

Ατομική Διπλωματική Εργασία

**Πρόβλεψη Ποδοσφαιρικών Αγώνων: Εφαρμογή με την χρήση
μοντέλου Transformer για την Εκτίμηση Over/Under 2,5 Τερμάτων**

Σολομωνίδης Γεώργιος

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μάιος 2026

Δηλώνω ότι η παρούσα διατριβή και οποιοδήποτε συνοδευτικό λογισμικό αποτελούν δικό μου πρωτότυπο έργο, εκπονημένο σε πλήρη συμμόρφωση με τον Κώδικα Δεοντολογίας για την Έρευνα του Πανεπιστημίου Κύπρου (<https://www.ucy.ac.cy/legislation/kodikas-deontologias-kai-politikes/>), και με πλήρη ευθύνη εκ μέρους μου για την ακρίβεια, την ακεραιότητα και την εγκυρότητα όλου του περιεχομένου.

Περίληψη

Η παρούσα διπλωματική εργασία εξετάζει την πρόβλεψη του αποτελέσματος Over/Under 2.5 γκολ σε ποδοσφαιρικούς αγώνες της αγγλικής Championship, με χρήση μοντέλων μηχανικής μάθησης και κύρια έμφαση σε Transformer-based αρχιτεκτονική. Το πρόβλημα διατυπώνεται ως δυαδική ταξινόμηση, όπου η κλάση Over αντιστοιχεί σε αγώνες με περισσότερα από 2.5 συνολικά γκολ, ενώ η κλάση Under αντιστοιχεί σε αγώνες με έως 2.5 γκολ.

Βασικός στόχος της εργασίας είναι η δημιουργία ενός αξιόπιστου συστήματος πρόβλεψης που αξιοποιεί αποκλειστικά πληροφορία διαθέσιμη πριν από την έναρξη του αγώνα. Για τον σκοπό αυτό χρησιμοποιήθηκαν ιστορικά δεδομένα αγώνων, στατιστικά χαρακτηριστικά ομάδων, δείκτες φόρμας, μεταβλητές σχετικές με την έδρα, καθώς και πληροφορία από στοιχηματικές αποδόσεις μέσω implied probabilities. Ιδιαίτερη έμφαση δόθηκε στην αποφυγή διαρροής μελλοντικής πληροφορίας, με χρονολογική ταξινόμηση των δεδομένων και αξιολόγηση σε μεταγενέστερη αγωνιστική περίοδο.

Στο πλαίσιο της εργασίας αξιολογήθηκαν διαφορετικές μορφές εισόδου με 26, 34 και 47 χαρακτηριστικά, καθώς και διαφορετικά μοντέλα, όπως MLP, RBF και Transformer Encoder. Η καλύτερη επίδοση επιτεύχθηκε από Transformer-based μοντέλο με είσοδο 47 χαρακτηριστικών, όπου η πληροφορία του αγώνα συμπυκνώνεται σε ένα ενιαίο pre-match token που περιγράφει το τρέχον fixture. Η συγκεκριμένη αναπαράσταση συνδυάζει engineered χαρακτηριστικά για τη γηπεδούχο και τη φιλοξενούμενη ομάδα, καθώς και πληροφορία της αγοράς από bookmaker odds. Το καλύτερο αποτέλεσμα ήταν test accuracy 60.78% και ROI 6.07%, γεγονός που δείχνει ότι η ποιότητα της αναπαράστασης εισόδου επηρεάζει καθοριστικά την τελική απόδοση του μοντέλου. Τα αποτελέσματα δείχνουν ότι η απλή αύξηση της πολυπλοκότητας του μοντέλου δεν αρκεί για ουσιαστική βελτίωση, αν δεν συνοδεύεται από κατάλληλη και μεθοδολογικά σωστή pre-match αναπαράσταση. Παράλληλα, η αξιολόγηση με χρηματοοικονομικούς δείκτες, όπως profit και ROI, ανέδειξε ότι ένα μοντέλο μπορεί να έχει ικανοποιητική ακρίβεια χωρίς απαραίτητα να είναι πάντα κερδοφόρο. Συνεπώς, η εργασία καταλήγει ότι η πρόβλεψη Over/Under 2.5 είναι ένα δύσκολο πρόβλημα, στο οποίο η σωστή επιλογή χαρακτηριστικών, η αποφυγή data leakage και η αξιολόγηση με πραγματικά στοιχηματικά κριτήρια είναι εξίσου σημαντικές με την επιλογή της αρχιτεκτονικής.

Περιεχόμενο

Περίληψη	3
Κεφάλαιο 1	6
1.1 Στόχος	6
1.2 Προηγούμενες προσπάθειες και σχετική έρευνα	8
Κεφάλαιο 2	13
2.1 Νευρωνικά Δίκτυα - Ιστορική Αναδρομή	14
2.2 Νευρώνες του Εγκεφάλου	17
2.3 Τεχνητά Νευρωνικά Δίκτυα	18
2.4 Αρχιτεκτονικές Νευρωνικών Δικτύων	20
2.5 MLP με Backpropagation	22
2.6 Δίκτυα Radial Basis Function	23
2.7 Hessian Free Optimization	26
2.8 Αρχιτεκτονική Transformer	28
Κεφάλαιο 3	34
3.1 Δεδομένα εισόδου	35
3.2 Το πρωτάθλημα της Championship	40
3.3 Λήψη και επεξεργασία δεδομένων	42
3.4 Κανονικοποίηση Τιμών	44
3.5 Σχεδιασμός και Υλοποίηση της Pipeline	46
3.5.1 Το αρχείο preprocessing_rawseq.py	48
3.5.2 Το αρχείο dataset.py	51
3.5.3 Το αρχείο transformer.py	52
3.5.4 Το αρχείο train.py	54
3.5.5 Τα βοηθητικά αρχεία ελέγχου και επιθεώρησης	57
3.5.6 Συνολική αποτίμηση της υλοποίησης	58
3.6 Πρόσθετες Πειραματικές Μεθοδολογικές Κατευθύνσεις	60
Κεφάλαιο 4	63
4.1 Αποτελέσματα MLP Backpropagation	67
4.1.1 MLP με 26 χαρακτηριστικά εισόδου	67
4.1.2 MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1	71
4.1.3 MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2	75
4.1.4 MLP με 47 χαρακτηριστικά εισόδου	79
4.2 Αποτελέσματα RBF	84

4.2.1 RBF με 26 χαρακτηριστικά εισόδου.....	84
4.2.2 RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.....	88
4.2.3 RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.....	92
4.2.4 RBF με 47 χαρακτηριστικά εισόδου.....	96
4.3 Transformer.....	101
4.3.1 Transformer με 26 χαρακτηριστικά εισόδου	101
4.3.2 Transformer με 34 χαρακτηριστικά εισόδου	105
4.3.3 Transformer με 47 χαρακτηριστικά εισόδου	109
4.4 Σχέση Ακρίβειας και Στοιχηματικού Κέρδους.....	115
4.5 Στοιχηματικό κέρδος με πραγματικές αποδόσεις.....	117
Κεφάλαιο 5	119
5.1 Συμπεράσματα	120
5.2 Μελλοντική Εργασία	123
References.....	125
Appendix.....	128

Κεφάλαιο 1

Εισαγωγή

1.1 Στόχος της έρευνας

1.2 Προηγούμενες προσπάθειες

1.1 Στόχος

Σκοπός της παρούσας εργασίας είναι η υλοποίηση, εκπαίδευση και συγκριτική αξιολόγηση μοντέλων μηχανικής μάθησης με έμφαση σε αρχιτεκτονικές τύπου Transformer, με στόχο την πρόβλεψη του πλήθους των τερμάτων που θα σημειωθούν σε έναν ποδοσφαιρικό αγώνα στο πλαίσιο του αγγλικού πρωταθλήματος Championship. Πιο συγκεκριμένα, το πρόβλημα διατυπώνεται ως δυαδική ταξινόμηση, όπου επιδιώκεται η εκτίμηση του κατά πόσο ο συνολικός αριθμός των τερμάτων θα υπερβεί ή όχι ένα προκαθορισμένο όριο (over/under 2.5 goals), το οποίο αποτελεί βασική αγορά στοιχηματισμού.

Στην εργασία αυτή μελετάται κυρίως η εφαρμογή ενός Transformer Encoder σε διαδοχικά δεδομένα αγώνων, με στόχο την αξιοποίηση της χρονικής πληροφορίας που ενσωματώνεται στην πρόσφατη αγωνιστική συμπεριφορά των ομάδων. Παράλληλα, εξετάζονται και συγκρίνονται πιο παραδοσιακές προσεγγίσεις μηχανικής μάθησης, όπως το Multilayer Perceptron - MLP, Radial Basis Function - RBF, προκειμένου να αξιολογηθεί αν η χρήση πιο σύνθετων αρχιτεκτονικών οδηγεί σε ουσιαστική βελτίωση της απόδοσης.

Ιδιαίτερη έμφαση δίνεται στον σχεδιασμό κατάλληλης αναπαράστασης των δεδομένων πριν τον αγώνα (pre-match representation), η οποία να ενσωματώνει πληροφορία σχετικά με τη φόρμα των ομάδων, τα στατιστικά από προηγούμενους αγώνες, καθώς και το πλαίσιο του εκάστοτε αγώνα, χωρίς να εισάγεται διαρροή πληροφορίας από το μέλλον. Η διατήρηση της χρονικής συνέπειας (chronology) αποτελεί βασική προϋπόθεση για την εξαγωγή αξιόπιστων και επιστημονικά τεκμηριωμένων αποτελεσμάτων.

Κύριος στόχος της διπλωματικής εργασίας είναι η ανάπτυξη ενός αξιόπιστου συστήματος πρόβλεψης, το οποίο να μπορεί να αξιολογηθεί τόσο με κλασικά μέτρα απόδοσης ταξινόμησης (όπως ακρίβεια, precision, recall, F1-score και Receiver Operating Characteristic – Area Under the Curve /ROC-AUC), όσο και με πρακτικά κριτήρια που σχετίζονται με τη λήψη αποφάσεων στον στοιχηματισμό, όπως η απόδοση επένδυσης (ROI – Return Of Investment). Μέσω αυτής της διπλής αξιολόγησης επιδιώκεται να διερευνηθεί κατά πόσο ένα μοντέλο μπορεί όχι μόνο να είναι στατιστικά ακριβές, αλλά και πρακτικά χρήσιμο.

1.2 Προηγούμενες προσπάθειες και σχετική έρευνα

Στη βιβλιογραφία έχουν προταθεί διάφορες προσεγγίσεις για την πρόβλεψη ποδοσφαιρικών αποτελεσμάτων. Οι πρώτες στατιστικές μέθοδοι βασίστηκαν κυρίως σε μοντέλα Poisson για την εκτίμηση των γκολ κάθε ομάδας. Ο Maher (1982) ανέπτυξε ένα από τα πρώτα μοντέλα βαθμολόγησης επιθετικής και αμυντικής ικανότητας ομάδων, ενώ οι Dixon και Coles (1997) επέκτειναν την προσέγγιση αυτή ενσωματώνοντας δυναμικές στοιχηματικές στρατηγικές. Στη συνέχεια, οι Rue και Salvesen (2000) εισήγαγαν Bayesian προσεγγίσεις με χρονικά μεταβαλλόμενες αξιολογήσεις ομάδων, ενώ οι Goddard (2005) και Karlis και Ntzoufras (2003) μελέτησαν εναλλακτικές μορφές στατιστικής μοντελοποίησης για ποδοσφαιρικά δεδομένα.

Με την εξέλιξη της μηχανικής μάθησης, εμφανίστηκαν πιο σύνθετες τεχνικές πρόβλεψης βασισμένες σε νευρωνικά δίκτυα, Bayesian Networks και ensemble μοντέλα. Οι Huang και Chang (2010) χρησιμοποίησαν νευρωνικά δίκτυα για την πρόβλεψη αγώνων του Παγκοσμίου Κυπέλλου, ενώ οι Joseph, Fenton και Neil (2006) αξιοποίησαν Bayesian Networks για την πρόβλεψη ποδοσφαιρικών αποτελεσμάτων. Επιπλέον, οι Hubáček, Šourek και Železný (2019) πέτυχαν ιδιαίτερα ανταγωνιστικά αποτελέσματα χρησιμοποιώντας Gradient Boosted Trees σε μεγάλο διαγωνισμό πρόβλεψης ποδοσφαίρου.

Τα τελευταία χρόνια, η έρευνα έχει στραφεί έντονα προς τη χρήση πιο πλούσιων αγωνιστικών χαρακτηριστικών πέρα από τα τελικά γκολ. Μελέτες έχουν δείξει ότι στατιστικά όπως τα shots, τα shots on target, τα corners και τα expected goals (xG) μπορούν να αποτυπώσουν καλύτερα την πραγματική αγωνιστική εικόνα μιας ομάδας σε σχέση με το απλό πλήθος γκολ, το οποίο επηρεάζεται σε μεγάλο βαθμό από τυχαίους παράγοντες.

Ιδιαίτερα σημαντική για την παρούσα εργασία είναι η μελέτη του Wheatcroft (2020), η οποία επικεντρώθηκε στην πρόβλεψη της αγοράς Over/Under 2.5 goals στο ποδόσφαιρο. Στην εργασία του αναπτύχθηκε ένα σύστημα αξιολόγησης ομάδων με την ονομασία Generalised Attacking Performance (GAP) ratings, το οποίο αποτυπώνει την επιθετική και αμυντική απόδοση κάθε ομάδας κατά τη διάρκεια της σεζόν. Σε αντίθεση με παραδοσιακές προσεγγίσεις που βασίζονται αποκλειστικά στα τελικά γκολ, ο Wheatcroft αξιοποίησε πιο λεπτομερή αγωνιστικά στατιστικά, όπως shots, shots on target και corners, υποστηρίζοντας ότι αυτά περιγράφουν καλύτερα την

πραγματική αγωνιστική εικόνα μιας ομάδας και μειώνουν την επίδραση της τύχης που υπάρχει στο σκοράρισμα.

Στη συνέχεια, χρησιμοποίησε logistic regression για την άμεση πρόβλεψη της πιθανότητας Over/Under 2.5 και αξιολόγησε το μοντέλο του όχι μόνο ως προς την ακρίβεια πρόβλεψης, αλλά και ως προς τη στοιχηματική κερδοφορία μέσω value betting στρατηγικών. Ο Wheatcroft (2020) ανέφερε ότι το μοντέλο του πέτυχε μέσο κέρδος περίπου 0.8% ανά στοίχημα σε διάστημα δώδεκα ετών και σε δέκα ευρωπαϊκά πρωταθλήματα, όταν χρησιμοποιήθηκαν ως είσοδοι μόνο χαρακτηριστικά όπως shots και corners και όχι τα ίδια τα γκολ. Τα αποτελέσματα της έρευνας αυτής δείχνουν ότι η χρήση προηγμένων στατιστικών χαρακτηριστικών μπορεί να οδηγήσει σε μικρό αλλά σταθερό θετικό profit over time, γεγονός που καθιστά τη μελέτη ιδιαίτερα σχετική με την παρούσα διπλωματική εργασία.

Πέρα από τη διεθνή βιβλιογραφία, σημαντικό υπόβαθρο για την παρούσα εργασία αποτέλεσαν και προηγούμενες διπλωματικές εργασίες που ασχολήθηκαν με παρόμοιο πρόβλημα πρόβλεψης ποδοσφαιρικών αγώνων, κυρίως στο πλαίσιο της αγοράς Over/Under 2.5. Οι εργασίες αυτές είχαν ιδιαίτερη αξία, καθώς προσέφεραν πρακτικά παραδείγματα υλοποίησης, επιλογής χαρακτηριστικών και αξιολόγησης μοντέλων σε δεδομένα της Championship. Σε ορισμένες από αυτές χρησιμοποιήθηκαν νευρωνικά δίκτυα όπως MLP, RBF και άλλες τεχνικές μηχανικής μάθησης, με στόχο είτε την πρόβλεψη του συνολικού αριθμού τερμάτων είτε την κατηγοριοποίηση των αγώνων σε Over και Under 2.5.

Ιδιαίτερα σημαντικές για τη δική μου μελέτη ήταν οι προσπάθειες του Ανδρέα Ιακώβου και του Μιχάλη Ξυδιά, καθώς ανέφεραν πολύ υψηλές επιδόσεις στο συγκεκριμένο πρόβλημα. Ο Ιακώβου εφάρμοσε Radial Basis Function Networks και Support Vector Machines, αναφέροντας ιδιαίτερα υψηλή ακρίβεια στα δεδομένα ελέγχου. Αντίστοιχα, ο Ξυδιάς μελέτησε MLP, RBF και Hessian-Free Optimization, αναφέροντας ακόμη υψηλότερες επιδόσεις, με το HFO να εμφανίζεται ως το πιο αποδοτικό μοντέλο της εργασίας του. Τα αποτελέσματα αυτά αποτέλεσαν αρχικά σημαντικό σημείο αναφοράς για την παρούσα διπλωματική, καθώς φαινόταν να υποδεικνύουν ότι το πρόβλημα Over/Under 2.5 μπορεί να επιλυθεί με ιδιαίτερα υψηλή ακρίβεια.

Ωστόσο, κατά τη μελέτη των προηγούμενων αυτών εργασιών, κρίθηκε απαραίτητο να εξεταστεί πιο προσεκτικά και η μεθοδολογική αξιοπιστία των αποτελεσμάτων. Ο λόγος είναι ότι οι αναφερόμενες επιδόσεις ήταν αρκετά υψηλές για ένα πρόβλημα που θεωρείται δύσκολο,

θορυβώδες και έντονα επηρεασμένο από την αποδοτικότητα των στοιχηματικών αγορών. Για τον σκοπό αυτό, στο πλαίσιο της παρούσας εργασίας αφιερώθηκε χρόνος στην ανάγνωση και ανάλυση του διαθέσιμου κώδικα των δύο αυτών προσεγγίσεων.

Στην περίπτωση της εργασίας του Μιχάλη Ξυδιά, η επιθεώρηση του κώδικα έδειξε πιθανή διαρροή πληροφορίας από τον ίδιο τον υπό πρόβλεψη αγώνα. Πιο συγκεκριμένα, ο τρέχων αγώνας φαινόταν να εισάγεται στο ιστορικό των δύο ομάδων πριν από τον υπολογισμό των στατιστικών χαρακτηριστικών που χρησιμοποιούνται ως είσοδος στο μοντέλο. Με αυτόν τον τρόπο, χαρακτηριστικά όπως οι μέσοι όροι γκολ υπέρ και κατά, τα ποσοστά σκοραρίσματος, τα ποσοστά παθητικού και τα ποσοστά Over/Under μπορούσαν να περιλαμβάνουν ήδη πληροφορία από το αποτέλεσμα του αγώνα που το μοντέλο καλούνταν να προβλέψει. Σε ένα πραγματικό pre-match περιβάλλον, τέτοια πληροφορία δεν θα ήταν διαθέσιμη. Επομένως, η παρουσία της στα χαρακτηριστικά εισόδου μπορεί να οδηγήσει σε τεχνητά αυξημένη ακρίβεια, καθώς το μοντέλο αποκτά έμμεση πρόσβαση στο target.

Στην περίπτωση της εργασίας του Ανδρέα Ιακώβου, το βασικό ζήτημα που εντοπίστηκε δεν αφορούσε την ίδια μορφή current-match feature leakage, αλλά πιθανή επιμόλυνση μεταξύ training και test set. Από την ανάγνωση του κώδικα προέκυψε ότι η διαδικασία εκπαίδευσης φαινόταν να περιλαμβάνει τη σεζόν που στη συνέχεια χρησιμοποιούνταν και για αξιολόγηση. Με άλλα λόγια, το μοντέλο φαινόταν να εκπαιδεύεται σε δεδομένα της ίδιας περιόδου στην οποία αργότερα ελεγχόταν. Αυτό μειώνει την ανεξαρτησία του test set και μπορεί επίσης να οδηγήσει σε υπεραισιόδοξη εκτίμηση της πραγματικής ικανότητας γενίκευσης.

Τα παραπάνω ευρήματα δεν αναιρούν τη σημασία των προηγούμενων εργασιών. Αντίθετα, οι εργασίες αυτές αποτέλεσαν ουσιαστική βάση για την κατανόηση του προβλήματος, για την επιλογή του πρωταθλήματος, για τη μελέτη πιθανών χαρακτηριστικών εισόδου και για τη διαμόρφωση της πειραματικής κατεύθυνσης της παρούσας διπλωματικής. Παράλληλα όμως, ανέδειξαν ένα από τα σημαντικότερα μεθοδολογικά ζητήματα στην πρόβλεψη ποδοσφαιρικών αγώνων: την ανάγκη αυστηρής αποφυγής data leakage και τη διασφάλιση ότι η αξιολόγηση γίνεται σε πραγματικά ανεξάρτητο χρονικό σύνολο.

Για τον λόγο αυτό, στην παρούσα εργασία δίνεται ιδιαίτερη έμφαση στη δημιουργία αυστηρά pre-match δειγμάτων. Κάθε δείγμα κατασκευάζεται αποκλειστικά με πληροφορία που θα ήταν διαθέσιμη πριν από τη σέντρα του αγώνα, ενώ η αξιολόγηση πραγματοποιείται χρονολογικά,

χρησιμοποιώντας μεταγενέστερη σεζόν ως σύνολο επαλήθευσης. Με αυτόν τον τρόπο, τα αποτελέσματα της παρούσας εργασίας είναι ενδεχομένως πιο συντηρητικά σε σχέση με ορισμένες προηγούμενες αναφορές, αλλά θεωρούνται πιο ρεαλιστικά και μεθοδολογικά αξιόπιστα.

Επομένως, η σύγκριση με προηγούμενες εργασίες δεν πρέπει να γίνεται μόνο με βάση το τελικό ποσοστό ακρίβειας, αλλά και με βάση τη διαδικασία με την οποία παράγονται και αξιολογούνται τα αποτελέσματα. Στο πρόβλημα Over/Under 2.5, η αποφυγή μελλοντικής πληροφορίας είναι κρίσιμη, καθώς ακόμη και μικρή διαρροή μπορεί να αυξήσει τεχνητά την απόδοση του μοντέλου. Η παρούσα εργασία προσπαθεί να αντιμετωπίσει το ζήτημα αυτό μέσω χρονολογικά συνεπούς προεπεξεργασίας, train-only normalization και αξιολόγησης σε ξεχωριστή μεταγενέστερη σεζόν.

Παρά τη σημαντική πρόοδο των παραδοσιακών και ensemble μεθόδων, οι σύγχρονες αρχιτεκτονικές βαθιάς μάθησης προσφέρουν νέες δυνατότητες στην ανάλυση ακολουθιακών δεδομένων. Ειδικότερα, οι αρχιτεκτονικές Transformer έχουν επικρατήσει τα τελευταία χρόνια σε προβλήματα που απαιτούν κατανόηση χρονικών εξαρτήσεων και σχέσεων μεταξύ στοιχείων μιας ακολουθίας. Σε αντίθεση με παλαιότερα νευρωνικά δίκτυα, οι Transformers χρησιμοποιούν μηχανισμούς self-attention, ώστε να μπορούν να επεξεργάζονται ολόκληρη την ακολουθία ταυτόχρονα και να εντοπίζουν σημαντικές σχέσεις μεταξύ διαφορετικών χρονικών στιγμών.

Η ιδιότητα αυτή είναι ιδιαίτερα χρήσιμη για το πρόβλημα της πρόβλεψης Over/Under 2.5, καθώς η αγωνιστική συμπεριφορά μιας ομάδας δεν καθορίζεται μόνο από έναν μεμονωμένο προηγούμενο αγώνα, αλλά από την εξέλιξη της φόρμας, της επιθετικής και αμυντικής συμπεριφοράς και των τάσεων σκοραρίσματος μέσα στον χρόνο. Ο μηχανισμός attention επιτρέπει στο μοντέλο να δώσει διαφορετική βαρύτητα σε διαφορετικά στοιχεία της εισόδου, γεγονός που καθιστά τους Transformers κατάλληλους για την ανάλυση χρονικά οργανωμένων pre-match δεδομένων.

Με βάση τα παραπάνω, η παρούσα διπλωματική εργασία τοποθετείται στο σημείο συνάντησης τριών βασικών κατευθύνσεων: της στατιστικής πρόβλεψης ποδοσφαιρικών αγώνων, της χρήσης πλούσιων προαγωνιστικών χαρακτηριστικών και της εφαρμογής σύγχρονων Transformer-based μοντέλων. Στόχος δεν είναι μόνο η επίτευξη υψηλής ακρίβειας, αλλά και η αξιολόγηση της πρακτικής αξίας των προβλέψεων μέσω profit και ROI, σε ένα πλαίσιο που αποφεύγει τη

διαρροή πληροφορίας και προσομοιώνει όσο το δυνατόν πιο ρεαλιστικά τη διαδικασία πρόβλεψης πριν από την έναρξη του αγώνα.

Κεφάλαιο 2

Υπόβαθρο

2.1 Νευρωνικά Δίκτυα – Ιστορική Αναδρομή

2.2 Νευρώνες Εγκεφάλου

2.3 Τεχνητά Νευρωνικά Δίκτυα

2.4 Αρχιτεκτονικές Νευρωνικών Δικτύων

2.5 MLP με Backpropagation

2.6 Radial Basis Function Networks

2.7 Hessian Free Optimization

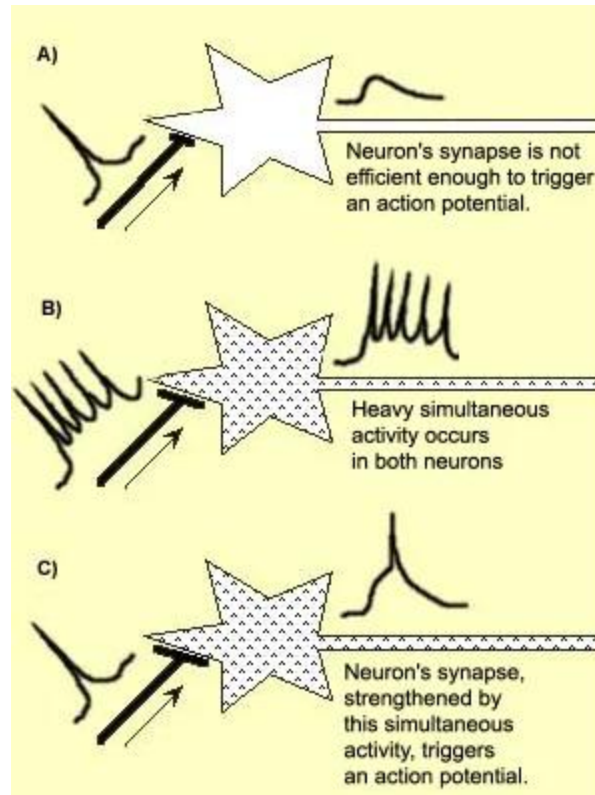
2.8 Transformer

2.1 Νευρωνικά Δίκτυα - Ιστορική Αναδρομή

Η εξέλιξη των νευρωνικών δικτύων αποτελεί ένα από τα βασικά θεωρητικά θεμέλια της σύγχρονης μηχανικής μάθησης. Παρότι η σημερινή βιβλιογραφία συνδέει τα νευρωνικά δίκτυα με πολύπλοκες αρχιτεκτονικές βαθιάς μάθησης, η αρχική τους σύλληψη ήταν σημαντικά απλούστερη: η δημιουργία υπολογιστικών μοντέλων που να προσεγγίζουν, έστω αφαιρετικά, τον τρόπο με τον οποίο οργανώνεται και επεξεργάζεται πληροφορία ο ανθρώπινος εγκέφαλος. Για τον λόγο αυτό, μια σύντομη ιστορική αναδρομή είναι χρήσιμη, επειδή επιτρέπει να τοποθετηθούν σε ευρύτερο πλαίσιο τόσο οι κλασικές προσεγγίσεις όσο και οι νεότερες αρχιτεκτονικές που αξιοποιούνται στην παρούσα εργασία.

