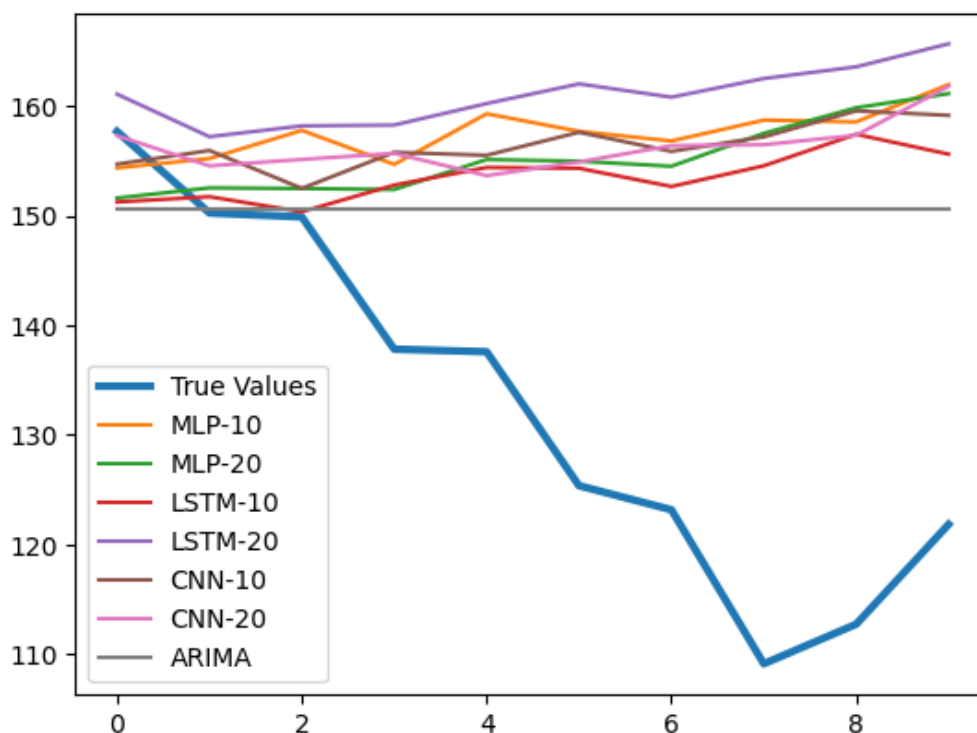
RMSE

	MLP-10	MLP-20	LSTM-10	LSTM-20	CNN-10	CNN-20	ARIMA
Tesla	25,63	23,88	22,25	28,41	24,43	23,85	21,43
Toyota	8,72	7,45	8,19	6,57	9,05	7,32	3,01
Mercedes	3,37	2,70	3,51	3,22	3,71	3,13	0,47
BMW	1,51	0,42	1,45	1,06	1,13	0,47	0,85

