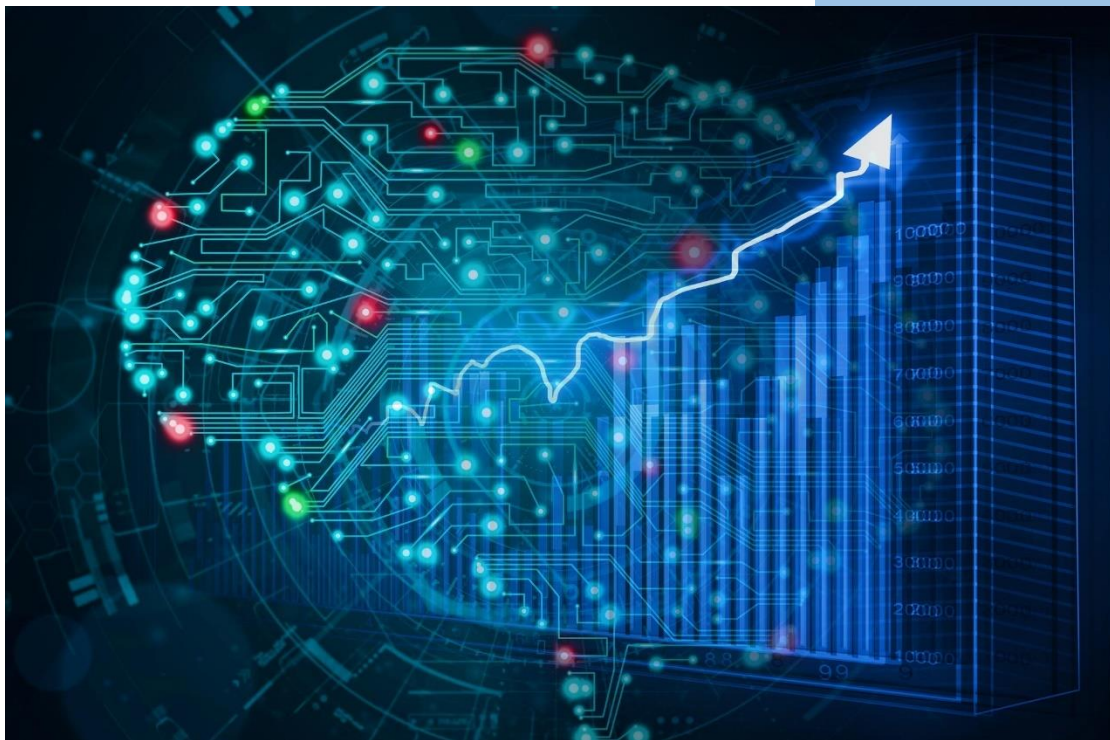




Πανεπιστήμιο Πειραιώς
Τμήμα Ψηφιακών Συστημάτων
Π.Μ.Σ. "Πληροφορικά Συστήματα & Υπηρεσίες"

Διπλωματική
εργασία

Διαδικτυακή εφαρμογή πρόβλεψης χρονοσειρών στο χρηματιστήριο με χρήση νευρωνικών δικτύων



Νικολάου Στέφανος (ΜΕ 1911)

Επιβλέπων: Φιλιππάκης Μιχαήλ

Πειραιάς,
Σεπτέμβριος 2021



Η παρούσα διπλωματική εργασία εκτελείται στο πλαίσιο του ΠΜΣ «Πληροφορικά Συστήματα & Υπηρεσίες» κατά το ακαδημαϊκό έτος 2020-21.

Copyright © Νικολάου Στέφανος, 2021. Με επιφύλαξη παντός δικαιώματος.



Περιεχόμενα

Περίληψη	6
Κεφάλαιο 1: Χρονοσειρές.....	7
1.1 Εισαγωγή	7
1.2 Χαρακτηριστικά των χρονοσειρών	8
1.2.1 Τάση (Trend)	8
1.2.2 Εποχικότητα (Seasonality)	8
1.2.3 Κυκλικές διακυμάνσεις (Cyclical variations).....	9
1.2.4 Ακανόνιστες διακυμάνσεις (Irregular variations)	9
1.3 Ανάλυση χρονοσειρών	10
1.3.1 Μέθοδοι ανάλυσης	10
1.3.1.1 Τομέας συχνοτήτων (Frequency domain)	10
1.3.1.2 Τομέας χρόνου (Time domain)	12
1.3.1.3 Παραμετρικές και μη παραμετρικές μέθοδοι	14
1.3.2 Τύποι ανάλυσης.....	15
1.3.2.1 Διερευνητική ανάλυση (Exploratory analysis).....	15
1.3.2.2 Προσαρμογή καμπύλης (Curve fitting).....	16
1.3.2.3 Κατηγοριοποίηση (Classification)	20
1.3.2.4 Τμηματοποίηση (Segmentation)	22
1.3.2.5 Προβλεπτική ανάλυση (Predictive analysis).....	24
Κεφάλαιο 2: Τεχνητά νευρωνικά δίκτυα.....	28
2.1 Εισαγωγή	28
2.2 Λειτουργία νευρωνικών δικτύων.....	29
2.2.1 Διάδοση πληροφορίας	30
2.2.2 Συνάρτηση ενεργοποίησης (Activation function)	31
2.2.2.1 Δυαδική συνάρτηση βήματος (Binary step function).....	31



2.2.2.2 Γραμμική συνάρτηση ενεργοποίησης (Linear activation function)	31
2.2.2.3 Σιγμοειδής συνάρτηση (Sigmoid activation function)	31
2.2.2.4 Συνάρτηση υπερβολικής εφαπτομένης (Tanh function)	32
2.2.2.5 Λειτουργία ενεργοποίησης ανορθωτή (Rectified linear unit – ReLU).....	32
2.2.3 Συνάρτηση κόστους (Cost function)	33
2.3 Κύριες κατηγορίες νευρωνικών δικτύων.....	34
2.3.1 Εμπρός-τροφοδοτούμενα νευρωνικά δίκτυα (Feedforward neural networks - FNN).....	34
2.3.1.1 Perceptron ενός επιπέδου (Single-layer perceptron – SLP)	34
2.3.1.2 Perceptron πολλών επιπέδων (Multi-layer perceptron - MLP)	35
2.3.2 Ανατροφοδοτούμενα νευρωνικά δίκτυα (Recurrent neural networks - RNN)	36
2.3.2.1 Νευρωνικά δίκτυα Μακράς Βραχύχρονης Μνήμης (Long Short-term Memory - LSTM).....	37
2.3.2.2 Επαναλαμβανόμενες μονάδες με πύλη (Gated recurrent units - GRUs)	39
2.3.2.3 Αμφίδρομα επαναλαμβανόμενα νευρωνικά δίκτυα (Bidirectional recurrent neural networks - BRNN).....	41
2.3.2 Συνελκτικά νευρωνικά δίκτυα (Convolutional neural networks - CNN)...	42
2.3.4 Αυτόματοι κωδικοποιητές (Autoencoders)	43
2.5 Αλγόριθμοι βελτιστοποίησης	44
2.5.1 Σύγκλιση με ελάττωση της παραγώγου (Gradient descent).....	45
2.5.2 Στοχαστική σύγκλιση με ελάττωση της παραγώγου (Stochastic Gradient Descent - SGD).....	45
2.5.3 Προσαρμοστικός αλγόριθμος κλίσης (Adaptive gradient -AdaGrad).....	45
2.5.4 RMSprop	46
2.5.5 Προσαρμοστική Εκτίμηση Στιγμής (Adaptive Moment Estimation - ADAM).....	47
Κεφάλαιο 3: Υλοποίηση εφαρμογής	48
3.1 Γενικές πληροφορίες	48



3.2 Αξιολόγηση μοντέλων.....	49
3.2.1 Σύνολα δεδομένων για επιλογή μοντέλου.....	49
3.2.2 Μέθοδος αξιολόγησης ακρίβειας	49
3.2.3 Περιγραφική ανάλυση δεδομένων.....	49
3.2.3.1 Χρονοσειρά μετοχής AMD	50
3.2.3.2 Χρονοσειρά μετοχής NVDA	51
3.2.3.3 Χρονοσειρά μετοχής TSLA.....	52
3.2.4 Πρόβλεψη με LSTM.....	53
3.2.4.1 Αποτελέσματα LSTM - AMD.....	54
3.2.4.2 Αποτελέσματα LSTM - NVDA.....	55
3.2.4.3 Αποτελέσματα LSTM - TSLA	56
3.2.5 Πρόβλεψη με MLP	57
3.2.5.1 Αποτελέσματα MLP – AMD.....	57
3.2.5.2 Αποτελέσματα MLP - NVDA.....	58
3.2.5.3 Αποτελέσματα MLP - TSLA.....	59
3.2.6 Πρόβλεψη με NeuralProphet.....	60
3.2.6.1 Αποτελέσματα NeuralProphet – AMD.....	61
3.2.6.2 Αποτελέσματα NeuralProphet - NVDA.....	62
3.2.6.3 Αποτελέσματα NeuralProphet - TSLA.....	63
3.3 Συνοπτική παρουσίαση εφαρμογής.....	64
Συμπεράσματα.....	67
Βιβλιογραφικές αναφορές	69



Περίληψη

Οι χρονοσειρές έχουν ευρείες εφαρμογές στον τομέα των οικονομικών όπως η εκτίμηση του ΑΕΠ, των επιτοκίων ή όπως στην περίπτωση μας, των μελλοντικών τιμών των μετοχών. Για την εκτίμηση αυτών των μεγεθών μπορούν να εφαρμοστούν διάφορα μοντέλα, λιγότερο ή περισσότερο πολύπλοκα. Παρόλο που τα οικονομικά δεδομένα κατά κύριο λόγο έχουν ένα διακριτό χαρακτήρα, τα οικονομετρικά μοντέλα συνηθίζουν να τα αντιμετωπίζουν ως συνεχείς παρατηρήσεις.

Τα τελευταία χρόνια έχουν διεξαχθεί διάφορες μελέτες σχετικά με την πρόβλεψη των μελλοντικών τιμών μετοχών στο χρηματιστήριο οι οποίες καταλήγουν στην πρόταση διαφόρων τεχνικών. Μπορούμε να πούμε πως οι προεξέχουσες μέθοδοι πρόβλεψης εμπίπτουν σε δυο κύριες κατηγορίες: στις στατιστικές μεθόδους και στις τεχνικές εύκαμπτης υπολογιστικής (soft computing).

Τα τεχνητά νευρωνικά δίκτυα ως μέθοδος εύκαμπτης υπολογιστικής θεωρούνται ως τα πιο εύστοχα και ευρέως χρησιμοποιούμενα προβλεπτικά μοντέλα. Αυτός είναι και ο λόγος για τον οποίο βρίσκουν συχνή εφαρμογή στην επίλυση προβλημάτων προβλεπτικής φύσεως. Στο πλαίσιο της παρούσας διπλωματικής εργασίας, αφού γίνει ανασκόπηση της σχετικής βιβλιογραφίας, θα υλοποιηθεί μια διαδικτυακή εφαρμογή η οποία θα είναι σε θέση να παράγει προβλέψεις για μελλοντικές τιμές στο χρηματιστήριο, εκμεταλλευόμενη την προβλεπτική ικανότητα μοντέλων βασισμένων σε νευρωνικά δίκτυα.



Κεφάλαιο 1: Χρονοσειρές

1.1 Εισαγωγή

Μια χρονολογική σειρά (ή αλλιώς χρονοσειρά) αποτελεί μια αλληλουχία αριθμητικών δεδομένων τα οποία συνήθως λαμβάνονται σε διαδοχικά και ισαπέχοντα χρονικά διαστήματα. Ωστόσο, υπάρχουν και περιπτώσεις κατά τις οποίες ο χρόνος πραγματοποίησης της δειγματοληψίας δεν είναι σταθερός, αλλά μεταβαλλόμενος. Τέτοιες περιπτώσεις αποτελούν οι χρονοσειρές που προκύπτουν από χρηματιστηριακά δεδομένα, όπως οι τιμές ενός Δείκτη Αξιών ή μεμονωμένων μετοχών, καθώς μεσολαβούν χρονικά διαστήματα (όπως Σαββατοκύριακα ή αργίες) κατά τα οποία οι αγορές παραμένουν κλειστές και έτσι ο χρόνος δειγματοληψίας αναγκαστικά μεταβάλλεται.

Οι χρονοσειρές ανάλογα με τον τρόπο κατά τον οποίο πραγματοποιήθηκε η μέτρηση των χρονικών σημείων που τις αποτελούν χωρίζονται σε δύο τύπους, τις συνεχείς και τις διακριτές. Ωστόσο, αξίζει να σημειωθεί πως και στις δυο περιπτώσεις η υπό εξέταση μεταβλητή μπορεί να είναι τόσο συνεχής όσο και διακριτή. Επί της ουσίας και ανεξαρτήτου χρόνου διενέργειας της δειγματοληψίας, μια χρονοσειρά αποτελεί ένα σύνολο δεδομένων στο οποίο οι παρατηρήσεις τοποθετούνται σε χρονολογική σειρά.

Η ανάλυση αυτού του είδους συνόλων δεδομένων μπορεί να έχει πολλούς διαφορετικούς σκοπούς ανάλογα με το πεδίο εφαρμογής. Οι αναλύσεις μπορούν να ποικίλουν από προβλέψεις μελλοντικών τιμών διαφορών μεγεθών, την εξαγωγή χρήσιμων πληροφοριών οι οποίες μπορεί να καλύπτονται από το θόρυβο των δεδομένων ή και την εκτίμηση της πιθανότητας πραγματοποίησης κάποιου συγκεκριμένου γεγονότος στο μέλλον.

Ιδίως η προβλεπτική ανάλυση των χρονοσειρών, η οποία και αποτελεί το κύριο αντικείμενο μελέτης της συγκεκριμένης εργασίας τυγχάνει πληθώρας εφαρμογών, όπως η συμβολή σε περιπτώσεις οικονομικού σχεδιασμού, η πρόβλεψη του ύψους των μελλοντικών πωλήσεων, ο έλεγχος των αποθεμάτων, η εκτίμηση και σύγκριση εναλλακτικών οικονομικών σχεδιασμών, καθώς και η διαχείριση του οικονομικού ρίσκου. [1], [2]

1.2 Χαρακτηριστικά των χρονοσειρών

Οι τιμές των παρατηρήσεων μιας χρονοσειράς μπορούν να επηρεαστούν από διάφορους παράγοντες, οι οποίοι χωρίζονται σε τέσσερις κατηγορίες και αποτελούν τα κύρια συστατικά ή χαρακτηριστικά της χρονοσειράς [3], [4], [5]:

1.2.1 Τάση (Trend)

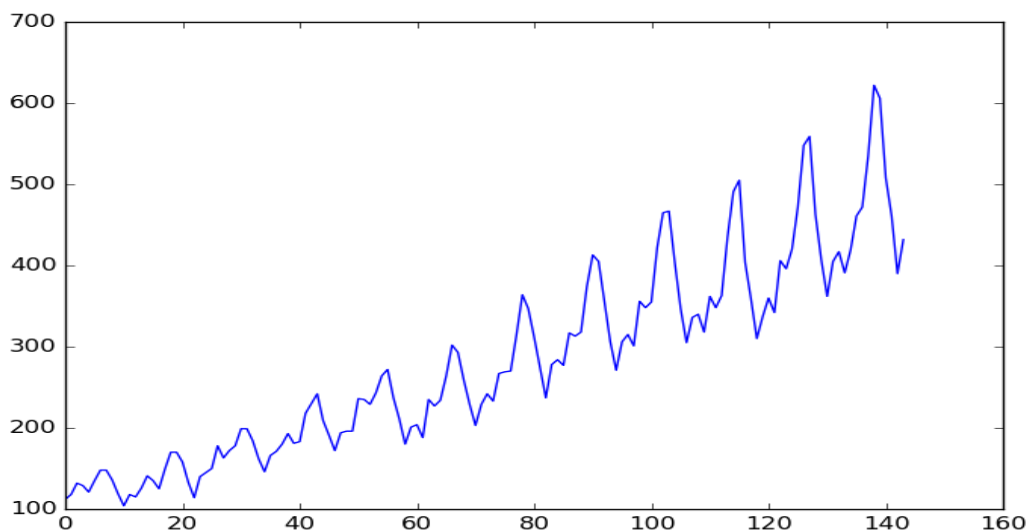
Το χαρακτηριστικό αυτό έχει μακροχρόνια επίδραση και συνιστά τη γενικότερη ροπή που έχουν τα δεδομένα της χρονοσειράς είτε προς αύξηση είτε προς μείωση. Οι βραχυπρόθεσμες «ταλαντώσεις» δεν λαμβάνονται υπόψιν και η τάση ουσιαστικά αποτελεί μια σταθερή και σταδιακή κίνηση της χρονοσειράς προς την ίδια κατεύθυνση για μεγάλο χρονικό διάστημα. Παραδείγματος χάριν, οι χρονοσειρές που αφορούν πληθυσμιακά δεδομένα ή στοιχεία πωλήσεων μεγάλων εταιριών παρουσιάζουν συνήθως ανοδική τάση. Ενώ, οι χρονοσειρές που αφορούν δεδομένα θανάτων συνηθίζουν να έχουν φθίνουσα τάση. Η τάση μπορεί να είναι είτε γραμμική είτε μη-γραμμική, ενώ χρονοσειρές οι οποίες δεν παρουσιάζουν κάποιο αυξανόμενο ή μειούμενο μοτίβο στο χρόνο, καλούνται στατικές.

1.2.2 Εποχικότητα (Seasonality)

Πολλές χρονοσειρές παρουσιάζουν τακτικές διακυμάνσεις στις τιμές των παρατηρήσεων οι οποίες επηρεάζονται από εποχιακούς παράγοντες σε περιόδους μικρότερες του έτους. Αυτού του είδους οι διακυμάνσεις γίνονται εύκολα αντιληπτές και είναι εύκολο είτε να μετρηθούν είτε ακόμη και να αφαιρεθούν από τα δεδομένα μας για λόγους που σχετίζονται με την ανάλυση αυτών. Η συντριπτική πλειοψηφία των χρονοσειρών που σχετίζονται με οικονομικά δεδομένα, όπως οι τιμές των μετοχών ή τα κέρδη ενός εμπορικού καταστήματος, παρουσιάζουν εποχικότητα. Οι εποχικές διακυμάνσεις αποτελούν απόρροια δυο κύριων αιτιών:

1. Φυσικές αιτίες: Οι διακυμάνσεις που επηρεάζονται από φυσικές αιτίες όπως η εναλλαγή των εποχών και των καιρικών συνθηκών. Παραδείγματος χάριν, οι πωλήσεις κλιματιστικών αυξάνονται κατά τους θερινούς μήνες μαζί με την άνοδο της θερμοκρασίας.
2. Ανθρωπογενείς αιτίες: Αποτελούν διακυμάνσεις οι οποίες προκαλούνται από παράγοντες οι οποίοι σχετίζονται με τον άνθρωπο,

όπως συνήθειες, τάσεις της μόδας, ήθη και έθιμα. Λόγου χάρη, οι πωλήσεις ανθών και γλυκών αυξάνονται την ημέρα του Αγίου Βαλεντίνου.



Εικόνα 1: Χρονοσειρά που παρουσιάζει ανοδική τάση και εποχικότητα

1.2.3 Κυκλικές διακυμάνσεις (Cyclical variations)

Οι διακυμάνσεις οι οποίες παρατηρούνται στα στοιχεία μιας χρονοσειράς σε μακροχρόνιες περιόδους μεγαλύτερες του ενός έτους ονομάζονται κυκλικές. Συνήθως, παρατηρούνται σε δεδομένα οικονομικού τύπου και σε περιόδους οι οποίες διαρκούν από πέντε έως 12 έτη ή και περισσότερο. Οι διακυμάνσεις αυτές συχνά συγχέονται με τις εποχιακές διακυμάνσεις, αλλά είναι αρκετά διαφορετικές. Επί της ουσίας, εάν οι διακυμάνσεις δεν παρατηρούνται σε σταθερές περιόδους και υπερβαίνουν το ένα έτος, τότε είναι κυκλικές.

1.2.4 Ακανόνιστες διακυμάνσεις (Irregular variations)

Ένα ακανόνιστο στοιχείο, γνωστό και ως υπόλοιπο (residual), είναι αυτό το οποίο μένει αφότου εκτιμήσουμε και στη συνέχεια απαλείψουμε τα στοιχεία τάσης και εποχικότητας μιας χρονοσειράς. Τα στοιχεία αυτά προκύπτουν ως αποτέλεσμα βραχυπρόθεσμων απρόβλεπτων διακυμάνσεων. Γεγονότα τα οποία μπορούν να προκαλέσουν τέτοιου είδους ακανόνιστες διακυμάνσεις μπορούν να σχετίζονται με φυσικές καταστροφές, απεργίες κ.ά. Οι διακυμάνσεις αυτές, οι οποίες είναι γνωστές



και ως «θόρυβος» (noise) διαρκούν για σύντομα και μη επαναλαμβανόμενα χρονικά διαστήματα, μικρότερα του ενός έτους.

1.3 Ανάλυση χρονοσειρών

Μια χρονοσειρά αποτελεί ίσως την απλούστερη μορφή συνόλου δεδομένων που μπορεί να συναντήσει κάποιος αναλυτής καθώς αποτελείται από δυο στήλες πληροφοριών, μια για τις παρατηρήσεις και άλλη μια για τις αντίστοιχες χρονικές περιόδους. Συνάμα, όμως, είναι και ένα από τα σημαντικότερα είδη συνόλου δεδομένων από πλευράς χρηστικότητας, καθώς περιέχει συμπυκνωμένα και με εύκολο τρόπο, μεγάλο όγκο χρήσιμων πληροφοριών. Οι πληροφορίες αυτές μπορούν να γίνουν αντιληπτές ακόμη και από χρήστες χωρίς ιδιαίτερες τεχνικές γνώσεις και να οδηγήσουν στη λήψη σημαντικών αποφάσεων βασισμένων στα δεδομένα. [6], [7]

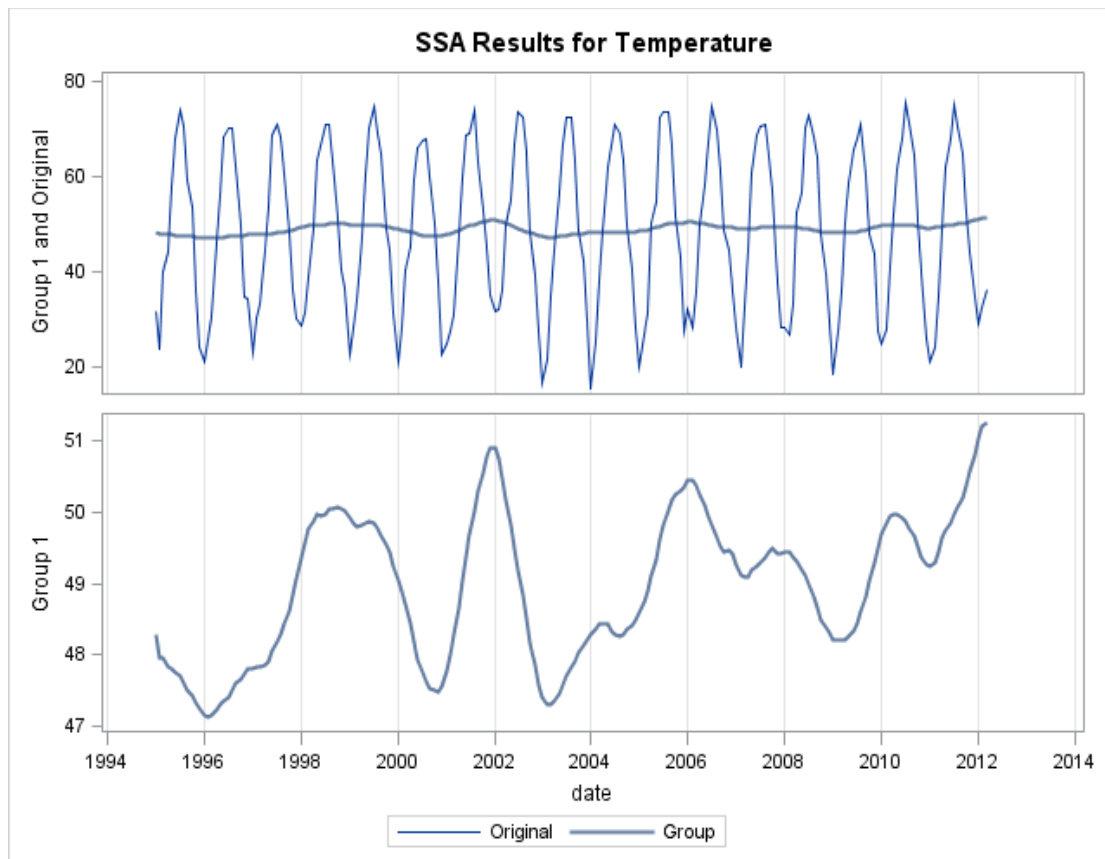
1.3.1 Μέθοδοι ανάλυσης

Η φύση των χρονοσειρών είναι τέτοια που δίνει τη δυνατότητα για πραγματοποίηση διαφόρων ειδών ανάλυσης ανάλογα με το πεδίο εφαρμογής και το στόχο του αναλυτή. Οι μέθοδοι ανάλυσης χρονοσειρών χωρίζονται σε δυο κύριες κατηγορίες: Σε αυτές που αναλύονται με βάση τη συχνότητα (Τομέας συχνοτήτων – Frequency domain) και σε αυτές που αναλύονται σε σχέση με το χρόνο (Τομέας χρόνου – Time domain). Η επιλογή μεθόδου έγκειται στους στόχους που θέλει να επιτύχει ο χρήστης μέσω της ανάλυσης. Ωστόσο, δεν είναι λίγες οι περιπτώσεις, όπου οι δυο μέθοδοι χρησιμοποιούνται συνδυαστικά. [2], [6], [8], [9].

1.3.1.1 Τομέας συχνοτήτων (Frequency domain)

Ο τομέας συχνοτήτων αναφέρεται στον αναλυτικό χώρο στον οποίο οι μαθηματικές συναρτήσεις ή σήματα εκφράζονται ως προς τη συχνότητα παρά ως προς το χρόνο [10]. Αυτού του είδους η προσέγγιση είναι χρήσιμη για αρκετούς λόγους, όπως οι περιπτώσεις όπου απαιτείται η εξάλειψη εποχικών διακυμάνσεων προτού εκτελεστεί κάποια ενέργεια ανάλυσης. Επιπλέον, η μέθοδος αυτή επιτρέπει την ανάλυση των συνεπειών διαφόρων διαδικασιών προσαρμογής. Τέτοιες συνέπειες μπορεί να είναι αρκετά σημαντικές, όπως η προσθήκη νέων χαρακτηριστικών στη χρονοσειρά ή η εξάλειψη επιθυμητών συνιστωσών μαζί με κάποιες ανεπιθύμητες [11]. Στις τεχνικές ανάλυσης χρονοσειρών με βάση τη συχνότητα συμπεριλαμβάνονται:

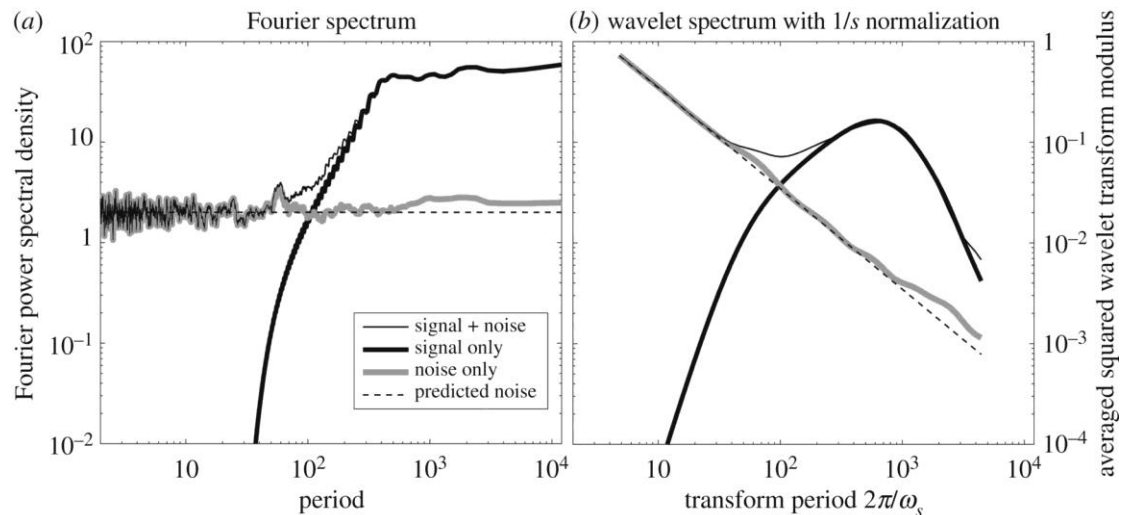
- Φασματική ανάλυση (spectral analysis): Πολλές χρονοσειρές παρουσιάζουν αρκετά σύνθετη περιοδική συμπεριφορά. Μέσω της φασματικής ανάλυσης μπορεί να γίνει αποσύνθεση μιας χρονοσειράς στα επιμέρους στοιχεία της με σκοπό τον εντοπισμό κυκλικότητας και περιοδικότητας στα δεδομένα. Αυτή η μορφή ανάλυσης είναι από τις πιο διαδεδομένες για δεδομένα που σχετίζονται με τους χώρους της γεωφυσικής, της ωκεανογραφίας, της αστρονομίας και της μηχανικής [12].



