



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ
ΤΜΗΜΑ ΜΗΧΑΝΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

Βραχυπρόθεσμη πρόβλεψη ηλεκτρικού φορτίου με χρήση νευρωνικών δικτύων

Μάριος Τασσιός

Διπλωματική Εργασία

Υπεβλήθη για την εκπλήρωση μέρους των απαιτήσεων για την απόκτηση του
Διπλώματος Μηχανολόγου Μηχανικού

Βόλος, Σεπτέμβριος 2023

© 2023 Μάριος Τασσιός

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανολόγων Μηχανικών της Πολυτεχνικής Σχολής του Πανεπιστημίου Θεσσαλίας δεν υποδηλώνει αποδοχή των απόψεων του συγγραφέα (Ν. 5343/32 αρ. 202 παρ. 2).

Εγκρίθηκε από τα μέλη της Τριμελούς Εξεταστικής Επιτροπής:

1. Δρ. Γεώργιος Λυμπερόπουλος
Καθηγητής Στοχαστικών Μεθόδων στη Διοίκηση Παραγωγής
Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
(Επιβλέπων)
2. Δρ. Κωνσταντίνος Αμπουντώλας
Αναπληρωτής Καθηγητής Συστημάτων Αυτομάτου Ελέγχου στη Μηχανολογία
Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας
3. Δρ. Δημήτριος Παντελής
Καθηγητής Στοχαστικών Προτύπων Επιχειρησιακής Έρευνας στη Βιομηχανική
Διοίκηση
Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Ευχαριστίες

Ευχαριστώ θερμά τον επιβλέποντα της διπλωματικής μου εργασίας, Καθηγητή Γεώργιο Λυμπερόπουλο που μου έδωσε την ευκαιρία να ενασχοληθώ με το συγκεκριμένο αντικείμενο, καθώς και για την πολύτιμη βοήθεια και καθοδήγηση, κατά τη διάρκεια της υλοποίησης της.

Θα ήθελα επίσης να ευχαριστήσω και τα υπόλοιπα μέλη της εξεταστικής επιτροπής της διπλωματικής μου εργασίας, Καθηγητές κ. Κωνσταντίνο Αμπουντώλα και κ. Δημήτριο Παντελή, για την προσεκτική ανάγνωση της και τις πολύτιμες υποδείξεις τους. Επίσης, θα ήθελα να ευχαριστήσω όλο το ακαδημαϊκό προσωπικό του τμήματος Μηχανολόγων Μηχανικών για τις πολύτιμες γνώσεις που μου παρείχαν κατά τη διάρκεια της φοίτησής μου.

Ευχαριστώ τους συναδέλφους μου στην εταιρία T.Q.C.C. για τη συνεχή υποστήριξή τους. Επίσης, θα ήθελα να εκφράσω την ευγνωμοσύνη μου στους φίλους μου για τις πολύ όμορφες στιγμές που μου χάρισαν και για την υποστήριξή τους όλα αυτά τα χρόνια των σπουδών μου.

Τέλος, περισσότερο από όλους, θα ήθελα να ευχαριστήσω τους γονείς μου, Αντιγόνη και Θωμά, και την αδερφή μου, Μυρτώ, καθώς και τους υπόλοιπους συγγενείς μου για την απεριόριστη αγάπη και υποστήριξη που λαμβάνω όλα αυτά χρόνια.

Βραχυπρόθεσμη πρόβλεψη ηλεκτρικού φορτίου με χρήση νευρωνικών δικτύων

Μάριος Τασσιός

Τμήμα Μηχανολόγων Μηχανικών, Πανεπιστήμιο Θεσσαλίας

Επιβλέπων: Δρ. Γεώργιος Λυμπερόπουλος

Καθηγητής Στοχαστικών Μεθόδων στη Διοίκηση Παραγωγής

Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την απευθείας βραχυπρόθεσμη πρόβλεψη του ηλεκτρικού φορτίου του Παναμά, με ορίζοντα πρόβλεψης τις 24 ώρες στο μέλλον, χρησιμοποιώντας και συγκρίνοντας διάφορους τύπους νευρωνικών δικτύων. Τέτοιοι τύποι είναι τα τεχνητά νευρωνικά δίκτυα και διάφοροι τύπου ανατροφοδοτούμενων νευρωνικών δικτύων, τα δίκτυα μακράς βραχύχρονης μνήμης (**LSTM**), τα δίκτυα **GRU (Gated Recurrent Unit)** και δύο τύπους ανατροφοδοτούμενων νευρωνικών δικτύων δύο κατευθύνσεων (**Bidirectional LSTM, GRU**). Αρχικά, πραγματοποιείται η προεπεξεργασία των δεδομένων και η δημιουργία της δομής που θα έχουν αυτά που θα τροφοδοτούνται στα μοντέλα. Έπειτα, τα μοντέλα εκπαιδεύονται και δοκιμάζονται χρησιμοποιώντας διάφορες ομάδες χαρακτηριστικών και η αποδοτικότητά τους ελέγχεται με χρήση αποτιμητών πρόβλεψης, συγκρίνοντας τις προβλέψεις με τις πραγματικές τιμές που βρίσκονται σε ένα σύνολο δεδομένων ελέγχου. Όλα τα παραπάνω πραγματοποιούνται με τη γλώσσα προγραμματισμού **Python**.

Λέξεις κλειδιά: Τεχνητά νευρωνικά δίκτυα, ανατροφοδοτούμενα νευρωνικά δίκτυα, βραχυπρόθεσμη απευθείας πρόβλεψη, πρόβλεψη χρονοσειρών.

Short-term electricity load forecasting using neural networks

Marios Tassios

Department of Mechanical Engineering, University of Thessaly

Supervisor: Prof. Georgios Liberopoulos

Professor of Stochastic Methods in Production Management

Abstract

This thesis is about the direct short-term forecasting of electricity load of Panama, with a prediction length of 24 hours in the future, using various types of neural networks. These types are the feed forward neural networks and different types of recurrent neural networks, the Long Short-Term Memory network (LSTM), the Gated Recurrent Unit network (GRU) and two types of Bidirectional recurrent neural networks (Bidirectional LSTM, GRU). Initially, the data used for this application were preprocessed and the structure from which they will be fed into the models was constructed. Afterwards, the models mentioned were trained and tested, using different groups of features and their efficiency was tested by comparing the forecasts with the real values, contained in a test set, using various evaluation metrics. The implementation of the above mentioned, was achieved via Python programming language.

Keywords: Feed forward neural networks, recurrent neural networks, direct short term forecasting, time series forecasting.

Περιεχόμενα

1	Εισαγωγή	1
1.1	Κίνητρο και υπόβαθρο	1
1.2	Έξυπνα ηλεκτρικά δίκτυα (Smart Grids)	2
1.3	Βιβλιογραφική ανασκόπηση	6
1.4	Οργάνωση διπλωματικής εργασίας	7
2	Τεχνητή νοημοσύνη, μηχανική μάθηση και βαθιά μάθηση	8
2.1	Τεχνητή νοημοσύνη	9
2.2	Μηχανική μάθηση	10
2.2.1	Κατασκευάζοντας ένα μοντέλο μηχανικής μάθησης	12
2.3	Βαθιά μάθηση	13
2.4	Το πρόβλημα πρόβλεψης χρονοσειρών	15
3	Θεωρητικό υπόβαθρο	16
3.1	Τεχνητά νευρωνικά δίκτυα (Feed-forward neural networks)	17
3.1.1	Συναρτήσεις ενεργοποίησης (Activation functions)	18
3.1.2	Multi-layer perceptrons (MLPs) και βαθιά FFNNs	20
3.1.3	Εκπαίδευση ενός FFNN	21
3.2	Γενικά για την εκπαίδευση νευρωνικών δικτύων	24
3.3	Ανατροφοδοτούμενα νευρωνικά δίκτυα (Recurrent Neural Networks - RNNs)	30
3.3.1	Εκπαίδευση ενός RNN	31
3.3.2	Το πρόβλημα της εκτόξευσης/ελάττωσης των μερικών παραγώγων	33
3.4	Δίκτυα μακράς βραχύχρονης μνήμης (LSTM)	33
3.5	Δίκτυα Gated Recurrent Unit (GRU)	35
3.6	Ανατροφοδοτούμενα νευρωνικά δίκτυα δύο κατευθύνσεων (Bidirectional RNNs)	36
3.7	Μετρητές αποτίμησης προβλέψεων (Evaluation metrics)	38
4	Περιβάλλον υλοποίησης και δεδομένα	41
4.1	Περιβάλλον υλοποίησης	42

4.2	Περιγραφή δεδομένων	43
4.3	Ανάλυση δεδομένων	44
5	Πειράματα και αποτελέσματα	58
5.1	Περιγραφή πειραμάτων	59
5.2	Μεθοδολογία και προεπεξεργασία δεδομένων	60
5.3	Αποτελέσματα	67
5.3.1	Απλοϊκή μέθοδος	67
5.3.2	1ο πείραμα	69
5.3.3	2ο πείραμα	73
5.3.4	3ο πείραμα	77
6	Συμπεράσματα και μελλοντικές κατευθύνσεις	86
A'	Παράρτημα	88
	Βιβλιογραφία	91

Κατάλογος Σχημάτων

1.1	Δομή ενός Smart Grid	4
1.2	Δομή ενός συμβατικού συστήματος παραγωγής ενέργειας	5
2.1	Τεχνητή νοημοσύνη, μηχανική μάθηση και βαθιά μάθηση	9
2.2	Machine Learning vs Classical Programming	10
2.3	Συνοπτικό διάγραμμα ροής εκπαίδευσης νευρωνικού δικτύου	14
3.1	Perceptron με πολλά inputs	17
3.2	Sigmoid activation function	18
3.3	Tanh activation function	19
3.4	ReLU activation function	19
3.5	Multi-layer perceptron με δύο hidden layers	20
3.6	Forward pass σε FFNN με δύο hidden layers	22
3.7	Backward pass για υπολογισμό παραγώγων σε FFNN με δύο hidden layers	23
3.8	Overfitting - επιθυμητή προσαρμογή - Underfitting	25
3.9	α) Ολόκληρο νευρωνικό δίκτυο, β) Νευρωνικό δίκτυο με dropout	26
3.10	Παράδειγμα early stopping	27
3.11	Ψευδοκώδικας του adam optimizer από το πρωτότυπο paper	28
3.12	Δομή ενός RNN	31
3.13	Δομή ενός LSTM	34
3.14	Δομή ενός GRU	36
3.15	Δομή ενός Bidirectional GRU	37
3.16	Δομή ενός Bidirectional LSTM	37
3.17	Περιγραφή της διαδικασίας εκπαίδευσης των Bidirectional RNNs από το πρωτότυπο paper	38
4.1	Δείγμα χαρακτηριστικών του συνόλου δεδομένων	45
4.2	Τύπος χαρακτηριστικών	45
4.3	Έλεγχος κενών τιμών	45
4.4	Φορτίο ηλεκτρικού ρεύματος - χρόνος	47
4.5	Boxplot ανά ημέρα της εβδομάδας	48
4.6	Boxplot ανά ώρα της ημέρας	48

4.7 Μέσος όρος φορτίου ανά ημέρα της εβδομάδας - ώρα της ημέρας	49
4.8 Παρατηρήσεις κάτω από 720 MWh	50
4.9 Διαγράμματα ημερών με outliers	50
4.10 ACF διάγραμμα	51
4.11 PACF διάγραμμα	52
4.12 Πίνακας συσχέτισης συνόλου δεδομένων (correlation matrix)	53
4.13 Ημερήσιο μοτίβο και εβδομαδιαία εποχικότητα	54
4.14 Μέσο μηνιαίο φορτίο ανά έτος	54
4.15 Μέσο μηνιαίο φορτίο όλων των ετών	55
4.16 Μέση μηνιαία θερμοκρασία ανά έτος	55
4.17 Μέση μηνιαία σχετική υγρασία ανά έτος	56
4.18 Μέσος μηνιαίος υετός ανά έτος	56
4.19 Μέση μηνιαία ταχύτητα αέρα ανά έτος	56
5.1 Παράδειγμα απευθείας πρόβλεψης χρησιμοποιώντας παρελθοντικές τιμές	61
5.2 Δομή δεδομένων κινούμενου παραθύρου	62
5.3 Τεχνική κινούμενου παραθύρου	62
5.4 Δεδομένα εισόδου στο πρόβλημα πρόβλεψης χρονοσειρών	63
5.5 Δομή μοντέλου FFNN	64
5.6 Δομή μοντέλων ανατροφοδοτούμενων δικτύων	64
5.7 Διαχωρισμός δεδομένων σε εκπαίδευσης και ελέγχου	64
5.8 Διαγράμματα σφάλματος εκπαίδευσης και επαλήθευσης - αριθμός εποχών	66
5.9 Ενδεικτικές προβλέψεις απλοϊκής μεθόδου	68
5.10 Ενδεικτικές προβλέψεις LSTM για <i>lookback</i> = 72	70
5.11 Ενδεικτικές προβλέψεις GRU για <i>lookback</i> = 24	71
5.12 Ενδεικτικές προβλέψεις Bidirectional LSTM για <i>lookback</i> = 24	71
5.13 Ενδεικτικές προβλέψεις Bidirectional GRU για <i>lookback</i> = 72	72
5.14 Ενδεικτικές προβλέψεις FFNN για <i>lookback</i> = 48	72
5.15 Ενδεικτικές προβλέψεις LSTM για 8 σημαντικότερα χαρακτηριστικά . .	74
5.16 Ενδεικτικές προβλέψεις GRU για 8 σημαντικότερα χαρακτηριστικά . . .	75
5.17 Ενδεικτικές προβλέψεις Bidirectional LSTM για 8 σημαντικότερα χαρα- κτηριστικά	75
5.18 Ενδεικτικές προβλέψεις Bidirectional GRU για 8 σημαντικότερα χαρα- κτηριστικά	76
5.19 Ενδεικτικές προβλέψεις FFNN για 8 σημαντικότερα χαρακτηριστικά . .	76
5.20 Ενδεικτικές προβλέψεις LSTM για feature engineering στα σημαντικά χαρακτηριστικά	79
5.21 Ενδεικτικές προβλέψεις GRU για feature engineering στα σημαντικά χα- ρακτηριστικά	79

5.22	Ενδεικτικές προβλέψεις Bidirectional LSTM για feature engineering στα σημαντικά χαρακτηριστικά	80
5.23	Ενδεικτικές προβλέψεις Bidirectional GRU για feature engineering στα σημαντικά χαρακτηριστικά	80
5.24	Ενδεικτικές προβλέψεις FFNN για feature engineering στα σημαντικά χαρακτηριστικά	81
5.25	Ενδεικτικές προβλέψεις LSTM για feature engineering σε όλα τα χαρακτηριστικά	82
5.26	Ενδεικτικές προβλέψεις GRU για feature engineering σε όλα τα χαρακτηριστικά	82
5.27	Ενδεικτικές προβλέψεις Bidirectional LSTM για feature engineering σε όλα τα χαρακτηριστικά	83
5.28	Ενδεικτικές προβλέψεις Bidirectional GRU για feature engineering σε όλα τα χαρακτηριστικά	83
5.29	Ενδεικτικές προβλέψεις FFNN για feature engineering σε όλα τα χαρακτηριστικά	84

Κατάλογος Πινάκων

4.1	Χαρακτηριστικά συνόλου δεδομένων και περιγραφές	44
4.2	Βασικά στατιστικά των χαρακτηριστικών	46
5.1	Υπερπαραμέτροι RNNs	65
5.2	Υπερπαραμέτροι FFNN	65
5.3	Υπερπαραμέτροι Early stopping RNNs	65
5.4	Υπερπαραμέτροι Early stopping FFNN	66
5.5	Αποτιμητές προβλέψεων για το σύνολο ελέγχου με την απλοϊκή μέθοδο	67
5.6	Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 1ου πειράματος με <i>lookback</i> = 24	69
5.7	Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 1ου πειράματος με <i>lookback</i> = 48	70
5.8	Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 1ου πειράματος με <i>lookback</i> = 72	70
5.9	Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 2ου πειράματος . .	74
5.10	Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 3ου πειράματος, χρησιμοποιώντας τα σημαντικά χαρακτηριστικά	78
5.11	Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 3ου πειράματος, χρησιμοποιώντας όλα τα χαρακτηριστικά	81

1.1 Κίνητρο και υπόβαθρο

Το ηλεκτρικό ρεύμα είναι, αδιαμφισβήτητα, ένα από τα πιο σημαντικά και απαραίτητα εργαλεία της καθημερινότητας. Είναι υπεύθυνο για το πολύ υψηλό επίπεδο ζωής του ανθρώπου, τη σημερινή εποχή, και η σημαντικότητα του φαίνεται σε διάφορους τομείς της, όπως στην ανάπτυξη των επιστημών, των τεχνολογιών και της οικονομίας, στην παραγωγή αγαθών και γενικότερα στην απλούστευση πάρα πολλών δραστηριοτήτων που πριν κάποια χρόνια, έμοιαζαν αρκετά πολύπλοκες για αυτόν.

Λόγω του ότι είναι ένα αναπόσπαστο κομμάτι της καθημερινότητας, υπάρχει μεγάλη ανάγκη για τη χρήση του παγκοσμίως, τόσο σε επίπεδο νοικοκυριών όσο και σε επίπεδο βιομηχανίας, με την κάθε περίπτωση να έχει μεγάλες διαφοροποιήσεις στις απαιτήσεις σε αυτό. Με τη συνεχή ανάπτυξη των τεχνολογιών, η κατανάλωση της ηλεκτρικής ενέργειας επιτυγχάνεται όλο και πιο εύκολα, αυξάνοντας ακόμη περισσότερο τις απαιτήσεις για τη διαθεσιμότητα της. Μερικά χαρακτηριστικά παραδείγματα είναι η χρήση μηχανών υψηλής τάσεως από τις βιομηχανίες, η θέρμανση και ο φωτισμός των κατοικιών, η χρήση οικιακών συσκευών, η φόρτιση ηλεκτρικών αυτοκινήτων και πολλά άλλα. Συνεπώς, για την αποφυγή υπερφόρτωσης του ηλεκτρικού δικτύου, το οποίο είναι υπεύθυνο για την διανομή του ηλεκτρικού ρεύματος στον τελικό χρήστη, αλλά και για την παροχή της απαιτούμενης ποσότητας του χωρίς να υπάρξουν ελλείψεις, είναι πολύ σημαντικός ο συνεχής σχεδιασμός και η οργάνωση του συστήματος ενέργειας, λαμβάνοντας υπόψη, για παράδειγμα, την παραγωγή του από τα αρμόδια εργοστάσια (εργοστάσια παραγωγής ηλεκτρικού ρεύματος από ορυκτά καύσιμα, από υδροηλεκτρική ενέργεια, από πυρηνική ενέργεια και από ανανεώσιμες πηγές ενέργειας), τη διατήρηση του εντός του δικτύου και τη ζήτησή του.

Ένα κομμάτι, και μάλιστα σημαντικό, του σχεδιασμού του ηλεκτρικού δικτύου, είναι και η πρόβλεψη της κατανάλωσης του ηλεκτρικού φορτίου και ειδικότερα η βραχυπρόθεσμη, αντικείμενο το οποίο εκπονείται στην παρούσα διπλωματική εργασία. Το αντικείμενο της πρόβλεψης βραχυπρόθεσμων τιμών ηλεκτρικού φορτίου (**Short-Term Load Forecasting**) μελετάται σε πολύ μεγάλο βαθμό τα τελευταία χρόνια, λόγω της ανάγκης για την ύπαρξη προβλεπτικών μοντέλων μεγαλύτερης ακρίβειας, τα οποία θα οδηγήσουν στον αποτελεσματικότερο σχεδιασμό και λειτουργία του συστήματος ηλεκτρισμού, ως προς την κάλυψη των απαιτήσεων των καταναλωτών. Η βραχυπρόθεσμη πρόβλεψη πραγματοποιείται από 1 έως 24 ώρες στο μέλλον, αλλά μέχρι και 1 εβδομάδα αργότερα. Μέσω αυτής δίνεται η δυνατότητα λήψης αποφάσεων σε σύντομο χρονικό διάστημα, όσον αφορά την οικονομική διανομή φορτίου από τα εργοστάσια παραγωγής ηλεκτρικής ενέργειας, τις αγορές καυσίμων, τις τιμές του ηλεκτρικού ρεύματος, ενώ πραγματοποιείται έλεγχος του ηλεκτρικού δικτύου σε πραγματικό χρόνο, αλλά και δύναται να αντιμετωπιστούν πιθανές μαζικές διακοπές ρεύματος. Με την ακριβή βραχυπρόθεσμη πρόβλεψη του φορτίου από μια έως και 24 ώρες μελλοντικά, δίνεται η δυνατότητα στους παρόχους να αντιληφθούν το ποσό της ενέργειας που πρέπει να παραχθεί, έτσι ώστε να καλυφθούν οι ανάγκες των καταναλωτών, όσο πιο αξιόπιστα και οικονομικά γίνεται. Κακή εκτίμηση του μπορεί να οδηγήσει σε διακοπές ρεύματος, αλλά και γενικότερα αναξιόπιστη λειτουργία του ηλεκτρικού δικτύου, ενώ πετυχαίνοντας ακριβείς προβλέψεις επιτυγχάνεται η καλύτερη διαχείριση του, αλλά και η μείωση του κόστους του ρεύματος από τον πάροχο. Βέβαια, η βραχυπρόθεσμη πρόβλεψη του ηλεκτρικού φορτίου μπορεί να θεωρηθεί ως ένα αρκετά απαιτητικό πρόβλημα, καθώς είναι αρκετά δύσκολη η μοντελοποίηση του, λόγω εμφάνισης μη γραμμικών μοτίβων. Επιπρόσθετα, υπάρχουν πολλοί παράγοντες, οι οποίοι συσχετίζονται με την τιμή της κατανάλωσης του ηλεκτρικού φορτίου, όπως η θερμοκρασία, η ταχύτητα του αέρα, η σχετική υγρασία, η ώρα της ημέρας κ.α..[1]

Ο στόχος της παρούσας διπλωματικής εργασίας είναι η χρήση διάφορων μοντέλων νευρωνικών δικτύων για την βραχυπρόθεσμη πρόβλεψη του φορτίου ηλεκτρικού ρεύματος, συγκεκριμένα 24 ωρών στο μέλλον, και η σύγκριση τους με χρήση διάφορων αποτιμητών, πάνω σε ένα σύνολο δεδομένων που προέρχεται από την ιστοσελίδα **Kaggle**, το οποίο αναφέρεται στο ηλεκτρικό δίκτυο του Παναμά.

1.2 Έξυπνα ηλεκτρικά δίκτυα (Smart Grids)

Στις αρχές του 20ου αιώνα εγκαθιδρύθηκαν τα πρώτα συστήματα παραγωγής ενέργειας, τα οποία μέχρι και σήμερα αποτελούν τον κύριο τρόπο διανομής του ηλεκτρικού ρεύματος στον τελικό καταναλωτή, μέσω των ηλεκτρικών δικτύων. Πιο αναλυτικά, ένα τέτοιο σύστημα αποτελείται από την παραγωγή, το μετασχηματισμό και την διανομή

του ηλεκτρικού ρεύματος. Αρχικά, η παραγωγή πραγματοποιείται κυρίως από συμβατικούς σταθμούς, όπως θερμικοί, υδροηλεκτρικοί, φυσικού αερίου και πυρηνικοί, με τιμές μεταξύ των 11 kV έως 33 kV. Έπειτα, μέσω του συστήματος μετασχηματιστών πραγματοποιείται η ανύψωση της τάσης του σε τιμές μεταξύ των 132-400 kV, ανάλογα με την απόσταση στην οποία δύναται να διανεμηθεί και οδηγείται σε ένα δευτερεύον σύστημα μετασχηματιστών, όπου γίνεται η μείωση της τάσης του σε τιμές περίπου 66 kV, αλλά και πραγματοποιείται η διανομή του σε καταναλωτές με υψηλές απαιτήσεις σε φορτίο (π.χ. μεγάλες βιομηχανίες). Επόμενο βήμα είναι η μεταφορά του στο σύστημα διανομής, πράγμα το οποίο επιτυγχάνεται μέσω του πρωτεύοντος συστήματος διανομής, στο οποίο περιέχεται ακόμη ένα σύστημα μετασχηματιστών που προκαλεί περεταίρω πτώση της τάσης του στην τιμή των 11-33 kV, το οποίο διανέμεται σε καταναλωτές με μεσαίες απαιτήσεις σε φορτίο, αλλά και μέσω του δευτερεύοντος συστήματος, όπου η πτώση είναι πολύ μεγαλύτερη και καταλήγει στις τιμές των 220-400 V, με τη διανομή του να γίνεται στους καταναλωτές με τις μικρότερες απαιτήσεις (οικιακή χρήση). Στην εικόνα 1.2, απεικονίζεται αυτό ακριβώς που περιγράφηκε παραπάνω.[2]

Το παραπάνω σύστημα είναι μεν αποδοτικό, αλλά αρκετά παθητικό, εκτός από την περίπτωση των καταναλωτών με αρκετά μεγάλη ζήτηση, δηλαδή διανέμεται συνεχώς ηλεκτρικό ρεύμα σε περιπτώσεις που δεν υπάρχει ανάγκη χωρίς αυτό να ελέγχεται και να ρυθμίζεται. Με τη μεγάλη ανάπτυξη των τηλεπικοινωνιών και του διαδικτύου δίνεται η δυνατότητα για δημιουργία ενός ηλεκτρικού δικτύου, το οποίο θα μπορέσει να αντιμετωπίσει προβλήματα σαν και αυτό (Εικόνα 1.1). Το δίκτυο αυτό ονομάζεται έξυπνο ηλεκτρικό δίκτυο και είναι μια μετατροπή του συμβατικού, έχοντας τα παρακάτω χαρακτηριστικά:

- Παρέχει την αναγκαία ποσότητα ηλεκτρικού φορτίου, σύμφωνα με τις ανάγκες των καταναλωτών, μέσω της χρήσης μετρητών, από τα οποία λαμβάνονται τόσο ενδείξεις της κατανάλωσης αλλά και του κόστους. Έχοντας γνώση για το φορτίο, το οποίο χρησιμοποιεί ένας καταναλωτής, έχει τη δυνατότητα τροποποίησης της κατανάλωσης του, για την αποφυγή προβλημάτων, όπως η υπερφόρτωση του ηλεκτρικού δικτύου.
- Περιλαμβάνει την προερχόμενη από τις ανανεώσιμες πηγές ενέργεια, αλλά και την αποθηκεύει με σκοπό την επίτευξη ελάττωσης της παραγωγής ηλεκτρικής ενέργειας με τον παραδοσιακό τρόπο.
- Έχει τη δυνατότητα εναλλαγής μετασχηματιστών και συστήματος διανομής.
- Μειώνει το κόστος παραγωγής ηλεκτρικής ενέργειας από τους σταθμούς, αφού έχει ως κύριο σκοπό τη χρήση των ανανεώσιμων πηγών ενέργειας και των αποθηκών αυτής.

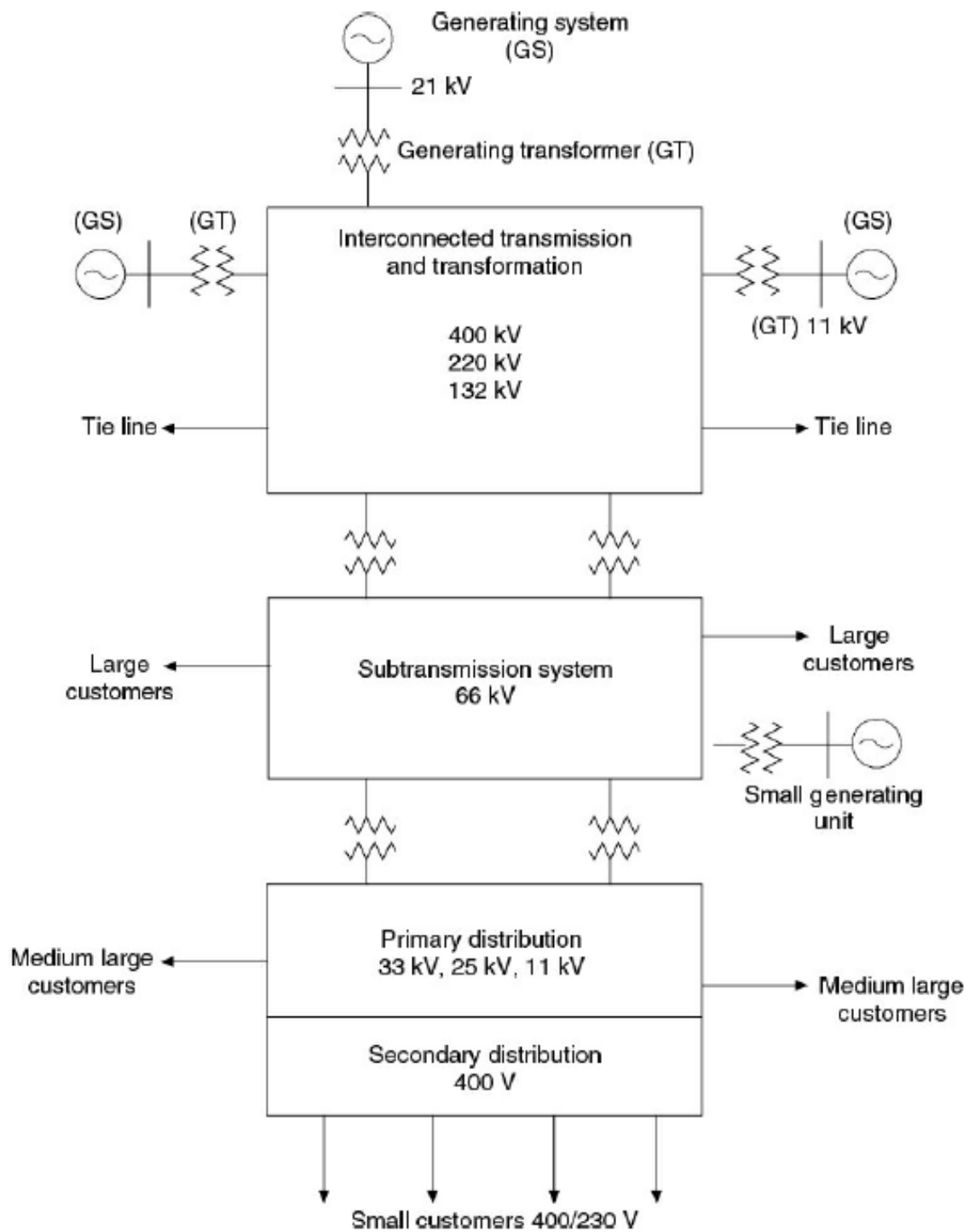
- Διασφαλίζει και βελτιώνει την αξιοπιστία και την ασφάλεια κατά την παροχή του ρεύματος, αναγνωρίζοντας τυχόν διαταραχές στο σύστημα, φυσικές καταστροφές, αλλά και τις προλαμβάνει μέσω της προληπτικής συντήρησης και τις αντιμετωπίζει με διορθωτικές ενέργειες. Γενικότερα, διασφαλίζει ότι το ρεύμα θα φτάσει στον καταναλωτή με ασφάλεια.
- Προστατεύει το περιβάλλον.

Smart Grid



Σχήμα 1.1: Δομή ενός Smart Grid

Μπορεί να γίνει, λοιπόν, εύκολα αντιληπτό ότι για το βέλτιστο σχεδιασμό ενός τέτοιου δικτύου, το οποίο βασίζεται στη λήψη πληροφοριών από τους καταναλωτές, έχει πολύ μεγάλη σημασία η γνώση της ζήτησης του ηλεκτρικού φορτίου εκ των προτέρων [3].



Σχήμα 1.2: Δομή ενός συμβατικού συστήματος παραγωγής ενέργειας

1.3 Βιβλιογραφική ανασκόπηση

Η βραχυπρόθεσμη πρόβλεψη φορτίου ηλεκτρικού ρεύματος είναι ένα αντικείμενο που συναντάται σε πολύ μεγάλο βαθμό στη βιβλιογραφία, καθώς έχει αποτελέσει προϊόν πολλών ερευνών. Τα προβλεπτικά μοντέλα, τα οποία χρησιμοποιούνται στη συγκεκριμένη εφαρμογή ποικίλλουν, από στατιστικά έως και μηχανικής και βαθιάς μάθησης, αλλά ακόμη και υβριδικά. Αρχικά, συναντώνται δύο μελέτες σχετικές με την βραχυπρόθεσμη πρόβλεψη του φορτίου του Παναμά. Η πρώτη διενεργήθηκε από τους **Bibi Ibrahim, Luis Rabelo et al** [1] χρησιμοποιώντας πέντε διαφορετικά μοντέλα μηχανικής μάθησης (**SVR, XGBoost, AdaBoost, Random Forest, LightGBM**) και ένα μοντέλο **feed forward neural network** για την πρόβλεψη μιας ώρας στο μέλλον, χρησιμοποιώντας για το τελευταίο το λογισμικό **Knime**, ένα λογισμικό κατασκευής μοντέλων βαθιάς μάθησης χωρίς τη χρήση κώδικα. Συμπληρωματικά, χρησιμοποιήθηκαν και δύο μοντέλα βαθιάς μάθησης (**Bi-LSTM, GRU**) για την πρόβλεψη 24 ωρών στο μέλλον, τα αποτελέσματα των οποίων θα μπορούσαν να αποτελέσουν ένα μέτρο σύγκρισης με την παρούσα διπλωματική εργασία. Η δεύτερη μελέτη, πάνω στο ίδιο σύνολο δεδομένων διενεργήθηκε από τους **Ernesto Aguilar Madrid και Nuno Antonio** [4] εξετάζοντας την απόδοση μοντέλων μηχανικής μάθησης στην πρόβλεψη 168 ωρών στο μέλλον με ένα κενό διάστημα 72 ωρών από το τελευταίο δεδομένο εισόδου. Τα μοντέλα αυτά είναι τα **MLR, KNN, SVR, RF, XGB**.

Μια ακόμη μελέτη παρουσιάζεται από τους Γεώργιο Σταματέλλο και Τάσο Σταματέλλο [5] στην οποία χρησιμοποιείται δύο μοντέλα νευρωνικών δικτύων, ένα με ένα κρυφό επίπεδο και ένα βαθύ με δύο κρυφά επίπεδα για την 24ωρη πρόβλεψη του φορτίου της Ελλάδας, χωρίς να χρησιμοποιείται η διόρθωση των απογευματινών ωρών που εφαρμόζεται πολύ συχνά για τη μείωση του σφάλματος των προβλέψεων κατά τις απογευματινές ώρες.

Μια μελέτη στην οποία προτιμάται η χρήση στατιστικών μοντέλων για τη βραχυπρόθεσμη πρόβλεψη 24 ωρών του φορτίου του Ιράν πραγματοποιήθηκε από τον **Nima Amjady** [6], όπου χρησιμοποιείται μια τροποποίηση του μοντέλου **ARIMA**, κατά το οποίο συνδυάζονται οι εκτιμήσεις των χειριστών του ενεργειακού συστήματος, οι οποίες αποτελούν την αρχική πρόβλεψη, με τα δεδομένα θερμοκρασίας και φορτίου, μέσω μιας παλινδρόμησης πολλών μεταβλητών για τη βελτίωση των προβλέψεων του παραδοσιακού **ARIMA**.

Τέλος, μια μελέτη της βιβλιογραφίας, όπου γίνεται χρήση ενός υβριδικού μοντέλου, πραγματοποιήθηκε από τους **Jinliang Zhang et al** [7]. Στην εν λόγω έρευνα χρησιμοποιούνται δεδομένα από το φορτίο της Αυστραλίας και της Νέας Υόρκης για την πρόβλεψη της τιμής της επόμενης ημέρας. Η μεθοδολογία βασίζεται στην ξεχωριστή

μελέτη και πρόβλεψη του γραμμικού μέρους της χρονοσειράς, το οποίο είναι η τάση της μέσω ενός μοντέλου **ARIMA** και το μη-γραμμικό μέρος μέσω ενός νευρωνικού δικτύου τύπου **wavelet**, το οποίο έχει υψηλή ικανότητα εντοπισμού του μη-γραμμικού μέρους της αρχικής χρονοσειράς.

1.4 Οργάνωση διπλωματικής εργασίας

Η υπόλοιπη διπλωματική εργασία χωρίζεται σε 5 επιπλέον Κεφάλαια. Συγκεκριμένα, στο Κεφάλαιο 2 πραγματοποιείται μια εισαγωγή στην τεχνητή νοημοσύνη, τη μηχανική μάθηση και τη βαθιά μάθηση. Γίνεται μια σύντομη περιγραφή τους, όσον αφορά τις βασικές αρχές τους και τον τρόπο με τον κατασκευάζονται τα μοντέλα μάθησης, αλλά και αναφέρονται διάφορες εφαρμογές στα οποία συναντώνται. Στο Κεφάλαιο 3, περιέχεται το θεωρητικό υπόβαθρο όλων των μοντέλων νευρωνικών δικτύων τα οποία δύναται να χρησιμοποιηθούν, καθώς και οι αποτιμητές των προβλέψεων. Στο Κεφάλαιο 4, παρατίθενται πληροφορίες σχετικές με το περιβάλλον υλοποίησης, παρουσιάζονται τα δεδομένα της εφαρμογής και πραγματοποιείται η εξερευνητική ανάλυση τους. Στο Κεφάλαιο 5 παρουσιάζεται η μεθοδολογία, η οποία χρησιμοποιήθηκε για την κατασκευή των μοντέλων, αλλά και τα αποτελέσματα των παραγόμενων προβλέψεων τους. Τέλος, στο Κεφάλαιο 6 παρουσιάζονται τα τελικά συμπεράσματα της διπλωματικής εργασίας και κάποιες πιθανές μελλοντικές κατευθύνσεις για περεταίρω έρευνα.

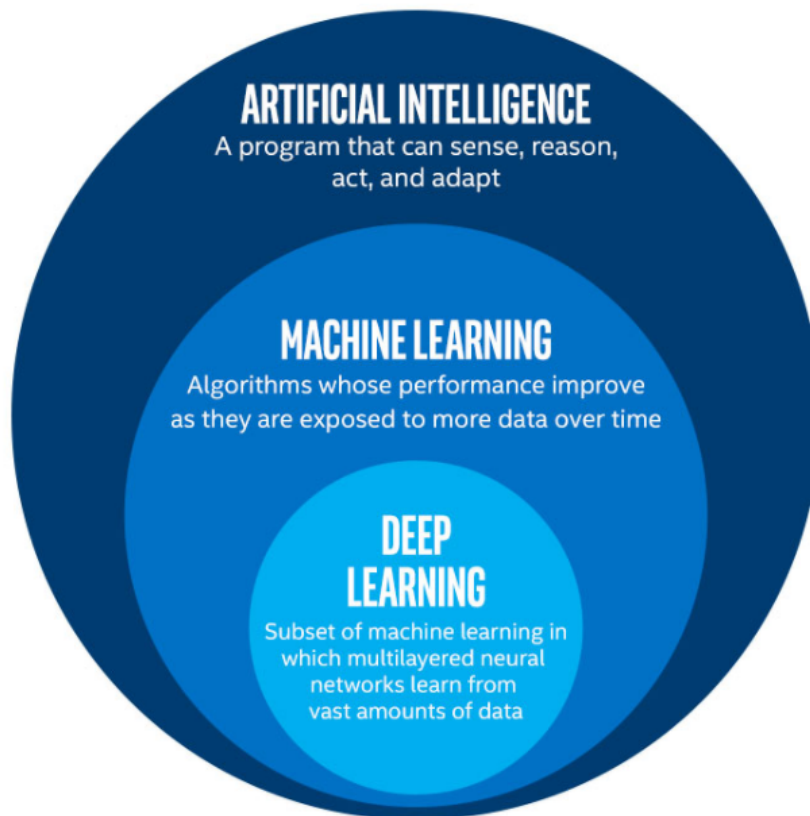
2

Τεχνητή νοημοσύνη, μηχανική μάθηση και βαθιά μάθηση

Ενότητες

2.1	Τεχνητή νοημοσύνη	9
2.2	Μηχανική μάθηση	10
2.3	Βαθιά μάθηση	13
2.4	Το πρόβλημα πρόβλεψης χρονοσειρών	15

Τα τελευταία χρόνια, η τεχνητή νοημοσύνη έχει εισχωρήσει για τα καλά στην καθημερινότητα του ανθρώπου, αλλάζοντας, ριζικά, τον τρόπο τον οποίο ζει, εργάζεται και κοινωνικοποιείται. Χρησιμοποιώντας τους υπολογιστές και τις μηχανές ως μέσα, έχει συμβάλει σε τεράστιο βαθμό στην ανάπτυξη και μεταμόρφωση των βιομηχανιών, αλλά και στον τρόπο με τον οποίο ο άνθρωπος έρχεται σε επαφή με την τεχνολογία. Επιπρόσθετα, τα υποπεδία της τεχνητής νοημοσύνης (μηχανική και βαθιά μάθηση) του έχουν δώσει τη δυνατότητα να χειρίζεται τις τεράστιες ποσότητες δεδομένων για την επίλυση περίπλοκων προβλημάτων.