Η πρώτη συστηματική διατύπωση της ιδέας των τεχνητών νευρωνικών δικτύων αποδίδεται στους McCulloch και Pitts το 1943. Οι ερευνητές αυτοί πρότειναν ένα αφηρημένο μαθηματικό μοντέλο νευρώνα, στο οποίο πολλαπλές είσοδοι συνδυάζονται και οδηγούν σε μια δυαδική έξοδο ενεργοποίησης. Η βασική σημασία της προσέγγισής τους δεν βρίσκεται στην υπολογιστική της ισχύ με τα σημερινά δεδομένα, αλλά στο γεγονός ότι έδειξε πως η λογική λειτουργία του νευρικού συστήματος μπορεί να περιγραφεί με τυπικούς κανόνες και να μεταφερθεί σε υπολογιστικό πλαίσιο. Με αυτή την έννοια, τέθηκαν οι πρώτες βάσεις για την ιδέα ότι η μάθηση και η επεξεργασία πληροφορίας μπορούν να προκύψουν από τη συνεργασία πολλών απλών μονάδων.

Καθοριστική υπήρξε και η συμβολή του Hebb το 1949, ο οποίος διατύπωσε την αρχή σύμφωνα με την οποία η σύνδεση μεταξύ δύο νευρώνων ενισχύεται όταν αυτοί ενεργοποιούνται επανειλημμένα με συσχετισμένο τρόπο. Η λογική αυτή, γνωστή ως Hebbian learning, παρείχε ένα από τα πρώτα ερμηνευτικά σχήματα για το πώς ένα δίκτυο θα μπορούσε να "μαθαίνει" από την εμπειρία μέσω της τροποποίησης των συνδέσεών του. Παρότι τα σύγχρονα αλγοριθμικά σχήματα εκπαίδευσης είναι πολύ πιο σύνθετα, η ιδέα ότι η μάθηση συνδέεται με την προσαρμογή των βαρών παραμένει μέχρι σήμερα κεντρική.



Εικόνα 2.1: Σχηματική αναπαράσταση της αρχής Hebbian learning

Στη συνέχεια, η έρευνα στράφηκε από τη θεωρητική διατύπωση προς πιο απτά υπολογιστικά μοντέλα. Στα τέλη της δεκαετίας του 1950, ο Rosenblatt παρουσίασε το perceptron, ένα από τα πρώτα πρακτικά νευρωνικά μοντέλα. Το perceptron μπορούσε να μαθαίνει απλούς γραμμικούς διαχωρισμούς και δημιούργησε μεγάλη αισιοδοξία ότι τα νευρωνικά δίκτυα θα μπορούσαν σύντομα να λύσουν ευρύ φάσμα προβλημάτων τεχνητής νοημοσύνης. Ωστόσο, ο αρχικός αυτός ενθουσιασμός περιορίστηκε όταν έγινε σαφές ότι το μονοεπίπεδο perceptron έχει ουσιώδεις περιορισμούς και αδυνατεί να επιλύσει μη γραμμικά διαχωρίσιμα προβλήματα.

Οι περιορισμοί αυτοί αναδείχθηκαν με ιδιαίτερη σαφήνεια από το έργο των Minsky και Papert. Η ανάλυσή τους έδειξε ότι τα απλά perceptrons δεν επαρκούν για πολλές κατηγορίες προβλημάτων, γεγονός που οδήγησε σε αισθητή μείωση του ενδιαφέροντος για τα νευρωνικά δίκτυα για ένα χρονικό διάστημα. Η περίοδος αυτή είναι σημαντική ιστορικά, επειδή δείχνει ότι η πρόοδος στον χώρο δεν ήταν γραμμική. Αντίθετα, η ανάπτυξη των νευρωνικών δικτύων χαρακτηρίστηκε από φάσεις ενθουσιασμού, κριτικής επανεκτίμησης και τεχνολογικής ωρίμανσης.

Το ενδιαφέρον για τα νευρωνικά δίκτυα αναζωπυρώθηκε τη δεκαετία του 1980. Ένα βασικό σημείο καμπής ήταν το έργο του Hopfield το 1982, το οποίο έδειξε ότι ένα νευρωνικό δίκτυο μπορεί να μελετηθεί και ως δυναμικό σύστημα με ιδιότητες μνήμης. Τα δίκτυα Hopfield ανέδειξαν ότι οι νευρωνικές αρχιτεκτονικές δεν είναι απλώς ταξινομητές εισόδου-εξόδου, αλλά μπορούν να εμφανίζουν σταθερές καταστάσεις, ανακατασκευή προτύπων και πιο σύνθετη εσωτερική συμπεριφορά. Η συμβολή αυτή επανέφερε στο προσκήνιο τη μελέτη των νευρωνικών μοντέλων ως σοβαρού επιστημονικού πεδίου.

Ακόμη σημαντικότερη για τη σύγχρονη πορεία του κλάδου ήταν η εδραίωση των πολυεπίπεδων δικτύων και της οπισθοδιάδοσης σφάλματος. Οι εργασίες των Rumelhart, McClelland και συνεργατών στα μέσα της δεκαετίας του 1980 κατέστησαν πρακτικά αξιοποιήσιμη την εκπαίδευση δικτύων με ενδιάμεσα κρυφά επίπεδα. Η μέθοδος backpropagation επέτρεψε τον υπολογισμό του σφάλματος από την έξοδο προς τα εσωτερικά επίπεδα και τη συστηματική προσαρμογή των βαρών, καθιστώντας εφικτή την εκμάθηση πολύ πιο σύνθετων συναρτήσεων από όσες μπορούσαν να χειριστούν τα απλούστερα προηγούμενα μοντέλα. Από το σημείο αυτό και έπειτα, τα νευρωνικά δίκτυα απέκτησαν σαφή υπολογιστική και ερευνητική δυναμική.

Κατά τις επόμενες δεκαετίες, η πρόοδος στην υπολογιστική ισχύ, η διαθεσιμότητα μεγάλων συνόλων δεδομένων και η ανάπτυξη αποδοτικότερων αλγορίθμων εκπαίδευσης οδήγησαν στη μετάβαση από τα κλασικά νευρωνικά δίκτυα στη σύγχρονη βαθιά μάθηση. Αναπτύχθηκαν εξειδικευμένες αρχιτεκτονικές, όπως πολυεπίπεδα δίκτυα πρόσθιας τροφοδότησης, συνελκτικά δίκτυα, αναδρομικά δίκτυα και, αργότερα, αρχιτεκτονικές προσοχής όπως οι Transformers. Παρότι οι αρχιτεκτονικές αυτές διαφέρουν σημαντικά ως προς τη δομή και τον τρόπο επεξεργασίας της πληροφορίας, όλες στηρίζονται στη βασική αρχή ότι η γνώση κωδικοποιείται στα βάρη ενός μεγάλου αριθμού διασυνδεδεμένων υπολογιστικών μονάδων.

Για τη δική μου εργασία, η ιστορική αυτή αναδρομή έχει ιδιαίτερη σημασία, επειδή το πρόβλημα της πρόβλεψης Over/Under 2.5 γκολ προσεγγίζεται μέσα από σύγχρονες νευρωνικές μεθόδους, με κύρια έμφαση σε Transformer-based μοντέλα. Η κατανόηση, όμως, αυτών των νεότερων μοντέλων προϋποθέτει γνώση των εννοιών που προηγήθηκαν: της αναπαράστασης μέσω βαρών, της μάθησης μέσω προσαρμογής συνδέσεων, της σημασίας των κρυφών επιπέδων και της ανάγκης για κατάλληλους μηχανισμούς εκπαίδευσης. Με άλλα λόγια, τα σημερινά

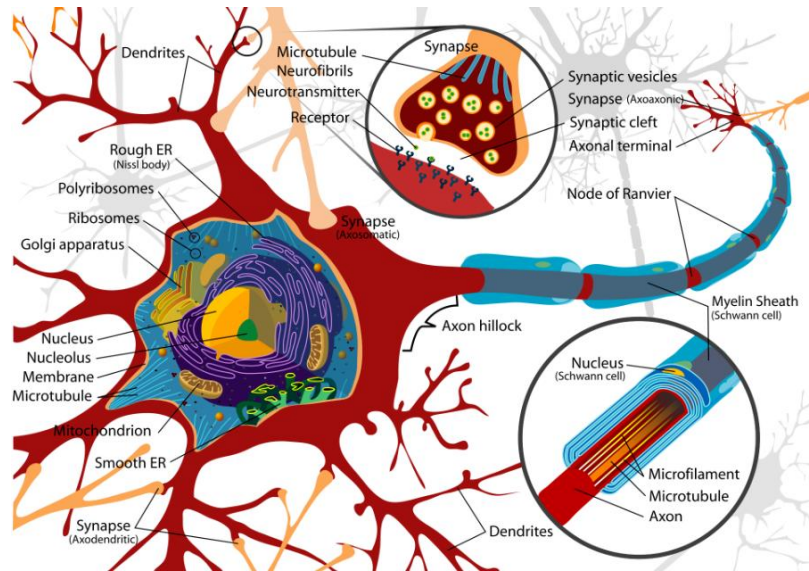
μοντέλα δεν εμφανίστηκαν αποκομμένα από το παρελθόν, αλλά αποτελούν συνέχεια μιας μακράς ερευνητικής διαδρομής.

Συνολικά, η ιστορία των νευρωνικών δικτύων δείχνει ότι η πρόοδος στον χώρο στηρίχθηκε τόσο στη βιολογική έμπνευση όσο και στη σταδιακή μαθηματική και υπολογιστική ωρίμανση των μοντέλων. Από τους πρώτους αφηρημένους νευρώνες έως τις σύγχρονες βαθιές αρχιτεκτονικές, ο βασικός στόχος παραμένει ο ίδιος: η ανάπτυξη συστημάτων ικανών να μαθαίνουν από δεδομένα και να αναπαριστούν σύνθετες σχέσεις. Στο πλαίσιο της παρούσας διπλωματικής, αυτή η εξέλιξη αποτελεί το θεωρητικό υπόβαθρο πάνω στο οποίο στηρίζεται η μελέτη των μοντέλων που χρησιμοποιούνται για την πρόβλεψη ποδοσφαιρικών αγώνων.

2.2 Νευρώνες του Εγκεφάλου

Ο βιολογικός νευρώνας αποτελεί τη βασική λειτουργική μονάδα του νευρικού συστήματος και το αρχικό πρότυπο πάνω στο οποίο βασίστηκε η ανάπτυξη των τεχνητών νευρωνικών δικτύων. Παρότι τα σύγχρονα υπολογιστικά μοντέλα δεν επιδιώκουν βιολογική ακρίβεια, η βασική αναλογία παραμένει σημαντική: η επεξεργασία πληροφορίας προκύπτει από τη συνεργασία πολλών απλών μονάδων.

Ένας νευρώνας αποτελείται από τρία βασικά μέρη. Οι δενδρίτες συλλέγουν ερεθίσματα, το κυτταρικό σώμα επεξεργάζεται την πληροφορία, και ο άξονας μεταφέρει το σήμα προς άλλους νευρώνες. Η επικοινωνία πραγματοποιείται μέσω των συνάψεων, όπου η μετάδοση της πληροφορίας γίνεται με τη βοήθεια νευροδιαβιβαστών, επιτρέποντας τη διάδοση σημάτων σε εκτεταμένα δίκτυα.



Εικόνα 2.2: Βασική δομή βιολογικού νευρώνα.

Η λειτουργία του εγκεφάλου βασίζεται στη συντονισμένη δράση μεγάλου αριθμού νευρώνων. Η μάθηση και η μνήμη συνδέονται με τη μεταβολή της ισχύος των συνδέσεων μεταξύ τους, δηλαδή με την ενίσχυση ή αποδυνάμωση των συναπτικών σχέσεων. Με αυτόν τον τρόπο, η γνώση αναπαρίσταται ως πρότυπο ενεργοποίησης μέσα στο δίκτυο.

Από υπολογιστική σκοπιά, η παραπάνω περιγραφή προσφέρει τη βασική διαίσθηση για τα τεχνητά νευρωνικά δίκτυα. Οι είσοδοι ενός τεχνητού νευρώνα αντιστοιχούν στα εισερχόμενα σήματα, τα βάρη εκφράζουν τη σημασία των συνδέσεων, ενώ η έξοδος αντιπροσωπεύει τη μεταδιδόμενη πληροφορία προς επόμενες μονάδες.

Στο πλαίσιο της παρούσας εργασίας, η αναφορά στους βιολογικούς νευρώνες λειτουργεί ως θεωρητική βάση για την κατανόηση των τεχνητών νευρωνικών δικτύων και, κατ' επέκταση, των σύγχρονων αρχιτεκτονικών όπως τα Transformer μοντέλα, τα οποία αξιοποιούνται για την πρόβλεψη ποδοσφαιρικών αγώνων.

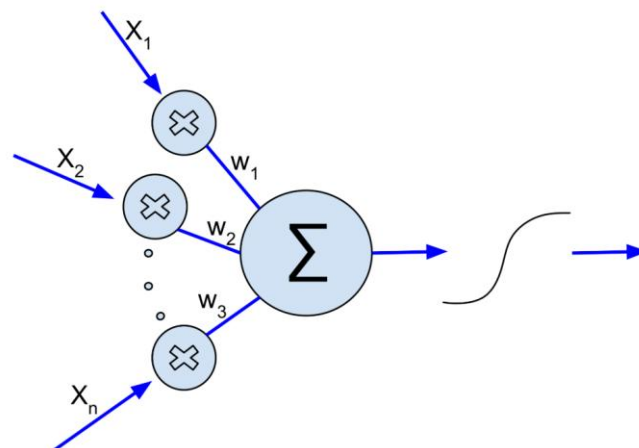
2.3 Τεχνητά Νευρωνικά Δίκτυα

Τα τεχνητά νευρωνικά δίκτυα αποτελούν υπολογιστικά μοντέλα εμπνευσμένα από τη βασική λειτουργία του βιολογικού νευρώνα. Στόχος τους δεν είναι η βιολογική προσομοίωση, αλλά η αξιοποίηση της κεντρικής ιδέας ότι πολύπλοκες σχέσεις μπορούν να προκύψουν από τη συνεργασία πολλών απλών, διασυνδεδεμένων μονάδων. Για τον λόγο αυτό, αποτελούν σήμερα

μία από τις βασικότερες κατηγορίες μοντέλων μηχανικής μάθησης, με εφαρμογές σε προβλήματα ταξινόμησης, παλινδρόμησης και πρόβλεψης.

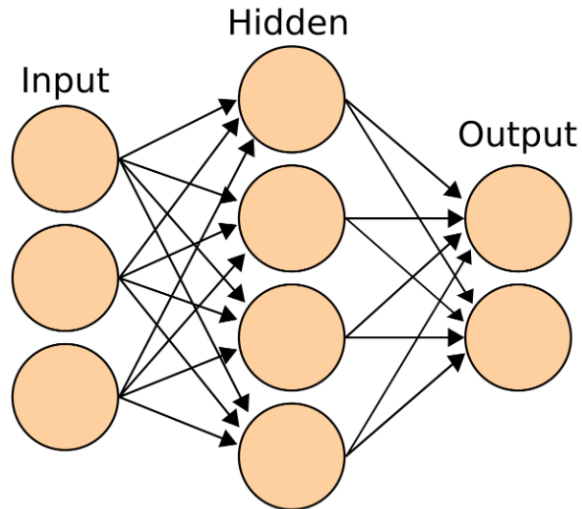
Σε ένα νευρωνικό δίκτυο, κάθε κόμβος λειτουργεί ως τεχνητός νευρώνας και συνδέεται με άλλους μέσω βαρών, τα οποία καθορίζουν τη σημασία κάθε εισόδου. Το δίκτυο δεν γνωρίζει εκ των προτέρων τις σχέσεις των δεδομένων, αλλά τις μαθαίνει προσαρμόζοντας τα βάρη του κατά τη διαδικασία εκπαίδευσης. Με τον τρόπο αυτό, η γνώση αποθηκεύεται κατανεμημένα στις συνδέσεις του δικτύου.

Η λειτουργία ενός τεχνητού νευρώνα περιλαμβάνει τρία βασικά στάδια: τον υπολογισμό ενός σταθμισμένου αθροίσματος των εισόδων, την προσθήκη ενός όρου μετατόπισης (bias) και την εφαρμογή μιας συνάρτησης ενεργοποίησης. Η συνάρτηση ενεργοποίησης, όπως η ReLU ή η sigmoid, εισάγει μη γραμμικότητα στο μοντέλο, επιτρέποντάς του να προσεγγίζει πολύπλοκες σχέσεις μεταξύ εισόδων και εξόδου.



Εικόνα 2.3: Μοντέλο τεχνητού νευρώνα.

Οι νευρώνες οργανώνονται σε επίπεδα, σχηματίζοντας δίκτυα με επίπεδο εισόδου, δύο κρυφά επίπεδα και επίπεδο εξόδου. Κατά την εκπαίδευση, το δίκτυο συγκρίνει την πρόβλεψή του με την πραγματική τιμή και προσαρμόζει τα βάρη του ώστε να μειώνει το σφάλμα, μέσω διαδικασιών βελτιστοποίησης.



Εικόνα 2.4: Αρχιτεκτονική πολυεπίπεδου νευρωνικού δικτύου.

Η βασική δύναμη των νευρωνικών δικτύων έγκειται στην ικανότητά τους να μαθαίνουν σύνθετες και μη γραμμικές σχέσεις χωρίς την ανάγκη ρητής διατύπωσης κανόνων. Στο πλαίσιο της παρούσας εργασίας, αυτό είναι ιδιαίτερα σημαντικό, καθώς η πρόβλεψη Over/Under 2.5 γκολ εξαρτάται από πολλαπλούς και αλληλεπιδρώντες παράγοντες, όπως η επιθετική και αμυντική απόδοση, η πρόσφατη φόρμα και οι στοιχηματικές πιθανότητες.

Παρά τα πλεονεκτήματά τους, η απόδοση των νευρωνικών δικτύων εξαρτάται από την ποιότητα των δεδομένων, την αναπαράσταση εισόδου και την αρχιτεκτονική του μοντέλου. Για τον λόγο αυτό, στην παρούσα εργασία δίνεται ιδιαίτερη έμφαση στη σωστή κατασκευή ακολουθιακών δεδομένων και στη χρήση κατάλληλων αρχιτεκτονικών, όπως τα Transformer μοντέλα, τα οποία επεκτείνουν τη βασική αυτή ιδέα σε πιο σύνθετες μορφές επεξεργασίας πληροφορίας.

2.4 Αρχιτεκτονικές Νευρωνικών Δικτύων

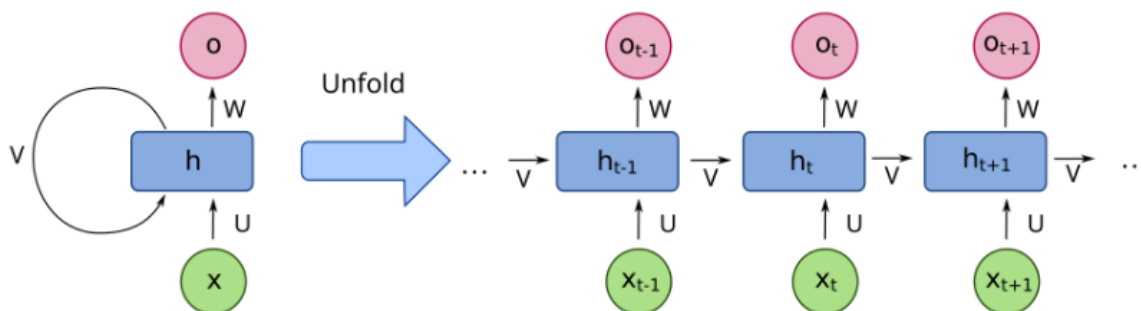
Η αρχιτεκτονική ενός νευρωνικού δικτύου αναφέρεται στον τρόπο με τον οποίο οργανώνονται και συνδέονται οι τεχνητοί νευρώνες. Η επιλογή της αποτελεί κρίσιμο παράγοντα, καθώς καθορίζει τόσο τη διαδικασία εκπαίδευσης όσο και το είδος των σχέσεων που το μοντέλο μπορεί να μάθει.

Η πιο βασική κατηγορία είναι τα δίκτυα εμπρόσθιας τροφοδότησης (feedforward neural networks), στα οποία η πληροφορία ρέει από το επίπεδο εισόδου προς τα κρυφά επίπεδα και

τελικά στο επίπεδο εξόδου, χωρίς αναδρομικές συνδέσεις. Τα μοντέλα αυτά είναι κατάλληλα για προβλήματα όπου τα δεδομένα μπορούν να αναπαρασταθούν ως σταθερού μήκους διανύσματα χαρακτηριστικών.

Μια απλή μορφή feedforward δικτύου είναι το δίκτυο με ένα κρυφό επίπεδο, ενώ πιο σύνθετες εκδοχές περιλαμβάνουν πολλαπλά κρυφά επίπεδα (deep neural networks). Η αύξηση του βάθους επιτρέπει την εκμάθηση πιο αφηρημένων και ιεραρχικών αναπαραστάσεων, αλλά συνοδεύεται από αυξημένες υπολογιστικές απαιτήσεις και κίνδυνο υπερπροσαρμογής.

Μια διαφορετική κατηγορία αποτελείται από τα αναδρομικά ή επαναληπτικά νευρωνικά δίκτυα (Recurrent Neural Networks - RNN), τα οποία εισάγουν την έννοια της χρονικής εξάρτησης. Σε αυτά τα μοντέλα, η πληροφορία από προηγούμενα χρονικά βήματα επανεισάγεται στο δίκτυο, επιτρέποντας τη μοντελοποίηση ακολουθιακών δεδομένων.



Εικόνα 2.5: Αναδίπλωση αναδρομικού νευρωνικού δικτύου στον χρόνο.

Η δυνατότητα αυτή είναι ιδιαίτερα σημαντική σε προβλήματα όπου η σειρά των παρατηρήσεων επηρεάζει την πρόβλεψη. Στο πλαίσιο της ποδοσφαιρικής ανάλυσης, για παράδειγμα, η πρόσφατη ακολουθία αγώνων, η φόρμα μιας ομάδας και η χρονική εξέλιξη των επιδόσεών της αποτελούν κρίσιμους παράγοντες που δεν μπορούν να αποτυπωθούν πλήρως με στατικά μοντέλα.

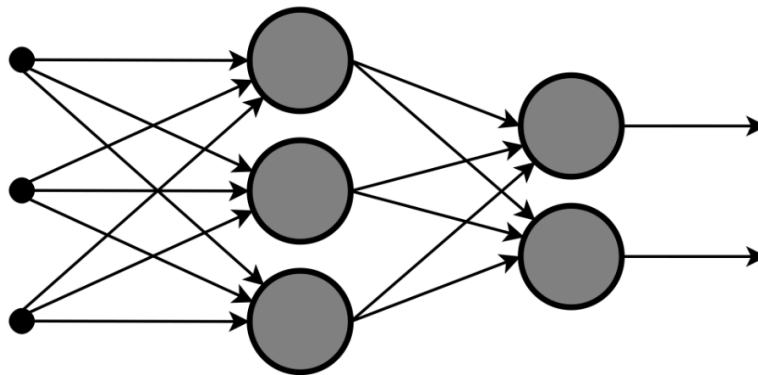
Παρά τα πλεονεκτήματά τους, τα RNN παρουσιάζουν δυσκολίες στην εκμάθηση μακροχρόνιων εξαρτήσεων, γεγονός που οδήγησε στην ανάπτυξη πιο εξελιγμένων αρχιτεκτονικών για την επεξεργασία ακολουθιών. Στο πλαίσιο αυτό εντάσσονται τα μοντέλα προσοχής (attention-based models) και ιδιαίτερα οι Transformers, οι οποίοι επιτρέπουν την αποδοτικότερη αξιοποίηση της πληροφορίας μέσα σε μια ακολουθία.

Δεν υπάρχει μία καθολικά βέλτιστη αρχιτεκτονική για όλα τα προβλήματα· η καταλληλότητά της εξαρτάται από τη μορφή των δεδομένων και τον στόχο της πρόβλεψης. Στην παρούσα εργασία, η έμφαση δίνεται σε ακολουθιακές αναπαραστάσεις αγώνων και σε αρχιτεκτονικές που μπορούν να αξιοποιήσουν τη χρονική πληροφορία, οδηγώντας στη χρήση Transformer-based μοντέλων για την πρόβλεψη Over/Under 2.5 γκολ.

2.5 MLP με Backpropagation

Τα Πολυεπίεδα Δίκτυα Αντίληψης (Multilayer Perceptrons - MLP) αποτελούν μία από τις πιο κλασικές αρχιτεκτονικές τεχνητών νευρωνικών δικτύων και χρησιμοποιούνται ευρέως για τη μοντελοποίηση μη γραμμικών σχέσεων μεταξύ εισόδων και εξόδου. Σε αντίθεση με το απλό perceptron, τα MLP περιλαμβάνουν ένα ή περισσότερα κρυφά επίπεδα, γεγονός που τους επιτρέπει να μαθαίνουν πιο σύνθετα πρότυπα.

Η δομή ενός MLP αποτελείται από επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα και επίπεδο εξόδου. Σε ένα πλήρως συνδεδεμένο δίκτυο, κάθε νευρώνας συνδέεται με όλους τους νευρώνες του επόμενου επιπέδου μέσω βαρών, τα οποία καθορίζουν τη συμβολή κάθε εισόδου στην τελική πρόβλεψη.



Εικόνα 2.6 Αρχιτεκτονική πολυεπίεδου νευρωνικού δικτύου.

Η λειτουργία του δικτύου βασίζεται στη διαδικασία της εμπρόσθιας τροφοδότησης (forward propagation), κατά την οποία τα δεδομένα διέρχονται διαδοχικά από τα επίπεδα. Σε κάθε βήμα υπολογίζεται ένα σταθμισμένο άθροισμα των εισόδων, προστίθεται ένας όρος μετατόπισης (bias) και εφαρμόζεται συνάρτηση ενεργοποίησης, επιτρέποντας στο δίκτυο να μοντελοποιεί μη γραμμικές σχέσεις.

Η εκπαίδευση πραγματοποιείται μέσω του αλγορίθμου οπισθοδιάδοσης σφάλματος (backpropagation). Αφού υπολογιστεί το σφάλμα της πρόβλεψης, αυτό διαδίδεται προς τα πίσω στο δίκτυο και χρησιμοποιείται για την ενημέρωση των βαρών μέσω μεθόδων βελτιστοποίησης, όπως ο gradient descent. Η επαναληπτική εφαρμογή των φάσεων forward και backward propagation επιτρέπει στο μοντέλο να βελτιώνει σταδιακά την απόδοσή του.

Οι συναρτήσεις ενεργοποίησης διαδραματίζουν καθοριστικό ρόλο, καθώς εισάγουν μη γραμμικότητα στο μοντέλο. Στη σύγχρονη πρακτική χρησιμοποιούνται κυρίως συναρτήσεις όπως η ReLU, οι οποίες επιτρέπουν πιο αποδοτική εκπαίδευση σε σχέση με παλαιότερες επιλογές όπως η sigmoid.

Τα MLP είναι ιδιαίτερα κατάλληλα όταν τα δεδομένα μπορούν να αναπαρασταθούν ως διανύσματα σταθερού μήκους. Στο πλαίσιο της παρούσας εργασίας, αυτό αντιστοιχεί σε περιπτώσεις όπου τα pre-match χαρακτηριστικά (όπως στατιστικά απόδοσης, δείκτες φόρμας και bookmaker probabilities) συμπυκνώνονται σε ενιαίο διάνυσμα εισόδου.

Για τον λόγο αυτό, τα MLP αποτελούν σημαντικό σημείο αναφοράς (baseline), καθώς μπορούν να αποδώσουν ικανοποιητικά χωρίς να απαιτούν ρητή μοντελοποίηση της χρονικής διάστασης. Ωστόσο, η προσέγγιση αυτή δεν αξιοποιεί την ακολουθιακή φύση των δεδομένων, γεγονός που περιορίζει την ικανότητά τους να συλλάβουν χρονικές εξαρτήσεις μεταξύ αγώνων.