Εικόνα 2 - Ανάλυση χρονοσειράς που αναπαριστά δεδομένα θερμοκρασίας, με χρήση μοναδικής ανάλυσης φάσματος (singular spectrum analysis)

Πηγή: <https://blogs.sas.com/content/iml/2012/04/10/a-singular-spectrum-analysis-of-a-temperature-time-series.html#prettyPhoto>

- Ανάλυση κυματιδίων (wavelet analysis): Τα εργαλεία τα οποία χρησιμοποιούν τεχνικές ανάλυσης κυματιδίων είναι ιδιαίτερος χρήσιμα για ανάλυση χρονοσειρών και ιδίως αυτών οι οποίες αφορούν δεδομένα τα οποία παρουσιάζουν στοιχεία περιοδικότητας. Τα κυματίδια δίνουν στο χρήστη χρήσιμες πληροφορίες σχετικά με τις διακυμάνσεις τοπικών μέσων, όπως η μέση μηνιαία θερμοκρασία του επιπέδου θαλάσσης, η μέση ετήσια θερμοκρασία στην ηπειρωτική Ελλάδα κ.ά. [13]



Εικόνα 3 – Αριστερά φαίνεται ο μετασχηματισμός Fourier και δεξιά ο μετασχηματισμός κυματιδίων μιας χρονοσειράς. Συμπεριλαμβάνονται φάσματα τόσο με θόρυβο όσο και χωρίς

Πηγή: <https://royalsocietypublishing.org/doi/10.1098/rspa.2016.0776>

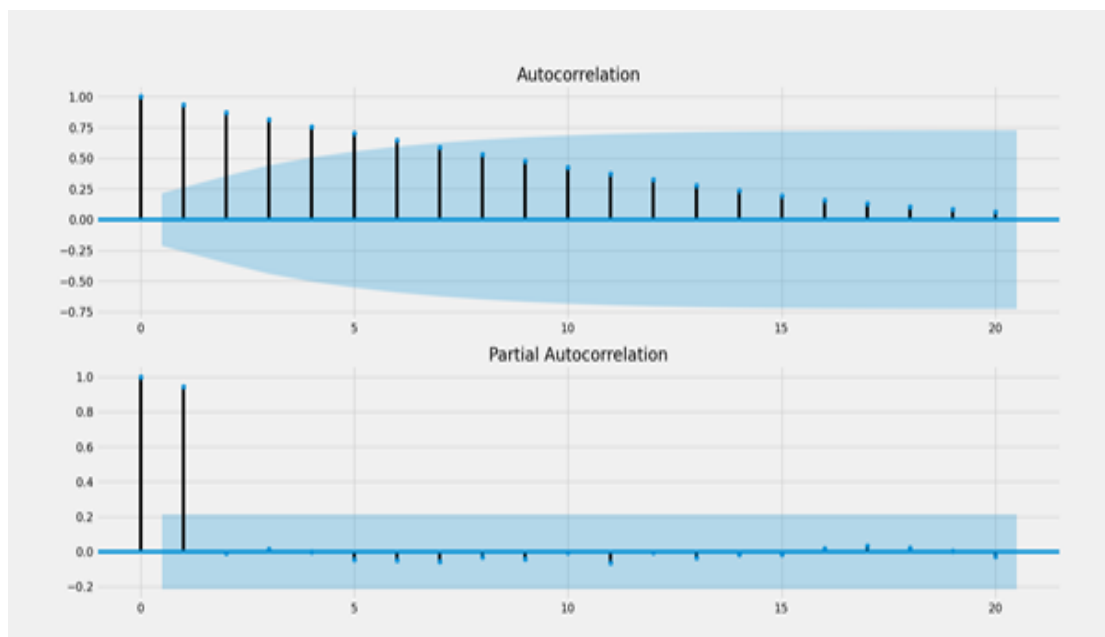
1.3.1.2 Τομέας χρόνου (Time domain)

Η ανάλυση μέσω του τομέα χρόνου αφορά την ανάλυση μαθηματικών συναρτήσεων, φυσικών σημάτων, ή χρονοσειρών που σχετίζονται με οικονομικά ή περιβαλλοντικά δεδομένα σε σχέση με το χρόνο. Στον τομέα χρόνου ένα σήμα εξετάζεται ως ο συνδυασμός πολλών κλιμακούμενων και μετατοπισμένων παλμών. Στις τεχνικές ανάλυσης χρονοσειρών με βάση το χρόνο συμπεριλαμβάνονται:

- Αυτοσυσχέτιση (autocorrelation): Η αυτοσυσχέτιση είναι μια μορφή σειριακής εξάρτησης (ή αλλιώς συσχέτισης) ενός σήματος ως συνάρτηση της χρονικής καθυστέρησης με ένα καθυστερημένο αντίγραφο του εαυτού του. Όταν σε μια αλληλουχία αριθμών είναι δυνατή η πρόβλεψη μελλοντικών τιμών βασισμένοι στις προηγούμενες τιμές, τότε μπορούμε να πούμε πως η σειρά παρουσιάζει αυτοσυσχέτιση. Η ανάλυση των δεδομένων μας για την ύπαρξη αυτοσυσχέτισης είναι ιδιαίτερως σημαντική:
 - Σχεδόν για οποιοδήποτε είδος ανάλυσης δεδομένων, ένα από τα πρώτα βήματα είναι η ανάλυση παλινδρόμησης, η οποία ωστόσο, προϋποθέτει ότι τα δεδομένα δεν παρουσιάζουν αυτοσυσχέτιση. Σε αντίθετη περίπτωση, η ανάλυση μπορεί να οδηγηθεί σε παραπλανητικά συμπεράσματα.
 - Επιπλέον, κάποιες μέθοδοι πρόβλεψης χρονοσειρών (ειδικά τα μοντέλα παλινδρόμησης) βασίζονται στην υπόθεση ότι δεν υπάρχει

αυτοσυσχέτιση στα υπόλοιπα (residuals). Οι αναλυτές χρησιμοποιούν συχνά τα υπόλοιπα για να αποφασίσουν εάν το προβλεπτικό μοντέλο που θέλουν να εφαρμόσουν είναι κατάλληλο για τα δεδομένα τους ή όχι. Επομένως, γίνεται κατανοητό πως η λανθασμένη αντίληψη περί ύπαρξης αυτοσυσχέτισης μπορεί να οδηγήσει σε εσφαλμένες αποφάσεις.

- Τέλος, η αυτοσυσχέτιση αποτελεί σύμμαχο του αναλυτή στην προσπάθεια να ανακαλύψει κρυμμένα μοτίβα στα υπό ανάλυση δεδομένα. Πιο συγκεκριμένα, είναι ιδιαίτερα χρήσιμη στον εντοπισμό τάσης και εποχικότητας σε δεδομένα χρονοσειράς. Επιπλέον, μέσω της χρήσης των συναρτήσεων αυτοσυσχέτισης (Autocorrelation function – ACF) και μερικής αυτοσυσχέτισης (Partial autocorrelation function – PACF) γίνεται η επιλογή του κατάλληλου Αυτοπαλίνδρομου ολοκληρωμένου μοντέλου κινητού μέσου (ARIMA).



Εικόνα 4 - Διαγράμματα αυτοσυσχέτισης και μερικής αυτοσυσχέτισης τα οποία περιλαμβάνουν και το 95% διάστημα εμπιστοσύνης

- Διασταυρούμενη αυτοσυσχέτιση (Cross-correlation): Είναι ένα μέτρο ομοιότητας για τη σχέση μεταξύ δυο σειρών (ή συνόλων δεδομένων) ως συνάρτηση της μετατόπισης της μιας σε σχέση με την άλλη. Στην απλούστερη μορφή της μπορεί να περιγραφεί ως εξής: Έστω ότι έχουμε μια ανεξάρτητη μεταβλητή X και δυο εξαρτημένες μεταβλητές Y και Z . Εάν η μεταβλητή X



επηρεάζει τη μεταβλητή Y και οι δυο τους είναι θετικά συσχετισμένες, τότε εάν η τιμή της μεταβλητής X αυξηθεί το ίδιο θα συμβεί και με την τιμή της μεταβλητής Y . Εάν τα παραπάνω ισχύουν και για τη σχέση μεταξύ των X και Z , τότε μπορούμε να πούμε πως οι μεταβλητές Y και Z παρουσιάζουν διασταυρούμενη αυτοσυσχέτιση, καθώς η συμπεριφορά τους είναι θετικά συσχετισμένη ως αποτέλεσμα της σχέσης εκάστης με την ανεξάρτητη μεταβλητή X .

1.3.1.3 Παραμετρικές και μη παραμετρικές μέθοδοι

Πέρα από τα όσα έχουν αναφερθεί έως τώρα στη συγκεκριμένη ενότητα, οι μέθοδοι ανάλυσης χρονοσειρών μπορούν να κατηγοριοποιηθούν περαιτέρω, σε παραμετρικές και μη παραμετρικές.

- Παραμετρικές: Αυτού του είδους οι προσεγγίσεις υποθέτουν ότι η υποκείμενη στοχαστική διαδικασία έχει συγκεκριμένη δομή, η οποία μπορεί να περιγραφεί χρησιμοποιώντας ένα μικρό αριθμό παραμέτρων. Σκοπός της προσέγγισης είναι η εκτίμηση των παραμέτρων του μοντέλου που περιγράφει τη στοχαστική διαδικασία.
- Μη παραμετρικές: Σε αντίθεση με τις παραμετρικές μεθόδους, η μη παραμετρική προσέγγιση προβαίνει σε εκτίμηση της συνδιακύμανσης ή του φάσματος της σειράς χωρίς να προϋποθέτει ότι αυτή έχει κάποια συγκεκριμένη δομή.

Οι μέθοδοι οι οποίες βασίζονται σε παραμετρικούς ελέγχους προϋποθέτουν συνήθως την εκπλήρωση αρκετών συνθηκών, ώστε το αποτέλεσμα του ελέγχου να θεωρείται αξιόπιστο. Αντιθέτως, οι μη παραμετρικοί έλεγχοι δεν βασίζονται σε κάποια κατανομή και μπορούν να εφαρμοστούν ακόμη και εάν οι παραμετρικές συνθήκες εγκυρότητας δεν πληρούνται. Ως εκ τούτου έχουν πιο ευρεία εφαρμογή.

1.3.2 Τύποι ανάλυσης

Η ανάλυση χρονοσειρών μπορεί να πραγματοποιηθεί με πολλούς διαφορετικούς τρόπους ανάλογα με το πεδίο εφαρμογής, αλλά και τους στόχους του εκάστοτε αναλυτή. Σε περιπτώσεις που σκοπός είναι η κατανόηση των δεδομένων γίνεται διερευνητική ανάλυση. Σε πεδία, όπως αυτό τη μηχανικής μάθησης βρίσκουν ιδιαίτερη εφαρμογή τύποι ανάλυσης όπως είναι η κατηγοριοποίηση (classification) και η συσταδοποίηση (segmentation). Ενώ στις περισσότερες των περιπτώσεων απαιτείται ο συνδυασμός περισσότερων του ενός τύπων για την επίτευξη του σκοπού της ανάλυσης.

1.3.2.1 Διερευνητική ανάλυση (Exploratory analysis)

Το σύνολο των πρακτικών, τόσο γραφικών όσο και μαθηματικών ή προγραμματιστικών, οι οποίες μας βοηθούν στην καλύτερη κατανόηση των δεδομένων καλείται διερευνητική ανάλυση. Η διενέργεια αυτού του είδους της ανάλυσης αποτελεί συνηθισμένη (και όχι άδικα) πρακτική, πριν την πραγματοποίηση οποιουδήποτε άλλου είδους ανάλυσης [14].

Τα διάφορα εργαλεία της διερευνητικής ανάλυσης μπορούν να χωριστούν σε δύο κύριες κατηγορίες: Τις γραφικές μεθόδους, οι οποίες περιλαμβάνουν μια συνολική απεικόνιση των δεδομένων μέσω γραφημάτων και τις ποσοτικές ή μη-γραφικές μεθόδους. Αυτές οι δυο μέθοδοι, διαιρούνται με τη σειρά τους σε δυο ακόμη κατηγορίες: Τις μονοπαραμετρικές μεθόδους, οι οποίες εξετάζουν μια μεταβλητή τη φορά και τις πολυπαραμετρικές, οι οποίες εξετάζουν περισσότερες.

- Γραφικές - Μονοπαραμετρικές: Σε αντίθεση με τις ποσοτικές, οι γραφικές μέθοδοι προσφέρουν μια συνολική εικόνα για τα υπό εξέταση δεδομένα. Οι τρεις κύριοι τύποι ανάλυσης που υπάγονται σε αυτές τις μεθόδους είναι το ιστόγραμμα (histogram), το φυλλογράφημα (stem and leaf plot) και το θηκόγραμμα (box plot). Το ιστόγραμμα αναπαριστά το σύνολο των περιπτώσεων για ένα εύρος τιμών. Το φυλλογράφημα μαζί με τις τιμές των δεδομένων δείχνει και το σχήμα της κατανομής αυτών. Το θηκόγραμμα απεικονίζει γραφικά και συνοπτικά το ελάχιστο, το μέγιστο, τη διάμεσο, το πρώτο τεταρτημόριο και το τρίτο τεταρτημόριο.
- Γραφικές - Πολυπαραμετρικές: Αυτός ο τύπος ανάλυσης παρουσιάζει τη σχέση μεταξύ δύο ή και περισσότερων συνόλων δεδομένων. Χαρακτηριστικό

παράδειγμα αποτελεί το ραβδόγραμμα, όπου κάθε ομάδα αναπαριστά ένα επίπεδο μιας εκ των μεταβλητών και κάθε ράβδος μέσα στην ομάδα αναπαριστά τα επίπεδα άλλων μεταβλητών.

- Ποσοτικές - Μονοπαραμετρικές: Αποτελούν την απλούστερη μορφή ανάλυσης δεδομένων μεταξύ των τεσσάρων επιλογών. Σε αυτό το είδος ανάλυσης, τα υπό ανάλυση δεδομένα αποτελούνται από μια μόνο μεταβλητή. Κύριος σκοπός της ανάλυσης είναι η περιγραφή των δεδομένων, καθώς και η εξέταση για ύπαρξη προτύπων.
- Ποσοτικές - Πολυπαραμετρικές: Οι ποσοτικοί πολυπαραμετρικοί τύποι ανάλυσης δεδομένων αναπαριστούν τη σχέση μεταξύ πολλαπλών μεταβλητών είτε μέσω διασταυρούμενων πινάκων (cross tabulation) είτε μέσω χρήσης στατιστικών μεθόδων.

1.3.2.2 Προσαρμογή καμπύλης (Curve fitting)

Στα μαθηματικά, η προσαρμογή καμπύλης είναι η διαδικασία κατασκευής μιας καμπύλης ή μαθηματικής συνάρτησης, η οποία έχει την καλύτερη προσαρμογή σε μια σειρά παρατηρήσεων. Επί της ουσίας αποτελεί ένα τρόπο βελτιστοποίησης, ο οποίος έχει στόχο την εύρεση των βέλτιστων παραμέτρων για τη δημιουργία μιας συνάρτησης που παρουσιάζει την καλύτερη προσαρμογή σε ένα δοθέν σύνολο παρατηρήσεων [15].

Η προσαρμογή καμπύλης μπορεί να χωριστεί σε δυο κύριους τύπους:

- Παρεμβολή (Interpolation): Στόχος είναι η εύρεση μιας συνάρτησης η οποία να ταιριάζει απόλυτα στα δεδομένα. Ωστόσο, το γεγονός ότι προϋποθέτει την απουσία σφάλματος μέτρησης, συνεπάγεται ότι έχει περιορισμένη εφαρμογή στην πράξη.
- Εξομάλυνση (Smoothing): Σε αυτή την περίπτωση δεν αναζητούμε ακριβή προσαρμογή, αλλά αποδεχόμαστε το περιθώριο λάθους και επιτρέπουμε στα σημεία δεδομένων να είναι κοντά στη γραμμή του μοντέλου μας, αλλά όχι απαραίτητα να συμπίπτουν απόλυτα.

Η μελέτη της σχέσης μεταξύ μιας εξαρτημένης μεταβλητής Y και μιας ή περισσότερων ανεξάρτητων μεταβλητών X , καλείται Ανάλυση Παλινδρόμησης. Ανάλογα με τον τρόπο προσέγγισης αυτής της σχέσης μπορούμε να προσδιορίσουμε δυο ευρύτερες κατηγορίες μοντέλων, τα γραμμικά και τα μη-γραμμικά.

Μοντέλο γραμμικής παλινδρόμησης

Συχνά σκεφτόμαστε τη σχέση μεταξύ δυο μεταβλητών ως μια ευθεία γραμμή, που σημαίνει ότι γνωρίζοντας την τιμή της μεταβλητής X μπορούμε να προσδιορίσουμε με ακρίβεια την τιμή της μεταβλητής Y . Τα δεδομένα τα οποία παρουσιάζουν αυτού του είδους τη σχέση μπορούν να αναπαρασταθούν με μια ευθεία γραμμή. Το μοντέλο αυτό ονομάζεται γραμμική παλινδρόμηση (linear regression) και εκφράζεται από τη σχέση (1):

$$(1) \quad Y = B_0 + B_1X,$$

όπου οι B_0 και B_1 καλούνται παράμετροι και είναι γνωστές σταθερές που συνδέουν τις μεταβλητές Y και X . Η σχέση (1) είναι ακριβής και τέτοιου είδους σχέσεις είναι δύσκολο να βρεθούν σε πρακτική εφαρμογή. Συνήθως, έχουμε να αντιμετωπίσουμε εμπειρικές προσεγγίσεις οι οποίες καθορίζονται από τα παρατηρηθέντα δεδομένα. Αυτού του είδους οι σχέσεις αναπαρίστανται ως εξής:

$$(2) \quad Y_i = B_0 + B_1X_i + e_i,$$

Όπου οι Y_i και X_i είναι οι i -οστές παρατηρηθείσες τιμές της εξαρτημένης και της ανεξάρτητης μεταβλητής, αντιστοίχως. Οι B_0 και B_1 είναι άγνωστες σταθερές οι οποίες πρέπει να εκτιμηθούν. Το e_i αναπαριστά το σφάλμα (απόκλιση) στην i -οστή μονάδα παρατήρησης. Στις περιπτώσεις που έχουμε μόνο μια ανεξάρτητη (ή ερμηνευτική) μεταβλητή, τότε το μοντέλο μας είναι αυτό της απλής γραμμικής παλινδρόμησης. Ενώ, όταν η μεταβλητή απόκρισης (ή εξαρτημένη) επηρεάζεται από περισσότερες από μια ερμηνευτικές μεταβλητές, τότε ακολουθείται το μοντέλο πολλαπλής γραμμικής παλινδρόμησης.

Η εκτίμηση των παραμέτρων για τα γραμμικά μοντέλα γίνεται με χρήση των μεθόδων ελαχίστων τετραγώνων και μέγιστης πιθανοφάνειας [16]. Η μέθοδος ελαχίστων τετραγώνων ελαχιστοποιεί το άθροισμα τετραγώνων των αποκλίσεων (σφαλμάτων), Q , το οποίο δίνεται από τον τύπο:

$$(3) \quad Q = \sum_{i=1}^n (Y_i - \hat{Y}_i)^2,$$

όπου $\hat{Y}_i = f(X_i; \hat{B}_1, \hat{B}_2, \dots)$ είναι η προβλεφθείσα τιμή για ένα συγκεκριμένο X_i κάνοντας χρήση των παραμέτρων που εκτιμήθηκαν μέσω των ελαχίστων τετραγώνων.

Εάν τα σφάλματα ακολουθούν την κανονική κατανομή, τότε οι εκτιμητές ελαχίστων τετραγώνων συμπίπτουν με τους εκτιμητές μεγίστης πιθανοφάνειας. Σε μοντέλα όπως αυτό της σχέσης (2) οι τιμές των παραμέτρων που ελαχιστοποιούν το σφάλμα μπορούν να βρεθούν με τον εξής τρόπο: Έστω ότι, η $f()$ είναι μια απλή συνάρτηση όπως αυτή της σχέσης (2). Τότε είναι δυνατή η εύρεση μιας αναλυτικής λύσης μέσω της διαφοροποίησης του Q ως προς τις σταθερές B_1, B_2, \dots, B_n , θέτοντας τα μερικά παράγωγα της συνάρτησης ίσα με το μηδέν και λύνοντας τις κανονικές εξισώσεις που προκύπτουν.

Μοντέλο μη γραμμικής παλινδρόμησης

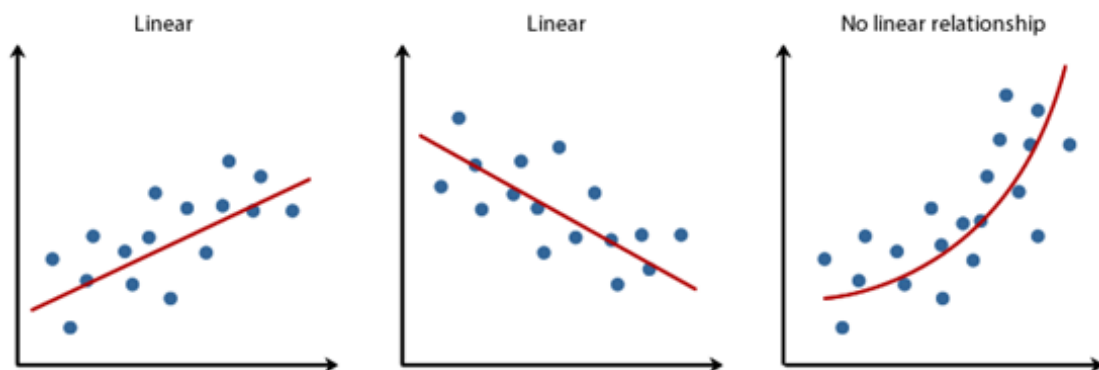
Μη γραμμικά είναι τα μοντέλα τα οποία αφορούν παραμέτρους οι σχέσεις των οποίων δεν είναι γραμμικές και ούτε είναι εφικτό να καταστούν γραμμικές, μέσω κάποιου μετασχηματισμού. Μια γενική αναπαράσταση αυτών των μοντέλων είναι η εξής:

$$(4) \quad Y_i = f(X_i, e_i; B_1, B_2, \dots, B_n),$$

όπου B_1, B_2, \dots, B_n είναι οι παράμετροι προς εκτίμηση και το e_i αναπαριστά το σφάλμα.

Σημειώνεται, πως το σφάλμα e_i δεν είναι κατ' ανάγκη σφαιρικό όπως στη σχέση (2), ωστόσο αποτελεί την ίσως συνηθέστερη μορφή. Ένα παράδειγμα σφαιρικού μοντέλου αποτελεί το εξής:

$$(5) \quad Y_i = B_0 e^{B_1(X_i)} + e_i$$



Εικόνα 5 – Παραδείγματα μοντέλων γραμμικής και μη γραμμικής παλινδρόμησης
Πηγή: <https://statistics.laerd.com/spss-tutorials/linear-regression-using-spss-statistics.php>



Τα γραμμικά μοντέλα συνήθως προτιμώνται συγκριτικά με τα μη γραμμικά μοντέλα για δυο κύριους λόγους:

- 1) Πρώτον, τα γραμμικά μοντέλα είναι ευκολότερα στη διαχείριση από μαθηματικής άποψης, αφού οι παράμετροι μπορούν να εκτιμηθούν με ρητές (explicit) εκφράσεις.
- 2) Δεύτερον, είναι συχνό το φαινόμενο ο αναλυτής να μη γνωρίζει την πραγματική μορφή της σχέσης μεταξύ των παραμέτρων και έτσι προχωρά σε μια προσέγγιση αυτής. Το γραμμικό μοντέλο είναι το συνηθέστερο σημείο εκκίνησης για αυτού του είδους τη διαδικασία.

Η εκτίμηση των παραμέτρων για τα μη γραμμικά μοντέλα μπορεί να γίνει με τις μεθόδους ελαχίστων τετραγώνων και μεγίστης πιθανοφάνειας, όπως και στα γραμμικά μοντέλα. Ωστόσο, ελάχιστα μη γραμμικά μοντέλα μπορούν να εκτιμηθούν με χρήση κανονικών εξισώσεων για την ελαχιστοποίηση του σφάλματος και για αυτό το λόγο χρησιμοποιούνται οι εξής διαδικασίες [16], [17]:

- Μέθοδος Gauss-Newton: Η μέθοδος Gauss-Newton, ή γραμμικοποίηση, χρησιμοποιεί την επέκταση μιας σειράς Taylor με σκοπό την προσέγγιση του μη γραμμικού μοντέλου με γραμμικούς όρους. Αυτοί μπορούν να εφαρμοστούν σε ένα μοντέλο γραμμικής παλινδρόμησης με σκοπό την εύρεση δοκιμαστικών παραμέτρων, οι οποίες με τη σειρά τους θα χρησιμοποιηθούν για τη δημιουργία νέων γραμμικών όρων. Η διαδικασία αυτή συνεχίζεται έως ότου βρεθεί λύση.
- Μέθοδος απότομης καθόδου: Η συγκεκριμένη μέθοδος αναζητεί το ελάχιστο άθροισμα τετραγώνων των σφαλμάτων επαναπροσδιορίζοντας διαρκώς την κατεύθυνση των αλλαγών στην εκτίμηση των παραμέτρων. Συγκεκριμένα, είναι πολύ χρήσιμη όταν οι αρχικές τιμές των παραμέτρων δεν είναι ιδιαίτερα βοηθητικές.
- Μέθοδος Levenberg-Marquadt: Ο συγκεκριμένος αλγόριθμος επί της ουσίας συνδυάζει τα καλύτερα χαρακτηριστικά των προηγούμενων δυο μεθόδων. Ωστόσο, καταλήγει στην εύρεση ενός τοπικού ελάχιστου, το οποίο δεν είναι και απαραίτητα το ολικό ελάχιστο.

1.3.2.3 Κατηγοριοποίηση (Classification)

Όπως έχουμε ήδη αναφέρει τα δεδομένα χρονοσειράς χρησιμοποιούνται ευρέως για να απεικονίσουν διάφορους τύπους δεδομένων, όπως τιμές μετοχών, δεδομένα καιρού, βιολογικές παρατηρήσεις κ.ά. Οι προσπάθειες εξαγωγής χρήσιμων συμπερασμάτων μέσα από τέτοιου είδους δεδομένα διαρκώς εντείνονται και αποκτούν μεγαλύτερη σημασία.