Σχήμα 2.1: Τεχνητή νοημοσύνη, μηχανική μάθηση και βαθιά μάθηση

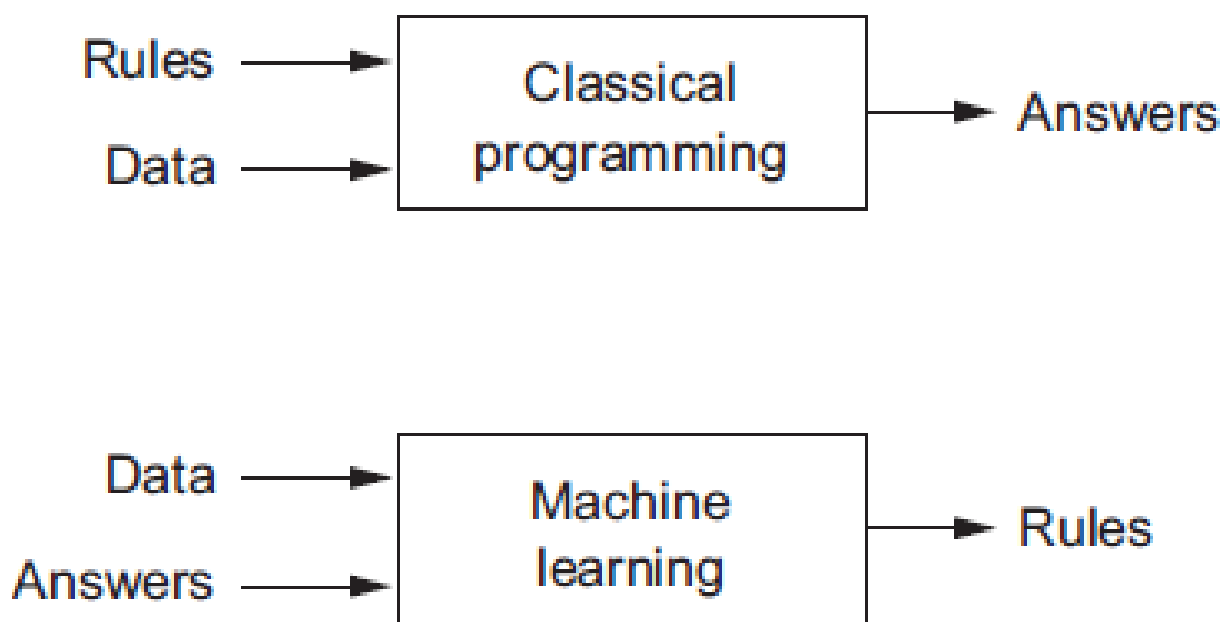
2.1 Τεχνητή νοημοσύνη

Οι πρώτες, σχετικές με την τεχνητή νοημοσύνη, έρευνες φαίνεται να πραγματοποιήθηκαν περίπου το 1950, όπου είχε γεννηθεί σε πολλούς επιστήμονες του τομέα της πληροφορικής, με πρωτοπόρο τον Άλαν Τούρινγκ, ο προβληματισμός του αν οι μηχανές έχουν τη δυνατότητα να αναπτύξουν ικανότητες σκέψης. Ο εν λόγω προβληματισμός, οδήγησε στην εγκαθίδρυση των βάσεων της επιστήμης της τεχνητής νοημοσύνης, η οποία τα τελευταία χρόνια έχει σημειώσει τεράστια ανάπτυξη. Ένας σύντομος ορισμός, ο οποίος θα μπορούσε να δοθεί είναι πως η τεχνητή νοημοσύνη είναι η επιστήμη που έχει

ως στόχο την ανάπτυξη ευφυών συστημάτων, τα οποία θα είναι ικανά να σκέφτονται και να ενεργούν σαν τον άνθρωπο. Διάφορες εφαρμογές της παρατηρούνται σε μηχανές αναζήτησης στο διαδίκτυο (π.χ. Google), σε συστήματα προτιμήσεων του χρήστη (π.χ. YouTube), συστήματα αυτόματου πιλότου (αυτοκίνητα, πλοία, αεροπλάνα), συστήματα αναγνώρισης φωνής, όραση υπολογιστών, στην βιομηχανία παιχνιδιών και πολλά άλλα.

2.2 Μηχανική μάθηση

Ένα πεδίο της τεχνητής νοημοσύνης είναι η μηχανική μάθηση, η οποία θα μπορούσε να οριστεί ως η ανάπτυξη αλγορίθμων ή μεθόδων που δίνουν τη δυνατότητα σε συστήματα να μαθαίνουν αυτόνομα, χωρίς δηλαδή το συνεχή προγραμματισμό ή την ανθρώπινη παρέμβαση, μέσω του χειρισμού των διαθέσιμων δεδομένων. Η συγκεκριμένη επιστήμη, διαφοροποιείται από τον κλασικό προγραμματισμό, καθώς στην προκειμένη δίνονται από το χρήστη τα δεδομένα που εισάγονται στο σύστημα, αλλά και τα αποτελέσματα με σκοπό την εύρεση του κανόνα από τον οποίο αυτά προκύπτουν, διαδικασία η οποία θυμίζει ταιριάζει πιο πολύ με τον όρο μάθηση και όχι τον όρο προγραμματισμός (Εικόνα 2.2). Οι πρώτες εφαρμογές της παρατηρήθηκαν περίπου το 1990, χαρακτηρίζοντας τη σχετικά νέα επιστήμη, αλλά έχει επιτευχθεί σε σύντομο χρονικό διάστημα η ανάπτυξη της, λόγω της μεγαλύτερης διαθεσιμότητας σε δεδομένα, αλλά και σε ισχυρότερα και γρηγορότερα υπολογιστικά συστήματα.



Σχήμα 2.2: Machine Learning vs Classical Programming

Ένας πολύ σημαντικός παράγοντας για την επιλογή του κατάλληλου αλγορίθμου για την εκμάθηση μέσω των δεδομένων είναι η αναγνώριση της κατηγορίας του εκάστοτε προβλήματος, σύμφωνα με το επιθυμητό στόχο του χειριστή. Υπάρχουν τέσσερις κατηγορίες μεθόδων μηχανικής μάθησης, η επιτηρούμενη μάθηση (**supervised learning**), η μη επιτηρούμενη μάθηση (**unsupervised learning**), η ημι-επιτηρούμενη μάθηση (**semi-supervised learning**) και η ενισχυτική μάθηση (**reinforcement learning**). Πιο αναλυτικά:

- **Επιτηρούμενη Μάθηση (Supervised Learning):** Η πιο διαδεδομένη περίπτωση μηχανικής μάθησης, κατά την οποία δίνονται δεδομένα εκπαίδευσης εισόδου μαζί με τις αναμενόμενες τιμές εξόδου. Έχει ως στόχο την εύρεση συσχέτισης μεταξύ αυτών των παραπάνω δεδομένων με σκοπό την πρόβλεψη τιμών από άλλα δεδομένα εισόδου. Τα προβλήματα που εντάσσονται σε αυτή την κατηγορία είναι αυτά της ταξινόμησης (**Classification**), όπου οι προβλεπόμενες μεταβλητές κατηγοριοποιούνται σε 2 ή περισσότερες κλάσεις και της παλινδρόμησης (**Regression**), όπου προβλέπεται η τιμή μια ή περισσότερων μεταβλητών. Διάφοροι τύποι μοντέλων που χρησιμοποιούνται στην προκειμένη περίπτωση είναι τα **Support Vector Machines**, **Random-Forests**, **Neural Networks**, **Boosting Algorithms** κ.α..
- **Μη Επιτηρούμενη Μάθηση (Unsupervised Learning):** Τύπος μηχανικής μάθησης κατά τον οποίο δεν χρησιμοποιούνται δεδομένα εξόδου για κάθε είσοδο, όπως στην επιτηρούμενη μάθηση, αλλά χρησιμοποιούνται τα δεδομένα εκπαίδευσης για την εύρεση της δομής ολόκληρου του σετ δεδομένων, η οποία θα εξηγήσει τις διαφοροποιήσεις που παρατηρούνται σε αυτά. Προβλήματα που εντάσσονται σε αυτή την κατηγορία είναι τα **Clustering**, **Anomaly Detection** και **Dimensionality Reduction**. Τύποι μοντέλων που μπορούν να χρησιμοποιηθούν είναι τα **K-means clustering**, **Auto-Encoders (Neural Networks)** κ.α..
- **Ημι-επιτηρούμενη Μάθηση (Semi-supervised Learning) :** Μια επέκταση των δύο προηγούμενων τύπων μάθησης, όπου εμφανίζεται στην περίπτωση που υπάρχει περιορισμένος αριθμός δεδομένων εξόδου που αντιστοιχούν σε δεδομένα εκπαίδευσης. Σε αυτή την περίπτωση χρησιμοποιούνται όλα τα διαθέσιμα δεδομένα εξόδου για την εκπαίδευση και έπειτα οι νέες τιμές εξόδου που θα προκύψουν θα χρησιμοποιηθούν ως δεδομένα εξόδου και για τα υπόλοιπα δεδομένα εκπαίδευσης. Για τη συγκεκριμένη περίπτωση μπορούν να χρησιμοποιηθούν τα μοντέλα που αναφέρθηκαν παραπάνω.
- **Ενισχυτική Μάθηση (Reinforcement Learning):** Τύπος μηχανικής μάθησης, όπου το η εκμάθηση γίνεται μέσω της αλληλεπίδρασης με το περιβάλλον. Το σύστημα, το οποίο καλείται και πράκτορας (**agent**) λαμβάνει πληροφορίες από το περιβάλλον του και μαθαίνει να αποφασίζει για κάποιες ενέργειες (**actions**) βελτιστοποιώντας μια συνάρτηση επιβράβευσης (**reward**). Ο συγκεκριμένος τύπος μηχανικής μάθη-

σης βρίσκει πολλές εφαρμογές σε βιντεοπαιχνίδια, αλλά και στην ρομποτική και τα αυτομάτως κινούμενα αυτοκίνητα.

2.2.1 Κατασκευάζοντας ένα μοντέλο μηχανικής μάθησης

Για τη δημιουργία ενός μοντέλου, το οποίο θα είναι αποτελεσματικό, όσον αφορά τις προβλέψεις του, απαιτείται να ακολουθηθεί μια αλληλουχία βημάτων. Τα βήματα αυτά είναι:

- Αναγνώριση του προβλήματος: Αρχικά, πριν από τη χρήση οποιουδήποτε αλγορίθμου και εργαλείου επίλυσης, είναι πολύ σημαντική η αναγνώριση του προς επίλυση προβλήματος, αλλά και σε ποια κατηγορία ανήκει, όπως και ο προσδιορισμός των προς εκπλήρωση στόχων με τη χρήση μηχανικής μάθησης.
- Συλλογή δεδομένων: Εύρεση δεδομένων σχετικών με το πρόβλημα, τα οποία θα χρησιμοποιηθούν για την εκπαίδευση του μοντέλου, λαμβάνοντας υπόψη την εγκυρότητά τους, το μέγεθός τους και το βαθμό στον οποίο εκπροσωπούν το πρόβλημα. Τα δεδομένα σε αυτό το στάδιο, συνήθως δεν είναι έτοιμα για την απευθείας χρήση.
- Προεπεξεργασία δεδομένων: Η προεπεξεργασία δεδομένων έχει ως στόχο το να μετατρέψει τα δεδομένα που συλλέχθηκαν σε μια μορφή κατάλληλη για την εκπαίδευση του μοντέλου. Μπορεί να περιλαμβάνει αφαίρεση ή αντικατάσταση κενών τιμών (NaN values), μετατροπή κατηγορικών μεταβλητών σε αριθμητικών, απομάκρυνση των απομακρυσμένων τιμών (Outliers), κανονικοποίηση των τιμών, Feature Engineering και πολλά άλλα. Είναι μια πολύ σημαντική διαδικασία, καθώς είναι αυτή που κυρίως θα προσδώσει την αποτελεσματικότητα στο μοντέλο.
- Επιλογή μοντέλου: Καθορίζεται από το είδος του προβλήματος και πολλές φορές χρειάζεται να επιλεγούν και να συγκριθούν αρκετά μοντέλα για την εύρεση του καταλληλότερου για το κάθε πρόβλημα. Παράγοντες επιλογής είναι ο βαθμός πολυπλοκότητας, υλοποίησης, αλλά και το υπολογιστικό κόστος.
- Εκπαίδευση του μοντέλου: Τροφοδοσία με δεδομένα και χρήση του αλγορίθμου του μοντέλου που επιλέχθηκε για την εκμάθηση των μοτίβων μεταξύ των δεδομένων εισόδου και των δεδομένων εξόδου. Τα δεδομένα αυτά είναι ένα μέρος των συνολικών. Η εκμάθηση αυτή πραγματοποιείται, μέσω της βελτιστοποίησης μιας συνάρτησης κόστους, η οποία αποτελείται από το σφάλμα, προσαρμόζοντας συνεχώς τις εσωτερικές παραμέτρους του.
- Επιλογή υπερπαραμέτρων: Με τη χρήση ενός άλλου μέρους των συνολικών δεδομένων, το οποίο ονομάζεται σετ επικύρωσης, πολύ μικρότερου από αυτά της

εκπαίδευσης, ελέγχεται η απόδοση του μοντέλου πάνω σε αυτό, με σκοπό την εύρεση των βέλτιστων υπερπαραμέτρων (παράμετροι που δεν αλλάζουν κατά την εκπαίδευση) που θα οδηγήσουν στην μεγιστοποίηση της απόδοσής του. Η επικύρωση πραγματοποιείται με την χρήση μέσων αποτίμησης (**evaluation metrics**), όπως το μέσο ποσοστιαίο σφάλμα.

- **Επικύρωση Μοντέλου:** Πραγματοποιείται στα δεδομένα επικύρωσης, δεδομένα τα οποία είναι άγνωστα για μοντέλο και σε αυτό το βήμα ελέγχεται η απόδοση του και η εγκυρότητα των προβλέψεων του, μέσω **metrics**.
- **Δημιουργία προβλέψεων:** Τροφοδοσία με έναν συγκεκριμένο αριθμό δεδομένων στο ήδη εκπαιδευμένο μοντέλο για τη δημιουργία προβλέψεων.

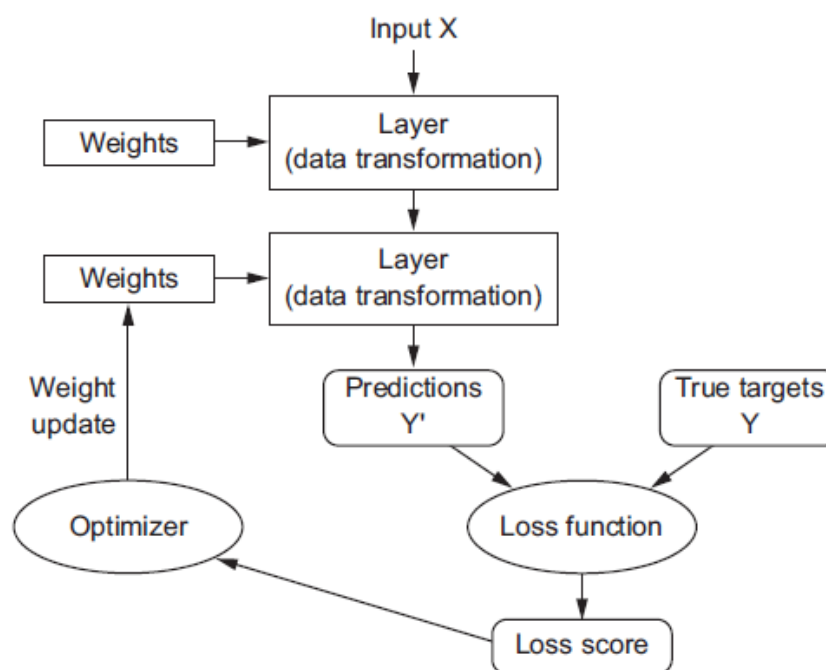
2.3 Βαθιά μάθηση

Ένα πολύ διαδεδομένο πεδίο της μηχανικής μάθησης, το οποίο έχει σημειώσει τεράστια ανάπτυξη τα τελευταία χρόνια είναι η βαθιά μάθηση. Πρόκειται για ένα είδος μάθησης, στο οποίο χρησιμοποιούνται τα τεχνητά νευρωνικά δίκτυα (**Artificial Neural Networks**), τα οποία αποτελούνται από μαθηματικά μοντέλα, με σκοπό την προσομοίωση της δομής των βιολογικών νευρώνων του εγκεφάλου. Η αρχή που επικρατεί στο συγκεκριμένο πεδίο είναι η ίδια με αυτή της μηχανικής μάθησης, δηλαδή ο χειρισμός των δεδομένων για τη δημιουργία προβλέψεων ή τον εντοπισμό συσχετίσεων μεταξύ τους και όλα τα είδη μάθησης που αναφέρθηκαν στην προηγούμενη ενότητα, μπορούν να προσεγγιστούν από αυτό. Η συνεισφορά της βαθιάς μάθησης φαίνεται σε πάρα πολλούς τομείς της σύγχρονης κοινωνίας και των επιστημών με εφαρμογές όπως η αναγνώριση φωνής και προσώπου, η ανάλυση εικόνας και γενικά η όραση των υπολογιστών, η παραγωγή κειμένου, η αυτόματη οδήγηση και πολλά ακόμα.

Τα τεχνητά νευρωνικά δίκτυα, αποτελούνται από διάφορους κόμβους ή νευρώνες (**neurons**), οι οποίοι συνδέονται μεταξύ τους και μέσω των στρωμάτων (**layers**) τα δεδομένα επεξεργάζονται και προκύπτουν προβλέψεις ή αποφάσεις. Οι εν λόγω μαθηματικές διατάξεις, έχουν το τεράστιο πλεονέκτημα σε σχέση με άλλα μοντέλα μηχανικής μάθησης ότι μπορούν μέσω των νευρώνων αυτών να εντοπίσουν, με μεγαλύτερη ευκολία, μη γραμμικότητες στις σχέσεις μεταξύ των δεδομένων, άρα και την πολυπλοκότητα (**complexity**) στο κάθε πρόβλημα πρόβλημα. Επίσης, υπάρχουν διάφορα είδη και αρχιτεκτονικές νευρωνικών δικτύων, τα οποία επιτυγχάνουν με πολύ μεγάλη επιτυχία τα παραπάνω, σε εξειδικευμένες εφαρμογές και προβλήματα. Τα δύο πιο αξιοσημείωτα είναι τα συνελικτικά νευρωνικά δίκτυα (**Convolutional Neural Networks - CNN**), τα οποία εξειδικεύονται σε προβλήματα όρασης υπολογιστών και αναγνώρισης και επεξεργασίας

εικόνων, αλλά και τα αναδρομικά νευρωνικά δίκτυα (**Recurrent Neural Networks - RNN**), και συγκεκριμένα κάποιες παραλλαγές τους, τα οποία εξειδικεύονται σε προβλήματα όπου τα δεδομένα βρίσκονται σε αλληλουχία, όπως αναγνώριση φωνής, επεξεργασία φυσικής γλώσσας και χρονοσειρές, η πρόβλεψη των οποίων είναι και το αντικείμενο της παρούσας διπλωματικής εργασίας.

Τα βήματα που ακολουθούνται στη βαθιά μάθηση είναι όμοια με αυτά της μηχανικής μάθησης, όσον αφορά την προσέγγιση ενός προβλήματος, με μια διαφοροποίηση στο κομμάτι της εκπαίδευσης, σχετικά με τους αλγόριθμους που χρησιμοποιούνται. Περιλιπτικά, κατά την εκπαίδευση, ο σκοπός είναι η βελτιστοποίηση των διάφορων βαρών (**weights**), παραμέτρων που περιέχονται εντός των νευρωνικών δικτύων. Η βελτιστοποίηση αυτή, επιτυγχάνεται, αρχικά, μέσω μιας πρόσθιας διέλευσης (**forward propagation**) των δεδομένων εισόδου, με το νευρωνικό δίκτυο να δίνει αυτόματα κάποιες αρχικές τιμές στα βάρη, και έπειτα να παράγονται κάποιες προβλέψεις, των οποίων το σφάλμα με τις πραγματικές τιμές δημιουργεί τη συνάρτηση σφάλματος. Η εν λόγω συνάρτηση είναι και η συνάρτηση κόστους ενός προβλήματος βελτιστοποίησης. Έπειτα, πραγματοποιείται μια οπίσθια διέλευση (**back propagation**) των προβλέψεων, όπου υπολογίζεται η παράγωγος της συνάρτησης κόστους, συναρτήσει των βαρών και μέσω ενός αλγόριθμου βελτιστοποίησης υπολογίζονται οι νέες τιμές των βαρών. Η διαδικασία επαναλαμβάνεται μέχρις ότου επιτευχθούν οι τιμές οι οποίες θα πετύχουν τις καλύτερες δυνατές προβλέψεις. Η παραπάνω διαδικασία απεικονίζεται από την εικόνα 2.3. [8][9]



Σχήμα 2.3: Συνοπτικό διάγραμμα ροής εκπαίδευσης νευρωνικού δικτύου

2.4 Το πρόβλημα πρόβλεψης χρονοσειρών

Οι χρονοσειρές, είναι μια αλληλουχία ιστορικών δεδομένων, τα οποία συνήθως καταγράφονται σε ένα διακριτό σταθερό χρόνο (ετήσια, μηνιαία, ωριαία κτλ.). Αυτός ο τύπος δεδομένων, έχει τεράστια συσχέτιση με τον κλάδο της στατιστικής, μέσω του οποίου μπορούν να ληφθούν πάρα πολλές πληροφορίες, όσον αφορά τη φύση και τα χαρακτηριστικά τους, αλλά και να χρησιμοποιηθούν διάφορα μοντέλα της για τη δημιουργία προβλέψεων και αναλύσεων. Ένα χαρακτηριστικό παραδείγμα στατιστικού μοντέλου προβλέψεων το οποίο έχει χρησιμοποιηθεί και σε μεγάλο βαθμό στη βιβλιογραφία για την πρόβλεψη ηλεκτρικού φορτίου, είναι το **ARIMA**. Βέβαια, εκτός από τις στατιστικές μεθόδους, η προσέγγιση του προβλήματος αυτού, μπορεί να πραγματοποιηθεί πολύ εύκολα και αποτελεσματικά και από μεθόδους μηχανικής μάθησης, άρα και βαθιάς μάθησης, καθώς εντάσσεται στην κατηγορία της επιβλεπόμενης μάθησης και πιο συγκεκριμένα στην παλινδρόμηση. Στα επόμενα κεφάλαια, θα περιγραφεί λεπτομερώς η διαδικασία με την οποία προσεγγίστηκε το εν λόγω πρόβλημα με μεθόδους βαθιάς μάθησης.

3

Θεωρητικό υπόβαθρο

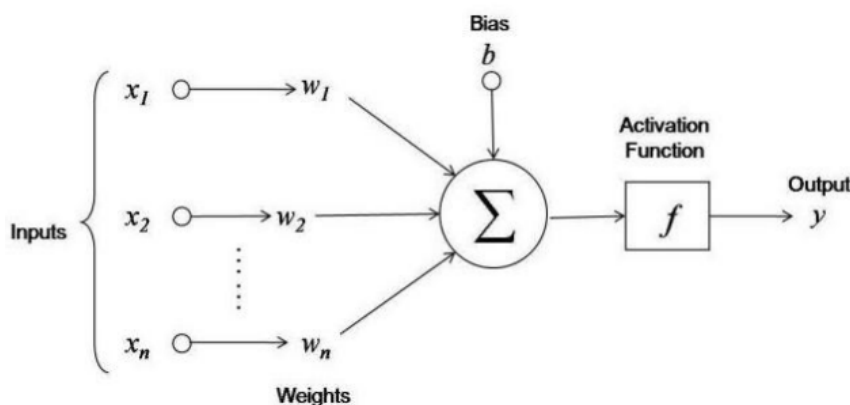
Ενότητες

3.1	Τεχνητά νευρωνικά δίκτυα (Feed-forward neural networks)	17
3.2	Γενικά για την εκπαίδευση νευρωνικών δικτύων	24
3.3	Ανατροφοδοτούμενα νευρωνικά δίκτυα (Recurrent Neural Networks - RNNs)	30
3.4	Δίκτυα μακράς βραχύχρονης μνήμης (LSTM)	33
3.5	Δίκτυα Gated Recurrent Unit (GRU)	35
3.6	Ανατροφοδοτούμενα νευρωνικά δίκτυα δύο κατευθύνσεων (Bidirectional RNNs)	36
3.7	Μετρητές αποτίμησης προβλέψεων (Evaluation metrics)	38

Στα πλαίσια της περιγραφής της βαθιάς μάθησης, η οποία στηρίζεται στη χρήση νευρωνικών δικτύων, έγινε μια περιληπτική αναφορά στον τρόπο με τον οποίο πραγματοποιείται η εκπαίδευση των μοντέλων αυτών, αλλά και στη δομή τους. Στο παρόν κεφάλαιο περιγράφονται σε μεγαλύτερο βάθος τα διάφορα μοντέλα που χρησιμοποιούνται για τη βραχυπρόθεσμη πρόβλεψη του ηλεκτρικού φορτίου, όσον αφορά τη δομή τους, αναφέροντας τις βασικές μαθηματικές σχέσεις που τα αποτελούν, αλλά και η διαδικασία εκπαίδευσής τους.

3.1 Τεχνητά νευρωνικά δίκτυα (Feed-forward neural networks)

Τα τεχνητά νευρωνικά δίκτυα (FFNNs) αποτελούν τον πρώτο τύπο νευρωνικών δικτύων που εισήλθε στον κόσμο της μηχανικής μάθησης. Η συνεχής αύξηση του πλήθους των δεδομένων, της υπολογιστικής ισχύος, αλλά και της αποδοτικότητας των αλγορίθμων τους, έχουν οδηγήσει στη συνεχή ανάγκη για χρήση τους σε εφαρμογές κάθε κατηγορίας μηχανικής μάθησης. Η αρχική ιδέα πίσω από τα FFNNs ήταν η εύρεση ενός υπολογιστικού συστήματος, το οποίο θα προσομοιώνει τη λειτουργία των βιολογικών νευρώνων του εγκεφάλου. Η πρώτη προσέγγιση έγινε από τους **Warren S. McCulloch** και **Walter Pitts** [10] το 1943, παρουσιάζοντας ένα μαθηματικό μοντέλο νευρωνικών δικτύων εμπνευσμένο από τα βιολογικά. Η συγκεκριμένη έρευνα έθεσε τα θεμέλια για την περεταίρω ανάπτυξη του θέματος και οδήγησε στην παρουσίαση του **perceptron** από τον **F. Rosenblatt** [11] το 1958, ο οποίος είναι και ένα από τα βασικά συστατικά του FFNN.



Σχήμα 3.1: Perceptron με πολλά inputs

Ο **perceptron** είναι η απλούστερη μορφή νευρωνικού δικτύου και η αρχική του δυνατότητα ήταν η επίλυση απλών προβλημάτων ταξινόμησης, εντάσσοντας τα δεδομένα σε κατηγορίες 0 και 1. Στα μοντέρνα νευρωνικά δίκτυα, ένας **perceptron** έχει τη μορφή της εικόνας 3.1 και για ένα διάνυσμα το οποίο αποτελείται από **inputs** της μορφής

$x_1, x_2, \dots, x_n$, το **output** υπολογίζεται από την παρακάτω εξίσωση:

$$y = f \left(\sum_{i=1}^n (x_i \cdot w_i) + b \right) \quad (3.1)$$

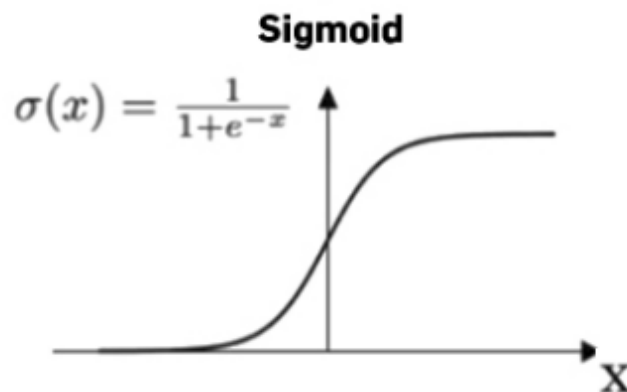
Όπου w_i είναι τα βάρη τα οποία πολλαπλασιάζουν το κάθε **input** και b είναι το **bias**, το οποίο είναι το βάρος ενός τεχνητού **input** που ισούται με τη μονάδα, και συμβάλει στην ενεργοποίηση του νευρώνα όταν τα βάρη πάρουν μηδενικές τιμές. Επιπλέον, για τη γραμμική σχέση της εξίσωσης 3.1, εφαρμόζεται μια συνάρτηση ενεργοποίησης f , της οποίας η λειτουργία θα αναλυθεί παρακάτω.

3.1.1 Συναρτήσεις ενεργοποίησης (Activation functions)

Οι συναρτήσεις ενεργοποίησης είναι συναρτήσεις που εφαρμόζονται στο άθροισμα των βαρών και των **inputs** κάθε νευρώνα, έτσι ώστε να δώσουν στο νευρωνικό δίκτυο τη μη γραμμική του συμπεριφορά. Υπάρχουν πάρα πολλά είδη συναρτήσεων ενεργοποίησης και χρησιμοποιούνται σε κάθε αρχιτεκτονική νευρωνικού δικτύου που συμπεριλαμβάνεται στην παρούσα διπλωματική εργασία. Τα πιο διαδεδομένα περιγράφονται παρακάτω.

Sigmoid/Logistic function Η sigmoid συνάρτηση ενεργοποίησης δίνει τιμές στο διάστημα $(0,1)$ και δίνεται από τον τύπο:

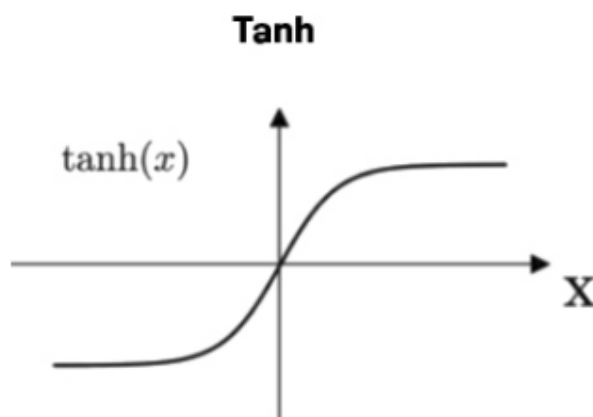
$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$



Σχήμα 3.2: Sigmoid activation function

Tanh function Η συνάρτηση υπερβολικής εφαπτομένης δίνει τιμές στο διάστημα $(-1,1)$ και δίνεται από τον τύπο:

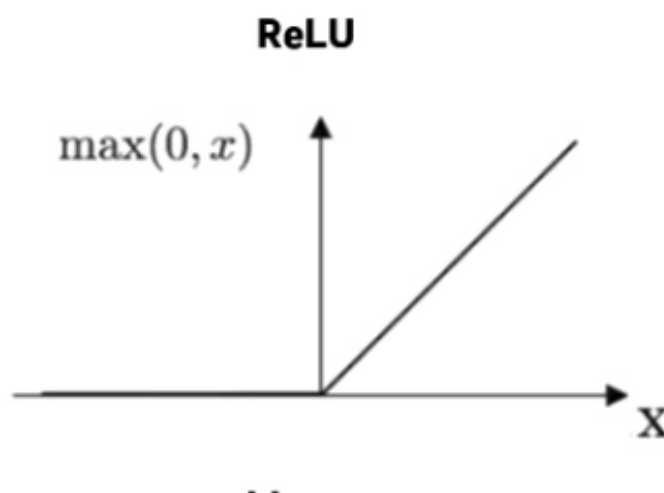
$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3)$$



Σχήμα 3.3: Tanh activation function

ReLU function Η συνάρτηση Rectified Linear Unit δίνει μηδενικές τιμές για αρνητικά x και θετικές τιμές για θετικά x και δίνεται από τον τύπο:

$$\text{ReLU}(x) = \max(0, x) \quad (3.4)$$

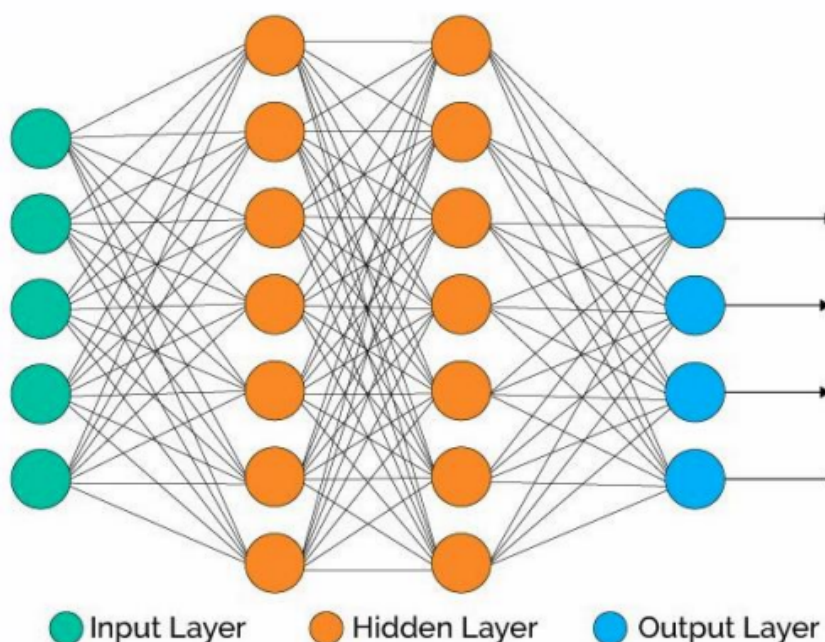


Σχήμα 3.4: ReLU activation function

Η επιλογή του είδους της συνάρτησης ενεργοποίησης εξαρτάται κάθε φορά από το είδος του προβλήματος, την αρχιτεκτονική του νευρωνικού δικτύου, αλλά και τη φύση των δεδομένων που χρησιμοποιούνται. Σε κάθε περίπτωση, όμως, πρόκειται για ένα σημαντικό μέρος ενός νευρωνικού δικτύου, καθώς εντάσσει τη μη-γραμμικότητα μεταξύ των δεδομένων.

3.1.2 Multi-layer perceptrons (MLPs) και βαθιά FFNNs

Καθώς η αρχιτεκτονική του **perceptron** έχει τη δυνατότητα επίλυσης πολύ απλών προβλημάτων, δημιουργήθηκε η ανάγκη εύρεσης μιας αρχιτεκτονικής που θα προσφέρει τη δυνατότητα αναγνώρισης πιο περίπλοκων μοτίβων μεταξύ των δεδομένων. Μια τέτοια αρχιτεκτονική είναι τα **Multi-layer perceptrons (MLPs)**. Τα **MLPs**, είναι νευρωνικά δίκτυα που περιέχουν **perceptrons** κατανεμημένους σε στρώματα (**layers**), τα οποία ονομάζονται **hidden layers**. Μέσω αυτών των στρωμάτων η πληροφορία που προσέρχεται από τα **inputs** τροφοδοτείται από τον ένα **perceptron** στον άλλο και από αυτόν στον επόμενο, καταλήγοντας στο **output layer**. Η διάταξη αυτή φαίνεται στην παρακάτω εικόνα (3.5).



Σχήμα 3.5: Multi-layer perceptron με δύο hidden layers

Αυξάνοντας τον αριθμό των νευρώνων και των **hidden layers**, δημιουργούμε το λεγόμενο βαθύ **FFNN**, μια αρχιτεκτονική η οποία έχει τεράστιες δυνατότητες, όσον αφορά την εκμάθηση από πολύ περίπλοκα δεδομένα. Όλες αυτές οι διατάξεις εντάσσονται στην κατηγορία των **FFNNs**, στα οποία η πληροφορία ρέει και επεξεργάζεται από **layer** σε **layer**, μέσω της εξίσωσης 3.1, έχοντας μόνο κατεύθυνση προς τα εμπρός, καταλήγοντας στο **output layer**. Κάθε **layer** περιέχει τον δικό του πίνακα βαρών, με στοιχεία τα βάρη από τον νευρώνα i στο νευρώνα j , αλλά και το διάνυσμα των **biases**, με στοιχεία το **bias** του κάθε νευρώνα που περιέχεται στο **layer**. Επομένως η εξίσωση 3.1 μπορεί

να γενικευθεί στο επίπεδο των βαθύων FFNNs ως:

$$x_j^{(\ell)} = f \left(\sum_{i=1}^{d^{(\ell-1)}} w_{ij}^{(\ell)} x_i^{(\ell-1)} + b_j^{(\ell)} \right) \quad (3.5)$$

Και σε μορφή πινάκων:

$$X^{(\ell)} = f \left(W^{(\ell)T} X^{(\ell-1)} + B^{(\ell)} \right), \quad (3.6)$$

Όπου χρησιμοποιήθηκαν τα:

$$X^{(\ell)} = \left(x_1^{(\ell)}, \dots, x_{d^\ell}^{(\ell)} \right)^T, \quad W^{(\ell)} = \left(w_{ij}^{(\ell)} \right)_{i,j}, \quad B^{(\ell)} = \left(b_1^{(\ell)}, \dots, b_{d^\ell}^{(\ell)} \right)^T,$$

με $1 \leq i \leq d^{(\ell-1)}$ και $1 \leq j \leq d^{(\ell)}$. Τα $X^{(0)} = \left(x_1^{(0)}, \dots, x_{d^0}^{(0)} \right)^T$ θεωρούνται ως το διάνυσμα των **inputs**, ενώ το διάνυσμα των **outputs** ως $X^{(L)} = \left(x_1^{(L)}, \dots, x_{d^L}^{(L)} \right)^T$. Τέλος, ως $d^{(\ell)}$ θεωρείται ο αριθμός των νευρώνων ενός **layer** ℓ . [12]

3.1.3 Εκπαίδευση ενός FFNN

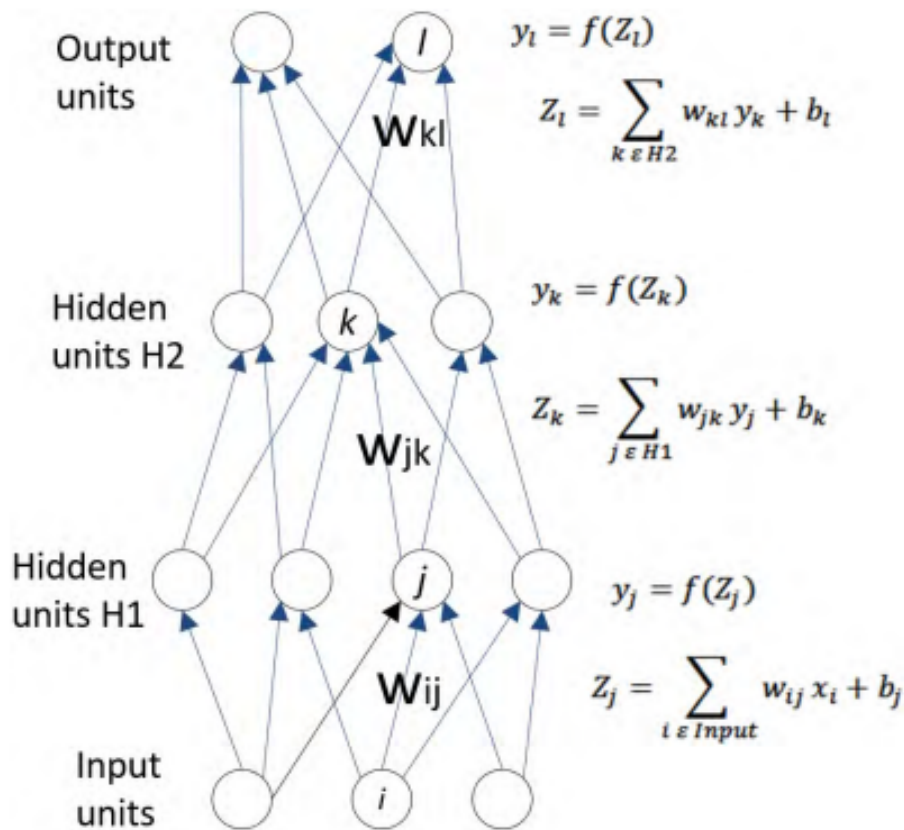
Πρωτού γίνει η περιγραφή της διαδικασίας εκπαίδευσης του συγκεκριμένου τύπου νευρωνικού δικτύου, αλλά και γενικότερα κάθε τύπου νευρωνικού δικτύου που χρησιμοποιείται στην παρούσα διπλωματική εργασία, καθώς οι διαφοροποιήσεις είναι μικρές, είναι πολύ σημαντικό να προσδιοριστεί ο επιθυμητός στόχος που προκύπτει από αυτή. Από την ανάλυση της δομής ενός νευρωνικού δικτύου και του νευρώνα, ο οποίος αποτελεί το υπολογιστικό μέρος του, γίνεται αντιληπτή η επιρροή των τιμών w_{ij} και b στις τιμές που θα λάβουν τα **outputs**. Οι συγκεκριμένες τιμές χαρακτηρίζονται ως παράμετροι του νευρωνικού δικτύου, οι οποίες μεταβάλλονται με τέτοιον τρόπο ώστε να προκύψουν τα οι βέλτιστες τιμές εξόδου. Συνεπώς, ο στόχος που προκύπτει από την εκπαίδευση είναι η εύρεση των βέλτιστων **weights** και **biases**, σύμφωνα με κάποιον κανόνα, ο οποίος είναι μια αντικειμενική συνάρτηση που θα αναλυθεί στη συνέχεια.