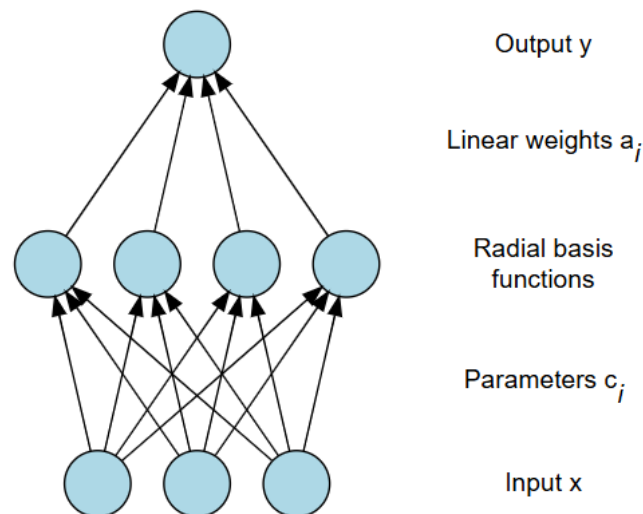
Συνεπώς, ενώ τα MLP αποτελούν ένα ισχυρό και ευρέως χρησιμοποιούμενο μοντέλο, η μετάβαση σε αρχιτεκτονικές που λαμβάνουν υπόψη τη χρονική πληροφορία, όπως τα Transformer-based μοντέλα, είναι απαραίτητη για την αποτελεσματικότερη αντιμετώπιση του προβλήματος της πρόβλεψης Over/Under 2.5 γκολ.

2.6 Δίκτυα Radial Basis Function

Τα δίκτυα Radial Basis Function (RBF) αποτελούν μια ιδιαίτερη κατηγορία τεχνητών νευρωνικών δικτύων εποπτευόμενης μάθησης, τα οποία χρησιμοποιούνται κυρίως για προβλήματα ταξινόμησης, παλινδρόμησης και προσέγγισης συναρτήσεων. Σε αντίθεση με τα κλασικά MLP, τα RBF δίκτυα βασίζονται στην ιδέα ότι η απόκριση του κρυφού επιπέδου δεν εξαρτάται από έναν καθολικό γραμμικό μετασχηματισμό της εισόδου, αλλά από την απόσταση της εισόδου από συγκεκριμένα σημεία αναφοράς, τα οποία ονομάζονται κέντρα. Με αυτή τη

λογική, τα RBF μοντέλα έχουν περισσότερο "τοπικό" χαρακτήρα, καθώς κάθε κρυφός νευρώνας ενεργοποιείται έντονα μόνο όταν η είσοδος βρίσκεται κοντά στο αντίστοιχο κέντρο του.

Η βασική αρχιτεκτονική ενός RBF δικτύου περιλαμβάνει τρία επίπεδα: το επίπεδο εισόδου, το κρυφό επίπεδο και το επίπεδο εξόδου. Το επίπεδο εισόδου μεταφέρει τα χαρακτηριστικά του προβλήματος χωρίς ιδιαίτερη επεξεργασία. Στο κρυφό επίπεδο βρίσκονται οι νευρώνες ακτινικής βάσης, οι οποίοι υπολογίζουν την απόσταση του διανύσματος εισόδου από τα κέντρα τους. Η τιμή αυτής της απόστασης εισάγεται σε μια μη γραμμική συνάρτηση βάσης, συνήθως Γκαουσιανή, ώστε να παραχθεί η ενεργοποίηση του νευρώνα. Τέλος, το επίπεδο εξόδου συνδυάζει γραμμικά τις αποκρίσεις των κρυφών νευρώνων για να δώσει την τελική πρόβλεψη.



Εικόνα 2.7: Βασική αρχιτεκτονική δικτύου Radial Basis Function

Η συνάρτηση ακτινικής βάσης είναι το στοιχείο που δίνει στο δίκτυο τον ιδιαίτερο χαρακτήρα του. Η πιο συνηθισμένη επιλογή είναι η Γκαουσιανή συνάρτηση, η οποία λαμβάνει μεγάλες τιμές όταν η είσοδος βρίσκεται κοντά στο κέντρο και μικρές τιμές όταν απομακρύνεται από αυτό. Κατ' αυτόν τον τρόπο, κάθε κρυφός νευρώνας λειτουργεί σαν τοπικός ανιχνευτής περιοχών του χώρου εισόδου. Το συνολικό δίκτυο δεν προσπαθεί να μάθει μια ενιαία καθολική αναπαράσταση όπως σε ένα βαθύ MLP, αλλά να συνθέσει την τελική έξοδο από ένα σύνολο τοπικών αποκρίσεων.

Η λογική αυτή συνδέεται και με την ευρύτερη ιδέα ότι ένα πολύπλοκο, μη γραμμικά διαχωρίσιμο πρόβλημα μπορεί να γίνει ευκολότερα διαχειρίσιμο αν μετασχηματιστεί σε χώρο μεγαλύτερης διαστατικότητας. Τα RBF δίκτυα επιτυγχάνουν ακριβώς έναν τέτοιο μετασχηματισμό, καθώς η είσοδος αναπαρίσταται πλέον μέσω των αποστάσεων της από πολλαπλά κέντρα. Στον νέο αυτό χώρο, η τελική έξοδος μπορεί συχνά να προκύπτει με απλούστερο, σχεδόν γραμμικό τρόπο.

Η εκπαίδευση ενός RBF δικτύου μπορεί να πραγματοποιηθεί με δύο βασικές προσεγγίσεις. Η πρώτη είναι η προσέγγιση σταθερών κέντρων, όπου τα κέντρα επιλέγονται εκ των προτέρων και παραμένουν αμετάβλητα κατά την εκπαίδευση, ενώ προσαρμόζονται μόνο τα βάρη του επιπέδου εξόδου. Σε αυτή την περίπτωση, η ποιότητα του μοντέλου εξαρτάται ιδιαίτερα από τη σωστή επιλογή του αριθμού και της θέσης των κέντρων, καθώς και από τη διασπορά της Γκαουσιανής συνάρτησης. Η δεύτερη προσέγγιση είναι η εκπαίδευση με μεταβλητά κέντρα, όπου προσαρμόζονται όχι μόνο τα βάρη εξόδου, αλλά και τα ίδια τα κέντρα και οι διασπορές τους, συνήθως μέσω gradient-based βελτιστοποίησης.

Η μέθοδος των σταθερών κέντρων είναι εννοιολογικά απλούστερη και συχνά χρήσιμη σε προβλήματα όπου ζητείται ακριβής παρεμβολή ή όπου ο χώρος εισόδου μπορεί να καλυφθεί ικανοποιητικά από προκαθορισμένα κέντρα. Αντίθετα, η μέθοδος των μεταβλητών κέντρων προσφέρει μεγαλύτερη ευελιξία, επειδή επιτρέπει στο δίκτυο να προσαρμόζει την ίδια τη γεωμετρία της αναπαράστασής του. Η ευελιξία αυτή μπορεί να βελτιώσει την απόδοση, αλλά συνοδεύεται και από μεγαλύτερη πολυπλοκότητα στην εκπαίδευση, καθώς πρέπει να βελτιστοποιηθούν περισσότερες παράμετροι.

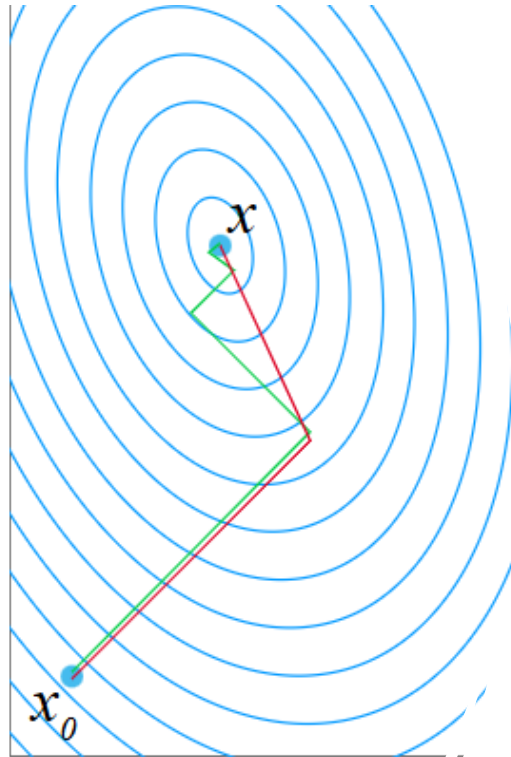
Σε σύγκριση με τα MLP, τα RBF δίκτυα έχουν ορισμένα χαρακτηριστικά πλεονεκτήματα. Συχνά είναι ευκολότερο να ερμηνευθούν, καθώς η λειτουργία των κρυφών νευρώνων συνδέεται άμεσα με περιοχές του χώρου εισόδου. Επιπλέον, το επίπεδο εξόδου είναι συνήθως γραμμικό, κάτι που κάνει την τελική χαρτογράφηση πιο απλή. Από την άλλη πλευρά, η απόδοσή τους εξαρτάται σε μεγάλο βαθμό από τη σωστή επιλογή του αριθμού των κέντρων, από τη χωρική κατανομή τους και από την κατάλληλη ρύθμιση των διασπορών. Αν αυτά δεν επιλεγούν σωστά, το δίκτυο μπορεί είτε να αδυνατεί να συλλάβει τη δομή του προβλήματος είτε να υπερπροσαρμόζεται σε τοπικές ιδιαιτερότητες των δεδομένων.

Στο πλαίσιο της παρούσας διπλωματικής, τα RBF δίκτυα έχουν ενδιαφέρον κυρίως ως εναλλακτική κλασική νευρωνική αρχιτεκτονική απέναντι στα MLP και στα νεότερα Transformer-based μοντέλα. Η ύπαρξή τους στη θεωρητική ανασκόπηση είναι σημαντική, επειδή αναδεικνύει ότι η νευρωνική μάθηση δεν περιορίζεται σε μία μόνο οικογένεια αρχιτεκτονικών, αλλά περιλαμβάνει διαφορετικές προσεγγίσεις ως προς τον τρόπο αναπαράστασης και επεξεργασίας της πληροφορίας. Ειδικά σε προβλήματα όπως η πρόβλεψη ποδοσφαιρικών αποτελεσμάτων, όπου η πληροφορία μπορεί να περιγραφεί είτε με στατικά engineered χαρακτηριστικά είτε με χρονικά οργανωμένες αναπαραστάσεις, η σύγκριση διαφορετικών οικογενειών μοντέλων είναι επιστημονικά χρήσιμη.

Συνολικά, τα RBF δίκτυα αποτελούν μια ενδιαφέρουσα και ιστορικά σημαντική μορφή νευρωνικών μοντέλων. Η βασική τους ιδέα είναι ότι η πρόβλεψη μπορεί να στηριχθεί στη μέτρηση τοπικής ομοιότητας μεταξύ της εισόδου και ενός συνόλου κέντρων αναφοράς. Παρότι σήμερα οι πιο σύγχρονες αρχιτεκτονικές έχουν κυριαρχήσει σε πολλά πεδία, τα RBF παραμένουν ουσιώδη για την κατανόηση της ποικιλίας των νευρωνικών προσεγγίσεων και για τη θεωρητική σύγκριση με άλλα μοντέλα. Η σύγκριση αυτή συμπληρώνεται στην επόμενη ενότητα με μια πιο προχωρημένη συζήτηση γύρω από τις στρατηγικές βελτιστοποίησης της εκπαίδευσης.

2.7 Hessian Free Optimization

Η Hessian Free Optimization (HFO) αποτελεί μία μέθοδο βελτιστοποίησης δεύτερης τάξης, η οποία αξιοποιεί πληροφορία καμπυλότητας της συνάρτησης σφάλματος, σε αντίθεση με μεθόδους πρώτης τάξης όπως η καθόδου κλίσης (gradient descent), οι οποίες βασίζονται αποκλειστικά στην κλίση.



Εικόνα 2.8: Σύγκριση διαδρομής βελτιστοποίησης πρώτης και δεύτερης τάξης.

Η βασική ιδέα προέρχεται από τη μέθοδο Newton, όπου η συνάρτηση στόχου προσεγγίζεται τοπικά μέσω ενός παραβολικού μοντέλου. Αν και η προσέγγιση αυτή μπορεί να οδηγήσει σε ταχύτερη σύγκλιση, η πλήρης εφαρμογή της είναι πρακτικά δύσκολη σε μεγάλα νευρωνικά δίκτυα, λόγω του υψηλού κόστους υπολογισμού και αποθήκευσης του πίνακα Hessian.

Η HFO αντιμετωπίζει το πρόβλημα αυτό αποφεύγοντας τον ρητό υπολογισμό του Hessian και βασιζόμενη σε γινόμενα του με διανύσματα. Με τον τρόπο αυτό, επιτυγχάνει την αξιοποίηση πληροφορίας δεύτερης τάξης με σημαντικά μικρότερο υπολογιστικό κόστος. Η επίλυση του αντίστοιχου τοπικού υποπροβλήματος πραγματοποιείται συνήθως μέσω της μεθόδου Conjugate Gradient, η οποία επιτρέπει τον εντοπισμό κατάλληλων κατευθύνσεων βελτιστοποίησης χωρίς πλήρη επίλυση του συστήματος.

Η χρήση τέτοιων μεθόδων μπορεί να προσφέρει πλεονεκτήματα σε περιπτώσεις όπου οι gradient-based τεχνικές παρουσιάζουν αργή ή ασταθή σύγκλιση, ιδιαίτερα σε βαθύτερα δίκτυα. Ωστόσο, στην πράξη οι μέθοδοι πρώτης τάξης παραμένουν πιο διαδεδομένες λόγω απλότητας και καλύτερης κλιμάκωσης σε μεγάλα δεδομένα.

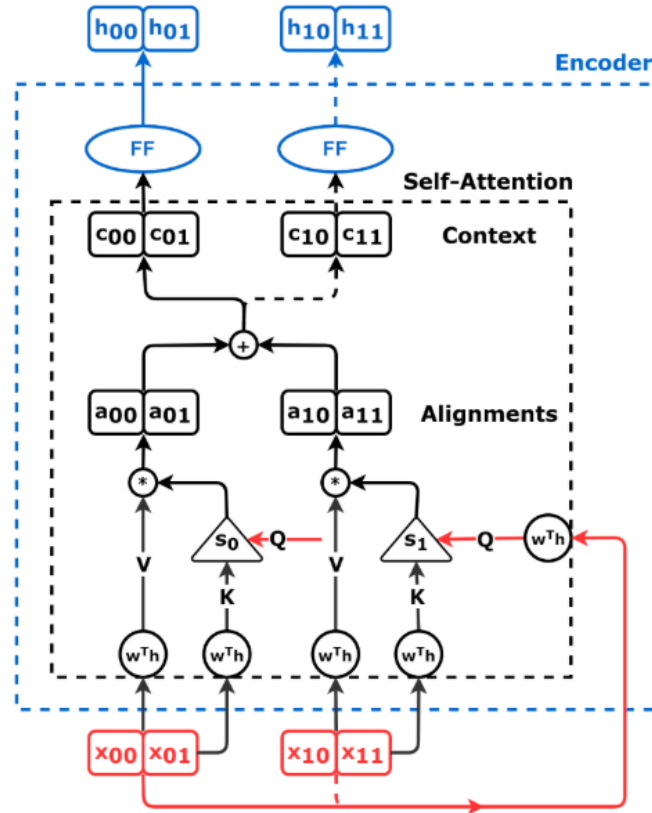
Στο πλαίσιο της παρούσας εργασίας, η HFO δεν χρησιμοποιείται ως κύρια μέθοδος εκπαίδευσης, αλλά αναφέρεται για λόγους πληρότητας και σύγκρισης. Η παρουσία της αναδεικνύει ότι η απόδοση ενός νευρωνικού μοντέλου εξαρτάται όχι μόνο από την αρχιτεκτονική του, αλλά και από τον αλγόριθμο βελτιστοποίησης. Η μελέτη τέτοιων προσεγγίσεων συμβάλλει σε μια πιο ολοκληρωμένη κατανόηση του πεδίου πριν από τη μετάβαση σε σύγχρονες αρχιτεκτονικές όπως οι Transformers.

2.8 Αρχιτεκτονική Transformer

Η αρχιτεκτονική Transformer αποτελεί το κεντρικό νευρωνικό υπόδειγμα της παρούσας διπλωματικής, καθώς πάνω σε αυτή στηρίχθηκε ο βασικός ερευνητικός άξονας της εργασίας για την πρόβλεψη Over/Under 2.5 γκολ. Σε αντίθεση με παλαιότερες νευρωνικές αρχιτεκτονικές που βασίζονται κυρίως σε διαδοχική ή τοπική επεξεργασία της πληροφορίας, ο Transformer σχεδιάστηκε ώστε να μπορεί να εξετάζει ταυτόχρονα όλα τα στοιχεία μιας ακολουθίας και να μαθαίνει ποιες σχέσεις μεταξύ τους είναι περισσότερο σημαντικές. Η ιδιότητα αυτή τον καθιστά ιδιαίτερα ελκυστικό για προβλήματα όπου η σειρά, η σχετική θέση και η αλληλεπίδραση διαφορετικών τμημάτων του ιστορικού παίζουν ουσιαστικό ρόλο.

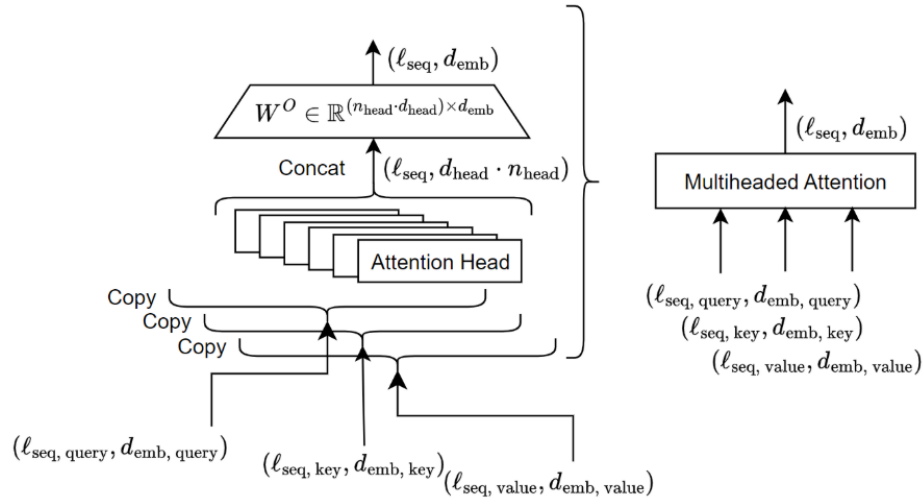
Η βασική καινοτομία του Transformer είναι ο μηχανισμός self-attention. Μέσω αυτού, κάθε στοιχείο μιας ακολουθίας μπορεί να "στρέφει την προσοχή του" προς όλα τα υπόλοιπα στοιχεία και να σταθμίζει δυναμικά τη σημασία τους. Αυτό σημαίνει ότι το μοντέλο δεν είναι υποχρεωμένο να επεξεργάζεται τα δεδομένα αυστηρά βήμα προς βήμα, όπως συμβαίνει στα κλασικά αναδρομικά δίκτυα, αλλά μπορεί να λαμβάνει υπόψη μακρινές εξαρτήσεις με πιο άμεσο τρόπο. Στην πράξη, ο μηχανισμός attention επιτρέπει στο μοντέλο να εντοπίζει ποιες προηγούμενες παρατηρήσεις είναι περισσότερο σχετικές για την τρέχουσα πρόβλεψη.

Η λειτουργία του self-attention βασίζεται στη δημιουργία τριών διαφορετικών αναπαραστάσεων για κάθε στοιχείο εισόδου: των queries, των keys και των values. Με απλά λόγια, το κάθε στοιχείο "ρωτά" ποια άλλα στοιχεία είναι σχετικά με αυτό μέσω του query, η σχετικότητα αξιολογείται συγκρίνοντας queries και keys, και η τελική πληροφορία που αντλείται προκύπτει από τα values. Η στάθμιση αυτή δεν είναι σταθερή αλλά μαθαίνεται από τα δεδομένα κατά την εκπαίδευση. Έτσι, το μοντέλο προσαρμόζει μόνο του σε ποια σημεία της ακολουθίας πρέπει να δώσει έμφαση για το εκάστοτε πρόβλημα.



Εικόνα 2.9: Μηχανισμός self-attention σε Transformer Encoder.

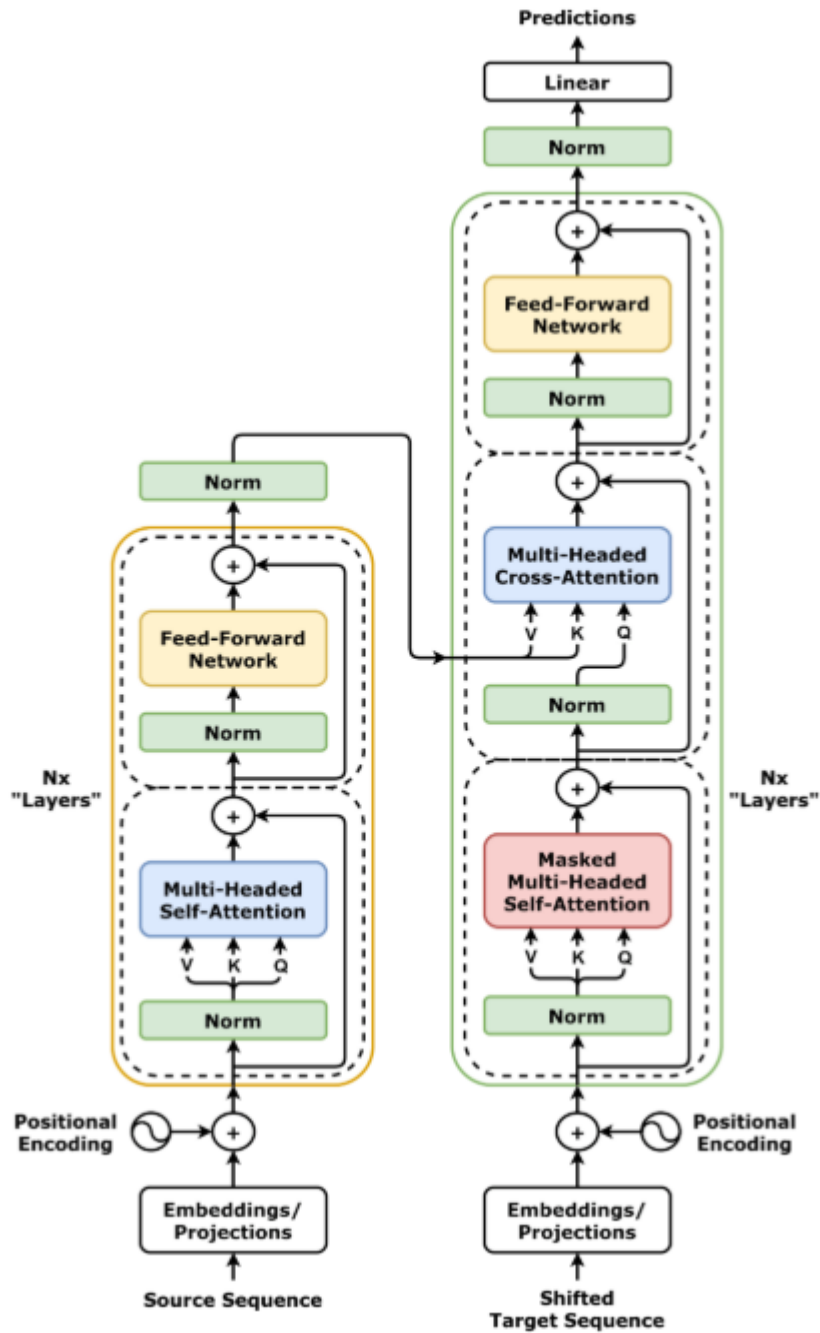
Ένα επιπλέον σημαντικό χαρακτηριστικό είναι το multi-head attention. Αντί να υπολογίζεται μία μόνο σχέση προσοχής, ο Transformer χρησιμοποιεί πολλαπλές "κεφαλές" προσοχής, καθεμία από τις οποίες μπορεί να εστιάζει σε διαφορετικό είδος σχέσης μέσα στην ακολουθία. Με αυτόν τον τρόπο, το μοντέλο αποκτά τη δυνατότητα να εξετάζει ταυτόχρονα διαφορετικές όψεις της ίδιας εισόδου. Σε ένα πρόβλημα ποδοσφαιρικής πρόβλεψης, αυτό είναι θεωρητικά ιδιαίτερα χρήσιμο, επειδή διαφορετικές κεφαλές θα μπορούσαν να εστιάζουν σε πρόσφατη φόρμα, σε διαφορές έδρας, σε επιθετική ή αμυντική συμπεριφορά, ή ακόμη και στη σχέση ανάμεσα σε ιστορικά παιχνίδια των δύο ομάδων και στο τρέχον fixture.



Εικόνα 2.10: Μηχανισμός multi-head attention.

Επειδή ο Transformer δεν διαθέτει από τη φύση του ενσωματωμένη έννοια χρονικής σειράς, απαιτείται η χρήση positional information. Οι αναπαραστάσεις θέσης επιτρέπουν στο μοντέλο να γνωρίζει ποιο στοιχείο προηγείται και ποιο έπεται μέσα στην ακολουθία. Χωρίς αυτές, το σύστημα θα έβλεπε την είσοδο ως απλό σύνολο στοιχείων χωρίς διάταξη. Στο δικό μου project, η χρονική διάσταση είναι καθοριστική, επειδή το pre-match ιστορικό πρέπει να διατηρεί αυστηρή χρονολογική σειρά και να μην αλλοιώνεται από μεταγενέστερη πληροφορία. Η χρήση positional encoding είναι επομένως απαραίτητη προϋπόθεση ώστε ο Transformer να αξιοποιεί πραγματικά το ιστορικό των αγώνων ως ακολουθία.

Σε επίπεδο αρχιτεκτονικής, ο Transformer συνήθως περιλαμβάνει ένα αρχικό στάδιο προβολής των χαρακτηριστικών εισόδου σε έναν χώρο embeddings, ακολουθούμενο από επίπεδα self-attention και feedforward μετασχηματισμών. Στη συνέχεια, οι παραγόμενες αναπαραστάσεις πρέπει να συγκεντρωθούν σε μία τελική περιγραφή του δείγματος, είτε μέσω ειδικού token τύπου [CLS] είτε μέσω τεχνικών pooling όπως ο μέσος όρος των token representations. Πάνω σε αυτή την τελική αναπαράσταση εφαρμόζεται η κεφαλή ταξινόμησης, η οποία στην παρούσα εργασία παράγει μια δυαδική πρόβλεψη για το αν ο αγώνας θα λήξει Over ή Under 2.5.



Εικόνα 2.11: Γενική αρχιτεκτονική Transformer.

Η αρχιτεκτονική αυτή είναι ιδιαίτερα σημαντική για τη συγκεκριμένη διπλωματική, επειδή το υπό μελέτη πρόβλημα έχει έντονο ακολουθιακό χαρακτήρα. Κάθε ποδοσφαιρικός αγώνας δεν είναι ένα ανεξάρτητο γεγονός, αλλά μέρος μιας χρονολογικής αλυσίδας αποτελεσμάτων, φορμών και μεταβαλλόμενων καταστάσεων ομάδας. Θεωρητικά, λοιπόν, ένας Transformer encoder μπορεί να μοντελοποιήσει το ιστορικό της γηπεδούχου και της

φιλοξενούμενης ομάδας με τρόπο πιο ευέλικτο από απλούστερα μοντέλα σταθερών χαρακτηριστικών. Αυτή ακριβώς η δυνατότητα ήταν ένας από τους βασικούς λόγους για τους οποίους ο Transformer επιλέχθηκε ως κύρια οικογένεια μοντέλων στο στην παρούσα εργασία.

Στην παρούσα εργασία, η αρχιτεκτονική Transformer αξιοποιείται ως βασικό νευρωνικό πλαίσιο για την επεξεργασία προαγωνιστικής πληροφορίας ποδοσφαιρικών αγώνων. Η χρήση της δεν περιορίζεται σε μια γενική θεωρητική αναφορά, αλλά ενσωματώνεται στην πειραματική διαδικασία μέσω συγκεκριμένων μορφών αναπαράστασης εισόδου, οι οποίες σχεδιάστηκαν ώστε να περιγράφουν την κατάσταση των δύο ομάδων πριν από την έναρξη του αγώνα.

Μία βασική κατεύθυνση αφορά την ακολουθιακή αναπαράσταση των δεδομένων. Σε αυτή την περίπτωση, κάθε δείγμα μπορεί να περιλαμβάνει προηγούμενους αγώνες της γηπεδούχου ομάδας, προηγούμενους αγώνες της φιλοξενούμενης ομάδας και μία επιπλέον περιγραφή του τρέχοντος αγώνα. Η λογική αυτής της προσέγγισης είναι ότι η πρόσφατη πορεία μιας ομάδας, η επιθετική και αμυντική της συμπεριφορά, καθώς και οι μεταβολές στη φόρμα της, αποτελούν πληροφορία με χρονική δομή. Ο Transformer Encoder μπορεί θεωρητικά να αξιοποιήσει αυτή τη δομή μέσω του μηχανισμού self-attention, σταθμίζοντας διαφορετικά τα στοιχεία της ακολουθίας.