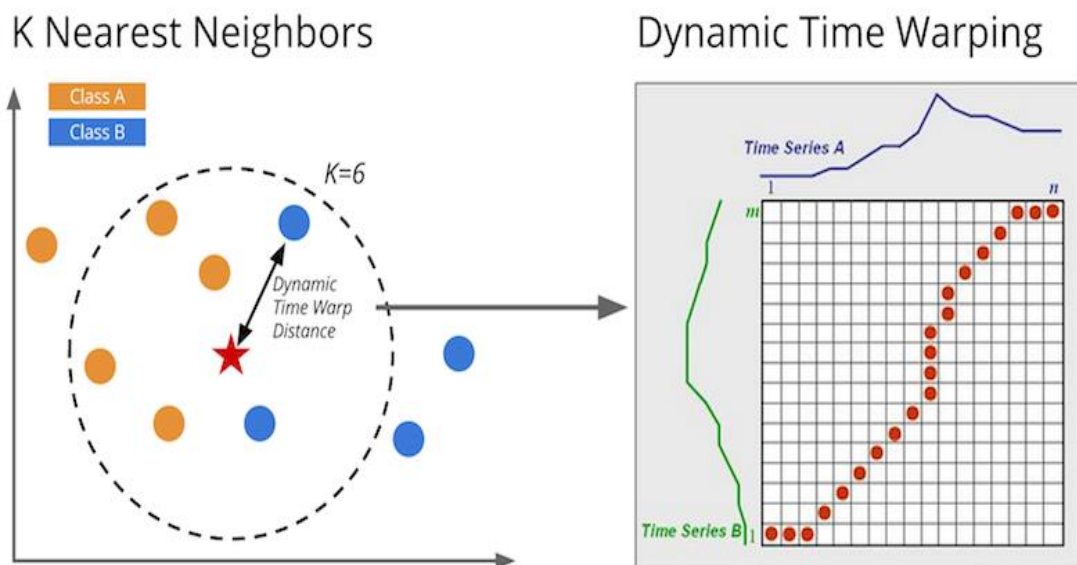
Η κατηγοριοποίηση χρονοσειρών είναι μια μορφή επιβλεπόμενης μάθησης που στοχεύει στην εύρεση της σχέσης ανάμεσα σε ένα γνώρισμα στόχο με ονομαστικές τιμές και σε ένα σύνολο άλλων γνωρισμάτων και θεωρείται ως μια από τις πιο απαιτητικές εργασίες στην εξόρυξη δεδομένων. Οι ερευνητές στην προσπάθειά τους να επιτύχουν την καλύτερη δυνατή κατηγοριοποίηση των δεδομένων χρονοσειρών έχουν εφαρμόσει πληθώρα μεθόδων [18].

Οι διάφορες προσεγγίσεις κατατάσσονται σε τρεις κύριες κατηγορίες: κατηγοριοποίηση με βάση την απόσταση, κατηγοριοποίηση με βάση τα χαρακτηριστικά και οι μέθοδοι συνόλων (ensemble-based) [19], [20]:

- Κατηγοριοποίηση με βάση την απόσταση: Η βασική ιδέα αυτής της προσέγγισης είναι η μέτρηση της ομοιότητας μεταξύ δυο χρονοσειρών. Οι συνηθέστεροι αλγόριθμοι που ακολουθούν αυτή τη λογική είναι οι K-πλησιέστεροι γείτονες (K-nearest neighbours – KNN), οι μηχανές διανυσματικής υποστήριξης (support vector machines – SVM) με πυρήνες βασιζόμενους στην ομοιότητα και η δυναμική χρονική στρέβλωση (dynamic time warping – DTW).
- Κατηγοριοποίηση με βάση τα χαρακτηριστικά: Οι αλγόριθμοι κατηγοριοποίησης με βάση τα χαρακτηριστικά προσπαθούν να εξάγουν ένα σύνολο χαρακτηριστικών, τα οποία θα δημιουργήσουν ένα σάκο λέξεων (Bag-of-Words - BoW) που θα χρησιμοποιηθεί από τους κατηγοριοποιητές. Τέτοιου είδους αλγόριθμοι είναι οι Bag-of-Features framework (TSBF), Bag-of-SFA-Symbols (BOSS) και η μέθοδος BOSSVS.
- Μέθοδοι συνόλων: Στόχος αυτής της προσέγγισης είναι ο συνδυασμός περισσοτέρων του ενός κατηγοριοποιητών με σκοπό τη μεγιστοποίηση της ακρίβειας του μοντέλου.

Δυναμική χρονική στρέβλωση (DTW) και 1-πλησιέστερος γείτονας (1-NN)

Για τουλάχιστον μια δεκαετία, ο αλγόριθμος της δυναμικής χρονικής στρέβλωσης (Dynamic Time Warping/ DTW) σε συνδυασμό με τον αλγόριθμο του 1-πλησιέστερου γείτονα (1-nearest neighbour/ 1-NN) αποτελούσε την αποτελεσματικότερη μέθοδο κατηγοριοποίησης χρονοσειρών και συνάμα ένα σημείο αναφοράς για την αποτελεσματικότητα των υπόλοιπων μεθόδων. Ο αλγόριθμος του 1-πλησιέστερου γείτονα είναι μια απλή τεχνική κατά την οποία έχουμε ένα σύνολο εκπαίδευσης (training set). Στη συνέχεια κατηγοριοποιούμε μια νέα εισερχόμενη χρονοσειρά βρίσκοντας στα δεδομένα εκπαίδευσης τη χρονοσειρά εκείνη η οποία παρουσιάζει τη μεγαλύτερη ομοιότητα και ταξινομώντας τη στην ίδια κατηγορία. Εκτός του 1-πλησιέστερου γείτονα μπορούν να χρησιμοποιηθούν και οι k-πλησιέστεροι γείτονες, όπου βρίσκουμε τις k όμοιες χρονοσειρές και επιλέγουμε την πιο συνηθισμένη κατηγορία ανάμεσά τους.



Εικόνα 6 – Διαγραμματική απεικόνιση αλγορίθμων K-NN και DTW

Πηγή: <https://github.com/markdregan/K-Nearest-Neighbors-with-Dynamic-Time-Warping>

Ο κυριότερος, ίσως, λόγος που παλαιότερα υπήρχε μεγάλη δυσκολία να ξεπεραστεί η αποτελεσματικότητα της μεθόδου DTW + 1-NN ήταν η έλλειψη δεδομένων, αφού τα πιο σύνθετα μοντέλα χρειάζονται μεγαλύτερο όγκο δεδομένων για να μπορέσουν να εκπαιδεύσουν όλες τις παραμέτρους. Ωστόσο, τα τελευταία χρόνια έχουν προκύψει μέθοδοι οι οποίες έχουν τη δυνατότητα να αποδίδουν καλύτερα από το συνδυασμό DTW + 1-NN. Ιδιαίτερο ενδιαφέρον και ολοένα αυξανόμενη χρήση στην επίλυση

προβλημάτων κατηγοριοποίησης παρουσιάζουν οι προσεγγίσεις της βαθιάς μηχανικής μάθησης (deep learning) με χρήση τεχνητών νευρωνικών δικτύων.

1.3.2.4 Τμηματοποίηση (Segmentation)

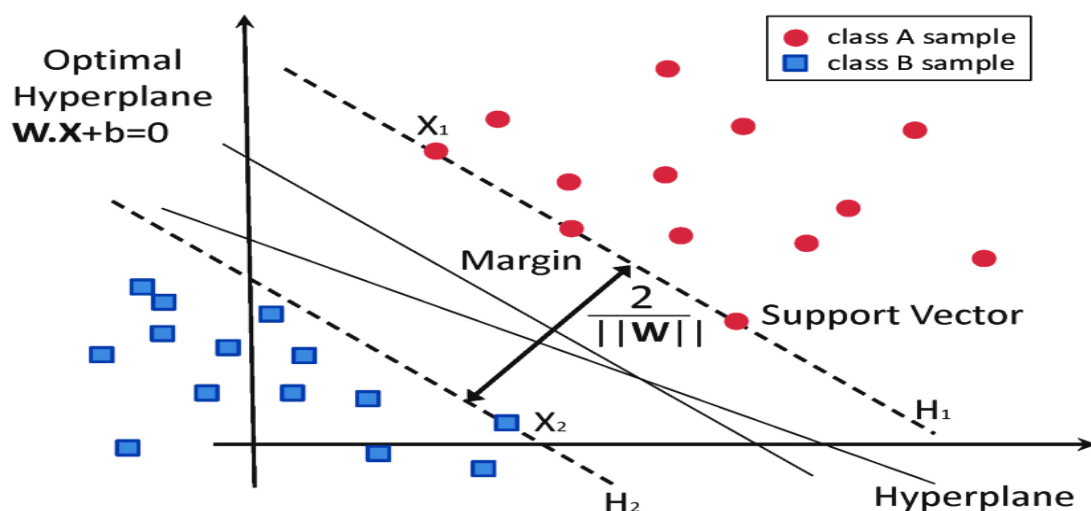
Η τμηματοποίηση είναι ένας τρόπος μοντελοποίησης των δεδομένων ο οποίος χρησιμοποιείται ευρέως και κυρίως σε τομείς όπως είναι το μάρκετινγκ και η ανάλυση δεδομένων. Κύριος στόχος της είναι η κατάτμηση της χρονοσειράς στόχου σε υποομάδες, βασιζόμενη σε κάποια κύρια κοινά χαρακτηριστικά. Παραδείγματος χάριν, εάν θέλαμε να εκτελέσουμε κάποιες εργασίες ανάλυσης σε δεδομένα παγκόσμιας γεωγραφικής κλίμακας θα μπορούσαμε να χρησιμοποιήσουμε την κατηγοριοποίηση για να εξάγουμε χρήσιμες πληροφορίες βασιζόμενοι σε χαρακτηριστικά όπως είναι η χώρα, η πόλη, η γλώσσα, ο πληθυσμός, το νόμισμα, το κλίμα κ.ά.

Στις περισσότερες των περιπτώσεων η τμηματοποίηση χρησιμοποιείται για δεδομένα τα οποία δεν είναι κατηγοριοποιημένα (unlabeled). Με βάση τις σχέσεις που παρατηρούνται μεταξύ των παρατηρήσεων, αυτές κατακερματίζονται περαιτέρω σε διαφορετικές συστάδες (ή ομάδες). Ακολουθεί συνοπτική παρουσίαση μερικών από τις πιο διαδεδομένες μεθόδους τμηματοποίησης [21], [22], [23], [24]:

- Αλγόριθμος K-μέσων (K-means): Ο αλγόριθμος των K-μέσων περιλαμβάνει ένα σύνολο k συστάδων (clusters) και ορίζει κάθε παρατήρηση σε μία συστάδα. Οι συστάδες αυτές αποτελούνται από παρόμοιες παρατηρήσεις, η ομοιότητα των οποίων βασίζεται στη μεταξύ τους απόσταση. Ο ορισμός κάθε συστάδας αποφασίζεται από τη θέση του κέντρου της στο n-διάστατο χώρο. Αυτή η θέση ονομάζεται κεντροειδής (centroid) της συστάδας. Αφότου αποφασιστεί ο αριθμός των συστάδων αρχικοποιούμε τυχαία του κεντροειδείς κάθε συστάδας. Στη συνέχεια, με βάση τον αλγόριθμο, η κάθε παρατήρηση ταξινομείται στην πλησιέστερη σε αυτή συστάδα.
- Χωρική Ομαδοποίηση Εφαρμογών με Θόρυβο (Density-Based Spatial Clustering of Applications with Noise - DBSCAN): Ο αλγόριθμος DB scan είναι ένας αλγόριθμος τμηματοποίησης με βάση την πυκνότητα. Μια συστάδα βασισμένη στην πυκνότητα αποτελεί ένα μέγιστο σύνολο σημείων συνδεδεμένων με βάση την έννοια της προσιτότητας της πυκνότητας (density reachability). Ο αλγόριθμος χρησιμοποιεί δυο παραμέτρους, το ε (eps) και το

minPoints, δηλαδή τον ελάχιστο αριθμό σημείων που απαιτούνται για τη δημιουργία μιας πυκνής περιοχής:

- eps: Ορίζει ποια είναι η απαιτούμενη απόσταση μεταξύ δοθέντων σημείων, ούτως ώστε αυτά να θεωρηθούν μέλη μιας συστάδας. Αυτό πρακτικά σημαίνει πως εάν η απόσταση μεταξύ δυο σημείων είναι μικρότερη ή ίση από το eps, τότε αυτά τα σημεία θεωρούνται γείτονες.
- minPoints: Ορίζει τον ελάχιστο απαιτούμενο αριθμό σημείων για τη διαμόρφωση μιας πυκνής περιοχής (dense region).
- Μηχανή Διανυσματικής Υποστήριξης (Support Vector Machine): Στις μηχανές διανυσματικής υποστήριξης, τα σημεία δεδομένων χαρτογραφούνται από το χώρο δεδομένων σε ένα χώρο υψηλής διάστασης μέσω ενός Γκαουσιανού πυρήνα (Gaussian kernel). Στο χώρο των χαρακτηριστικών, αναζητείται η μικρότερη σφαίρα που περικλείει την εικόνα των δεδομένων. Στη συνέχεια, αυτή η σφαίρα χαρτογραφείται πίσω στο χώρο των δεδομένων όπου δημιουργεί ένα σύνολο περιγραμμάτων που περικλείουν δεδομένα και τα οποία ερμηνεύονται ως τα όρια της συστάδας. Όσο το πλάτος του Γκαουσιανού πυρήνα μειώνεται, ο αριθμός των αποσυνδεδεμένων περιγραμμάτων στο χώρο των δεδομένων αυξάνεται, οδηγώντας έτσι σε αυξανόμενο αριθμό συστάδων και κατά συνέπεια περαιτέρω τμηματοποίηση.



Εικόνα 7 – Κατηγοριοποίηση δεδομένων με χρήση Μηχανής Διανυσματικής Υποστήριξης

Πηγή: *Hard-Rock Stability Analysis for Span Design in Entry-Type Excavations with Learning Classifiers* - Scientific Figure on ResearchGate. Available from:
https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323 [accessed 20 Aug, 2021]

- Ιεραρχική ομαδοποίηση (Agglomerative clustering/ Hierarchical clustering): Η βασική ιδέα της ιεραρχικής ομαδοποίησης βασίζεται στο γεγονός πως μια παρατήρηση σχετίζεται περισσότερο με τις πλησιέστερες σε αυτή παρατηρήσεις. Μια συστάδα μπορεί εν πολλοίς να περιγραφεί ως η μέγιστη απόσταση που απαιτείται για τη σύνδεση των μερών που την απαρτίζουν. Διαφορετικές συστάδες δημιουργούνται σε διαφορετικές αποστάσεις και μπορούν να αναπαρασταθούν μέσω ενός δενδρογράμματος εξ ου και η ονομασία της ιεραρχικής ομαδοποίησης.

1.3.2.5 Προβλεπτική ανάλυση (Predictive analysis)

Στα μαθηματικά και ειδικότερα στη στατιστική, οι προβλέψεις μελλοντικών τιμών καλούνται παρέκταση (extrapolation). Αυτός ο τύπος εκτίμησης βρίσκει εφαρμογή σε πολλά σύγχρονα πεδία και αναφέρεται ως προβλεπτική ανάλυση χρονοσειρών. Η πρόβλεψη περιλαμβάνει τη χρήση παρελθοντικών δεδομένων με σκοπό τη χρησιμοποίηση αυτών για την πρόβλεψη μελλοντικών παρατηρήσεων.

Μπορούμε να πούμε πως οι προεξέχουσες μέθοδοι πρόβλεψης εμπίπτουν σε δυο κύριες κατηγορίες: στις στατιστικές μεθόδους και στις τεχνικές εύκαμπτης υπολογιστικής (soft computing). Όπως έχει ήδη αναφερθεί, τα νευρωνικά δίκτυα ξεχωρίζουν από τις υπόλοιπες μεθόδους soft computing για την προβλεπτική τους ικανότητα και αποτελούν κύριο αντικείμενο της παρούσας εργασίας. Επειδή, λοιπόν, το επόμενο κεφάλαιο είναι εξ ολοκλήρου αφιερωμένο στα τεχνητά νευρωνικά δίκτυα, σε αυτή την ενότητα θα γίνει μια σύντομη αναφορά σε δυο εκ των σημαντικότερων στατιστικών μεθόδων πρόβλεψης χρονοσειρών, τα μοντέλα ARIMA και Holt-Winters.

Αυτοπαλινδρομικό Μοντέλο Ενσωματωμένου Κινητού Μέσου Όρου - (AutoRegressive Integrated Moving Average)

Ένα Αυτοπαλινδρομικό Μοντέλο Ενσωματωμένου Κινητού Μέσου Όρου (AutoRegressive Integrated Moving Average) ή όπως είναι ευρέως γνωστό ως αρκτικόλεξο της αγγλικής πλήρους ονομασίας του, ARIMA, είναι ένα μοντέλο στατιστικής ανάλυσης το οποίο χρησιμοποιεί δεδομένα χρονοσειρών με σκοπό είτε την καλύτερη κατανόηση αυτών είτε για την πρόβλεψη μελλοντικών τάσεων. Αναπτύχθηκε κατά τη δεκαετία του 1970 από τους George Box και Gwilym Jenkins (για αυτό και συναντάται συχνά με την ονομασία «Μοντέλο Box-Jenkins»), σε μια

προσπάθεια να περιγράψουν τις αλλαγές που προκύπτουν σε δεδομένα χρονοσειρών, χρησιμοποιώντας μαθηματική προσέγγιση. Το μοντέλο ARIMA συνδυάζει τα μοντέλα αυτοπαλινδρόμησης και κινητού μέσου όρου, ενώ με την εισαγωγή του μοντέλου διαφορίσης επιτυγχάνεται η διασφάλιση της στασιμότητας της χρονοσειράς. Το ακρωνύμιο είναι περιγραφικό και αντικατοπτρίζει τα κύρια στοιχεία του μοντέλου [25], [26], [27]:

- **AR** - Autoregression (Αυτοπαλινδρόμηση): Ένα στατιστικό μοντέλο είναι αυτοπαλινδρόμο εάν οι προβλεφθείσες μελλοντικές τιμές βασίζονται σε παρελθούσες τιμές. Χαρακτηριστικό παράδειγμα αποτελεί το αντικείμενο μελέτης της παρούσας εργασίας, όπου επιχειρείται η πρόβλεψη των μελλοντικών τιμών μιας μετοχής με βάση την απόδοσή της στο παρελθόν.
- **I** - Integrated (Ολοκληρωμένο): Γίνεται χρήση της διαφορίσης για την επίτευξη της μετατροπή της χρονοσειράς σε στάσιμη.
- **MA** - Moving Average (Κινούμενος Μέσος): Το μοντέλο κινούμενου μέσου χρησιμοποιεί την εξάρτηση μεταξύ μιας παρατήρησης και της εκτίμησης σφάλματος (residual error) από ένα μοντέλο κινούμενου μέσου εφαρμοσμένου σε ύστερες παρατηρήσεις.

Για την υλοποίηση ενός μη-εποχικού μοντέλου ARIMA πρέπει να λάβουμε υπόψιν τρεις παραμέτρους p , d και q :

- **p**: Είναι η τάξη (order) του μοντέλου Αυτοπαλινδρόμησης (Autoregressive model – AR(p)). Συχνά καλείται και «τάξη υστέρησης»
- **d**: Είναι ο αριθμός των διαφορών που πρέπει να πάρουμε για να επιτύχουμε στασιμότητα.
- **q**: Είναι η τάξη του μοντέλου Κινητού Μέσου (Moving Average model – MA(q)).

Η συνάρτηση πρόβλεψης για το μοντέλο ARIMA προκύπτει ως εξής:

Έστω ότι Y είναι η αρχική σειρά, y η σειρά που προκύπτει από τη διαφορίση και d ο αριθμός των διαφορών που απαιτούνται για την επίτευξη στασιμότητας, τότε:

- Εάν $d = 0$: $y_t = Y_t$
- Εάν $d = 1$: $y_t = Y_t - Y_{t-1}$

- Εάν $d = 2$: $y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Ως προς το y , η γενική συνάρτηση πρόβλεψης που προκύπτει είναι:

$$(1) \hat{y}_t = \mu + \varphi_1 y_{t-1} + \dots + \varphi_p y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q}$$

Όπου μ είναι η σταθερά, φ οι όροι του μοντέλου αυτοπαλινδρόμησης και θ οι όροι του μοντέλου κινητού μέσου όρου, οι οποίοι με βάση των κανόνα που θεσπίστηκε από τους Box και Jenkins ορίζονται με αρνητικό πρόσημο στη συνάρτηση.

Μέθοδος εκθετικής εξομάλυνσης Holt-Winters

Η μέθοδος Holt-Winters, είναι μια στατιστική μέθοδος για πρόβλεψη χρονοσειρών, η οποία αποτελεί ουσιαστικά την προέκταση του μοντέλου εκθετικής εξομάλυνσης (exponential smoothing) το οποίο ανέπτυξε ο καθηγητής Charles C. Holt στο πανεπιστήμιο του Τέξας στο Όστιν, το 1957.

Η συγκεκριμένη μέθοδος χρησιμοποιείται πολύ συχνά για την επίλυση προβλημάτων πρόβλεψης χρονοσειρών και διακρίνεται για την ικανότητα που έχει να αντιμετωπίζει δεδομένα τα οποία παρουσιάζουν τόσο τάση όσο και εποχικότητα. Η μέθοδος αποτελεί ουσιαστικά ένα συνδυασμό τριών απλούστερων μεθόδων εκθετικής εξομάλυνσης και σταθμισμένων μέσων όρων [28], [29]:

- Σταθμισμένοι μέσοι όροι (Weighted Average): Ένας σταθμισμένος μέσος όρος είναι επί της ουσίας ο μέσος όρος n αριθμών, όπου ο κάθε αριθμός έχει ένα συγκεκριμένο βάρος και ο παρονομαστής προκύπτει ως το άθροισμα των n βαρών. Τα βάρη συνήθως ορίζονται από κάποια συνάρτηση βάρους, όπως η λογαριθμική, η τετραγωνική, η εκθετική και ούτω καθεξής. Η χρήση του μέσου όρου ως μέθοδος πρόβλεψης χρονοσειρών έχει την ιδιότητα της εξομάλυνσης της διακύμανσης των ιστορικών τιμών, ενώ γίνεται ο υπολογισμός της πρόβλεψης.
- Απλή Εκθετική Εξομάλυνση (Simple Exponential Smoothing): Σε αυτή τη μέθοδο, η πρόβλεψη μιας μελλοντικής τιμής γίνεται με τη χρήση του σταθμισμένου μέσου όρου όλων των προηγούμενων τιμών, με τα βάρη να φθίνουν εκθετικά από την πιο πρόσφατη στην πιο παλιά τιμή. Η απλή εκθετική εξομάλυνση βασίζεται στην υπόθεση ότι οι πιο πρόσφατες τιμές της



χρονοσειράς είναι αρκετά πιο σημαντικές (έχουν μεγαλύτερο βάρος) από τις παλαιότερες παρατηρήσεις. Σημειώνεται, ότι η συγκεκριμένη μέθοδος δεν μπορεί να χρησιμοποιηθεί σε δεδομένα που παρουσιάζουν είτε τάση είτε εποχικές διακυμάνσεις.

- Μέθοδος Εκθετικής Εξομάλυνσης Holt (Holt's Exponential Smoothing): Το μοντέλο του Holt πηγαίνει τη μέθοδο της εκθετικής εξομάλυνσης ένα βήμα παραπέρα, αφού επιτρέπει την προβλεπτική ανάλυση χρονοσειρών οι οποίες παρουσιάζουν το χαρακτηριστικό της τάσης. Ωστόσο, ούτε αυτό το μοντέλο μπορεί να χειριστεί εποχικές χρονοσειρές.
- Μέθοδος Εκθετικής Εξομάλυνσης Winter (Winter's Exponential Smoothing): Η μέθοδος εκθετικής εξομάλυνσης του Winter αποτελεί μια επέκταση του μοντέλου του Holt, το οποίο δίνει τη δυνατότητα στον αναλυτή να χειριστεί δεδομένα τα οποία παρουσιάζουν εποχικότητα. Το μοντέλο αυτό είναι γνωστό με την ονομασία Holt-Winters.

Η μέθοδος Holt-Winters συναντάται σε δυο παραλλαγές, οι οποίες διαφέρουν ανάλογα με τη φύση της εποχικότητας που παρουσιάζει η χρονοσειρά ενδιαφέροντος:

- Προσθετική μέθοδος (Additive method): Προτιμάται σε περιπτώσεις όπου οι εποχικές διακυμάνσεις είναι σχετικά συνεχείς και σταθερές. Με τη μέθοδο αυτή, το εποχικό στοιχείο εκφράζεται σε απόλυτους όρους στην κλίμακα των παρατηρήσεων. Όσον αφορά τη συνάρτηση επιπέδου της χρονοσειράς, αυτή προσαρμόζεται αφαιρώντας το εποχικό στοιχείο. Μέσα σε ένα έτος το άθροισμα του εποχικού στοιχείου θα προσεγγίσει το 0.
- Πολλαπλασιαστική μέθοδος (Multiplicative method): Σε αντίθεση με την προσθετική μέθοδο, προτιμάται σε περιπτώσεις όπου οι εποχικές διακυμάνσεις μεταβάλλονται αναλογικά με το επίπεδο της σειράς. Με τη μέθοδο αυτή, το εποχικό στοιχείο εκφράζεται ως ποσοστό και η εποχική προσαρμογή της χρονοσειράς γίνεται με διαίρεση από το εποχικό στοιχείο, το άθροισμα του οποίου σε διάστημα ενός έτους προσεγγίσει το m , όπου m η σχετική συχνότητα του εποχικού στοιχείου.



Κεφάλαιο 2: Τεχνητά νευρωνικά δίκτυα

2.1 Εισαγωγή

Ο ανθρώπινος εγκέφαλος αποτελείται από ένα σύνολο συνδεδεμένων δικτύων νευρώνων. Τα τεχνητά νευρωνικά δίκτυα (Artificial Neural Networks – ANN), όπως εύκολα γίνεται αντιληπτό μέσω της ονομασίας τους και όπως το οραματίστηκαν οι πρωτοπόροι της μηχανικής μάθησης επιχειρούν να αναπαράγουν τις λειτουργίες του ανθρώπινου εγκεφάλου, ο οποίος αποτελεί το κατά τεκμήριο εξυπνότερο και πολυπλοκότερο δημιούργημα της φύσης. Αυτός είναι και ο λόγος που ο κύριος όγκος της ονοματολογίας του πεδίου (συμπεριλαμβανομένου του όρου «τεχνητή νοημοσύνη») συνδέεται άμεσα με έννοιες που σχετίζονται με τον ανθρώπινο νου.

Χαρακτηριστικό γνώρισμα των τεχνητών νευρωνικών δικτύων αποτελεί η ικανότητά τους να εγκαθιδρύουν εμπειρικές σχέσεις μεταξύ ανεξάρτητων και εξαρτημένων μεταβλητών και να εξάγουν χρήσιμες και πολύπλοκες πληροφορίες από τα σύνολα δεδομένων. Οι σχέσεις μεταξύ των εξαρτημένων και ανεξάρτητων μεταβλητών μπορούν να οριστούν χωρίς αυτό να είναι απαραίτητο να αναπαρασταθεί μέσω κάποιας μαθηματικής σχέσης. Τα μοντέλα νευρωνικών δικτύων προσφέρουν συγκεκριμένα πλεονεκτήματα έναντι των παλινδρομικών μοντέλων, όπως είναι η ικανότητα που έχουν στο χειρισμό δεδομένων με θόρυβο.