Πέραν των παραμέτρων που αναφέρθηκαν, οι οποίες είναι μεταβαλλόμενες εσωτερικές παράμετροι του νευρωνικού δικτύου, υπάρχουν και οι λεγόμενες υπερπαραμέτροι (**hyperparameters**), οι οποίες επιλέγονται, αρχικά και δεν μεταβάλλονται κατά την εκπαίδευση. Αυτές σχετίζονται τόσο με την αρχιτεκτονική του νευρωνικού δικτύου (π.χ. αριθμός νευρώνων και **layers**), όσο και με την εκπαίδευση (π.χ. **learning rate**, **batch size**, **number of epochs** κτλ) και η σωστή επιλογή τους συμβάλλει στην αποδοτικότητα του μοντέλου, αλλά και στο πόσο θα καταφέρει αυτό να εκπαιδευτεί. Επομένως, είναι αρκετά σημαντική οι σωστή επιλογή τους και μάλιστα η διαδικασία αυτή χαρακτηρίζεται

ως **hyperparameter tuning**. Στη συνέχεια, περιγράφεται η διαδικασία εκπαίδευσης ενός FFNN, η οποία θα σχετίζεται με ένα πρόβλημα επιτηρούμενης μάθησης, που είναι και το αντικείμενο της παρούσας διπλωματικής εργασίας.

Ο πιο διαδεδομένος αλγόριθμος που χρησιμοποιείται για τη διαδικασία εκπαίδευσης ονομάζεται **backpropagation algorithm** [13] και βασίζεται στον υπολογισμό των μερικών παραγώγων της αντικειμενικής συνάρτησης ως προς τις παραμέτρους προς βελτιστοποίηση. Τα βήματα που ακολουθούνται είναι, το **forward pass**, το **backward pass** και ο υπολογισμός των παραγώγων και τέλος η χρήση ενός αλγόριθμου βελτιστοποίησης, μέσω του οποίου πραγματοποιείται η ανανέωση των τιμών των παραμέτρων. Πιο αναλυτικά:

1. Forward pass : Πρώτο βήμα του αλγορίθμου, στο οποίο τα **inputs** επεξεργάζονται μέσω των **hidden layers**, σύμφωνα με τις εξισώσεις 3.5 ή 3.6 και υπολογίζονται τα **outputs**. Πιο απλά, είναι μια διαδικασία που τα **outputs** εκφράζονται, μέσω των παραπάνω εξισώσεων, από τα **inputs** του νευρωνικού δικτύου. Η συγκεκριμένη διαδικασία πραγματοποιείται δίνοντας τυχαίες τιμές στα **weights** και **biases** (**weight initialization**). Το βήμα αυτό μπορεί να ονομαστεί και ως **forward propagation**. Στην εικόνα 3.6 (Πηγή: [14]) παρουσιάζεται ένα παράδειγμα **forward pass** ενός FFNN με 3 hidden layers.

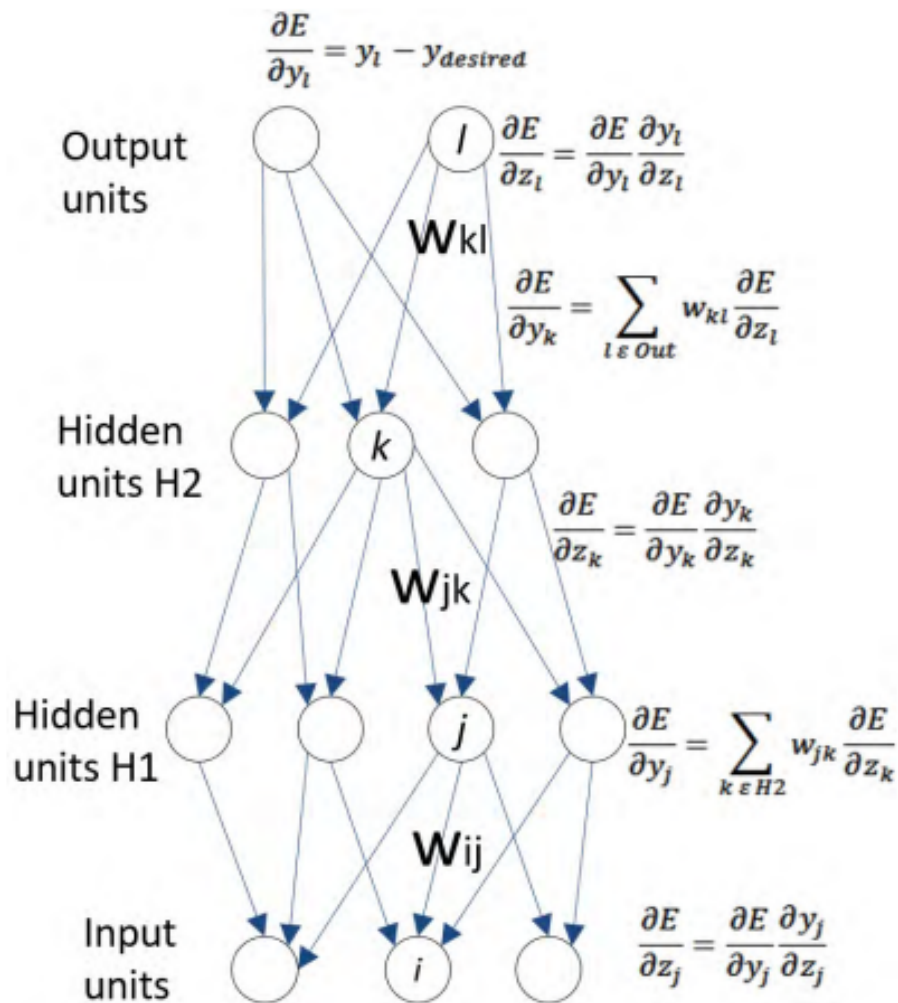


Σχήμα 3.6: Forward pass σε FFNN με δύο hidden layers

2. Cost function : Με την ολοκλήρωση του υπολογισμού των **outputs** , σειρά παίρνει ο υπολογισμός της συνάρτησης κόστους, η οποία είναι και η αντικειμενική συνάρτηση του προβλήματος βελτιστοποίησης. Η συνάρτηση αυτή αποτελείται από τη συνάρτηση σφάλματος και σε περίπτωση που χρησιμοποιείται κάποια τεχνική **regularization** , η οποία συμβάλλει στην αποφυγή του **overfitting** των δεδομένων, πρόβλημα το οποίο θα αναλυθεί στη συνέχεια. Ανάλογα με το είδος της εφαρμογής, χρησιμοποιούνται και διαφορετικά ήδη συνάρτησης σφάλματος. Στην προκειμένη, για μια εφαρμογή **regression** , όπως είναι η προβλέψεις χρονοσειρών, χρησιμοποιείται το μέσο τετραγωνικό σφάλμα (**Mean Squared Error - MSE**) που δίνεται από τον τύπο:

$$\text{Loss} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.7)$$

όπου y_i η πραγματική τιμή, $\hat{y}_i$ η προβλεπόμενη τιμή, δηλαδή αυτή που προέρχεται από το forward pass και n ο αριθμός των **outputs**.



Σχήμα 3.7: Backward pass για υπολογισμό παραγώγων σε FFNN με δύο hidden layers

3. Backward pass : Βήμα στο οποίο ξεκινάει η διαδικασία της βελτιστοποίησης. Μετά τον υπολογισμό της συνάρτησης κόστους, στόχος είναι και η εύρεση των παραμέτρων που θα δώσουν και το ελάχιστο σφάλμα, πράγμα που επιτυγχάνεται με τον υπολογισμό των μερικών παραγώγων της ως προς τα **weights** και **biases**. Έπειτα οι ποσότητες αυτές χρησιμοποιούνται σε κάποιον αλγόριθμο βελτιστοποίησης (**gradient descent**, **stochastic gradient descent** κτλ.), μέσω του οποίου εφαρμόζεται η ανανέωση των τιμών των παραμέτρων αυτών. Ο υπολογισμός των μερικών παραγώγων πραγματοποιείται με τον κανόνα της αλυσίδας, ξεκινώντας από το **output layer** και καταλήγοντας στο **input layer** ακολουθώντας μια αντίθετη από το **forward pass** διαδρομή. Ένα παράδειγμα του υπολογισμού των παραγώγων του σφάλματος ακολουθώντας μια οπίσθια διαδρομή παρατηρείται στην εικόνα 3.7. Οι παράγωγοι που υπολογίζονται συμβάλλουν και στον υπολογισμό της συνάρτησης κόστους, συναρτήσεως των προς βελτιστοποίηση παραμέτρων.

4. Weights and biases update : Το τελευταίο βήμα του αλγορίθμου, στο οποίο ανανεώνονται οι τιμές των παραμέτρων, έτσι ώστε να συγκλίνει σε αυτές που θα δώσουν τις πιο αποτελεσματικές προβλέψεις. Η ανανέωση αυτή πραγματοποιείται με τη χρήση κάποιου **optimizer**. Το πιο απλό είδος **optimizer** είναι ο **Gradient Descent** [15], κατά το οποίο οι αλλαγές των τιμών πραγματοποιούνται σύμφωνα με τους τύπους:

$$\theta = \theta - \alpha \cdot \nabla_{\theta} C(\theta) \quad (3.8)$$

όπου θ οι παράμετροι **weights** και **biases** και α ο ρυθμός εκμάθησης (**learning rate**). Με το πέρασμα των ετών, άρχισαν να χρησιμοποιούνται πιο περίπλοκοι και, κάποιες φορές, πιο αποτελεσματικοί **optimizers** ως παραλλαγές του, όπως ο **Adam** που χρησιμοποιείται και στην εργασία αυτή.

Τα βήματα, τα οποία περιγράφηκαν παραπάνω αποτελούν μια μόνο επανάληψη του αλγορίθμου και συνήθως χρειάζονται πολλές επαναλήψεις για να επιτευχθεί η σύγκλιση. Όταν ολοκληρωθεί ένα 'πέρασμα' όλων των δεδομένων που περιέχονται στο **training dataset**, τότε ολοκληρώνεται και μια εποχή. Ο αριθμός των εποχών επιλέγεται πριν την έναρξη της εκπαίδευσης. Με την άφιξη του αλγορίθμου στο μέγιστο αριθμό εποχών που επιλέχθηκε, τότε ολοκληρώνεται και η εκπαίδευση.

3.2 Γενικά για την εκπαίδευση νευρωνικών δικτύων

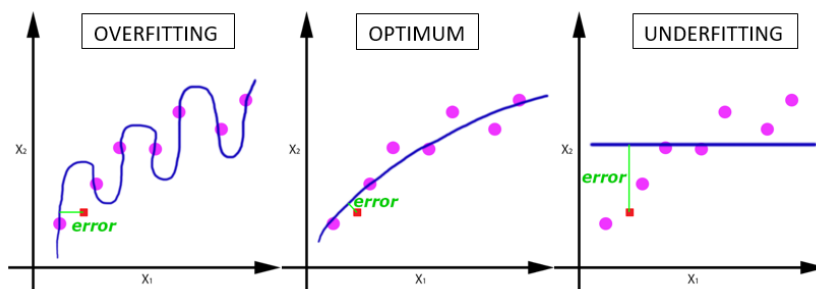
Στην ενότητα αυτή, ως συνέχεια στην προηγούμενη υποενότητα, αναφέρονται έννοιες και τεχνικές, οι οποίες σχετίζονται με την εκπαίδευση κάθε τύπου και αρχιτεκτονικής νευρωνικού δικτύου και έχουν χρησιμοποιηθεί στο πειραματικό μέρος της συγκεκρι-

μένης διπλωματικής εργασίας.

Overfitting - Underfitting : Δύο προβλήματα που προκύπτουν κατά την εκπαίδευση μοντέλων τόσο μηχανικής, όσο και βαθιάς μάθησης. Γενικά, κατά την εκπαίδευση ενός μοντέλου, χρησιμοποιείται ένα μέρος (συνήθως 10%-20%) του **training dataset**, για τον έλεγχο της εκπαίδευσης του μοντέλου σε δεδομένα τα οποία δεν έχει “αναγνώσει”, το **validation dataset**. Συνήθως, για τον εντοπισμό των δύο αυτών προβλημάτων είναι σημαντική η ταυτόχρονη χρήση των σετ δεδομένων αυτών. Ένας πολύ καλός τρόπος εντοπισμού τους είναι μέσω της κατασκευής ενός διαγράμματος της τιμής της συνάρτησης κόστους συναρτήσει των εποχών που θα περιέχει τόσο το σφάλμα του **training** όσο και αυτό του **validation dataset**.

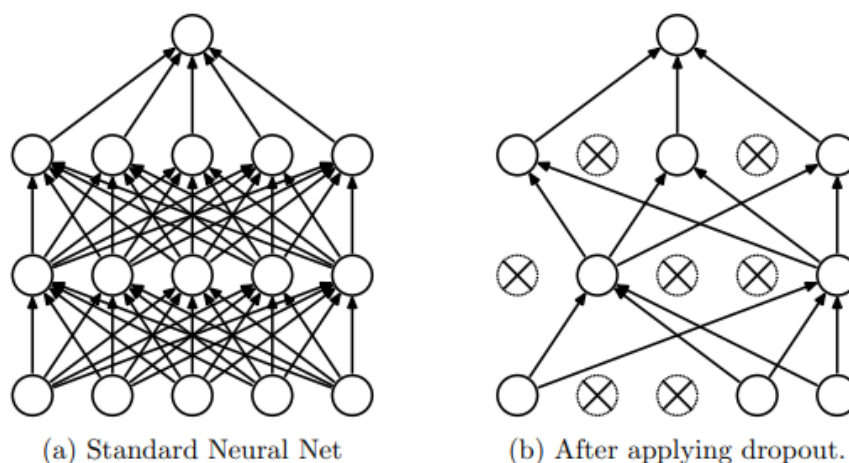
- **Overfitting**: Ορίζεται ως η υπερβολική προσαρμογή των δεδομένων εκπαίδευσης, δηλαδή την υπερβολική εκμάθηση των δεδομένων αυτών, όντας, κατά συνέπεια, μη αποτελεσματικό σε άγνωστα στο μοντέλο δεδομένα, όπως αυτά του **validation dataset**. Οι συνηθέστεροι λόγοι για τους οποίους εμφανίζεται είναι η επιλογή μιας πολύ περίπλοκης αρχιτεκτονικής νευρωνικού δικτύου, αλλά και η χρήση πολύ μεγάλου αριθμού εποχών κατά την εκπαίδευση.
- **Underfitting**: Ορίζεται ως η πολύ φτωχή προσαρμογή του μοντέλου στα δεδομένα εκπαίδευσης, δηλαδή η αδυναμία του εντοπίσει τα μοτίβα μεταξύ των **inputs** και των **outputs**. Εμφανίζεται μεγάλη αναποτελεσματικότητα και στο **training** αλλά και στο **validation dataset**. Συνήθως, συναντάται λόγω χρήσης πολύ απλής αρχιτεκτονικής νευρωνικού δικτύου ή και πολύ μικρού αριθμού εποχών κατά την εκπαίδευση, αλλά και η επιλογή πολύ μικρού αριθμού δεδομένων εκπαίδευσης.

Επομένως, για την επιτυχημένη εκπαίδευση ενός μοντέλου μηχανικής ή βαθιάς μάθησης, είναι πολύ σημαντικό να βρεθεί η “χρυσή τομή” ανάμεσα σε αυτά τα δύο προβλήματα. Στην εικόνα 3.8 παρατηρείται ένα παράδειγμα των παραπάνω



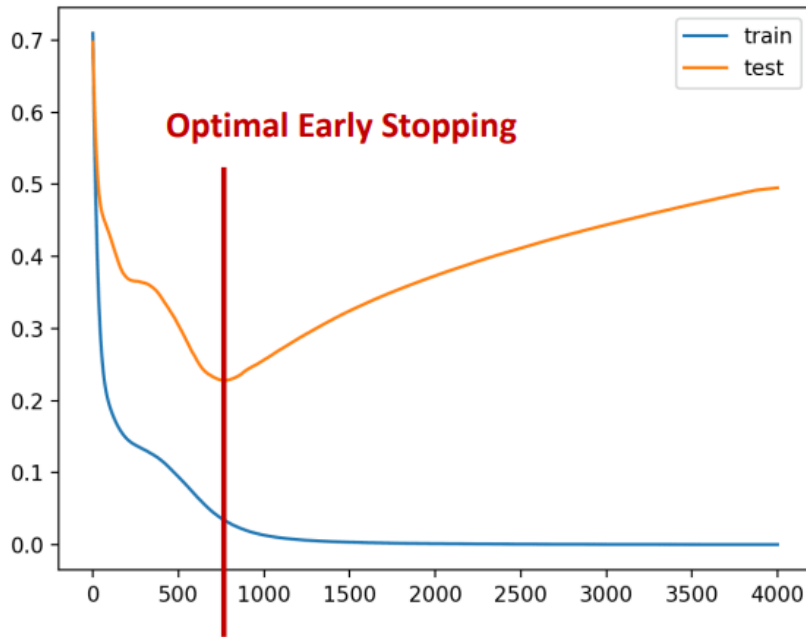
Σχήμα 3.8: Overfitting - επιθυμητή προσαρμογή - Underfitting

Dropout : Μια τεχνική με την οποία μπορεί να αντιμετωπιστεί το **overfitting** , είναι το **dropout** [16] , η οποία είναι μια τεχνική **regularization**. Γενικά, τέτοιου είδους τεχνικές βρίσκουν ευρεία χρήση, καθώς μπορούν να αντιμετωπίσουν πολύ αποτελεσματικά το συγκεκριμένο πρόβλημα. Πρόκειται για μια τυχαία απόρριψη νευρώνων του συνολικού νευρωνικού δικτύου, με σκοπό τη μείωση του μεγέθους των δεδομένων εκπαίδευσης, καθώς η χρήση ενός πολύ μεγάλου μεγέθους συνήθως δημιουργεί το εν λόγω πρόβλημα. Το ποσοστό νευρώνων που απορρίπτεται επιλέγεται πριν από την έναρξη της εκπαίδευσης, αλλά υπάρχουν και διάφορες τεχνικές που αποφασίζει μόνο του το μοντέλο πόσους νευρώνες πρέπει να κρατήσει ή να απορρίψει. Η επιλογή των νευρώνων που θα απορριφθούν προκύπτει από μια τυχαία μεταβλητή που ακολουθεί την κατανομή **Bernoulli** και εφαρμόζεται σε κάθε **layer** ξεχωριστά.



Σχήμα 3.9: α) Ολόκληρο νευρωνικό δίκτυο, β) Νευρωνικό δίκτυο με **dropout**

Early stopping : Μια ακόμη τεχνική για την αντιμετώπιση του **overfitting** είναι το **early stopping**. Στενάζεται στη χρήση ενός **validation dataset** κατά τη διάρκεια της εκπαίδευσης ελέγχοντας την τιμή της συνάρτησης κόστους του σε συνάρτηση με τον αριθμό των εποχών εκπαίδευσης. Ο συνηθέστερος τρόπος με τον οποίο χρησιμοποιείται είναι μέσω του εντοπισμού (**monitor**) της χαμηλότερης τιμής του, έχοντας επιλέξει ένα διάστημα υπομονής (**patience**). Με το πέρας του διαστήματος αυτού αν δεν έχει βρεθεί κάποια χαμηλότερη τιμή του σφάλματος του **validation dataset** , ολοκληρώνεται η διαδικασία της εκπαίδευσης πριν την άφιξη του επιλεγμένου εξ αρχής αριθμού εποχών. Με αυτόν τον τρόπο, επιτυγχάνεται η επιλογή του κατάλληλου αριθμού εποχών, έτσι ώστε το νευρωνικό δίκτυο να δίνει τις πιο αποτελεσματικές προβλέψεις σε άγνωστα δεδομένα και να γενικεύει καλύτερα. Αν επιλεγεί η τεχνική αυτή, προτείνεται ο αριθμός των μέγιστων εποχών να είναι μεγάλος.



Σχήμα 3.10: Παράδειγμα early stopping

Στην προηγούμενη ενότητα, στην προσπάθεια περιγραφής των βημάτων της εκπαίδευσης ενός FFNN, έγινε αναφορά σε κάποιες έννοιες, σχετικές με το βαθμό εκμάθησης, την επιλογή των ενακτήριων τιμών των **weights** και **biases**, τους **optimizers** και τις συναρτήσεις κόστους. Προσθέτοντας μια ακόμη υπερπαραμέτρο - λειτουργία που μπορεί να χρησιμοποιηθεί στη διαδικασία της εκπαίδευσης, στη συνέχεια θα γίνει μια προσπάθεια ανάλυσης των εννοιών αυτών.

Learning rate : Ο βαθμός εκμάθησης αποτελεί μια από τις σημαντικότερες υπερπαραμέτρους ενός νευρωνικού δικτύου, καθώς καθορίζει τον τρόπο με τον οποίο θα γίνεται η ανανέωση των τιμών των **weights** και **biases**. Η επιλογή ενός άστοχου **learning rate** οδηγεί στο να μην μπορεί να 'μαθαίνει' το νευρωνικό δίκτυο από τα δεδομένα κατά τη διαδικασία της εκπαίδευσης. Πολλές φορές για την επιλογή ενός **learning rate** που θα ταιριάζει σε ένα συγκεκριμένο πρόβλημα εφαρμόζεται η τεχνική του **learning rate scheduling**, κατά την οποία οι τιμές του μεταβάλλονται μέσα σε ένα εύρος τιμών, με αρχικά πολύ μικρές τιμές και σταδιακά μεγαλύτερες εντοπίζοντας το σημείο, όπου το σφάλμα εκπαίδευσης θα αρχίσει να εμφανίζει αύξηση τιμών μετά την αρχική μείωση με το πέρασμα των εποχών. Η πιο συνηθισμένη τιμή του είναι 10^{-3} , αλλά η επιλογή του εξαρτάται πάντα από την αρχιτεκτονική του νευρωνικού δικτύου, τον τύπο των δεδομένων και την κατηγορία της εφαρμογής.

Glorot uniform (Xavier initialization) : Το forward pass είναι μια διαδικασία που πραγματοποιείται από όλους τους τύπους νευρωνικών δικτύων και όχι μόνο από αυτό που έχει αναφερθεί μέχρι αυτό το σημείο. Έτσι, όπως προαναφέρθηκε, για την

επίτευξη του πρώτου **forward pass** χρησιμοποιούνται τυχαίες τιμές στα **weights** και τα **biases**, στα οποία, βέβαια, πρέπει να επιλέγονται τιμές που δεν θα αποκλείουν σε μεγάλο βαθμό από τις βέλτιστες. Για αυτό το λόγο, χρησιμοποιούνται διάφοροι αλγόριθμοι **weight initialization**. Ένας από τους πιο συχνά χρησιμοποιούμενους είναι ο **Glorot uniform** ή **Xavier initialization** [17]. Κατά τον αλγόριθμο αυτό οι ενακτήριες τιμές που επιλέγονται για τα **biases** είναι μηδενικές, ενώ οι τιμές ακολουθούν την ομοιόμορφη κατανομή τα παρακάτω άκρα:

$$W \sim U \left(-\sqrt{\frac{6}{n_j + n_{j+1}}}, \sqrt{\frac{6}{n_j + n_{j+1}}} \right) \quad (3.9)$$

όπου n_j είναι ο αριθμός των **inputs** σε ένα **layer** και n_{j+1} ο αριθμός των **outputs** από αυτό.

Adam optimizer : Στην προηγούμενη ενότητα, κατά την περιγραφή του βήματος ανανέωσης τιμών των παραμέτρων ενός **FFNN**, έγινε αναφορά στον αλγόριθμο βελτιστοποίησης **Gradient Descent**, ο οποίος είναι από τους πιο απλούς και διαδεδομένους αλγορίθμους σε προβλήματα μηχανικής μάθησης. Ο εν λόγω αλγόριθμος αποτέλεσε τη βάση για τη δημιουργία νέων, με την ιδιότητα της πιο γρήγορης στη βέλτιστη λύση σύγκλισης. Ένας από αυτούς είναι και ο **Adam optimizer** [18]. Ο αλγόριθμος αυτός, βασίζεται στον υπολογισμό δύο ποσοτήτων, των λεγόμενων **moments**, οι οποίες περιέχουν τη σχετική με τις μερικές παραγώγους της συνάρτησης κόστους πληροφορία. Παρακάτω, απεικονίζεται ο ψευδοκώδικας του αλγορίθμου αυτού:

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t .

Require: α : Stepsize

Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates

Require: $f(\theta)$: Stochastic objective function with parameters θ

Require: θ_0 : Initial parameter vector

$m_0 \leftarrow 0$ (Initialize 1st moment vector)

$v_0 \leftarrow 0$ (Initialize 2nd moment vector)

$t \leftarrow 0$ (Initialize timestep)

while θ_t not converged **do**

$t \leftarrow t + 1$

$g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)

$m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)

$v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)

$\hat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)

$\hat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)

$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon)$ (Update parameters)

end while

return θ_t (Resulting parameters)

Σχήμα 3.11: Ψευδοκώδικας του **adam optimizer** από το πρωτότυπο **paper**

Αρχικά, στην πρώτη του επανάληψη, στο μηδενικό βήμα, οι ποσότητες m_t και v_t τίθενται ίσες με το μηδέν. Επιπρόσθετα, μέσω του **backward pass** έχουν υπολογιστεί οι μερικές παράγωγοι της συνάρτησης κόστους, συναρτήσει των **weights** και **biases**, όπου συμβολίζονται με θ . Έπειτα, μέσω των παρακάτω τύπων υπολογίζονται οι ποσότητες των m_t και v_t , οι οποίες είναι τα διανύσματα του 1ου και 2ου **moment** αντίστοιχα:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (3.10)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (3.11)$$

όπου β_1 και β_2 , οι υπερπαράμετροι που ελέγχουν τον εκθετικό ρυθμό μείωσης των m_t και v_t με τιμές $[0, 1)$ και g_t οι μερικές παράγωγοι της συνάρτησης κόστους συναρτήσει των παραμέτρων. Στη συνέχεια, υπολογίζονται οι διορθωμένες τιμές των **moments**, ως:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (3.12)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (3.13)$$

Αφού υπολογιστούν οι ποσότητες αυτές, πραγματοποιείται, τελικά, η ανανέωση των τιμών των παραμέτρων θ ως:

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t \quad (3.14)$$

όπου α ο ρυθμός εκμάθησης και ϵ μια πολύ μικρή τιμή για την αποφυγή της διαίρεσης με το μηδέν.

Ως t προσδιορίζεται το βήμα του αλγορίθμου. Οι υπερπαράμετροι $\{\beta_1, \beta_2, \alpha, \epsilon\}$ συνήθως επιλέγεται να έχουν τις παρακάτω τιμές $\{0.9, 0.999, 0.001, 10^{-8}\}$. Τέλος, σημειώνεται ότι ο συγκεκριμένος αλγόριθμος, όπως και κάθε αλγόριθμος βελτιστοποίησης δεν ταιριάζει με όλες τις εφαρμογές και η επιλογή του εξαρτάται από τους ίδιους παράγοντες με τις προηγούμενες περιπτώσεις.

Batch training : Λόγω της χρήσης ενός πολύ μεγάλου αριθμού δεδομένων κατά τη διαδικασία της εκπαίδευσης, πολύ συχνά επιλέγεται η εκπαίδευση σε παρτίδες (**batch training** ή **mini-batch training**). Σε αυτή την περίπτωση, το σύνολο δεδομένων χωρίζεται σε παρτίδες, ενός αριθμού που επιλέγεται, από τις οποίες προκύπτουν ξεχωριστοί υπολογισμοί, δηλαδή η μια παρτίδα ενός αριθμού δεδομένων μπορεί να θεωρηθεί ως ένα δεδομένο εισόδου. Πιο συγκεκριμένα, τα δεδομένα που περιέχονται σε μια παρτίδα χρησιμοποιούνται για τον υπολογισμό των συναρτήσεων κόστους και των μερικών παραγώγων, από τις οποίες υπολογίζεται ο μέσος όρος και χρησιμοποιείται στη διαδικασία ανανέωσης των παραμέτρων. Είναι μια διαδικασία, η οποία εκτός από υπο-

λογιστικό κέρδος, ενισχύει την ικανότητα γενίκευσης στις προβλέψεις του νευρωνικού δικτύου, δηλαδή τη μεγαλύτερη αποτελεσματικότητα σε άγνωστα για αυτό δεδομένα. Η επιλογή του αριθμού παρτίδων που θα επιλέγεται κάθε φορά, καθορίζεται κυρίως από τον αριθμό των προς εκπαίδευση δεδομένων, αλλά και άλλους παράγοντες όπως η αρχιτεκτονική του νευρωνικού δικτύου και η φύση των δεδομένων. Μια λανθασμένη επιλογή αριθμού παρτίδας δύναται να οδηγήσει είτε σε **overfitting** είτε σε **underfitting**.

3.3 Ανατροφοδοτούμενα νευρωνικά δίκτυα (Recurrent Neural Networks - RNNs)

Ένας άλλος αξιοσημείωτος τύπος νευρωνικών δικτύων και μάλιστα ιδιαίτερα χρήσιμος στην πρόβλεψη χρονοσειρών είναι τα ανατροφοδοτούμενα νευρωνικά δίκτυα (RNNs) και οι διάφορες παραλλαγές τους. Οι πρώτες έρευνες σχετικά με τη συγκεκριμένη αρχιτεκτονική έγιναν αρχικά από τον Michael I. Jordan [19] και σε συνέχεια της έρευνας του πρώτου από τον Jeffrey L. Elman [20] το 1986 και το 1990 αντίστοιχα και έθεσαν τα θεμέλια για την έναρξη της ανάπτυξης της. Η βασική διαφορά με τα FFNNs είναι στην ικανότητα τους να χρησιμοποιούν συνδέσεις ανατροφοδότησης (**feedback connections**) που είναι ιδιαίτερα χρήσιμες για την εκμάθηση μοτίβων από δεδομένα και για την επεξεργασία τους μέσα από τα **hidden layers** που βρίσκονται σε μορφή χρονικής ακολουθίας (**sequential data**). Τέτοια δεδομένα είναι οι χρονοσειρές, κείμενα, ήχος, ακόμα και εικόνες, αν αναλυθούν σε μορφή ακολουθίας και οι κυριότερες εφαρμογές στις οποίες χρησιμοποιούνται είναι η πρόβλεψη χρονοσειρών, η αναγνώριση φωνής, η δημιουργία κειμένου, η αυτόματη μετάφραση και πολλές άλλες ακόμα.

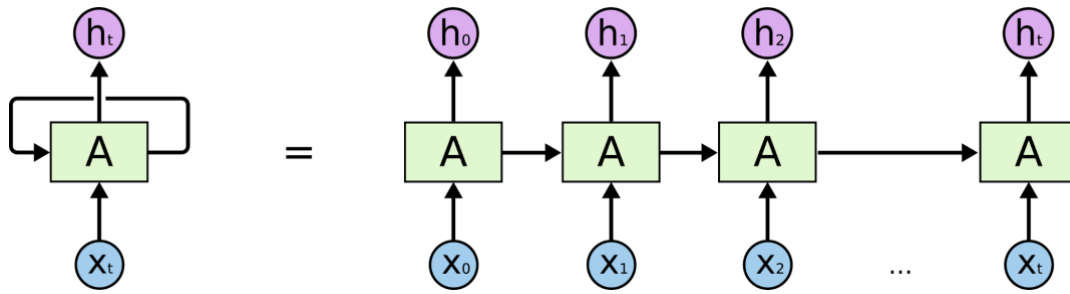
Οι συνδέσεις ανατροφοδότησης περιέχονται στα **hidden layers** και δίνουν, στην ουσία, τη δυνατότητα στο νευρωνικό δίκτυο να έχει μνήμη και να μπορεί να αποθηκεύει πληροφορία από προηγούμενα χρονικά βήματα. Επιπρόσθετα, ένα ακόμη πλεονέκτημα που έχουν σε σχέση με τα απλά τεχνητά νευρωνικά δίκτυα είναι ότι, εκτός από την αποθήκευση πληροφορίας από προηγούμενα χρονικά βήματα, είναι αρκετά αποτελεσματικά στην πρόβλεψη μιας ακολουθίας τιμών. Γενικά, υπάρχουν διάφοροι τύποι προβλέψεων που μπορούν να πραγματοποιηθούν με το συγκεκριμένο τύπο νευρωνικών δικτύων, όπως:

- Ένα προς ένα (**One-to-one**): Δέχεται μια τιμή ως **input** και προβλέπει μια τιμή ως **output**.
- Ένα προς πολλά (**One-to-many**): Δέχεται μια τιμή ως **input** και προβλέπει πολλές τιμές ως **outputs**.
- Πολλά προς ένα (**Many-to-one**): Δέχεται πολλές τιμές ως **inputs** και προβλέπει

μια τιμή ως **output**.

- Πολλά προς πολλά (**Many-to-many** ή **sequence-to-sequence**): Δέχεται πολλές τιμές ως **inputs** και προβλέπει πολλές τιμές ως **outputs**.

Παρακάτω (Εικόνα 3.12) απεικονίζεται ένα παράδειγμα RNN καθώς και το πως φαίνεται η δομή του όταν αυτό ‘‘ξεδιπλώνεται’’ στο χρόνο.



Σχήμα 3.12: Δομή ενός RNN

Στην περίπτωση που, για παράδειγμα, το RNN δεχόταν 3 τιμές ως **inputs**, τότε αυτό θα ‘‘ξεδιπλώνοταν’’ 3 φορές πίσω και κάθε τιμή θα είχε αντιστοιχούσε σε ένα **layer**.

3.3.1 Εκπαίδευση ενός RNN

Τα βήματα εκπαίδευσης ενός τέτοιου τύπου νευρωνικών δικτύων, αλλά και των παραλλαγών αυτού, είναι αρκετά όμοια με αυτά που περιγράφηκαν στην ενότητα 3.1. Η μόνη διαφορά σε αυτή την περίπτωση είναι η ανάμειξη του χρονικού παράγοντα που προέρχεται από την φύση των δεδομένων που χρησιμοποιούνται. Επομένως, πρέπει να χρησιμοποιηθεί ένα διαφορετικό είδος αλγορίθμου από τον **backpropagation algorithm**. Ο συχνότερα χρησιμοποιούμενος αλγόριθμος εκπαίδευσης, αποτελεί μια παραλλαγή του προηγούμενου και ονομάζεται **backpropagation through time (BPTT)** [21], κατά τον οποίο πραγματοποιείται ουσιαστικά το ‘‘ξεδίπλωμα’’ των **hidden layers**, κατασκευάζοντας ένα παραδοσιακό FFNN, με σκοπό να εφαρμοστεί ο αλγόριθμος **backpropagation** [22]. Τα βήματα που πραγματοποιούνται σε αυτόν τον αλγόριθμο είναι κοινά με τον προηγούμενο, με διαφοροποιήσεις στο κομμάτι των εξισώσεων που τον χαρακτηρίζουν. Πιο αναλυτικά:

1. Forward pass : Όμοια, πρώτο βήμα του αλγορίθμου με τη διαφορά ότι η πληροφορία από τα **inputs** επεξεργάζεται με διαφορετικό τρόπο, λόγω της διαφορετικής αρχιτεκτονικής του νευρωνικού δικτύου. Αυτή η περίπτωση περιγράφεται από τις παρακάτω εξισώσεις, σε μορφή πινάκων, για ένα χρονικό βήμα t :

$$H_t = f_h(X_t W_{xh} + H_{t-1} W_{hh} + b_h) \quad (3.15)$$

$$O_t = f_o(H_t W_{ho} + b_o) \quad (3.16)$$

όπου $H_t \in \mathbb{R}^{n \times h}$, ο πίνακας των τιμών του **hidden state** και $X_t \in \mathbb{R}^{n \times d}$, ο πίνακας των **inputs** με n το συνολικό αριθμό των δεδομένων που περιέχονται στο **training dataset** d τον αριθμό των **inputs** (τα λεγόμενα **timesteps** ή **lookback period**) και h τον αριθμό των νευρώνων που περιέχονται στο **hidden layer** (τώρα **hidden state**). Επίσης, $W_{xh} \in \mathbb{R}^{d \times h}$, ο πίνακας των **weights** από τα **inputs** στο **hidden state**, $W_{hh} \in \mathbb{R}^{h \times h}$, ο πίνακας των **weights** από το **hidden state** στο επόμενο **hidden state**, $W_{ho} \in \mathbb{R}^{h \times o}$ ο πίνακας των **weights** από το **hidden state** στα **outputs** με o , τον αριθμό των **outputs** και $b_h \in \mathbb{R}^{1 \times h}$ και $b_o \in \mathbb{R}^{1 \times o}$, τα διανύσματα των **biases** στο **hidden state** και **output layer** αντίστοιχα.

Μέσω των παραπάνω εξισώσεων, γίνεται αντιληπτός ο τρόπος με τον οποίο αποθηκεύεται η πληροφορία, μέσω του H_{t-1} , πίνακας ο οποίος περιέχει τις τιμές του **hidden state** του προηγούμενου χρονικού βήματος. Σημειώνεται ότι οι παραπάνω εξισώσεις ισχύουν για την περίπτωση ενός κρυφού στρώματος, ενώ αν υπάρχουν περισσότερα το **output** του ενός λειτουργεί ως **input** για το επόμενο.

2. Cost function : Για τη συνάρτηση κόστους ισχύει ότι ακριβώς αναφέρθηκε στην ενότητα 3.1 με τη μόνη διαφοροποίηση στον υπολογισμό της, καθώς γίνεται με βάση το χρόνο. Η συνάρτηση κόστους, χωρίς όρους **regularization** δίνεται από:

$$C(\theta) = \frac{1}{T} \sum_{t=1}^T (y_t - \hat{y}_t)^2 \quad (3.17)$$

όπου T ο συνολικός αριθμός των **timesteps**, y_t η πραγματική τιμή στο **timestep** t και $\hat{y}_t$ η προβλεπόμενη τιμή στο **timestep** t .

3. Backward pass : Βήμα, στο οποίο πρέπει να υπολογιστούν οι μερικές παράγωγοι της συνάρτησης κόστους, συναρτήσει των τριών διαφορετικών **weights** και των δύο **biases**. Στην προκειμένη, οι παραπάνω υπολογισμοί πραγματοποιούνται για κάθε χρονικό βήμα ξεχωριστά και στο τέλος αθροίζονται όλες οι ποσότητες για την κάθε παράγωγο ξεχωριστά. Για τον υπολογισμό των παραγώγων αυτών χρησιμοποιείται ο κανόνας της αλυσίδας, όπως και στην προηγούμενη περίπτωση. Στο σημείο αυτό του αλγορίθμου συνήθως εμφανίζεται ένα μεγάλο πρόβλημα, το οποίο θα αναλυθεί στην επόμενη ενότητα.

4. Weights and biases update : Τελευταίο βήμα του αλγορίθμου, στο οποίο χρησιμοποιείται κάποιος αλγόριθμος βελτιστοποίησης από αυτούς που έχουν ήδη αναφερθεί (**Gradient Descent**, **Adam**) ή και κάποιους άλλους, για την ανανέωση των παραμέτρων του **RNN**. Σε αυτό το βήμα ισχύει, ακριβώς, ότι αναφέρθηκε και στην ενότητα 3.1

Τα παραπάνω βήματα αποτελούν μέρος μιας επαναληπτικής διαδικασίας. Όλες οι αναφορές που έγιναν στις ενότητες 3.1 και 3.2 ισχύουν και για την περίπτωση της εκπαίδευσης των ανατροφοδοτούμενων νευρωνικών δικτύων και των παραλλαγών τους.