Παράλληλα, εξετάζεται και μια πιο συμπυκνωμένη μορφή εισόδου, στην οποία η προαγωνιστική πληροφορία του αγώνα οργανώνεται σε ένα ενιαίο token. Το token αυτό δεν αντιστοιχεί σε ακατέργαστη πληροφορία ενός μόνο γεγονότος, αλλά σε προσεκτικά κατασκευασμένη αναπαράσταση που συνδυάζει στατιστικά χαρακτηριστικά της γηπεδούχου ομάδας, στατιστικά χαρακτηριστικά της φιλοξενούμενης ομάδας και πληροφορία της αγοράς που προκύπτει από τις αποδόσεις των bookmakers. Με αυτόν τον τρόπο, η ιστορική συμπεριφορά των ομάδων συμπυκνώνεται σε προαγωνιστικά χαρακτηριστικά σταθερού μήκους, τα οποία μπορούν να δοθούν ως είσοδος στον Transformer.

Σε τεχνικό επίπεδο, τα αριθμητικά χαρακτηριστικά εισόδου προβάλλονται αρχικά σε χώρο embeddings, ώστε να αποκτήσουν την κατάλληλη διάσταση για επεξεργασία από τον Transformer Encoder. Στις ακολουθιακές εκδοχές προστίθεται πληροφορία θέσης μέσω positional encoding, ώστε το μοντέλο να μπορεί να διακρίνει τη σειρά των προηγούμενων αγώνων. Στη συνέχεια, οι αναπαραστάσεις εισέρχονται στον Transformer Encoder και η τελική έξοδος οδηγείται σε κεφαλή ταξινόμησης, η οποία παράγει την πιθανότητα ο αγώνας να λήξει σε Over 2.5.

Η χρήση του Transformer στην παρούσα εργασία είναι επομένως διπλή. Από τη μία πλευρά, επιτρέπει τη διερεύνηση της χρονικής πληροφορίας μέσα από ακολουθίες προηγούμενων αγώνων. Από την άλλη, παρέχει ένα ευέλικτο νευρωνικό πλαίσιο για την επεξεργασία πλούσιων προαγωνιστικών χαρακτηριστικών του τρέχοντος αγώνα. Και στις δύο περιπτώσεις, ο στόχος παραμένει ο ίδιος: η αξιοποίηση διαθέσιμης pre-match πληροφορίας για τη δυαδική πρόβλεψη Over/Under 2.5, χωρίς χρήση στοιχείων που προκύπτουν μετά την έναρξη του αγώνα.

Με αυτόν τον τρόπο, ο Transformer αποτελεί το βασικό μοντέλο βαθιάς μάθησης της παρούσας διπλωματικής. Η σημασία του δεν περιορίζεται μόνο στην ικανότητά του να επεξεργάζεται ακολουθίες, αλλά και στη δυνατότητά του να ενσωματώνεται σε διαφορετικές μορφές αναπαράστασης εισόδου, επιτρέποντας τη συστηματική σύγκριση μεταξύ ακολουθιακής και συμπυκνωμένης προαγωνιστικής πληροφορίας.

Κεφάλαιο 3

Σχεδιασμός και Υλοποίηση

3.1 Δεδομένα Εισόδου

3.2 Το πρωτάθλημα της Championship

3.3 Λήψη και επεξεργασία δεδομένων

3.4 Κανονικοποίηση Τιμών

3.5 Σχεδιασμός και Υλοποίηση του Δυκτίου

3.1 Δεδομένα εισόδου

Στην ενότητα αυτή παρουσιάζονται τα δεδομένα εισόδου που χρησιμοποιήθηκαν για την εκπαίδευση και αξιολόγηση των μοντέλων πρόβλεψης Over/Under 2.5 γκολ. Η περιγραφή των δεδομένων δεν περιορίζεται μόνο στην πηγή από την οποία αντλήθηκαν, αλλά επεκτείνεται στον τρόπο επιλογής, επεξεργασίας και μετατροπής τους σε κατάλληλη προαγωνιστική αναπαράσταση για τα μοντέλα μηχανικής μάθησης.

Η βασική πηγή των δεδομένων είναι η ιστοσελίδα football-data.co.uk, η οποία παρέχει ιστορικά αρχεία ποδοσφαιρικών αγώνων σε μορφή CSV. Στην παρούσα εργασία χρησιμοποιήθηκαν δεδομένα της αγγλικής Championship από πολλαπλές αγωνιστικές περιόδους, καλύπτοντας ένα μεγάλο χρονικό εύρος αγώνων. Η επιλογή της Championship έγινε επειδή το συγκεκριμένο πρωτάθλημα προσφέρει μεγάλο αριθμό αγώνων ανά σεζόν, επαρκές ιστορικό βάθος και διαθέσιμα πεδία που σχετίζονται τόσο με τα αποτελέσματα όσο και με τις στοιχηματικές αποδόσεις. Παράλληλα, η ανταγωνιστικότητα του πρωταθλήματος καθιστά το πρόβλημα πρόβλεψης Over/Under 2.5 ιδιαίτερα απαιτητικό και κατάλληλο για πειραματική μελέτη.

Κάθε γραμμή των αρχικών αρχείων αντιστοιχεί σε έναν ποδοσφαιρικό αγώνα. Από τα διαθέσιμα πεδία αξιοποιούνται κυρίως η ημερομηνία διεξαγωγής, η γηπεδούχος και η φιλοξενούμενη ομάδα, τα τελικά γκολ των δύο ομάδων, τα γκολ ημιχρόνου όπου είναι διαθέσιμα, καθώς και μεταβλητές που σχετίζονται με τις στοιχηματικές αγορές. Ιδιαίτερη σημασία έχουν οι αποδόσεις της αγοράς 1X2 και οι αποδόσεις της αγοράς Over/Under 2.5, επειδή αποτυπώνουν τη συλλογική προαγωνιστική εκτίμηση της αγοράς για τον αγώνα.

Παρότι θα μπορούσαν θεωρητικά να χρησιμοποιηθούν και άλλες μεταβλητές, όπως τραυματισμοί, καιρικές συνθήκες, ψυχολογικοί παράγοντες ή ειδησεογραφικά δεδομένα, τέτοιου είδους πληροφορία δεν είναι διαθέσιμη με ομοιόμορφο και αξιόπιστο τρόπο για όλες τις σεζόν. Για τον λόγο αυτό, η εργασία ακολουθεί μια πιο ελεγχόμενη και μεθοδολογικά ασφαλή προσέγγιση, εστιάζοντας σε χαρακτηριστικά που μπορούν να υπολογιστούν με συνέπεια από τα διαθέσιμα ιστορικά δεδομένα και τις προαγωνιστικές αποδόσεις.

Βασική αρχή της προεπεξεργασίας είναι ότι κάθε δείγμα εισόδου πρέπει να κατασκευάζεται αποκλειστικά από πληροφορία διαθέσιμη πριν από την έναρξη του αγώνα. Επομένως, τα δεδομένα ταξινομούνται χρονολογικά και κάθε δείγμα δημιουργείται χρησιμοποιώντας μόνο το ιστορικό των δύο ομάδων μέχρι το αμέσως προηγούμενο χρονικό σημείο. Με αυτόν τον τρόπο

αποφεύγεται η διαρροή μελλοντικής πληροφορίας, η οποία θα μπορούσε να οδηγήσει σε μη ρεαλιστικά υψηλή απόδοση του μοντέλου.

Κατά τη διάρκεια της εργασίας εξετάστηκαν περισσότερες από μία μορφές αναπαράστασης εισόδου. Μία κατεύθυνση βασίστηκε σε ακολουθίες προηγούμενων αγώνων, όπου κάθε token περιέγραφε έναν παλαιότερο αγώνα από την οπτική της αντίστοιχης ομάδας. Παράλληλα, εξετάστηκε και μια πιο συμπυκνωμένη προαγωνιστική αναπαράσταση, στην οποία η ιστορική συμπεριφορά των δύο ομάδων συνοψίζεται σε στατιστικά χαρακτηριστικά σταθερού μήκους. Η τελική αναπαράσταση των 47 χαρακτηριστικών συνδυάζει πληροφορία για τη γηπεδούχο ομάδα, πληροφορία για τη φιλοξενούμενη ομάδα και στοιχεία του τρέχοντος αγώνα.

Η επιλογή των χαρακτηριστικών βασίζεται στη λογική ότι η πιθανότητα εμφάνισης Over ή Under 2.5 γκολ επηρεάζεται από την επιθετική συμπεριφορά των ομάδων, την αμυντική τους σταθερότητα, τη φόρμα τους, τη συμπεριφορά τους εντός ή εκτός έδρας και την εκτίμηση της αγοράς πριν από τον αγώνα. Επομένως, τα χαρακτηριστικά οργανώνονται γύρω από τις ακόλουθες βασικές κατηγορίες:

1. μέσοι όροι γκολ υπέρ,
2. μέσοι όροι γκολ κατά,
3. ποσοστά αγώνων στους οποίους η ομάδα σκόραρε,
4. ποσοστά αγώνων στους οποίους η ομάδα δέχθηκε γκολ,
5. ποσοστά αγώνων που έληξαν Over 2.5,
6. πρόσφατη φόρμα σε μικρό χρονικό ορίζοντα,
7. πρόσφατη τάση προς Over ή Under,
8. συνεχόμενα σερί Over ή Under,
9. διάκριση εντός και εκτός έδρας συμπεριφοράς,
10. προαγωνιστική πληροφορία από στοιχηματικές αποδόσεις.

Τα παραπάνω χαρακτηριστικά υπολογίζονται ξεχωριστά για τη γηπεδούχο και για τη φιλοξενούμενη ομάδα. Αυτό είναι σημαντικό, επειδή η συμπεριφορά μιας ομάδας εντός έδρας

μπορεί να διαφέρει ουσιαστικά από τη συμπεριφορά της εκτός έδρας. Επομένως, η είσοδος του μοντέλου δεν περιγράφει γενικά την ποιότητα των δύο ομάδων, αλλά προσπαθεί να αποτυπώσει την κατάστασή τους μέσα στο συγκεκριμένο πλαίσιο του αγώνα.

Εκτός από τα στατιστικά χαρακτηριστικά των ομάδων, η αναπαράσταση περιλαμβάνει και πληροφορία του τρέχοντος αγώνα. Σε αυτήν περιλαμβάνονται κυρίως οι πιθανότητες της αγοράς 1X2, μετασχηματισμένες σε οπτική ομάδα, καθώς και οι implied probabilities της αγοράς Over/Under 2.5 μετά από κανονικοποίηση του bookmaker margin. Η χρήση αυτών των μεταβλητών είναι σημαντική, επειδή οι αποδόσεις των bookmakers ενσωματώνουν μεγάλο μέρος της διαθέσιμης προαγωνιστικής πληροφορίας.

Η τελική προαγωνιστική αναπαράσταση αποτελείται από 47 χαρακτηριστικά. Τα πρώτα 17 αφορούν τη γηπεδούχο ομάδα, τα επόμενα 17 αφορούν τη φιλοξενούμενη ομάδα και τα τελευταία 13 περιγράφουν το τρέχον fixture και την προαγωνιστική πληροφορία της αγοράς.

1. Μέσος όρος γκολ υπέρ της γηπεδούχου ομάδας στη σχετική συνθήκη έδρας.
2. Ποσοστό αγώνων στα οποία η γηπεδούχος ομάδα σκόραρε.
3. Μέσος όρος γκολ κατά της γηπεδούχου ομάδας στη σχετική συνθήκη έδρας.
4. Ποσοστό αγώνων στα οποία η γηπεδούχος ομάδα δέχθηκε γκολ.
5. Ποσοστό αγώνων της γηπεδούχου ομάδας που έληξαν Over 2.5 στη σχετική συνθήκη έδρας.
6. Μέσος όρος γκολ υπέρ της γηπεδούχου ομάδας στα τελευταία τρία αντίστοιχα παιχνίδια.
7. Ποσοστό πρόσφατων αγώνων στα οποία η γηπεδούχος ομάδα σκόραρε.
8. Μέσος όρος γκολ κατά της γηπεδούχου ομάδας στα τελευταία τρία αντίστοιχα παιχνίδια.
9. Ποσοστό πρόσφατων αγώνων στα οποία η γηπεδούχος ομάδα δέχθηκε γκολ.
10. Ποσοστό πρόσφατων αγώνων της γηπεδούχου ομάδας που έληξαν Over 2.5.
11. Συνολικός μέσος όρος γκολ υπέρ της γηπεδούχου ομάδας ανεξάρτητα από έδρα.
12. Συνολικός μέσος όρος γκολ κατά της γηπεδούχου ομάδας ανεξάρτητα από έδρα.

13. Συνολικό ποσοστό αγώνων της γηπεδούχου ομάδας που έληξαν Over 2.5.
14. Ποσοστό Over 2.5 στα τελευταία έξι παιχνίδια της γηπεδούχου ομάδας.
15. Τρέχον σερί Over της γηπεδούχου ομάδας.
16. Τρέχον σερί Under της γηπεδούχου ομάδας.
17. Κανονικοποιημένο μήκος διαθέσιμου ιστορικού της γηπεδούχου ομάδας.
18. Μέσος όρος γκολ υπέρ της φιλοξενούμενης ομάδας στη σχετική συνθήκη έδρας.
19. Ποσοστό αγώνων στα οποία η φιλοξενούμενη ομάδα σκόραρε.
20. Μέσος όρος γκολ κατά της φιλοξενούμενης ομάδας στη σχετική συνθήκη έδρας.
21. Ποσοστό αγώνων στα οποία η φιλοξενούμενη ομάδα δέχθηκε γκολ.
22. Ποσοστό αγώνων της φιλοξενούμενης ομάδας που έληξαν Over 2.5 στη σχετική συνθήκη έδρας.
23. Μέσος όρος γκολ υπέρ της φιλοξενούμενης ομάδας στα τελευταία τρία αντίστοιχα παιχνίδια.
24. Ποσοστό πρόσφατων αγώνων στα οποία η φιλοξενούμενη ομάδα σκόραρε.
25. Μέσος όρος γκολ κατά της φιλοξενούμενης ομάδας στα τελευταία τρία αντίστοιχα παιχνίδια.
26. Ποσοστό πρόσφατων αγώνων στα οποία η φιλοξενούμενη ομάδα δέχθηκε γκολ.
27. Ποσοστό πρόσφατων αγώνων της φιλοξενούμενης ομάδας που έληξαν Over 2.5.
28. Συνολικός μέσος όρος γκολ υπέρ της φιλοξενούμενης ομάδας ανεξάρτητα από έδρα.
29. Συνολικός μέσος όρος γκολ κατά της φιλοξενούμενης ομάδας ανεξάρτητα από έδρα.
30. Συνολικό ποσοστό αγώνων της φιλοξενούμενης ομάδας που έληξαν Over 2.5.
31. Ποσοστό Over 2.5 στα τελευταία έξι παιχνίδια της φιλοξενούμενης ομάδας.
32. Τρέχον σερί Over της φιλοξενούμενης ομάδας.

33. Τρέχον σερί Under της φιλοξενούμενης ομάδας.
34. Κανονικοποιημένο μήκος διαθέσιμου ιστορικού της φιλοξενούμενης ομάδας.
35. Ένδειξη οπτικής γηπεδούχου ομάδας για το τρέχον fixture.
36. Θέση γκολ υπέρ του τρέχοντος αγώνα, η οποία λαμβάνει μηδενική τιμή πριν από τη σέντρα.
37. Θέση γκολ κατά του τρέχοντος αγώνα, επίσης με μηδενική τιμή στο pre-match στάδιο.
38. Θέση γκολ υπέρ ημιχρόνου του τρέχοντος αγώνα, μηδενική πριν από τον αγώνα.
39. Θέση γκολ κατά ημιχρόνου του τρέχοντος αγώνα, μηδενική πριν από τον αγώνα.
40. Θέση συνολικών γκολ του τρέχοντος αγώνα, μηδενική στο pre-match στάδιο.
41. Θέση ένδειξης Over 2.5 του τρέχοντος αγώνα, μηδενική πριν από την πραγματοποίηση του αγώνα.
42. Θέση αποτελέσματος του τρέχοντος αγώνα, μηδενική επειδή η έκβαση δεν είναι γνωστή εκ των προτέρων.
43. Πιθανότητα νίκης της ομάδας από την αγορά 1X2 σε οπτική ομάδας.
44. Πιθανότητα ισοπαλίας από την αγορά 1X2.
45. Πιθανότητα ήττας της ομάδας από την αγορά 1X2 σε οπτική ομάδας.
46. Κανονικοποιημένη implied probability του Over 2.5.
47. Κανονικοποιημένη implied probability του Under 2.5.

Τα χαρακτηριστικά 36 έως 42 αντιστοιχούν σε θέσεις που σχετίζονται με πληροφορία του ίδιου του αγώνα, όπως γκολ, αποτέλεσμα ή ένδειξη Over/Under. Στο pre-match στάδιο οι τιμές αυτές δεν είναι διαθέσιμες και επομένως αρχικοποιούνται με μηδενικές τιμές. Η επιλογή αυτή προέκυψε αρχικά από τις ακολουθιακές εκδοχές του μοντέλου, όπου η χρήση πραγματικών τιμών για το τρέχον fixture θα οδηγούσε σε διαρροή πληροφορίας (data leakage). Για λόγους συνέπειας μεταξύ των διαφορετικών πειραματικών προσεγγίσεων, η ίδια δομή εισόδου διατηρήθηκε και στις πιο συμπυκνωμένες προαγωνιστικές αναπαραστάσεις.

Με την παραπάνω αναπαράσταση, το μοντέλο λαμβάνει ως είσοδο ένα πλούσιο προαγωνιστικό διάλυμα, το οποίο συνδυάζει μακροπρόθεσμη και βραχυπρόθεσμη φόρμα, επιθετική και αμυντική συμπεριφορά, διάκριση έδρας, ιστορική τάση Over/Under και πληροφορία της αγοράς. Έτσι, κάθε δείγμα περιγράφει τον αγώνα πριν από την έναρξή του, χωρίς να χρησιμοποιεί πληροφορία που προκύπτει από τον ίδιο τον υπό πρόβλεψη αγώνα.

3.2 Το πρωτάθλημα της Championship

Η επιλογή του πρωταθλήματος στο οποίο εφαρμόζεται ένα σύστημα πρόβλεψης αποτελεί σημαντικό μεθοδολογικό στοιχείο της έρευνας. Στην παρούσα διπλωματική εργασία η μελέτη επικεντρώνεται στην αγγλική Championship, δηλαδή στη δεύτερη κατηγορία του αγγλικού ποδοσφαίρου. Η επιλογή αυτή δεν βασίστηκε αποκλειστικά στη διαθεσιμότητα των δεδομένων, αλλά και στα ιδιαίτερα χαρακτηριστικά του πρωταθλήματος, τα οποία το καθιστούν κατάλληλο πεδίο μελέτης για το πρόβλημα πρόβλεψης Over/Under 2.5 γκολ.

Η Championship αποτελείται από 24 ομάδες, με κάθε ομάδα να αγωνίζεται 46 φορές ανά σεζόν. Συνολικά, κάθε αγωνιστική περίοδος περιλαμβάνει 552 αγώνες. Ο μεγάλος αυτός αριθμός αναμετρήσεων είναι ιδιαίτερα χρήσιμος για μοντέλα μηχανικής μάθησης, καθώς προσφέρει επαρκές πλήθος παρατηρήσεων για την εξαγωγή στατιστικών μοτίβων. Στο πλαίσιο της παρούσας εργασίας, όπου χρησιμοποιούνται ιστορικά δεδομένα ομάδων, rolling χαρακτηριστικά και προαγωνιστικές αναπαραστάσεις, το πλήθος των αγώνων ανά ομάδα αποτελεί σημαντικό πλεονέκτημα.

Ένας ακόμη λόγος επιλογής της Championship είναι το μεγάλο ιστορικό βάθος των διαθέσιμων δεδομένων. Η χρήση δεδομένων από πολλές διαδοχικές αγωνιστικές περιόδους επιτρέπει την εκπαίδευση και αξιολόγηση των μοντέλων σε ευρύ χρονολογικό πλαίσιο. Αυτό είναι ιδιαίτερα σημαντικό, επειδή η παρούσα εργασία δεν βασίζεται σε τυχαία ανάμειξη παλαιότερων και νεότερων αγώνων, αλλά σε χρονολογικά συνεπή αξιολόγηση. Με αυτόν τον τρόπο, η διαδικασία προσομοιάζει καλύτερα ένα πραγματικό σενάριο πρόβλεψης, όπου το μοντέλο εκπαιδεύεται σε παλαιότερα δεδομένα και αξιολογείται σε μεταγενέστερους αγώνες.

Η εστίαση σε ένα μόνο πρωτάθλημα μειώνει επίσης τη μεθοδολογική ετερογένεια. Εάν χρησιμοποιούνταν ταυτόχρονα δεδομένα από διαφορετικές χώρες ή διοργανώσεις, θα εισάγονταν πρόσθετοι παράγοντες διαφοροποίησης, όπως διαφορετικός ρυθμός παιχνιδιού, διαφορετικές σκοραριστικές τάσεις, διαφορετική αγωνιστική κουλτούρα και διαφορετικός βαθμός αποτελεσματικότητας των στοιχηματικών αγορών. Η χρήση μίας ενιαίας διοργάνωσης επιτρέπει πιο καθαρή αξιολόγηση του βασικού ερευνητικού ερωτήματος, δηλαδή κατά πόσο τα διαθέσιμα προαγωνιστικά δεδομένα μπορούν να υποστηρίξουν αξιόπιστη πρόβλεψη Over/Under 2.5.

Η Championship παρουσιάζει επίσης υψηλό πρακτικό ενδιαφέρον για την αγορά Over/Under 2.5. Πρόκειται για ένα ανταγωνιστικό πρωτάθλημα, στο οποίο η αγωνιστική φόρμα των ομάδων μπορεί να μεταβάλλεται σημαντικά μέσα στη σεζόν και αρκετές αναμετρήσεις παρουσιάζουν υψηλό βαθμό αβεβαιότητας. Το γεγονός αυτό καθιστά την πρόβλεψη του συνολικού αριθμού τερμάτων απαιτητική, καθώς η έκβαση Over ή Under δεν είναι πάντοτε προφανής από απλά στατιστικά ή από τη γενική δυναμικότητα των ομάδων.

Επιπλέον, η δομή του πρωταθλήματος επιτρέπει την κατασκευή πλούσιων χαρακτηριστικών εισόδου. Ο μεγάλος αριθμός αγώνων ανά ομάδα διευκολύνει τον υπολογισμό στατιστικών που σχετίζονται με την έδρα, την πρόσφατη φόρμα, την επιθετική και αμυντική συμπεριφορά, τις τάσεις Over/Under και τα σερί αποτελεσμάτων. Παράλληλα, η παρουσία στοιχηματικών αποδόσεων στα διαθέσιμα αρχεία επιτρέπει την ενσωμάτωση προαγωνιστικής πληροφορίας της αγοράς, στοιχείο ιδιαίτερα χρήσιμο για την αξιολόγηση τόσο της προβλεπτικής όσο και της στοιχηματικής απόδοσης των μοντέλων.

Πρέπει, ωστόσο, να σημειωθεί ότι δεν αξιοποιούνται όλοι οι αγώνες κάθε σεζόν ως έγκυρα δείγματα εισόδου. Επειδή η παρούσα εργασία βασίζεται σε ιστορικά χαρακτηριστικά που πρέπει να είναι διαθέσιμα πριν από τον αγώνα, απαιτείται να υπάρχει επαρκές προηγούμενο ιστορικό για κάθε ομάδα. Επομένως, οι πρώτες αγωνιστικές κάθε σεζόν δεν μπορούν πάντα να χρησιμοποιηθούν με τον ίδιο τρόπο, καθώς δεν έχει ακόμη συσσωρευθεί αρκετή πληροφορία για την κατασκευή rolling χαρακτηριστικών ή ιστορικών αναπαραστάσεων.

Συνολικά, η αγγλική Championship αποτελεί κατάλληλη και μεθοδολογικά υπερασπίσιμη επιλογή για τη μελέτη του προβλήματος Over/Under 2.5. Ο μεγάλος αριθμός αγώνων, το ιστορικό βάθος, η σχετική ομοιογένεια των δεδομένων, η ανταγωνιστικότητα του πρωταθλήματος και η δυνατότητα κατασκευής πλούσιων προαγωνιστικών χαρακτηριστικών την καθιστούν κατάλληλο πεδίο αξιολόγησης για τα μοντέλα της παρούσας εργασίας.

3.3 Λήψη και επεξεργασία δεδομένων

Η λήψη και επεξεργασία των δεδομένων αποτελεί ένα από τα πιο κρίσιμα στάδια της παρούσας εργασίας, καθώς η απόδοση των μοντέλων εξαρτάται σε μεγάλο βαθμό από την ποιότητα και τη μεθοδολογική ορθότητα των δειγμάτων εισόδου. Στο συγκεκριμένο πρόβλημα δεν αρκεί η απλή χρήση ιστορικών αποτελεσμάτων αγώνων. Αντίθετα, απαιτείται η μετατροπή των ακατέργαστων δεδομένων σε χρονικά συνεπείς και αποκλειστικά προαγωνιστικές αναπαραστάσεις, ώστε κάθε πρόβλεψη να βασίζεται μόνο σε πληροφορία διαθέσιμη πριν από την έναρξη του αγώνα.

Τα δεδομένα αντλήθηκαν από την ιστοσελίδα football-data.co.uk και περιλαμβάνουν αρχεία CSV της αγγλικής Championship για την περίοδο 2006–2025. Από κάθε αρχείο αξιοποιήθηκαν βασικά αγωνιστικά στοιχεία, όπως οι ομάδες, η ημερομηνία διεξαγωγής, το τελικό σκορ και το σκορ ημιχρόνου, καθώς και δεδομένα από τις στοιχηματικές αγορές 1X2 και Over/Under 2.5. Τα στοιχεία αυτά αποτέλεσαν τη βάση τόσο για την κατασκευή των ιστορικών χαρακτηριστικών των ομάδων όσο και για την ενσωμάτωση προαγωνιστικής πληροφορίας της αγοράς.

Πριν από τη δημιουργία των τελικών δειγμάτων πραγματοποιήθηκαν έλεγχοι ποιότητας στα αρχικά αρχεία. Οι έλεγχοι αυτοί αφορούσαν τη διαθεσιμότητα των βασικών πεδίων, τη μορφή των αποδόσεων, την ύπαρξη ελλিপών τιμών και τη συνέπεια των ονομάτων των ομάδων μεταξύ διαφορετικών σεζόν. Η διαδικασία αυτή είναι απαραίτητη, επειδή τα ιστορικά αρχεία δεν έχουν πάντοτε απόλυτα ομοιόμορφη δομή και ενδέχεται να παρουσιάζουν διαφοροποιήσεις στις διαθέσιμες στήλες ή στις ονομασίες των πεδίων.

Κεντρικό βήμα της προεπεξεργασίας είναι η χρονολογική ταξινόμηση των αγώνων. Δεδομένου ότι το πρόβλημα είναι προγνωστικό, κάθε δείγμα πρέπει να κατασκευάζεται με βάση το

διαθέσιμο ιστορικό μέχρι τη χρονική στιγμή πριν από τον υπό πρόβλεψη αγώνα. Για τον λόγο αυτό, οι αγώνες οργανώνονται με βάση την ημερομηνία τους και η παραγωγή των δειγμάτων ακολουθεί αυστηρά τη χρονική σειρά. Με αυτόν τον τρόπο αποφεύγεται η χρήση μελλοντικής πληροφορίας, δηλαδή το φαινόμενο data leakage, το οποίο θα μπορούσε να οδηγήσει σε μη ρεαλιστικά υψηλή απόδοση.