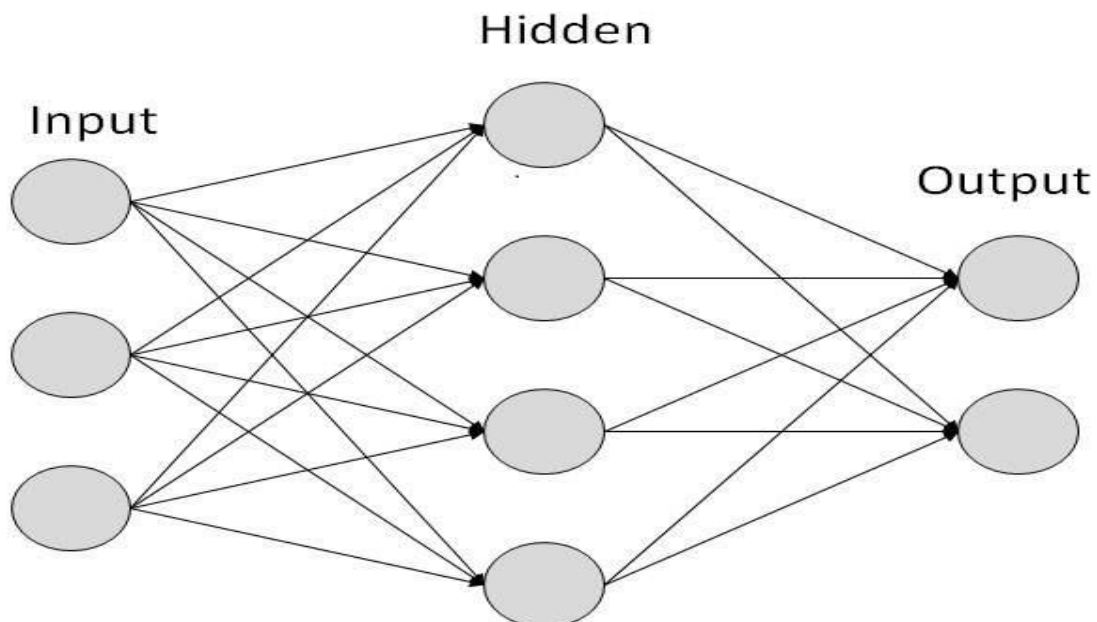
Η σύλληψη της ιδέας του πρώτου τεχνητού νευρωνικού δικτύου ανήκει στους Warren S. McCulloch και Walter Pitts, οι οποίοι σε έρευνα που δημοσιεύτηκε το 1943 ήταν οι πρώτοι που προσπάθησαν να κατανοήσουν τη λειτουργία των νευρώνων και δημιούργησαν ένα απλό νευρωνικό δίκτυο χρησιμοποιώντας ηλεκτρικά κυκλώματα. Το 1958, ο Frank Rosenblatt ανέπτυξε τον νευρώνα perceptron πηγαίνοντας τη μελέτη των McCulloch και Pitts ένα βήμα παραπέρα με την προσθήκη βαρών στην εξίσωση. Ενώ, το 1975 ο Kunihiko Fukushima ήταν ο πρώτος ο οποίος ανέπτυξε ένα πραγματικά πολυεπίπεδο νευρωνικό δίκτυο.

Ο αρχικός σκοπός των τεχνητών νευρωνικών δικτύων ήταν η δημιουργία συστήματος το οποίο θα μπορούσε να λύνει προβλήματα με τρόπο όμοιο με εκείνο του ανθρώπινου εγκεφάλου. Ωστόσο, με την πάροδο των ετών και με το ολοένα αυξανόμενο ερευνητικό ενδιαφέρον, άρχισαν να υπάρχουν αποκλίσεις από την καθαρά βιολογική προσέγγιση.

Τα τεχνητά νευρωνικά δίκτυα είναι μια από τις πιο επιδραστικές τεχνολογίες της τελευταίας δεκαετίας και χρησιμοποιούνται για διάφορες εργασίες, όπως μηχανική όραση (computer vision), αναγνώριση ομιλίας (speech recognition), μηχανική μετάφραση (machine translation) κ.ά. Όσο αυξάνεται ο όγκος των δεδομένων και μιλάμε πλέον για μεγάλα δεδομένα (big data), τόσο αυξάνονται και οι υπολογιστικές απαιτήσεις και έτσι έχουν δημιουργηθεί συστήματα βαθιάς μάθησης (deep learning), τα οποία είναι επί της ουσίας νευρωνικά δίκτυα με πολλά επίπεδα. Εδώ θα πρέπει να σημειωθεί πως συχνά συγχέονται οι έννοιες της βαθιάς μάθησης και των νευρωνικών δικτύων και χρησιμοποιούνται ως ταυτόσημες. Ωστόσο, ένα νευρωνικό δίκτυο με δυο ή τρία επίπεδα δεν μπορεί να θεωρηθεί ως ένας αλγόριθμος βαθιάς μάθησης, αλλά είναι απλώς ένα τεχνητό νευρωνικό δίκτυο.

2.2 Λειτουργία νευρωνικών δικτύων

Τα τεχνητά νευρωνικά δίκτυα αποτελούνται από διάφορα επίπεδα, κύριο συστατικό των οποίων είναι οι τεχνητοί νευρώνες. Ένα απλό νευρωνικό δίκτυο αποτελείται από τρία επίπεδα: το επίπεδο εισόδου (input layer), ένα ή περισσότερα κρυφά επίπεδα (hidden layer) και το επίπεδο εξόδου (output layer).



Εικόνα 8 - Ένα τυπικό νευρωνικό δίκτυο

Τα τεχνητά νευρωνικά δίκτυα προσομοιάζουν με κατευθυνόμενους γράφους με βάρη (weighted directed graphs), στους οποίους οι τεχνητοί νευρώνες είναι οι κόμβοι



(nodes) και οι κατευθυνόμενες ακμές (με βάρη) είναι οι συνδέσεις μεταξύ των νευρώνων εξόδου και των νευρώνων εισόδου. Κάθε σύνδεση εκτός από το σχετιζόμενο με αυτή βάρος έχει και ένα κατώφλι. Εάν το αποτέλεσμα ενός κόμβου είναι πάνω από το ορισμένο κατώφλι, τότε ο κόμβος ενεργοποιείται και στέλνει δεδομένα στο επόμενο επίπεδο του νευρωνικού δικτύου. Ειδικά, δεν μεταφέρονται δεδομένα στο επόμενο επίπεδο [30]. Τα παραπάνω αναπαριστάνεται μαθηματικά από τη σχέση:

$$y = \theta \left(\sum_{j=1}^n w_j x_j - u \right),$$

όπου θ είναι το βήμα και w_j είναι το ορισμένο βάρος που σχετίζεται με την j -οστή τιμή εισόδου. Αφότου έχουν καθοριστεί τα βάρη, τα οποία ορίζουν και τη σημαντικότητα που έχει η κάθε μεταβλητή στην έκβαση του αποτελέσματος, αυτά πολλαπλασιάζονται με τις αντίστοιχες τιμές εισόδου και στη συνέχεια αθροίζονται. Ορισμένα μοντέλα έχουν και μια σταθερά, bias, η οποία βοηθά στην καλύτερη προσαρμογή του μοντέλου στα δεδομένα:

$$Z = bias + W_1 X_1 + W_2 X_2 + \dots + W_n X_n,$$

όπου W_i τα βάρη, X_i οι τιμές εισόδου και $bias = W_0$.

2.2.1 Διάδοση πληροφορίας

Η διάδοση της πληροφορίας και συνάμα η εκπαίδευση του νευρωνικού δικτύου γίνεται με δυο τρόπους:

- Διάδοση προς τα εμπρός (forward propagation): Είναι ο υπολογισμός και η αποθήκευση των ενδιάμεσων μεταβλητών με φορά από το επίπεδο εισόδου προς το επίπεδο εξόδου (από αριστερά προς τα δεξιά) με σκοπό την κατάλληλη τροποποίηση των βαρών για την πρόβλεψη του αποτελέσματος.
- Διάδοση προς τα πίσω ή οπισθοδιάδοση (backpropagation): Είναι η μέθοδος η οποία υπολογίζει την κλίση των παραμέτρων του νευρωνικού δικτύου. Επί της ουσίας, ο αλγόριθμος αυτός «διασχίζει» το νευρωνικό δίκτυο με αντίστροφη φορά, από το επίπεδο εξόδου προς το επίπεδο εισόδου και αποθηκεύει όποιες ενδιάμεσες μεταβλητές (μερικές παραγώγους) απαιτούνται, ενώ υπολογίζει την κλίση των παραμέτρων με σκοπό την ελαχιστοποίηση του σφάλματος μεταξύ των πραγματικών τιμών και των προβλέψεων.

2.2.2 Συνάρτηση ενεργοποίησης (Activation function)

Η μεταφορά της παραγόμενης πληροφορίας από το επίπεδο εισόδου στο επίπεδο εξόδου γίνεται μέσω μιας συνάρτησης ενεργοποίησης. Η χρήση της συνάρτησης ενεργοποίησης είναι ιδιαίτερος σημαντική, αφού χωρίς αυτή το παραγόμενο αποτέλεσμα θα είναι μια απλή γραμμική συνάρτηση, η οποία επί της ουσίας είναι ένα απλό πολυώνυμο πρώτου βαθμού. Ακολουθούν ορισμένα από τα βασικότερα είδη συναρτήσεων ενεργοποίησης [31]:

2.2.2.1 Δυαδική συνάρτηση βήματος (Binary step function)

Αποτελεί την πιο απλή συνάρτηση ενεργοποίησης και μαθηματικά εκφράζεται ως:

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

Σημειώνεται, πως δεν μπορεί να χρησιμοποιηθεί σε περιπτώσεις πολλαπλής κατηγοριοποίησης της μεταβλητής στόχου. Ενώ, και η βαθμίδα (gradient) του δυαδικού βήματα είναι 0, που σημαίνει ότι μπορεί να δημιουργηθεί πρόβλημα κατά την οπισθοδιάδοση (backpropagation).

2.2.2.2 Γραμμική συνάρτηση ενεργοποίησης (Linear activation function)

Η γραμμική συνάρτηση σχετίζεται άμεσα με την τιμή εισόδου και περιγράφεται ως:

$$F_x = ax$$

Σε αντίθεση με τη δυαδική συνάρτηση βήματος, η βαθμίδα δεν είναι μηδέν, αλλά ισούται με τη σταθερά που έχει χρησιμοποιηθεί και είναι ανεξάρτητη της τιμής εισόδου x . Ωστόσο, η χρησιμότητά της θεωρείται περιορισμένη αφού σε κάθε επανάληψη χρησιμοποιείται η ίδια βαθμίδα και έτσι δεν βελτιώνεται το σφάλμα. Επιπλέον, η χρήση αυτής της συνάρτησης δεν βοηθά στον εντοπισμό σύνθετων προτύπων στα δεδομένα.

2.2.2.3 Σιγμοειδής συνάρτηση (Sigmoid activation function)

Αποτελεί τη συχνότερα χρησιμοποιούμενη συνάρτηση ενεργοποίησης. Είναι μια μη-γραμμική συνάρτηση που μετατρέπει τις τιμές σε 0 και 1 και ορίζεται ως:

$$f(x) = 1/e^{-x}$$



Η σιγμοειδής είναι μια συνεχώς διαφορίσιμη συνάρτηση και η παράγωγός της προκύπτει ως:

$$f_x = 1 - \text{sigmoid}(x)$$

Επιπλέον, η σιγμοειδής συνάρτηση δεν είναι συμμετρική γύρω από το μηδέν, που σημαίνει ότι όλες οι τιμές εξόδου από τους νευρώνες θα είναι ομόσημες. Το ζήτημα αυτό μπορεί να επιλυθεί μέσω της κλιμάκωσης της συνάρτησης.

2.2.2.4 Συνάρτηση υπερβολικής εφαπτομένης (Tanh function)

Η συνάρτηση υπερβολικής εφαπτομένης είναι παρόμοια με τη σιγμοειδή, με τη διαφορά ότι παρουσιάζει συμμετρία ως προς σημείο (παραμένει δηλαδή ίδια όταν αντικατοπτρίζεται τόσο στον άξονα των X όσο και των Y). Αυτό σημαίνει πως τα αποτελέσματα των προηγούμενων επιπέδων τα οποία θα χρησιμοποιηθούν ως μεταβλητές εισόδου στο επόμενο επίπεδο θα έχουν διαφορετικό πρόσημο. Η συνάρτηση ορίζεται μαθηματικά ως εξής:

$$f_x = 2\text{sigmoid}(2x) - 1$$

Η συνάρτηση tanh είναι συνεχής και διαφορίσιμη και παίρνει τιμές από -1 έως 1. Συγκριτικά με τη σιγμοειδή είναι πιο απότομη και γενικότερα προτιμάται αφού περιέχει κλίσεις, οι διακυμάνσεις των οποίων δεν περιορίζονται προς μια συγκεκριμένη κατεύθυνση.

2.2.2.5 Λειτουργία ενεργοποίησης ανορθωτή (Rectified linear unit – ReLU)

Η λειτουργία ενεργοποίησης ανορθωτή ή αλλιώς ReLU είναι μια μη-γραμμική συνάρτηση, η οποία παρουσιάζει το συγκριτικό πλεονέκτημα πως δεν ενεργοποιεί όλους τους νευρώνες ταυτόχρονα. Αυτό πρακτικά σημαίνει πως ένας κόμβος απενεργοποιείται μόνο όταν το αποτέλεσμα του γραμμικού μετασχηματισμού είναι 0.

Η μαθηματική σχέση που την περιγράφει ορίζεται ως εξής:

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

Σημειώνεται, πως σε ορισμένες περιπτώσεις η τιμή της κλίσης είναι 0 που σημαίνει ότι τα βάρη και το αντίστοιχο bias δεν ενημερώνονται κατά το βήμα της οπισθοδιάδοσης.

2.2.3 Συνάρτηση κόστους (Cost function)

Η συνάρτηση κόστους ή συνάρτηση σφάλματος ελέγχου χρησιμοποιείται για να υπολογίσει την απώλεια/ διαφορά μεταξύ των πραγματικών και των παρατηρηθεισών τιμών και επομένως αποτελεί ένα μέτρο για το πόσα αποτελεσματικό είναι το νευρωνικό δίκτυο στην πρόβλεψη των τιμών με βάση το δοθέν δείγμα εκπαίδευσης και τα εκτιμώμενα αποτελέσματα [32]. Η συνάρτηση κόστους είναι μοναδική τιμή και όχι διάνυσμα, επειδή μετρά την ακρίβεια του νευρωνικού δικτύου στο σύνολο του και έχει τη μορφή:

$$C(W, B, S^r, E^r),$$

όπου το W αναπαριστά τα βάρη του νευρωνικού δικτύου, το B τα biases, το S^r είναι η τιμή εισόδου ενός δείγματος εκπαίδευσης και το E^r το επιθυμητό αποτέλεσμα του αντίστοιχου δείγματος. Στην οπισθοδιάδοση, συνάρτηση κόστους χρησιμοποιείται για να υπολογίσει το σφάλμα του επιπέδου εξόδου, δ^L , ως εξής:

$$\delta_j^L = \frac{\partial C}{\partial \alpha_j^L} \sigma'(z_j^L)$$

Για να μπορεί να χρησιμοποιηθεί στην οπισθοδιάδοση, μια συνάρτηση κόστους θα πρέπει να πληροί δυο προϋποθέσεις:

- Θα πρέπει να μπορεί να γραφτεί ως μέσος όρος:

$$C = \frac{1}{n} \sum_x C_x$$

- Θα πρέπει να είναι ανεξάρτητη από κάθε τιμή ενεργοποίησης του νευρωνικού δικτύου εκτός από τις τιμές εξόδου α^L .

Μερικές από τις πιο συχνά χρησιμοποιούμενες συναρτήσεις είναι: Τετραγωνικό κόστος (γνωστή και ως Root Mean Square Error), Cross Entropy, Εκθετικό κόστος (Exponential cost), Απόσταση Hellinger (Hellinger distance), Απόκλιση Kullback–Leibler (Kullback–Leibler divergence) και Απόσταση Itakura–Saito (Itakura–Saito distance).

2.3 Κύριες κατηγορίες νευρωνικών δικτύων

Υπάρχουν πολλά και διαφορετικά είδη τεχνητών νευρωνικών δικτύων καθένα εκ των οποίων έχει τόσο πλεονεκτήματα όσο και μειονεκτήματα ανάλογα με τη χρήση του. Σε αυτή την ενότητα θα γίνει παρουσίαση ορισμένων εκ των σημαντικότερων ειδών καθώς και μια προσπάθεια κατηγοριοποίησης τους.

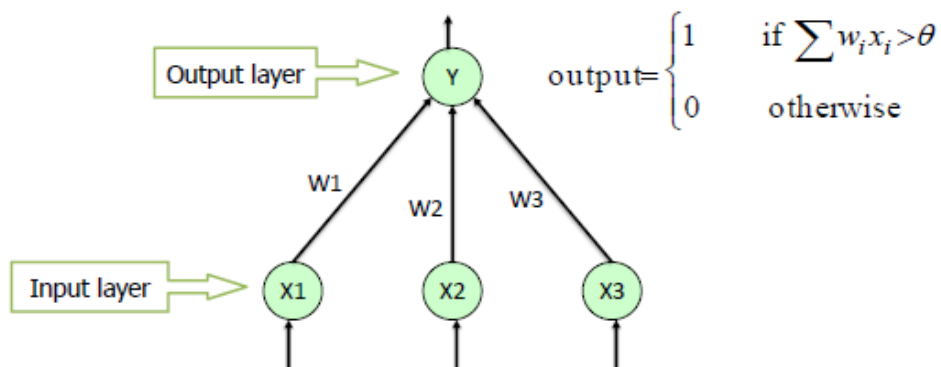
2.3.1 Εμπρός-τροφοδοτούμενα νευρωνικά δίκτυα (Feedforward neural networks - FNN)

Αποτελούν τον πρώτο τύπο τεχνητού νευρωνικού δικτύου που δημιουργήθηκε και είναι το μοντέλο με τη συχνότερη εφαρμογή σήμερα. Αυτού του είδους τα δίκτυα ονομάζονται εμπρός-τροφοδοτούμενα επειδή η ροή της πληροφορίας γίνεται μόνο προς μια κατεύθυνση (από αριστερά προς τα δεξιά, δηλαδή προς τα εμπρός). Έτσι, δεν υπάρχουν βρόχοι ανάδρασης (feedback loops), δηλαδή τα αποτελέσματα του δικτύου δεν χρησιμοποιούνται ως τιμές εισόδου σε επόμενα βήματα.

2.3.1.1 Perceptron ενός επιπέδου (Single-layer perceptron – SLP)

Όπως έχει ήδη αναφερθεί εισήχθη από τον Frank Rosenblatt το 1958 και αποτελεί την απλούστερη μορφή νευρωνικού δικτύου. Ο αλγόριθμος αυτός αποτελεί ένα γραμμικό ταξινομητή (linear classifier) που μπορεί να κατηγοριοποιήσει μόνο γραμμικά διαχωρίσιμες περιπτώσεις με δυαδικό αποτέλεσμα (0,1), δηλαδή οι τιμές εισόδου του μπορούν να χωριστούν σε μία ή σε δυο κατηγορίες.

Single Layer Perceptron



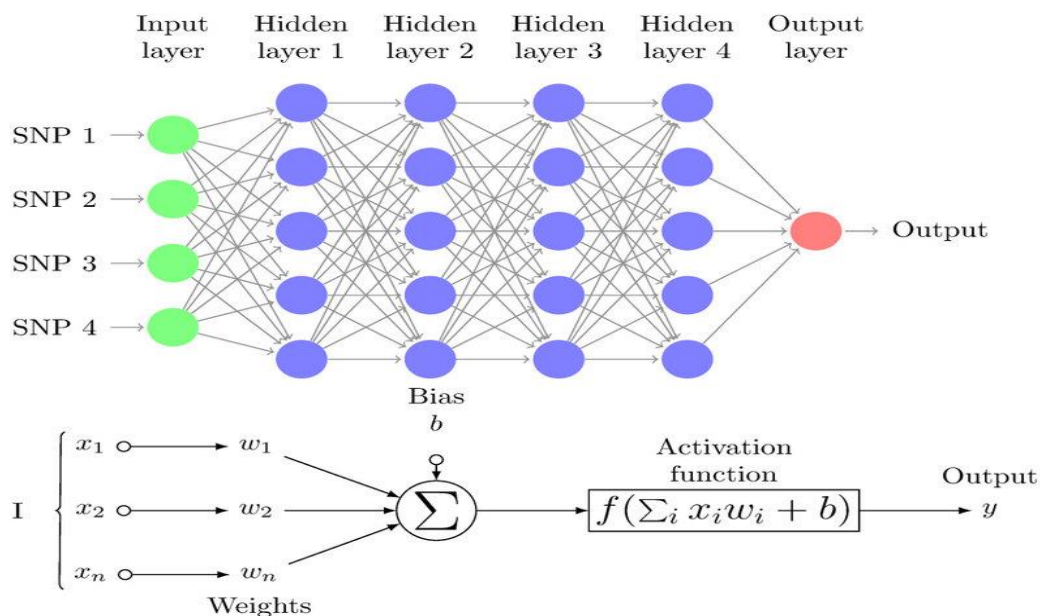
Εικόνα 9 – Single-layer perceptron

Πηγή: https://www.saedsayad.com/artificial_neural_network_bkp.htm

Το SLP δεν περιλαμβάνει κρυφά επίπεδα, τα οποία επιτρέπουν στο νευρωνικό δίκτυο να δημιουργήσει μια ιεραρχία των χαρακτηριστικών. Είναι, ως εκ τούτου, ένα ρηχό δίκτυο το οποίο δεν μπορεί να εκτελέσει μη-γραμμική κατηγοριοποίηση. Επιπλέον, δεν έχει εκ των προτέρων γνώση, που σημαίνει ότι τα αρχικά βάρη ανατίθενται τυχαία. Αφού οριστούν τα σχετιζόμενα βάρη, το μοντέλο αθροίζει όλες τις τιμές εισόδου και εάν το άθροισμα είναι μεγαλύτερο του προκαθορισμένου κατωφλιού, τότε το δίκτυο ενεργοποιείται. Όπως έχει ήδη αναφερθεί, το SLP είναι ένας γραμμικός ταξινομητής, που σημαίνει ότι δεν μπορεί να λύσει μη-γραμμικά διαχωρίσιμα προβλήματα, χαρακτηριστικό παράδειγμα των οποίων αποτελεί το πρόβλημα XOR. Για την επίλυση τέτοιου είδους προβλημάτων μπορεί να γίνει χρήση ενός perceptron πολλών επιπέδων.

2.3.1.2 Perceptron πολλών επιπέδων (Multi-layer perceptron - MLP)

Η δημοφιλέστερη κατηγορία πολυεπίπεδων και εμπρός-τροφοδοτούμενων νευρωνικών δικτύων είναι το perceptron πολλών επιπέδων (Multi-layer perceptron - MLP). Το MLP είναι ένα βαθύ νευρωνικό δίκτυο το οποίο αποτελείται από περισσότερα του ενός perceptrons, τα οποία με τη σειρά τους αποτελούνται από ένα επίπεδο εισόδου (input layer), ένα επίπεδο εξόδου (output layer) και στο ενδιάμεσο αυτών ένα ή περισσότερα κρυφά επίπεδα (hidden layer) [33].



Εικόνα 10 - MLP με 4 κρυφά επίπεδα

Πηγή: https://www.researchgate.net/figure/Multi-Layer-Perceptron-MLP-diagram-with-four-hidden-layers-and-a-collection-of-single_fig1_334609713



Η μετάδοση της πληροφορίας στα δίκτυα MLP γίνεται μονάχα προς τα εμπρός, από το επίπεδο εισόδου προς το επίπεδο εξόδου, δηλαδή. Ενώ, η εκπαίδευση των νευρώνων του μοντέλου γίνεται μέσω του αλγορίθμου της οπισθοδιάδοσης (backpropagation) και του κανόνα της αλυσίδας (chain calculus). Τα δίκτυα MLP είναι σχεδιασμένα με τρόπο τέτοιο ώστε να μπορούν να προσεγγίσουν οποιαδήποτε συνεχή συνάρτηση και να λύνουν προβλήματα τα οποία δεν είναι γραμμικά διαχωρίσιμα. Ακολουθούν οι σχέσεις που περιγράφουν του υπολογισμούς που λαμβάνουν χώρα σε κάθε νευρώνα του δικτύου τόσο στο επίπεδο εξόδου όσο και στα κρυφά επίπεδα:

$$o(x) = G(b(2) + W(2)h(x))$$

$$h(x) = \Phi(x) = s(b(1) + W(1)x),$$

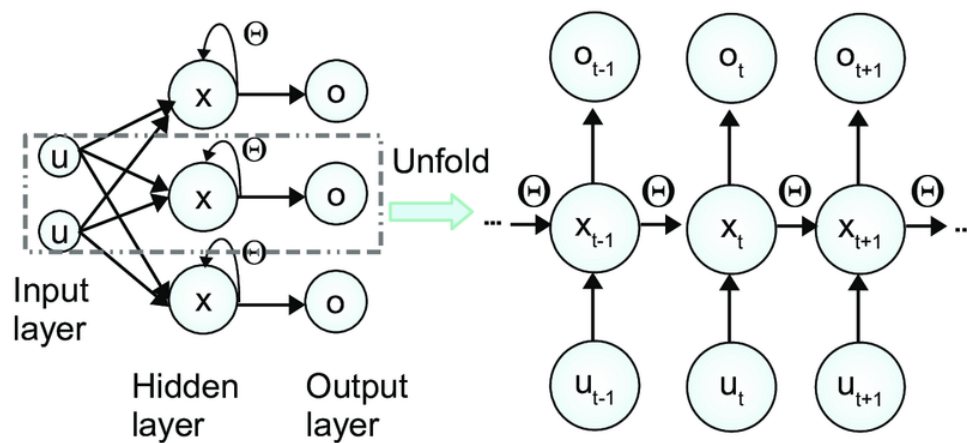
όπου $b(1)$ και $b(2)$ αναπαριστούν το bias, $W(1)$ και $W(2)$ τα βάρη και τα G και s τις συναρτήσεις ενεργοποίησης. Οι συνηθέστερες συναρτήσεις ενεργοποίησης που χρησιμοποιούνται σε μοντέλα MLP είναι η tanh και η σιγμοειδής. Τα δίκτυα MLP είναι κατάλληλα για προβλήματα πρόβλεψης κατηγοριοποίησης, όπου ανατίθεται κάποια συγκεκριμένη κατηγορία στις τιμές εισόδου.

2.3.2 Ανατροφοδοτούμενα νευρωνικά δίκτυα (Recurrent neural networks - RNN)

Σε αυτού του είδους τα νευρωνικά, το αποτέλεσμα ενός επιπέδου αποθηκεύεται και μεταφέρεται πίσω στο επίπεδο εισόδου. Με αυτό τον τρόπο, οι κόμβοι ενός συγκεκριμένου επιπέδου «θυμούνται» κάποιο μέρος της πληροφορίας των προηγούμενων βημάτων/ επαναλήψεων, το οποίο και μπορούν να χρησιμοποιήσουν αργότερα. Σε κάθε βήμα, το RNN κάνει μια σειρά από υπολογισμούς, το αποτέλεσμα των οποίων είναι γνωστό και ως κρυφή κατάσταση (hidden state) και το οποίο συνδυάζεται με την επόμενη τιμή εισόδου για να παράγει το επόμενο αποτέλεσμα [34].

Ένα από τα κύρια χαρακτηριστικά των ανατροφοδοτούμενων νευρωνικών δικτύων είναι το γεγονός ότι μοιράζονται τις ίδιες παραμέτρους σε όλα τα επίπεδα του δικτύου, σε αντίθεση με τα εμπρός τροφοδοτούμενα νευρωνικά που έχουν διαφορετικά βάρη σε κάθε κόμβο. Τα βάρη αυτά, ωστόσο, εξακολουθούν να προσαρμόζονται κατά τις διαδικασίες backpropagation και gradient descent για να διευκολύνουν τη διαδικασία της ενισχυτικής μάθησης (reinforcement learning). Τα δίκτυα RNN χρησιμοποιούν την

οπισθοδιάδοση στο χρόνο (backpropagation through time – BPTT) για να καθορίσουν τις κλίσεις. Οι βασικές αρχές της BPTT είναι οι ίδιες με τις παραδοσιακής οπισθοδιάδοσης, όπου το μοντέλο εκπαιδεύεται υπολογίζοντας σφάλματα από το επίπεδο εξόδου στο επίπεδο εισόδου. Εκεί που διαφέρει η BPTT είναι στο γεγονός ότι αθροίζει τα σφάλματα σε κάθε χρονικό βήμα, ενώ στην παραδοσιακή οπισθοδιάδοση που χρησιμοποιείται στα FNN δεν γίνεται αυτό, αφού οι παράμετροι δεν είναι κοινές μεταξύ των επιπέδων.



Εικόνα 11 – Ένα δίκτυο RNN και η ανάλυση της δομής του

Πηγή: https://www.researchgate.net/figure/A-recurrent-neural-network-and-its-unfolded-structure-where-TH-is-the-weight-matrix-x-is_fig1_333570397

Τα RNN συναντούν δυο κύρια προβλήματα στην εφαρμογή τους γνωστά ως: κλίσεις που εκρήγνυνται (exploding gradients) και εξαφανιστική διαβάθμιση (vanishing gradients), όταν οι κλίσεις τείνουν να αυξάνονται ή να μειώνονται υπερβολικά, αντίστοιχα. Η δημοφιλέστερη εφαρμογή του μοντέλου RNN είναι σε συστήματα μετατροπής κειμένου σε ομιλία (text-to-speech).