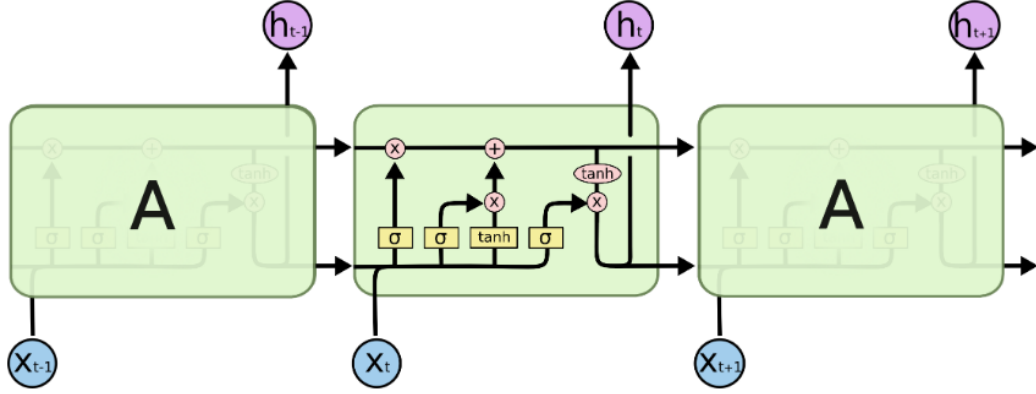
3.3.2 Το πρόβλημα της εκτόξευσης/ελάττωσης των μερικών παραγώγων

Τα RNNs, όπως αναφέρθηκε, έχουν την ικανότητα να δέχονται έναν αριθμό χρονικά εξαρτημένων διακριτών δεδομένων και να τον αποθηκεύουν μέσα στα **hidden states** τους. Βέβαια, όσο πιο μεγάλος είναι ο αριθμός αυτών, δηλαδή το μήκος του χρονικού βήματος των **inputs**, εμφανίζεται και μεγαλύτερη δυσκολία στο να επιτευχθεί η αποθήκευση αυτή. Αυτό το πρόβλημα εμφανίζεται κατά τον υπολογισμό των μερικών παραγώγων στον αλγόριθμο **BPTT** και χαρακτηρίζεται ως το πρόβλημα της εκτόξευσης/ελάττωσης των μερικών παραγώγων (**exploding/vanishing gradient problem**) [23]. Οι μερικές παράγωγοι, που υπολογίζονται με βάση των κανόνα της αλυσίδας, περιέχουν πληροφορία από τα **hidden states**, η οποία σε περίπτωση μεγάλου μήκους **inputs** οδηγεί στον πολλαπλασιασμό πολλών ποσοτήτων μεταξύ τους. Σε περίπτωση που κάποιες τιμές των ποσοτήτων αυτών, οι οποίες αποτελούν στοιχεία πινάκων, λάβουν μεγάλες τιμές (> 1) ή μικρές τιμές (< 1) τότε πραγματοποιείται η εκτόξευση ή ελάττωση των τιμών αντίστοιχα. Επομένως, σταματάει και η συνεισφορά από αυτά και τα μετέπειτα χρονικά βήματα. Το συγκεκριμένο πρόβλημα μπορεί να παρατηρηθεί και στην περίπτωση του **FFNN**, αν χρησιμοποιηθούν πολλά **hidden layers** ή ακόμα και αν γίνει ανεπαρκής έναρξη τιμών των **weights**. Η ανακάλυψη νέων αρχιτεκτονικών **RNN** (παραλλαγών τους) έχει οδηγήσει στην αντιμετώπιση του εν λόγω προβλήματος, έχοντας την ικανότητα απομνημόνευσης μεγαλύτερου μήκους χρονικά βήματα. Αυτού του είδους **RNNs** είναι τα **LSTMs**, **GRUs** και **Bidirectional RNNs**, τα οποία θα περιγραφούν στη συνέχεια.

3.4 Δίκτυα μακράς βραχύχρονης μνήμης (LSTM)

Η αδυναμία εκμάθησης δεδομένων από τα συμβατικά ανατροφοδοτούμενα νευρωνικά δίκτυα που περιγράφηκαν στην προηγούμενη ενότητα, οδήγησε στην αναζήτηση νέων τεχνικών εκμάθησης, αλλά και αρχιτεκτονικών νευρωνικών δικτύων. Μια από αυτές είναι και τα δίκτυα μακράς βραχύχρονης μνήμης (**Long Short-Term Memory networks - LSTM**), τα οποία συστήθηκαν αρχικά από τους **Sepp Hochreiter** και **Jurgen Schmidhuber** [24] το 1997 και έπειτα εξελίχθηκαν μέχρι να πάρουν και τη μοντέρνα μορφή τους. Ο συγκεκριμένος τύπος νευρωνικού δικτύου αποτελεί μια παραλλαγή του συμβατικού **RNN**, στην οποία πραγματοποιήθηκε η αντικατάσταση του **hidden state** με ένα στοιχείο μνήμης (**memory cell**). Το συγκεκριμένο στοιχείο είναι και αυτό που του προσδίδει τη δυνατότητα αποθήκευσης της πληροφορίας που θεωρεί σημαντική και χρήσιμη από τα προηγούμενα χρονικά βήματα. Γενικά, ένα στοιχείο **LSTM (LSTM cell)** αποτελείται

από διάφορους νευρώνες, οι οποίοι συνδυάζονται μεταξύ τους, έτσι ώστε να πραγματοποιηθεί η εν λόγω αποθήκευση, αλλά και το **forward pass**, για τη διαδικασία της εκπαίδευσης. Στην εικόνα 3.13 παρατηρείται η δομή ενός δικτύου LSTM.



Σχήμα 3.13: Δομή ενός LSTM

Έστω x_t ένα διάνυσμα των **inputs** στο χρονικό βήμα t , h_{t-1} το διάνυσμα των **outputs** από το προηγούμενο στοιχείο (**hidden output**) και s_{t-1} το διάνυσμα που περιέχει την αποθηκευμένη πληροφορία από το προηγούμενο στοιχείο. Τα πρώτα δύο συνδυάζονται για τον υπολογισμό του διανύσματος του **forget gate**, μέσω του οποίου αποφασίζεται ποια πληροφορία είναι σημαντική για αποθήκευση, "προβάλλοντας" στο s_{t-1} τιμές από 0 έως 1, με εφαρμογή της **sigmoid activation function**, με το 0 να σημαίνει ότι η πληροφορία δεν είναι σημαντική και το 1 να σημαίνει ότι πρέπει οπωσδήποτε να αποθηκευτεί. Αυτό περιγράφεται μέσω της εξίσωσης:

$$f_t = \sigma(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \quad (3.18)$$

Έπειτα, μέσω των i_t και g_t , τα οποία είναι διανύσματα που σχετίζονται με τη νέα πληροφορία, αλλά και τη χρήση του **forget gate**, αποφασίζεται ποια πληροφορία θα οδηγηθεί στο **memory cell**, μέσω των τύπων:

$$i_t = \sigma(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \quad (3.19)$$

$$g_t = \phi(W_{gx}x_t + W_{gh}h_{t-1} + b_g) \quad (3.20)$$

$$s_t = g_t \odot i_t + s_{t-1} \odot f_t \quad (3.21)$$

όπου ϕ η συνάρτηση ενεργοποίησης **tanh** και $\odot$ ο πολλαπλασιασμός ανά στοιχείο (**element-wise multiplication**). Τέλος, υπολογίζεται το **output** h_t του **cell**, το οποίο

μεταβιβάζεται στο επόμενο χρονικό βήμα, ως:

$$\begin{aligned} \mathbf{o}_t &= \sigma(W_{ox}\mathbf{x}_t + W_{oh}\mathbf{h}_{t-1} + \mathbf{b}_o) \\ \mathbf{h}_t &= \phi(\mathbf{s}_t) \odot \mathbf{o}_t \end{aligned} \quad (3.22)$$

όπου ϕ , ξανά, η συνάρτηση ενεργοποίησης $\tanh$. Τα $W_{ix}, W_{ih}, W_{gx}, W_{gh}, W_{ox}, W_{oh}$ είναι οι πίνακες των **weights** και τα b_f, b_i, b_g, b_o τα διανύσματα των **biases**, σε κάθε gate [25].

Τα παραπάνω αποτελούν το **forward pass** του **LSTM** για ένα τυχαίο χρονικό βήμα, του οποίου η πληροφορία θα χρησιμοποιηθεί, μέσω του αλγορίθμου **BPTT**, για την εκπαίδευσή του. Για την εφαρμογή του αλγορίθμου αυτού το **LSTM** πραγματοποιεί ‘ξεδίπλωση’ των **states** του, όπως και στα συμβατικά **RNNs**.

3.5 Δίκτυα Gated Recurrent Unit (GRU)

Μια ακόμα, μοντέρνα, παραλλαγή του συμβατικού **RNN**, η οποία δημιουργήθηκε για την αντιμετώπιση του προβλήματος της ελάττωσης των τιμών των μερικών παραγώγων, αλλά και για την ικανότητα αποθήκευσης πληροφορίας από μεγάλα χρονικά βήματα, προήλθε από τον **Kyunghyun Cho** το 2014 και είναι το δίκτυο **Gated Recurrent Unit (GRU)** [26]. Πρόκειται, για μια αρχιτεκτονική εμπνευσμένη από το **LSTM**, κατά την οποία δύο θύρες (**gates**) παίζουν το ρόλο του στοιχείου μνήμης, η θύρα επαναφοράς (**reset gate**) και η θύρα ανανέωσης (**update gate**). Οι εξισώσεις που είναι υπεύθυνες για το **forward pass**, σε ένα τυχαίο χρονικό βήμα t , είναι:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (3.23)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (3.24)$$

$$\hat{h}_t = \phi(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (3.25)$$

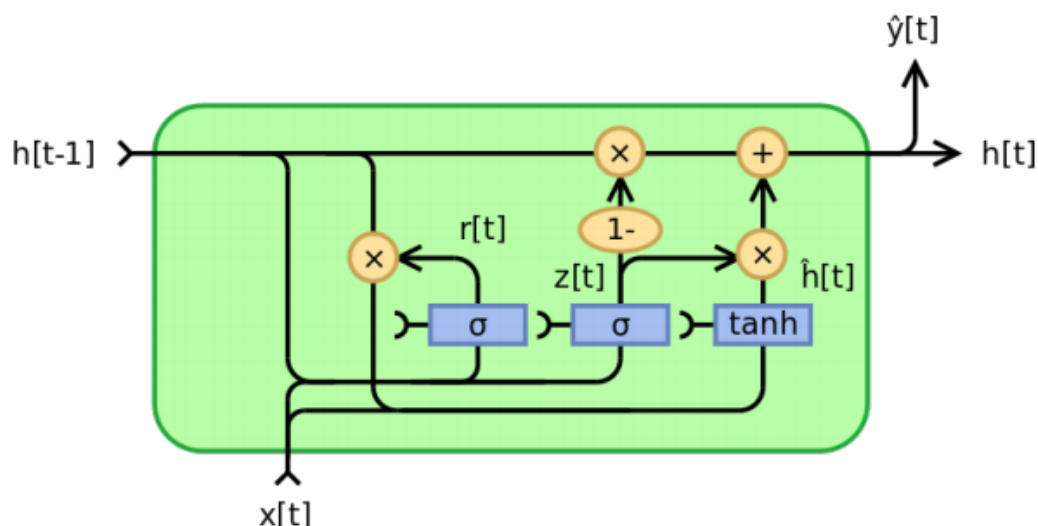
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \hat{h}_t \quad (3.26)$$

όπου z_t το διάνυσμα **update gate**, το r_t το διάνυσμα **reset gate**, το διάνυσμα $\hat{h}_t$, το νέο **hidden state**, το οποίο περιέχει την πληροφορία από τα νέα **inputs** και το h_t το διάνυσμα των **outputs**, το οποίο θα είναι και το **input**, του επόμενου χρονικού βήματος. Τα W, U και b είναι οι πίνακες και τα διανύσματα των **weights** και **biases** σε κάθε gate αντίστοιχα [27].

Αρχικά, η πληροφορία η οποία εισέρχεται από το προηγούμενο **hidden state (time step)** ελέγχεται από το **reset gate**, το οποίο επιλέγει ποια πρέπει να απορριφθεί και ποια πρέπει να περάσει στο επόμενο **hidden state**, δίνοντας τιμές στο $(0,1)$, με τη 0 να οδηγεί στην

απόρριψη. Έπειτα, υπολογίζεται το νέο **hidden state** $\hat{h}_t$, στο οποίο περιέχεται η νέα πληροφορία από τα **inputs**, αλλά και αυτή που είναι αποδεκτή από το **reset gate**. Σε αυτήν την περίπτωση, χρησιμοποιείται η συνάρτηση ενεργοποίησης *tanh*. Στη συνέχεια, μέσω του **update gate** επιλέγεται η ποσότητα της πληροφορίας που πρέπει να περάσει από το προηγούμενο **hidden state** στο νέο και τέλος υπολογίζεται το **output** της μονάδας, μέσω της πληροφορίας από τα **update gate** και νέο **hidden state**.

Ο συγκεκριμένος τύπος δικτύου, συγκριτικά με το **LSTM**, έχει μια πιο απλή δομή, γεγονός που οδηγεί σε μια εκπαίδευση με λιγότερο υπολογιστικό κόστος. Βέβαια, η απλότητα της δομής του αποτελεί έναν περιορισμό για την εκμάθηση πιο περίπλοκων μοτίβων μεταξύ των δεδομένων. Άρα, είναι μια πολύ αξιόπιστη και γρήγορη λύση για πιο απλούς τύπους δεδομένων. Για τη διαδικασία της εκπαίδευσης ακολουθούνται τα ίδια βήματα με τους προηγούμενους τύπους ανατροφοδοτούμενων νευρωνικών δικτύων.

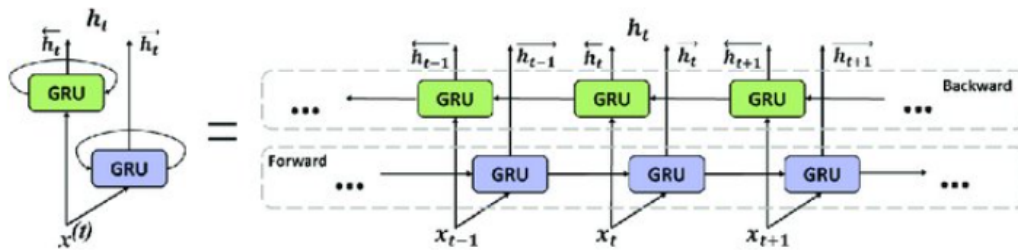


Σχήμα 3.14: Δομή ενός GRU

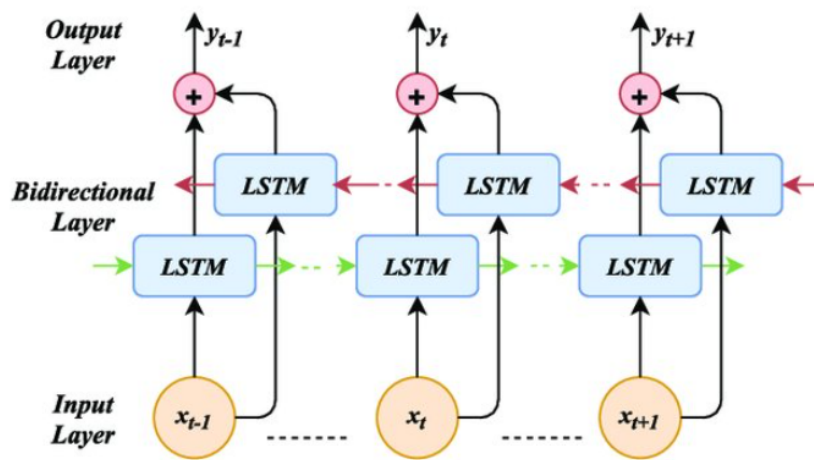
3.6 Ανατροφοδοτούμενα νευρωνικά δίκτυα δύο κατευθύνσεων (Bidirectional RNNs)

Ένας ακόμα τρόπος για την αντιμετώπιση των περιορισμών που προκύπτουν κατά την εκπαίδευση των συμβατικών RNNs είναι με τη χρήση των ανατροφοδοτούμενων νευρωνικών δικτύων δύο κατευθύνσεων είτε συμβατικών είτε δικτύων που περιέχουν **LSTM** και **GRU cells**. Ο συγκεκριμένος τύπος δικτύου δημιουργήθηκε από την έρευνα των **Mike Schuster** και **Kuldip K. Paliwal** [28] το 1997. Στις αρχιτεκτονικές που έχουν περιγραφεί έως αυτό το σημείο, χρησιμοποιούνταν ως **inputs** δεδομένα ακολουθίας της μορφής $X_t = \{X_1, X_2, \dots, X_T\}$, όπου T το μήκος του χρονικού βήματος, των οποίων η πληροφορία έρρεε από την πρώτη χρονική στιγμή έως την τελευταία μέσα από τα **hidden**

states του κάθε ανατροφοδοτούμενου νευρωνικού δικτύου. Όμως, στη συγκεκριμένη περίπτωση η πληροφορία από αυτά κινείται ως προς δύο κατευθύνσεις, μια προς τα εμπρός και μια προς τα πίσω. Ο τρόπος με τον οποίο αυτό επιτυγχάνεται, είναι μέσω της ύπαρξης δύο **hidden state cells** από τα οποία το ένα συνδέεται με αυτό του επόμενου χρονικού βήματος και ένα με αυτό του προηγούμενου χρονικού βήματος. Στις παρακάτω εικόνες (3.15, 3.16) απεικονίζεται αυτό ακριβώς που περιγράφηκε παραπάνω για **LSTM** και **GRU cells**, τα οποία είναι και αυτά που χρησιμοποιούνται στην παρούσα διπλωματική εργασία.



Σχήμα 3.15: Δομή ενός Bidirectional GRU



Σχήμα 3.16: Δομή ενός Bidirectional LSTM

Για τη διαδικασία εκπαίδευσης χρησιμοποιείται ο ίδιος ακριβώς αλγόριθμος (BPTT), κατά τον οποίο το νευρωνικό δίκτυο αυτό “ξεδιπλώνεται” έτσι ώστε να αποκτήσει τη μορφή του FFNN. Πιο συγκεκριμένα, αρχικά, εκτελείται ένα **forward pass** από $t = 1$ έως $t = T$ για την προς τα εμπρός κατεύθυνση και από $t = T$ έως $t = 1$ για την πίσω κατεύθυνση, αποθηκεύοντας κάποια πληροφορία τόσο στο **hidden state** της μιας κατεύθυνσης όσο και της άλλης. Έπειτα, για τον υπολογισμό του **output** πραγματοποιείται μια συγχώνευση στα **hidden states** των δύο κατευθύνσεων, σε ένα το οποίο θα περιέχει τις πληροφορίες και των δύο. Έχοντας τα **outputs** υπολογίζεται η συνάρτηση

κόστους και στη συνέχεια πραγματοποιείται το **backward pass** χρησιμοποιώντας τη συγχωνευμένη πληροφορία των 2 κατευθύνσεων. Μια άλλη προσέγγιση είναι αυτή που αναγράφεται στο πρωτότυπο **paper** (Εικόνα 3.17), κατά την οποία η διαδικασία εκπαίδευσης πραγματοποιείται διαχωρίζοντας εξόλοκληρου τις δύο κατευθύνσεις.

Ο συγκεκριμένος τύπος δικτύου εκτός από την ικανότητα εκμάθησης μεγάλου μήκους χρονικά βήματα, πλεονεχτεί σε εφαρμογές όπου τα δεδομένα εξαρτώνται και από τις δύο κατευθύνσεις, όπως η μετάφραση, η αναγνώριση φωνής και η ανάλυση κειμένου. Έπειτα, προσδίδει τη δυνατότητα για εντοπισμό πολύ περίπλοκων μοτίβων μεταξύ των δεδομένων, καθώς περιέχονται πιο πολλές μεταβλητές και παράμετροι. Βέβαια, λόγω του ιδιαίτερου αυτού τρόπου εκπαίδευσης του, αυξάνεται αρκετά το υπολογιστικό κόστος, σε σχέση με τα δίκτυα μιας κατεύθυνσης

1) FORWARD PASS

Run all input data for one time slice $1 \leq t \leq T$ through the BRNN and determine all predicted outputs.

- a) Do forward pass just for forward states (from $t = 1$ to $t = T$) and backward states (from $t = T$ to $t = 1$).
- b) Do forward pass for output neurons.

2) BACKWARD PASS

Calculate the part of the objective function derivative for the time slice $1 \leq t \leq T$ used in the forward pass.

- a) Do backward pass for output neurons.
- b) Do backward pass just for forward states (from $t = T$ to $t = 1$) and backward states (from $t = 1$ to $t = T$).

3) UPDATE WEIGHTS

Σχήμα 3.17: Περιγραφή της διαδικασίας εκπαίδευσης των **Bidirectional RNNs** από το πρωτότυπο **paper**

3.7 Μετρητές αποτίμησης προβλέψεων (Evaluation metrics)

Κατά τη δημιουργία προβλεπτικών μοντέλων μηχανικής και βαθιάς μάθησης, μετά το πέρας της διαδικασίας της εκπαίδευσης, ένα πολύ σημαντικό βήμα είναι ο έλεγχος της αποτελεσματικότητας του μοντέλου. Η διαδικασία αυτή γίνεται με τη χρήση ενός δεύτερου σετ δεδομένων, το οποίο περιέχει άγνωστες για το εκπαιδευμένο μοντέλο ποσότητες, το **test dataset**. Ο τρόπος με τον οποίο πραγματοποιείται αυτό, είναι μέσω της σύγκρισης των προβλεπόμενων από το μοντέλο τιμών με τις πραγματικές, οι οποίες περιέχονται στο σετ που προαναφέρθηκε, χρησιμοποιώντας διάφορους μετρητές απο-

τίμησης, με την επιλογή τους να γίνεται ανάλογα με την εκάστοτε εφαρμογή. Για ένα πρόβλημα πρόβλεψης χρονοσειρών, το οποίο θεωρείται ότι είναι ένα πρόβλημα παλινδρόμησης, οι πιο διαδεδομένοι είναι, για συνολικό αριθμό προβλέψεων n , προβλεπόμενη τιμή $\hat{y}$, πραγματική τιμή y και μέση τιμή των πραγματικών τιμών $\bar{y}$:

RMSE : Το RMSE ή με την πλήρη ονομασία τετραγωνική ρίζα του μέσου τετραγωνικού σφάλματος (Root Mean Squared Error) είναι ένα μέσο αποτίμησης των προβλέψεων σε μια μεγενθυμένη κλίμακα, συγκριτικά με την πραγματική και δίνεται από τον τύπο:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (3.27)$$

MSE : Αντίστοιχο είδος σφάλματος με το παραπάνω είναι και το μέσο τετραγωνικό σφάλμα, όπου δεν χρησιμοποιείται η τετραγωνική ρίζα. Αυτό το είδος σφάλματος, χρησιμοποιείται ως συνάρτηση σφάλματος σε προβλήματα παλινδρόμησης, όπως είναι η προβλέψεις χρονοσειρών. Δίνεται από τον τύπο:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.28)$$

MAE : Το MAE ή με την πλήρη ονομασία μέσο απόλυτο σφάλμα (Mean Absolute Error) είναι ένα μέσο αποτίμησης των προβλέψεων σε μια πραγματική κλίμακα, καθώς πρόκειται για τη μέση διαφορά τους με την πραγματική τιμή σε απόλυτη τιμή και δίνεται από τον τύπο:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.29)$$

MAPE : Το MAPE ή με την πλήρη ονομασία μέσο ποσοστιαίο απόλυτο σφάλμα (Mean Absolute Percentage Error) είναι ένα μέσο αποτίμησης των προβλέψεων, το οποίο δίνει μια τιμή ποσοστού σφάλματος. Στις περιπτώσεις που οι πραγματικές τιμές είναι πολύ κοντά ή ισούται με το μηδέν, αποφεύγεται η χρήση του, καθώς η τιμή του τείνει προς το άπειρο είτε δεν μπορεί να υπολογιστεί. Η μέγιστη τιμή που μπορεί να πάρει είναι το 100%. Δίνεται από τον τύπο:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \quad (3.30)$$

Για μια αποτελεσματική πρόβλεψη, οι παραπάνω τιμές είναι επιθυμητό να έχουν όσο μικρότερη τιμή γίνεται, με την τιμή μηδέν να αντιστοιχεί στην ιδανική περίπτωση, όπου η προβλέψεις ισοδυναμούν με τις πραγματικές προβλέψεις.

R-squared (R^2) Coefficient : Το μέτρο αποτίμησης **R-squared** , είναι μια ποσότητα που αντικατροπτίζει το πόσο οι προβλέψεις ενός μοντέλου ταιριάζουν με τις πραγματικές τιμές. Ορίζεται ως, μια μέτρηση που αναπαριστά την ποσότητα της διακύμανσης μιας εξαρτημένης μεταβλητής, η οποία μπορεί να προβλεφθεί από ανεξάρτητες μεταβλητές. Δίνεται από τον τύπο:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3.31)$$

4

Περιβάλλον υλοποίησης και δεδομένα

Ενότητες

4.1	Περιβάλλον υλοποίησης	42
4.2	Περιγραφή δεδομένων	43
4.3	Ανάλυση δεδομένων	44

Στο παρόν κεφάλαιο γίνεται αναφορά στο περιβάλλον υλοποίησης των μοντέλων, καθώς επίσης και η παρουσίαση του συνόλου δεδομένων που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία. Περιέχονται πληροφορίες σχετικά με τη γλώσσα και το περιβάλλον προγραμματισμού, με τις βιβλιοθήκες που χρησιμοποιήθηκαν, αλλά και με την προέλευση και τα χαρακτηριστικά των δεδομένων, μέσω της περιγραφής και ανάλυσής τους.

4.1 Περιβάλλον υλοποίησης

Η κατασκευή και εκπαίδευση των μοντέλων των νευρωνικών δικτύων, καθώς και η ανάλυση και η επεξεργασία των δεδομένων πραγματοποιήθηκαν σε περιβάλλον υπολογιστή Acer Aspire A515-45 laptop με AMD Ryzen 7 5700U with Radeon Graphics @1.80 GHz, 8,00 GB RAM και λειτουργικό σύστημα x64 bit Windows 11 Home. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Python 3.9.7 και το περιβάλλον προγραμματισμού το Spyder και τα Jupyter Notebooks του Anaconda 3. Ο χειρισμός των δεδομένων και η κατασκευή των μοντέλων πραγματοποιήθηκε με τη βοήθεια των βιβλιοθηκών που αναφέρονται παρακάτω:

- **Pandas:** Μια από τις πιο διαδεδομένες και χρήσιμες βιβλιοθήκες για το χειρισμό και την ανάγνωση δεδομένων. Χρησιμοποιείται κατά κόρων για την ανάλυση δεδομένων, μέσω της κατάταξης των δεδομένων σε πίνακες για την επεξεργασία, τον καθαρισμό και την διερεύνησή τους, αλλά και της χρήσης των διάφορων στατιστικών εργαλείων που περιέχονται σε αυτή. [29]
- **NumPy:** Μια βιβλιοθήκη, η οποία χρησιμοποιείται ευρέως από τους χρήστες της Python και παρέχει υπηρεσίες αριθμητικών υπολογισμών. Δίνει τη δυνατότητα στο χρήστη να κατασκευάσει και να περιλάβει δεδομένα σε μορφή διανυσμάτων, πινάκων και τανυστών πολλών διαστάσεων, καθώς και να πραγματοποιήσει υπολογισμούς με αυτά. [30]
- **TensorFlow:** Πρόκειται για μια βιβλιοθήκη που παρέχει στον χρήστη δυνατότητες κατασκευής μοντέλων μηχανικής και βαθιάς μάθησης, προεπεξεργασίας των δεδομένων, αλλά και οπτικοποίησης τους και επαλήθευσης των προβλέψεων που προκύπτουν από αυτά. Επιπλέον, λόγω της μεγάλης φορητότητας του, παρέχει τη δυνατότητα εκπαίδευσης των μοντέλων αυτών σε πληθώρα συσκευών και πλατφόρμων. [31]
- **Keras:** Μια βιβλιοθήκη βασισμένη στην TensorFlow για την κατασκευή μοντέλων βαθιάς μάθησης. Παρέχει δυνατότητες εύκολης υλοποίησης τους, αλλά και μεγάλη ταχύτητα και αποτελεσματικότητα στην εκπαίδευσή τους. Διαθέτει έτοιμες κλάσεις που αναλαμβάνουν εξολοκλήρου τη διαδικασία της εκπαίδευσης. [32]

- **Matplotlib:** Μια βιβλιοθήκη, η οποία παρέχει υπηρεσίες οπτικοποίησης των δεδομένων, περιέχοντας μεγάλη ποικιλία διαγραμμάτων, όπως γραμμής, διασποράς, ιστογράμματα, **boxplots** και πολλά ακόμα. [33]
- **Seaborn:** Ακόμα, μια βιβλιοθήκη οπτικοποίησης δεδομένων, η οποία έχει ως βάση τη **Matplotlib**, και διαθέτει όμοιες λειτουργίες, με έμφαση στα διαγράμματα στατιστικής. [34]
- **Statsmodels:** Μια βιβλιοθήκη που παρέχει μια τεράστια ποικιλία εργαλείων στατιστικής, που εξυπηρετούν στην ανάλυση δεδομένων, αλλά και χρήση στατιστικών προβλεπτικών μοντέλων, ιδιαίτερα χρήσιμη για τη μελέτη χρονοσειρών. Περιέχει πολλά στατιστικά τεστ, μοντέλα όπως το **ARIMA** και στατιστικά διαγράμματα, όπως τα διαγράμματα αυτοσυσχέτισης. [35]
- **Scikit-learn:** Μια βιβλιοθήκη υλοποίησης και κατασκευής μοντέλων μηχανικής μάθησης. Παρέχει υπηρεσίες εκπαίδευσής τους, διαχωρισμού σε σύνολα εκπαίδευσης και επαλήθευσης, αλλά και της συναρτήσεων, των περισσότερων ευρέως χρησιμοποιούμενων, μετρητών αποτίμησης. [36]
- **Math:** Μια βιβλιοθήκη της **Python** που περιέχει βασικές μαθηματικές συναρτήσεις, όπως ρίζες, λογαρίθμους και πολλά ακόμη.

4.2 Περιγραφή δεδομένων

Για κατασκευή προβλεπτικών μοντέλων βαθιάς μάθησης ένας πολύ σημαντικός παράγοντας είναι εύρεση και εξόρυξη ενός έγκυρου συνόλου δεδομένων με μεγάλο πλήθος παρατηρήσεων. Μια ιστοσελίδα η οποία περιέχει τέτοιου είδους βάσεις δεδομένων είναι η ιστοσελίδα **Kaggle**, από την οποία χρησιμοποιήθηκε το σύνολο δεδομένων με όνομα "Panama Electricity Load Forecasting" [37], για την εκπόνηση της παρούσας διπλωματικής εργασίας. Πρόκειται, για ένα σύνολο δεδομένων χρονοσειρών που περιέχει τις τιμές του ηλεκτρικού φορτίου της χώρας του Παναμά, μαζί με διάφορα ιστορικά περιβαλλοντικά δεδομένα, αλλά και δεδομένα που υποδεικνύουν διάφορες πληροφορίες σχετικές με τις εορταστικές και τις εργάσιμες μέρες. Στο εν λόγω σύνολο, περιέχονται τρία διαφορετικά αρχεία **.csv**, τα οποία περιέχουν ιστορικές παρατηρήσεις σε διακριτό ωριαίο χρόνο για τις τιμές που αναφέρθηκαν. Από αυτά, το πρώτο που ονομάζεται "train.csv", περιέχει τιμές από την 3η Ιανουαρίου του 2015 και ώρα 01:00:00 έως την 31η Δεκεμβρίου του 2019 και ώρα 23:00:00 και είναι το σετ, το οποίο προτείνεται από τον δημιουργό του για την εκπαίδευση του εκάστοτε μοντέλου που δύναται να χρησιμοποιηθεί. Επιπλέον, τα υπόλοιπα δύο, με ονομασίες "Test_Jan" και "Predict_Jan", περιέχουν ξεχωριστά για το μήνα Ιανουάριο του 2020 τα περιβαλλοντικά και τα υπόλοιπα χαρακτηριστικά (**features**) και τις τιμές του ηλεκτρικού ρεύματος αντίστοιχα,

τα οποία προτείνει ο δημιουργός για την επαλήθευση των προβλέψεων. Η χρήση του συνόλου αυτού, διαφοροποιείται από την προτεινόμενη του δημιουργού και έτσι πραγματοποιήθηκε η συγχώνευσή τους, με το τελικό να περιέχει παρατηρήσεις από την 3η Ιανουαρίου του 2015 και ώρα 01:00:00 έως την 31η Ιανουαρίου του 2020 και ώρα 23:00:00. Στον πίνακα 4.1, αναγράφονται οι ονομασίες των χαρακτηριστικών του συνόλου δεδομένων, καθώς και σύντομες περιγραφές για αυτά.

Όνομα χαρακτηριστικού	Περιγραφή
datetime	Ημερομηνία και ώρα της ζώνης ώρας του Παναμά UTC-05:00 (index)
nat_demand	Εθνικό φορτίο ηλεκτρικού ρεύματος (Στόχος) (MWh)
T2M_toc	Θερμοκρασία στα 2 μέτρα υψόμετρο στην πόλη Τόκιουμιν (°C)
QV2M_toc	Σχετική υγρασία στα 2 μέτρα υψόμετρο στην πόλη Τόκιουμιν (%)
TQL_toc	Υετός στην πόλη Τόκιουμιν (liters/m ²)
W2M_toc	Ταχύτητα του αέρα στην πόλη Τόκιουμιν (m/s)
T2M_san	Θερμοκρασία στα 2 μέτρα υψόμετρο στην πόλη Σαντιάγκο (°C)
QV2M_san	Σχετική υγρασία στα 2 μέτρα υψόμετρο στην πόλη Σαντιάγκο (%)
TQL_san	Υετός στην πόλη Σαντιάγκο (liters/m ²)
W2M_san	Ταχύτητα του αέρα στην πόλη Σαντιάγκο (m/s)
T2M_dav	Θερμοκρασία στα 2 μέτρα υψόμετρο στην πόλη Ντέιβιντ (°C)
QV2M_dav	Σχετική υγρασία στα 2 μέτρα υψόμετρο στην πόλη Ντέιβιντ (%)
TQL_dav	Υετός στην πόλη Ντέιβιντ (liters/m ²)
W2M_dav	Ταχύτητα του αέρα στην πόλη Ντέιβιντ (m/s)
Holiday_ID	Απαρίθμηση, με ακέραιο αριθμό, του είδους της εορταστικής ημέρας
holiday	Διαδικός δείκτης εορταστικής ημέρας (1 = Εορταστική ημέρα και 0 = Κανονική ημέρα)
school	Διαδικός δείκτης ημέρας λειτουργίας σχολείου (1 = Σχολείο, 0 = Αργία)

Πίνακας 4.1: Χαρακτηριστικά συνόλου δεδομένων και περιγραφές

Έτσι, στο τελικό σύνολο δεδομένων, περιέχονται τα παραπάνω χαρακτηριστικά με έναν συνολικό αριθμό τις 44519 γραμμές και 16 στήλες.

4.3 Ανάλυση δεδομένων

Πριν την κατασκευή οποιουδήποτε μοντέλου μηχανικής μάθησης, αλλά και γενικότερα χειρισμού δεδομένων, μια πολύ χρήσιμη ενέργεια είναι η διερεύνηση και ανάλυση τους, με σκοπό την άντληση πληροφοριών από αυτά, οι οποίες βοηθούν στην κατανόηση της δομής τους, τον εντοπισμό κενών τιμών, οι οποίες πρέπει να 'καθαριστούν' από το σύνολό τους, την επιλογή των πιο επιδραστικών για το μοντέλο χαρακτηριστικών, την οπτικοποίηση, η οποία βοηθάει και αυτή στην κατανόηση, αλλά και την αναγνώριση μοτίβων μεταξύ τους και, γενικότερα, τη δημιουργία αποτελεσματικότερων προβλεπτικών μοντέλων.

Έτσι, μέσω της ενότητας αυτής, παρουσιάζονται μερικές χρήσιμες πληροφορίες για

το σύνολο δεδομένων, που προαναφέρθηκε, από την ανάλυσή του. Αρχικά, στο σύνολο αυτό, όπως αναφέρεται και στην περιγραφή του δημιουργού του, περιέχονται μόνο αριθμητικές τιμές, αλλά και διατίθενται όλες οι τιμές, δηλαδή δεν υπάρχουν κενές τιμές (null values). Παρόλα αυτά, πραγματοποιείται έλεγχος για τα παραπάνω, αλλά και παρατηρούνται οι πρώτες γραμμές του:

	datetime	nat_demand	T2M_toc	QV2M_toc	TQL_toc	W2M_toc	T2M_san	QV2M_san	TQL_san	W2M_san	T2M_dav	QV2M_dav	TQL_dav	W2M_dav	t
0	03-01-2015 01:00	970.3450	25.865259	0.018576	0.016174	21.850546	23.482446	0.017272	0.001855	10.328949	22.662134	0.016562	0.096100	5.364148	
1	03-01-2015 02:00	912.1755	25.899255	0.018653	0.016418	22.166944	23.399255	0.017265	0.001327	10.681517	22.578943	0.016509	0.087646	5.572471	
2	03-01-2015 03:00	900.2688	25.937280	0.018768	0.015480	22.454911	23.343530	0.017211	0.001428	10.874924	22.531030	0.016479	0.078735	5.871184	
3	03-01-2015 04:00	889.9538	25.957544	0.018890	0.016273	22.110481	23.238794	0.017128	0.002599	10.518620	22.512231	0.016487	0.068390	5.883621	
4	03-01-2015 05:00	893.6865	25.973840	0.018981	0.017281	21.186089	23.075403	0.017059	0.001729	9.733589	22.481653	0.016456	0.064362	5.611724	

Holiday_ID	holiday	school
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

Σχήμα 4.1: Δείγμα χαρακτηριστικών του συνόλου δεδομένων

```

datetime      object
nat_demand    float64
T2M_toc       float64
QV2M_toc      float64
TQL_toc       float64
W2M_toc       float64
T2M_san       float64
QV2M_san      float64
TQL_san       float64
W2M_san       float64
T2M_dav       float64
QV2M_dav      float64
TQL_dav       float64
W2M_dav       float64
Holiday_ID    int64
holiday       int64
school        int64
dtype: object

```

Σχήμα 4.2: Τύπος χαρακτηριστικών

```

datetime      0
nat_demand    0
T2M_toc       0
QV2M_toc      0
TQL_toc       0
W2M_toc       0
T2M_san       0
QV2M_san      0
TQL_san       0
W2M_san       0
T2M_dav       0
QV2M_dav      0
TQL_dav       0
W2M_dav       0
Holiday_ID    0
holiday       0
school        0
dtype: int64

```

Σχήμα 4.3: Έλεγχος κενών τιμών

Επομένως, επιβεβαιώνονται τα σχόλια του δημιουργού, όσον αφορά τα παραπάνω. Έπειτα, παρατηρήθηκε ότι ο τύπος του χαρακτηριστικού **datetime** είναι **object**, επομένως πραγματοποιήθηκε η μετατροπή του σε μεταβλητή **datetime64**, της μορφής (έτος - μήνας

- ημέρα ώρα : λεπτά : δευτερόλεπτα) και χρησιμοποιήθηκε ως δείκτης (index) του συνόλου, δίνοντας του την ιδιότητα της χρονοσειράς.