Για κάθε ομάδα διατηρείται ένα δυναμικό ιστορικό προηγούμενων αγώνων, το οποίο ενημερώνεται σταδιακά καθώς προχωρά η χρονολογική επεξεργασία των δεδομένων. Όταν κατασκευάζεται ένα νέο δείγμα, χρησιμοποιείται μόνο το ιστορικό που έχει ήδη δημιουργηθεί μέχρι εκείνη τη στιγμή. Το αποτέλεσμα του τρέχοντος αγώνα προστίθεται στο ιστορικό μόνο αφού ολοκληρωθεί η δημιουργία του αντίστοιχου pre-match δείγματος. Η σειρά αυτή είναι κρίσιμη, διότι διασφαλίζει ότι το μοντέλο δεν αποκτά πρόσβαση, άμεσα ή έμμεσα, σε πληροφορία που δεν θα ήταν διαθέσιμη σε πραγματικές συνθήκες πρόβλεψης.

Η πληροφορία που αποθηκεύεται στο ιστορικό κάθε ομάδας περιλαμβάνει στοιχεία όπως η έδρα, ο αντίπαλος, τα γκολ υπέρ και κατά, καθώς και η ένδειξη Over/Under 2.5. Από αυτό το ιστορικό παράγονται δύο βασικές μορφές αναπαράστασης. Η πρώτη είναι ακολουθιακή και διατηρεί πληροφορία από επιμέρους προηγούμενους αγώνες. Η δεύτερη είναι πιο συμπυκνωμένη και βασίζεται σε engineered χαρακτηριστικά, τα οποία συνοψίζουν τη συμπεριφορά της ομάδας σε διαφορετικά χρονικά παράθυρα και συνθήκες έδρας.

Παράλληλα, πραγματοποιείται ειδική επεξεργασία των στοιχηματικών αποδόσεων. Οι αποδόσεις δεν χρησιμοποιούνται μόνο στην αρχική αριθμητική τους μορφή, αλλά μετατρέπονται σε implied probabilities, ώστε να εκφράζουν πιθανότητες που μπορούν να αξιοποιηθούν πιο άμεσα από τα μοντέλα. Στην αγορά 1X2, η πληροφορία μετασχηματίζεται σε οπτική ομάδας, δηλαδή ως πιθανότητα νίκης, ισοπαλίας και ήττας της αντίστοιχης ομάδας. Με αυτόν τον τρόπο επιτυγχάνεται πιο συνεπής περιγραφή τόσο των ιστορικών όσο και των τρεχόντων δεδομένων.

Για κάθε αγώνα κατασκευάζεται επίσης μία προαγωνιστική περιγραφή του τρέχοντος fixture. Η περιγραφή αυτή περιλαμβάνει την πληροφορία έδρας και τις αντίστοιχες πιθανότητες της αγοράς. Αντίθετα, τα χαρακτηριστικά που σχετίζονται με το πραγματικό αποτέλεσμα του αγώνα,

όπως τα γκολ ή η τελική ένδειξη Over/Under, διατηρούνται σε μηδενικές τιμές στο pre-match στάδιο. Η επιλογή αυτή επιτρέπει τη διατήρηση σταθερής δομής εισόδου, χωρίς να εισάγεται πληροφορία από το μέλλον.

Από τον συνδυασμό των παραπάνω στοιχείων προκύπτουν οι τελικές μορφές εισόδου που χρησιμοποιούνται στα πειράματα της εργασίας. Σε ορισμένες εκδοχές, τα δείγματα οργανώνονται ως ακολουθίες ιστορικών αγώνων. Σε άλλες, η πληροφορία συμπυκνώνεται σε ενιαίες προαγωνιστικές αναπαραστάσεις, οι οποίες συνδυάζουν στατιστικά χαρακτηριστικά των δύο ομάδων και πληροφορία του τρέχοντος αγώνα. Και στις δύο περιπτώσεις, η βασική μεθοδολογική αρχή παραμένει ίδια: όλα τα χαρακτηριστικά πρέπει να είναι διαθέσιμα πριν από την έναρξη του αγώνα και να υπολογίζονται με χρονολογικά συνεπή τρόπο.

Συνολικά, η διαδικασία λήψης και επεξεργασίας των δεδομένων μετασχηματίζει τα ακατέργαστα αρχεία αγώνων σε δομημένες και αξιόπιστες προαγωνιστικές αναπαραστάσεις. Η διαδικασία αυτή αποτελεί θεμέλιο της πειραματικής μεθοδολογίας, καθώς εξασφαλίζει ότι τα μοντέλα εκπαιδεύονται και αξιολογούνται σε δεδομένα που προσομοιάζουν πραγματικές συνθήκες πρόβλεψης.

3.4 Κανονικοποίηση Τιμών

Η κανονικοποίηση των τιμών εισόδου αποτελεί κρίσιμο στάδιο στην εκπαίδευση νευρωνικών δικτύων, καθώς τα χαρακτηριστικά ενός προβλήματος συχνά παρουσιάζουν διαφορετικές κλίμακες, διαφορετική διασπορά και διαφορετική αριθμητική συμπεριφορά. Εάν οι διαφορές αυτές δεν αντιμετωπιστούν κατάλληλα, ορισμένες μεταβλητές μπορεί να επηρεάσουν δυσανάλογα τη διαδικασία εκπαίδευσης, όχι επειδή περιέχουν περισσότερη πληροφορία, αλλά επειδή έχουν μεγαλύτερο αριθμητικό εύρος.

Στην παρούσα εργασία, η ανάγκη για κανονικοποίηση είναι ιδιαίτερα σημαντική, επειδή η είσοδος των μοντέλων συνδυάζει διαφορετικές κατηγορίες χαρακτηριστικών. Συγκεκριμένα, χρησιμοποιούνται στατιστικά χαρακτηριστικά ομάδων, δείκτες πρόσφατης φόρμας, μεταβλητές που σχετίζονται με την έδρα, πληροφορία από στοιχηματικές αποδόσεις, καθώς και, σε

ορισμένες εκδοχές, ακολουθιακές αναπαραστάσεις προηγούμενων αγώνων. Η συνύπαρξη αυτών των διαφορετικών τύπων πληροφορίας απαιτεί μετασχηματισμό των τιμών σε πιο συμβατές αριθμητικές κλίμακες.

Σε απλούστερες εκδοχές της υλοποίησης χρησιμοποιείται min-max normalization, κατά το οποίο κάθε χαρακτηριστικό μετασχηματίζεται στο διάστημα $[0,1]$. Η προσέγγιση αυτή είναι απλή και διαισθητική, καθώς διατηρεί τη σχετική σειρά των τιμών και περιορίζει όλα τα χαρακτηριστικά σε κοινό εύρος. Είναι ιδιαίτερα πρακτική όταν τα δεδομένα οργανώνονται ως σταθερού μήκους διανύσματα engineered χαρακτηριστικών.

Ωστόσο, στις πιο ολοκληρωμένες εκδοχές της προεπεξεργασίας υιοθετείται train-only normalization. Συγκεκριμένα, οι παράμετροι κανονικοποίησης, όπως ο μέσος όρος και η τυπική απόκλιση, υπολογίζονται αποκλειστικά από το σύνολο εκπαίδευσης. Στη συνέχεια, οι ίδιες παράμετροι εφαρμόζονται στα δεδομένα επαλήθευσης ή ελέγχου. Η διαδικασία αυτή αντιστοιχεί σε standardization και έχει ιδιαίτερη μεθοδολογική σημασία, επειδή αποτρέπει την έμμεση χρήση πληροφορίας από μελλοντικά δεδομένα κατά το στάδιο της προεπεξεργασίας.

Η επιλογή train-only normalization είναι ιδιαίτερα σημαντική για τα Transformer-based μοντέλα. Στις ακολουθιακές εκδοχές, τα δεδομένα εισόδου οργανώνονται σε tensors που περιλαμβάνουν πολλαπλά χαρακτηριστικά ανά token. Αν οι τιμές αυτών των χαρακτηριστικών έχουν πολύ διαφορετικές κλίμακες, η εκπαίδευση μπορεί να γίνει λιγότερο σταθερή και πιο ευαίσθητη στις αρχικές τιμές των παραμέτρων. Με την κανονικοποίηση, διευκολύνεται η αριθμητική σταθερότητα και η πιο ομαλή σύγκλιση του μοντέλου.

Εκτός από την κανονικοποίηση των στατιστικών χαρακτηριστικών, εφαρμόζεται και ειδικός μετασχηματισμός των στοιχηματικών αποδόσεων. Οι αποδόσεις της αγοράς 1X2 και της αγοράς Over/Under 2.5 μετατρέπονται σε implied probabilities, ώστε να εκφράζουν την εκτίμηση της αγοράς σε μορφή πιθανότητας. Επιπλέον, όπου απαιτείται, οι πιθανότητες αυτές κανονικοποιούνται ώστε να μειώνεται η επίδραση του bookmaker margin. Με αυτόν τον τρόπο, οι αποδόσεις μετατρέπονται σε πιο καθαρά και συγκρίσιμα προαγωνιστικά χαρακτηριστικά.

Στην αγορά 1X2, η πληροφορία εκφράζεται σε οπτική ομάδα, δηλαδή ως πιθανότητα νίκης, ισοπαλίας και ήττας της αντίστοιχης ομάδας. Η επιλογή αυτή βελτιώνει τη συνοχή της αναπαράστασης, επειδή επιτρέπει την περιγραφή των δύο ομάδων με ενιαία λογική, ανεξάρτητα από το αν εξετάζεται η γηπεδούχος ή η φιλοξενούμενη πλευρά.

Συνολικά, η κανονικοποίηση στην παρούσα εργασία δεν αποτελεί απλώς τεχνική λεπτομέρεια, αλλά βασικό μέρος της μεθοδολογίας. Συμβάλλει στη σταθερότητα της εκπαίδευσης, στη δίκαιη συνύπαρξη χαρακτηριστικών διαφορετικής προέλευσης και στη διατήρηση της χρονολογικής ορθότητας της αξιολόγησης. Με τον τρόπο αυτό, τα μοντέλα εκπαιδεύονται σε δεδομένα που είναι αριθμητικά πιο ισορροπημένα και μεθοδολογικά πιο αξιόπιστα.

3.5 Σχεδιασμός και Υλοποίηση της Pipeline

Η παρούσα ενότητα περιγράφει τη συνολική δομή της υλοποίησης που χρησιμοποιήθηκε για την εκτέλεση των πειραμάτων της εργασίας. Σε αντίθεση με μια μονολιθική προσέγγιση, όπου όλα τα στάδια της διαδικασίας ενσωματώνονται σε ένα ενιαίο αρχείο, η υλοποίηση οργανώθηκε σε επιμέρους υποσυστήματα με διακριτούς ρόλους. Η λογική αυτή επιτρέπει σαφέστερο διαχωρισμό ανάμεσα στην προεπεξεργασία των δεδομένων, την κατασκευή των datasets, τον ορισμό του νευρωνικού μοντέλου, την εκπαίδευση, την αξιολόγηση και τους βοηθητικούς ελέγχους ποιότητας.

Η εργασία υλοποιήθηκε σε Python με χρήση του PyTorch framework, το οποίο παρέχει τα απαραίτητα εργαλεία για την κατασκευή και εκπαίδευση νευρωνικών δικτύων. Η επιλογή του PyTorch είναι κατάλληλη για την παρούσα εργασία, καθώς επιτρέπει ευέλικτο ορισμό μοντέλων, άμεση διαχείριση tensors, χρήση DataLoader για την τροφοδότηση των δεδομένων και αποτελεσματική υλοποίηση αρχιτεκτονικών τύπου Transformer Encoder.



Εικόνα 3.2: Λογότυπο του PyTorch framework.

Η δομή της υλοποίησης βασίζεται σε ανεξάρτητα αρχεία, καθένα από τα οποία αναλαμβάνει συγκεκριμένο μέρος της διαδικασίας. Με αυτόν τον τρόπο, η δημιουργία των προαγωνιστικών δειγμάτων, η μετατροπή τους σε μορφή κατάλληλη για PyTorch, ο ορισμός του Transformer μοντέλου και η διαδικασία εκπαίδευσης μπορούν να περιγραφούν και να ελεγχθούν ξεχωριστά. Η οργάνωση αυτή είναι σημαντική τόσο από τεχνική όσο και από μεθοδολογική άποψη, επειδή καθιστά τη διαδικασία πιο διαφανή, επεκτάσιμη και αναπαραγώγιμη.

Η συνολική ροή του συστήματος ξεκινά από την ανάγνωση των ακατέργαστων CSV αρχείων των αγωνιστικών περιόδων. Τα δεδομένα ταξινομούνται χρονολογικά και μετατρέπονται σε έγκυρα προαγωνιστικά δείγματα, τα οποία δεν περιλαμβάνουν πληροφορία από τον υπό πρόβλεψη αγώνα ή από μεταγενέστερα παιχνίδια. Στη συνέχεια, τα δείγματα οργανώνονται σε μορφή PyTorch Dataset, ώστε να μπορούν να τροφοδοτηθούν στο μοντέλο μέσω DataLoader.

Ακολούθως, το νευρωνικό μοντέλο εκπαιδεύεται πάνω στα δεδομένα με συγκεκριμένες επιλογές ως προς τον διαχωρισμό εκπαίδευσης και επαλήθευσης, την κανονικοποίηση, τον ρυθμό μάθησης, το early stopping και την παρακολούθηση των μετρικών απόδοσης. Μετά την εκπαίδευση, το μοντέλο αξιολογείται τόσο με κλασικές μετρικές δυαδικής ταξινόμησης, όπως accuracy, precision, recall, F1-score και ROC-AUC, όσο και με χρηματοοικονομικούς δείκτες, όπως profit, total return και ROI. Η διπλή αυτή αξιολόγηση είναι απαραίτητη, επειδή ο στόχος της εργασίας δεν είναι μόνο η στατιστική πρόβλεψη του Over/Under 2.5, αλλά και η διερεύνηση της πρακτικής αξίας των προβλέψεων σε στοιχηματικό περιβάλλον.

Συνολικά, η υλοποίηση της παρούσας εργασίας συγκροτεί μια ολοκληρωμένη πειραματική διαδικασία, η οποία καλύπτει όλα τα στάδια από τα αρχικά δεδομένα έως την τελική αξιολόγηση

των μοντέλων. Η modular οργάνωση του κώδικα διευκολύνει την κατανόηση, τον έλεγχο και τη σύγκριση διαφορετικών πειραματικών προσεγγίσεων, στοιχείο ιδιαίτερα σημαντικό για μια εργασία που εξετάζει πολλαπλές μορφές αναπαράστασης εισόδου και διαφορετικές αρχιτεκτονικές μοντέλων.

3.5.1 Το αρχείο `preprocessing_rawseq.py`

Το αρχείο `preprocessing_rawseq.py` αποτελεί τον βασικό πυρήνα της προεπεξεργασίας των δεδομένων. Ο ρόλος του είναι να διαβάζει τα αρχικά αρχεία CSV, να οργανώνει τους αγώνες χρονολογικά, να διατηρεί ιστορικό για κάθε ομάδα και τελικά να μετατρέπει κάθε αγώνα σε έγκυρο προαγωνιστικό δείγμα. Με τον τρόπο αυτό, κάθε είσοδος που δημιουργείται για το μοντέλο βασίζεται αποκλειστικά σε πληροφορία διαθέσιμη πριν από την έναρξη του αντίστοιχου αγώνα.

Στην αρχή του αρχείου ορίζονται βασικές παράμετροι της διαδικασίας, όπως ο φάκελος των αρχικών δεδομένων, το πλήθος των προηγούμενων αγώνων που θα χρησιμοποιηθούν, το πιθανό φίλτρο πρωταθλήματος και ο τρόπος οργάνωσης της ακολουθίας. Οι παράμετροι αυτές επιτρέπουν την εκτέλεση διαφορετικών πειραματικών εκδοχών χωρίς να απαιτείται αλλαγή στη βασική λογική της προεπεξεργασίας.

Ιδιαίτερη σημασία έχουν οι βοηθητικές συναρτήσεις που σχετίζονται με τις στοιχηματικές αποδόσεις. Οι συναρτήσεις `extract_ou_odds()` και `extract_ou_probs()` αναλαμβάνουν την εξαγωγή των αποδόσεων και των αντίστοιχων implied probabilities για την αγορά Over/Under 2.5. Επειδή τα διαθέσιμα αρχεία διαφορετικών σεζόν δεν έχουν πάντα ακριβώς την ίδια δομή, η αναζήτηση των στηλών γίνεται με σειρά προτεραιότητας. Έτσι, το σύστημα μπορεί να επιλέγει το πρώτο έγκυρο διαθέσιμο ζεύγος αποδόσεων, ακόμη και όταν οι ονομασίες των στηλών διαφοροποιούνται μεταξύ σεζόν.

Η συνάρτηση `_to_float()` χρησιμοποιείται για την ασφαλή μετατροπή τιμών σε αριθμητική μορφή. Η λειτουργία αυτή είναι σημαντική, επειδή τα αρχικά αρχεία μπορεί να περιλαμβάνουν ελλειπείς ή μη έγκυρες τιμές. Αντίστοιχα, οι συναρτήσεις `_imp_prob()` και `_norm3()` μετατρέπουν τις αποδόσεις σε implied probabilities και κανονικοποιούν τις πιθανότητες της αγοράς 1X2. Με

αυτόν τον τρόπο, οι αποδόσεις δεν χρησιμοποιούνται απλώς ως ακατέργαστοι αριθμοί, αλλά ως πιο καθαρά προαγωνιστικά σήματα σε μορφή πιθανοτήτων.

Η συνάρτηση `parse_match_dates()` εξασφαλίζει τη σωστή ερμηνεία των ημερομηνιών των αγώνων. Η χρονολογική ακρίβεια είναι απαραίτητη, επειδή ολόκληρη η διαδικασία βασίζεται στην αρχή ότι κάθε δείγμα πρέπει να δημιουργείται μόνο από πληροφορία που προηγείται χρονικά του υπό πρόβλεψη αγώνα. Εάν οι ημερομηνίες δεν αναλυθούν σωστά, υπάρχει κίνδυνος να αλλοιωθεί η σειρά των αγώνων και να δημιουργηθεί διαρροή πληροφορίας από το μέλλον.

Η συνάρτηση `build_current_match_token()` κατασκευάζει την αναπαράσταση του τρέχοντος αγώνα. Σε αυτήν περιλαμβάνεται η πληροφορία έδρας, οι προαγωνιστικές πιθανότητες από την αγορά 1X2, καθώς και οι implied probabilities της αγοράς Over/Under 2.5. Τα πεδία που σχετίζονται με το πραγματικό σκορ ή την τελική έκβαση του αγώνα διατηρούνται σε μηδενικές τιμές, επειδή δεν είναι διαθέσιμα πριν από τη διεξαγωγή του αγώνα. Η επιλογή αυτή επιτρέπει τη διατήρηση σταθερής δομής εισόδου χωρίς εισαγωγή μελλοντικής πληροφορίας.

Μία από τις σημαντικότερες συναρτήσεις του αρχείου είναι η `build_engineered_features()`. Η συνάρτηση αυτή δέχεται το ιστορικό μιας ομάδας και παράγει ένα σύνολο 17 στατιστικών χαρακτηριστικών. Τα χαρακτηριστικά αυτά περιλαμβάνουν μέσους όρους γκολ υπέρ και κατά, ποσοστά σκοραρίσματος και παθητικού, ποσοστά Over 2.5, δείκτες πρόσφατης φόρμας, σερί Over ή Under και κανονικοποιημένο μήκος διαθέσιμου ιστορικού. Με αυτόν τον τρόπο, το ακατέργαστο ιστορικό μιας ομάδας μετατρέπεται σε συμπυκνωμένη στατιστική περιγραφή της αγωνιστικής της κατάστασης.

Η συνάρτηση `build_match_dict()` δημιουργεί μια συνοπτική περιγραφή ενός προηγούμενου αγώνα από την οπτική της ομάδας. Η περιγραφή αυτή περιλαμβάνει στοιχεία όπως γκολ υπέρ, γκολ κατά, ένδειξη Over 2.5, ένδειξη σκοραρίσματος, ένδειξη παθητικού και πληροφορία έδρας. Η δομή αυτή λειτουργεί ως βασική μονάδα ιστορικού, πάνω στην οποία στηρίζεται ο υπολογισμός των μεταγενέστερων στατιστικών χαρακτηριστικών.

Για τις ακολουθιακές εκδοχές της εισόδου, η συνάρτηση `build_past_match_raw13()` δημιουργεί μια αναπαράσταση 13 χαρακτηριστικών για κάθε παρελθοντικό αγώνα. Σε αυτήν κωδικοποιούνται πληροφορίες όπως γκολ υπέρ και κατά, γκολ ημιχρόνου, συνολικά γκολ, ένδειξη Over/Under, αποτέλεσμα και διαθέσιμες προαγωνιστικές πιθανότητες. Η αναπαράσταση αυτή επιτρέπει στο μοντέλο να επεξεργάζεται επιμέρους προηγούμενους αγώνες ως tokens μέσα σε μια ακολουθία.

Η συνάρτηση `build_hybrid_token()` μπορεί να περιγραφεί ως η διαδικασία σύνθεσης της τελικής προαγωνιστικής αναπαράστασης. Συγκεκριμένα, συνενώνει τα στατιστικά χαρακτηριστικά της μίας ομάδας, τα αντίστοιχα χαρακτηριστικά του αντιπάλου και την αναπαράσταση του τρέχοντος αγώνα, δημιουργώντας ένα διάνυσμα 47 χαρακτηριστικών. Η αναπαράσταση αυτή χρησιμοποιείται στις πιο συμπυκνωμένες εκδοχές εισόδου, όπου η πληροφορία του αγώνα δίνεται στο μοντέλο ως ένα ενιαίο προαγωνιστικό διάνυσμα.

Η συνάρτηση `build_history_entry()` δημιουργεί μια πιο πλήρη δομή για κάθε ιστορικό token. Εκτός από τα αριθμητικά χαρακτηριστικά, περιλαμβάνει και μεταδεδομένα όπως η ταυτότητα της ομάδας, η ταυτότητα του αντιπάλου, η ένδειξη έδρας και η θέση του αγώνα στην ακολουθία. Η ύπαρξη αυτών των μεταδεδομένων είναι σημαντική, επειδή επιτρέπει στο μοντέλο να διατηρεί σωστό πλαίσιο για κάθε παρελθοντικό αγώνα.

Η `assemble_sequence()` είναι υπεύθυνη για τη σύνθεση της τελικής ακολουθίας εισόδου από το ιστορικό της γηπεδούχου ομάδας, το ιστορικό της φιλοξενούμενης ομάδας και την αναπαράσταση του τρέχοντος αγώνα. Υποστηρίζονται διαφορετικοί τρόποι οργάνωσης της ακολουθίας, όπως διαδοχική τοποθέτηση των ιστορικών, εναλλαγή μεταξύ των δύο ομάδων ή συγχώνευση σε κοινή χρονική σειρά. Η δυνατότητα αυτή επιτρέπει τη σύγκριση διαφορετικών τρόπων παρουσίασης της ιστορικής πληροφορίας στον Transformer.

Τέλος, η `preprocess_raw_sequences()` αποτελεί την κεντρική συνάρτηση που ενοποιεί όλα τα παραπάνω βήματα. Διαβάζει τα αρχεία των σεζόν, εφαρμόζει τα απαραίτητα φίλτρα, αναλύει τις ημερομηνίες, ταξινομεί τους αγώνες χρονολογικά και δημιουργεί τις τελικές μήτρες εισόδου, τις ετικέτες στόχου, τα αναγνωριστικά ομάδων και τις αποδόσεις που χρησιμοποιούνται στην

αξιολόγηση. Το πιο κρίσιμο σημείο της διαδικασίας είναι ότι κάθε δείγμα κατασκευάζεται πριν ενημερωθεί το ιστορικό των ομάδων με το αποτέλεσμα του ίδιου αγώνα. Με αυτόν τον τρόπο διασφαλίζεται ότι η προεπεξεργασία παραμένει χρονικά ορθή και δεν εισάγει διαρροή πληροφορίας στο μοντέλο.

3.5.2 Το αρχείο `dataset.py`

Το αρχείο `dataset.py` περιλαμβάνει την κλάση `FootballRawSeqDataset`, η οποία υλοποιεί τη λογική του `Dataset` του PyTorch. Ο ρόλος της κλάσης είναι να λειτουργεί ως ενδιάμεσο επίπεδο ανάμεσα στην προεπεξεργασία των δεδομένων και στη διαδικασία εκπαίδευσης του μοντέλου. Συγκεκριμένα, λαμβάνει τις αναπαραστάσεις που έχουν δημιουργηθεί στο στάδιο της προεπεξεργασίας και τις μετατρέπει σε μορφή κατάλληλη για χρήση από το PyTorch.

Στο εσωτερικό της κλάσης, τα αριθμητικά χαρακτηριστικά των δειγμάτων μετατρέπονται σε `tensors` τύπου `float32`, ώστε να μπορούν να χρησιμοποιηθούν ως είσοδος στο νευρωνικό μοντέλο. Αντίστοιχα, τα αναγνωριστικά των ομάδων, των αντιπάλων και η ένδειξη έδρας μετατρέπονται σε `tensors` τύπου `long`, καθώς χρησιμοποιούνται από `embedding layers`. Οι ετικέτες στόχου αποθηκεύονται επίσης σε μορφή `float32`, ώστε να είναι συμβατές με τη συνάρτηση απώλειας `BCEWithLogitsLoss`, η οποία χρησιμοποιείται για τη δυαδική ταξινόμηση Over/Under 2.5.

Ένα επιπλέον στοιχείο που διατηρείται στο `dataset` είναι οι στοιχηματικές αποδόσεις για Over και Under. Οι τιμές αυτές δεν χρησιμοποιούνται ως είσοδος στο `forward pass` του μοντέλου, αλλά αξιοποιούνται στο στάδιο της αξιολόγησης για την προσομοίωση στοιχηματικής στρατηγικής. Με αυτόν τον τρόπο, το ίδιο `dataset` υποστηρίζει τόσο την εκπαίδευση του μοντέλου όσο και τη μεταγενέστερη αξιολόγηση της χρηματοοικονομικής απόδοσης των προβλέψεων.

Οι μέθοδοι `__len__()` και `__getitem__()` παρέχουν τη βασική λειτουργικότητα που απαιτεί ο `DataLoader` του PyTorch. Η `__len__()` επιστρέφει το πλήθος των διαθέσιμων δειγμάτων, ενώ η `__getitem__()` επιστρέφει ένα πλήρες δείγμα εισόδου, το οποίο περιλαμβάνει τα χαρακτηριστικά, τα απαραίτητα μεταδεδομένα, την ετικέτα στόχου και τις αντίστοιχες

στοιχηματικές αποδόσεις. Με αυτόν τον τρόπο εξασφαλίζεται ότι κάθε batch που τροφοδοτείται στο μοντέλο έχει σταθερή και προβλέψιμη δομή.

Παρότι η κλάση `FootballRawSeqDataset` είναι σχετικά μικρή σε έκταση, έχει σημαντικό ρόλο στη συνολική υλοποίηση. Επιβάλλει μια ενιαία μορφή δείγματος για όλα τα στάδια της εκπαίδευσης και αξιολόγησης, διευκολύνει τη συνεργασία με τον `DataLoader` και διασφαλίζει ότι τα δεδομένα μεταφέρονται στο μοντέλο με τους κατάλληλους τύπους. Έτσι, το `dataset.py` λειτουργεί ως το σημείο σύνδεσης ανάμεσα στις προαγωνιστικές αναπαραστάσεις και στο νευρωνικό μοντέλο.

3.5.3 Το αρχείο `transformer.py`

Το αρχείο `transformer.py` περιλαμβάνει την υλοποίηση του νευρωνικού μοντέλου που χρησιμοποιείται ως βασική αρχιτεκτονική βαθιάς μάθησης στην παρούσα εργασία. Η υλοποίηση βασίζεται σε `Transformer Encoder` και είναι σχεδιασμένη ώστε να μπορεί να επεξεργάζεται τόσο ακολουθιακές αναπαραστάσεις προηγούμενων αγώνων όσο και πιο συμπυκνωμένες προαγωνιστικές αναπαραστάσεις του τρέχοντος αγώνα.

Το αρχείο περιλαμβάνει δύο βασικές κλάσεις: την `PositionalEncoding` και την `TransformerModel`. Η κλάση `PositionalEncoding` είναι υπεύθυνη για την προσθήκη πληροφορίας θέσης στα tokens της εισόδου. Η λειτουργία αυτή είναι απαραίτητη στις ακολουθιακές εκδοχές του μοντέλου, επειδή ο `Transformer` δεν διαθέτει από μόνος του ενσωματωμένη αντίληψη της χρονικής σειράς. Μέσω της προσθήκης `positional encoding`, το μοντέλο μπορεί να διακρίνει ποια στοιχεία της εισόδου αντιστοιχούν σε παλαιότερους και ποια σε πιο πρόσφατους αγώνες.