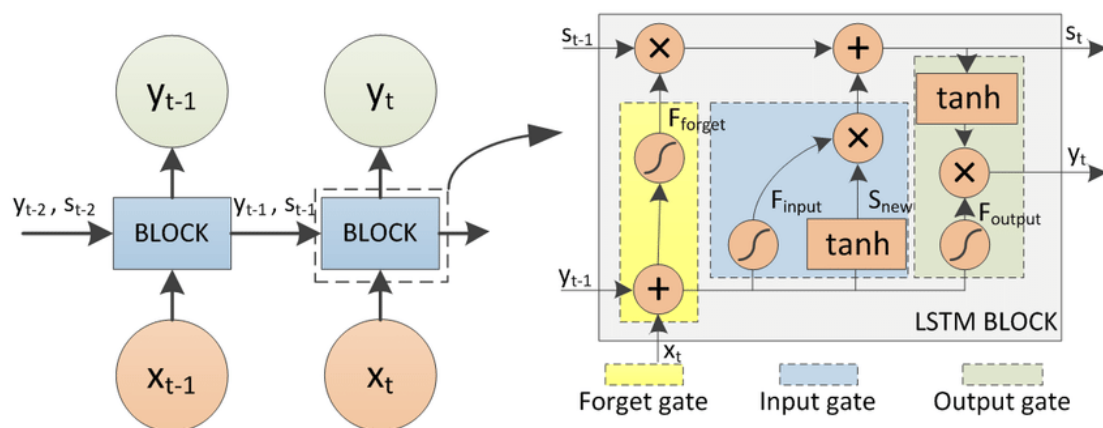
2.3.2.1 Νευρωνικά δίκτυα Μακράς Βραχύχρονης Μνήμης (Long Short-term Memory - LSTM)

Τα νευρωνικά δίκτυα Μακράς Βραχύχρονης Μνήμης (ή απλώς “LSTM”, όπως προκύπτει από το αρκτικόλεξο της αγγλικής τους ονομασίας), αποτελούν ένα είδος ανατροφοδοτούμενων νευρωνικών δικτύων το οποίο παρουσιάζει πληθώρα πρακτικών εφαρμογών. Το μοντέλο αυτό αποτελεί μία από τις πιο δημοφιλείς αρχιτεκτονικές ανατροφοδοτούμενων νευρωνικών δικτύων και παρουσιάστηκε από τους Serr

Hochreiter και Juergen Schmidhuber το 1997, ως μία λύση στο πρόβλημα της εξαφανιστικής διαβάθμισης (vanishing gradient) και το πρόβλημα των μακροχρόνιων εξαρτήσεων [35]. Όταν η πρότερη κατάσταση, η οποία επηρεάζει την παρούσα πρόβλεψη δεν είναι στο πρόσφατο παρελθόν, το μοντέλο RNN δεν είναι σε θέση να προβλέψει την παρούσα κατάσταση με ακρίβεια.

Για την αντιμετώπιση του ζητήματος της βραχυπρόθεσμης μνήμης τα μοντέλα LSTM διαθέτουν ειδικές μονάδες που καλούνται μπλοκ μνήμης (memory blocks) και τα οποία αποτελούνται από κελιά μνήμης (memory cells) στα κρυφά επίπεδα του νευρωνικού δικτύου. Τα κελιά αυτά διαθέτουν συνδέσεις που αποθηκεύουν την πρόσκαιρη κατάσταση του δικτύου, καθώς και ειδικές πολλαπλασιαστικές μονάδες, τις λεγόμενες πύλες, οι οποίες ελέγχουν τη ροή της πληροφορίας στο δίκτυο:

- Πύλη εισόδου (input gate): Ελέγχει τη ροή των ενεργοποιήσεων εισόδου στο κελί μνήμης.
- Πύλη εξόδου (output gate): Ελέγχει τη ροή των αποτελεσμάτων της ενεργοποίησης των κελιών στο υπόλοιπο δίκτυο.
- Πύλη λήθης (forget gate): Προστέθηκε αργότερα, με σκοπό να αντιμετωπίσει την αδυναμία του μοντέλου LSTM να επεξεργαστεί συνεχόμενες ροές δεδομένων, οι οποίες δεν έχουν κατακερματιστεί σε υπο-ακολουθίες. Η πύλη λήθης κλιμακώνει την εσωτερική κατάσταση του κελιού πριν την προσθέσει ως τιμή εισόδου και με αυτό τον τρόπο επιτυγχάνει την προσαρμοστική επαναφορά της μνήμης του κελιού.



Εικόνα 12 – Αρχιτεκτονική LSTM

Πηγή: https://www.researchgate.net/figure/LSTM-network-structure_fig7_328777599

Η μοντέρνα αρχιτεκτονική LSTM περιέχει συνδέσεις peephole από τα εσωτερικά κελιά στις πύλες του ίδιου κελιού με σκοπό τον προσδιορισμό του ακριβή χρόνου των αποτελεσμάτων. Ένα δίκτυο LSTM χαρτογραφεί μια ακολουθία εισόδου x σε μια ακολουθία εξόδου y υπολογίζοντας τις ενεργοποιήσεις των μονάδων του δικτύου επαναλαμβάνοντας τις ακόλουθες εξισώσεις από το $t=1$ έως το T :

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f) \quad (2)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_{t-1} + b_o) \quad (4)$$

$$m_t = o_t \odot h(c_t) \quad (5)$$

$$y_t = \varphi(W_{ym}m_t + b_y) \quad (6),$$

όπου οι όροι W είναι διαγώνιοι πίνακες των βαρών των συνδέσεων peephole, το b δηλώνει το bias, το σ είναι η λογιστική σιγμοειδής συνάρτηση. Τα i , f , o αντιστοιχούν στις πύλες εισόδου, λήθης και εξόδου, αντίστοιχα. Το m είναι το διάνυσμα ενεργοποίησης του κελιού εξόδου και τα g και h οι συναρτήσεις ενεργοποίησης.

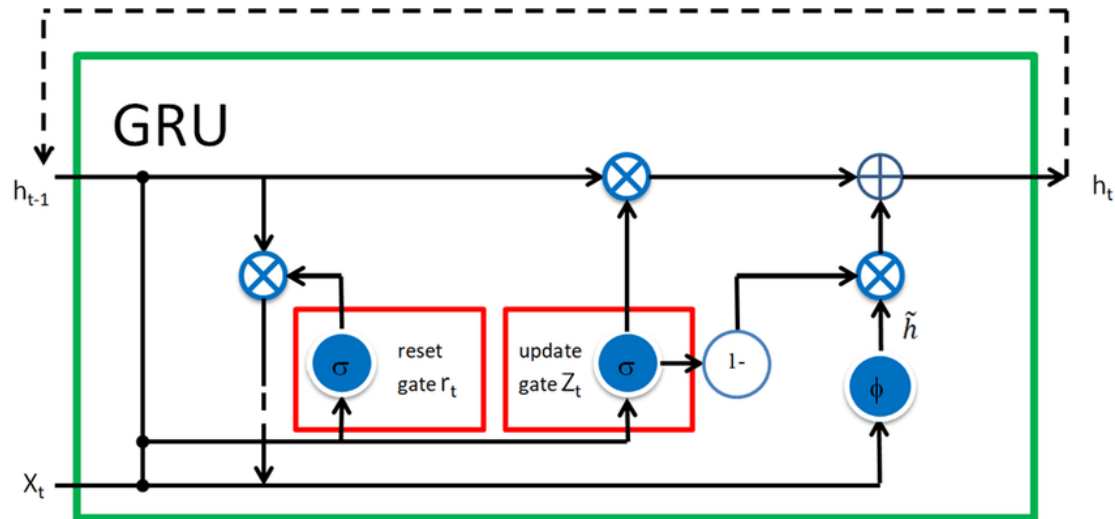
2.3.2.2 Επαναλαμβανόμενες μονάδες με πύλη (Gated recurrent units - GRUs)

Αποτελούν άλλη μια παραλλαγή ανατροφοδοτούμενου νευρωνικού δικτύου και παρουσιάζουν πολλές ομοιότητες με τα μοντέλα LSTM. Όπως γίνεται αντιληπτό και από την ονομασία τους, χρησιμοποιούν πύλες για να ελέγξουν τη ροή της πληροφορίας μεταξύ των κελιών του δικτύου. Είναι μια πρόσφατη αρχιτεκτονική η οποία εισήχθη μόλις το 2014 από τον Kyunghyun Cho [36].

Όπως έχουμε ήδη αναφέρει, τα GRU χρησιμοποιούν πύλες όπως τα LSTM για να λύσουν το πρόβλημα της εξαφανιστικής διαβάθμισης που παρουσιάζουν τα κλασσικά RNN. Ωστόσο, σε αντίθεση με τα LSTM, μεταφέρουν μόνο μια (αντί για δυο) κρυφή κατάσταση μεταξύ των χρονικών βημάτων. Η κρυφή κατάσταση στα GRU είναι σε θέση να κρατά τόσο τις βραχυπρόθεσμες όσο και τις μακροπρόθεσμες εξαρτήσεις.

Κάθε κελί του μοντέλου, σε αντίθεση με το LSTM, περιλαμβάνει δυο πύλες: την πύλη ενημέρωσης (update gate) και την πύλη επαναφοράς (reset gate). Αυτές οι πύλες είναι επί της ουσίας διανύσματα τα οποία περιέχουν τιμές από το 0 έως το 1 και οι οποίες

πολλαπλασιάζονται με τις τιμές εισόδου ή τις κρυφές καταστάσεις. Η σημαντικότητα των παρατηρήσεων είναι ανάλογη αυτών των τιμών, με το 0 σημαίνει πολύ χαμηλή σημαντικότητα και το 1 πολύ υψηλή.



Εικόνα 13 – Απεικόνιση ενός κελιού GRU

Πηγή: https://www.researchgate.net/figure/The-diagram-of-a-GRU-cell_fig3_323570767

Η δομή του GRU μπορεί να αναλυθεί σε τρία βήματα, ως εξής:

1. Αρχικά, υπολογίζεται το διάνυσμα της πύλης επαναφοράς (reset gate), το οποίο προκύπτει πολλαπλασιάζοντας την προηγούμενη κρυφή κατάσταση και την παρούσα τιμή εισόδου με τα συνδεδεμένα βάρη και αθροίζοντας τα προτού περάσουν σε μια σιγμοειδή συνάρτηση, η οποία θα μετατρέψει το αποτέλεσμα σε μια τιμή μεταξύ 0 και 1. Όταν ολοκληρω το δίκτυο εκπαιδευτεί, μέσω της οπισθοδιάδοσης τα βάρη θα ενημερωθούν αναλόγως. Το τελικό αποτέλεσμα της πύλης προκύπτει μετά την εφαρμογή μιας συνάρτησης tanh:

$$r = \tanh(\text{gate}_{reset} \odot (W_{h1} \cdot h_{t-1}) + W_{x1} \cdot x_t)$$

2. Στη συνέχεια υπολογίζεται το διάνυσμα της πύλης ενημέρωσης (update gate), το οποίο δημιουργείται με τον ίδιο τρόπο με αυτόν της πύλης επαναφοράς. Σημειώνεται, ωστόσο, ότι τα βάρη που χρησιμοποιούνται στον πολλαπλασιασμό είναι μοναδικά για κάθε πύλη κι έτσι τα αποτελέσματα που προκύπτουν ανά πύλη διαφέρουν. Το τελικό αποτέλεσμα προκύπτει ως το προϊόν Hadamard (element-wise product) του διανύσματος με την προηγούμενη κρυφή κατάσταση:

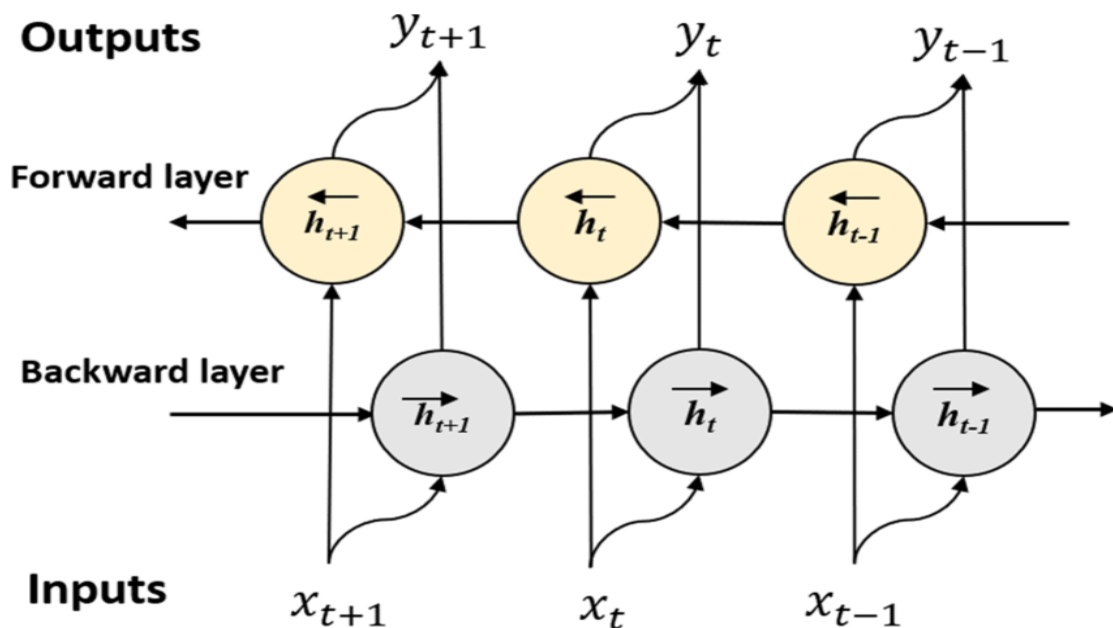
$$u = gate_{update} \odot h_{t-1}$$

3. Στο τέλος, υπολογίζεται η νέα και ενημερωμένη κρυφή κατάσταση, η οποία προκύπτει ως το άθροισμα του προϊόντος Hadamard του αποτελέσματος της πύλης επαναφοράς r και της αντίστροφης έκδοσης του αντίστοιχου αποτελέσματος της πύλης ενημέρωσης και του αποτελέσματος της πύλης ενημέρωσης στο προηγούμενο βήμα, u :

$$h_t = r \odot (1 - gate_{update}) + u$$

2.3.2.3 Αμφίδρομα επαναλαμβανόμενα νευρωνικά δίκτυα (Bidirectional recurrent neural networks - BRNN)

Τα δίκτυα BRNN συνδέουν δυο κρυφά επίπεδα με αντίθετες κατευθύνσεις σε ένα αποτέλεσμα, επιτρέποντάς τους να λαμβάνουν πληροφορίες τόσο από παρούσες όσο και από παρελθοντικές καταστάσεις. Αυτή η τεχνική συναντάται συχνότερα σε προσεγγίσεις επιβλεπόμενης μάθησης (supervised learning), επειδή είναι δύσκολο να υπολογιστεί ένα αξιόπιστο μοντέλο πιθανοτήτων [37].



Εικόνα 14 – Αρχιτεκτονική μοντέλου BRNN

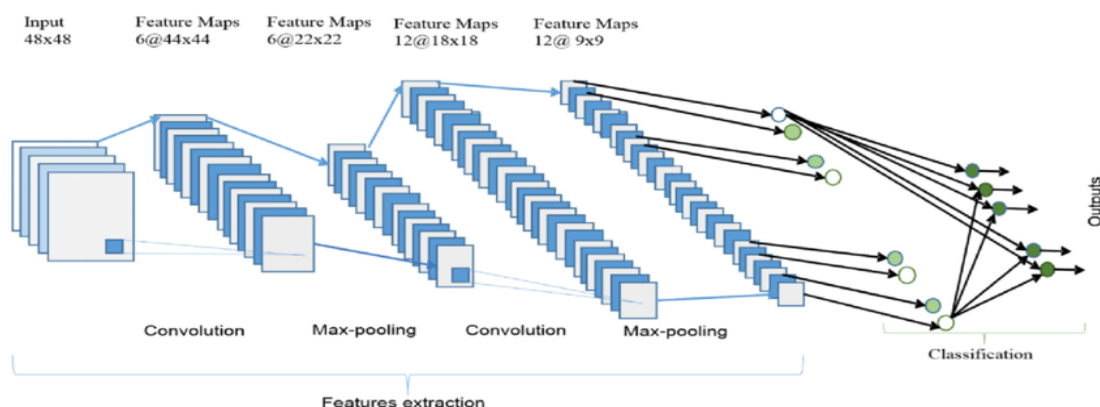
Πηγή: https://www.researchgate.net/figure/Bidirectional-Recurrent-Neural-Network-BRNN_fig5_339490382

Τα BRNN εκπαιδεύονται με παρόμοιο τρόπο με τα παραδοσιακά RNN, αφού οι δυο νευρώνες δεν αλληλοεπιδρούν μεταξύ τους. Στις περιπτώσεις που χρησιμοποιείται η

οπισθοδιάδοση, απαιτείται μια πρόσθετη διαδικασία, αφού τα επίπεδα εισόδου και εξόδου δεν μπορούν να ενημερωθούν ταυτόχρονα. Σε αντίθεση με τα παραδοσιακά RNN, τα BRNN εκπαιδεύονται για να μπορούν να προβλέψουν τόσο τις θετικές όσο και τις αρνητικές κατευθύνσεις του χρόνου την ίδια στιγμή. Οι νευρώνες χωρίζονται σε δυο κατευθύνσεις, μια για τις επόμενες καταστάσεις (forward states) και μια για τις πρότερες (backward state). Καμία από αυτές τις καταστάσεις αποτελέσματος δεν συνδέεται με τιμές εισόδου της αντίθετης κατεύθυνσης. Χρησιμοποιώντας δυο κατευθύνσεις του χρόνου την ίδια ακριβώς στιγμή, το αποτέλεσμα μπορεί να υπολογιστεί χρησιμοποιώντας δεδομένα εισόδου για το παρόν χρονικό πλαίσιο, τόσο από το παρελθόν όσο και από το μέλλον. Αντιθέτως, στα κλασσικά RNN απαιτείται ένα επιπλέον στρώμα για να μπορεί να χρησιμοποιηθεί μελλοντική πληροφορία.

2.3.2 Συνελκτικά νευρωνικά δίκτυα (Convolutional neural networks - CNN)

Τα συνελκτικά νευρωνικά δίκτυα αποτελούνται από πέντε διαφορετικούς τύπους στρωμάτων: εισόδου (input), συνέλιξης (convolution), συγκέντρωσης (pooling), πλήρως συνδεμένα (fully connected) και εξόδου (output), σε συνδυασμό με μια μέθοδο μείωσης δειγματοληψίας (downsampling). Τα CNN μοιάζουν με τα εμπρός τροφοδοτούμενα δίκτυα και λόγω της δυνατότητας που έχουν για χωρική κατηγοριοποίηση των χαρακτηριστικών του δικτύου βρίσκουν πολύ συχνή εφαρμογή στην κατηγοριοποίηση εικόνας (image classification) και στη μηχανική όραση (computer vision) [38].



Εικόνα 15 – Παρουσίαση της συνολικής αρχιτεκτονικής ενός συνελκτικού νευρωνικού δικτύου

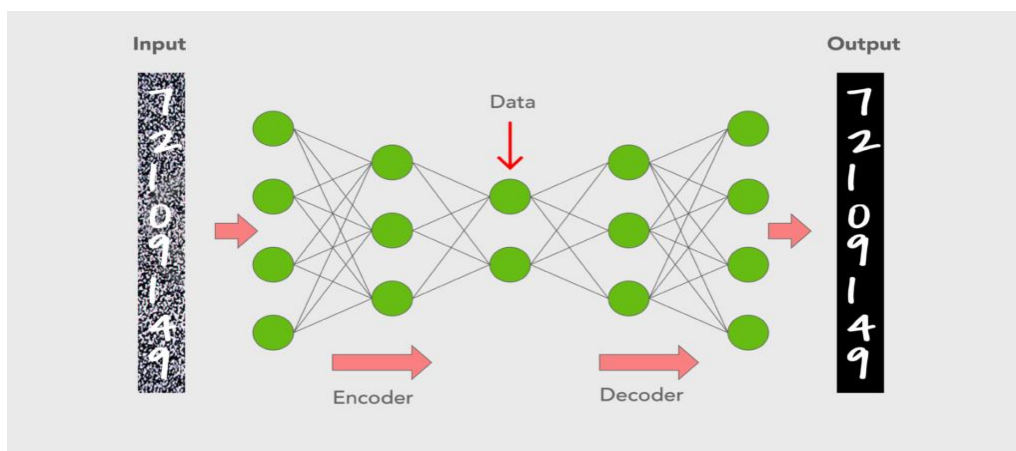
Πηγή: https://www.researchgate.net/figure/The-overall-architecture-of-the-Convolutional-Neural-Network-CNN-includes-an-input_fig4_331540139

Κάθε στρώμα εξυπηρετεί ένα συγκεκριμένο σκοπό. Τα στρώματα συνέλιξης δημιουργούν χαρτογραφήσεις χαρακτηριστικών, οι οποίες χρησιμεύουν στην ερμηνεία των τιμών εισόδου με διαφορετικούς τρόπους, ενώ τα στρώματα συγκέντρωσης συμπιέζουν τις χωρικές διαστάσεις, ελαττώνοντας τον αριθμό των παραμέτρων που απαιτούνται για την εξαγωγή χαρακτηριστικών στα στρώματα που ακολουθούν. Στην κατηγοριοποίηση εικόνας, η αναπαράσταση της αρχικής εικόνας ελαττώνεται σταδιακά και καταλήγει σε μια βαθιά στοίβα ξεχωριστών παραστάσεων της. Σε αυτό το σημείο η πληροφορία τροφοδοτεί τα νέα πλήρως συνδεδεμένα στρώματα για να αποφασιστεί η πιθανότητα εμφάνισης κάθε κατηγορίας αποτελέσματος στην εικόνα.

Τα δίκτυα αυτού του είδους αξιοποιούν αρχές της γραμμικής άλγεβρας και ιδιαίτερα τον πολλαπλασιασμό πινάκων και αποτελούν πλούσιους εξαγωγείς χαρακτηριστικών, που βρίσκουν ιδιαίτερη εφαρμογή στην κατηγοριοποίηση και τμηματοποίηση εικόνας, καθώς και στον εντοπισμό προτύπων. Υπάρχουν διάφορες αρχιτεκτονικές CNN που υπάγονται σε δυο κύριες κατηγορίες: τις κλασσικές (LeNet-5, AlexNet, VGG 16) και τις σύγχρονες (Inception, ResNet, ResNeXt, DenseNet).

2.3.4 Αυτόματοι κωδικοποιητές (Autoencoders)

Αποτελούν μια αρχιτεκτονική νευρωνικών δικτύων ικανή να εντοπίσει τη δομή των δεδομένων με σκοπό να αναπτύξει μια συμπιεσμένη αναπαράσταση της τιμής εισόδου. Παρόλο που παρουσιάζουν ομοιότητες με τα πιο παραδοσιακά νευρωνικά δίκτυα, οι αυτόματοι κωδικοποιητές στοχεύουν στη μοντελοποίηση των δεδομένων εισόδου και για αυτό η μέθοδος θεωρείται μη επιβλεπόμενη [39].



Εικόνα 16 - Δομή αυτόματου κωδικοποιητή

Πηγή: <https://www.mygreatlearning.com/blog/autoencoder/>



Υπάρχουν πολλές παραλλαγές της γενικής αρχιτεκτονικής του αυτόματου κωδικοποιητή, όλες με σκοπό να διασφαλίσουν ότι η συμπιεσμένη αναπαράσταση αντιπροσωπεύει σημαντικά γνωρίσματα των αρχικών δεδομένων εισόδου. Ένας αυτόματος κωδικοποιητής αποτελείται από τρία μέρη: τον κωδικοποιητή (encoder), τον κώδικα (code) και τον αποκωδικοποιητή (decoder). Ο κωδικοποιητής συμπιέζει την πληροφορία της τιμής εισόδου και παράγει τον κώδικα τον οποίο και χρησιμοποιεί ο αποκωδικοποιητής για την ανακατασκευή της τιμής εισόδου. Το αποτέλεσμα του μοντέλου δεν θα είναι ακριβώς το ίδιο με την τιμή εισόδου, αλλά θα είναι μια κοντινή αναπαράσταση. Για να κατασκευαστεί το μοντέλο απαιτείται μια μέθοδος κωδικοποίησης, μια μέθοδος αποκωδικοποίησης και μια συνάρτηση κόστους.

Ένας τρόπος για να μάθει το μοντέλο χρήσιμα χαρακτηριστικά είναι να κρατήσουμε το μέγεθος του στρώματος του κώδικα όσο το δυνατόν μικρότερο. Ένας άλλος τρόπος είναι η προσθήκη τυχαίου θορύβου στα δεδομένα εισόδου. Με αυτό τον τρόπο ο autoencoder δεν θα μπορεί απλά να αντιγράψει την τιμή εισόδου στο αποτέλεσμα επειδή θα περιέχεται θόρυβος, ο οποίος θα πρέπει να αφαιρεθεί. Αυτά τα μοντέλα ονομάζονται denoising encoders. Ένας τρίτος τρόπος είναι μέσω της κανονικοποίησης. Το μοντέλο μπορεί να κανονικοποιηθεί μέσω ενός περιορισμού αραιότητας (sparsity constraint), με τρόπο τέτοιο ώστε μόνο ένα μικρό μέρος των κόμβων να έχει μη μηδενικές τιμές. Αυτό επιτυγχάνεται με την προσθήκη ενός όρου ποινής στη συνάρτηση κόστους. Η μέθοδος αυτή ονομάζεται sparse encoder και λειτουργεί καλά ακόμη και εάν το στρώμα του κώδικα είναι μεγάλο, αφού μόνο ένας μικρός αριθμός κόμβων είναι ενεργός κάθε φορά.

2.5 Αλγόριθμοι βελτιστοποίησης

Οι αλγόριθμοι βελτιστοποίησης ή αλλιώς βελτιστοποιητές (optimizers) είναι αλγόριθμοι ή μέθοδοι, που χρησιμοποιούνται για την τροποποίηση χαρακτηριστικών του νευρωνικού δικτύου, όπως είναι τα βάρη και ο ρυθμός μάθησης. Σκοπός αυτών των τροποποιήσεων δεν είναι άλλος από την ελαχιστοποίηση της συνάρτησης απώλειας για την παροχή των ακριβέστερων δυνατών αποτελεσμάτων. Στη συνέχεια αυτής της ενότητας γίνεται παρουσίαση κάποιων εκ των σημαντικότερων αλγορίθμων.

2.5.1 Σύγκλιση με ελάττωση της παραγώγου (Gradient descent)

Ο αλγόριθμος σύγκλισης με ελάττωση της παραγώγου (gradient descent) αποτελεί τον απλούστερο και συνάμα συχνότερα χρησιμοποιούμενο αλγόριθμο βελτιστοποίησης, με κύρια εφαρμογή στη γραμμική παλινδρόμηση και σε αλγόριθμους κατηγοριοποίησης. Επίσης, αρκετά συχνά χρησιμοποιείται σε αλγόριθμους οπισθοδιάδοσης (backpropagation) στα τεχνητά νευρωνικά δίκτυα. Η σύγκλιση με ελάττωση της παραγώγου είναι ένας αλγόριθμος βελτιστοποίησης πρώτης τάξης που εξαρτάται από την παράγωγο πρώτης τάξης της συνάρτησης απώλειας. Επί της ουσίας, υπολογίζει τον τρόπο με τον οποίο θα πρέπει να τροποποιηθούν τα βάρη ώστε να ελαχιστοποιηθεί η συνάρτηση. Μέσω της οπισθοδιάδοσης, η απώλεια μεταφέρεται από το ένα επίπεδο στο άλλο και οι παράμετροι του μοντέλου μεταβάλλονται αναλόγως με σκοπό την ελαχιστοποίηση της απώλειας.