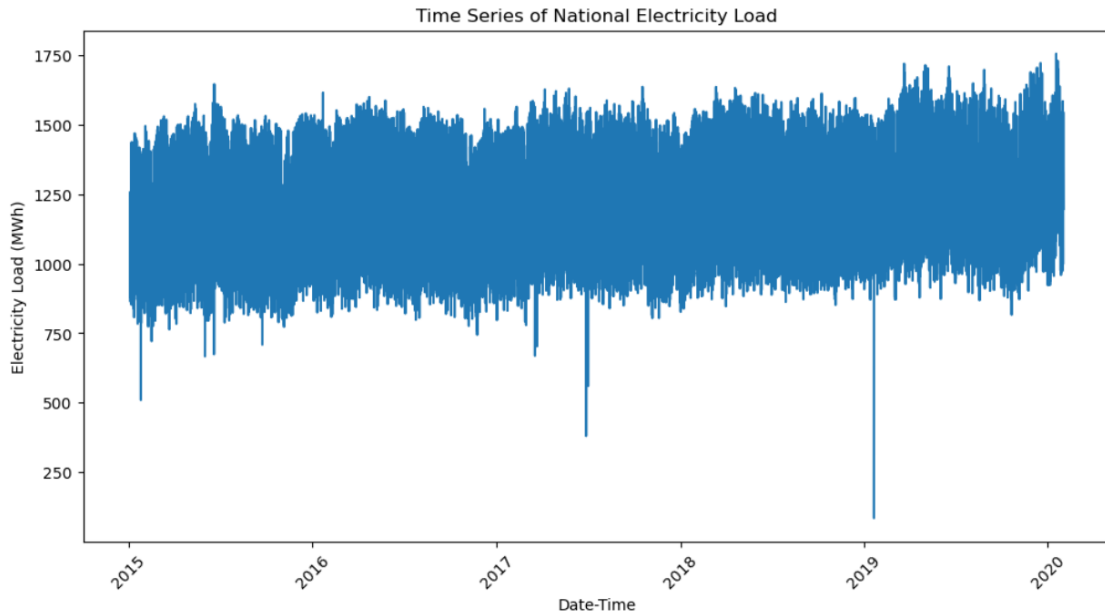
Χρησιμοποιώντας την εντολή `describe()`, λαμβάνονται διάφορα στατιστικά του κάθε χαρακτηριστικού ξεχωριστά της χρονοσειράς, τα οποία δίνουν πληροφορία για το συνολικό αριθμό παρατηρήσεων, τη μέση τιμή και την τυπική απόκλιση, την ελάχιστη και μέγιστη τιμή και οι τιμές 25%, 50% και 75% που ορίζονται ως τεταρτημόρια (quartiles). Στον παρακάτω πίνακα αναγράφονται οι τιμές που αναφέρθηκαν:

	count	mean	std	min	25%	50%	75%	max
nat_demand	44519.000	1185.647	193.856	85.192	1018.565	1175.276	1334.062	1754.882
T2M_toc	44519.000	27.374	1.657	22.953	26.156	27.102	28.522	35.040
QV2M_toc	44519.000	0.018	0.002	0.012	0.017	0.019	0.020	0.023
TQL_toc	44519.000	0.082	0.066	0.000	0.028	0.067	0.119	0.521
W2M_toc	44519.000	13.133	7.179	0.009	7.460	11.901	18.207	39.230
T2M_san	44519.000	26.855	2.961	19.765	24.754	26.133	28.642	39.063
QV2M_san	44519.000	0.018	0.002	0.010	0.016	0.018	0.019	0.022
TQL_san	44519.000	0.109	0.086	0.000	0.039	0.088	0.160	0.485
W2M_san	44519.000	6.966	4.062	0.060	3.933	5.923	9.211	24.484
T2M_dav	44519.000	24.672	2.370	19.934	22.945	23.976	26.174	34.216
QV2M_dav	44519.000	0.017	0.002	0.010	0.016	0.018	0.018	0.021
TQL_dav	44519.000	0.148	0.088	0.000	0.079	0.133	0.204	0.478
W2M_dav	44519.000	3.567	1.709	0.015	2.303	3.412	4.660	10.289
Holiday_ID	44519.000	0.720	3.219	0.000	0.000	0.000	0.000	22.000
holiday	44519.000	0.063	0.242	0.000	0.000	0.000	0.000	1.000
school	44519.000	0.727	0.445	0.000	0.000	1.000	1.000	1.000

Πίνακας 4.2: Βασικά στατιστικά των χαρακτηριστικών

Από τον πίνακα 4.2 αξίζει να σημειωθούν ο μέσος όρος της μεταβλητής στόχου που είναι τα 1185.647 MWh και η τυπική απόκλιση που είναι τα 193.856 MWh γεγονός που προϋποθέτει για ομαλότητα στις τιμές του ηλεκτρικού φορτίου, καθώς υπάρχει μεγάλη διαφορά σε αυτές τις δύο τιμές. Επιπρόσθετα, η ελάχιστη τιμή του ισούται με 85.192 MWh, η οποία απέχει αρκετά από το μέσο όρο και τη μέση τιμή που ισούται με το 50%, άρα πολύ πιθανόν να πρόκειται για κάποια διακοπή ρεύματος (outage). Για τα υπόλοιπα χαρακτηριστικά, φαίνεται να υπάρχει κάποιος θόρυβος στα περιβαλλοντικά δεδομένα του καιρού και πιο συγκεκριμένα στην ταχύτητα του αέρα και ελαφρώς λιγότερος στις θερμοκρασίες.

Στη συνέχεια, χρησιμοποιώντας τις βιβλιοθήκες **matplotlib** και **seaborn**, δημιουργήθηκε μια πληθώρα γραφημάτων, από τα οποία αντλούνται πολλές πληροφορίες που βοηθούν στην κατανόηση της εν λόγω χρονοσειράς. Η γραφική παράσταση του φορτίου ηλεκτρικού ρεύματος σε συνάρτηση με το χρόνο είναι:

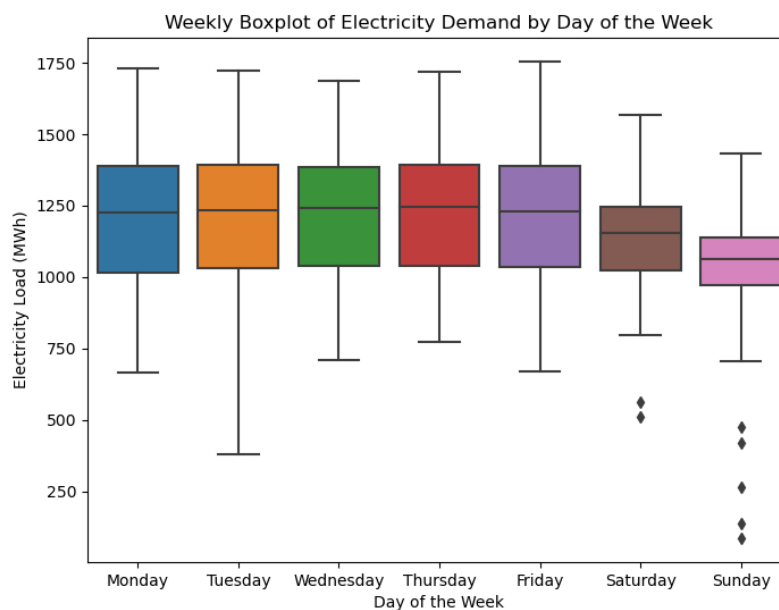


Σχήμα 4.4: Φορτίο ηλεκτρικού ρεύματος - χρόνος

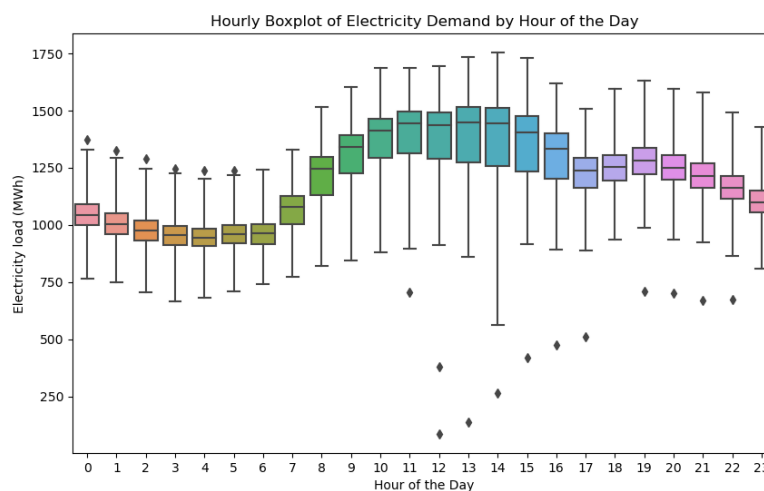
Λόγω του μεγάλου πλήθους των δεδομένων δεν διακρίνεται με ευκολία κάποιο μοτίβο κατά μήκος του χρονικού ορίζοντα, παρά μόνο μια ελαφριά αύξηση των τιμών με το πέρασμα του χρόνου. Μια πληροφορία που μπορεί να ληφθεί, βέβαια, είναι ο εντοπισμός των τιμών, οι οποίες απέχουν αρκετά από τις συνηθέστερες παρατηρήσεις τις χρονοσειράς και πιο συγκεκριμένα είναι αρκετά μικρότερες. Αυτού του είδους οι τιμές καλούνται έκτοπες (**outliers**) και στη συγκεκριμένη περίπτωση, αφού τα χρησιμοποιούμενα δεδομένα χαρακτηρίζουν την ωριαία κατανάλωση ηλεκτρικού ρεύματος, είναι πολύ πιθανό να ταυτίζονται με κάποιες διακοπές ρεύματος είτε πολύωρες αν είναι συνεχόμενες αυτού του είδους οι τιμές είτε στιγμιαίες/ωριαίες. Ένα πιθανό πρόβλημα που ενδέχεται να προκαλέσουν οι εν λόγω τιμές είναι ο αποσυντονισμός του μοντέλου κατά την εκπαίδευση, καθώς το δυσκολεύουν, στην ουσία, να εντοπίσει το μοτίβο στις συγκεκριμένες περιπτώσεις και γιαυτό το λόγο χρειάζονται κάποια ιδιαίτερη μεταχείριση.

Ένας τύπος διαγράμματος, ο οποίος προσφέρει, οπτικά, στατιστικές πληροφορίες για την χρονοσειρά είναι τα λεγόμενα **boxplots**, τα οποία απεικονίζουν την κεντρική τάση των τιμών, δηλαδή τη μέση τιμή, το πόσο αυτές διασκορπίζονται, αλλά και κάποια πιθανά **outliers**. Αποτελούνται από ένα κουτί με μια κεντρική γραμμή, η οποία δηλώνει τη μέση τιμή και οι πάνω και άνω άκρες του τις τιμές 75% και 25% αντίστοιχα. Ε-

πιπλέον, επιλέγεται μια τιμή που ονομάζεται **whiskers**, η οποία δηλώνει το εύρος των τιμών που διασκορπίζονται πέρα από τις άκρες των κουτιών, με τυπική τιμή το 1.5. Τέλος, απεικονίζονται κάποια σημεία έξω από το εύρος αυτό, τα οποία χαρακτηρίζουν τις τιμές που είναι **outliers**. Ένας πιο αποτελεσματικός τρόπος, για το λόγο ότι το σύνολο αυτό αποτελείται από ωραίες παρατηρήσεις, για τη μελέτη των διαγραμμάτων είναι οπτικοποιώντας σε ημερίσια, εβδομαδιαία και ακόμα και μηναία πλαίσια. Έτσι, αξίζει να κατασκευαστούν τα **boxplots** του φορτίου ανά ημέρα της εβδομάδας και ανά ώρα της ημέρας, τα οποία είναι:

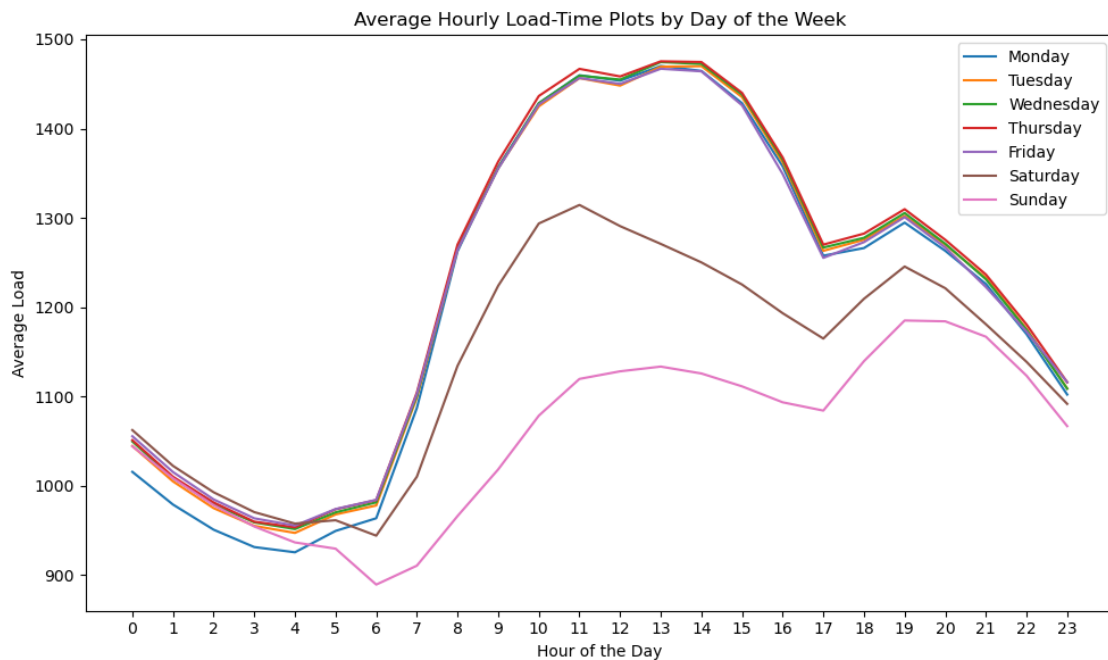


Σχήμα 4.5: Boxplot ανά ημέρα της εβδομάδας



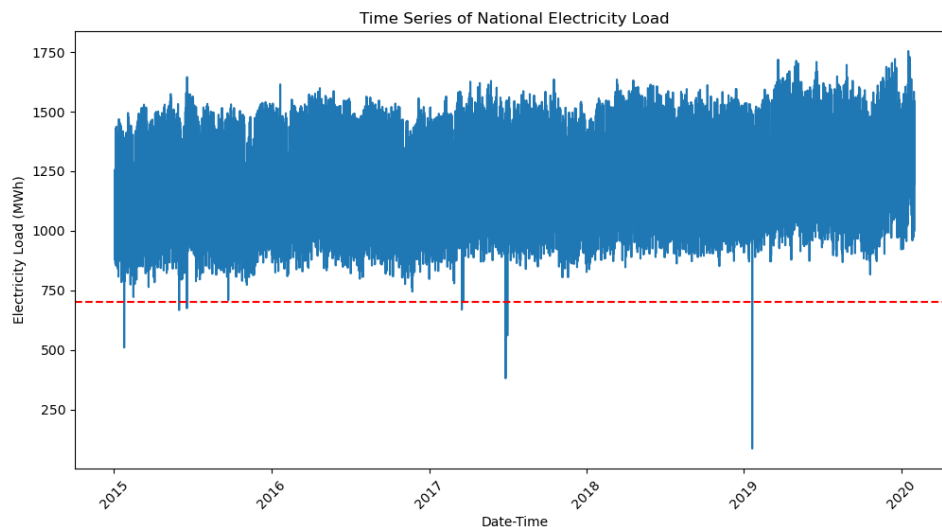
Σχήμα 4.6: Boxplot ανά ώρα της ημέρας

Από τα διαγράμματα φαίνεται ότι οι τιμές του φορτίου ηλεκτρικού ρεύματος εμφανίζουν μια διαφοροποίηση κατά το σαββατοκύριακο, ενώ τις καθημερινές είναι κατά βάση όμοιες, πράγμα το οποίο είναι λογικό, καθώς πολλές βιομηχανίες υψηλής κατανάλωσης, δεν βρίσκονται σε λειτουργία εκείνες τις ημέρες. Επιπλέον, όσον αφορά την ώρα της ημέρας, οι ώρες, κατά τις οποίες πραγματοποιούνται οι εργασιακές βάρδιες (07:00:00 - 17:00:00) εμφανίζουν τη μεγαλύτερη κατανάλωση ηλεκτρικής ενέργειας, με μέγιστες τιμές (**peaks**) να παρατηρούνται κατά τις ώρες 13:00:00 και 15:00:00, αλλά και τις ώρες κατά τις οποίες χρησιμοποιούνται σε μεγάλο βαθμό οι οικιακές συσκευές κατά τις βραδινές ώρες 19:00:00 και 20:00:00. Επίσης, στο διάγραμμα 4.6 απεικονίζεται το μέσο μοτίβο του φορτίου στον χρονικό ορίζοντα μιας ημέρας. Τα παραπάνω συμπεράσματα, μπορούν να γίνουν ακόμα περισσότερο κατανοητά κατασκευάζοντας ένα μέσο διάγραμμα των τιμών του φορτίου ανά ημέρα της εβδομάδας συναρτήσει της ώρας της ημέρας:

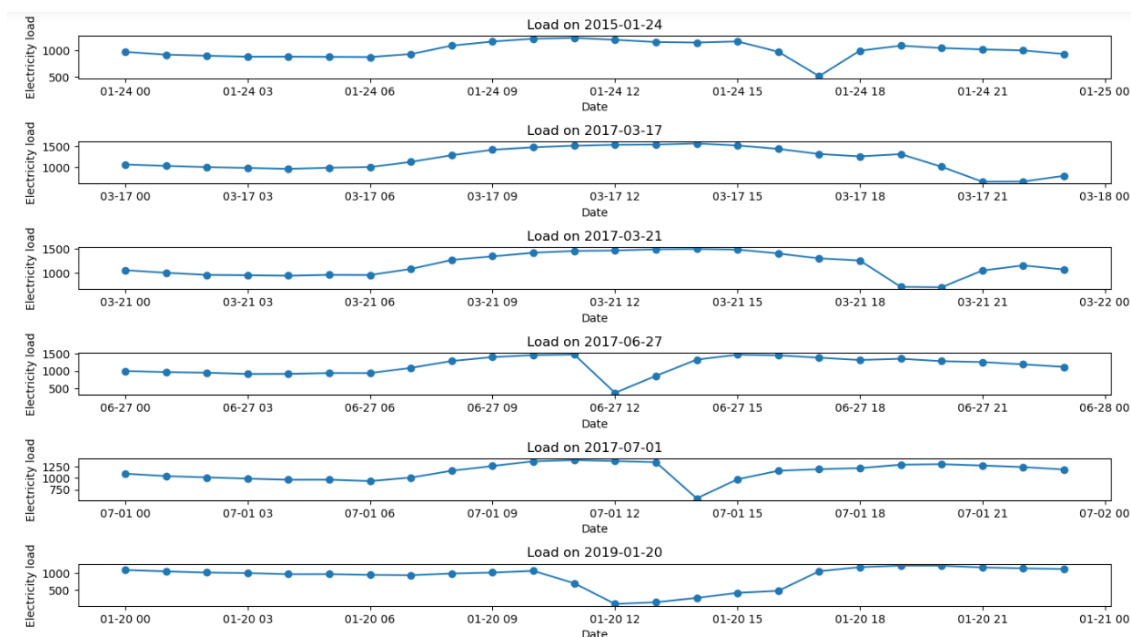


Σχήμα 4.7: Μέσος όρος φορτίου ανά ημέρα της εβδομάδας - ώρα της ημέρας

Τώρα, συνδυάζοντας τα διαγράμματα 4.4, 4.5 και 4.6 υπάρχει η δυνατότητα εντοπισμού των **outliers**. Υπάρχουν διάφοροι τρόποι εντοπισμού τους, αλλά στην προκειμένη προτιμήθηκε ο οπτικός. Στη γραφική παράσταση του φορτίου, παρατηρούνται μεμονομένες τιμές περίπου στην τιμή 700 MWh και κάτω, οι οποίες εμφανίζουν μεγάλη απόκλιση από τις συνηθέστερες, ενώ δεν φαίνεται να υπάρχει κάποια τέτοια άλλη τιμή που να απέχει τόσο από το αναμενόμενο μοτίβο (εικόνα 4.8). Ο λόγος εμφάνισης της εν λόγω απόκλισης είναι άγνωστος και κρίζει περεταίρω διερεύνησης, με έναν αποτελεσματικό τρόπο να είναι οι κατασκευή ημερησίων γραφικών παραστάσεων των ημερών που αυτές παρατηρήθηκαν (εικόνα 4.9).



Σχήμα 4.8: Παρατηρήσεις κάτω από 720 MWh



Σχήμα 4.9: Διαγράμματα ημερών με outliers

Σε όλες τις ημέρες που απεικονίζονται παραπάνω, παρατηρείται μια απότομη πτώση της τιμής του φορτίου σε ώρες, τις οποίες αναμένεται να υπάρχει μια πολύ μεγάλη, αν όχι η μεγαλύτερη, κατανάλωση μέσα στην ημέρα. Επομένως, όπως έχει αναφερθεί, οι συγκεκριμένες τιμές ίσως να προκύπτουν από κάποια πιθανή διακοπή ρεύματος και άρα μπορούν να χαρακτηριστούν ως **outliers**.

Δύο πολύ σημαντικά διαγράμματα στην ανάλυση χρονοσειρών είναι τα διαγράμματα των συναρτήσεων αυτοσυσχέτισης και μερικής αυτοσυσχέτισης, από τα οποία προκύπτει οποιαδήποτε συσχέτιση μεταξύ της τιμής μιας μεταβλητής με την αντίστοιχη

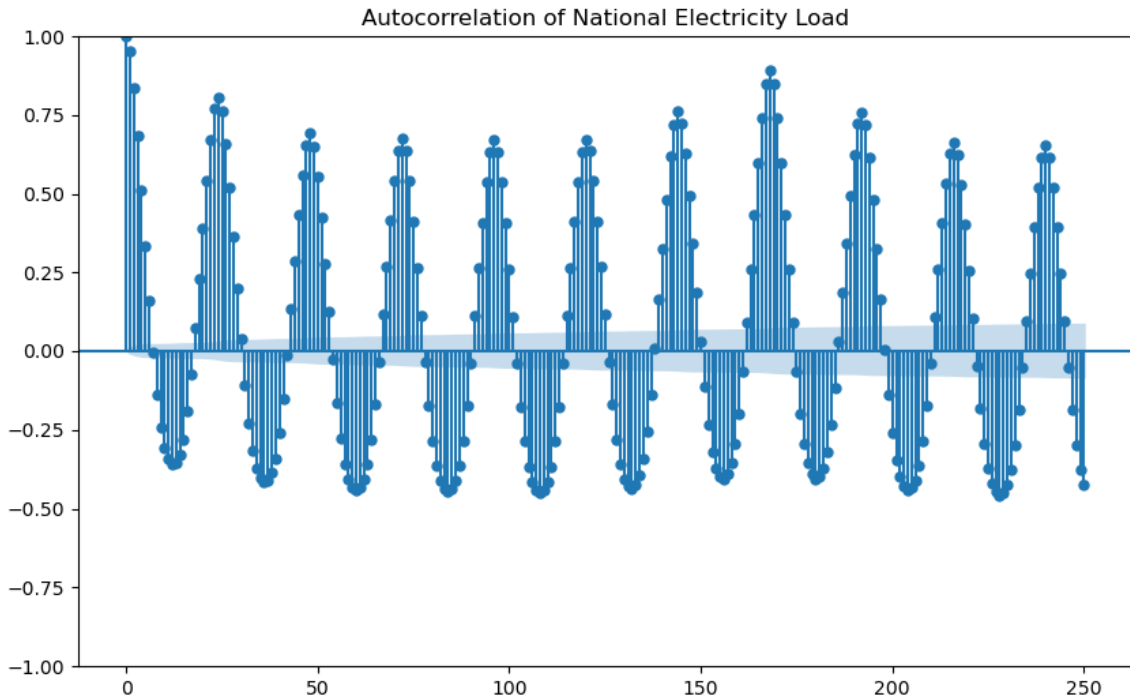
τιμή κάποια χρονικά βήματα πίσω (lags). Το πρώτο κατασκευάζεται μέσω του δειγματικού συντελεστή αυτοσυσχέτισης για καθυστέρηση k και χρόνο t :

$$\rho(k) = \frac{\sum_{t=k+1}^n (X_t - \bar{X})(X_{t-k} - \bar{X})}{\sum_{t=1}^n (X_t - \bar{X})^2} \quad (4.1)$$

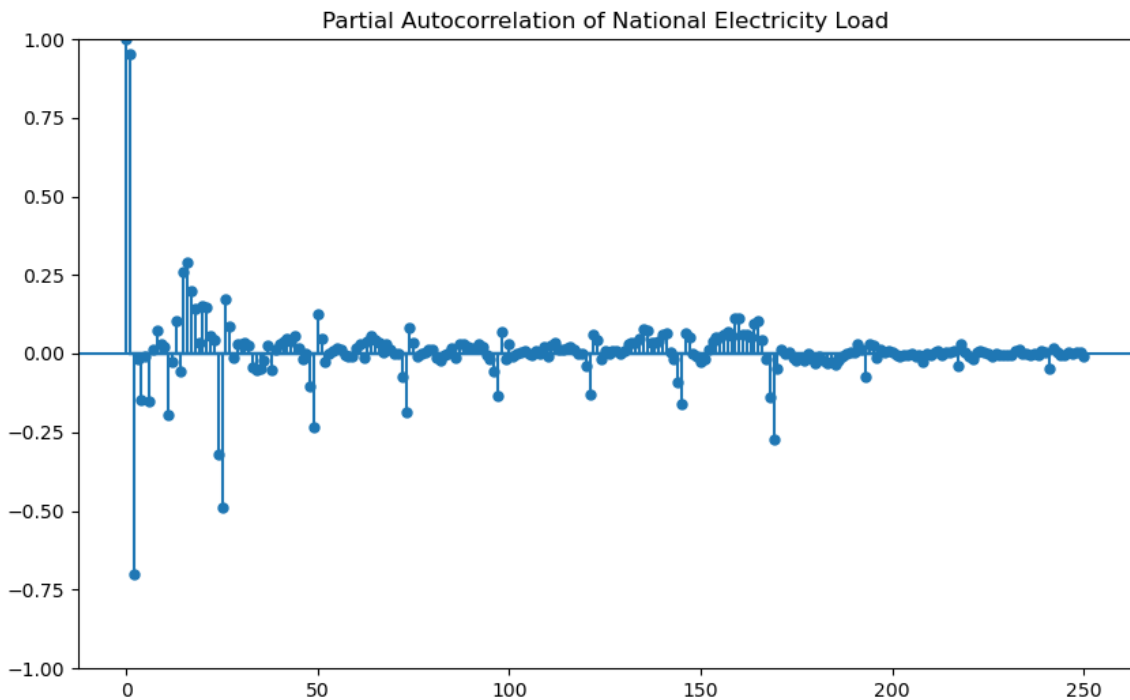
και δείχνει πόσο και πως μια τιμή τη χρονική στιγμή t συσχετίζεται με την τιμή της χρονικής τιμής $t - k$ παίρνοντας τιμές από $[-1, 1]$. Το δεύτερο κατασκευάζεται μέσω του δειγματικού συντελεστή μερικής αυτοσυσχέτισης :

$$\phi_{n,n} = \frac{\rho(n) - \sum_{k=1}^{n-1} \phi_{n-1,k} \rho(n-k)}{1 - \sum_{k=1}^{n-1} \phi_{n-1,k} \rho(k)} \quad (4.2)$$

και δείχνει το πόσο και πως μια τιμή τη χρονική στιγμή t συσχετίζεται με την τιμή της χρονικής τιμής $t - k$, αφαιρώντας όμως την αυτοσυσχέτιση των μεταξύ τους καθυστερήσεων παίρνοντας τιμές από $[-1, 1]$. Πρόκειται, δηλαδή, για μια καθαρή συσχέτιση μεταξύ 2 χρονικών βημάτων. Τα διαγράμματα που προκύπτουν είναι:



Σχήμα 4.10: ACF διάγραμμα



Σχήμα 4.11: PACF διάγραμμα

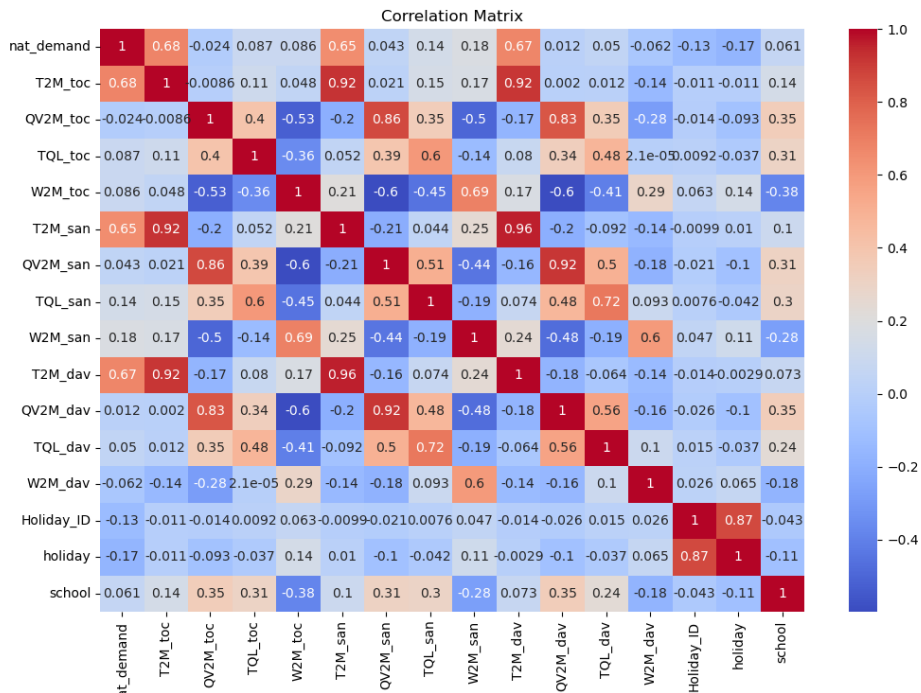
Από τα δύο αυτά διαγράμματα, γίνεται αντιληπτή η ημερίσια αυτοσυσχέτιση με την πιο ισχυρή, βέβαια, να παρατηρείται για $k = 168$ ώρες, δηλαδή μια εβδομάδα. Οι πληροφορίες αυτές, πολύ πιθανό να αποτελέσουν έναν πολύ σημαντικό παράγοντα στην εκπαίδευση των μοντέλων νευρωνικών δικτύων και την αύξηση της αποτελεσματικότητας των προβλέψεών τους.

Ένα ακόμη πολύ σημαντικό κομμάτι στην ανάλυση δεδομένων, όταν υπάρχει πληθώρα χαρακτηριστικών είναι η εύρεση αυτών που έχουν επίδραση στις μεταβολές των τιμών της μεταβλητής στόχου, δηλαδή του φορτίου ηλεκτρικού ρεύματος. Αυτό μπορεί να πραγματοποιηθεί με την κατασκευή ενός πίνακα αυτοσυσχέτισης, ο οποίος συμβολίζει τη γραμμική σχέση μεταξύ της μεταβλητής στόχου και οποιοδήποτε άλλου χαρακτηριστικού που περιέχεται στο σύνολο και τα στοιχεία του υπολογίζονται μέσω του συντελεστή συσχέτισης **Pearson** :

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (4.3)$$

Παίρνει τιμές στο $[-1, 1]$ και πρακτικά δείχνει πόσο και πως μεταβάλλεται η τιμή της μιας μεταβλητής, όταν μεταβάλλεται η άλλη. Ο αρνητικός συντελεστής χαρακτηρίζει αρνητική γραμμική σχέση μεταξύ των μεταβλητών, ενώ ο θετικός θετική γραμμική

σχέση. Έτσι ο πίνακας συσχέτισης είναι:

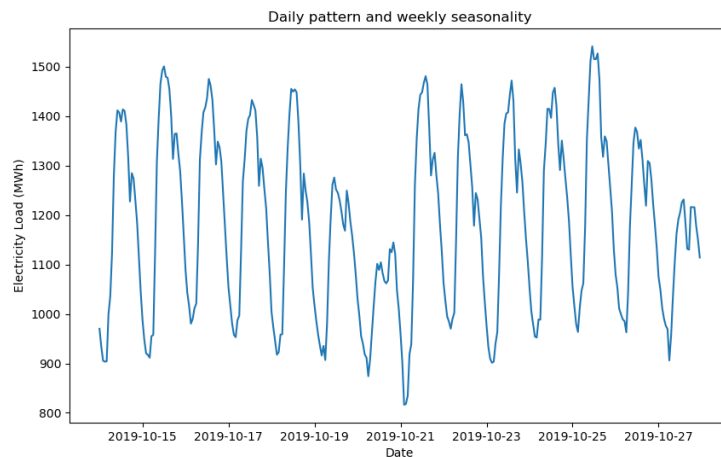


Σχήμα 4.12: Πίνακας συσχέτισης συνόλου δεδομένων (correlation matrix)

Από τον πίνακα αυτό και παρατηρώντας την πρώτη στήλη, οι θερμοκρασίες στις τρεις διαφορετικές πόλεις φαίνεται να είναι τα πιο σημαντικά χαρακτηριστικά με υψηλούς συντελεστές συσχέτισης (περίπου 0.7), ενώ κάποια μικρή αλλά σημαντική επίδραση εμφανίζουν η ταχύτητα του αέρα και ο υετός στην πόλη του Σαντιάγκο. Τέλος, μια αρνητική συσχέτιση εξίσου μικρή φαίνεται στα χαρακτηριστικά που σχετίζονται με τις εορταστικές ημέρες, γεγονός που είναι λογικό, καθώς τις εορταστικές ημέρες αναμένονται αρκετές διαφοροποιήσεις στην κατανάλωση (μικρότερη) του ηλεκτρικού ρεύματος σε σχέση με τη μέση αναμενόμενη. Άρα, θεωρώντας ότι η επίδραση στην μεταβλητή στόχο εμφανίζεται για συντελεστή με απόλυτη τιμή από 0.1 και άνω, τα πιο σημαντικά χαρακτηριστικά είναι τα T2M_toc, T2M_dav, T2M_san, W2M_san, TQL_san, holiday και Holiday_ID.

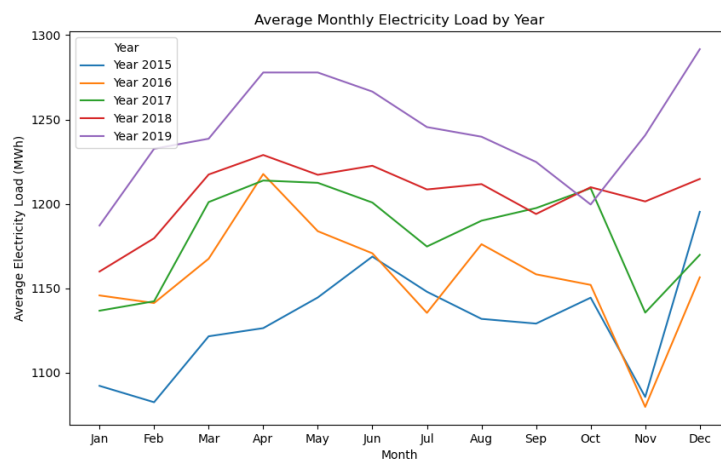
Επιπροσθέτως, μια πολύ σημαντική πληροφορία κατά τη μελέτη και την ανάλυση χρονοσειρών είναι η εύρεση πιθανών επαναλαμβανόμενων μοτίβων σε συγκεκριμένα χρονικά διαστήματα, τα οποία μπορούν να χαρακτηριστούν και ως διαστήματα εποχικότητας. Προηγουμένως, εντοπίστηκαν οι διαφορές, για παράδειγμα, των μοτίβων των καθημερινών σε σχέση με τα σαββατοκύριακα. Κάτι αντίστοιχο συμβαίνει και με τις αργίες. Εκτός αυτού εντοπίστηκαν και μέσω του διαγράμματος αυτοσυσχέτισης ομοιότητες και υψηλή αυτοσυσχέτιση στο φορτίο ίδιων ημερών σε διάστημα μιας εβδομάδας, γεγονός

που παραπέμπει σε εβδομαδιαία εποχικότητα. Επίσης, μια άλλη πληροφορία που λήφθηκε ήταν η εύρεση του μοτίβου κατά τη διάρκεια της ημέρας. Στο παρακάτω διάγραμμα, εναποτίθεται ένα παράδειγμα μιας τυχαίας εβδομάδας, όπου δεν υπάρχει κάποια αργία, όπου παρατηρούνται όσα αναφέρθηκαν παραπάνω.

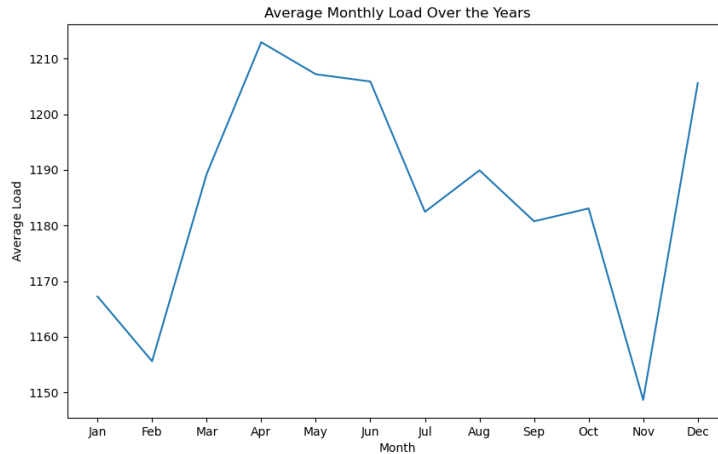


Σχήμα 4.13: Ημερήσιο μοτίβο και εβδομαδιαία εποχικότητα

Σε συνέχεια των προηγούμενων, εκτός από τα εβδομαδιαία μοτίβα, η κατανάλωση του φορτίου αναμένεται να διαφοροποιείται ανάλογα την εποχή του έτους, λόγω της ύπαρξης διαφορετικών αναγκών σε κάθε περίοδο στα περισσότερα κράτη. Στην περίπτωση, βέβαια, του Παναμά, όπου επικρατεί τροπικό κλίμα, ενδέχεται να υπάρχουν διαφοροποιήσεις. Στα παρακάτω διαγράμματα απεικονίζονται το μέσο φορτίο ανά μήνα ανά ξεχωριστό έτος και το μέσο φορτίο στη διάρκεια ολόκληρου του συνόλου δεδομένων.



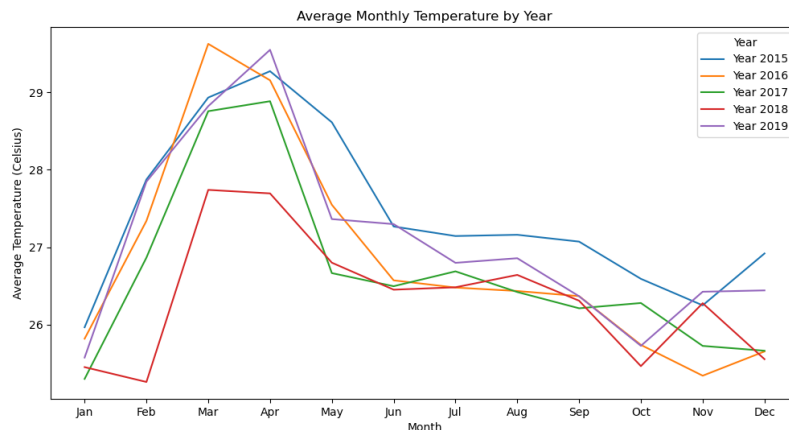
Σχήμα 4.14: Μέσο μηνιαίο φορτίο ανά έτος



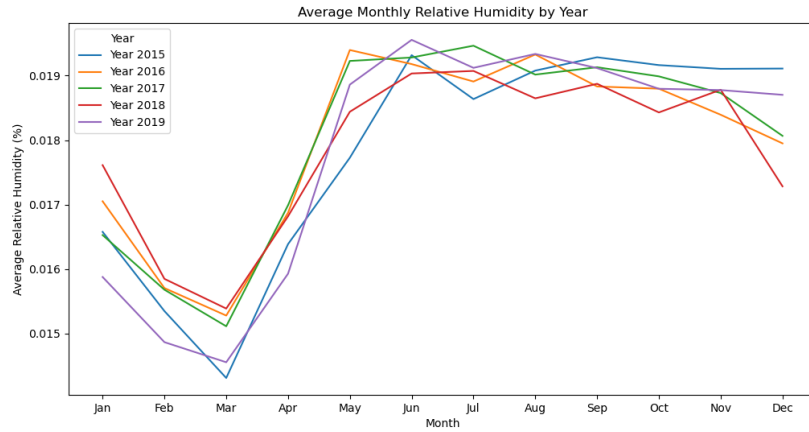
Σχήμα 4.15: Μέσο μηνιαίο φορτίο όλων των ετών

Αρχικά, παρατηρείται η αύξηση των τιμών με το πέρασμα των ετών, πράγμα που πα-
ρεπέμπει στην ύπαρξη μιας θετικής τάσης της χρονοσειράς. Η επόμενη πληροφορία
που λαμβάνεται είναι ότι σε γενικές γραμμές δεν υπάρχουν τεράστιες διαφορές στην
κατανάλωση μεταξύ των μηνών, αλλά η μέγιστη μέση κατανάλωση σημειώνεται κατά
τους μήνες Απρίλιο και Μάιο και η ελάχιστη κατά τους μήνες Νοέμβριο και Ιανουάριο
και Φεβρουάριο. Τα παραπάνω, ενδέχεται να οφείλονται σε πολλούς παράγοντες, με
τους περιβαλλοντικούς και την τουριστική περίοδο να ασκούν τη μεγαλύτερη επιρροή.

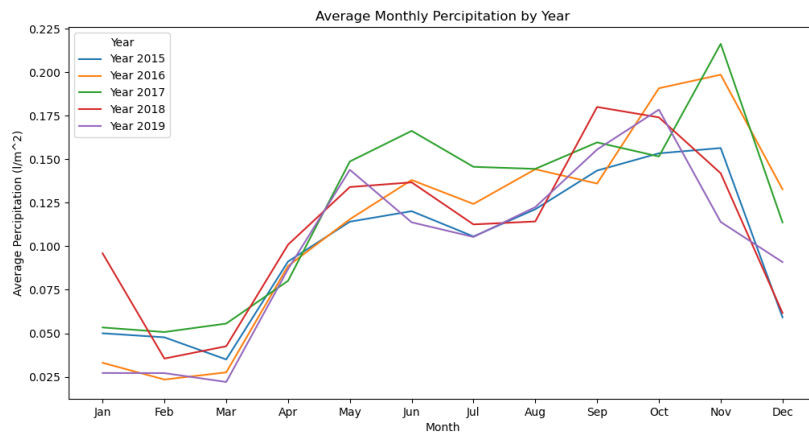
Τέλος, εκτός από το τη μεταβλητή στόχο, είναι αρκετά σημαντική και η εξερεύνηση
και άντληση πληροφορίας από τα υπόλοιπα χαρακτηριστικά. Παρακάτω εναποτίθενται
αντίστοιχα διαγράμματα σχετικά με τη θερμοκρασία, τη σχετική υγρασία, τον υετό
και την ταχύτητα του αέρα μιας από τις τρεις πόλεις που περιέχονται στο σύνολο των
δεδομένων.