Η κλάση `TransformerModel` αποτελεί την κύρια υλοποίηση του μοντέλου. Στο αρχικό στάδιο, τα αριθμητικά χαρακτηριστικά εισόδου περνούν από ένα γραμμικό επίπεδο προβολής, ώστε να μετατραπούν από τον αρχικό χώρο χαρακτηριστικών σε χώρο `embeddings` κατάλληλης διάστασης. Παράλληλα, μπορούν να ενσωματωθούν επιπλέον `embeddings` που σχετίζονται με την ταυτότητα της ομάδας, την ταυτότητα του αντιπάλου και την ένδειξη έδρας. Με αυτόν τον τρόπο, το μοντέλο δεν αξιοποιεί μόνο τα αριθμητικά χαρακτηριστικά, αλλά και πληροφορία που περιγράφει το πλαίσιο κάθε δείγματος.

Ο βασικός κορμός του μοντέλου υλοποιείται με τις κλάσεις `nn.TransformerEncoderLayer` και `nn.TransformerEncoder` του PyTorch. Η παραμετροποίηση του Transformer περιλαμβάνει τη διάσταση αναπαράστασης (`d_model`), τον αριθμό κεφαλών προσοχής, τον αριθμό επιπέδων, το μέγεθος του feedforward μέρους, τη συνάρτηση ενεργοποίησης και το dropout. Η χρήση αυτών των παραμέτρων επιτρέπει τον έλεγχο της πολυπλοκότητας του μοντέλου, ώστε να μπορεί να προσαρμοστεί στις ανάγκες των πειραμάτων χωρίς να γίνεται υπερβολικά βαθύ ή δύσκολο στην εκπαίδευση.

Για τη συγκέντρωση της πληροφορίας της εισόδου σε μία τελική αναπαράσταση χρησιμοποιούνται διαφορετικές επιλογές pooling. Στην περίπτωση του cls pooling, προστίθεται ένα ειδικό learnable [CLS] token, το οποίο χρησιμοποιείται για τη συνολική αναπαράσταση του δείγματος. Εναλλακτικά, μπορεί να χρησιμοποιηθεί mean pooling, όπου η τελική αναπαράσταση προκύπτει από τον μέσο όρο των token representations. Η επιλογή pooling καθορίζει τον τρόπο με τον οποίο η πληροφορία που παράγει ο Transformer Encoder μετατρέπεται σε είσοδο για την τελική ταξινόμηση.

Η τελική κεφαλή ταξινόμησης αποτελείται από ένα μικρό πλήρως συνδεδεμένο δίκτυο, το οποίο λαμβάνει την τελική αναπαράσταση του Transformer και παράγει μία μοναδική αριθμητική έξοδο. Η έξοδος αυτή αντιστοιχεί στο logit της δυαδικής πρόβλεψης Over/Under 2.5. Κατά την εκπαίδευση χρησιμοποιείται κατάλληλη συνάρτηση απώλειας για δυαδική ταξινόμηση, ενώ κατά την αξιολόγηση το logit μετατρέπεται σε πιθανότητα μέσω sigmoid συνάρτησης.

Η μέθοδος `forward()` περιγράφει τη συνολική ροή υπολογισμού του μοντέλου. Αρχικά εφαρμόζεται η προβολή των χαρακτηριστικών στον χώρο embeddings και ενσωματώνονται, όπου χρησιμοποιούνται, τα embeddings ομάδας, αντιπάλου και έδρας. Στις ακολουθιακές αναπαραστάσεις προστίθεται positional encoding, ώστε να διατηρείται η πληροφορία σειράς.

Στη συνέχεια, η είσοδος περνά από τον Transformer Encoder και η τελική αναπαράσταση οδηγείται στην κεφαλή ταξινόμησης. Με αυτόν τον τρόπο, το μοντέλο συνδυάζει αριθμητικά

χαρακτηριστικά, πληροφορία ομάδας, πληροφορία έδρας και, όπου υπάρχει, χρονική δομή σε ένα ενιαίο υπολογιστικό πλαίσιο.

Συνολικά, το `transformer.py` αποτελεί το κεντρικό αρχείο ορισμού του μοντέλου. Η υλοποίησή του επιτρέπει την πειραματική αξιολόγηση διαφορετικών μορφών εισόδου, διατηρώντας σταθερό τον βασικό μηχανισμό μάθησης. Αυτό είναι σημαντικό για την παρούσα εργασία, επειδή επιτρέπει τη σύγκριση μεταξύ ακολουθιακών και συμπυκνωμένων προαγωνιστικών αναπαραστάσεων κάτω από κοινό Transformer-based πλαίσιο.

3.5.4 Το αρχείο `train.py`

Το αρχείο `train.py` αποτελεί το κεντρικό αρχείο εκτέλεσης των πειραμάτων. Σε αυτό συγκεντρώνονται τα βασικά στάδια της διαδικασίας, από τον ορισμό των παραμέτρων του πειράματος μέχρι την εκπαίδευση, την αξιολόγηση και την αποθήκευση των αποτελεσμάτων. Με τον τρόπο αυτό, το αρχείο λειτουργεί ως ο κύριος συνδετικός κρίκος ανάμεσα στην προεπεξεργασία των δεδομένων, στο μοντέλο Transformer και στη διαδικασία αξιολόγησης.

Η συνάρτηση `parse_args()` ορίζει τις βασικές παραμέτρους που μπορούν να μεταβληθούν κατά την εκτέλεση ενός πειράματος. Μέσα από αυτή τη διαδικασία καθορίζονται, μεταξύ άλλων, το μήκος ιστορικού, ο τρόπος οργάνωσης της εισόδου, το ποσοστό διαχωρισμού εκπαίδευσης και επαλήθευσης, το batch size, ο αριθμός εποχών, το patience για early stopping, το seed, καθώς και οι υπερπαραμέτροι του Transformer, όπως η διάσταση αναπαράστασης, ο αριθμός κεφαλών προσοχής, ο αριθμός επιπέδων και το dropout. Επιπλέον, ορίζονται παράμετροι που σχετίζονται με τον ρυθμό μάθησης, την κανονικοποίηση, τη στρατηγική threshold και την αξιολόγηση με βάση στοιχηματικούς δείκτες. Η δυνατότητα παραμετροποίησης είναι σημαντική, επειδή επιτρέπει την εκτέλεση πολλών πειραμάτων με ελεγχόμενο και αναπαραγώγιμο τρόπο.

Η συνάρτηση `set_seed()` χρησιμοποιείται για τον έλεγχο της στοχαστικότητας στη διαδικασία εκπαίδευσης. Συγκεκριμένα, θέτει σταθερές αρχικές τιμές για τις βιβλιοθήκες Python, NumPy και PyTorch. Η λειτουργία αυτή είναι απαραίτητη, επειδή τα νευρωνικά δίκτυα μπορεί να παρουσιάζουν διαφοροποιήσεις στην απόδοση λόγω τυχαίας αρχικοποίησης βαρών ή τυχαίας

σειράς δειγμάτων κατά την εκπαίδευση. Με τη χρήση σταθερού seed, τα πειράματα γίνονται πιο αναπαραγώγιμα και τα αποτελέσματα μπορούν να συγκριθούν με μεγαλύτερη αξιοπιστία.

Η συνάρτηση `normalize_sequences()` εφαρμόζει κανονικοποίηση στα δεδομένα εισόδου με βάση αποκλειστικά το σύνολο εκπαίδευσης. Συγκεκριμένα, υπολογίζει τον μέσο όρο και την τυπική απόκλιση από τα training samples και στη συνέχεια εφαρμόζει τις ίδιες παραμέτρους κανονικοποίησης και στα υπόλοιπα δεδομένα. Η επιλογή αυτή είναι ιδιαίτερα σημαντική, καθώς βελτιώνει τη σταθερότητα της εκπαίδευσης και ταυτόχρονα αποτρέπει την έμμεση χρήση πληροφορίας από το validation ή test σύνολο.

Η συνάρτηση `build_dataloaders()` συνδέει το στάδιο της προεπεξεργασίας με το στάδιο της εκπαίδευσης. Αρχικά καλεί τη διαδικασία δημιουργίας των προαγωνιστικών δειγμάτων, στη συνέχεια υπολογίζει το σημείο διαχωρισμού των δεδομένων σε training και validation σύνολο, εφαρμόζει την κανονικοποίηση και δημιουργεί τα αντίστοιχα Dataset και DataLoader αντικείμενα. Με τον τρόπο αυτό, τα δεδομένα μετατρέπονται σε μορφή κατάλληλη για τροφοδότηση στο μοντέλο κατά τη διάρκεια της εκπαίδευσης.

Η συνάρτηση `build_model()` είναι υπεύθυνη για την κατασκευή του μοντέλου Transformer με βάση τις παραμέτρους που έχουν οριστεί για το συγκεκριμένο πείραμα. Η ύπαρξη ξεχωριστής συνάρτησης για την αρχικοποίηση του μοντέλου βοηθά στη διατήρηση καθαρής δομής στον κώδικα, καθώς απομονώνει τη λογική δημιουργίας του μοντέλου από την υπόλοιπη διαδικασία εκπαίδευσης.

Η συνάρτηση `compute_metrics()` υπολογίζει τις βασικές μετρικές ταξινόμησης, όπως accuracy, precision, recall, F1-score και ROC-AUC. Οι μετρικές αυτές επιτρέπουν την αξιολόγηση της προβλεπτικής ικανότητας του μοντέλου όχι μόνο ως προς το ποσοστό σωστών προβλέψεων, αλλά και ως προς την ισορροπία μεταξύ των δύο κλάσεων Over και Under.

Η συνάρτηση `find_best_threshold()` σχετίζεται με την επιλογή του threshold που χρησιμοποιείται για τη μετατροπή των πιθανοτήτων του μοντέλου σε τελικές δυαδικές προβλέψεις. Ανάλογα με τη στρατηγική του πειράματος, μπορεί να χρησιμοποιηθεί σταθερό threshold, όπως 0.5, ή να

αναζητηθεί καλύτερο threshold με βάση συγκεκριμένη μετρική στο training σύνολο. Η διάκριση αυτή είναι σημαντική, επειδή διαχωρίζει την εκτίμηση πιθανότητας από την τελική απόφαση ταξινόμησης.

Ιδιαίτερη σημασία έχει η συνάρτηση `simulate_betting()`, η οποία ενσωματώνει το πρακτικό μέρος της αξιολόγησης. Η συνάρτηση αυτή μεταφράζει τις προβλέψεις του μοντέλου σε στοιχηματικές αποφάσεις και υπολογίζει δείκτες όπως πλήθος στοιχημάτων, συνολικό ποσό πονταρίσματος, συνολική επιστροφή, καθαρό κέρδος και ROI. Οι αποδόσεις δεν χρησιμοποιούνται για την εκπαίδευση του μοντέλου σε αυτό το στάδιο, αλλά αξιοποιούνται για να εκτιμηθεί κατά πόσο οι προβλέψεις θα μπορούσαν να έχουν πρακτική αξία σε ένα ρεαλιστικό στοιχηματικό πλαίσιο.

Η συνάρτηση `evaluate()` υλοποιεί τη συνολική διαδικασία αξιολόγησης ενός εκπαιδευμένου μοντέλου. Συλλέγει τις προβλέψεις, τις πραγματικές ετικέτες, τις πιθανότητες και τις διαθέσιμες αποδόσεις, υπολογίζει τις μετρικές ταξινόμησης και στη συνέχεια καλεί τη διαδικασία προσομοίωσης στοιχηματισμού. Με αυτόν τον τρόπο, η αξιολόγηση δεν περιορίζεται μόνο σε στατιστικά μέτρα απόδοσης, αλλά επεκτείνεται και σε χρηματοοικονομικούς δείκτες.

Η συνάρτηση `log_result()` αποθηκεύει τα αποτελέσματα κάθε πειράματος σε δομημένη μορφή, μαζί με τις παραμέτρους εκτέλεσης, τα βασικά μεταδεδομένα και τις καλύτερες μετρικές που επιτεύχθηκαν. Η αποθήκευση αυτή επιτρέπει την οργανωμένη παρακολούθηση διαφορετικών πειραμάτων και τη μεταγενέστερη σύγκριση μεταξύ διαφορετικών ρυθμίσεων.

Τέλος, η συνάρτηση `main()` συνδέει όλα τα παραπάνω στάδια σε μια ενιαία διαδικασία. Αρχικά αναλύει τις παραμέτρους εκτέλεσης, θέτει το seed, δημιουργεί τα δεδομένα και το μοντέλο, ορίζει τον optimizer, τη συνάρτηση απώλειας και τον scheduler, και στη συνέχεια εκτελεί το training loop. Σε κάθε εποχή παρακολουθούνται οι μετρικές απόδοσης, εφαρμόζεται αξιολόγηση στο validation σύνολο και χρησιμοποιείται early stopping ώστε να αποφευχθεί περιττή ή επιζήμια εκπαίδευση. Με αυτή τη δομή, το `train.py` αποτελεί το κεντρικό σημείο ελέγχου της πειραματικής διαδικασίας της παρούσας εργασίας.

3.5.5 Τα βοηθητικά αρχεία ελέγχου και επιθεώρησης

Εκτός από τα βασικά αρχεία προεπεξεργασίας, εκπαίδευσης και αξιολόγησης, η υλοποίηση περιλαμβάνει και ορισμένα βοηθητικά αρχεία, τα οποία χρησιμοποιούνται για τον έλεγχο και την επιθεώρηση των αρχικών δεδομένων. Τα αρχεία αυτά δεν συμμετέχουν άμεσα στην εκπαίδευση του μοντέλου, αλλά έχουν σημαντικό ρόλο στη διασφάλιση της ποιότητας και της συνέπειας των δεδομένων.

Το αρχείο `data_quality_check.py` χρησιμοποιείται για τον έλεγχο ποιότητας των αρχικών αρχείων κάθε σεζόν. Η βασική του λειτουργία είναι να εξετάζει αν υπάρχουν οι απαραίτητες στήλες, να εντοπίζει ελλείψεις τιμές και να ελέγχει ποιες στήλες στοιχηματικών αποδόσεων είναι διαθέσιμες για την αγορά Over/Under 2.5. Επιπλέον, υπολογίζει το ποσοστό των γραμμών που διαθέτουν έγκυρες αποδόσεις και εκτιμά το μέσο `bookmaker margin` όπου αυτό είναι δυνατό. Με αυτόν τον τρόπο, επιτρέπει την αξιολόγηση της καταλληλότητας των δεδομένων πριν από την έναρξη της εκπαίδευσης.

Το αρχείο `name_validation.py` χρησιμοποιείται για την επιθεώρηση των ονομάτων των ομάδων στα ιστορικά αρχεία. Συγκεκριμένα, εξάγει και εμφανίζει τα μοναδικά ονόματα ομάδων που εμφανίζονται στα δεδομένα. Η λειτουργία αυτή είναι χρήσιμη, επειδή σε δεδομένα πολλών αγωνιστικών περιόδων μπορεί να υπάρχουν μικρές διαφοροποιήσεις ή ασυνέπειες στην ονοματοδοσία των ομάδων. Ο έλεγχος αυτός βοηθά στη διατήρηση συνεπούς ιστορικού για κάθε ομάδα.

Το αρχείο `print_column.py` χρησιμοποιείται για την επιθεώρηση της δομής των CSV αρχείων. Εμφανίζει τον συνολικό αριθμό στηλών και εντοπίζει ιδιαίτερα τις στήλες που σχετίζονται με τις αγορές Over/Under 2.5 και 1X2. Η λειτουργία αυτή είναι σημαντική για την επιβεβαίωση της διαθεσιμότητας των απαραίτητων πεδίων, καθώς τα `bookmaker-related columns` μπορεί να διαφέρουν μεταξύ σεζόν.

Συνολικά, τα βοηθητικά αυτά αρχεία συμβάλλουν στη διαφάνεια και αξιοπιστία της διαδικασίας. Μέσω των ελέγχων ποιότητας, της επιθεώρησης ονομάτων και της επιβεβαίωσης των διαθέσιμων στηλών, μειώνεται ο κίνδυνος σφαλμάτων στα δεδομένα εισόδου και ενισχύεται η αξιοπιστία των πειραματικών αποτελεσμάτων.

3.5.6 Συνολική αποτίμηση της υλοποίησης

Η συνολική σχεδίαση της υλοποίησης δείχνει ότι η παρούσα εργασία δεν περιορίζεται στην απλή εκπαίδευση ενός μεμονωμένου νευρωνικού μοντέλου, αλλά βασίζεται σε μια ολοκληρωμένη πειραματική διαδικασία. Η προεπεξεργασία των δεδομένων, η κατασκευή των προαγωνιστικών αναπαραστάσεων, η μετατροπή τους σε μορφή κατάλληλη για PyTorch, ο ορισμός του Transformer μοντέλου, η εκπαίδευση, η αξιολόγηση και η καταγραφή των αποτελεσμάτων αποτελούν διακριτά αλλά αλληλένδετα στάδια της ίδιας μεθοδολογίας.

Η οργάνωση σε ανεξάρτητα αρχεία επιτρέπει σαφή διαχωρισμό αρμοδιοτήτων. Το `preprocessing_rawseq.py` αναλαμβάνει τη δημιουργία των προαγωνιστικών δειγμάτων, το `dataset.py` τα μετατρέπει σε μορφή κατάλληλη για χρήση από το PyTorch, το `transformer.py` ορίζει την αρχιτεκτονική του μοντέλου και το `train.py` υλοποιεί τη διαδικασία εκπαίδευσης και αξιολόγησης. Τα βοηθητικά αρχεία υποστηρίζουν τον έλεγχο ποιότητας των αρχικών δεδομένων, την επιθεώρηση των διαθέσιμων στηλών και τη συνέπεια των ονομάτων των ομάδων.

Η συγκεκριμένη δομή είναι σημαντική τόσο από τεχνική όσο και από μεθοδολογική άποψη. Από τεχνικής πλευράς, διευκολύνει την κατανόηση, τον έλεγχο και την επέκταση της υλοποίησης. Από μεθοδολογικής πλευράς, εξασφαλίζει ότι κάθε στάδιο της διαδικασίας μπορεί να περιγραφεί και να αξιολογηθεί ξεχωριστά, γεγονός που ενισχύει τη διαφάνεια και την αναπαραγωγιμότητα των πειραμάτων.

Ιδιαίτερη σημασία έχει το γεγονός ότι η υλοποίηση ενσωματώνει βασικές αρχές που είναι κρίσιμες για το πρόβλημα της ποδοσφαιρικής πρόβλεψης. Η χρονολογική επεξεργασία των αγώνων, η αποφυγή διαρροής μελλοντικής πληροφορίας, η κανονικοποίηση με βάση αποκλειστικά το training σύνολο και η αξιολόγηση με πολλαπλές μετρικές αποτελούν στοιχεία που καθιστούν την πειραματική διαδικασία πιο αξιόπιστη. Παράλληλα, η ενσωμάτωση

χρηματοοικονομικών δεικτών, όπως profit και ROI, επιτρέπει την αξιολόγηση των μοντέλων όχι μόνο σε στατιστικό αλλά και σε πρακτικό επίπεδο.

Συνολικά, ο κώδικας της παρούσας εργασίας λειτουργεί ως δομικό μέρος της επιστημονικής μεθοδολογίας και όχι απλώς ως τεχνικό εργαλείο υλοποίησης. Η σαφής οργάνωση των επιμέρους σταδίων επιτρέπει τη συστηματική διερεύνηση διαφορετικών αναπαραστάσεων εισόδου και διαφορετικών πειραματικών ρυθμίσεων, δημιουργώντας ένα σταθερό πλαίσιο πάνω στο οποίο βασίζονται τα αποτελέσματα και τα συμπεράσματα της διπλωματικής.

3.6 Πρόσθετες Πειραματικές Μεθοδολογικές Κατευθύνσεις

Κατά τη διάρκεια της παρούσας εργασίας εξετάστηκαν περισσότερες πειραματικές κατευθύνσεις από αυτές που τελικά παρουσιάζονται αναλυτικά στο κεφάλαιο των αποτελεσμάτων. Οι δοκιμές αυτές είχαν σημαντικό ρόλο στη διαμόρφωση της τελικής μεθοδολογίας, καθώς βοήθησαν να αξιολογηθεί όχι μόνο ποιο μοντέλο αποδίδει καλύτερα, αλλά και ποια μορφή αναπαράστασης της προαγωνιστικής πληροφορίας είναι πιο κατάλληλη για το πρόβλημα Over/Under 2.5.

Μία από τις πρώτες κατευθύνσεις ήταν η χρήση ακολουθιακής αναπαράστασης προηγούμενων αγώνων. Σε αυτή την προσέγγιση, ο Transformer δεν λάμβανε ως είσοδο ένα ενιαίο διάνυσμα χαρακτηριστικών, αλλά μια ακολουθία από tokens. Κάθε δείγμα μπορούσε να περιλαμβάνει (K) προηγούμενους αγώνες της γηπεδούχου ομάδας, (K) προηγούμενους αγώνες της φιλοξενούμενης ομάδας και ένα επιπλέον token για το τρέχον fixture. Επομένως, η είσοδος είχε μορφή ακολουθίας μήκους $(2K+1)$, όπου κάθε θέση αντιστοιχούσε είτε σε έναν παρελθοντικό αγώνα είτε στο προαγωνιστικό πλαίσιο του τρέχοντος αγώνα. Η προσέγγιση αυτή ήταν θεωρητικά ελκυστική, επειδή επέτρεπε στον Transformer να αξιοποιήσει άμεσα τη χρονική δομή των δεδομένων.

Στο πλαίσιο των ακολουθιακών πειραμάτων εξετάστηκαν και διαφορετικοί τρόποι οργάνωσης του ιστορικού. Στη στρατηγική concat, το ιστορικό της μίας ομάδας τοποθετείται πρώτα και ακολουθεί το ιστορικό της άλλης ομάδας. Στη στρατηγική interleave, τα ιστορικά των δύο ομάδων εναλλάσσονται, ώστε το μοντέλο να βλέπει πιο άμεσα τη σχέση μεταξύ των δύο πλευρών. Τέλος, στη στρατηγική merged, τα ιστορικά των δύο ομάδων συγχωνεύονται σε μία κοινή χρονολογική σειρά. Οι δοκιμές αυτές πραγματοποιήθηκαν ώστε να εξεταστεί αν η διάταξη των tokens μέσα στην ακολουθία επηρεάζει την ικανότητα του Transformer να αξιοποιεί τη χρονική πληροφορία.

Παράλληλα με τις ακολουθιακές αναπαραστάσεις, δοκιμάστηκαν και πιο συμπυκνωμένες προαγωνιστικές αναπαραστάσεις. Σε αυτές, η πληροφορία από το ιστορικό των ομάδων δεν δίνεται ως ξεχωριστή ακολουθία αγώνων, αλλά μετατρέπεται σε στατιστικά χαρακτηριστικά σταθερού μήκους. Η πιο πλήρης εκδοχή αυτής της κατεύθυνσης είναι η αναπαράσταση των 47 χαρακτηριστικών, όπου τα χαρακτηριστικά της γηπεδούχου ομάδας, τα χαρακτηριστικά της φιλοξενούμενης ομάδας και η πληροφορία του τρέχοντος αγώνα συνδυάζονται σε ένα ενιαίο pre-match token. Σε αυτή την περίπτωση, η παράμετρος (K) δεν αντιστοιχεί στο μήκος της

ακολουθίας που εισέρχεται στο μοντέλο, αλλά στο πλήθος προηγούμενων αγώνων που χρησιμοποιούνται για τον υπολογισμό rolling στατιστικών, όπως μέσοι όροι γκολ, πρόσφατη φόρμα και τάσεις Over/Under.

Σημαντική πειραματική κατεύθυνση αποτέλεσε επίσης η σύγκριση διαφορετικού πλήθους χαρακτηριστικών εισόδου. Η αρχική αναπαράσταση των 26 χαρακτηριστικών περιλάμβανε βασικά στατιστικά για τη γηπεδούχο και τη φιλοξενούμενη ομάδα, όπως μέσους όρους γκολ υπέρ και κατά, ποσοστά σκοραρίσματος, ποσοστά παθητικού και ποσοστά Over 2.5. Η αναπαράσταση των 34 χαρακτηριστικών επεκτείνει την προηγούμενη προσθέτοντας streak-based πληροφορία, δηλαδή χαρακτηριστικά που αποτυπώνουν συνεχόμενες εμφανίσεις Over ή Under. Τέλος, η αναπαράσταση των 47 χαρακτηριστικών προσθέτει πιο ολοκληρωμένη πληροφορία για το τρέχον fixture, όπως implied probabilities από τις αγορές 1X2 και Over/Under 2.5.

Μία ακόμη μεθοδολογική κατεύθυνση αφορούσε τη μετατροπή των στοιχηματικών αποδόσεων σε χαρακτηριστικά οπτικής ομάδας. Αντί οι bookmaker πιθανότητες να χρησιμοποιούνται απλώς ως ανεξάρτητες τιμές αγοράς, μετασχηματίστηκαν σε team-perspective μορφή, ώστε να εκφράζουν την πιθανότητα νίκης, ισοπαλίας ή ήττας από την οπτική της ομάδας που περιγράφεται. Με τον τρόπο αυτό, η πληροφορία της αγοράς έγινε πιο συνεπής με τα υπόλοιπα χαρακτηριστικά της εισόδου.

Εξετάστηκαν επίσης δυναμικές και venue-aware αναπαραστάσεις. Η venue-aware λογική βασίζεται στην ιδέα ότι η απόδοση μιας ομάδας μπορεί να διαφέρει σημαντικά εντός και εκτός έδρας. Για τον λόγο αυτό, δοκιμάστηκαν χαρακτηριστικά που διαχωρίζουν home και away συμπεριφορά, όπως μέσοι όροι γκολ, scoring rates, conceding rates, Over rates, δείκτες φόρμας και streaks ανά συνθήκη έδρας. Η κατεύθυνση αυτή είναι ιδιαίτερα σχετική με το ποδόσφαιρο, όπου η έδρα μπορεί να επηρεάζει τόσο την επιθετική όσο και την αμυντική συμπεριφορά μιας ομάδας.

Σε ορισμένες φάσεις του project εξετάστηκαν και πιο σύνθετες ιδέες, όπως δυναμικά ratings τύπου Elo, χαρακτηριστικά που προσεγγίζουν την επιθετική και αμυντική δυναμική των ομάδων, καθώς και βοηθητικοί στόχοι μάθησης. Για παράδειγμα, δοκιμάστηκαν κατευθύνσεις όπου το μοντέλο δεν στόχευε μόνο στην πρόβλεψη Over/Under 2.5, αλλά και σε βοηθητικές προβλέψεις, όπως συνολικό πλήθος γκολ, γκολ γηπεδούχου και φιλοξενούμενης ομάδας ή

κατηγορίες συνολικών γκολ. Παρότι αυτές οι προσεγγίσεις δεν αποτέλεσαν την τελική κύρια λύση, συνέβαλαν στην κατανόηση του πώς διαφορετικοί στόχοι μπορούν να επηρεάσουν τη μάθηση του μοντέλου.

Επιπλέον, ιδιαίτερη σημασία δόθηκε στη σύγκριση με απλούστερα baseline μοντέλα και bookmaker-based baselines. Η σύγκριση αυτή είναι απαραίτητη, επειδή ένα σύνθετο νευρωνικό μοντέλο έχει αξία μόνο αν μπορεί να ξεπεράσει απλούστερες προσεγγίσεις ή τουλάχιστον να αξιοποιήσει πιο αποτελεσματικά τη διαθέσιμη πληροφορία. Στο πλαίσιο αυτό εξετάστηκαν παραδοσιακές μέθοδοι μηχανικής μάθησης, όπως MLP και RBF, καθώς και η πληροφορία που περιέχεται ήδη στις αποδόσεις της αγοράς.

Πέρα από τη μορφή της εισόδου, δοκιμάστηκαν και διαφορετικές επιλογές αξιολόγησης και εκπαίδευσης. Χρησιμοποιήθηκε χρονολογικός διαχωρισμός των δεδομένων, ώστε το μοντέλο να εκπαιδεύεται σε παλαιότερες σεζόν και να αξιολογείται σε μεταγενέστερη περίοδο. Επιπλέον, εφαρμόστηκε train-only normalization, όπου οι παράμετροι κανονικοποίησης υπολογίζονται αποκλειστικά από το training set, ώστε να αποφευχθεί διαρροή πληροφορίας από τα validation ή test δεδομένα. Δοκιμάστηκαν επίσης seed-controlled experiments, early stopping, scheduler-based training, threshold tuning και calibration experiments, με στόχο τη σταθερότερη και πιο αξιόπιστη αξιολόγηση των μοντέλων.