2.5.2 Στοχαστική σύγκλιση με ελάττωση της παραγώγου (Stochastic Gradient Descent - SGD)

Αποτελεί μια παραλλαγή της σύγκλισης με ελάττωση της παραγώγου και αυτό που προσπαθεί να επιτύχει είναι η συχνότερη ενημέρωση των παραμέτρων του μοντέλου. Σε αυτό τον αλγόριθμο, οι παράμετροι του μοντέλου τροποποιούνται μετά τον υπολογισμό της απώλειας σε κάθε παράδειγμα εκπαίδευσης. Έτσι, λοιπόν, εάν το σύνολο δεδομένων περιέχει 1000 παρατηρήσεις, ο αλγόριθμος SGD θα ενημερώσει το μοντέλο 1000 φορές σε ένα κύκλο του συνόλου δεδομένων σε αντίθεση με την απλή σύγκλιση με ελάττωση της παραγώγου που θα το κάνει μία φορά:

$$\theta = \theta - \eta \cdot \nabla_{\theta} J(\theta; x^i; y^i)$$

Ο αλγόριθμος SGD κάνει συχνές ενημερώσεις με υψηλή διακύμανση, κάτι το οποίο έχει ως αποτέλεσμα να παρατηρούνται υψηλές διακυμάνσεις στην υποκείμενη συνάρτηση.

2.5.3 Προσαρμοστικός αλγόριθμος κλίσης (Adaptive gradient - AdaGrad)

Είναι ένας αλγόριθμος ο οποίος προσαρμόζει το ρυθμό μάθησης (learning rate) των παραμέτρων, πραγματοποιώντας μικρότερες ενημερώσεις για παραμέτρους που

αφορούν χαρακτηριστικά με υψηλή συχνότητα και μεγαλύτερες για παραμέτρους που αφορούν λιγότερα συχνά χαρακτηριστικά [40]. Αυτός είναι και ο λόγος που η μέθοδος αυτή είναι ιδιαίτερος αποτελεσματική όταν εφαρμόζεται σε αραιά δεδομένα. Στον κανόνα ενημέρωσής του, ο AdaGrad τροποποιεί τον γενικό ρυθμό μάθησης η , κάθε χρονικό βήμα t , για κάθε παράμετρο θ_i , με βάση τις προηγούμενες κλίσεις που έχουν υπολογιστεί για την παράμετρο θ_i :

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i}$$

Ένα από τα κύρια πλεονεκτήματα του αλγόριθμου AdaGrad είναι η εξάλειψη της ανάγκης για χειροκίνητη ρύθμιση του ρυθμού μάθησης. Ενώ, η κύρια αδυναμία του εντοπίζεται στο γεγονός της συσσώρευσης των τετραγωνικών κλίσεων στον παρονομαστή, αφού κάθε όρος είναι θετικός και το σύνολο διαρκώς αυξάνεται κατά την εκπαίδευση.

2.5.4 RMSprop

Ο αλγόριθμος RMSprop αποτελεί μια τεχνική βελτιστοποίησης βασισμένη στην κλίση (gradient-based). Εισήχθη από τον Αγγλο-Καναδό Geoffrey Hinton, καθηγητή του πανεπιστημίου του Τορόντο και μέλος της ομάδας τεχνητής νοημοσύνης της Google, Google Brain. Ο Hinton μεταξύ άλλων έχει σημαντική συμβολή στην εφαρμογή της οπισθοδιάδοσης σε πολυεπίπεδα νευρωνικά δίκτυα [41].

Οι κλίσεις πολύπλοκων συναρτήσεων όπως είναι τα νευρωνικά δίκτυα έχουν την τάση να εξαφανίζονται όσο τα δεδομένα διαδίδονται στη συνάρτηση (πρόβλημα διαβάθμισης που εξαφανίζεται – vanishing gradient problem). Ο αλγόριθμος RMSprop αντιμετωπίζει το παραπάνω πρόβλημα χρησιμοποιώντας εάν μέσο όρο τετραγωνικών κλίσεων για την κανονικοποίηση της κλίσης. Αυτή η κανονικοποίηση έχει ως αποτέλεσμα την τροποποίηση του βήματος της εκπαίδευσης, ελαττώνοντας το όταν υπάρχουν μεγάλες κλίσεις και αυξάνοντας το όταν η κλίση είναι μικρή. Με απλά λόγια, ο αλγόριθμος χρησιμοποιεί ένα προσαρμοζόμενο ρυθμό εκπαίδευσης και δεν τον χειρίζεται ως μια υπερπαράμετρο του μοντέλου:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g]_t^2 + \epsilon}} \cdot g_t$$

Ο RMSprop αποτελεί μια παραλλαγή του αλγορίθμου AdaGrad, ο οποίος χρησιμοποιώντας εκθετικούς κινούμενους μέσους προηγούμενων τετραγωνικών κλίσεων μετριάζει την ταχύτητα μείωσης του ρυθμού εκπαίδευσης.

2.5.5 Προσαρμοστική Εκτίμηση Στιγμής (Adaptive Moment Estimation - ADAM)

Ο ADAM είναι ένας αλγόριθμος κλίσης πρώτης τάξης που βασίζεται σε προσαρμοστικές εκτιμήσεις ροπών χαμηλότερης τάξης. Η συγκεκριμένη μέθοδος, η οποία εισήχθη από τους Kingma και Ba το 2014, αποτελεί έναν αποτελεσματικό τρόπο στοχαστικής βελτιστοποίησης που απαιτεί μόνο κλίσεις πρώτου βαθμού και λίγη μνήμη [42]. Η μέθοδος υπολογίζει τους προσαρμοστικούς ρυθμούς μάθησης για κάθε παράμετρο, χρησιμοποιώντας τις πρώτες και δεύτερες παραγώγους των κλίσεων και επί της ουσίας συνδυάζει τα πλεονεκτήματα των αλγορίθμων AdaGrad και RMSProp. Όπως και ο αλγόριθμος RMSprop, αποθηκεύει έναν εκθετικά μειούμενο μέσο όρο προηγούμενων τετραγωνικών κλίσεων u_t , αλλά επιπροσθέτως αποθηκεύει και έναν εκθετικά μειούμενο μέσο όρο προηγούμενων κλίσεων m_t . Ο κανόνας ενημέρωσης του αλγορίθμου έχει ως εξής:

$$\theta_{t+1} = \theta_t - \frac{n}{\sqrt{\hat{u}_t + \varepsilon}} \hat{m}_t,$$

όπου

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \text{ και } \hat{u}_t = \frac{u_t}{1 - \beta_2^t}$$

Οι Kingma και Ba προτείνουν ως προκαθορισμένες τιμές 0,9 για το β_1 , 0,999 για το β_2 και 10^{-8} για το ε . Ο ADAM αποτελεί μια από τις πιο σύγχρονες και συνάμα αποτελεσματικότερες προσαρμοστικές μεθόδους βελτιστοποίησης.

Κεφάλαιο 3: Υλοποίηση εφαρμογής

3.1 Γενικές πληροφορίες

Τόσο για την υλοποίηση της εφαρμογής όσο και για την εκτέλεση όλων των μοντέλων τεχνητών νευρωνικών δικτύων, χρησιμοποιήθηκε αποκλειστικά η γλώσσα προγραμματισμού Python. Ακολουθεί συνοπτική αναφορά των κυριότερων εργαλείων που χρησιμοποιήθηκαν:

- **Γλώσσα προγραμματισμού:** Python (έκδοση 3.7)
- **Περιβάλλον ανάπτυξης:** Χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment - IDE), Pycharm.
- **Κύρια πακέτα και βιβλιοθήκες:** Κύριο χαρακτηριστικό της γλώσσας προγραμματισμού Python είναι η πληθώρα βιβλιοθηκών που διαθέτει, προς διευκόλυνση του προγραμματιστή. Οι κυριότερες που χρησιμοποιήθηκαν στο πλαίσιο της παρούσας εργασίας είναι:
 - **Pandas:** Οι κύριες διαδικασίες χειρισμού και ανάλυσης των δεδομένων πραγματοποιήθηκαν με χρήση της συγκεκριμένης βιβλιοθήκης.
 - **Numpy:** Χρησιμοποιήθηκε για την κάλυψη των αναγκών αριθμητικής ανάλυσης.
 - **Matplotlib, Plotly:** Οι συγκεκριμένες βιβλιοθήκες χρησιμοποιήθηκαν για τα γραφήματα τόσο των μοντέλων όσο και της εφαρμογής.
 - **Keras, Tensorflow:** Η βιβλιοθήκη Keras είναι μια βιβλιοθήκη ανοιχτού κώδικα, η οποία χρησιμοποιήθηκε ως διεπαφή της βιβλιοθήκης Tensorflow (που αναπτύχθηκε από την ομάδα τεχνητής νοημοσύνης της Google), η οποία χρησιμοποιήθηκε για την εφαρμογή των προβλεπτικών μοντέλων LSTM και MLP.
 - **Streamlit:** Αποτελεί ένα δωρεάν και ανοιχτού κώδικα πλαίσιο λογισμικού στο οποίο στηρίχθηκε η υλοποίηση της εφαρμογής.
 - **PyTorch, NeuralProphet:** Είναι μια βιβλιοθήκη μηχανικής εκμάθησης ανοιχτού κώδικα που αναπτύχθηκε κυρίως από το εργαστήριο AI Research της εταιρίας Facebook και πάνω στην οποία βασίζεται η βιβλιοθήκη μοντελοποίησης χρονοσειρών NeuralProphet.

3.2 Αξιολόγηση μοντέλων

Στην προηγούμενη ενότητα παρουσιάστηκαν τα κυριότερα εργαλεία που χρησιμοποιήθηκαν για την τεχνική υλοποίηση. Αυτό που απομένει να αποφασιστεί είναι το προβλεπτικό μοντέλο το οποίο θα χρησιμοποιηθεί για την εξυπηρέτηση του σκοπού της εφαρμογής. Τα μοντέλα που θα εξεταστούν στη συνέχεια είναι τρία και συγκεκριμένα: 1) Long Short-term Memory - LSTM, 2) Multi-layer Perceptron – MLP και 3) NeuralProphet (AR-Net).

3.2.1 Σύνολα δεδομένων για επιλογή μοντέλου

Με σκοπό την εύρεση και επιλογή του καταλληλότερου προβλεπτικού μοντέλου, χρησιμοποιήθηκαν τρία σύνολα δεδομένων που περιέχουν πληροφορίες σχετικά με την τιμή κλεισίματος των μετοχών των εταιριών Nvidia (NVDA), Advanced Micro Devices (AMD) και TESLA (TSLA) για τη χρονική περίοδο από 01/01/2011 έως και 31/12/2020. Καθένα από τα τρία σύνολα δεδομένων αποτελείται από συνολικά 2515 παρατηρήσεις.

3.2.2 Μέθοδος αξιολόγησης ακρίβειας

Για την αξιολόγηση της ακρίβειας όλων των προβλεπτικών μοντέλων χρησιμοποιήθηκε ως μέτρο η ρίζα του Μέσου Τετραγωνικού Σφάλματος (Root Mean Square Error – RMSE) των υπολοίπων. Τα υπόλοιπα είναι ουσιαστικά τα σφάλματα της πρόβλεψης και αποτελούν ένα μέτρο απόστασης των σημείων της πρόβλεψης από τη γραμμή παλινδρόμησης. Το RMSE είναι η τυπική απόκλιση των υπολοίπων, δηλαδή μετρά το πόσο απλωμένα είναι τα υπόλοιπα στο χώρο των δεδομένων. Με άλλα λόγια, το RMSE μας δείχνει το μέγεθος της συγκέντρωσης των υπολοίπων γύρω από τη γραμμή βέλτιστης προσαρμογής

3.2.3 Περιγραφική ανάλυση δεδομένων

Για τη διαγραμματική απεικόνιση των δεδομένων χρησιμοποιήθηκε ένα διάγραμμα περιοχής (area chart), το οποίο επί της ουσίας αποτελεί ένα συνδυασμό του διαγράμματος γραμμής και του ραβδογράμματος. Αυτού του είδους τα γραφήματα χρησιμοποιούνται ιδιαίτερος συχνά για τη διαγραμματική απεικόνιση χρηματοοικονομικών δεδομένων. Επιπλέον, χρησιμοποιήθηκαν γραφήματα για την ανάλυση των κύριων χαρακτηριστικών της κάθε χρονοσειράς, όπως είναι η τάση, η

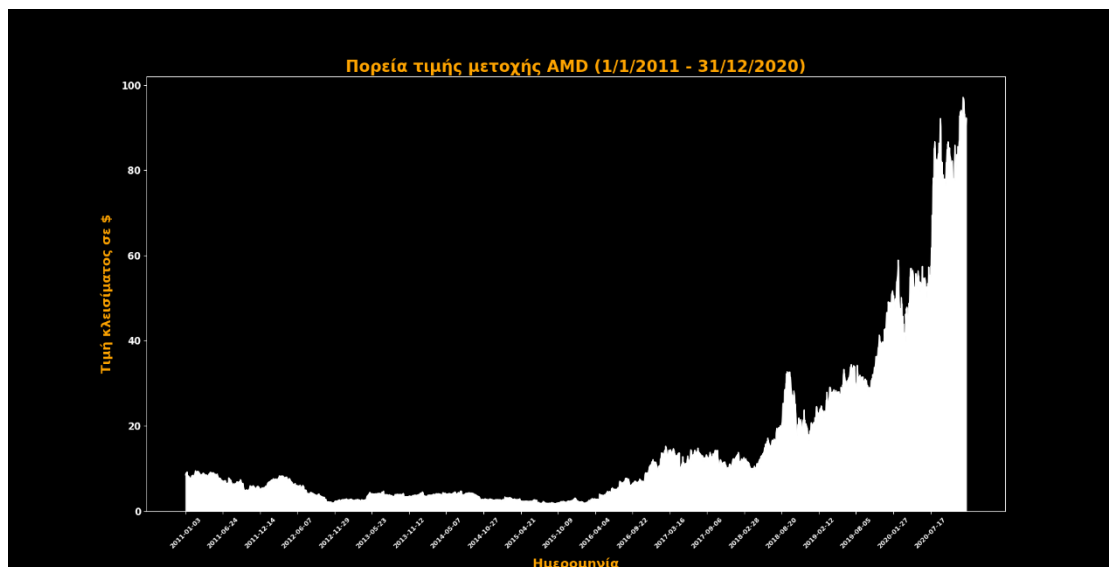
εποχικότητα και οι ακανόνιστες διακυμάνσεις (υπόλοιπα). Τέλος, έγινε και έλεγχος στασιμότητας μέσω της εφαρμογής του επαυξημένου ελέγχου Dickey-Fuller (Augmented Dickey-Fuller - ADF).

Οι δυο υποθέσεις του ελέγχου είναι οι εξής [43]:

- Μηδενική υπόθεση (H_0): Η χρονοσειρά μας έχει μια μονάδα ρίζας, που σημαίνει ότι δεν είναι στάσιμη.
- Εναλλακτική υπόθεση (H_1): Η χρονοσειρά μας δεν έχει κάποια μονάδα ρίζας. Άρα, είναι στάσιμη και δεν έχει δομή που να εξαρτάται από το χρόνο.

3.2.3.1 Χρονοσειρά μετοχής AMD

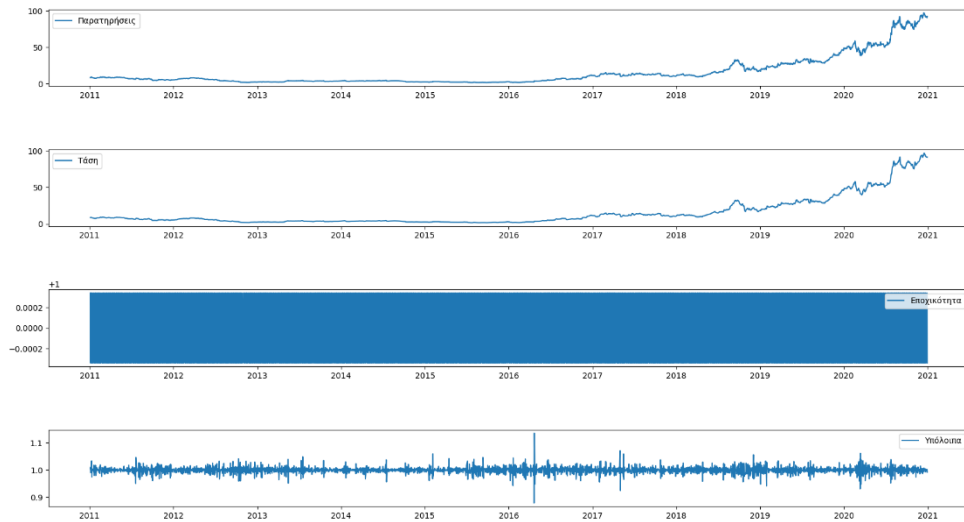
Όπως φαίνεται ξεκάθαρα από το διάγραμμα της **εικόνας 17**, η μετοχή της εταιρείας Advanced Micro Devices (AMD) παρουσιάζει μια διαρκώς αυξανόμενη τάση για το διάστημα το οποίο μελετάμε. Ενώ με τη βοήθεια του γραφήματος της **εικόνας 18** παρατηρούμε ότι δεν υπάρχει εποχικότητα. Ενδιαφέρον παρουσιάζει το διάγραμμα των υπολοίπων, όπου παρατηρούμε κάποιες περιόδους με πολύ υψηλές διακυμάνσεις όπως είναι το πρώτο τρίμηνο του 2016 (ραγδαία αύξηση λόγω της αρχιτεκτονικής Zen) και οι αρχές του 2020 (απότομη πτώση λόγω της πανδημίας του κορονοϊού).



Εικόνα 17 - Διάγραμμα περιοχής (area chart) για την τιμή κλεισίματος της μετοχής της εταιρείας Advanced Micro Devices (AMD) για το χρονικό διάστημα 01/01/2011 – 31/12/2020

Με βάση τα αποτελέσματα του ADF προκύπτει ότι σε επίπεδο σημαντικότητας $\alpha = 0,05$, το p-value ισούται με 0,988175. Οπότε δεν μπορούμε να απορρίψουμε τη

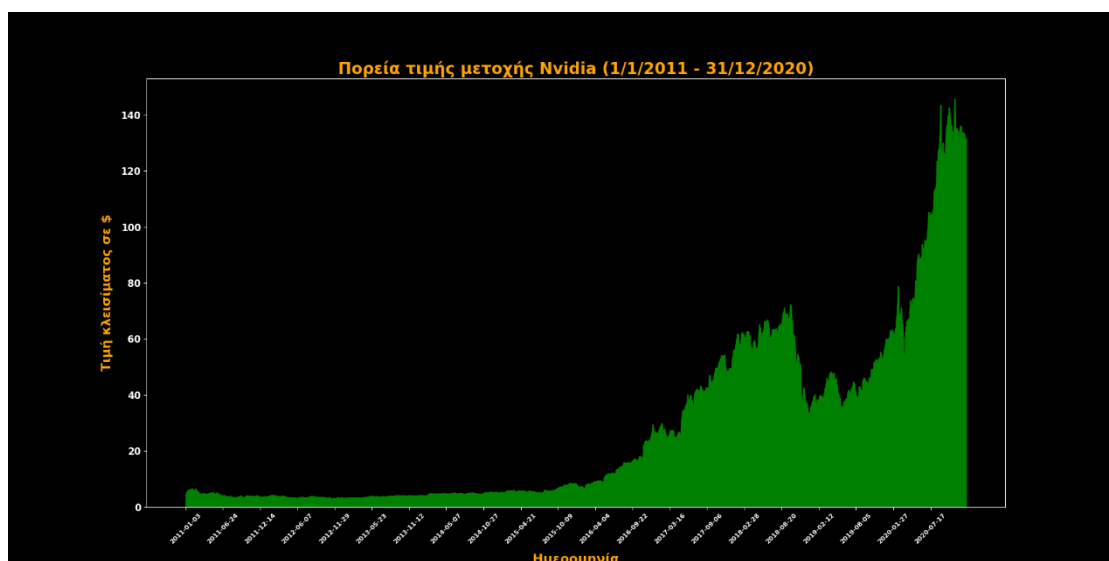
μηδενική υπόθεση ότι τα δεδομένα μας δεν είναι στάσιμα. Η άποψη αυτή ενισχύεται και από το γεγονός ότι η στατιστική τιμή της επαυξημένης δοκιμής Dickey – Fuller ισούται με 0,623260 και είναι μεγαλύτερη από την κρίσιμη τιμή του 95% διαστήματος εμπιστοσύνης, η οποία ισούται με -2,862690.



Εικόνα 18 - Ανάλυση χρονοσειράς AMD

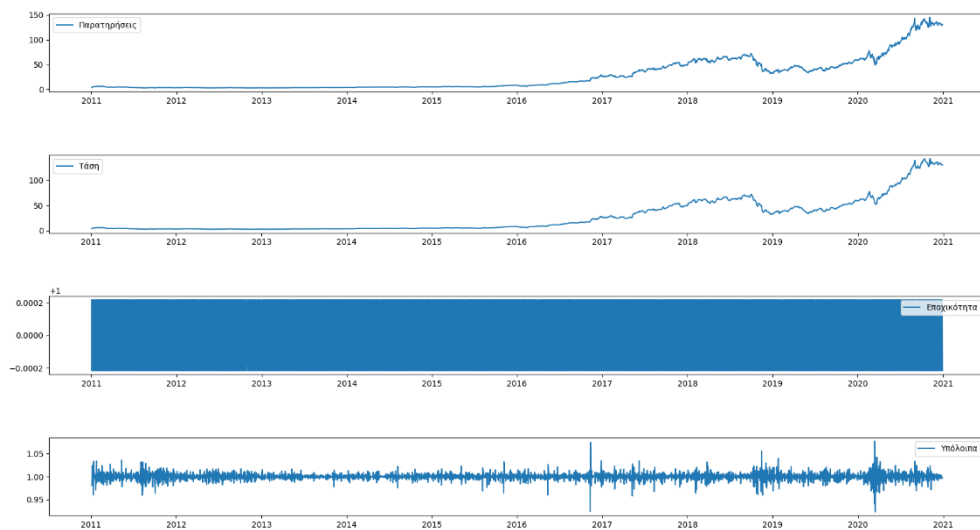
3.2.3.2 Χρονοσειρά μετοχής NVDA

Μια γενικότερη ανοδική τάση παρουσιάζουν και τα δεδομένα που αναπαριστούν την τιμή μετοχής της εταιρείας Nvidia (NVDA), όπως φαίνεται στο ραβδόγραμμα της εικόνας 19.



Εικόνα 19 - Διάγραμμα περιοχής (area chart) για την τιμή κλεισίματος της μετοχής της εταιρείας Nvidia (NVDA) για το χρονικό διάστημα 01/01/2011 – 31/12/2020

Όπως συμβαίνει και με τη μετοχή της AMD, δεν παρατηρείται εποχικότητα ενώ υπάρχουν υψηλές διακυμάνσεις κατά διαστήματα (**εικόνα 20**), όπως τα τέλη του 2018 (μεγάλη πτώση ως απόρροια του εμπορικού πολέμου μεταξύ ΗΠΑ και Κίνας) και οι αρχές του 2020 (ελέω κορονοϊού). Η εφαρμογή του επαυξημένου ελέγχου Dickey – Fuller επιβεβαίωσε τη μη-στασιμότητα και της χρονοσειράς της μετοχής NVDA, με το p-value να ισούται με 0,995706, τη στατιστική τιμή ίση με 1,161115 και την κρίσιμη τιμή του 95% ΔΕ ίση με -2,862695.

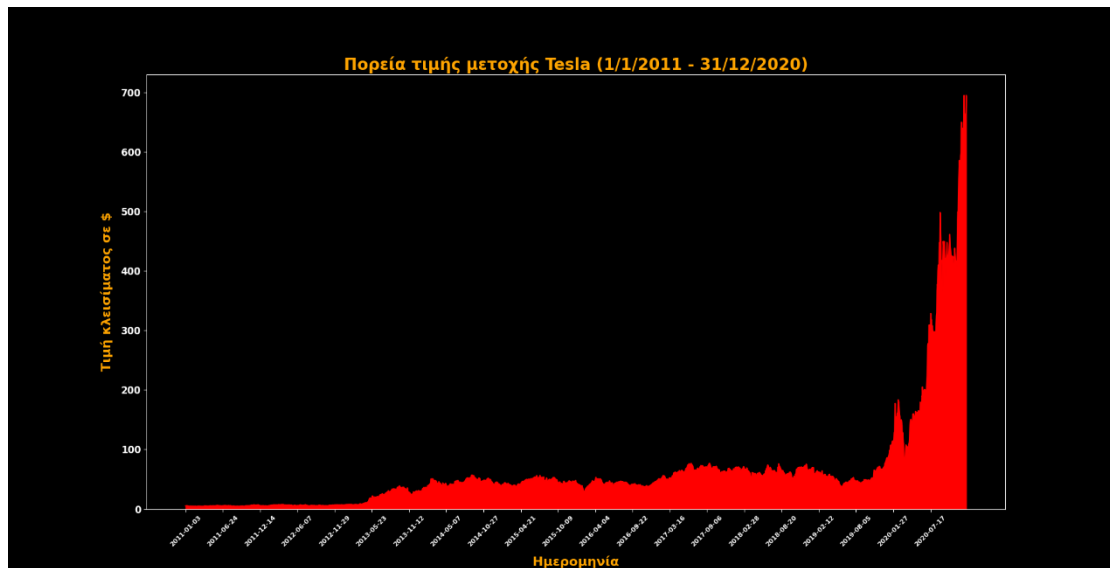


Εικόνα 20 – Ανάλυση χρονοσειράς NVDA

3.2.3.3 Χρονοσειρά μετοχής TSLA

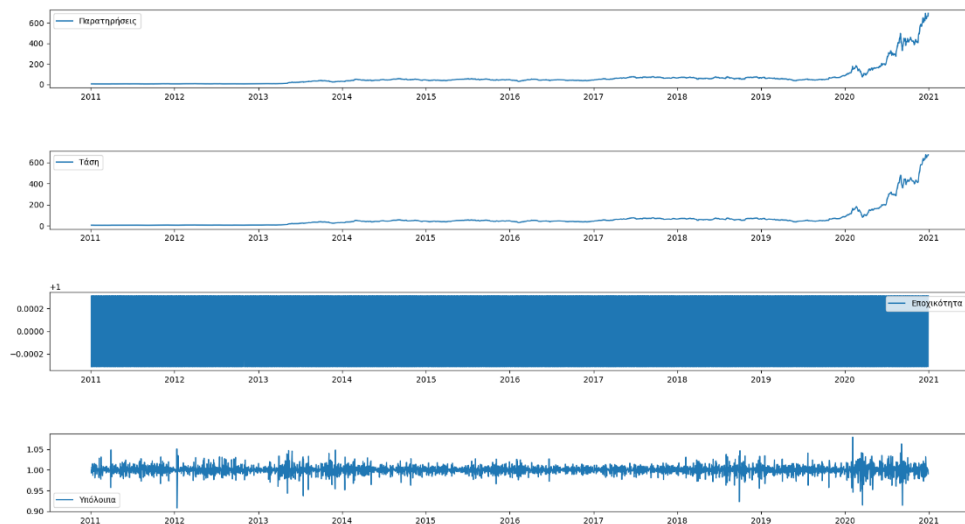
Η μετοχή της εταιρείας Tesla (TSLA) σε αντίθεση με τα προηγούμενα δυο σύνολα δεδομένων δεν είχε να επιδείξει ιδιαίτερη ανοδική τάση μέχρι πρότινος. Αυτό, ωστόσο, άλλαξε από τα τέλη του 2019 όταν και παρουσίασε ραγδαία αύξηση, η οποία απεικονίζεται ξεκάθαρα στο ραβδόγραμμα της **εικόνας 21**.

Όσον αφορά την εποχικότητα της χρονοσειράς, το μοτίβο παραμένει το ίδιο με τα άλλα δυο σύνολα δεδομένων και έτσι δεν παρατηρούνται εποχικές διακυμάνσεις. Η διαφορά που παρατηρείται συγκριτικά με τις άλλες δυο χρονοσειρές είναι στη συχνότητα εμφάνισης υψηλών διακυμάνσεων. Όπως φαίνεται στο διάγραμμα υπολοίπων της **εικόνας 22**, η μετοχή της Tesla παρουσιάζει πιο συχνά υψηλές διακυμάνσεις στην τιμή της συγκριτικά με τις άλλες δυο μετοχές που εξετάζουμε.



Εικόνα 21 - Διάγραμμα περιοχής (area chart) για την τιμή κλεισίματος της μετοχής της εταιρείας Tesla (TSLA) για το χρονικό διάστημα 01/01/2011 – 31/12/2020

Όπως συνέβη και με τα άλλα δυο σύνολα δεδομένων, το κριτήριο ADF επιβεβαίωσε τη μη-στασιμότητα της χρονοσειράς της Tesla. Το p-value είναι ίσο με 0,988175, η στατιστική τιμή ίση με 7,726334 και η κρίσιμη τιμή του 95% ΔΕ ίση με -2,862702.