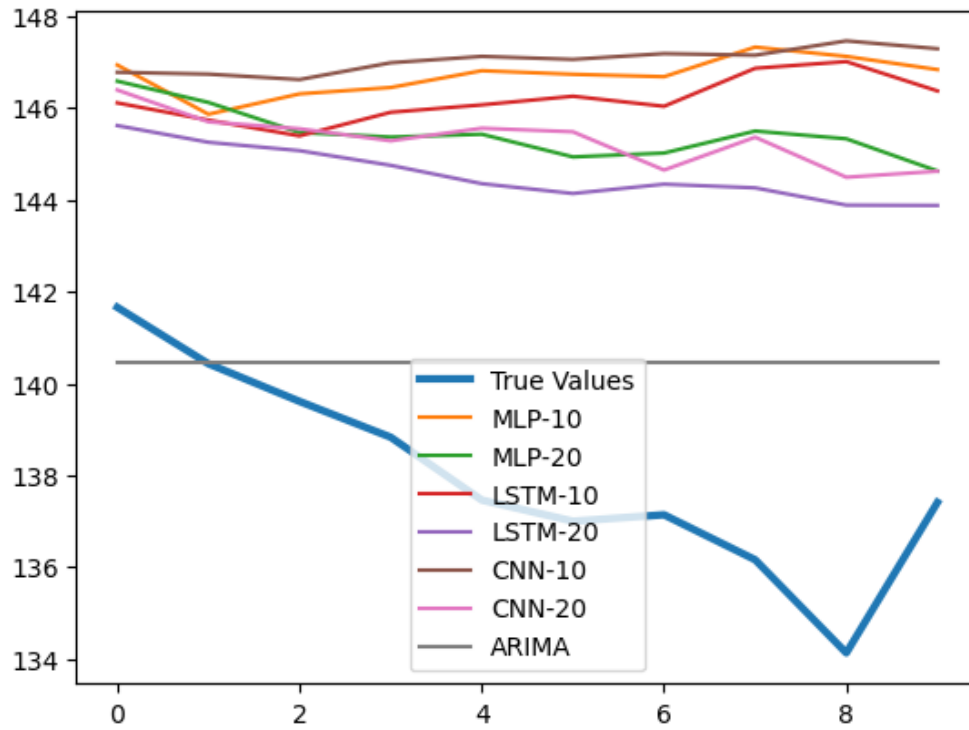
MAE

Η μόνη περίπτωση για την οποία υπάρχουν νευρωνικά δίκτυα που έδωσαν καλύτερο RMSE και MAE σε σχέση με τα ARIMA μοντέλα είναι αυτή της μετοχής της BMW και το καλύτερο εξ αυτών ήταν ο MLP με window=20.

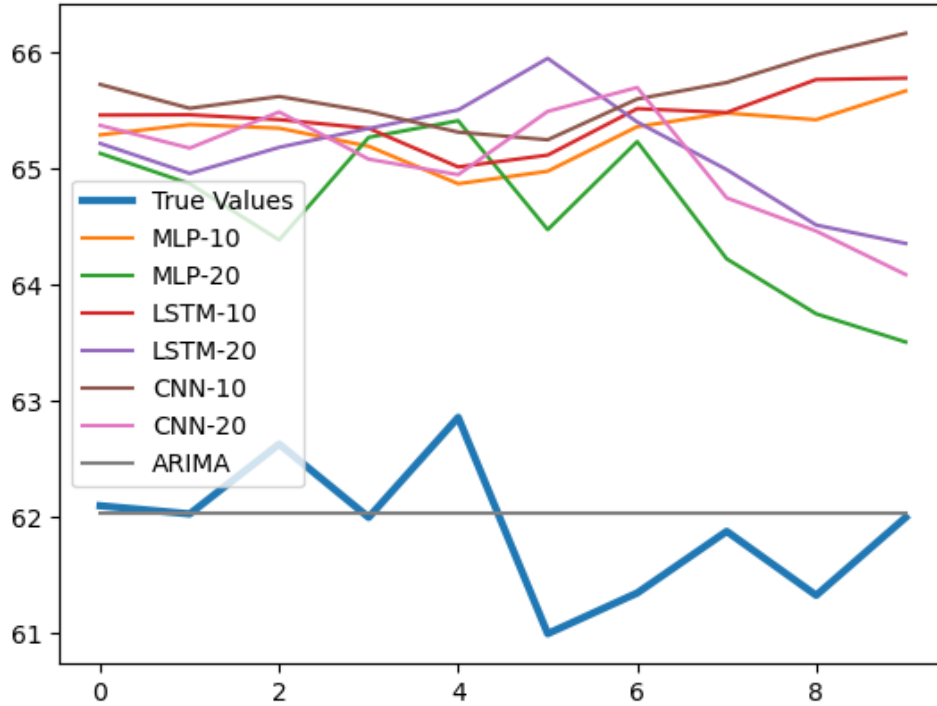
Παρακάτω δίνονται τα διαγράμματα πρόβλεψης για κάθε μετοχή