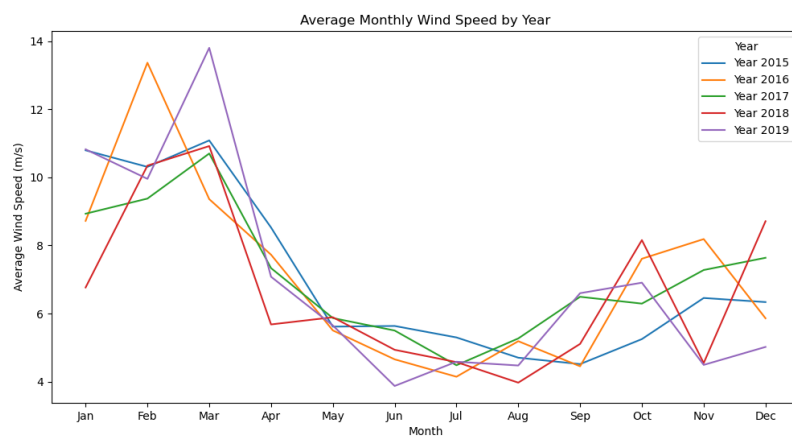
Σχήμα 4.16: Μέση μηνιαία θερμοκρασία ανά έτος



Σχήμα 4.17: Μέση μηνιαία σχετική υγρασία ανά έτος



Σχήμα 4.18: Μέσος μηνιαίος νετός ανά έτος



Σχήμα 4.19: Μέση μηνιαία ταχύτητα αέρα ανά έτος

Από τα παραπάνω περιβαλλοντικά διαγράμματα μπορούν να πραγματοποιηθούν κάποιες

εκτιμήσεις που θα δικαιολογήσουν τις τιμές των φορτίων σε μηνιαία κλίμακα των προηγούμενων διαγραμμάτων. Αρχικά, οι υψηλότερες θερμοκρασίες φαίνεται να εμφανίζονται κατά το μήνα Απρίλιο και έτσι η εμφάνιση της μεγαλύτερης κατανάλωσης εκείνο το μήνα μπορεί να οφείλεται σε αυτό το γεγονός, καθώς θα αυξάνονται οι ανάγκες για χρήση κλιματιστικών και συστημάτων ψύξης γενικότερα. Επιπρόσθετα, για το μήνα Νοέμβριο η ύπαρξη υψηλής σχετικής υγρασίας, αλλά και υετού, όπου δηλώνει τα ποσοστά βροχοπτώσεων, είναι πιθανό να οδηγεί στη χαμηλότερη κατανάλωση, λόγω του μειωμένου αριθμού τουριστών (αφιλόξενο κλίμα), αλλά και της μη ύπαρξης ανάγκης του προηγούμενου εξοπλισμού. Ο μήνας Δεκέμβριος θεωρείται από τους πιο δημοφιλείς μήνες για διακοπές, λόγω των Χριστουγέννων μέχρι και τις πρώτες ημέρες του Ιανουαρίου, γι'αυτό και παρατηρείται μια σχετική αύξηση του φορτίου μετά το Νοέμβριο. Οι μήνες Ιανουάριος και Φεβρουάριος, λόγω του ότι ανήκουν σε περιόδους όπου δεν υπάρχουν αργίες και χαμηλότερος τουρισμός σε συνδυασμό και με τις χαμηλότερες θερμοκρασίες είναι μήνες που σημειώνουν και αυτοί με τη σειρά τους χαμηλότερη κατανάλωση φορτίου.

Εν κατακλείδι, ένα γενικό συμπέρασμα είναι ότι η κατανάλωση του φορτίου ηλεκτρικού ρεύματος εμφανίζει όμοια ημερήσια, εβδομαδιαία και μηνιαία μοτίβα, με εξαίρεση τις περιπτώσεις των ειδικών ημερών (αργιών), όπως επίσης και θετική τάση με το πέρασμα των ετών.

5

Πειράματα και αποτελέσματα

Ενότητες

5.1	Περιγραφή πειραμάτων	59
5.2	Μεθοδολογία και προεπεξεργασία δεδομένων	60
5.3	Αποτελέσματα	67

Το παρόν κεφάλαιο πραγματεύεται τη διενέργεια διαφορετικών υπολογιστικών πειραμάτων με τη χρήση των τύπων νευρωνικών δικτύων που αναλύθηκαν και περιγράφηκαν στο Κεφάλαιο 3, χρησιμοποιώντας το σύνολο δεδομένων που αναφέρθηκε στο Κεφάλαιο 4. Αρχικά, εναποτίθεται μια περιγραφή των τριών πειραμάτων που διενεργούνται, όπου ο παράγοντας που μεταβάλλεται είναι τα δεδομένα εισόδου. Έπειτα, περιγράφεται η μεθοδολογία κατασκευής των μοντέλων (αριθμός νευρώνων και στρωμάτων, επιλογή υπερπαραμέτρων και τεχνικών), όπως και η βασική προεπεξεργασία που γίνεται στα δεδομένα, έτσι ώστε να τροφοδοτηθούν στο εκάστοτε νευρωνικό δίκτυο, και παρουσιάζονται τα αποτελέσματα, τα οποία προκύπτουν από τη σύγκριση των προβλέψεων με ένα σύνολο δεδομένων ελέγχου (Test set).

5.1 Περιγραφή πειραμάτων

Στην παρούσα διπλωματική εργασία πραγματοποιήθηκαν τρία πειράματα, δίνοντας κυρίως έμφαση στο είδος και στο πλήθος των δεδομένων εισόδου τα οποία χρησιμοποιούνται για την εκπαίδευση των νευρωνικών δικτύων, διατηρώντας σταθερή και την αρχιτεκτονική τους, καθώς και τις περισσότερες από τις υπερπαραμέτρους τους. Αρχικά, όπως σε κάθε μοντέλο μηχανικής μάθησης, συνίσταται η κατασκευή ενός μοντέλου βάσης (baseline model) ή η χρήση κάποιου προϋπάρχοντος σημείου αναφοράς (benchmark) από τη βιβλιογραφία. Έτσι, στη συγκεκριμένη περίπτωση κατασκευάστηκε το πιο απλό είδος μοντέλου πρόβλεψης, η λεγόμενη απλοϊκή μέθοδος (naive method) κατά την οποία γίνεται η υπόθεση ότι η πρόβλεψη τη χρονική στιγμή t θα ισούται την τιμή του ηλεκτρικού φορτίου της χρονικής στιγμής $t - 168$, δηλαδή με την τιμή του φορτίου της προηγούμενης εβδομάδας. Η επιλογή της τιμής αυτής, γίνεται λόγω του ότι η πρόβλεψη από τα επόμενα μοντέλα γίνεται απευθείας για 24 ώρες στο μέλλον και επομένως μπορούν να χρησιμοποιηθούν δεδομένα χρονολογικά 24 ώρες και πριν από την προβλεπόμενη τιμή. Επιπλέον, δεν επιλέγεται η τιμή της προηγούμενης ημέρας, καθώς κατά τις ίδιες ημέρες τις εβδομάδας εμφανίζονται όμοια μοτίβα, άρα η παραπάνω θεώρηση είναι η βέλτιστη. Σημειώνεται πως η ομοιότητα αυτή ενδέχεται να δώσει ικανοποιητικές προβλέψεις, με εξαιρέσεις τις ημέρες, τις οποίες είναι εμφανής κάποια τάση μεταξύ των εβδομαδιαίων τιμών, αλλά και των εορταστικών ημερών.

Έπειτα, τα πέντε διαφορετικά είδη νευρωνικών δικτύων που αναφέρθηκαν στο Κεφάλαιο 3 (FFNN, LSTM, GRU, Bidirectional LSTM και Bidirectional GRU) δοκιμάζονται σε όλα τα χαρακτηριστικά του συνόλου δεδομένων, τροφοδοτώντας τα με χρονικά βήματα 24, 48 και 72. Δηλαδή για inputs $\{X_{t-n}, X_{t-n+1}, \dots, X_t\}$, όπου n είναι το μήκος του χρονικού βήματος που χρησιμοποιείται για την πρόβλεψη των $\{X_{t+1}, \dots, X_{t+m}\}$, με m τον ορίζοντα πρόβλεψης. Στην προκειμένη χρησιμοποιήθηκαν $n = \{24, 48, 72\}$ και $m = 24$ για το κάθε μοντέλο ξεχωριστά. Έχοντας χωρίσει ήδη τα δεδομένα σε δεδομένα εκ-

παίδευσης και δεδομένα ελέγχου, χρησιμοποιήθηκε αρχικά η τεχνική **Early stopping**, προσδιορίζοντας ως σετ επαλήθευσης το 20% των δεδομένων εκπαίδευσης για ένα συγκεκριμένο μήκος χρονικού βήματος από τα παραπάνω με σκοπό την εύρεση του κατάλληλου αριθμού εποχών εκπαίδευσης. Έπειτα, η εκπαίδευση πραγματοποιήθηκε με ολόκληρο το σετ δεδομένων εκπαίδευσης στο μήκος εποχών, το οποίο επιλέχθηκε από την προηγούμενη ενέργεια, για όλα τα μοντέλα που αναφέρθηκαν παραπάνω.

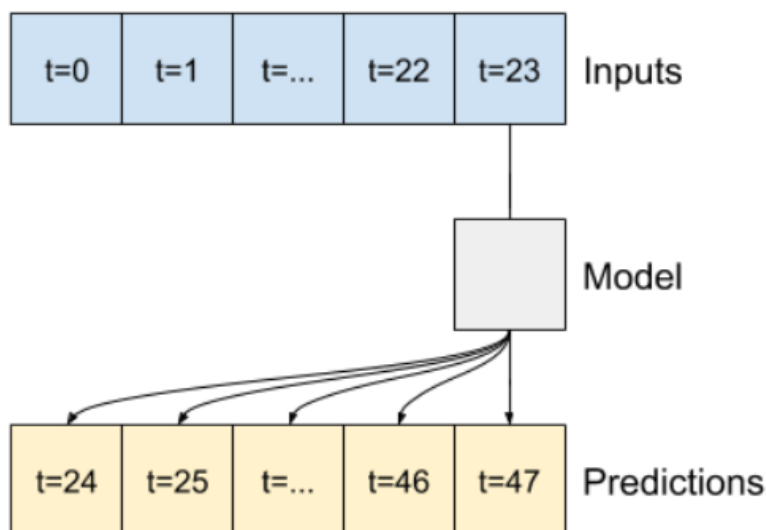
Στη συνέχεια, συγκρίνοντας τα αποτελέσματα των παραπάνω επιλέγεται ένα μήκος χρονικού βήματος τόσο για την αποτελεσματικότητά του, όσο και για το υπολογιστικό του κόστος και πραγματοποιείται το δεύτερο πείραμα, στο οποίο χρησιμοποιούνται ακριβώς τα ίδια μοντέλα, με τη διαφορά ότι αυτή τη φορά τα δεδομένα εκπαίδευσης, θα περιλαμβάνουν μόνο τα χαρακτηριστικά, τα οποία θεωρούνται πιο σημαντικά, σύμφωνα με τον πίνακα συσχέτισης 4.12.

Τέλος, το τρίτο πείραμα, έχει ως σκοπό την αύξηση της αποτελεσματικότητας των προβλέψεων των παραπάνω μοντέλων, πραγματοποιώντας το λεγόμενο χειρισμό των χαρακτηριστικών (**feature engineering**). Σε αυτή την τεχνική περιλαμβάνεται η χρήση όλων των χαρακτηριστικών που προέρχονται από το δημιουργό του συνόλου δεδομένων (πείραμα 1) και η χρήση των πιο σημαντικών (πείραμα 2), σε συνδυασμό με τη δημιουργία νέων με σκοπό να βοηθήσουν το εκάστοτε μοντέλο να εντοπίσει πιο αποτελεσματικά τα μοτίβα μεταξύ των δεδομένων εισόδου και δεδομένων εξόδου, τα οποία προκύπτουν από τα ήδη υπάρχοντα δεδομένα. Εκτός αυτού, αντιμετωπίζονται και τα **outliers**, τα οποία, όπως προαναφέρθηκε, πολύ πιθανόν να αναφέρονται σε κάποια διακοπή ρεύματος, αν η καταγραφή των δεδομένων είναι έγκυρη.

5.2 Μεθοδολογία και προεπεξεργασία δεδομένων

Τα βήματα που ακολουθούνται για την κατασκευή και την υλοποίηση προβλεπτικών μοντέλων νευρωνικών δικτύων με εφαρμογή στις χρονοσειρές δεν διαφοροποιούνται, σε γενικές γραμμές, από αυτά σε οποιαδήποτε άλλα μοντέλα μηχανικής μάθησης, όπως περιγράφηκαν και στο Κεφάλαιο 2. Η μεθοδολογία που περιγράφεται στην παρούσα ενότητα αναφέρεται στην απευθείας πρόβλεψη 24 ωρών στο μέλλον, λαμβάνοντας ως δεδομένα εισόδου τιμές που αναφέρονται σε προηγούμενες από 24 έως και 72 ώρες (Εικόνα 5.1). Ο τρόπος με τον οποίο κατασκευάζεται η δομή δεδομένων της εφαρμογής αυτής, βασίζεται σε μια συγκεκριμένη τεχνική, η οποία είναι μέρος της προεπεξεργασίας και θα αναλυθεί στη συνέχεια. Γενικά, οι υπερπαραμέτροι των νευρωνικών δικτύων, οι αλγόριθμοι εκπαίδευσης και βελτιστοποίησης, οι τεχνικές προεπεξεργασίας και το πλήθος των δεδομένων δεν μεταβάλλεται ανάμεσα στα πειράματα, καθώς εστιάζεται η διαφορά στην αποτελεσματικότητα των μοντέλων, σύμφωνα με τη φύση των δεδομένων

εισόδου που χρησιμοποιούνται.



Σχήμα 5.1: Παράδειγμα απευθείας πρόβλεψης χρησιμοποιώντας παρελθοντικές τιμές

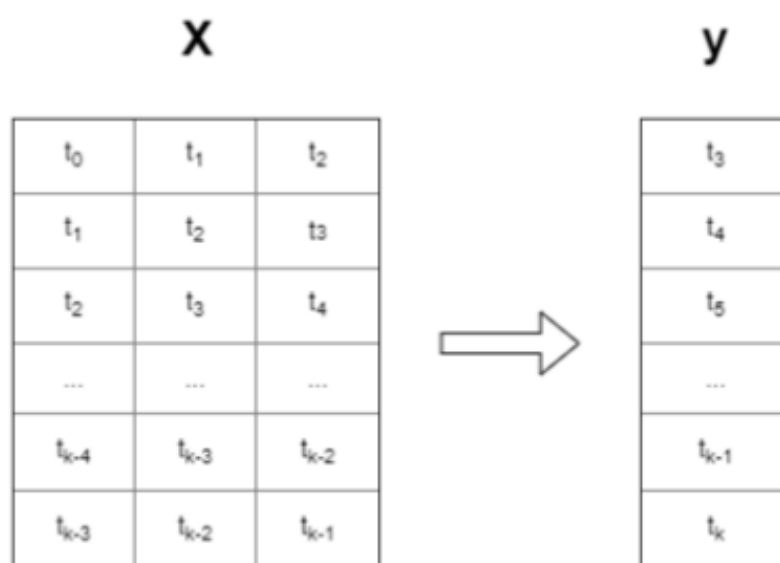
Αρχικά, το σύνολο δεδομένων που περιγράφηκε και αναλύθηκε στο Κεφάλαιο 4, χωρίζεται σε δύο υποσύνολα: ένα εκπαίδευσης (training set) και ένα ελέγχου (test set). Το ποσοστό διαχωρισμού είναι περίπου 80%-20%, καθώς χρησιμοποιούνται τα δεδομένα του 2015, 2016, 2017 και 2018 για εκπαίδευση και τα δεδομένα του 2019, μαζί με αυτά του Ιανουαρίου του 2020 για έλεγχο (Εικόνα 5.7). Έπειτα, πραγματοποιείται μια κανονικοποίηση στα δεδομένα, έτσι ώστε να διευκολυνθεί η διαδικασία της εκπαίδευσης των νευρωνικών δικτύων, λόγω της τροφοδοσίας με δεδομένα που θα βρίσκονται στην ίδια κλίμακα. Υπάρχουν διάφορα είδη κανονικοποίησης, που τα οποία ανάλογα με το είδος των δεδομένων δύναται να αυξήσουν την αποτελεσματικότητα των προβλέψεων, αλλά στην παρούσα διπλωματική εργασία επιλέχθηκε ένα από τα πιο διαδεδομένα είδη κανονικοποίησης, η κανονικοποίηση μέγιστης και ελάχιστης τιμής (MinMax scaler). Η παρακάτω εξίσωση περιγράφει αυτό το είδος κανονικοποίησης:

$$X_{\text{scaled}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (5.1)$$

όπου επιλέχθηκε οι τιμές να κανονικοποιηθούν στο διάστημα $[0, 1]$. Τονίζεται ότι οι προβλέψεις που προκύπτουν είναι κανονικοποιημένες, επομένως πρέπει να χρησιμοποιηθεί ο αντίστροφος μετασχηματισμός, έτσι ώστε να πραγματοποιηθεί η σύγκριση με τις πραγματικές τιμές του test set.

Επόμενο βήμα είναι η δημιουργία της δομής δεδομένων που περιγράφηκε παραπάνω. Για την κατασκευή της, χρησιμοποιείται η τεχνική του κινούμενου παραθύρου (sliding window), κατά την οποία κατασκευάζονται δείγματα του κάθε συνόλου από τα δε-

δομένα, που περιλαμβάνουν τα δεδομένα εισόδου ανάλογα με το μήκος του χρονικού βήματος που επιλέγεται και δεδομένα εξόδου ανάλογα με τον ορίζοντα πρόβλεψης. Στη μια δομή περιέχονται σε μια γραμμή τα δεδομένα εισόδου από το πιο απομακρισμένο χρονικό βήμα στο πιο πρόσφατο, έχοντας τον αριθμό των χαρακτηριστικών σε 2η διάσταση, ενώ στη δεύτερη δομή βρίσκονται τα δεδομένα εξόδου. Όταν ολοκληρώνεται το πέρασμα της μιας παρτίδας, η δεύτερη παρτίδα περιέχει την ίδια δομή, με τη μόνη διαφορά ότι τα δεδομένα εισόδου και εξόδου έχουν μετατοπιστεί κατά μια ώρα στο μέλλον, δηλαδή η πρώτη τιμή της γραμμής προβλέψεων περιέχεται στη γραμμή των δεδομένων εκπαίδευσης.



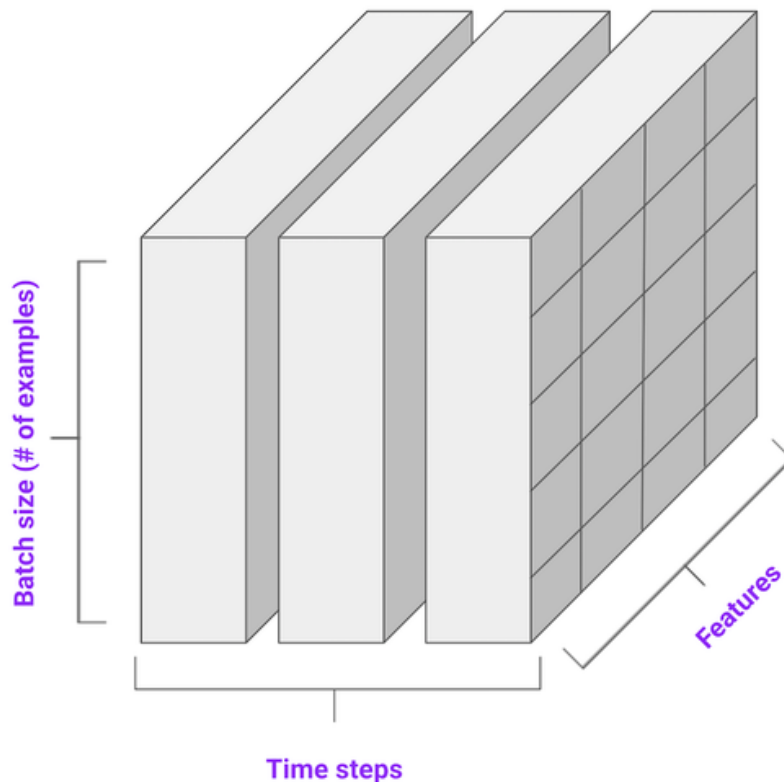
Σχήμα 5.2: Δομή δεδομένων κινούμενου παραθύρου



Σχήμα 5.3: Τεχνική κινούμενου παραθύρου

Στην ουσία, η συγκεκριμένη τεχνική επιτυγχάνει τη μετατροπή του προβλήματος της

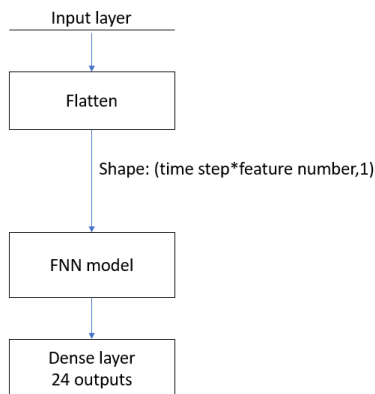
πρόβλεψης χρονοσειρών σε ένα πρόβλημα εποπτευόμενης μάθησης, καθώς χρησιμοποιείται ένας συγκεκριμένος αριθμός εισόδου, με σκοπό την πρόβλεψη ενός συγκεκριμένου αριθμού εξόδου. Επίσης, με την τεχνική αυτή, επιτυγχάνεται η ανάγνωση μεγαλύτερου αριθμού δεδομένων από το νευρωνικό δίκτυο κατά την εκπαίδευση, η λήψη και αναγνώριση των σχέσεων των δεδομένων ακολουθίας, αλλά και την προσέγγιση πραγματικών συνθηκών πρόβλεψης, καθώς τις περισσότερες φορές διατίθεται μόνο ο πιο πρόσφατος αριθμός δεδομένων για την πρόβλεψη μελλοντικών τιμών, αλλά και υπάρχει ανθυπαρσία σχετικά με ποιες είναι οι επιθυμητές μελλοντικές τιμές προς πρόβλεψη. Οι διαστάσεις της δομής των δεδομένων στα οποία έχει χρησιμοποιηθεί η παραπάνω τεχνική είναι της μορφής (αριθμός συνολικών δειγμάτων, μήκος χρονικού βήματος, αριθμός χαρακτηριστικών), σε μορφή τρισδιάστατου τανυστή (Εικόνα 5.4) και κατασκευάζεται με τη βοήθεια της βιβλιοθήκης **numpy** σε μορφή **numpy.array** και η μετατροπή σε τανυστή των διαστάσεων αυτών με τη βοήθεια της εντολής **reshape**. Οι συγκεκριμένες διαστάσεις είναι και αυτές που ζητούνται από τη βιβλιοθήκη **Keras**, για την τροφοδοσία των ανατροφοδοτούμενων νευρωνικών δικτύων.



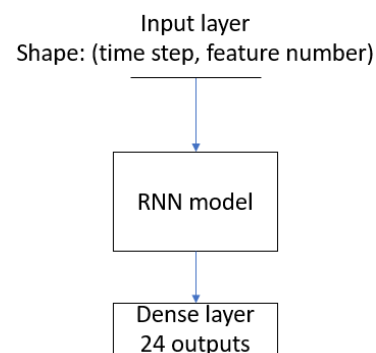
Σχήμα 5.4: Δεδομένα εισόδου στο πρόβλημα πρόβλεψης χρονοσειρών

Όσον αφορά την κατασκευή των νευρωνικών δικτύων, για όλους τους τύπους χρησιμοποιήθηκε μια προτεινόμενη από τη βιβλιογραφία αρχιτεκτονική, πάνω σε αυτόν τον τύπο δεδομένων [1]. Πιο συγκεκριμένα, χρησιμοποιήθηκε μια διάταξη τριών στρωμάτων αποτελούμενα από 80 νευρώνες το καθένα ξεχωριστά. Στην περίπτωση του **FFNN**, χρη-

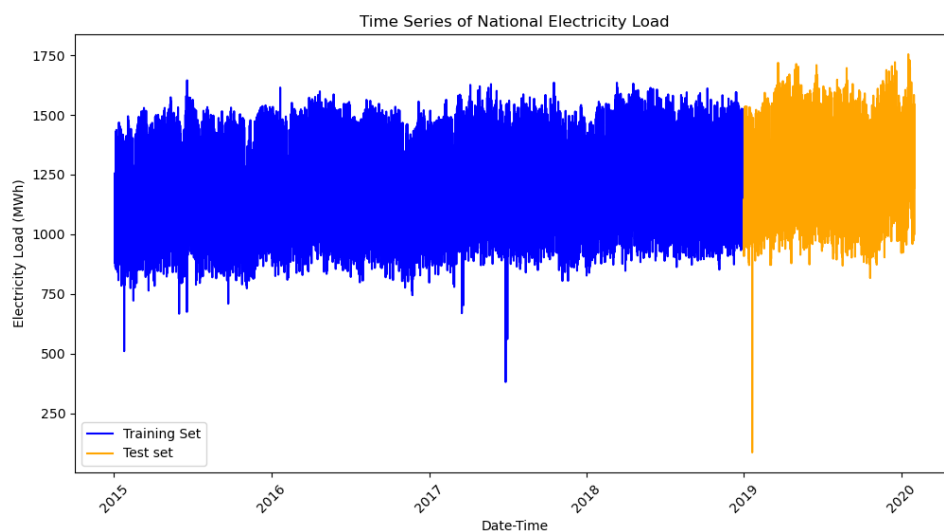
σιμοποιείται και ένα στρώμα **Flatten** πριν το πρώτο **hidden layer**, καθώς απαιτείται δύο διαστάσεων τανυστής, επομένως γίνεται η επιπεδοποίηση της τρισδιάστατης δομής. Τα δεδομένα χωρίζονται ξανά σε δείγματα διαστάσεων (αριθμός χρονικών βημάτων·αριθμός χαρακτηριστικών,1), ενώ στην περίπτωση των ανατροφοδοτούμενων σε (αριθμό χρονικών βημάτων, αριθμό χαρακτηριστικών). Και στις δύο αυτές περιπτώσεις χρησιμοποιείται ένα τελικό στρώμα τύπου **Dense**, το οποίο αποτελείται από 24 νευρώνες και δημιουργεί το **output layer**.



Σχήμα 5.5: Δομή μοντέλου FFNN



Σχήμα 5.6: Δομή μοντέλων ανατροφοδοτούμενων δικτύων



Σχήμα 5.7: Διαχωρισμός δεδομένων σε εκπαίδευσης και ελέγχου

Επιπρόσθετα, για την κατασκευή των μοντέλων νευρωνικών δικτύων, στους παρακάτω πίνακες αναγράφονται οι υπερπαράμετροι που προτείνονται για την συγκεκριμένη εφαρμογή και παραμένουν σταθερές και στα 3 πειράματα:

Υπερπαράμετρος	Τιμή
Αριθμός hidden layers	3
Αριθμός νευρώνων στο κάθε layer	80
Αλγόριθμος βελτιστοποίησης	Adam
Συνάρτηση ενεργοποίησης	<i>tanh</i>
Dropout σε κάθε layer	0.2
Βαθμός εκμάθησης	0.001
Αριθμός παρτίδων	128
Ενεργοποίηση weights (default)	Glorot uniform

Πίνακας 5.1: Υπερπαράμετροι RNNs

Υπερπαράμετρος	Τιμή
Αριθμός hidden layers	3
Αριθμός νευρώνων στο κάθε layer	80
Αλγόριθμος βελτιστοποίησης	Adam
Συνάρτηση ενεργοποίησης	ReLU
Dropout σε κάθε layer	0.2
Βαθμός εκμάθησης	0.0001
Αριθμός παρτίδων	128
Ενεργοποίηση weights (default)	Glorot uniform

Πίνακας 5.2: Υπερπαράμετροι FFNN

Όλες οι υπόλοιπες υπερπαράμετροι που χρησιμοποιήθηκαν, δεν έχουν επιλεγεί και επομένως έχουν επιλεχθεί οι προκαθορισμένες από τη βιβλιοθήκη **Keras**. Όσον αφορά τη συνάρτηση κόστους σε όλους τους τύπους νευρωνικών δικτύων επιλέγεται ως συνάρτηση λάθους η **Mean Squared Error (MSE)**. Τέλος, επιλέγεται ο αριθμός εποχών εκπαίδευσης των νευρωνικών δικτύων. Για μια εύστοχη επιλογή αυτής της υπερπαραμέτρου χρησιμοποιήθηκε η τεχνική **Early stopping**, με υπερπαραμέτρους:

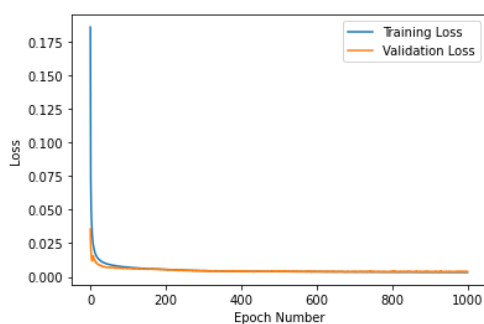
Υπερπαράμετρος	Τιμή
epochs	500
patience	200
validation_split	0.2
monitor	val_loss
mode	min
restore_best_weights	True

Πίνακας 5.3: Υπερπαράμετροι Early stopping RNNs

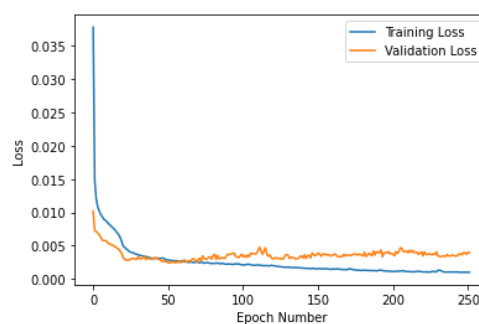
Υπερπαράμετρος	Τιμή
epochs	1000
patience	400
validation_split	0.2
monitor	val_loss
mode	min
restore_best_weights	True

Πίνακας 5.4: Υπερπαράμετροι Early stopping FFNN

Χρησιμοποιώντας το 20% των τελευταίων δεδομένων εκπαίδευσης και μήκος παρελθοντικού βήματος 24 ωρών στα αρχικά χαρακτηριστικά, ολοκληρώθηκε η εκπαίδευση του FFNN και από τα RNNs του LSTM και κατασκευάστηκαν τα παρακάτω διαγράμματα:



(α') Early stopping FFNN



(β') Early stopping LSTM

Σχήμα 5.8: Διαγράμματα σφάλματος εκπαίδευσης και επαλήθευσης - αριθμός εποχών

Στην περίπτωση (α') παρατηρείται ότι χρησιμοποιήθηκαν και οι 1000 εποχές με το σφάλμα επαλήθευσης να ταυτίζεται σχεδόν με το σφάλμα εκπαίδευσης. Βέβαια, περίπου στον αριθμό των 600 εποχών φαίνεται μια 'αποκόλληση' του σφάλματος εκπαίδευσης και με το πέρας των εποχών η σταδιακή ελάττωση του με έναν πολύ μικρό ρυθμό. Με αυτόν τον τρόπο, υποδεικνύεται ότι οι τιμές των σφαλμάτων παραμένουν σχεδόν σταθερές, δηλαδή δεν υπάρχουν τόσο μεγάλες διαφοροποιήσεις στις προβλέψεις τόσο σε άγνωστα (επαλήθευσης) όσο και γνωστά για το νευρωνικό δίκτυο δεδομένα. Επομένως επιλέχθηκε μια τιμή εποχών ανάμεσα στο 600 και το 1000 με αυτήν να είναι οι 750 εποχές. Στην περίπτωση (β') παρατηρείται ότι η τεχνική ολοκλήρωσε την εκπαίδευση λίγο μετά τις 250 εποχές, επομένως βρέθηκε το ελάχιστο σφάλμα επαλήθευσης περίπου λίγο πάνω από τις 50 εποχές, γεγονός που παρατηρείται και από το διάγραμμα. Βέβαια, από εκείνη τη στιγμή και έπειτα το σφάλμα εκπαίδευσης εξακολουθεί να μειώνεται σταδιακά, ενώ το σφάλμα επαλήθευσης ταλαντώνεται γύρω από τις τιμές 0.003 με 0.004 περίπου, δείχνοντας ότι παραμένει σε γενικές γραμμές σταθερό. Βέβαια, ο αριθμός των 50 εποχών που βρέθηκε το βέλτιστο σφάλμα, θεωρείται μη ικανοποιητικός, επομένως επιλέγεται ο αριθμός 100, στον οποίο υπάρχει μεγαλύτερη βεβαιότητα

τόσο για τις προβλέψεις σε γνωστά δεδομένα, όσο και για τις προβλέψεις σε άγνωστα δεδομένα και με όλα τα διαφορετικά παρελθοντικά βήματα που θα χρησιμοποιηθούν. Δεν επιλέγεται μεγαλύτερος αριθμός εποχών από αυτόν, καθώς είναι αρκετά πιθανή η εμφάνιση **overfitting**, καθώς όπως φαίνεται και από το διάγραμμα το σφάλμα επαλήθευσης εμφανίζει σταδιακή αύξηση στις τιμές του με την αύξηση του αριθμού εποχών. Επομένως, για όλα τα διαφορετικά χρονικά βήματα χρησιμοποιούνται 750 εποχές, όσον αφορά το **FFNN** και 100 εποχές, όσον αφορά τους διαφορετικούς τύπους **RNNs**, χρησιμοποιώντας ολόκληρο το σύνολο δεδομένων εκπαίδευσης, στη συγκεκριμένη εφαρμογή.

Σημειώνεται πως, οι δομές του συνόλου δεδομένων ελέγχου είναι παρόμοιες με αυτές των δεδομένων εκπαίδευσης, δηλαδή χρησιμοποιείται ένας ταχυστής που περιέχει τα **inputs**, που αναφέρονται στα δεδομένα ελέγχου και έναν πίνακα που περιέχει στις γραμμές του τις 24 πραγματικές τιμές, οι οποίες συγκρίνονται με τις προβλέψεις των μοντέλων. Οι ακριβείς διαστάσεις των δομών αυτών αναφέρονται στην επόμενη ενότητα. Τέλος, μετά την ολοκλήρωση της εκπαίδευσης ενός μοντέλου πραγματοποιείται επανεκκίνηση του πυρήνα (**kernel**) και η διαγραφή των προϋπάρχοντων μεταβλητών από το περιβάλλον υλοποίησης, **spyder**.

5.3 Αποτελέσματα

Τα αποτελέσματα των προβλεπτικών μοντέλων προκύπτουν από τη χρήση των αποτιμητών προβλέψεων που αναφέρθηκαν στην ενότητα 3 συγκρίνοντας τις προβλέψεις με τις πραγματικές τιμές που περιέχονται στο σύνολο δεδομένων ελέγχου. Τόσο οι προβλέψεις όσο και οι πραγματικές τιμές, σε όλα τα πειράματα, περιέχονται σε πίνακες διαστάσεων (9481,24) και οι αποτιμητές προκύπτουν από το μέσο όρο των σφαλμάτων όλων των γραμμών. Η κάθε γραμμή περιέχει 24 τιμές, ενώ η κάθε στήλη των συνολικό αριθμό των ωρών που περιέχονται στο **Test set** μετατοπισμένο μια ώρα στο μέλλον ανά γραμμή.

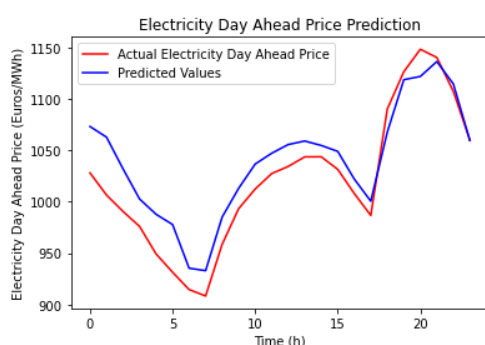
5.3.1 Απλοϊκή μέθοδος

Κατά την απλοϊκή μέθοδο γίνεται η θεώρηση ότι η τιμή του ηλεκτρικού φορτίου προς πρόβλεψη θα ισούται με την τιμή του ηλεκτρικού φορτίου την προηγούμενη εβδομάδα. Το μοντέλο αυτό κατασκευάστηκε με τη βοήθεια της εντολής **iloc** της βιβλιοθήκης **pandas**. Τα αποτελέσματα που προκύπτουν είναι:

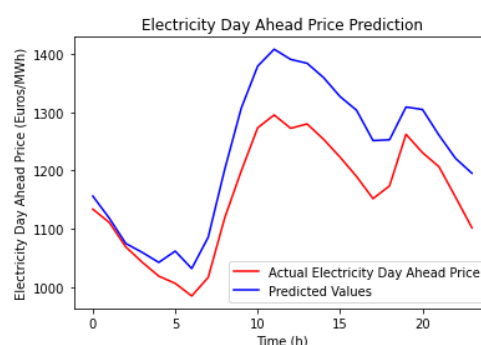
Μοντέλο	RMSE	MAE	MAPE	R^2
Απλοϊκή μέθοδος	103,2491	70,83966	0,059424	0,703194

Πίνακας 5.5: Αποτιμητές προβλέψεων για το σύνολο ελέγχου με την απλοϊκή μέθοδο

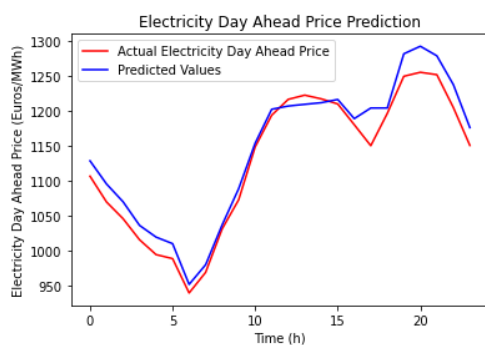
Όπως αναφέρθηκε και προηγουμένως η πρόβλεψη με τη συγκεκριμένη μέθοδο είναι ικανοποιητική, αλλά δεν μοιάζει δύσκολο να νικηθεί από κάποια αποτελεσματικότερη. Παρατηρώντας τα αποτελέσματα της έρευνας [1], αναμένονται αξιοσημείωτες διαφορές στους αποτιμητές των προβλέψεων των μοντέλων νευρωνικών δικτύων συγκριτικά με αυτές της απλοϊκής μεθόδου. Με $MAPE = 5.94\%$ και $R^2 = 0.7$, το συγκεκριμένο μοντέλο έχει μια καλή απόδοση και ταιριάζει σε γενικές γραμμές με τα δεδομένα του συνόλου ελέγχου. Τα παρακάτω διαγράμματα απεικονίζουν κάποιες προβλέψεις από κανονικές ημέρες, απλά σαββατοκύριακα και εορταστικές ημέρες (κινούμενες και σταθερές).



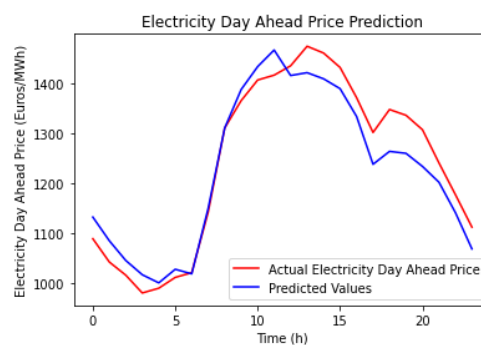
(α') 1 Ιανουαρίου (Σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)



(γ') 7 Απριλίου (Κυριακή)



(δ') 16 Οκτωβρίου (Καθημερινή)

Σχήμα 5.9: Ενδεικτικές προβλέψεις απλοϊκής μεθόδου

Τα ημερήσια μοτίβα εντοπίζονται αποτελεσματικά, με τη μεγαλύτερη απόκλιση στις κινούμενες αλλά και σταθερές αργίες, με εξαιρέσεις στις περιπτώσεις που τυγχάνει δύο αργίες να έχουν, χρονικά, απόσταση μιας εβδομάδας. Στο μεγαλύτερο μέρος των τιμών φορτίου, δεν εμφανίζονται τόσο μεγάλες διαφορές στην τάση σε διάρκεια μιας εβδομάδας άρα και δεν αναμένονται μεγάλες διαφορές όσον αφορά το ύψος της καμπύλης προβλέψεων.