Εικόνα 22 - Ανάλυση χρονοσειράς TSLA

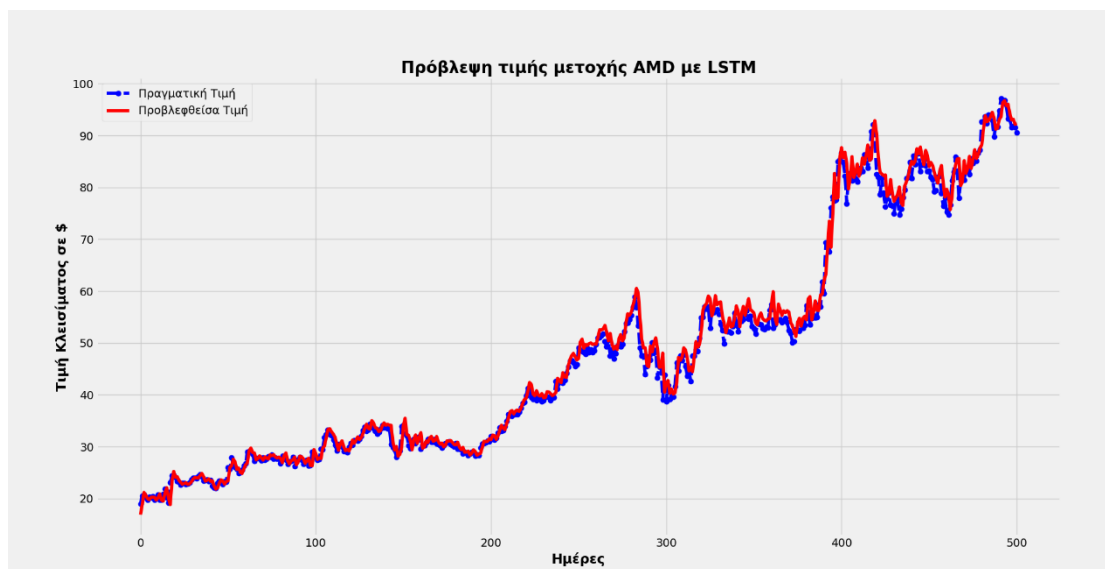
3.2.4 Πρόβλεψη με LSTM

Η ακρίβεια του μοντέλου το οποίο δημιουργήθηκε εξετάστηκε σε διάφορες παραλλαγές, με τροποποίηση τόσο των κύκλων εκπαίδευσης (epochs) όσο και των αλγορίθμων βελτιστοποίησης. Πιο συγκεκριμένα, εξετάστηκαν τέσσερις διαφορετικές

τιμές για τον αριθμό των κύκλων εκπαίδευσης: 50, 100, 200 και 400. Επιπλέον, χρησιμοποιήθηκαν τρεις διαφορετικοί αλγόριθμοι βελτιστοποίησης: Οι ADAM, RMSprop και SGD. Ο αριθμός των παραδειγμάτων εκπαίδευσης της κάθε παρτίδας (batch size) ορίστηκε στο 64 και ως συνάρτηση ενεργοποίησης χρησιμοποιήθηκε η 'ReLU. Τέλος, το σύνολο δεδομένων χωρίστηκε σε σύνολο εκπαίδευσης (ή εκμάθησης – training set) και σύνολο ελέγχου (ή επικύρωσης – test set) σε αναλογία 80/20 και ως συνάρτηση απώλειας ορίστηκε η 'MSE'.

3.2.4.1 Αποτελέσματα LSTM - AMD

Το μοντέλο που δημιουργήθηκε παρουσίασε ιδιαίτερα ικανοποιητική προβλεπτική ικανότητα, όπως φαίνεται και στο γράφημα της **εικόνας 23**. Συγκεκριμένα, το ακριβέστερο μοντέλο ήταν αυτό που χρησιμοποίησε τον αλγόριθμο βελτιστοποίησης, ADAM, σε 200 κύκλους εκπαίδευσης.



Εικόνα 23 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου LSTM για τη μετοχή AMD, με χρήση βελτιστοποιητή ADAM και 200 κύκλους εκπαίδευσης (epochs)

Τα αποτελέσματα για κάθε παραμετροποίηση που δοκιμάστηκε κατά την εκτέλεση του μοντέλου παρουσιάζονται αναλυτικά στον **πίνακα 1**. Αρχικά, αυτό που παρατηρείται είναι πως η χρήση του βελτιστοποιητή RMSprop επηρέασε αρνητικά την προβλεπτική ικανότητα του μοντέλου συγκριτικά με τους άλλους δυο αλγόριθμους, οι οποίοι παρουσίασαν πολύ υψηλότερη ακρίβεια, με τον ADAM να είναι ο πιο αποτελεσματικός. Επιπλέον, αυτό που παρατηρήθηκε είναι ότι η αύξηση των epochs

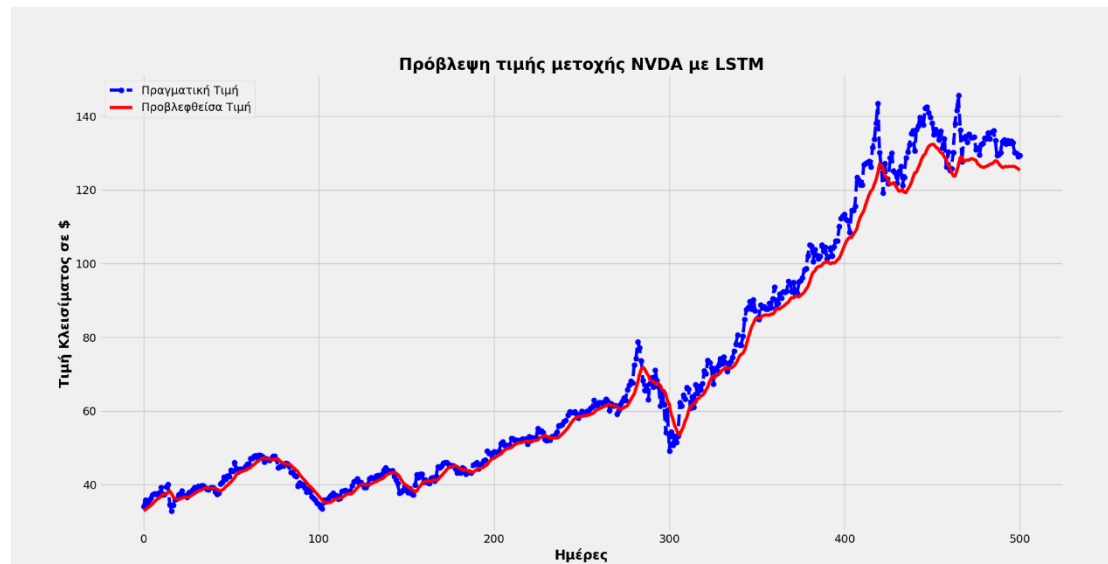
μέχρι και τους 200 συνέβαλε στην αύξηση της ακρίβειας με τους αλγόριθμους ADAM και RMSprop ενώ για τον βελτιστοποιητή SGD ισχύει το αντίστροφο.

Πίνακας 1: Αποτελέσματα μοντέλου LSTM για τη μετοχή AMD

epochs	optimizer	RMSE
50	ADAM	4,6397
100	ADAM	2,7606
200	ADAM	1,9857
400	ADAM	2,8335
50	RMSprop	13,4106
100	RMSprop	6,1487
200	RMSprop	5,2293
400	RMSprop	6,6049
50	SGD	5,4104
100	SGD	5,8406
200	SGD	6,1738
400	SGD	4,6401

3.2.4.2 Αποτελέσματα LSTM - NVDA

Η εφαρμογή του μοντέλου LSTM για την μετοχή της Nvidia, μπορεί να μην ήταν τόσο αποτελεσματική όσο αυτή της AMD, ωστόσο τα αποτελέσματα και σε αυτή την περίπτωση είναι ικανοποιητικά, κάτι το οποίο απεικονίζει και το διάγραμμα της εικόνας 24.



Εικόνα 24 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου LSTM για τη μετοχή NVDA, με χρήση βελτιστοποιητή SGD σε 100 κύκλους εκπαίδευσης (epochs)

Αυτή τη φορά, παρατηρούμε πως ο αποτελεσματικότερος βελτιστοποιητής για το μοντέλο μας είναι ο SGD και το ακριβέστερο μοντέλο επιτεύχθηκε μετά από 100

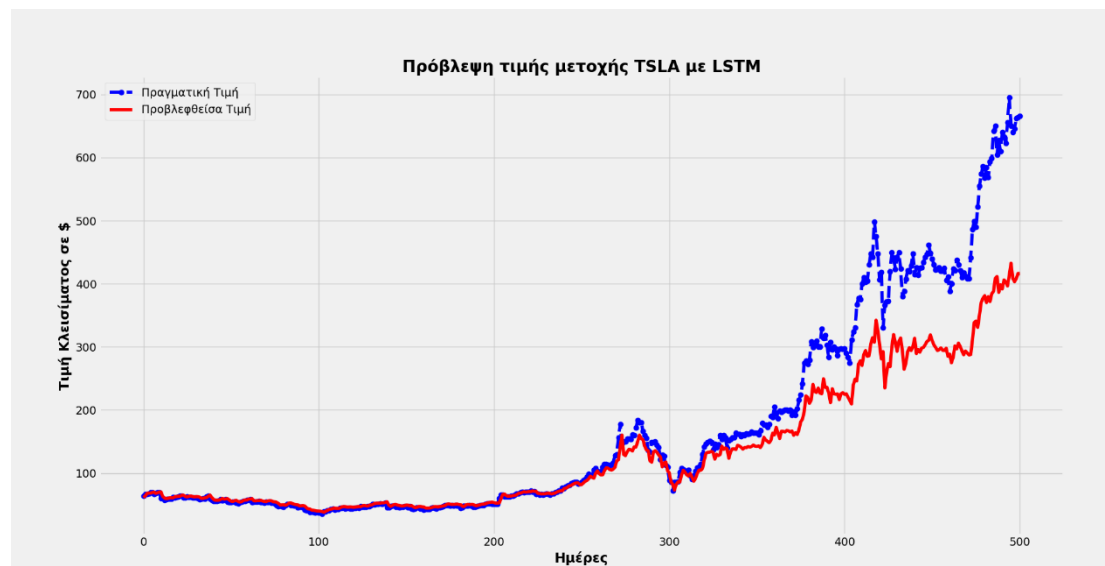
κύκλους εκπαίδευσης, ενώ ενδιαφέρον παρουσιάζει το γεγονός πως η αύξηση των κύκλων εκπαίδευσης με χρήση του αλγορίθμου RMSprop παρουσίασε αρνητική συσχέτιση με την ακρίβεια πρόβλεψης.

Πίνακας 6: Αποτελέσματα μοντέλου LSTM για τη μετοχή NVDA

epochs	optimizer	RMSE
50	ADAM	9,7205
100	ADAM	8,7236
200	ADAM	10,4406
400	ADAM	6,8119
50	RMSprop	13,5300
100	RMSprop	9,1645
200	RMSprop	8,0839
400	RMSprop	7,5062
50	SGD	5,4284
100	SGD	4,5691
200	SGD	5,6677
400	SGD	5,5352

3.2.4.3 Αποτελέσματα LSTM - TSLA

Σε αντίθεση με τις χρονοσειρές των μετοχών AMD και NVDA, το μοντέλο που κατασκευάστηκε με χρήση LSTM, δεν παρουσίασε την ίδια υψηλή προβλεπτική ικανότητα, κάτι το οποίο αποτυπώνεται ξεκάθαρα στο γράφημα της εικόνας 25.



Εικόνα 25 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου LSTM για τη μετοχή TSLA, με χρήση βελτιστοποιητή ADAM σε 50 κύκλους εκπαίδευσης (epochs)

Το ακριβέστερο μοντέλο ήταν αυτό που χρησιμοποίησε τον αλγόριθμο βελτιστοποίησης, ADAM, σε 50 κύκλους εκπαίδευσης. Αυτό που παρατηρήθηκε και χρήζει αναφοράς, είναι το γεγονός πως όσο αυξάνονταν οι κύκλοι εκπαίδευσης του μοντέλου τόσο μειωνόταν η προβλεπτική του ικανότητα.

Πίνακας 7: Αποτελέσματα μοντέλου LSTM για τη μετοχή TSLA

epochs	optimizer	RMSE
50	ADAM	70,6878
100	ADAM	78,5167
200	ADAM	86,1894
400	ADAM	113,1085
50	RMSprop	95,2633
100	RMSprop	98,4086
200	RMSprop	98,5439
400	RMSprop	111,1869
50	SGD	86,6767
100	SGD	89,1287
200	SGD	97,1874
400	SGD	84,3668

3.2.5 Πρόβλεψη με MLP

Οι παραμετροποίηση του μοντέλου MLP δεν διαφέρει από αυτή του LSTM. Ενώ και πάλι εξετάστηκαν τέσσερις διαφορετικές τιμές για τον αριθμό των κύκλων εκπαίδευσης: 50, 100, 200 και 400. Επιπλέον, χρησιμοποιήθηκαν οι ίδιοι τρεις αλγόριθμοι βελτιστοποίησης: ADAM, RMSprop και SGD. Το batch size παρέμεινε στο 64, ενώ και η αναλογία του συνόλου εκπαίδευσης προς το σύνολο ελέγχου παρέμεινε στο 80/20.

3.2.5.1 Αποτελέσματα MLP – AMD

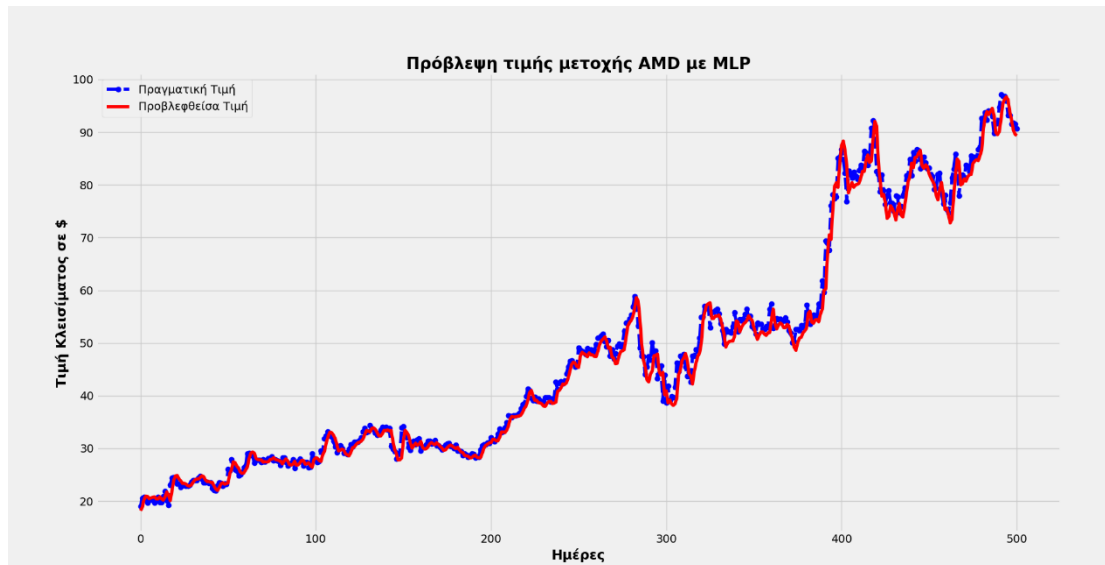
Το μοντέλο MLP που εφαρμόστηκε για τη μετοχή AMD παρουσίασε παρόμοια αποτελέσματα με το LSTM, με τις προβλέψεις του να είναι ελαφρώς πιο ανακριβείς.

Πίνακας 5: Αποτελέσματα μοντέλου MLP για τη μετοχή AMD

epochs	optimizer	RMSE
50	ADAM	2,5216
100	ADAM	2,3284
200	ADAM	2,0429
400	ADAM	2,1773
50	RMSprop	6,4395
100	RMSprop	7,3452
200	RMSprop	9,8309

400	RMSprop	10,3043
50	SGD	3,6677
100	SGD	3,1526
200	SGD	2,5553
400	SGD	2,4140

Όπως και στο LSTM, το ακριβέστερο μοντέλο ήταν αυτό που χρησιμοποίησε τον αλγόριθμο βελτιστοποίησης ADAM σε 200 κύκλους εκπαίδευσης.



Εικόνα 26 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου MLP για τη μετοχή AMD, με χρήση βελτιστοποιητή ADAM σε 200 κύκλους εκπαίδευσης (epochs)

3.2.5.2 Αποτελέσματα MLP - NVDA

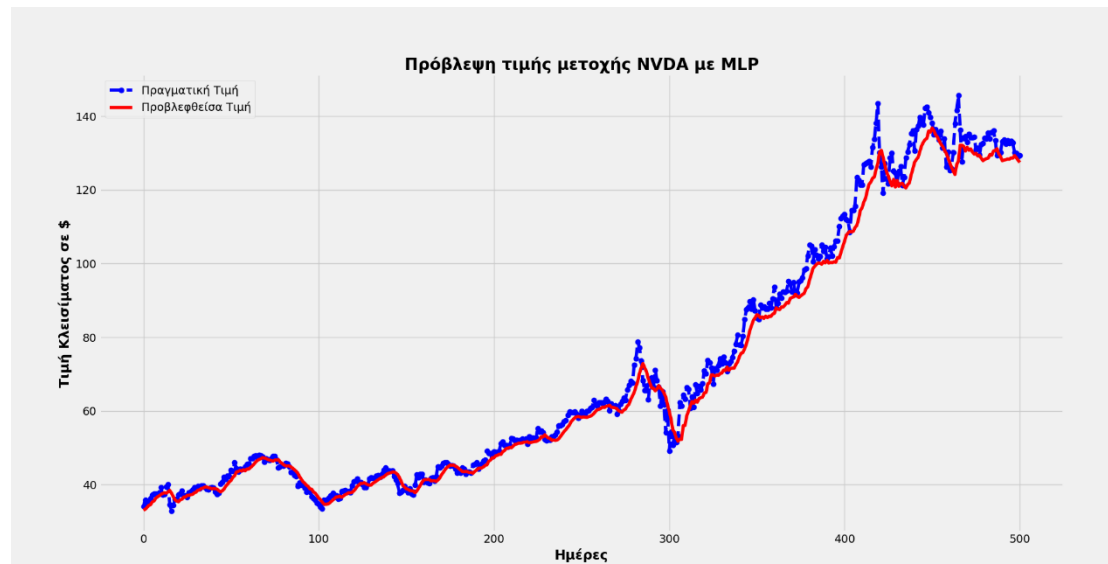
Όσον αφορά τη χρονοσειρά της Nvidia, το MLP παρουσίασε ελαφρώς καλύτερα αποτελέσματα συγκριτικά με το LSTM με το RMSE να είναι 4,5691 και 3,4834, αντιστοίχως.

Πίνακας 6: Αποτελέσματα μοντέλου MLP για τη μετοχή NVDA

epochs	optimizer	RMSE
50	ADAM	5,7955
100	ADAM	7,4459
200	ADAM	8,8673
400	ADAM	6,5192
50	RMSprop	5,6812
100	RMSprop	6,8601
200	RMSprop	5,8290
400	RMSprop	5,2178
50	SGD	4,2571
100	SGD	4,2441

200	SGD	3,9505
400	SGD	3,4834

Όπως και στο LSTM, έτσι και σε αυτό το μοντέλο ο βελτιστοποιητής SGD είχε τη θετικότερη επίδραση, ωστόσο απαιτήθηκαν περισσότεροι κύκλοι εκπαίδευσης για την επίτευξη της ακριβέστερης πρόβλεψης.



Εικόνα 27 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου MLP για τη μετοχή NVDA, με χρήση βελτιστοποιητή SGD σε 400 κύκλους εκπαίδευσης (epochs)

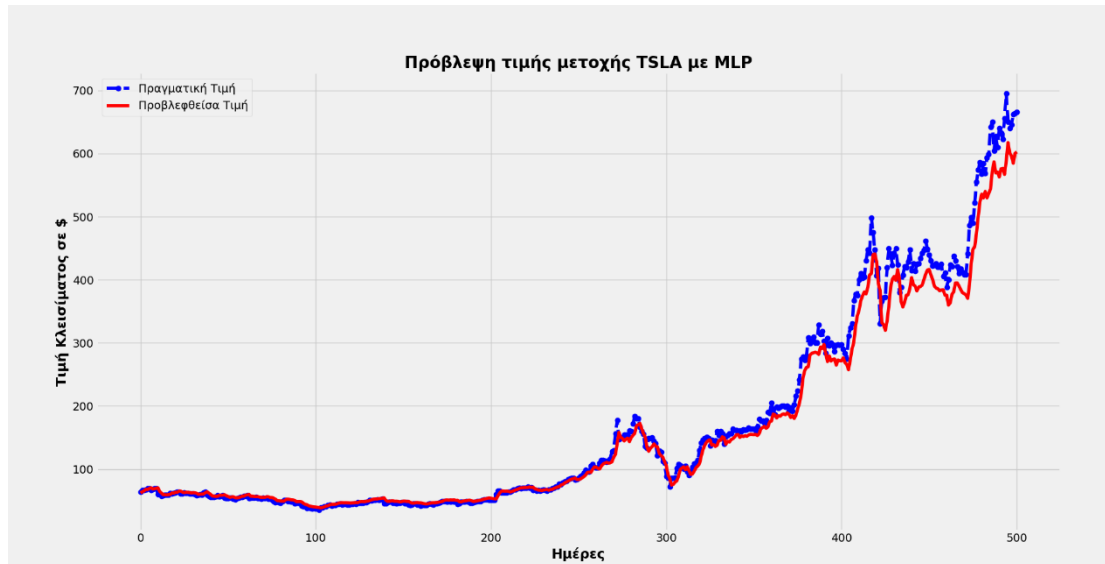
3.2.5.3 Αποτελέσματα MLP - TSLA

Στα προηγούμενα σύνολα δεδομένων, τα δυο μοντέλα παρήγαγαν παρόμοια αποτελέσματα. Όσον αφορά, ωστόσο, τη μετοχή της Tesla υπήρξε σημαντική απόκλιση μεταξύ της ακρίβειας των δυο μοντέλων.

Πίνακας 7: Αποτελέσματα μοντέλου MLP για τη μετοχή TSLA

epochs	optimizer	RMSE
50	ADAM	24,2598
100	ADAM	25,2384
200	ADAM	23,6185
400	ADAM	23,0925
50	RMSprop	39,0652
100	RMSprop	34,9822
200	RMSprop	42,2170
400	RMSprop	35,6872
50	SGD	22,1642
100	SGD	17,6757
200	SGD	16,3642
400	SGD	15,4375

Σε αντίθεση με το μοντέλο LSTM το οποίο δυσκολεύτηκε πολύ (χαμηλότερο RMSE = 70,6878), το MLP έκανε σαφώς καλύτερη δουλειά με το RMSE να είναι σχεδόν πέντε φορές μικρότερο.



Εικόνα 28 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου MLP για τη μετοχή TSLA, με χρήση βελτιστοποιητή SGD σε 400 κύκλους εκπαίδευσης (epochs)

3.2.6 Πρόβλεψη με NeuralProphet

Το τρίτο μοντέλο το οποίο θα μελετήσουμε είναι το NeuralProphet. Η συγκεκριμένη βιβλιοθήκη της Python χρησιμοποιείται για την ανάλυση χρονοσειρών με χρήση νευρωνικών δικτύων και βασίζεται στη βιβλιοθήκη μηχανικής εκμάθησης, PyTorch. Όπως γίνεται αντιληπτό από το όνομά της, είναι εμπνευσμένη από το λογισμικό ανοιχτού κώδικα, Prophet, το οποίο αναπτύχθηκε από την ομάδα μηχανικής μάθησης της εταιρείας Facebook [44].

Το νευρωνικό δίκτυο που χρησιμοποιείται από το μοντέλο είναι το AR-Net, ένα αυτοπαλίνδρομο και εμπρός τροφοδοτούμενο νευρωνικό δίκτυο (Auto-Regressive Feed-Forward Neural Network) για χρονοσειρές. Το AR-Net είναι το ίδιο ερμηνεύσιμο με ένα κλασικό αυτοπαλίνδρομο μοντέλο αλλά έχει τη δυνατότητα κλιμάκωσης σε μακροπρόθεσμες εξαρτήσεις [45].

Η συγκεκριμένη βιβλιοθήκη διαθέτει μια πλειάδα υπερπαραμέτρων (hyperparameters), οι οποίες δίνουν τη δυνατότητα στον αναλυτή να προβεί σε πολλούς πειραματισμούς για την εύρεση του καταλληλότερου μοντέλου. Σε αυτό το σημείο θα αναφέρουμε συνοπτικά μερικές από τις σημαντικότερες παραμέτρους και τη λειτουργία τους:

- *n_lags*: Αυτή η παράμετρος αποφασίζει για την ενεργοποίηση ή μη του AR-Net και η τιμή της είναι αυτή που αποφασίζει για το πόσο πίσω στο παρελθόν θα πρέπει να λαμβάνονται υπόψιν οι αυτοπαλίνδρομες εξαρτήσεις. Για να ενεργοποιηθεί το νευρωνικό θα πρέπει η τιμή αυτή να είναι μεγαλύτερη του μηδενός, ενώ προτείνεται να είναι και μεγαλύτερη από το *n_forecasts* (αριθμός προβλέψεων) το οποίο και έχουμε αφήσει στην default τιμή ίση με 1. Στο πείραμα που εκτελέστηκε δοκιμάστηκαν 4 διαφορετικές τιμές: 2, 5, 10 και 20.
- *n_changepoints*: Η συγκεκριμένη παράμετρος ορίζει τον αριθμό των σημείων στον οποίο μπορεί να γίνει αλλαγή της τάσης. Όπως παρατηρήθηκε, η τιμή της παραμέτρου παίζει πολύ σημαντικό ρόλο στην προβλεπτική ικανότητα του μοντέλου. Στο πείραμα που εκτελέστηκε δοκιμάστηκαν τέσσερις διαφορετικές τιμές: 10, 20 και 40.
- Όσον αφορά τις υπόλοιπες παραμέτρους, ο αριθμός των παραδειγμάτων εκπαίδευσης της κάθε παρτίδας (batch size) ορίστηκε στο 64, οι κύκλοι εκπαίδευσης (epochs) παρατηρήθηκε πως δεν έχουν ιδιαίτερη επίδραση στην προβλεπτική ικανότητα του μοντέλου και αποφασίστηκε να επιλεγούν αυτόματα από τη βιβλιοθήκη ενώ ορίστηκε ένα κρυφό επίπεδο (hidden layer). Τέλος, ως συνάρτηση απώλειας ορίστηκε η 'huber'.

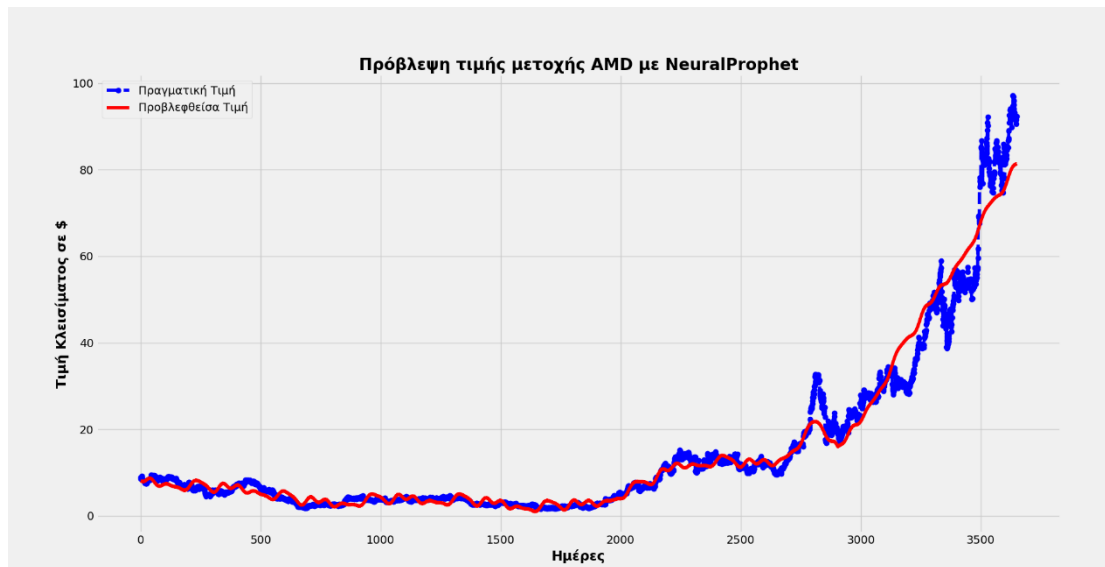
3.2.6.1 Αποτελέσματα NeuralProphet – AMD

Η εφαρμογή του μοντέλου NeuralProphet για τη μετοχή της AMD παρουσίασε υψηλότερο σφάλμα πρόβλεψης συγκριτικά με τα άλλα δυο μοντέλα, με την καλύτερη πρόβλεψη να φτάνει στο 3,6024.