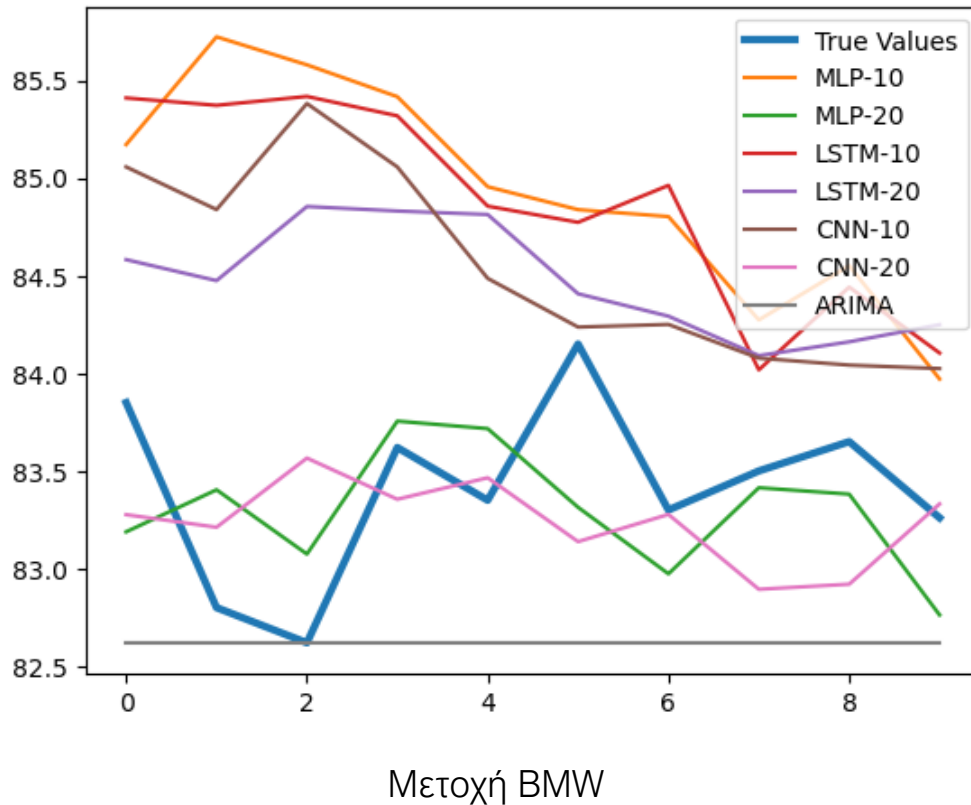
Μετοχή Tesla



Μετοχή Toyota



Μετοχή Mercedes



Απ' τα παραπάνω διαγράμματα φαίνεται η αδυναμία των ARIMA μοντέλων που εξετάστηκαν να εντοπίσουν μελλοντική διακύμανση. Αυτό οφείλεται στον όγκο των δεδομένων και την έλλειψη στασιμότητας στις χρονοσειρές.

## Κεφάλαιο 5: Συμπεράσματα

Στη συγκεκριμένη διπλωματική εργασία εξετάστηκαν διάφορες αρχιτεκτονικές τεχνητών νευρωνικών δικτύων και μελετήθηκε η απόδοση αυτών για τρεις εφαρμογές που αφορούν δεδομένα μετοχών στο χρηματιστήριο. Πιο συγκεκριμένα εξετάστηκαν οι αρχιτεκτονικές των Multilayer Perceptrons, LSTMs και CNNs σχετικά με την πρόβλεψη τιμών κλεισίματος των μετοχών της επόμενης ημέρας, με την πρόβλεψη βραχυχρόνιας τάσης (1 και 5 ημερών) και η μακροχρόνια πρόβλεψη τιμών κλεισίματος των μετοχών (10 ημέρες). Για την τελευταία εφαρμογή τα νευρωνικά δίκτυα συγκρίθηκαν με τη κλασική μέθοδο ανάλυσης χρονοσειρών ARIMA.



Σχετικά με την πρόβλεψη τιμής κλεισίματος της επόμενης ημέρας, κρίνετε εκ των αποτελεσμάτων ότι εν γένει η χρήση LSTM δικτύων είναι η καλύτερη, στη συνέχεια ακολουθεί η χρήση MLP δικτύων και τελευταία έρχεται η χρήση CNN. Απ' τον πειραματισμό φαίνεται επίσης ότι η χρήση μεγαλύτερου εύρους ημερών παρατήρησης ( $window=20$ ), είναι εξαιρετικά χρήσιμη, ειδικά για μετοχές μικρότερης διακύμανσης παρατηρήσεων. Η διαφορά στα σφάλματα που παίρνουμε από τις μετοχές σε κάθε περίπτωση, οφείλεται ακριβώς στη διαφορά των διακυμάνσεων των μετοχών, καθότι η μετοχή της Tesla που έχει μεγαλύτερη διακύμανση, παρουσιάζει και μεγαλύτερα σφάλματα, στη συνέχεια ακολουθεί η μετοχή της Toyota, ενώ οι μετοχές των Mercedes και BMW, λόγω μικρής διακύμανσης, παρουσιάζουν τα μικρότερα σφάλματα.

Σχετικά με την πρόβλεψη βραχυχρόνια τάσης, λύνοντας το πρόβλημα της δυαδικής ταξινόμησης (αυξητική/πτωτική τάση), παρατηρήθηκε ότι το σύνολο των μοντέλων που μελετήθηκαν κρίνεται ανεπαρκές, καθότι οι αντίστοιχες ακρίβειες ταξινόμησης δεν υπερβαίνουν το 54%. Αυτό οφείλεται σε δύο παράγοντες: α) οι ταξινομητές ταξινομούσαν μόνο σε μία κλάση, β) τα σφάλματα ταξινόμησης τύπου I και II ήταν μεγάλα. Θεωρούμε ότι η συγκεκριμένη εφαρμογή/πρόβλημα είναι μεγάλης πολυπλοκότητας κι ο μόνος τρόπος για να πάρουμε καλύτερα αποτελέσματα, είναι η χρήση περισσότερων δεδομένων ή/και πιο βαθιών αρχιτεκτονικών δικτύων. Εν γένει το πρόβλημα πρόβλεψης της τάσης μιας μετοχής δεν θα μπορούσε να είναι απλό λόγω της πολυπλοκότητάς του και της μη ντετερμινιστικής φύσης των αγορών που τις ορίζουν.