Συνολικά, οι πρόσθετες αυτές πειραματικές κατευθύνσεις έπαιξαν σημαντικό ρόλο στη διαμόρφωση της τελικής εργασίας. Παρότι δεν οδήγησαν όλες στα καλύτερα αποτελέσματα, βοήθησαν να αναδειχθεί ένα βασικό συμπέρασμα: στο πρόβλημα Over/Under 2.5, η επιτυχία δεν εξαρτάται μόνο από την επιλογή της αρχιτεκτονικής, αλλά κυρίως από την ποιότητα, τη συνέπεια και τη χρονική ορθότητα της προαγωνιστικής αναπαράστασης. Η τελική επιλογή της πιο πλούσιας ενιαίας αναπαράστασης 47 χαρακτηριστικών προέκυψε ακριβώς μέσα από αυτή τη διαδικασία πειραματικής διερεύνησης.

Κεφάλαιο 4

Αποτελέσματα και Συζήτηση

4.1 Αποτελέσματα MLP Backpropagation

4.2 Αποτελέσματα Radial Basis Function (RBF)

4.3 Αποτελέσματα Transformer

4. Πειραματικά αποτελέσματα και αξιολόγηση

Στο παρόν κεφάλαιο παρουσιάζονται τα βασικά πειράματα που πραγματοποιήθηκαν στο πλαίσιο της εργασίας, καθώς και τα αποτελέσματα που προέκυψαν από αυτά. Στόχος δεν είναι μόνο η παράθεση αριθμητικών μετρήσεων, αλλά και η ερμηνεία τους, ώστε να εξαχθούν ουσιαστικά συμπεράσματα σχετικά με την απόδοση των διαφορετικών μοντέλων και μορφών εισόδου.

Η παρούσα εργασία επικεντρώνεται αποκλειστικά σε πρόβλημα δυαδικής ταξινόμησης. Το ζητούμενο είναι η πρόβλεψη του αν ένας αγώνας θα λήξει σε Over ή Under 2.5 συνολικά γκολ. Συγκεκριμένα, η έξοδος κωδικοποιείται ως 1 όταν στον αγώνα σημειώνονται περισσότερα από 2.5 γκολ και ως 0 όταν σημειώνονται λιγότερα ή ίσα με 2.5 γκολ. Με αυτόν τον τρόπο, η πρόβλεψη συνδέεται άμεσα με τη στοιχηματική αγορά Over/Under 2.5.

Για την αξιολόγηση των μοντέλων χρησιμοποιούνται κλασικές μετρικές δυαδικής ταξινόμησης, όπως accuracy, precision, recall, F1-score και ROC-AUC (Receiver Operating Characteristic – Area Under the Curve). Οι μετρικές αυτές παρέχουν μια συνολική εικόνα της προβλεπτικής ικανότητας των μοντέλων, τόσο ως προς το ποσοστό σωστών προβλέψεων όσο και ως προς την ικανότητα διαχωρισμού των δύο κλάσεων.

Ωστόσο, επειδή το πρόβλημα έχει άμεση εφαρμογή στον χώρο του στοιχηματισμού, η αξιολόγηση δεν περιορίζεται μόνο σε στατιστικές μετρικές. Σε όλα τα πειράματα πραγματοποιείται και προσομοίωση στοιχηματικής στρατηγικής με βάση τις προβλέψεις των μοντέλων. Για κάθε αγώνα θεωρείται σταθερό ποντάρισμα. Όταν η πρόβλεψη είναι σωστή, η επιστροφή υπολογίζεται με βάση την αντίστοιχη στοιχηματική απόδοση. Όταν η πρόβλεψη είναι λανθασμένη, καταγράφεται απώλεια ίση με το ποσό του πονταρίσματος.

Με βάση αυτή τη διαδικασία υπολογίζονται το συνολικό ποσό πονταρίσματος, οι συνολικές επιστροφές, το καθαρό κέρδος και ο δείκτης ROI (Return on Investment). Ο δείκτης ROI εκφράζει την αναλογία του καθαρού κέρδους προς το συνολικό ποσό που έχει πονταριστεί και αποτελεί έναν πιο πρακτικό δείκτη της χρησιμότητας των προβλέψεων. Με αυτόν τον τρόπο, τα μοντέλα αξιολογούνται όχι μόνο ως ταξινομητές, αλλά και ως συστήματα υποστήριξης αποφάσεων σε ένα ρεαλιστικό στοιχηματικό πλαίσιο.

$$\text{profit} = \text{total_return} - \text{total_spent}$$

$$\text{ROI}(\%) = \frac{\text{profit}}{\text{total_spent}} \times 100$$

Εικόνα 4.1: Υπολογισμός ROI

Στα πειράματα που ακολουθούν χρησιμοποιούνται τρεις βασικές μορφές εισόδου, οι οποίες διαφοροποιούνται ως προς τον αριθμό και τη δομή των χαρακτηριστικών. Η αναλυτική περιγραφή των χαρακτηριστικών παρουσιάστηκε στην Ενότητα 3.1, ενώ εδώ γίνεται σύντομη υπενθύμιση ώστε να είναι σαφές τι συγκρίνεται στα αποτελέσματα.

Η πρώτη μορφή εισόδου περιλαμβάνει 26 χαρακτηριστικά. Τα χαρακτηριστικά αυτά περιγράφουν βασικούς στατιστικούς δείκτες επιθετικής και αμυντικής συμπεριφοράς για τη γηπεδούχο και τη φιλοξενούμενη ομάδα. Περιλαμβάνουν μέσους όρους γκολ υπέρ και κατά, ποσοστά αγώνων στους οποίους η ομάδα σκόραρε ή δέχθηκε γκολ, ποσοστά Over 2.5, δείκτες πρόσφατης φόρμας και συνολικούς δείκτες απόδοσης.

Η δεύτερη μορφή εισόδου επεκτείνει την προηγούμενη σε 34 χαρακτηριστικά. Η βασική προσθήκη αφορά μεταβλητές που αποτυπώνουν τη δυναμική συμπεριφορά των ομάδων μέσω συνεχόμενων εμφανίσεων Over ή Under 2.5. Οι μεταβλητές αυτές επιτρέπουν στο μοντέλο να λαμβάνει υπόψη όχι μόνο τους μέσους όρους, αλλά και πρόσφατες τάσεις που μπορεί να επηρεάζουν την πιθανότητα εμφάνισης υψηλού ή χαμηλού σκορ.

Η τρίτη μορφή εισόδου αποτελεί την πιο πλήρη προαγωνιστική αναπαράσταση και περιλαμβάνει 47 χαρακτηριστικά. Σε αυτή την περίπτωση, τα στατιστικά χαρακτηριστικά της γηπεδούχου και της φιλοξενούμενης ομάδας συνδυάζονται με πληροφορία του τρέχοντος αγώνα και με πιθανότητες που προκύπτουν από τις στοιχηματικές αποδόσεις. Συγκεκριμένα, περιλαμβάνονται χαρακτηριστικά για τις δύο ομάδες, ένδειξη έδρας, μηδενικές θέσεις για πληροφορία που δεν

είναι διαθέσιμη πριν από τον αγώνα, implied probabilities από την αγορά 1X2 και implied probabilities από την αγορά Over/Under 2.5.

Η σύγκριση αυτών των τριών μορφών εισόδου είναι σημαντική, επειδή επιτρέπει την αξιολόγηση του κατά πόσο η αύξηση και η καλύτερη οργάνωση της προαγωνιστικής πληροφορίας βελτιώνει την απόδοση των μοντέλων. Ιδιαίτερη έμφαση δίνεται στην αναπαράσταση των 47 χαρακτηριστικών, καθώς αποτελεί την πιο ολοκληρωμένη μορφή εισόδου που εξετάστηκε στην παρούσα εργασία.

4.1 Αποτελέσματα MLP Backpropagation

4.1.1 MLP με 26 χαρακτηριστικά εισόδου

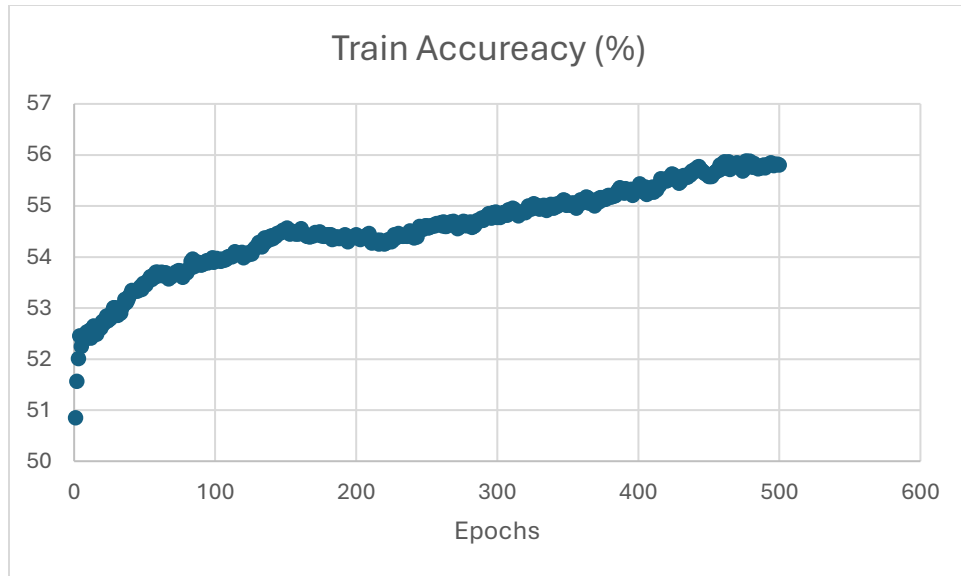
Στο πρώτο πείραμα χρησιμοποιήθηκε ένα νευρωνικό δίκτυο τύπου MLP με 26 νευρώνες στο επίπεδο εισόδου. Η επιλογή των χαρακτηριστικών βασίζεται στη λογική προηγούμενων εργασιών, με μικρές προσαρμογές ως προς ορισμένες μεταβλητές.

Η αρχιτεκτονική του δικτύου περιλαμβάνει δύο κρυφά επίπεδα, καθένα από τα οποία αποτελείται από 10 νευρώνες, καθώς και ένα νευρώνα στο επίπεδο εξόδου για τη δυαδική πρόβλεψη του προβλήματος. Ο ρυθμός μάθησης ορίστηκε σε 0.05, ενώ η εκπαίδευση πραγματοποιήθηκε για συνολικά 500 εποχές.

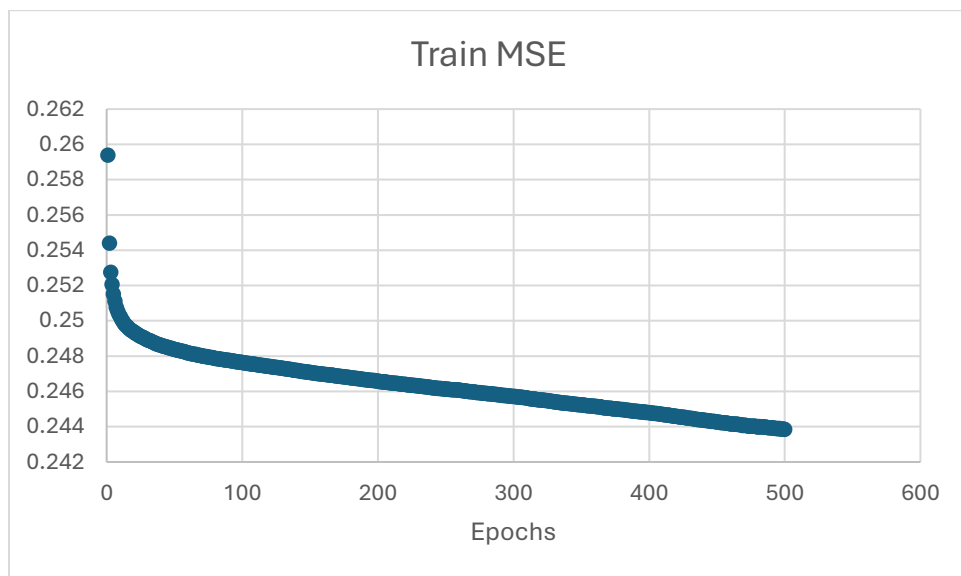
Τα δεδομένα εκπαίδευσης καλύπτουν τη χρονική περίοδο από τη σεζόν 2009/10 έως και τη σεζόν 2023/24, ενώ η αξιολόγηση του μοντέλου πραγματοποιήθηκε στη σεζόν 2024/25.

Συνοπτικά, η δομή του μοντέλου έχει ως εξής:

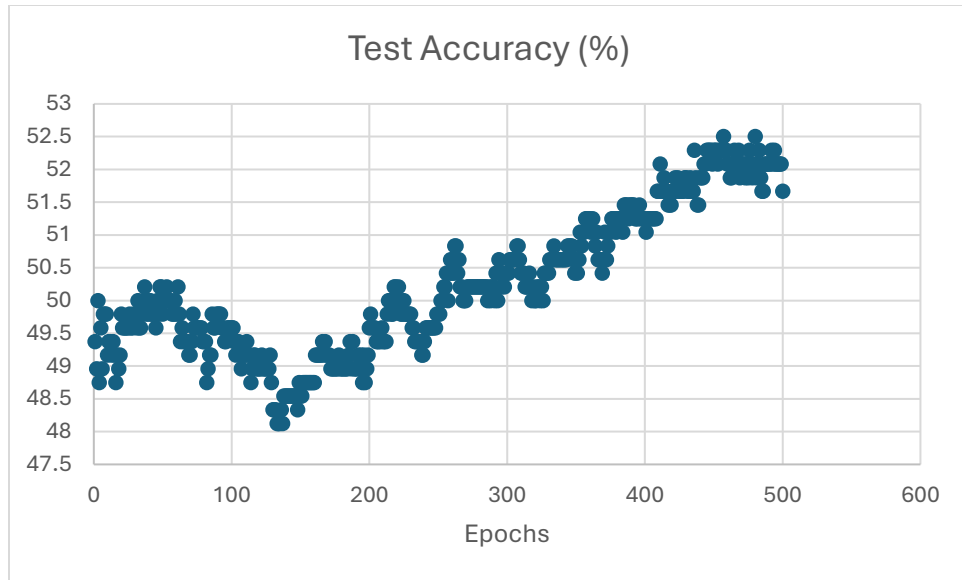
- Νευρώνες εισόδου: 26
- Κρυφά επίπεδα: 2
- Νευρώνες 1ου κρυφού επιπέδου: 10
- Νευρώνες 2ου κρυφού επιπέδου: 10
- Νευρώνες εξόδου: 1
- Ρυθμός μάθησης: 0.05
- Εποχές εκπαίδευσης: 500
- Δεδομένα εκπαίδευσης: 2009/10 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



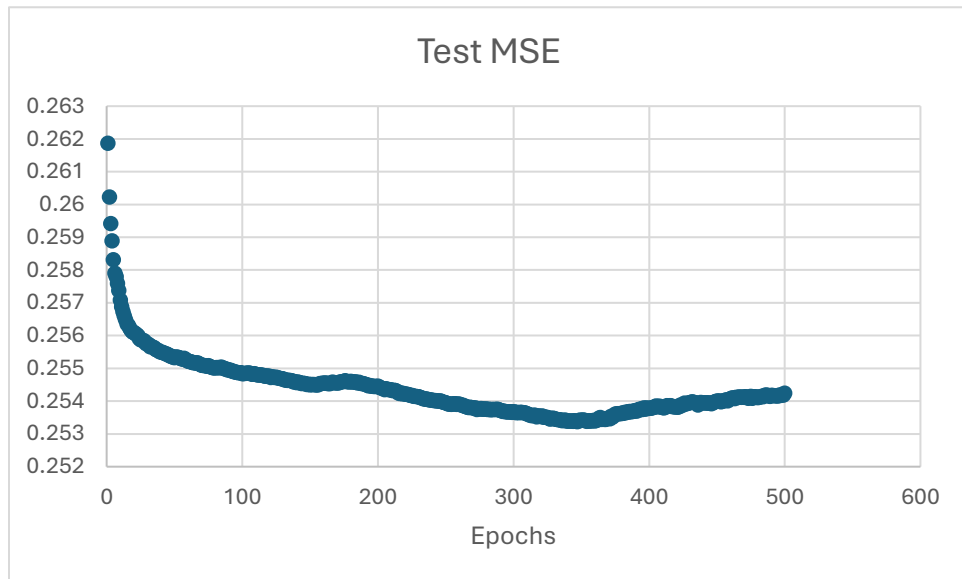
Εικόνα 4.1: Εξέλιξη του Train Accuracy για το MLP με 26 χαρακτηριστικά εισόδου.



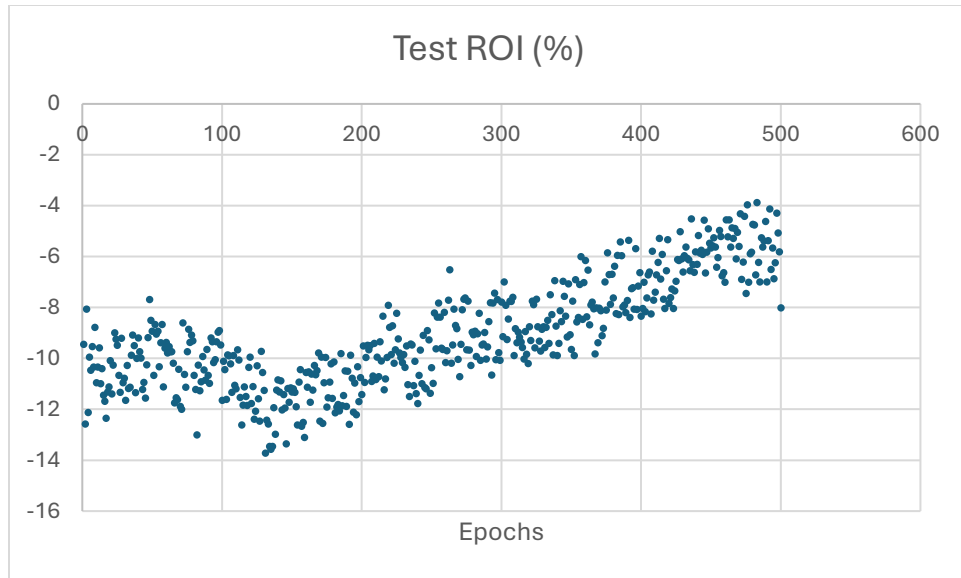
Εικόνα 4.2: Εξέλιξη του Train MSE για το MLP με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.3: Εξέλιξη του Test Accuracy για το MLP με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.4: Εξέλιξη του Test MSE για το MLP με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.5: Εξέλιξη του Test ROI για το MLP με 26 χαρακτηριστικά εισόδου.

Όπως φαίνεται στις Εικόνες 4.1–4.5, από τα γραφήματα του MLP με 26 χαρακτηριστικά εισόδου παρατηρείται σταδιακή βελτίωση της ακρίβειας στα δεδομένα ελέγχου κατά τη διάρκεια της εκπαίδευσης. Η καλύτερη τιμή test accuracy φτάνει το 52.5%, ενώ το Test MSE μειώνεται στις πρώτες εποχές και στη συνέχεια σταθεροποιείται με μικρές διακυμάνσεις. Παρόλα αυτά, το ROI παραμένει αρνητικό, με καλύτερη τιμή περίπου -4%, γεγονός που δείχνει ότι η μικρή βελτίωση στην προβλεπτική ακρίβεια δεν είναι αρκετή ώστε το μοντέλο να ξεπεράσει το περιθώριο των στοιχηματικών αποδόσεων. Επομένως, το συγκεκριμένο MLP λειτουργεί ως βασικό σημείο αναφοράς, αλλά δεν παρουσιάζει ακόμη ικανοποιητική πρακτική απόδοση σε στοιχηματικό επίπεδο.

4.1.2 MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1

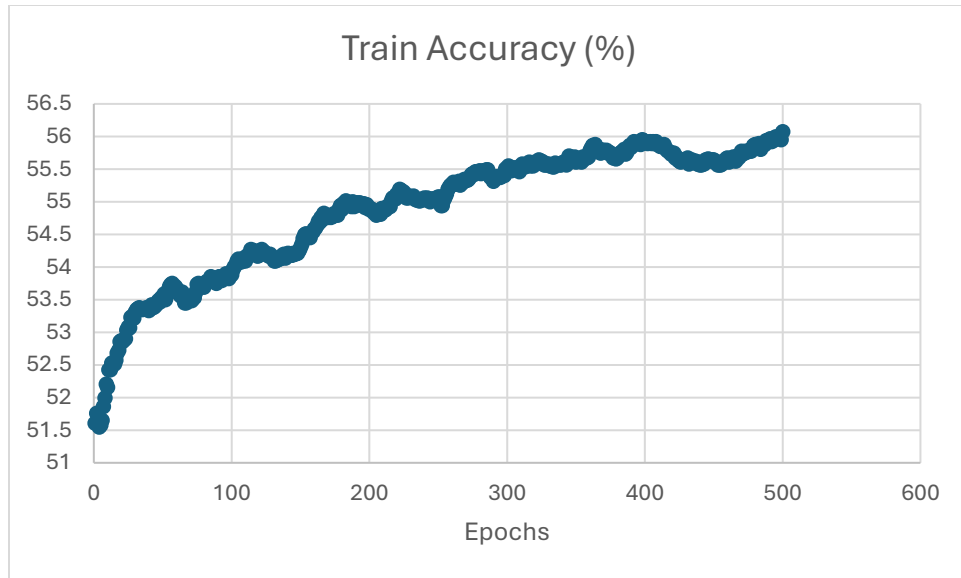
Στο δεύτερο πείραμα χρησιμοποιήθηκε ένα νευρωνικό δίκτυο τύπου MLP με 34 νευρώνες στο επίπεδο εισόδου, οι οποίοι αντιστοιχούν σε εμπλουτισμένη μορφή των αρχικών χαρακτηριστικών.

Η αρχιτεκτονική του δικτύου παραμένει ίδια με την προηγούμενη περίπτωση, περιλαμβάνοντας δύο κρυφά επίπεδα με 10 νευρώνες το καθένα και έναν νευρώνα εξόδου για τη δυαδική ταξινόμηση. Ο ρυθμός μάθησης ορίστηκε σε 0.05, ενώ η εκπαίδευση πραγματοποιήθηκε για 500 εποχές.

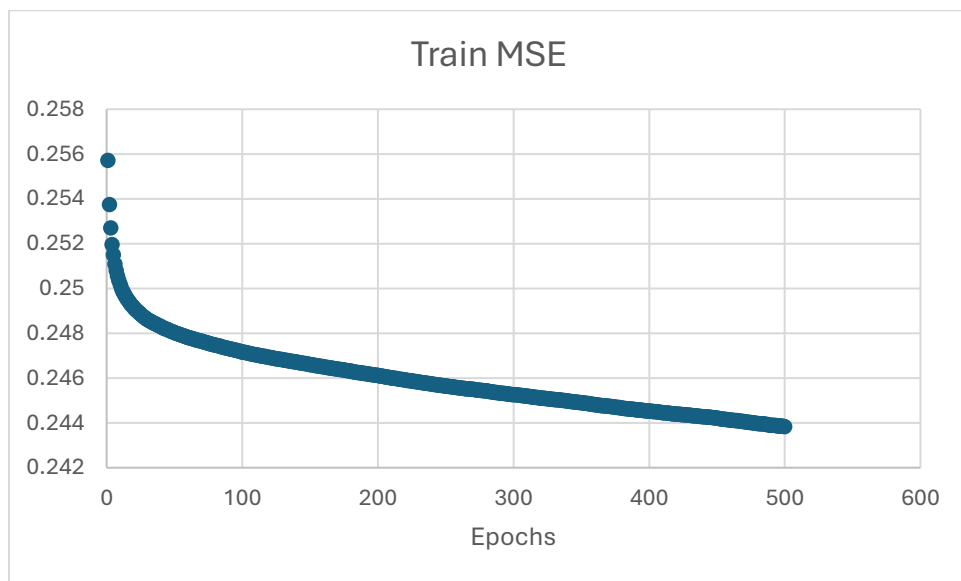
Τα δεδομένα εκπαίδευσης καλύπτουν τη χρονική περίοδο από τη σεζόν 2009/10 έως και τη σεζόν 2023/24, ενώ η αξιολόγηση πραγματοποιήθηκε στη σεζόν 2024/25.

Συνοπτικά, τα βασικά χαρακτηριστικά της αρχιτεκτονικής είναι τα εξής:

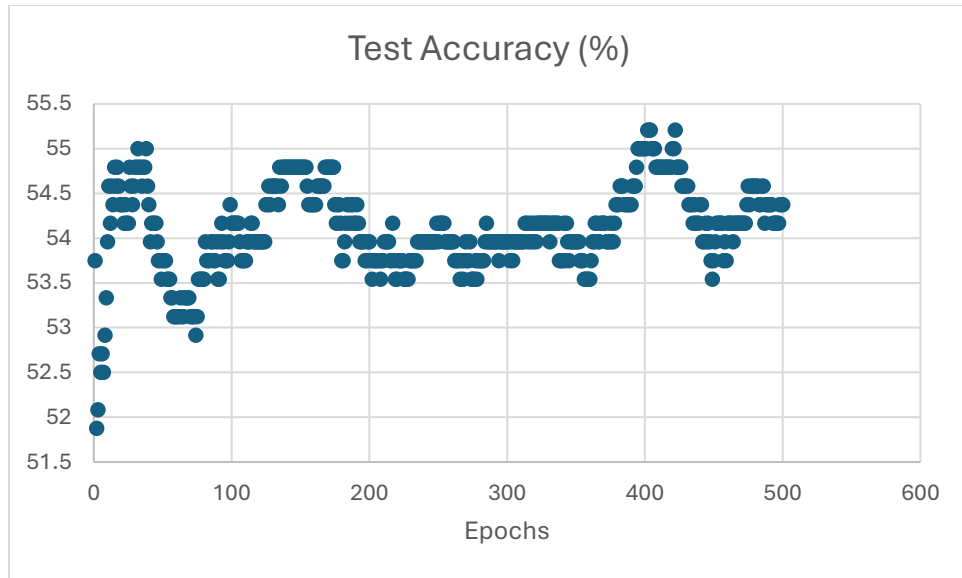
- Νευρώνες εισόδου: 34
- Κρυφά επίπεδα: 2
- Νευρώνες 1ου κρυφού επιπέδου: 10
- Νευρώνες 2ου κρυφού επιπέδου: 10
- Νευρώνας εξόδου: 1
- Ρυθμός μάθησης: 0.05
- Εποχές εκπαίδευσης: 500
- Δεδομένα εκπαίδευσης: 2009/10 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



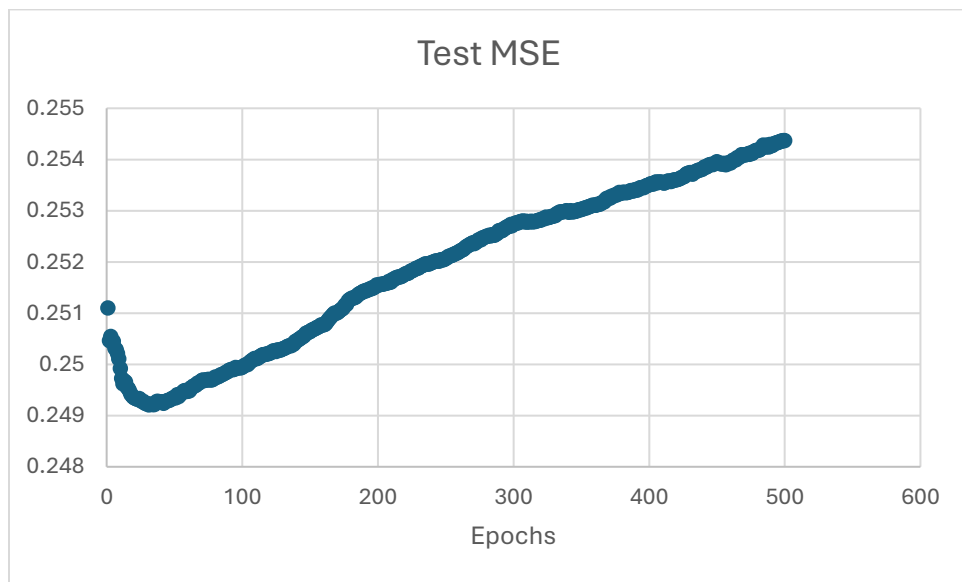
Εικόνα 4.6: Εξέλιξη του Train Accuracy για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



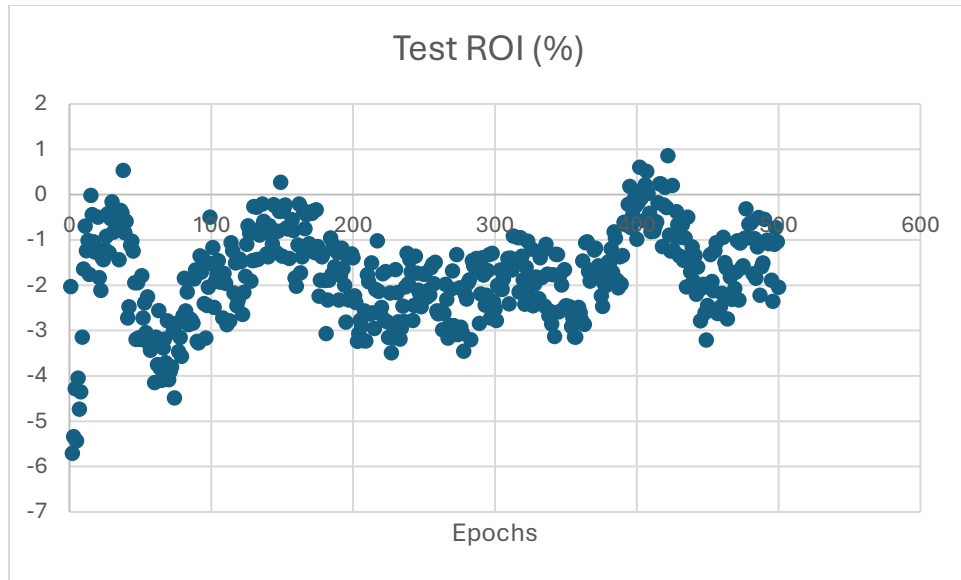
Εικόνα 4.7: Εξέλιξη του Train MSE για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



Εικόνα 4.8: Εξέλιξη του Test Accuracy για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



Εικόνα 4.9: Εξέλιξη του Test MSE για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



Εικόνα 4.10: Εξέλιξη του Test ROI για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.