Πίνακας 8: Αποτελέσματα NeuralProphet για τη μετοχή AMD

lags	changepoints	RMSE
2	10	4,8233
5	10	4,8282
10	10	4,7862
20	10	4,7893
2	20	4,1655
5	20	4,1630
10	20	4,1606
20	20	4,1724
2	40	3,6314
5	40	3,6024
10	40	3,6348
20	40	3,6381

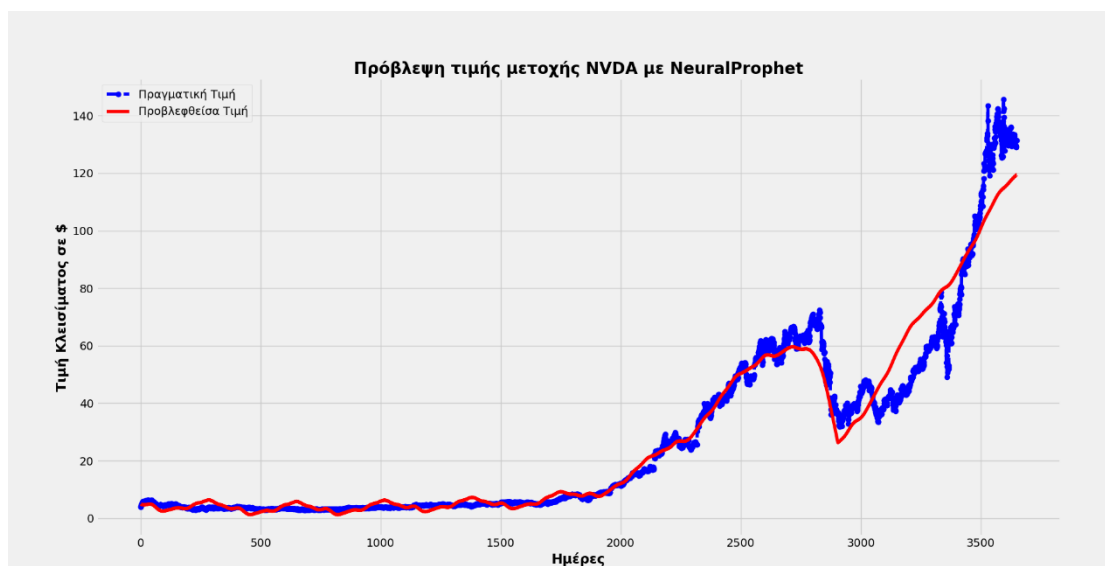
Αυτό που παρατηρήθηκε είναι πως η ακρίβεια της προβλεπτικής ικανότητας του μοντέλου είναι ανάλογη με την αύξηση του αριθμού των changepoints.



Εικόνα 29 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου NeuralProphet για τη μετοχή AMD, με 5 lags και 40 changepoints

3.2.6.2 Αποτελέσματα NeuralProphet - NVDA

Το μοντέλο παρουσίασε και πάλι μεγαλύτερο προβλεπτικό σφάλμα σε σχέση με τα άλλα δυο μοντέλα, ωστόσο παρατηρούμε πως προέβλεψε με απόλυτη σχεδόν ακρίβεια τη ραγδαία πτώση του 2018.



Εικόνα 30 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου NeuralProphet για τη μετοχή NVDA, με 20 lags και 40 changepoints

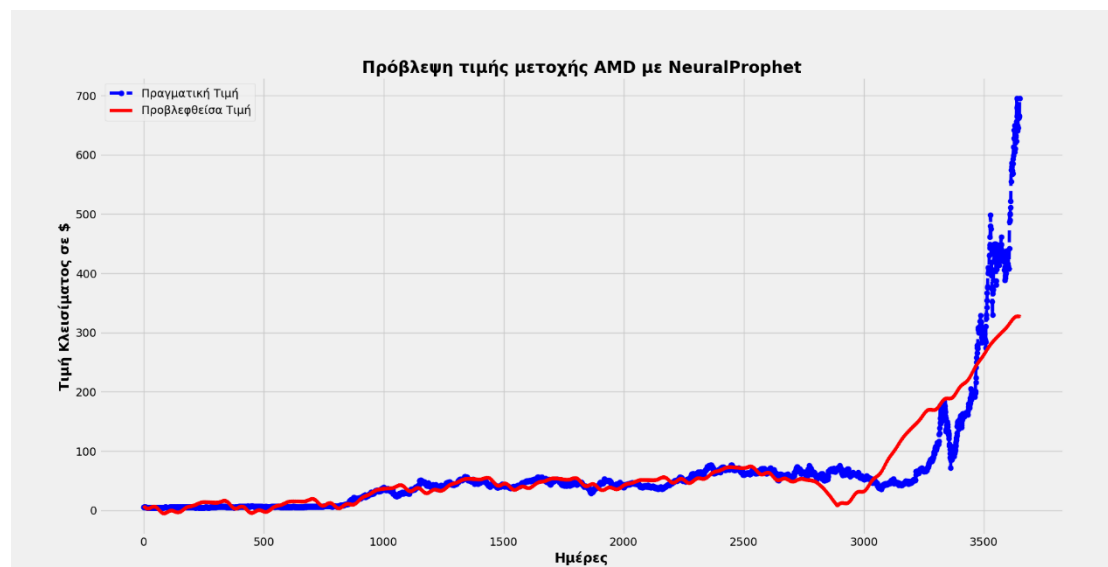
Όπως και με τη μετοχή της AMD, η αύξηση του αριθμού των changepoints συνέβαλε στη βελτίωση της ακρίβειας του μοντέλου.

Πίνακας 9: Αποτελέσματα NeuralProphet για τη μετοχή NVDA

lags	changepoints	RMSE
2	10	12,2775
5	10	12,2778
10	10	12,2830
20	10	12,3076
2	20	10,4568
5	20	10,4876
10	20	10,4730
20	20	10,4853
2	40	6,8758
5	40	6,8752
10	40	6,8571
20	40	6,8364

3.2.6.3 Αποτελέσματα NeuralProphet - TSLA

Το NeuralProphet παρουσίασε καλύτερα αποτελέσματα από το LSTM, όχι όμως και από το MLP, με το RMSE να είναι 45,6817.



Εικόνα 31 - Διαγραμματική απεικόνιση προβλεπτικού μοντέλου NeuralProphet για τη μετοχή TSLA, με 5 lags και 40 changepoints

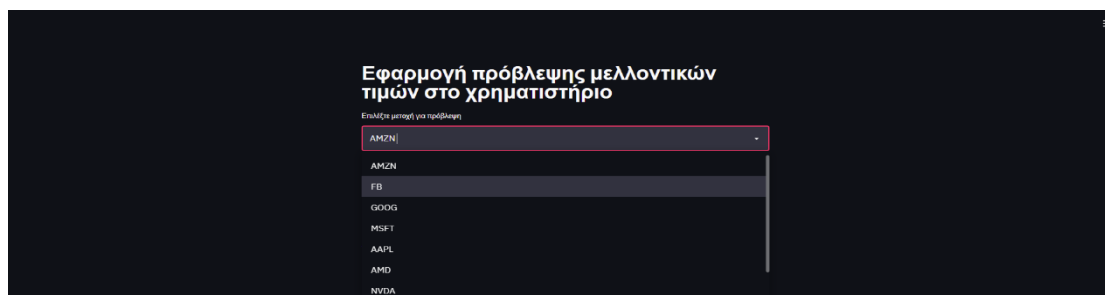
Σημαντικό είναι το γεγονός πως το μοντέλο προέβλεψε την αλλαγή της τάσης και την επικείμενη άνοδο της μετοχής από το 2019 και έπειτα, χωρίς να καταφέρει, ωστόσο, να προβλέψει απόλυτα το πλήρες μέγεθος αυτής.

Πίνακας 9: Αποτελέσματα NeuralProphet για τη μετοχή NVDA

lags	changepoints	RMSE
2	10	58,2183
5	10	58,2245
10	10	58,2253
20	10	58,2197
2	20	48,0465
5	20	48,0453
10	20	47,9974
20	20	48,0413
2	40	45,6903
5	40	45,6817
10	40	45,6949
20	40	45,7024

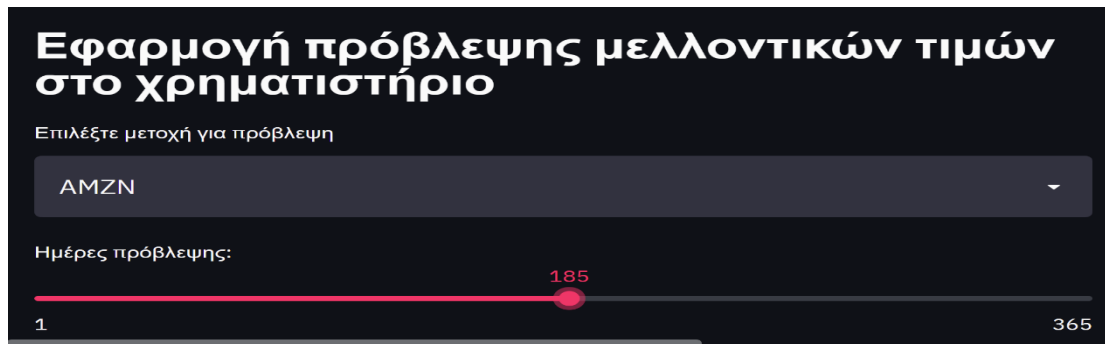
3.3 Συνοπτική παρουσίαση εφαρμογής

Για την υλοποίηση της εφαρμογής αποφασίστηκε η χρήση του μοντέλου NeuralProphet. Παρά το γεγονός ότι το μοντέλο παρουσίασε μικρότερη προβλεπτική ικανότητα κατά το πειραματικό σκέλος, παρουσίασε τη μεγαλύτερη ευαισθησία στην πρόβλεψη μελλοντικών αλλαγών στην τάση της τιμής. Επίσης πολύ σημαντικό είναι και το γεγονός ότι προσφέρει περισσότερες επιλογές παραμετροποίησης (άρα και περισσότερες πιθανότητες μελλοντικής βελτίωσης), ενώ είναι και το λιγότερο απαιτητικό από πλευράς υπολογιστικής ισχύος ανάμεσα στα τρία μοντέλα. Όπως έχει ήδη αναφερθεί, η εφαρμογή είναι διαδικτυακή και στηρίχθηκε πάνω στο πλαίσιο streamlit. Επιπλέον, μέσω της χρήσης της βιβλιοθήκης yfinance προσφέρεται η δυνατότητα στην εφαρμογή να έχει ροή δεδομένων από την ομώνυμη ιστοσελίδα (Yahoo Finance) οικονομικού περιεχομένου. Ο σχεδιασμός διεπαφής χρήστη είναι ιδιαίτερος απλός και παρέχει βασικές πληροφορίες στον επισκέπτη, ο οποίος επιλέγει την μετοχή για την οποία ενδιαφέρεται μέσω ενός σύνθετου πλαισίου (combobox).



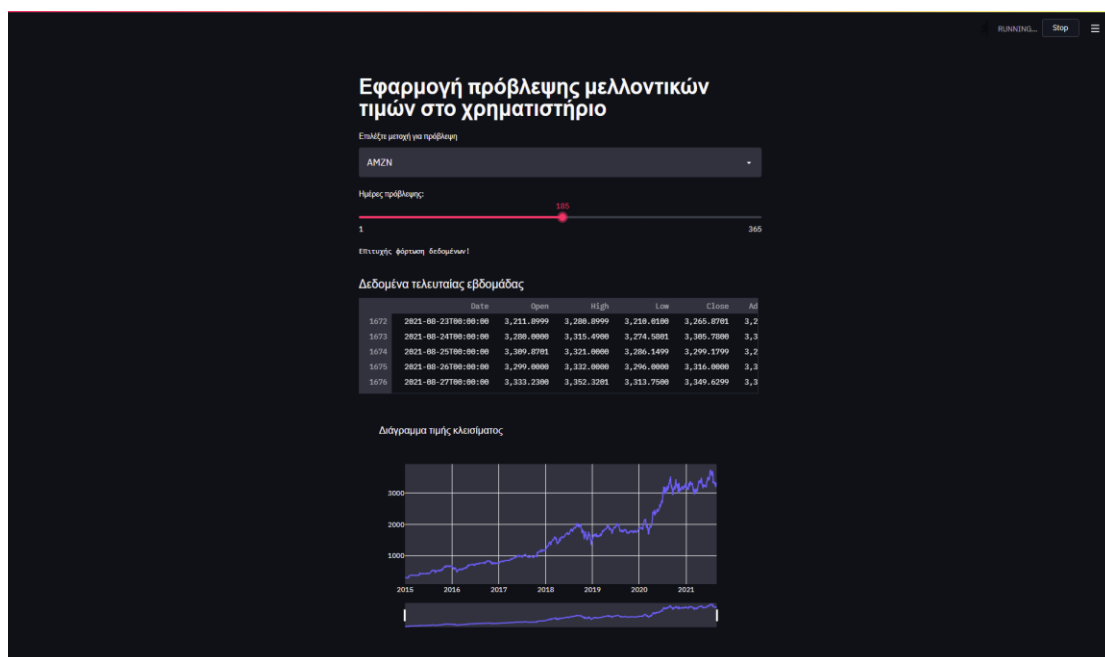
Εικόνα 32 - Επιλογή μετοχής

Αφού έχει επιλέξει μετοχή, στη συνέχεια επιλέγει το χρονικό ορίζοντα πρόβλεψης μέσω ενός κυλιόμενου διακόπτη (εικόνα23).



Εικόνα 33 – Επιλογή χρονικού ορίζοντα πρόβλεψης

Επί της παρούσης έχει επιλεγεί να δοθεί η δυνατότητα πρόβλεψης σε ορίζοντα ενός ημερολογιακού έτους (365 ημέρες), ωστόσο το διάστημα αυτό μπορεί να μεταβληθεί. Αφού, λοιπόν, ο χρήστης έχει επιλέξει τόσο τη μετοχή η οποία τον ενδιαφέρει όσο και το επιθυμητό διάστημα πρόβλεψης, η εφαρμογή του παρουσιάζει κάποιες χρήσιμες πληροφορίες σχετικά με την τιμή της μετοχής την τελευταία εβδομάδα.



Εικόνα 34 – Πληροφορίες για την τιμή της επιλεγμένης μετοχής

Επιπλέον, παρέχεται και ένα διαδραστικό διάγραμμα που απεικονίζει την τιμή κλεισίματος και δίνει τη δυνατότητα χρονικής αναπροσαρμογής των δεδομένων μέσω ενός κυλιόμενου διακόπτη (εικόνα 25).



Εικόνα 35 – Διαδραστικό διάγραμμα τιμής κλεισίματος

Στην εικόνα 30 βλέπουμε την τελική πληροφορία που λαμβάνει ο χρήστης και απαρτίζεται από ένα πίνακα με τις πέντε πιο πρόσφατες τιμές πρόβλεψης για το χρονικό διάστημα επιλογής, καθώς επίσης και μια διαγραμματική απεικόνιση της πορείας της μετοχής, συμπεριλαμβανομένης της πρόβλεψης του μοντέλου. Στο παράδειγμα μας έχουμε εκτελέσει μια πρόβλεψη για την τιμή της μετοχής της Amazon και σύμφωνα με το μοντέλο, η τιμή της εταιρίας αναμένεται να φτάσει τα \$4234,6885 στις 28/02/2022.



Εικόνα 36 – Πληροφορίες πρόβλεψης

Συμπεράσματα

Κατά τη διάρκεια του πειραματικού σκέλους τα μοντέλα LSTM και MLP παρουσίασαν μικρότερα σφάλματα πρόβλεψης στο σύνολό τους και κατά συνέπεια μεγαλύτερη προβλεπτική ικανότητα συγκριτικά με το NeuralProphet. Ωστόσο, παρατηρήθηκε πως η πιθανότητα να παρουσιαστεί το φαινόμενο του overfitting σε ρηγά δίκτυα, είναι μεγαλύτερη για τα μοντέλα MLP και LSTM και άρα απαιτείται μεγαλύτερη προσοχή στην εφαρμογή του για την αποφυγή του. Overfitting έχουμε όταν τα αποτελέσματα της ανάλυσης (πρόβλεψης) είναι πολύ κοντά (ή ακόμη και σχεδόν ίδια) με τις πραγματικές τιμές του συνόλου δεδομένων, κάτι το οποίο σημαίνει πως το μοντέλο θα δυσκολευτεί να παράγει ακριβείς προβλέψεις όταν το γενικεύσουμε.

Παρόλο, βέβαια, που το NeuralProphet παρουσίασε υψηλότερα σφάλματα πρόβλεψης, αξίζει να σημειωθεί ότι οι πολύ περισσότερες επιλογές παραμετροποίησης που διαθέτει συγκριτικά με τα άλλα μοντέλα προσφέρουν τη δυνατότητα μεγαλύτερης βελτίωσης του μοντέλου, ειδικά σε αναλυτές που έχουν βαθιά γνώση του τομέα τον οποίο αφορούν τα δεδομένα της πρόβλεψης. Επιπλέον, το μοντέλο NeuralProphet παρουσίασε μεγαλύτερη ευαισθησία όσον αφορά την πρόβλεψη μελλοντικών αλλαγών στην τάση των δεδομένων της χρονοσειράς και ήταν σε θέση να εντοπίζει τέτοιοι είδους αλλαγές νωρίτερα από τα άλλα μοντέλα.

Μια άλλη σημαντική παρατήρηση έχει να κάνει με το γεγονός πως το LSTM χρειάζεται σημαντικά υψηλότερο χρόνο εκπαίδευσης και είναι πιο απαιτητικό υπολογιστικά σε σχέση με τα άλλα δυο μοντέλα, με το MLP να ακολουθεί και το NeuralProphet να είναι το ταχύτερο εκ των τριών. Όσον αφορά τους αλγορίθμους βελτιστοποίησης, ο ADAM και ο SGD ήταν αυτοί που παρουσίασαν τα καλύτερα αποτελέσματα, με τον πρώτο να δείχνει ως καλύτερη επιλογή για το LSTM και τον δεύτερο για το MLP. Σχετικά με τους κύκλους εκπαίδευσης αυτό που παρατηρήθηκε είναι ότι η αύξησή τους δεν παίζει ιδιαίτερο ρόλο στην αύξηση της προβλεπτικής ικανότητας του μοντέλου LSTM, σε αντίθεση με το MLP στο οποίο παρατηρήθηκε πως είχε θετική επίδραση.

Εν κατακλείδι, το μοντέλο LSTM και ειδικά το μοντέλο MLP παρουσίασαν μεγαλύτερη προβλεπτική ικανότητα συγκριτικά με το NeuralProphet. Ωστόσο, αξίζει να σημειωθεί πως διαβλέπονται πολύ καλές προοπτικές για το μοντέλο NeuralProphet,



το οποίο βρίσκεται ακόμα σε φάση beta και σε πολύ πρώιμο στάδιο συγκριτικά με τα άλλα δυο μοντέλα. Το γεγονός πως δίνει πολλές επιλογές παραμετροποίησης σε συνδυασμό με την ταχύτητα εκπαίδευσής του, αλλά και την ευκολία υλοποίησης του το καθιστούν μια ελκυστική επιλογή για κάθε αναλυτή, η οποία ακόμα έχει αρκετά περιθώρια βελτίωσης. Στο πλαίσιο της παρούσας εργασίας, το μοντέλο εφαρμόστηκε με διάφορες παραλλαγές παραμέτρων. Ωστόσο, υπάρχουν ακόμη πολλές επιλογές παραμετροποίησης οι οποίες δεν καλύφθηκαν και θα μπορούσαν να αποτελέσουν αντικείμενο μελλοντικής μελέτης, με σκοπό την περαιτέρω βελτίωση του μοντέλου, η οποία με βάση τα όσα παρατηρήθηκαν κατά το πειραματικό σκέλος είναι απόλυτα εφικτή.



Βιβλιογραφικές αναφορές

- [1] Chatfield Christopher. 2000. *Time-series forecasting*. Chapman & Hall/CRC, Boca Raton, Florida.
- [2] P.J. Brockwell. 2010. *Time Series Analysis*. International Encyclopedia of Education (Third Edition), Elsevier, Pages 474-481.
- [3] Springer-Verlag. 2008. *Time Series. In: The Concise Encyclopedia of Statistics*. Springer, New York, NY. DOI: https://doi.org/10.1007/978-0-387-32833-1_401
- [4] Muhammad Imdad Ullah. 2020. *Components of Time Series*. <https://itfeature.com/time-series-analysis-and-forecasting/components-of-time-series>
- [5] Australian Bureau of Statistics. *Time Series Analysis: The Basics*. <https://www.abs.gov.au/websitedbs/d3310114.nsf/home/time+series+analysis:+the+basics>
- [6] Robert H. Shumway, David S. Stoffer. 2014. *Time Series Analysis and its Applications, 3rd edition*. <https://sites.ualberta.ca/~dwiens/stat479/tsa3EZ.pdf>
- [7] Jian Zhao, Fanny Chevalier, Emmanuel Pietriga, Ravin Balakrishnan. 2011. *Exploratory Analysis of Timeseries with ChronoLenses*. IEEE Transactions on Visualization and Computer Graphics, Institute of Electrical and Electronics Engineers.
- [8] influxdata.com. *Time series analysis methods*. <https://www.influxdata.com/time-series-analysis-methods/>
- [9] Fulcher BD, Little MA, Jones NS. 2013. *Highly comparative time-series analysis: the empirical structure of time series and their methods*. DOI: <http://dx.doi.org/10.1098/rsif.2013.0048>
- [10] Lambert H. Koopmans. 1995. *The Spectral Analysis of Time Series*. DOI: <https://doi.org/10.1016/B978-0-12-419251-5.X5000-5>
- [11] O. Brandes, J. Farley, M. Hinich and U. Zackrisson. 1968. *The Time Domain and the Frequency Domain in Time Series Analysis*. <https://doi.org/10.2307/3438983>
- [12] Michael G. Sovereign, Richard L. Nolan, James P. Mandel. 1971. *Applications of spectral analysis*. <https://doi.org/10.1111/j.1540-5915.1971.tb01595.x>
- [13] Don Percival. 2000. *An Introduction to the Wavelet Analysis of Time Series*.
- [14] Howard J. Seltman. 2018. *Experimental Design and Analysis*. <https://www.stat.cmu.edu/~hseltman/309/Book/Book.pdf>
- [15] Mulaik, Stanley. 2001. *The Curve-Fitting Problem: An Objectivist View*. *Philosophy of Science*. 68. 218. 10.1086/392874.



- [16] Ελένη Ι. Κουτσουδάκη. 2016. *Γραμμική και μη γραμμική παλινδρόμηση με εφαρμογές στην R*.
- [17] Robert M. Freund. 2004. *The Steepest Descent Algorithm for Unconstrained Optimization and a Bisection Line-search Method*. Massachusetts Institute of Technology.
- [18] Pierre Geurts. 2001. *Pattern Extraction for Time Series Classification*. University of Liege.
- [19] Yuxia Lei & Zhongqiang Wu. 2020. *Time series classification based on statistical features*. <https://doi.org/10.1186/s13638-020-1661-4>.
- [20] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller. 2019. *Deep learning for time series classification: a review*. <https://doi.org/10.1007/s10618-019-00619-1>.
- [21] M. Riyadi, D. Pratiwi, A. Irawan, K. Fithriasari. 2017. *Clustering Stationary and Non-stationary TimeSeries Based on Auto-correlation Distance of Hierarchical and K-means Algorithms* I. International Journal of Advances in Intelligent Informatics, vol. 3, pp. 154-160, 2017.
- [22] Vihar Kurama. 2020. *An Introduction to Segmentation, Correlation and Time Series Modeling*. <https://builtin.com/data-science/introduction-segmentation-correlation-time-series-modeling>
- [23] Kaushik C Shete, Dr. Amar Buchade. 2019. *Time Series Segmentation Using Two-Stage Clustering Approach*. International Journal of Scientific & Technology Research, vol. 8, issue 09, ISSN 2277-8616
- [24] William S Noble. 2006. *What is a support vector machine*. Nature Biotechnology, vol. 24, no 12.
- [25] George Box, Gwilym Jenkins, Gregory Reinsel, Greta Ljung. 2015. *Time Series Analysis: Forecasting and Control (5th. ed.)*. Wiley, New York, NY.
- [26] Eric Brown. 2019. *Stationary Data Tests for Time Series Forecasting*. <https://pythondata.com/stationary-data-tests-for-time-series-forecasting/>
- [27] Robert Nau. 2020. *Statistical forecasting: notes on regression and time series analysis*. <https://people.duke.edu/~rnau/411home.htm>
- [28] Hyndman, R.J., & Athanasopoulos, G. 2018. *Forecasting: principles and practice, 2nd edition*. OTexts: Melbourne, Australia. [OTexts.com/fpp2](https://www.otexts.com/fpp2).
- [29] Jeffrey Strickland, Ph.D. 2015. *What is Time Series Analysis, Part II*. <https://bicornor.com/2015/05/02/what-is-time-seires-analysis-part-ii/>
- [30] A. K. Jain, Jianchang Mao and K. M. Mohiuddin. 1996. "Artificial neural networks: a tutorial," in *Computer*, vol. 29, no. 3, pp. 31-44. doi: 10.1109/2.485891.



- [31] Sharma, Siddharth & Sharma, Simone & Athaiya, Anidhya. 2020. *ACTIVATION FUNCTIONS IN NEURAL NETWORKS*. International Journal of Engineering Applied Sciences and Technology. 04. 310-316. 10.33564/IJEAST.2020.v04i12.054.
- [32] Michael Nielsen. 2019. *Neural Networks and Deep Learning*. <http://neuralnetworksanddeeplearning.com/index.html>
- [33] S. Abirami, P. Chitra. 2020. *Advances in Computers, Volume 117, Issue 1, Chapter Fourteen*. <https://doi.org/10.1016/bs.adcom.2019.09.007>
- [34] Tomas Mikolov, Stefan Kombrink, Lukas Burget, Jan “Honza” Cernocky, Sanjeev Khudanpur. 2011. *EXTENSIONS OF RECURRENT NEURAL NETWORK LANGUAGE MODEL*. John Hopkins University
- [35] Hasim Sak, Andrew Senior, Francoise Beaufays. 2014. *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling*. Google, USA
- [36] Cho et. al. 2014. *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation*.
- [37] Mike Schuster, Kuldip K. Paliwal. 1997. *Bidirectional Recurrent Neural Networks*. IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 11.
- [38] Keiron O’Shea, Ryan Nash. 2015. *An Introduction to Convolutional Neural Networks*. <https://arxiv.org/pdf/1511.08458.pdf>
- [39] Jeremy Jordan. 2018. *Introduction to autoencoders*. <https://www.jeremyjordan.me/autoencoders/>
- [40] Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V, ... Ng, A. Y. 2012. *Large Scale Distributed Deep Networks*. *NIPS 2012: Neural Information Processing Systems, 1–11*.
- [42] Geoffrey Hinton, Nitish Srivastava, Kevin Swersky. *Neural Networks for Machine Learning – Lecture 6a*. http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
- [42] Kingma, D. P., & Ba, J. L. 2015. *Adam: A Method for Stochastic Optimization*. *International Conference on Learning Representations, 1–13*.
- [43] Eric Brown. 2019. *Stationary Data Tests for Time Series Forecasting*. <https://pythondata.com/stationary-data-tests-for-time-series-forecasting/>
- [44] <http://neuralprophet.com/>
- [45] Oskar Triebe, Nikolay Laptev, Ram Rajagopal. 2019. *AR-Net: A simple Auto-Regressive Neural Network for time-series*. arXiv:1911.12436