Σχετικά με το πρόβλημα της μακροχρόνιας πρόβλεψης τιμών κλεισίματος παρατηρούνται μεγαλύτερα σφάλματα, ειδικά στις μετοχές των Tesla, Toyota και Mercedes, εξαιτίας της προαναφερθείσας μεγάλης διακύμανσης των μετοχών. Από την άλλη, βλέπουμε ότι η έλλειψη στασιμότητας των χρονοσειρών που εξετάζονται, οδηγεί σε αδυναμία του ARIMA μοντέλου να συλλάβει τη μακροχρόνια τάση, δίνοντας όμως μια (συγκριτικά με τα υπόλοιπα μοντέλα για τις μετοχές μεγάλης διακύμανσης) καλή μέση τιμή.

Κλείνοντας, σε αυτή τη διπλωματική εργασία εξετάσαμε διάφορες εφαρμογές νευρωνικών δικτύων σχετικά με την πρόβλεψη χρονοσειρών στο χρηματιστήριο. Είδαμε ότι η χρήση των αρχιτεκτονικών των δικτύων που εξετάστηκαν είχε καλά αποτελέσματα για το πρόβλημα της πρόβλεψης τιμής κλεισίματος της επόμενης ημέρας, καθώς επίσης και της μακροχρόνιας πρόβλεψης τιμών κλεισίματος για

μετοχές με μικρή διακύμανση, όπως αυτή της BMW. Η χρήση αυτών των αρχιτεκτονικών από την άλλη, φαίνεται να είναι αδύναμη στην πρόβλεψη μακροχρόνιας/βραχυχρόνιας τάσης, στην προσπάθειά μας να λύσουμε το πρόβλημα της δυαδικής ταξινόμησης (αύξηση-μείωση τάσης στις 5 επόμενες ημέρες). Για το λόγο αυτό κρίνεται σκόπιμο σε μελλοντική έρευνα να εξεταστούν περισσότερες αρχιτεκτονικές νευρωνικών δικτύων, καθώς και η προσπάθεια κατασκευής πολυμεταβλητών (multivariate) μοντέλων που θεωρούμε ότι θα βελτιώσει τις ζητούμενες μετρικές.

## Βιβλιογραφία

1. Nils J. Nilsson (1998) Introduction to Machine Learning, Stanford University, κεφάλαιο 4
2. Shai Shalev-Shwartz, Shai Ben-David (2014) Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, κεφάλαιο 9.2, 9.3, 15, 20, 22
3. Haykin, S. (1999) Neural Networks: A Comprehensive Foundation, Prentice Hall
4. [https://en.wikipedia.org/wiki/Artificial\\_neural\\_network](https://en.wikipedia.org/wiki/Artificial_neural_network)
5. Williams, Ronald J.; Hinton, Geoffrey E.; Rumelhart, David E. (October 1986). "Learning representations by back-propagating errors". *Nature*. 323 (6088): 533–536.
6. Ian J. Goodfellow, Yoshua Bengio and Aaron Courville (2016) Deep Learning, MIT Press, κεφάλαιο 6, 9, 10
7. Mitchell Tom M. (1997) Machine learning. Vol. 1. , bk. 9. : McGraw-hill New York, κεφάλαιο 4.6.5
8. [https://en.wikipedia.org/wiki/Convolutional\\_neural\\_network](https://en.wikipedia.org/wiki/Convolutional_neural_network)
9. Sepp Hochreiter, Jürgen Schmidhuber; Long Short-Term Memory. *Neural Comput* 1997; 9 (8): 1735–1780.
10. Douglas C. Montgomery, Cheryl L. Jennings, Murat Kulahci (2015) Introduction to Time Series Analysis and Forecasting, 2nd Edition, Wiley
11. R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. <https://www.R-project.org/>.
12. Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia

13. <https://www.python.org>
14. <https://keras.io/>
15. Shen, J., Shafiq, M.O. Short-term stock market price trend prediction using a comprehensive deep learning system. J Big Data 7, 66 (2020).