5.3.2 1ο πείραμα

Σε αυτό το πείραμα χρησιμοποιήθηκαν όλα τα χαρακτηριστικά που περιέχονται στο σύνολο δεδομένων εξ αρχής δοκιμάζοντας 5 διαφορετικά μοντέλα νευρωνικών δικτύων με τις παραμέτρους που αναφέρθηκαν στην προηγούμενη ενότητα. Η δομή που χρησιμοποιείται για την εκπαίδευση έχει διαστάσεις:

- (34968,24,16), δομή δεδομένων εισόδου για *lookback_period* = 24 και (34968,24,1), δομή δεδομένων εξόδου
- (34944,48,16), δομή δεδομένων εισόδου για *lookback_period* = 48 και (34944,24,1), δομή δεδομένων εξόδου
- (34920,72,16), δομή δεδομένων εισόδου για *lookback_period* = 72 και (34920,24,1), δομή δεδομένων εξόδου

Η δομή των δεδομένων εισόδου ελέγχου έχει διαστάσεις (9481,24,16), (9481,48,16) και (9481,72,16) αντίστοιχα με τις διαστάσεις των πραγματικών τιμών να είναι ίδιες με αυτές που προαναφέρθηκαν στην αρχή της ενότητας.

Τα αποτελέσματα παρουσιάζονται στους παρακάτω πίνακες για κάθε μήκος παρελθοντικού χρονικού βήματος (*lookback period*) ξεχωριστά, καθώς και τα διαγράμματα των μοντέλων στα μήκη κατά τα οποία έδωσαν τις πιο αποτελεσματικές προβλέψεις με κύριο γνώμονα το μέσο απόλυτο ποσοστιαίο σφάλμα.

Για *lookback* = 24:

Μοντέλο	RMSE	MAE	MAPE	R^2	Χρόνος εκπαίδευσης (s)
LSTM	78,70842682	54,24206604	0,045136256	0,827500615	2101,6
GRU	77,84258544	53,73543615	0,044882811	0,831270386	1977
BiLSTM	79,70957852	54,03767555	0,045265559	0,823079285	9200,2
BiGRU	82,99863711	57,33486313	0,047422569	0,808177931	4843,9
FFNN	95,06142303	72,23373776	0,058415729	0,748385774	606,64

Πίνακας 5.6: Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 1ου πειράματος με *lookback* = 24

Για $lookback = 48$:

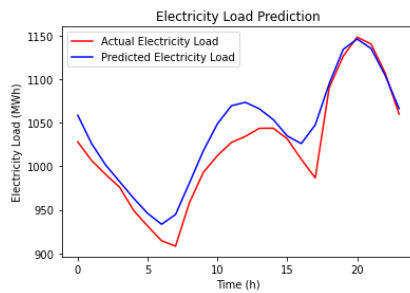
Μοντέλο	RMSE	MAE	MAPE	R^2	Χρόνος εκπαίδευσης (s)
LSTM	78,6154034	54,27314207	0,045190527	0,827908794	3937,1
GRU	82,06868607	57,30066007	0,047866615	0,812457163	3239,8
BiLSTM	79,51297493	54,75537916	0,045947587	0,823951472	19865
BiGRU	82,71750232	57,50250374	0,047741151	0,809473845	15090
FFNN	94,72523475	70,2143162	0,058043224	0,75015715	476,48

Πίνακας 5.7: Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 1ου πειράματος με $lookback = 48$

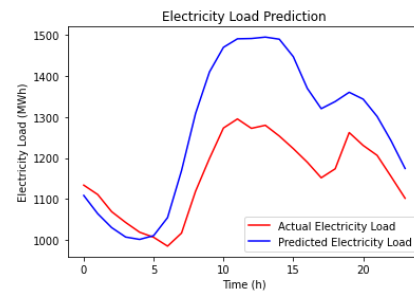
Για $lookback = 72$:

Μοντέλο	RMSE	MAE	MAPE	R^2	Χρόνος εκπαίδευσης (s)
LSTM	77,95583589	53,31929522	0,044758653	0,830782483	8204,1
GRU	86,20633196	59,5850478	0,049465226	0,793068279	6487,8
BiLSTM	82,43305837	58,25843368	0,048612062	0,810784913	43081
BiGRU	80,41687878	56,29784742	0,046749657	0,819931315	26669
FFNN	109,6604727	85,46252836	0,068346641	0,66517093	676,41

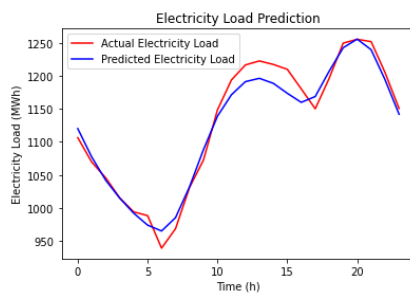
Πίνακας 5.8: Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 1ου πειράματος με $lookback = 72$



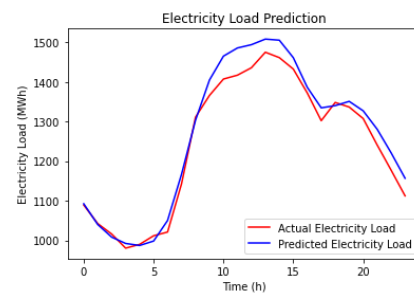
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

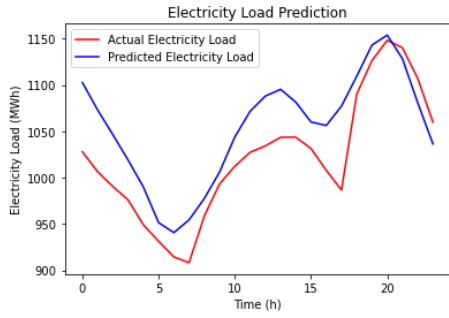


(γ') 7 Απριλίου (Κυριακή)

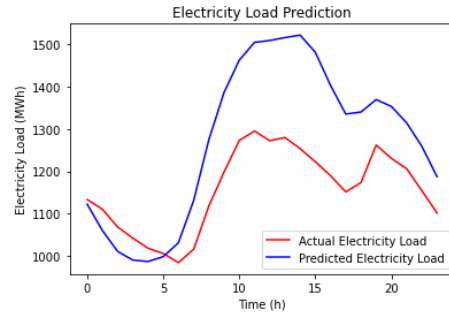


(δ') 16 Οκτωβρίου (Καθημερινή)

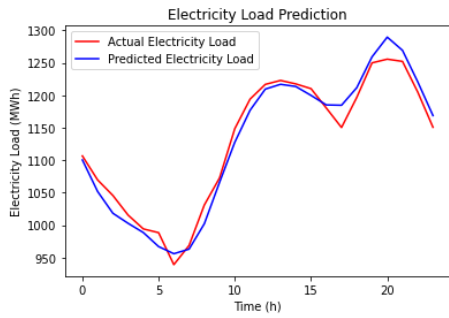
Σχήμα 5.10: Ενδεικτικές προβλέψεις LSTM για $lookback = 72$



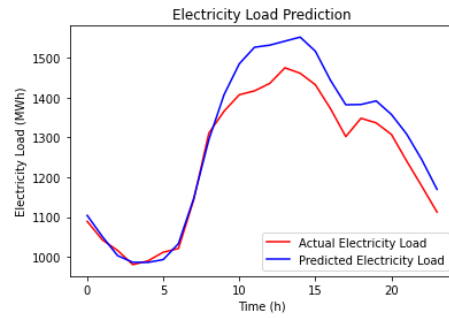
(α') 1 Ιανουαρίου (Σταθερή αργία)



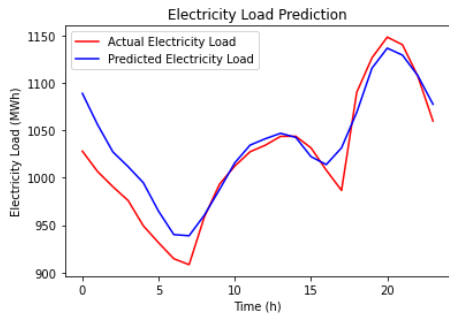
(β') 2 Μαρτίου (Κινητή αργία)



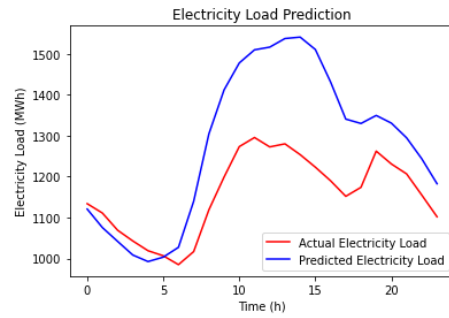
(γ') 7 Απριλίου (Κυριακή)



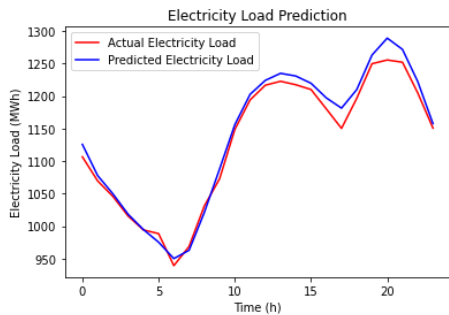
(δ') 16 Οκτωβρίου (Καθημερινή)

Σχήμα 5.11: Ενδεικτικές προβλέψεις GRU για $lookback = 24$ 

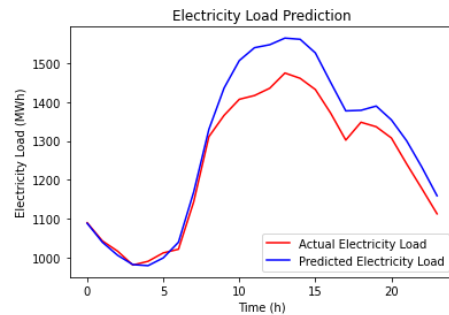
(α') 1 Ιανουαρίου (Σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

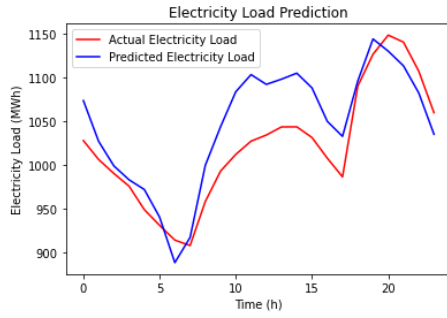


(γ') 7 Απριλίου (Κυριακή)

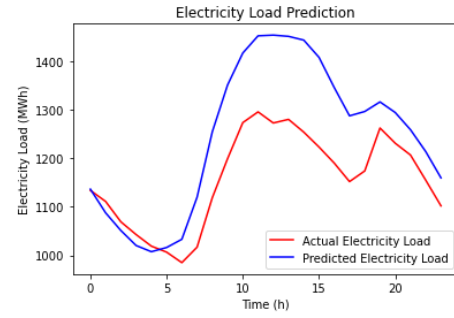


(δ') 16 Οκτωβρίου (Καθημερινή)

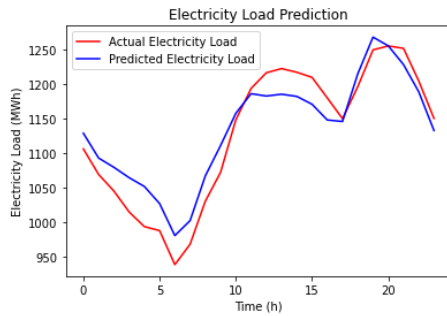
Σχήμα 5.12: Ενδεικτικές προβλέψεις Bidirectional LSTM για $lookback = 24$



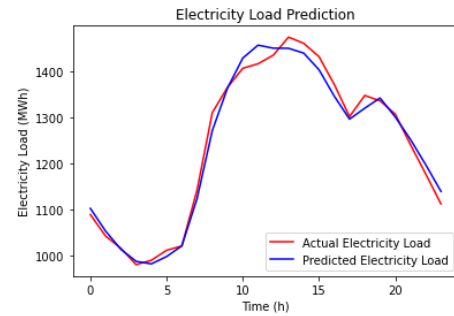
(α') 1 Ιανουαρίου (Σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

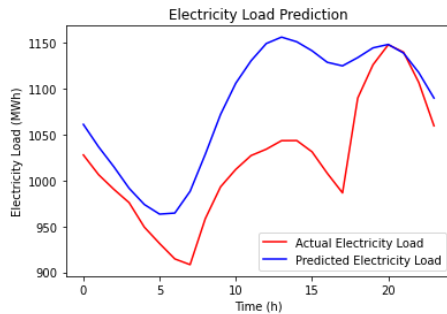


(γ') 7 Απριλίου (Κυριακή)

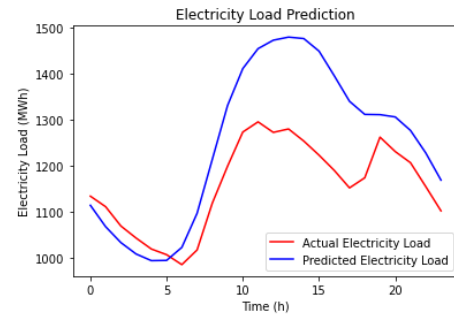


(δ') 16 Οκτωβρίου (Καθημερινή)

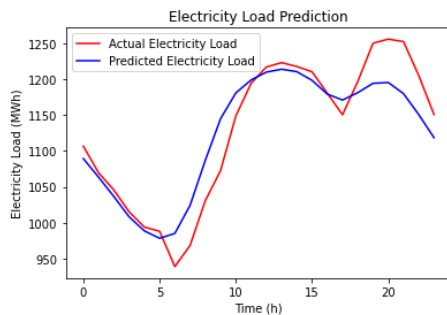
Σχήμα 5.13: Ενδεικτικές προβλέψεις Bidirectional GRU για $lookback = 72$



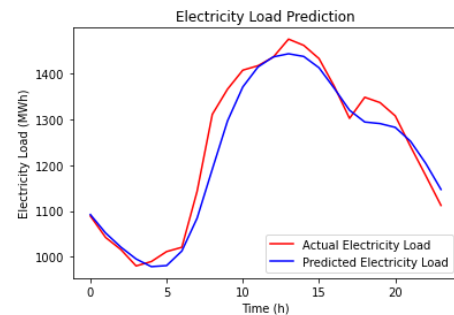
(α') 1 Ιανουαρίου (Σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)



(γ') 7 Απριλίου (Κυριακή)



(δ') 16 Οκτωβρίου (Καθημερινή)

Σχήμα 5.14: Ενδεικτικές προβλέψεις FFNN για $lookback = 48$

Τα παραπάνω διαγράμματα των ενδεικτικών προβλέψεων είναι ένα πολύ καλό δείγμα

για την απόδοση των μοντέλων σε διαφορετικές περιστάσεις μέσα στη διάρκεια ενός έτους. Αρχικά, τονίζεται ότι όλα τα παραπάνω μοντέλα νευρωνικών δικτύων ξεπερνούν το αρχικό μοντέλο βάσης σε αποτελεσματικότητα, καθώς οι βέλτιστες προβλέψεις, έχοντας ως γνώμονα το **MAPE**, δίνονται από το **LSTM** χρησιμοποιώντας ως **inputs**, **lookback=72** με σφάλμα περίπου ίσο με 4,47%. Αξίζει να σημειωθεί ότι σε όλες τις περιπτώσεις έχουν εντοπισθεί αρκετά αποτελεσματικά τα μοτίβα των καθημερινών ημερών, καθώς και ένα πολύ μεγάλο ποσοστό των σαββατοκυριάκων, με την αδυναμία στις προβλέψεις να παρατηρείται ως επί το πλείστον στις σταθερές και κινητές αργίες. Πιο συγκεκριμένα, παρατηρείται ότι η χειρότερη απόδοση σημειώθηκε από το **FFNN**, γεγονός που είναι αναμενόμενο, καθώς δεν διαθέτει τη δυνατότητα συγκράτησης πληροφορίας από προηγούμενα χρονικά βήματα στη μνήμη του και με αύξηση της αστοχίας στις προβλέψεις κατά τη χρήση του μεγαλύτερου αριθμού **inputs**. Όσον αφορά το **LSTM**, η ικανότητα του για αποθήκευση πληροφορίας από μεγάλα χρονικά βήματα, σημειώνει τις πιο αποτελεσματικές του προβλέψεις για το μεγαλύτερο αριθμό χρονικών βημάτων, επιτυγχάνοντας την εκμετάλλευση όλης της επιπρόσθετης πληροφορίας. Βέβαια, η διαφορά με τα προηγούμενα **lookback periods** δεν είναι σημαντικά αισθητή. Σε συνέχεια, το μοντέλο **GRU** σημειώνει τις αποδοτικότερες του προβλέψεις με χρήση του μικρότερου χρονικού βήματος ως **inputs**, άλλο ένα αναμενόμενο γεγονός, λόγω της μεγαλύτερης του απλότητας σε σχέση με τα υπόλοιπα, άρα και αποθηκεύεται και επεξεργάζεται με μεγαλύτερη ακρίβεια η μικρότερη ποσότητα πληροφορίας. Επιπρόσθετα, οι περιπτώσεις των **LSTM** και **GRU** δύο κατευθύνσεων σημειώνουν χειρότερη απόδοση συγκριτικά με αυτά των μονών, καταλήγοντας στο συμπέρασμα ότι ταιριάζουν ελαφρώς λιγότερο με τη συγκεκριμένη εφαρμογή. Το **Bidirectional LSTM** σημειώνει τον βέλτιστο αποτιμητή για το χαμηλότερο χρονικό βήμα, ενώ το **Bidirectional GRU** για το υψηλότερο, έχοντας την αντίθετη ακριβώς ανταπόκριση από τα αντίστοιχα μοντέλα μιας κατεύθυνσης. Τέλος, δύο ακόμα σημεία που αξίζει να τονιστούν είναι ότι σε γενικές γραμμές όλα τα μοντέλα νευρωνικών δικτύων ταιριάζουν με τα δεδομένα του συνόλου ελέγχου σημειώνοντας αρκετά υψηλές τιμές R^2 , με τη χειρότερη να εμφανίζεται στα μοντέλα **FFNN** και ότι ο χρόνος εκπαίδευσης, άρα και το υπολογιστικό κόστος, αυξάνεται με την αύξηση της πληροφορίας εισόδου που δέχεται το κάθε νευρωνικό δίκτυο.

5.3.3 2ο πείραμα

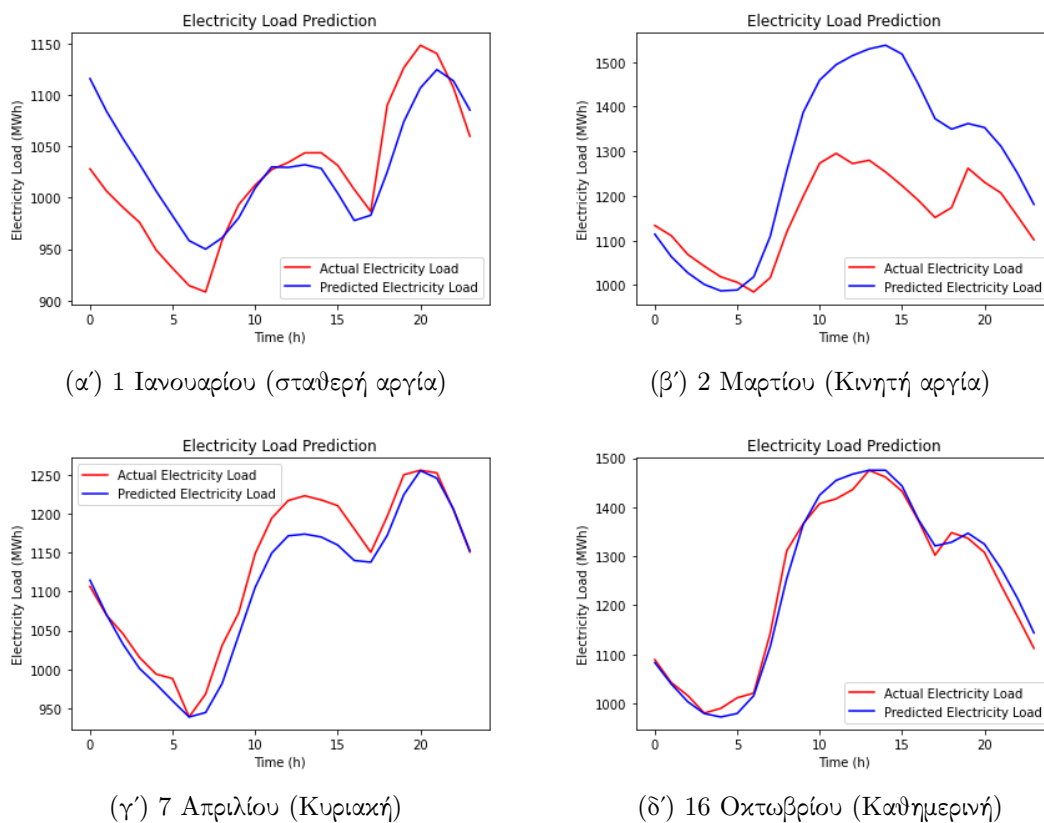
Στην περίπτωση του δεύτερου πειράματος επαναλαμβάνεται η κατασκευή των παραπάνω τύπων νευρωνικών δικτύων, αυτή τη φορά, όμως, χρησιμοποιούνται ως δεδομένα εισόδου τα επιλεγμένα πιο σημαντικά χαρακτηριστικά. Το κριτήριο επιλογής είναι ο συντελεστής **Pearson correlation**, θεωρώντας ως ένα σημαντικό για την εφαρμογή χαρακτηριστικό, αυτό που εμφανίζει συντελεστή μεγαλύτερο από την απόλυτη τιμή του 0.1. Επομένως, τα χαρακτηριστικά που χρησιμοποιούνται στο 2ο πείραμα είναι τα εξής:

T2M_toc, T2M_dav, T2M_san, W2M_san, TQL_san, holiday και Holiday_ID. Αυτά τα χαρακτηριστικά συμπληρώνουν τις ιστορικές τιμές του ωριαίου ηλεκτρικού φορτίου. Η δοκιμή των μοντέλων, αυτή τη φορά πραγματοποιείται για *lookback* = 48, για του οποίου την επιλογή συμπεριλήφθηκαν παράγοντες αποτελεσματικότητας σε όλα τα μοντέλα, αλλά και υπολογιστικούς κόστους. Επομένως, χρησιμοποιώντας αυτή τη φορά δεδομένα εισόδου εκπαίδευσης διαστάσεων (34944,48,8) και δεδομένα εισόδου ελέγχου (9481,48,8), τα αποτελέσματα που λαμβάνονται είναι:

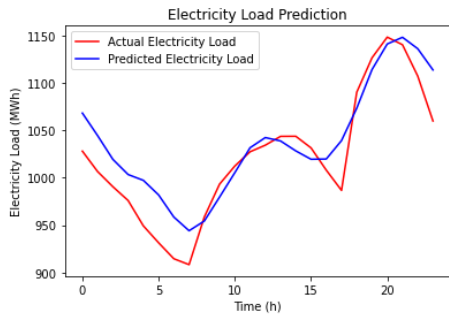
Μοντέλο	RMSE	MAE	MAPE	R^2	Χρόνος εκπαίδευσης (s)
LSTM	75,34494	52,60372	0,043764	0,841926	3694,933
GRU	75,4414	51,83262	0,042675	0,841519	3362,906
BiLSTM	76,28361	52,29342	0,043562	0,837958	19789,6
BiGRU	76,7902	53,34946	0,044529	0,835802	13429,46
FFNN	84,46448	62,16493	0,051338	0,80135	431,2924

Πίνακας 5.9: Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 2ου πειράματος

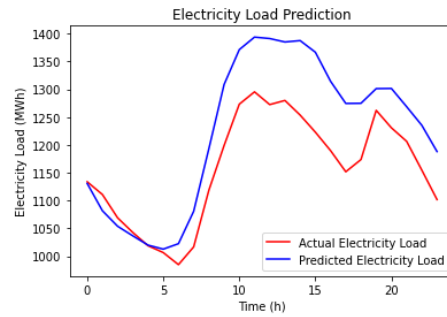
Τα διαγράμματα των ίδιων ενδεικτικών ημερών για το κάθε μοντέλο είναι:



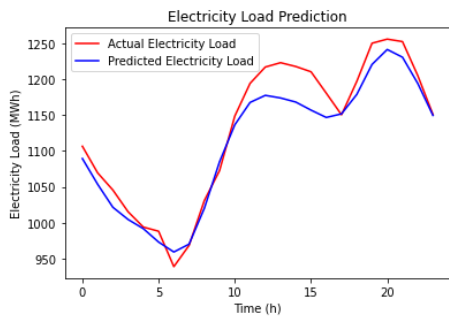
Σχήμα 5.15: Ενδεικτικές προβλέψεις LSTM για 8 σημαντικότερα χαρακτηριστικά



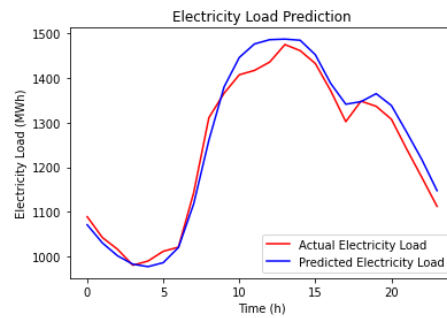
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

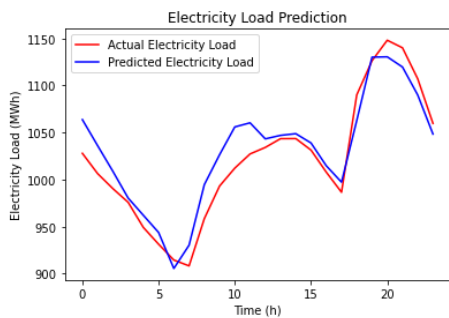


(γ') 7 Απριλίου (Κυριακή)

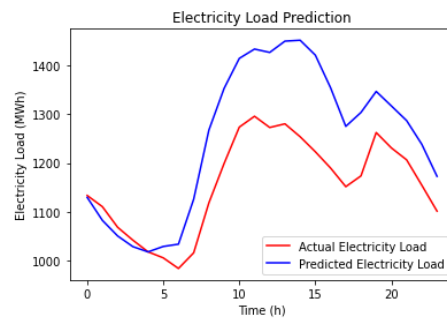


(δ') 16 Οκτωβρίου (Καθημερινή)

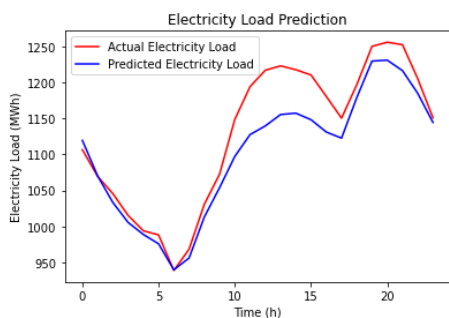
Σχήμα 5.16: Ενδεικτικές προβλέψεις GRU για 8 σημαντικότερα χαρακτηριστικά



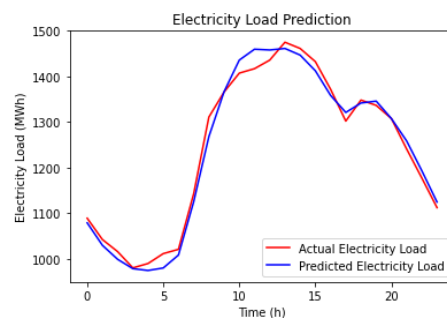
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

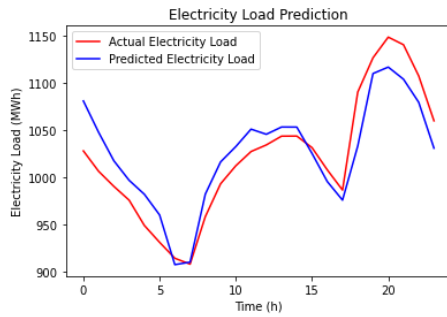


(γ') 7 Απριλίου (Κυριακή)

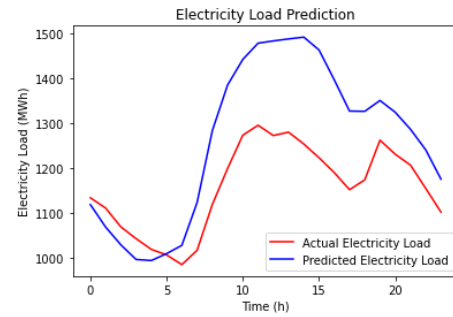


(δ') 16 Οκτωβρίου (Καθημερινή)

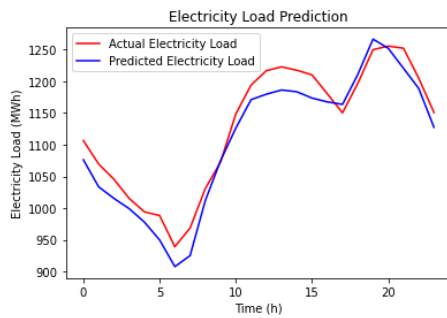
Σχήμα 5.17: Ενδεικτικές προβλέψεις Bidirectional LSTM για 8 σημαντικότερα χαρακτηριστικά



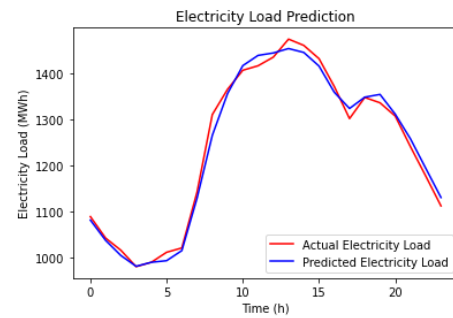
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

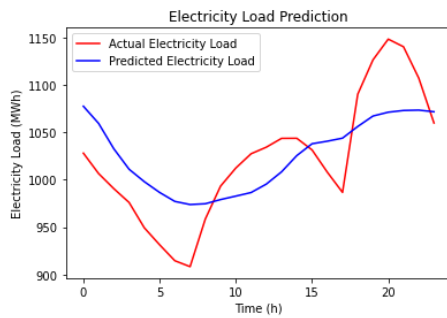


(γ') 7 Απριλίου (Κυριακή)

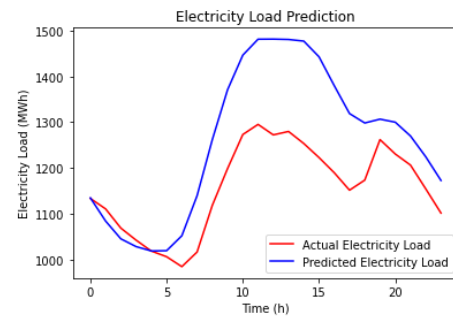


(δ') 16 Οκτωβρίου (Καθημερινή)

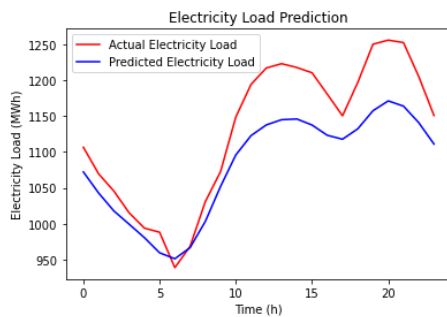
Σχήμα 5.18: Ενδεικτικές προβλέψεις Bidirectional GRU για 8 σημαντικότερα χαρακτηριστικά



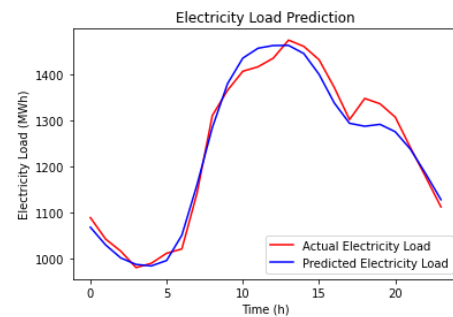
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)



(γ') 7 Απριλίου (Κυριακή)



(δ') 16 Οκτωβρίου (Καθημερινή)

Σχήμα 5.19: Ενδεικτικές προβλέψεις FFNN για 8 σημαντικότερα χαρακτηριστικά

Η χρήση των πιο σημαντικών δεδομένων οδήγησε στη βελτίωση των αποτελεσμάτων όλων των μοντέλων, με τη μεγαλύτερη να σημειώνεται στο **FFNN**. Το γεγονός αυτό, οφείλεται στην τροφοδοσία του μοντέλου με μικρότερο αριθμό δεδομένων, αλλά σημαντικών, διευκολύνοντας έτσι το έργο της εκπαίδευσης του μέσω του αλγορίθμου **back-propagation**. Από το προηγούμενο πείραμα έγινε αντιληπτό ότι όσο μεγαλύτερη ήταν η τροφοδοσία τόσο μεγαλύτερη ήταν και η δυσκολία επεξεργασίας της πληροφορίας. Βέβαια, λόγω της ανικανότητας αποθήκευσης της πληροφορίας, υστερεί και σε αυτή την περίπτωση σε σχέση με τους υπόλοιπους τύπους νευρωνικών δικτύων. Τώρα, όσον αφορά τα ανατροφοδοτούμενα μοντέλα, το βέλτιστο **MAPE** σημειώνεται από το **GRU**, με τιμή το 4.27%, αλλά οι διαφορές με τα υπόλοιπα είναι αρκετά μικρές. Ελαττώνοντας τον αριθμό των χαρακτηριστικών εισόδου, ελαττώνεται και ο πιθανός εισερχόμενος θόρυβος σε αυτά, που πολλές φορές μπορεί να προκαλέσει τον αποσυντονισμό τους κατά την προσπάθεια εντοπισμού των μοτίβων. Σε όλες τις περιπτώσεις παρατηρείται μια μικρή αύξηση της αποτελεσματικότητας στις προβλέψεις των καθημερινών και των σαββατοκυριακών, καθώς ήταν αρκετά μεγάλη και στην προηγούμενη περίπτωση, αλλά σημειώθηκαν και μικρές βελτιώσεις κατά την πρόβλεψη των αργιών, εξαιρώντας την περίπτωση του **FFNN** που στο συγκεκριμένο παράδειγμα αδυνατεί να εντοπίσει κάποιο μοτίβο, με τις κινητές να αποτελούν ακόμη τη μεγαλύτερη πρόκληση. Τέλος, όσον αφορά το χρόνο εκπαίδευσης, φαίνεται να είναι περίπου ο ίδιος ανεξαρτήτως χρήσης λιγότερης πληροφορίας.

5.3.4 3ο πείραμα

Το 3ο και τελευταίο πείραμα που διεξάγεται στην παρούσα διπλωματική εργασία έχει ως σκοπό τη βελτίωση των αποτελεσμάτων των προηγούμενων πειραμάτων, χωρίς να πραγματοποιηθεί κάποια μεταβολή στις υπερπαραμέτρους που χρησιμοποιήθηκαν, αλλά μόνο με μεταβολές στο χειρισμό των χαρακτηριστικών του συνόλου δεδομένων. Πιο συγκεκριμένα, πραγματοποιείται ο καθαρισμός των έκτοπων τιμών που εντοπίστηκαν κατά την ανάλυση δεδομένων του Κεφαλαίου 4, με την αντικατάσταση αυτών που βρίσκονται στο σύνολο δεδομένων εκπαίδευσης με τις τιμές που δίνει η γραμμική παρεμβολή των φορτίων της προηγούμενης και της επόμενης ώρας. Έπειτα, εντάσσονται επιπλέον χαρακτηριστικά, τα οποία προκύπτουν από τα ήδη υπάρχοντα, τα οποία δύναται να βοηθήσουν τα μοντέλα νευρωνικών δικτύων να εντοπίσουν αποτελεσματικότερα τα μοτίβα. Οι ενέργειες που αναφέρθηκαν ονομάζονται **feature engineering**. Τα χαρακτηριστικά αυτά είναι:

- **hour_of_the_day**: Δηλώνει την ώρα κατά τη διάρκεια της ημέρας. Χαρακτηριστικό πολύ σημαντικό για τον εντοπισμό των αυξομειώσεων του φορτίου ανάλογα με την ώρα της ημέρας.

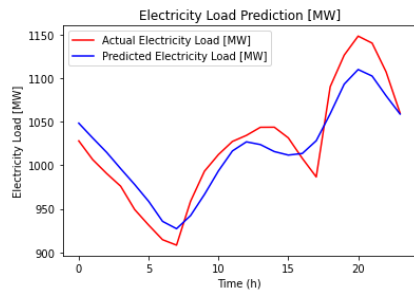
- **day_of_week**: Δηλώνει την ημέρα της εβδομάδας. Χαρακτηριστικό σημαντικό για τον εντοπισμό των διαφοροποιήσεων των μοτίβων του φορτίου ανάλογα με τη μέρα της εβδομάδας
- **month**: Δηλώνει το μήνα του έτους. Χαρακτηριστικό που βοηθάει και στην ετήσια εποχικότητα, καθώς σε κάθε μήνα καταναλώνεται διαφορετική ποσότητα ηλεκτρικού φορτίου.
- **weekend**: Διαδική μεταβλητή που δηλώνει τα σαββατοκύριακα (0: για καθημερινή και 1: για σαββατοκύριακο)
- **prev_24h_avg_load**: Υπολογίζεται χρησιμοποιώντας ένα παράθυρο 24 ωρών και υπολογίζοντας το μέσο όρο του. Το παράθυρο αυτό βρίσκεται 24 ώρες πίσω από την κάθε τιμή του φορτίου. Βοηθάει τόσο στον εντοπισμό πιθανής τάσης αλλά και ομαλοποίησης των δεδομένων φορτίου.
- **prev_24h_lagged_load**: Δηλώνει την τιμή του ηλεκτρικού φορτίου ακριβώς 24 ώρες πριν. Επιλέγεται λόγω της μεγάλης αυτοσυσχέτισης που υπάρχει με τις 24 καθυστερήσεις.
- **prev_168h_lagged_load**: Δηλώνει την τιμή του ηλεκτρικού φορτίου ακριβώς 168 ώρες ή μια εβδομάδα πριν. Επιλέγεται λόγω της μεγάλης αυτοσυσχέτισης που υπάρχει με τις 168 καθυστερήσεις.

Σε αυτή την περίπτωση η δομή των δεδομένων εκπαίδευσης ελαττώνεται σε γραμμές και αυξάνεται σε στήλες. Για την περίπτωση του 1ου πειράματος (όλα τα δεδομένα) η δομή των δεδομένων εισόδου έχει διαστάσεις (34776,48,15) και των δεδομένων ελέγχου (9481,48,15), ενώ στην άλλη περίπτωση (σημαντικά δεδομένα) (34776,48,23) και (9481,48,23) αντίστοιχα. Η μείωση στις γραμμές εντοπίζεται λόγω της ρίψης των κενών τιμών που προκύπτουν στις αρχικές τιμές του συνόλου δεδομένων, για την κατασκευή και υπολογισμό των νέων χαρακτηριστικών. Τα αποτελέσματα βάση των αποτιμητών προβλέψεων για την εφαρμογή **feature engineering** στα πιο σημαντικά χαρακτηριστικά είναι:

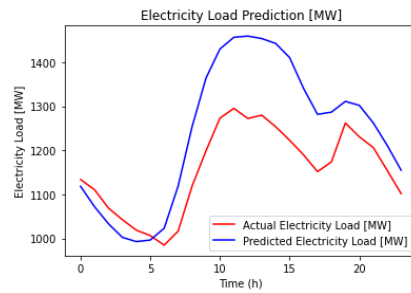
Μοντέλο	RMSE	MAE	MAPE	R^2	Χρόνος εκπαίδευσης (s)
LSTM	67,97855	45,08669	0,038739	0,871324	4221,516
GRU	65,78258	44,72692	0,03844	0,879502	4183,638
BiLSTM	67,02285	45,4116	0,038663	0,874915	20093,56
BiGRU	68,63993	47,54881	0,040086	0,868805	12842,98
FFNN	68,68909	49,68587	0,041454	0,868627	755,7886

Πίνακας 5.10: Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 3ου πειράματος, χρησιμοποιώντας τα σημαντικά χαρακτηριστικά

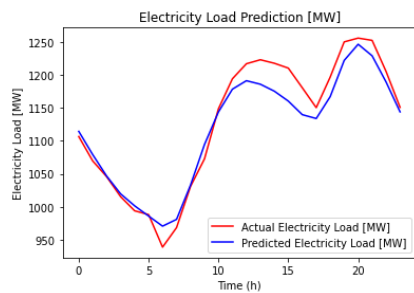
Τα αντίστοιχα ενδεικτικά διαγράμματα για αυτή την περίπτωση είναι:



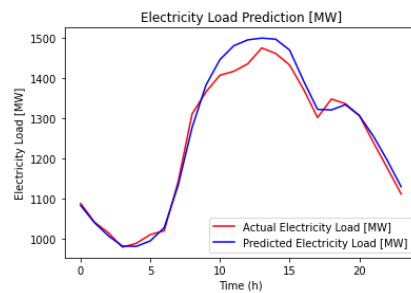
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

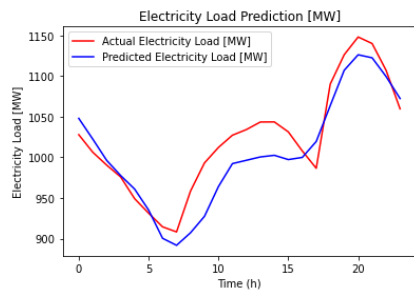


(γ') 7 Απριλίου (Κυριακή)

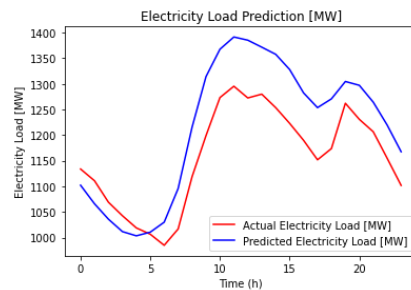


(δ') 16 Οκτωβρίου (Καθημερινή)

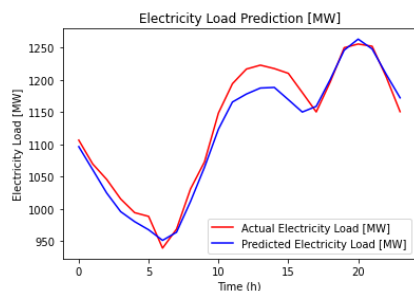
Σχήμα 5.20: Ενδεικτικές προβλέψεις LSTM για feature engineering στα σημαντικά χαρακτηριστικά



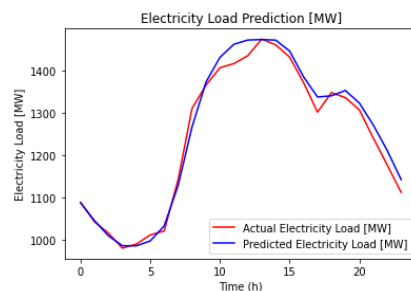
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

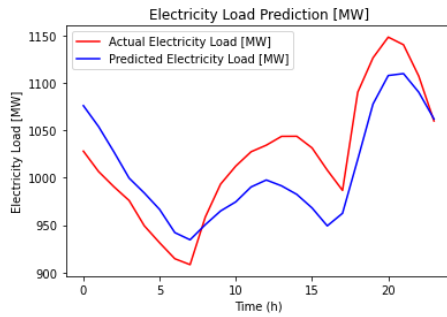


(γ') 7 Απριλίου (Κυριακή)

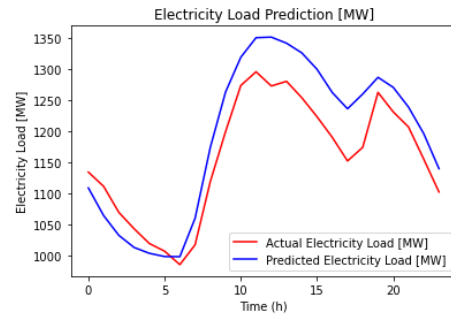


(δ') 16 Οκτωβρίου (Καθημερινή)

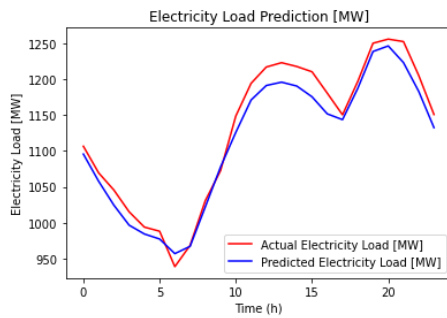
Σχήμα 5.21: Ενδεικτικές προβλέψεις GRU για feature engineering στα σημαντικά χαρακτηριστικά



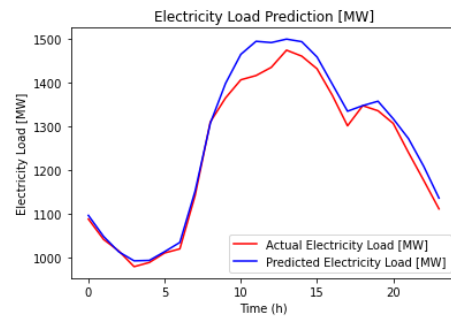
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

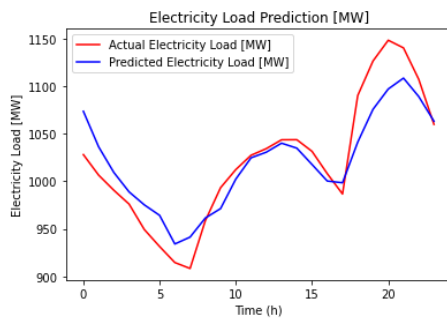


(γ') 7 Απριλίου (Κυριακή)

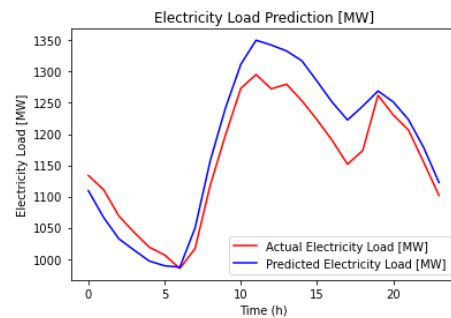


(δ') 16 Οκτωβρίου (Καθημερινή)

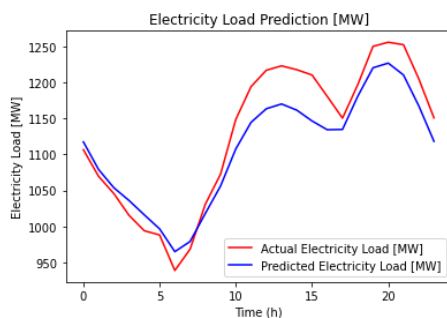
Σχήμα 5.22: Ενδεικτικές προβλέψεις Bidirectional LSTM για feature engineering στα σημαντικά χαρακτηριστικά



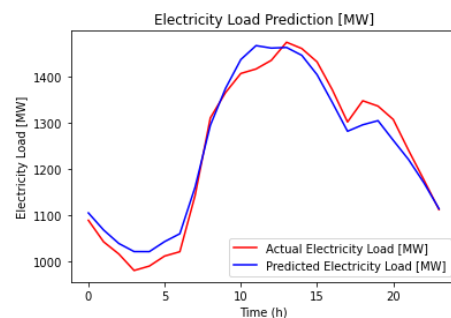
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

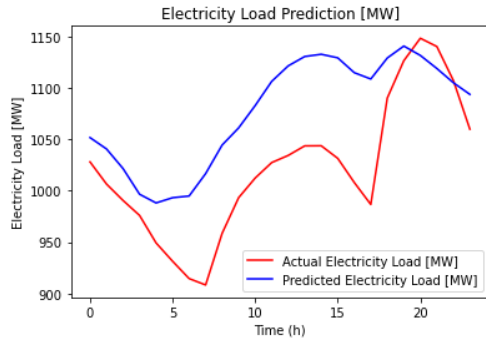


(γ') 7 Απριλίου (Κυριακή)

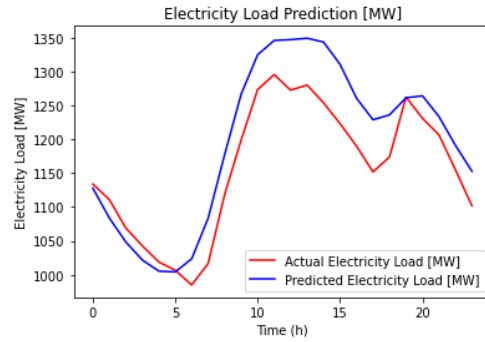


(δ') 16 Οκτωβρίου (Καθημερινή)

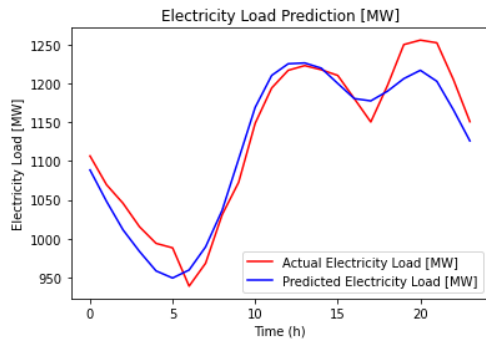
Σχήμα 5.23: Ενδεικτικές προβλέψεις Bidirectional GRU για feature engineering στα σημαντικά χαρακτηριστικά



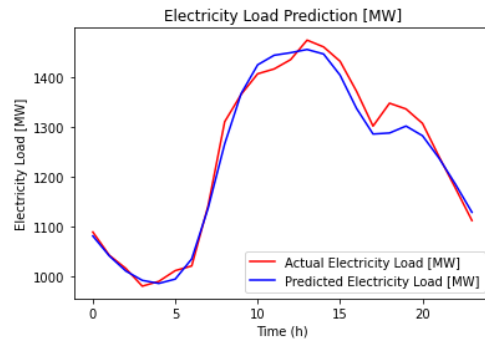
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)



(γ') 7 Απριλίου (Κυριακή)



(δ') 16 Οκτωβρίου (Καθημερινή)

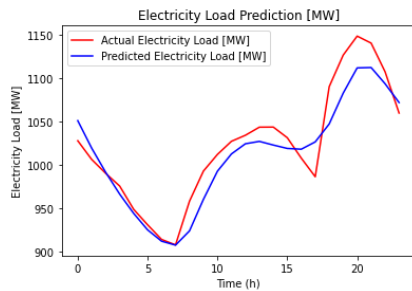
Σχήμα 5.24: Ενδεικτικές προβλέψεις FFNN για feature engineering στα σημαντικά χαρακτηριστικά

Ενώ τα αποτελέσματα με την εφαρμογή του feature engineering σε όλα τα χαρακτηριστικά του συνόλου, με βάση τους αποτιμητές προβλέψεων είναι:

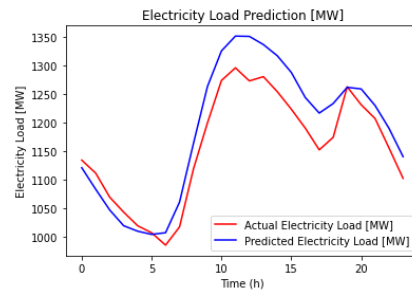
Μοντέλο	RMSE	MAE	MAPE	R^2	Χρόνος εκπαίδευσης (s)
LSTM	61,72077	42,0456	0,036052	0,893925	4176,361
GRU	63,2244	43,52274	0,037388	0,888695	4151,362
BiLSTM	66,14566	45,32702	0,03844	0,878169	19104,91
BiGRU	66,62826	45,66703	0,039105	0,876386	13128,95
FFNN	85,48749	67,01242	0,054531	0,796512	684,0914

Πίνακας 5.11: Αποτιμητές προβλέψεων για το σύνολο ελέγχου του 3ου πειράματος, χρησιμοποιώντας όλα τα χαρακτηριστικά

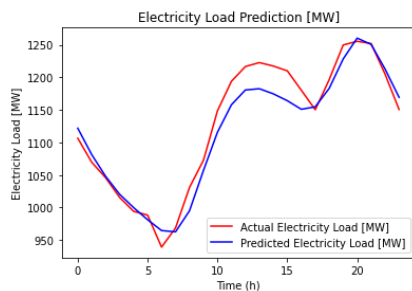
Τα αντίστοιχα ενδεικτικά διαγράμματα είναι:



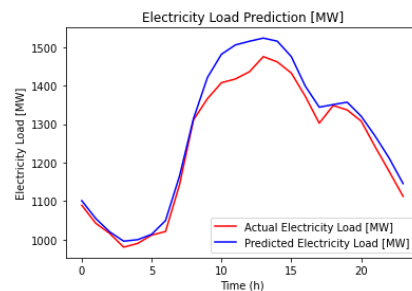
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

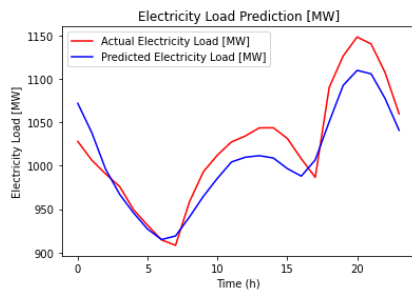


(γ') 7 Απριλίου (Κυριακή)

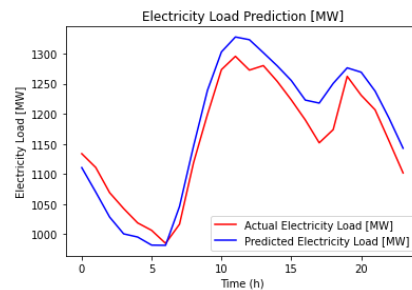


(δ') 16 Οκτωβρίου (Καθημερινή)

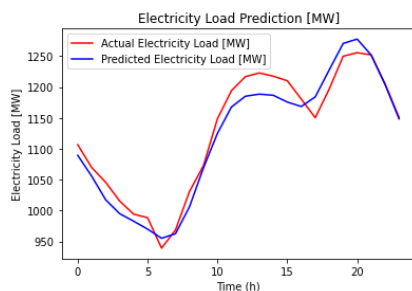
Σχήμα 5.25: Ενδεικτικές προβλέψεις LSTM για feature engineering σε όλα τα χαρακτηριστικά



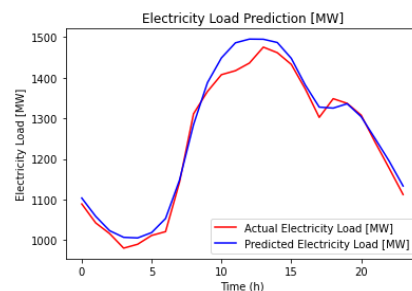
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

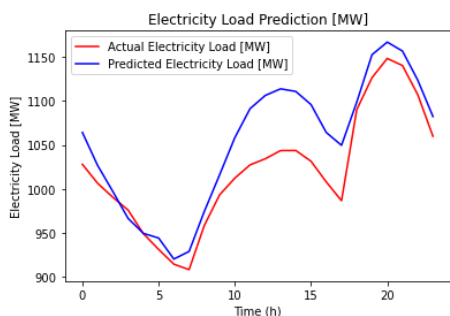


(γ') 7 Απριλίου (Κυριακή)

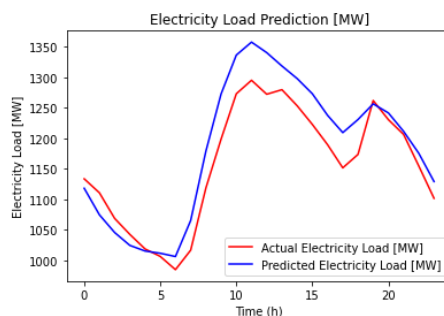


(δ') 16 Οκτωβρίου (Καθημερινή)

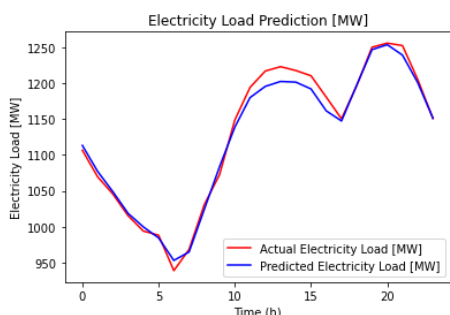
Σχήμα 5.26: Ενδεικτικές προβλέψεις GRU για feature engineering σε όλα τα χαρακτηριστικά



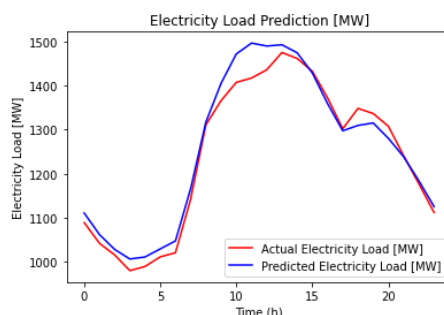
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

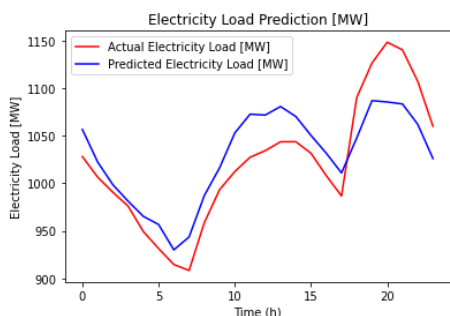


(γ') 7 Απριλίου (Κυριακή)

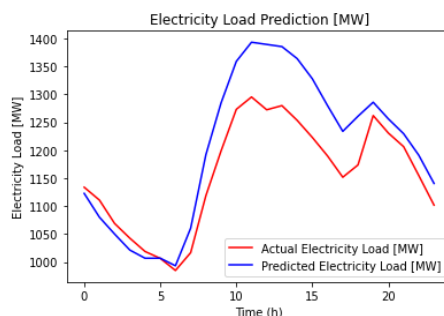


(δ') 16 Οκτωβρίου (Καθημερινή)

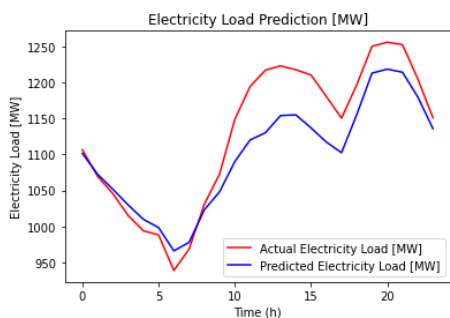
Σχήμα 5.27: Ενδεικτικές προβλέψεις Bidirectional LSTM για feature engineering σε όλα τα χαρακτηριστικά



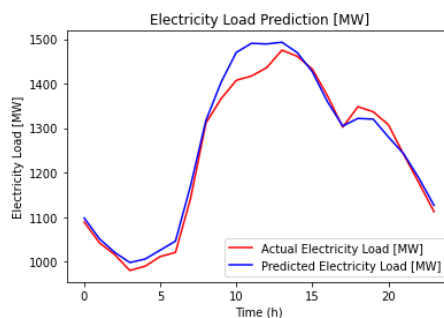
(α') 1 Ιανουαρίου (σταθερή αργία)



(β') 2 Μαρτίου (Κινητή αργία)

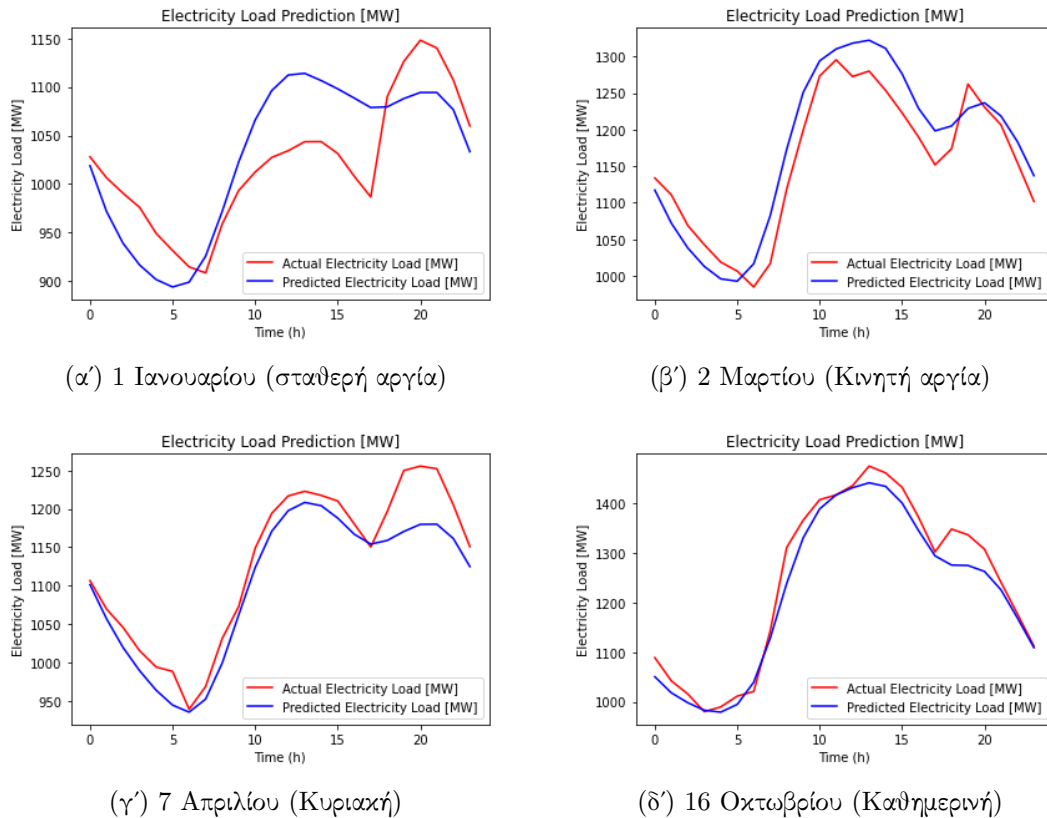


(γ') 7 Απριλίου (Κυριακή)



(δ') 16 Οκτωβρίου (Καθημερινή)

Σχήμα 5.28: Ενδεικτικές προβλέψεις Bidirectional GRU για feature engineering σε όλα τα χαρακτηριστικά



Σχήμα 5.29: Ενδεικτικές προβλέψεις FFNN για feature engineering σε όλα τα χαρακτηριστικά

Το 3ο πείραμα, βελτιώνει σε έναν ικανοποιητικό βαθμό τις προβλέψεις των προηγούμενων 2 πειραμάτων με τη βέλτιστη πρόβλεψη να πραγματοποιείται από το **LSTM** της περίπτωσης όλων των χαρακτηριστικών με $MAPE = 3.6\%$. Σε όλες τις περιπτώσεις, ακόμη και σε αυτήν του **FFNN** όλων των χαρακτηριστικών, παρατηρείται μια αισθητή βελτίωση στις ημέρες τις οποίες εμφάνιζαν αδυναμία τα μοντέλα των προηγούμενων δύο βημάτων, δηλαδή της κινητές και σταθερές αργίες. Επίσης, αξίζει να τονιστεί ότι τα αποδοτικότερα μοντέλα προβλέψεων, όσον αφορά τα ανατροφοδοτούμενα νευρωνικά δίκτυα, προήλθαν από την περίπτωση που χρησιμοποιήθηκαν όλα τα χαρακτηριστικά, γεγονός το οποίο δεν ήταν το αναμενόμενο, λόγω της εισαγωγής περισσότερου θορύβου στο εκάστοτε μοντέλο. Το συμπέρασμα είναι ότι με χρήση όλων των προϋπάρχοντων χαρακτηριστικών, αλλά και την προσθήκη νέων και πολύ βοηθητικών, όπως αποδείχθηκε, υπήρξε μεγαλύτερη συσχέτιση και σύνδεση μεταξύ τους για τον εντοπισμό των μοτίβων. Επίσης, πραγματοποιώντας το **feature engineering** η πληροφορία, με το χαμηλό συντελεστή συσχέτισης αποδείχθηκε τελικά χρήσιμη και η μη χρήση της οδηγεί σε λιγότερο αποτελεσματικές προβλέψεις στην παρούσα εφαρμογή. Ένα άλλο σημαντικό γεγονός, είναι ότι οι προβλέψεις όλων των μοντέλων που χρησιμοποιήθηκαν στο πείραμα αυτό ταιριάζουν σε πολύ μεγάλο βαθμό $R^2 = 0.89$ με τα πραγματικά δεδομένα. Εν αντιθέση με τα ανατροφοδοτούμενα μοντέλα, το **FFNN** παρουσίασε τη μεγαλύτερη

αποτελεσματικότητα όταν εφαρμόστηκε το **feature engineering** στα πιο σημαντικά χαρακτηριστικά, γεγονός που είναι λογικό για τους ίδιους λόγους που ανταποκρίθηκε καλύτερα και στο δεύτερο πείραμα. Το συγκεκριμένο, μοντέλο στην περίπτωση αυτή είναι υπολογίσιμο, καθώς είναι αρκετά ικανοποιητικό και έχει πολύ μικρότερο υπολογιστικό κόστος από τα ανατροφοδοτούμενα μοντέλα. Τέλος, τα ανατροφοδοτούμενα δύο κατευθύνσεων σημείωσαν και αυτά μια αξιοσημείωτη βελτίωση στις προβλέψεις τους, αλλά εξακολουθούν να είναι ελαφρώς λιγότερο αποτελεσματικά από τα μονής κατεύθυνσης, λαμβάνοντας υπόψη και τον παράγοντα υπολογιστικού κόστους. Είναι αρκετά πιθανό πως μια λιγότερο περίπλοκη αρχιτεκτονική (αριθμός νευρώνων και στρωμάτων), θα βελτίωνε τις προβλέψεις των εν λόγω μοντέλων, γεγονός που ισχύει για τη συμμετοχή τους σε όλα τα πειράματα.

6

Συμπεράσματα και μελλοντικές κατευθύνσεις

Σε αυτήν τη διπλωματική εργασία μελετήθηκε η βραχυπρόθεσμη απευθείας πρόβλεψη κατανάλωσης - ζήτησης ηλεκτρικού φορτίου για την πόλη του Παναμά, με ορίζοντα πρόβλεψης 24 ώρες στο μέλλον. Για την παραγωγή των προβλέψεων αναπτύχθηκαν και υλοποιήθηκαν διάφορα μοντέλα βαθιάς μηχανικής μάθησης, δηλαδή νευρωνικών δικτύων. Οι τύποι που χρησιμοποιήθηκαν είναι τα τεχνητά και διάφορων ειδών ανατροφοδοτούμενα νευρωνικά δίκτυα. Για την εκπαίδευση των μοντέλων αυτών χρησιμοποιήθηκε ένα σύνολο δεδομένων, το οποίο περιείχε ιστορικές παρατηρήσεις σε μορφή χρονοσειράς της κατανάλωσης ηλεκτρικού φορτίου του Παναμά σε διάστημα πέντε ετών, καθώς και 15 επιπρόσθετα χαρακτηριστικά, 12 σχετικά με περιβαλλοντικές μετρήσεις και 3 σχετικά με τις ημέρες που χαρακτηρίζονται ως αργίες και Σαββατοκύριακα στη χώρα του Παναμά. Οι καταγραφή των παρατηρήσεων πραγματοποιήθηκε σε ωραία κλίμακα. Τα μοντέλα αυτά αξιολογήθηκαν σε μέρος των δεδομένων αυτών που χρησιμοποιήθηκαν ως σύνολο ελέγχου, με χρήση αποτιμητών πρόβλεψης.

Από τη μελέτη αυτή προέκυψαν διάφορα συμπεράσματα. Αρχικά, για την κατασκευή των παραπάνω μοντέλων είναι αναγκαία η διαθεσιμότητα ενός μεγάλου αριθμού ιστορικών δεδομένων για χρήση κατά την εκπαίδευσή τους. Σε περιπτώσεις όπου υπήρχε σε διαθεσιμότητα μεγάλο πλήθος δεδομένων, όπως οι καθημερινές και οι ενδιάμεσες μεταξύ τους ώρες, αλλά και τα απλά σαββατοκύριακα, παρατηρήθηκε πολύ μεγάλη αποτελεσματικότητα στις προβλέψεις. Εν αντιθέσει με τα προηγούμενα, στις περιπτώσεις των αργιών, όπου εμφανίζονται μόνο για 22 ημέρες ανά έτος στον Παναμά, παρατηρήθηκε η παραγωγή λιγότερο αποδοτικών προβλέψεων. Επιπλέον, χρησιμοποιώντας όλα τα χαρακτηριστικά που περιέχονταν στο σύνολο των δεδομένων αρχικά, παράχθηκαν οι λιγότερο αποδοτικές προβλέψεις, αλλά ήταν αποτελεσματικές σε γενικές γραμμές. Η επιλογή των πιο σημαντικών χαρακτηριστικών μείωσε το θόρυβο των δεδομένων και χρησιμοποιήθηκαν δεδομένα με υψηλή συσχέτιση βελτιώνοντας ελαφρώς τις προηγούμενες προβλέψεις. Τέλος, το πιο αξιοσημείωτο συμπέρασμα, ήταν ότι κατά την ένταξη καινούργιων χαρακτηριστικών στα μοντέλα βελτιώθηκαν αισθητά οι προβλέψεις των α-

νατροφοδοτούμενων νευρωνικών δικτύων στην περίπτωση όπου χρησιμοποιήθηκαν όλα τα αρχικά χαρακτηριστικά, γεγονός που οφείλεται στη μεγαλύτερη συσχέτιση και τον αποτελεσματικότερο συνδυασμό των νέων χαρακτηριστικών με όλα τα υπόλοιπα. Επομένως, η πληροφορία που περιέχονταν αρχικά στο σύνολο θεωρείται χρήσιμη, κατά την προσθήκη των νέων χαρακτηριστικών. Βέβαια, στην περίπτωση του τεχνητού νευρωνικού δικτύου, λόγω της έλλειψης ικανότητας αποθήκευσης πληροφορίας από προηγούμενα χρονικά βήματα, παρατηρήθηκε η μεγαλύτερη αποδοτικότητα στην περίπτωση, όπου τα νέα χαρακτηριστικά συνδυάστηκαν μόνο με τα πιο σημαντικά.

Οι 24ωρες προβλέψεις που παράγονται από τα παραπάνω μοντέλα μπορούν να χρησιμοποιηθούν για τον σωστό προγραμματισμό της παραγωγής του ηλεκτρικού ρεύματος από όλους τους διαθέσιμους πόρους, αλλά και της διανομής του, καθώς δίνουν πληροφορία για της ανάγκες των καταναλωτών σε αυτό το διάστημα. Επιπλέον, σύμφωνα με τον προγραμματισμό των παραπάνω, μπορούν να παρθούν αποφάσεις για την οικονομικότερη διάθεση του στους καταναλωτές, αλλά και τις ανάγκες για συντήρηση των εγκαταστάσεων παραγωγής και διανομής του, ειδικά σε περιπτώσεις όπου αναμένεται μεγάλη ζήτηση. Τα μοντέλα αυτά μπορούν να χρησιμοποιηθούν με αντίστοιχο τρόπο και σε προβλέψεις κατανάλωσης ηλεκτρικού φορτίου, ηλεκτρικών δικτύων άλλων κρατών.

Κάποιες πιθανές μελλοντικές κατευθύνσεις είναι η μελέτη μεγαλύτερου χρονικού ορίζοντα στο μέλλον απευθείας ή και προβλέποντας σε 24 ορίζοντα όλα τα χαρακτηριστικά και τροφοδοτώντας τα ξανά στο εκάστοτε μοντέλο για την πρόβλεψη του επόμενου μήκους πρόβλεψης. Επιπρόσθετα, θα μπορούσαν να μελετηθούν διαφορετικοί τύποι νευρωνικών δικτύων, αλλά και υβριδικά μοντέλα (συνδυασμός διαφορετικών μοντέλων), όπως και άλλων μοντέλων μηχανικής μάθησης ή στατιστικής. Επιπλέον, θα μπορούσε να μελετηθούν διάφορες τεχνικές υπερπαραμετροποίησης (**hyperparameter tuning**) για την εύρεση των βέλτιστων υπερπαραμέτρων στη συγκεκριμένη εφαρμογή. Τέλος, κάποια περεταίρω μελέτη θα μπορούσε να πραγματοποιηθεί στη δημιουργία επιπρόσθετων χαρακτηριστικών που θα μπορούσαν να αυξήσουν ακόμη περισσότερο την αποδοτικότητα των μοντέλων και ιδιαίτερα στις περιπτώσεις των αργιών, όπου συναντάται και η μεγαλύτερη πρόκληση, καθώς και η δοκιμή διαφορετικών τεχνικών για την επιλογή των σημαντικότερων χαρακτηριστικών.

A'

Παράρτημα

Στην ενότητα αυτή παρουσιάζονται τα βασικά κομμάτια κώδικα από την υλοποίηση ενός μοντέλου νευρωνικών δικτύων, από αυτά που χρησιμοποιήθηκαν στην παρούσα διπλωματική εργασία. Η κατασκευή μοντέλων διαφορετικών τύπων μπορεί να πραγματοποιηθεί με παρόμοιο τρόπο.

```
1 #Train-test split
2 index2019 = main_df.index.get_loc('2019-01-01 00:00:00')
3 lookback = 48 #Time steps tested 24,48,72
4 prediction_length = 24
5 train_data = main_df.iloc[:index2019,:].values
6 test_data = main_df.iloc[index2019-lookback:,:].values
7
8 #Data normalization using MinMaxScaler
9 from sklearn.preprocessing import MinMaxScaler
10 scaler = MinMaxScaler(feature_range = (0,1))
11 train_datasc = scaler.fit_transform(train_data)
12 test_datasc = scaler.transform(test_data)
13
14 #Creation of sliding window data structure function
15 def sliding_window(data,lookback,prediction_length):
16     X=[]
17     y=[]
18     for i in range(len(data)-lookback-prediction_length+1):
19         X.append(data[i:i+lookback,:])
20         y.append(data[i+lookback:i+lookback+
21                     prediction_length,0:1])
22     X,y = np.array(X),np.array(y)
23     return X,y
```

```

23 X_train , y_train = sliding_window ( train_datasc , lookback ,
    prediction_length )
24 X_test , y_test = sliding_window ( test_datasc , lookback ,
    prediction_length )
25
26 #Implementation of LSTM network with Early stopping and
    training time calculation
27 from keras.models import Sequential
28 from keras.layers import Dense, LSTM, GRU , Bidirectional
29 from keras.layers import Dropout
30 from keras.callbacks import EarlyStopping
31 import time
32
33 dropout = 0.2
34 b_size = 128
35 n_epochs = 500
36 n_patience = 200
37 split = 0.2
38 model = Sequential ()
39 model.add ( LSTM ( units=80 , return_sequences = True ,
    input_shape=( X_train . shape [ 1 ] , X_train . shape [ 2 ] ) ) )
40 model.add ( Dropout ( dropout ) )
41 model.add ( LSTM ( units=80 , return_sequences=True ) )
42 model.add ( Dropout ( dropout ) )
43 model.add ( LSTM ( units=80 , return_sequences=False ) )
44 model.add ( Dropout ( dropout ) )
45 model.add ( Dense ( units = prediction_length ) )
46 model.compile ( optimizer = 'adam' , loss = '
    mean_squared_error' )
47 es = EarlyStopping ( monitor="val_loss" , mode='min' , patience
    =n_patience , restore_best_weights=True )
48 t0 = time.time ()
49 model.fit ( X_train , y_train , epochs=n_epochs , batch_size=b_size
    , validation_split=split , callbacks=( [ es ] ) , shuffle = False
    )
50 t = time.time ()
51 training_time = t-t0
52

```

```
53 #Plotting of training and validation loss according to the
    number of epochs
54 plt.plot(range(len(model.history.history['loss'])), model.
    history.history['loss'],label='Training Loss')
55 plt.plot(model.history.history['val_loss'],label='
    Validation Loss')
56 plt.legend()
57 plt.xlabel('Epoch Number')
58 plt.ylabel('Loss')
59 plt.show()
60
61 #Predictions and inversion of scaled values
62 inputs = X_test
63 predictions_scaled = model.predict(inputs)
64 scaler_new = MinMaxScaler(feature_range = (0,1))
65 target_sc = scaler_new.fit_transform(train_data[:,0:1])
66 predictions = scaler_new.inverse_transform(
    predictions_scaled)
```

Βιβλιογραφία

- [1] B. Ibrahim et al. “Machine Learning for Short-Term Load Forecasting in Smart Grids”. In: *Energies* 15 (2022), p. 19.
- [2] S. Singh. *Electric Power Generation, Transmission and Distribution*. PHI Learning, 2012.
- [3] J. B. Ekanayake and N. Jenkins. *Smart Grid: Technology and Applications*. Wiley, 2012.
- [4] E. Aguilar Madrid and N. Antonio. “Short-Term Electricity Load Forecasting with Machine Learning”. In: *Information* 12.2 (2021). ISSN: 2078-2489. DOI: [10.3390/info12020050](https://doi.org/10.3390/info12020050). URL: <https://www.mdpi.com/2078-2489/12/2/50>.
- [5] G. Stamatellos and T. Stamatelos. “Short-Term Load Forecasting of the Greek Electricity System”. In: *Applied Sciences* 13.4 (2023). ISSN: 2076-3417. DOI: [10.3390/app13042719](https://doi.org/10.3390/app13042719). URL: <https://www.mdpi.com/2076-3417/13/4/2719>.
- [6] N. Amjady. “Short-term hourly load forecasting using time-series modeling with peak load estimation capability”. In: *IEEE Transactions on Power Systems* 16.3 (2001), pp. 498–505. DOI: [10.1109/59.932287](https://doi.org/10.1109/59.932287).
- [7] J. Zhang et al. “Short term electricity load forecasting using a hybrid model”. In: *Energy* 158 (2018), pp. 774–781. ISSN: 0360-5442. DOI: <https://doi.org/10.1016/j.energy.2018.06.012>. URL: <https://www.sciencedirect.com/science/article/pii/S036054421831065X>.
- [8] F. Chollet. *Deep Learning with Python*. Manning, 2017.
- [9] V. Assimakopoulos. *Forecasting techniques (Lecture 11) Machine Learning [Course Presentation]*. <https://www.fsu.gr/el/component/jdownloads/finish/6/1717>. NTUA - School of Electrical and Computer Engineering.
- [10] W. S. McCulloch and W. Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *Bulletin of Mathematical Biology* 52 (1943), pp. 99–115.
- [11] F. Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain”. In: *Psychological Review* 65 (1958), pp. 386–408.
- [12] O. Calin. *Deep learning architectures*. Springer, 2020.

- [13] R. Hecht-Nielsen. “Theory of the Backpropagation Neural Network”. In: *Proceedings of the International Joint Conference on Neural Networks* 1 (1989), pp. 593–611.
- [14] A. Shrestha and A. Mahmood. “Review of Deep Learning Algorithms and Architectures”. In: *IEEE Access* 7 (2019), pp. 53040–53065.
- [15] S. Ruder. *An overview of gradient descent optimization algorithms*. 2017. arXiv: [1609.04747 \[cs.LG\]](https://arxiv.org/abs/1609.04747).
- [16] N. Srivastava et al. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958.
- [17] X. Glorot and Y. Bengio. “Understanding the difficulty of training deep feedforward neural networks”. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Y. W. Teh and M. Titterton. Vol. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, May 2010, pp. 249–256. URL: <https://proceedings.mlr.press/v9/glorot10a.html>.
- [18] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: [1412.6980 \[cs.LG\]](https://arxiv.org/abs/1412.6980).
- [19] M. I. Jordan. “Chapter 25 - Serial Order: A Parallel Distributed Processing Approach”. In: *Neural-Network Models of Cognition*. Ed. by J. W. Donahoe and V. Packard Dorsel. Vol. 121. Advances in Psychology. North-Holland, 1997, pp. 471–495. DOI: [https://doi.org/10.1016/S0166-4115\(97\)80111-2](https://doi.org/10.1016/S0166-4115(97)80111-2). URL: <https://www.sciencedirect.com/science/article/pii/S0166411597801112>.
- [20] J. L. Elman. “Finding Structure in Time”. In: *Cognitive Science* 14 (1990), pp. 179–211.
- [21] P. J. Werbos. “Backpropagation through time: What It Does and How to Do It”. In: *Proceedings of the IEE* 78 (1990), pp. 1550–1560.
- [22] R. M. Schmidt. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. arXiv: [1912.05911 \[cs.LG\]](https://arxiv.org/abs/1912.05911).
- [23] Y. Bengio, P. Simard, and P. Frasconi. “Learning long-term dependencies with gradient descent is difficult”. In: *IEEE Transactions of Neural Networks* 5 (1994), pp. 157–166.
- [24] S. Hochreiter and J. Schmidhuber. “Long Short-term Memory”. In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [25] W. Kong et al. “Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network”. In: *IEEE Transactions on Smart Grid* 10.1 (2019), pp. 841–851. DOI: [10.1109/TSG.2017.2753802](https://doi.org/10.1109/TSG.2017.2753802).

-
- [26] K. Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. 2014. arXiv: [1406.1078 \[cs.CL\]](#).
- [27] W. Contributors. *Gated recurrent unit*. Wikipedia, Feb. 2019. URL: https://en.wikipedia.org/wiki/Gated_recurrent_unit.
- [28] M. Schuster and K. Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681. DOI: [10.1109/78.650093](#).
- [29] T. pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: [10.5281/zenodo.3509134](#). URL: <https://doi.org/10.5281/zenodo.3509134>.
- [30] C. R. Harris et al. “Array programming with NumPy”. In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: [10.1038/s41586-020-2649-2](#). URL: <https://doi.org/10.1038/s41586-020-2649-2>.
- [31] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <https://www.tensorflow.org/>.
- [32] F. Chollet et al. *Keras*. <https://keras.io>. 2015.
- [33] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](#).
- [34] M. L. Waskom. “seaborn: statistical data visualization”. In: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: [10.21105/joss.03021](#). URL: <https://doi.org/10.21105/joss.03021>.
- [35] S. Seabold and J. Perktold. “statsmodels: Econometric and statistical modeling with python”. In: *9th Python in Science Conference*. 2010.
- [36] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [37] *Panama Electricity Load Forecasting Dataset*. <https://www.kaggle.com/datasets/pateljay731/panama-electricity-load-forecasting>.