Όπως φαίνεται στις Εικόνες 4.6–4.10, στα γραφήματα του MLP με 34 χαρακτηριστικά εισόδου παρατηρείται ότι η ακρίβεια στα δεδομένα ελέγχου είναι υψηλότερη σε σχέση με το προηγούμενο πείραμα, φτάνοντας περίπου στο 55.22%. Ωστόσο, το Test MSE μειώνεται μόνο στις πρώτες εποχές και στη συνέχεια παρουσιάζει σταδιακή αύξηση. Η συμπεριφορά αυτή δείχνει ότι, παρότι το μοντέλο πετυχαίνει σχετικά καλύτερη ταξινόμηση, οι παραγόμενες πιθανότητες γίνονται λιγότερο καλά βαθμονομημένες όσο συνεχίζεται η εκπαίδευση. Με άλλα λόγια, το μοντέλο μπορεί να προβλέπει σωστά περισσότερους αγώνες, αλλά ταυτόχρονα να αποδίδει μεγαλύτερη βεβαιότητα σε ορισμένες λανθασμένες προβλέψεις, κάτι που αυξάνει το MSE.

Το ROI παραμένει κυρίως αρνητικό, αν και σε ορισμένες εποχές πλησιάζει ή ξεπερνά οριακά το μηδέν(μέγιστο 0.42%). Αυτό δείχνει ότι η αύξηση των χαρακτηριστικών από 26 σε 34 βελτίωσε την προβλεπτική απόδοση, αλλά η βελτίωση αυτή δεν είναι ακόμη αρκετά σταθερή ώστε να οδηγήσει σε ξεκάθαρα κερδοφόρα στοιχηματική στρατηγική.

4.1.3 MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2

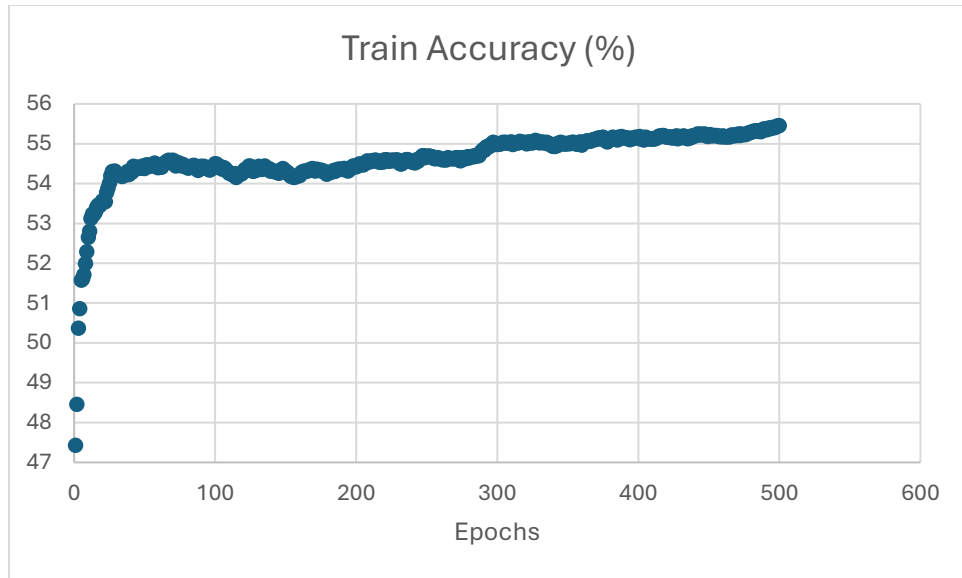
Στο συγκεκριμένο πείραμα εξετάστηκε μια εναλλακτική αρχιτεκτονική MLP με 34 χαρακτηριστικά εισόδου, με στόχο να διερευνηθεί αν η μείωση της πολυπλοκότητας στο δεύτερο κρυφό επίπεδο μπορεί να βελτιώσει τη γενίκευση του μοντέλου

Η αρχιτεκτονική διαφοροποιείται ελαφρώς σε σχέση με τα προηγούμενα πειράματα, καθώς το πρώτο κρυφό επίπεδο περιλαμβάνει 10 νευρώνες, ενώ το δεύτερο μειώνεται σε 5 νευρώνες. Η αλλαγή αυτή αποσκοπεί στον καλύτερο έλεγχο της πολυπλοκότητας του μοντέλου και φαίνεται να συμβάλλει στη βελτίωση της απόδοσης. Ο ρυθμός μάθησης ορίστηκε σε 0.05, ενώ η εκπαίδευση πραγματοποιήθηκε για 500 εποχές.

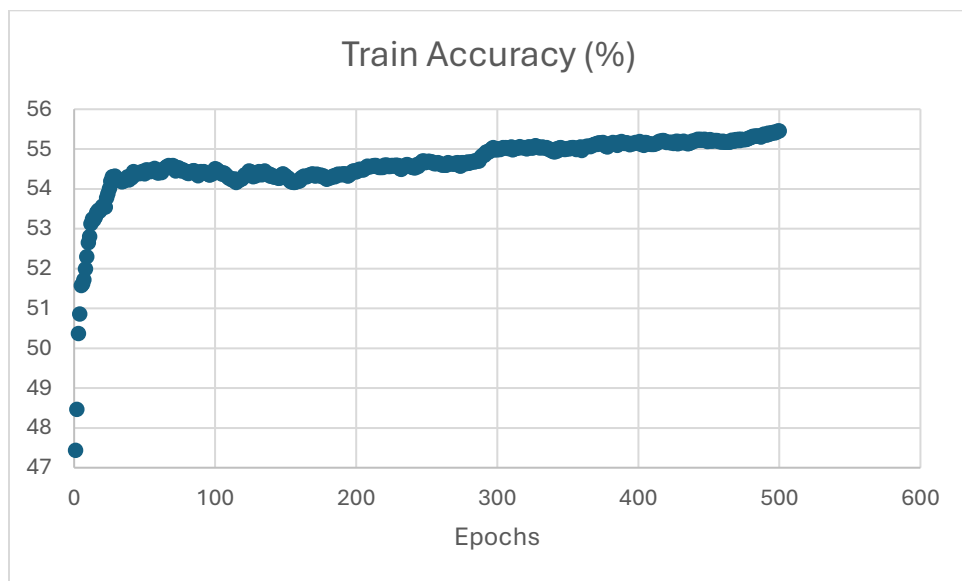
Τα δεδομένα εκπαίδευσης καλύπτουν τη χρονική περίοδο από τη σεζόν 2009/10 έως και τη σεζόν 2023/24, ενώ η αξιολόγηση πραγματοποιήθηκε στη σεζόν 2024/25.

Συνοπτικά, η αρχιτεκτονική και οι παράμετροι του μοντέλου είναι οι εξής:

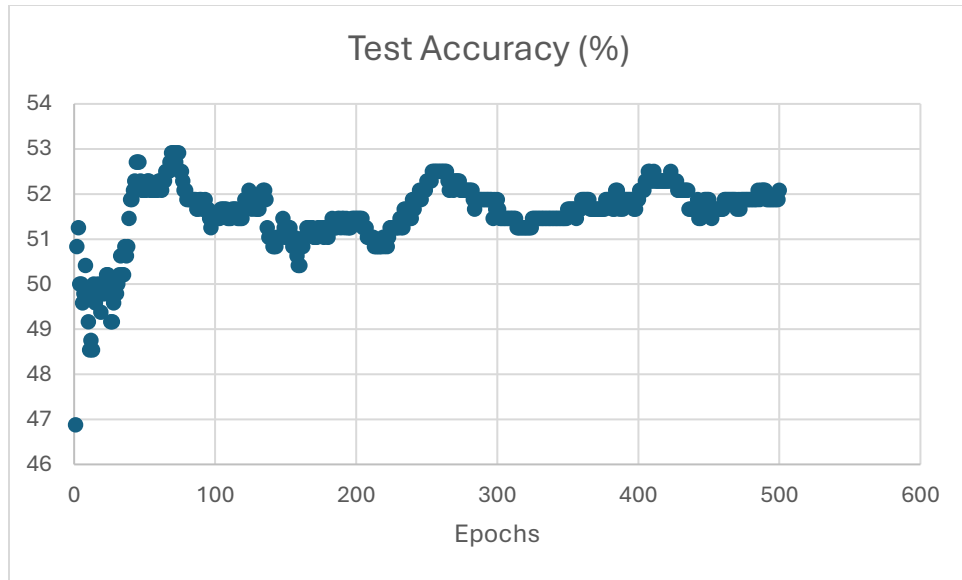
- Νευρώνες εισόδου: 34
- Κρυφά επίπεδα: 2
- Νευρώνες 1ου κρυφού επιπέδου: 10
- Νευρώνες 2ου κρυφού επιπέδου: 5
- Νευρώνες εξόδου: 1
- Ρυθμός μάθησης: 0.05
- Εποχές εκπαίδευσης: 500
- Δεδομένα εκπαίδευσης: 2009/10 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



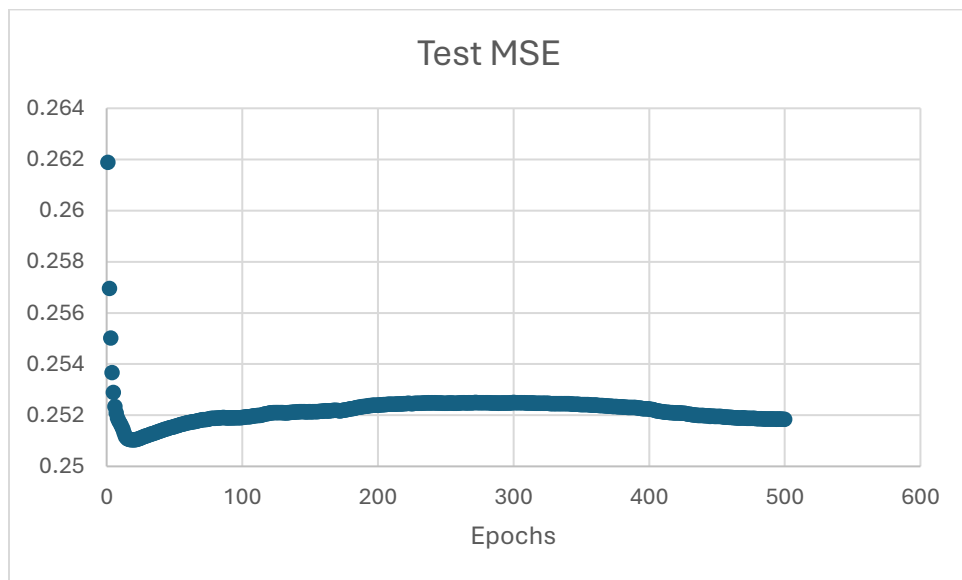
Εικόνα 4.11: Εξέλιξη του Train Accuracy για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



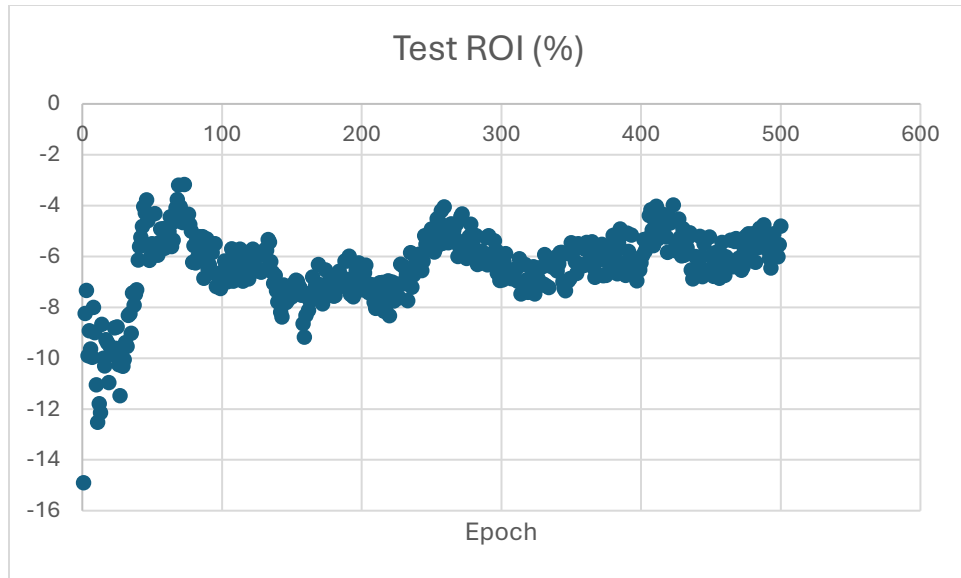
Εικόνα 4.12: Εξέλιξη του Train MSE για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



Εικόνα 4.13: Εξέλιξη του Test Accuracy για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



Εικόνα 4.14: Εξέλιξη του Test MSE για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



Εικόνα 4.15: Εξέλιξη του Test ROI για το MLP με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.

Όπως φαίνεται στις Εικόνες 4.11–4.15, από τα αποτελέσματα της δεύτερης προσπάθειας με 34 χαρακτηριστικά παρατηρείται ότι η μείωση των νευρώνων στο δεύτερο κρυφό επίπεδο δεν οδήγησε σε ουσιαστική βελτίωση της ακρίβειας. Η καλύτερη τιμή Test Accuracy έφτασε το 53.08%, ενώ το ROI παρέμεινε αρνητικό, με καλύτερη τιμή -2.68%. Παρότι το μοντέλο παρουσιάζει σχετικά σταθερή συμπεριφορά μετά τις πρώτες εποχές, η απόδοσή του δεν είναι αρκετή ώστε να οδηγήσει σε κερδοφόρα στοιχηματική στρατηγική. Συνεπώς, η απλή μείωση της πολυπλοκότητας του MLP δεν φαίνεται να βελτιώνει σημαντικά τη γενίκευση στο συγκεκριμένο πρόβλημα.

4.1.4 MLP με 47 χαρακτηριστικά εισόδου

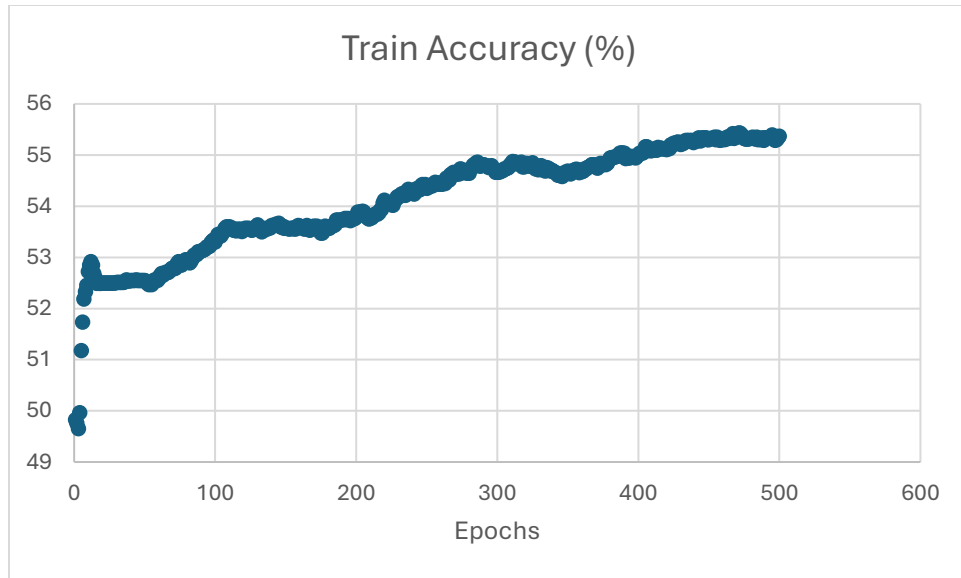
Στο επόμενο πείραμα χρησιμοποιήθηκε η ίδια αρχιτεκτονική MLP, με τη βασική διαφοροποίηση να αφορά το πλήθος και τη δομή των χαρακτηριστικών εισόδου. Συγκεκριμένα, το μοντέλο δέχεται 47 νευρώνες στο επίπεδο εισόδου.

Η αρχιτεκτονική του δικτύου παραμένει ίδια με το προηγούμενο πείραμα, περιλαμβάνοντας δύο κρυφά επίπεδα με 10 και 10 νευρώνες αντίστοιχα, καθώς και έναν νευρώνα εξόδου για τη δυαδική πρόβλεψη του προβλήματος. Ο ρυθμός μάθησης διατηρείται στο 0.05, ενώ η εκπαίδευση πραγματοποιείται για 500 εποχές.

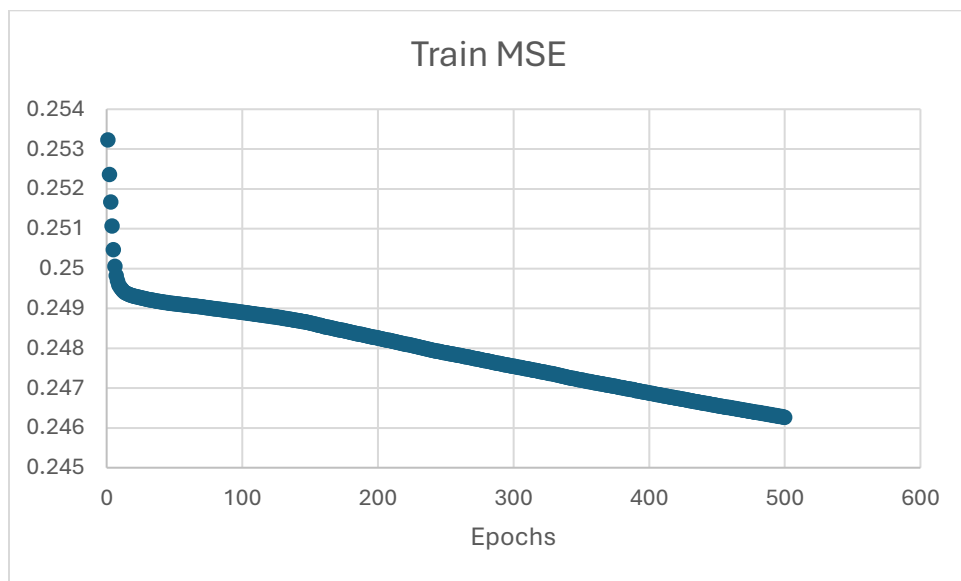
Τα δεδομένα εκπαίδευσης καλύπτουν τη χρονική περίοδο από τη σεζόν 2009/10 έως και τη σεζόν 2023/24, ενώ η αξιολόγηση πραγματοποιείται στη σεζόν 2024/25, διασφαλίζοντας χρονική συνέπεια και αποφυγή χρήσης μελλοντικής πληροφορίας.

Συνοπτικά, η αρχιτεκτονική και οι παράμετροι του μοντέλου είναι οι εξής:

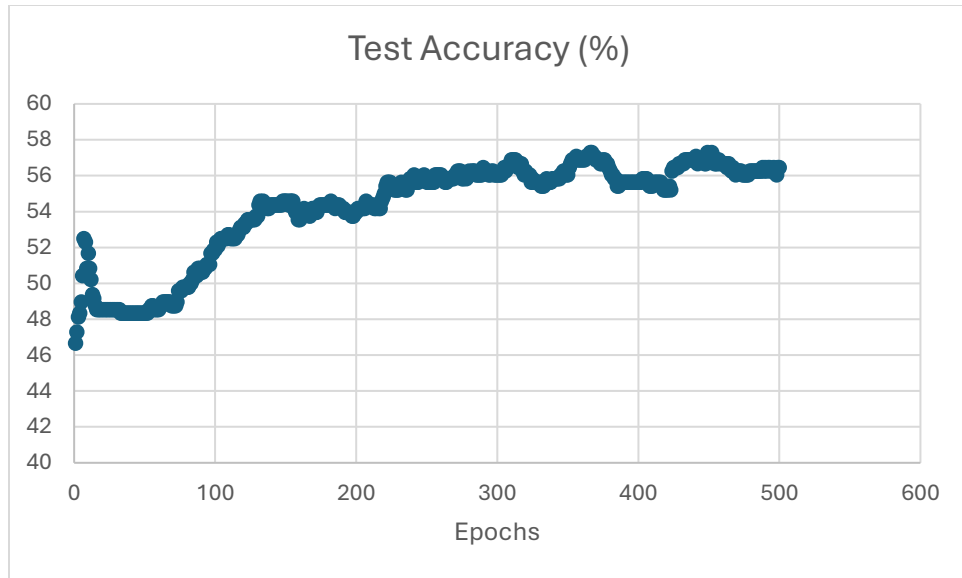
- Νευρώνες εισόδου: 47
- Κρυφά επίπεδα: 2
- Νευρώνες 1ου κρυφού επιπέδου: 10
- Νευρώνες 2ου κρυφού επιπέδου: 10
- Νευρώνες εξόδου: 1
- Ρυθμός μάθησης: 0.05
- Εποχές εκπαίδευσης: 500
- Δεδομένα εκπαίδευσης: 2009/10 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



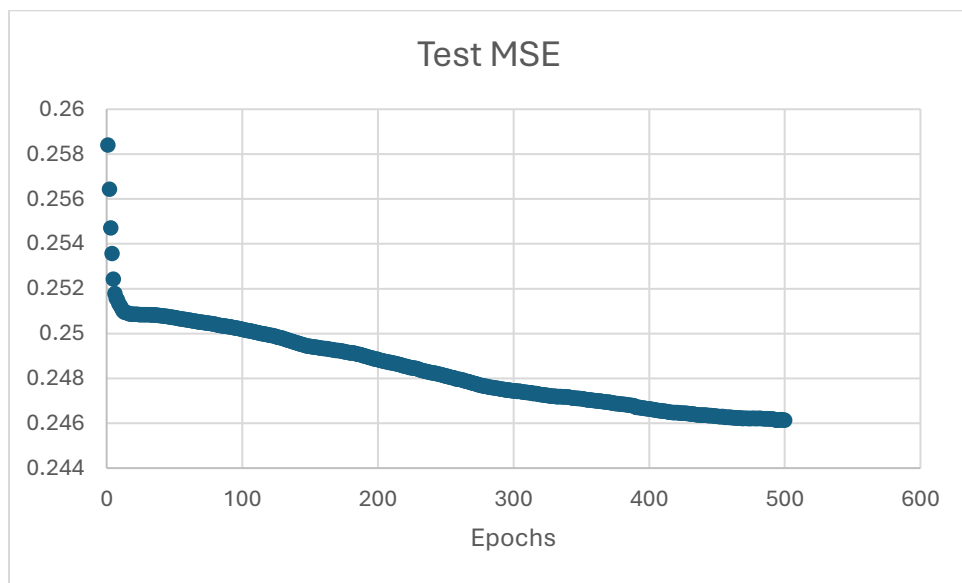
Εικόνα 4.16: Εξέλιξη του Train Accuracy για το MLP με 47 χαρακτηριστικά εισόδου.



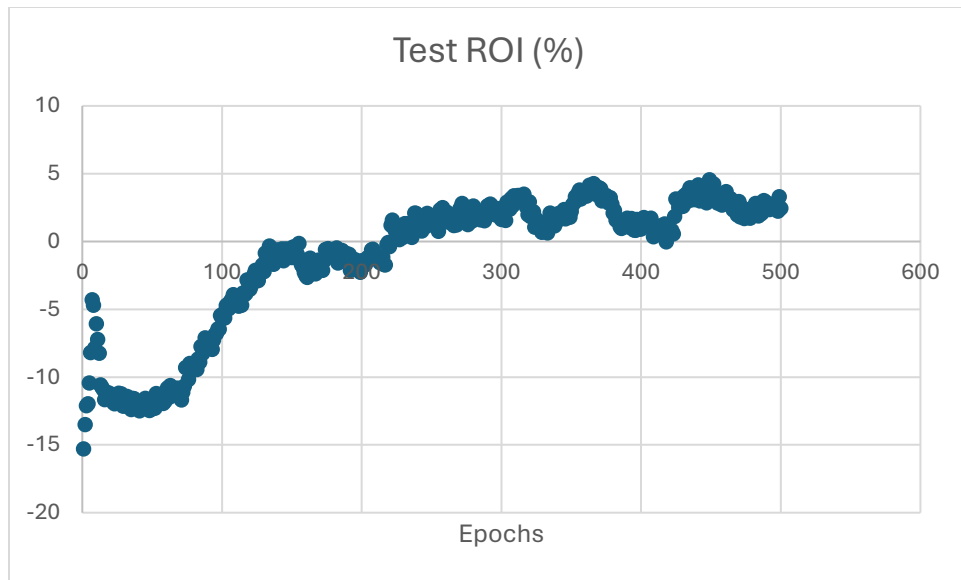
Εικόνα 4.17: Εξέλιξη του Train MSE για το MLP με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.18: Εξέλιξη του Test Accuracy για το MLP με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.19: Εξέλιξη του Test MSE για το MLP με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.20: Εξέλιξη του Test ROI για το MLP με 47 χαρακτηριστικά εισόδου.

Όπως φαίνεται στις Εικόνες 4.16–4.20, από τα αποτελέσματα του MLP με 47 χαρακτηριστικά προκύπτει σαφής βελτίωση σε σχέση με τις προηγούμενες εκδοχές των 26 και 34 χαρακτηριστικών. Η αύξηση του πλήθους των εισόδων φαίνεται ότι εμπλούτισε ουσιαστικά την πληροφορία που δέχεται το μοντέλο, επιτρέποντάς του να αποτυπώσει καλύτερα τη σχέση ανάμεσα στα αγωνιστικά στατιστικά και την τελική έκβαση Over/Under 2.5. Η καμπύλη του Test Accuracy παρουσιάζει σταθερή ανοδική πορεία κατά την εκπαίδευση, ενώ και το Test MSE μειώνεται προοδευτικά, γεγονός που δείχνει πιο σταθερή και αποτελεσματική μάθηση.

Η καλύτερη επίδοση του μοντέλου έφτασε σε Best Test Accuracy 57.84%, που αποτελεί την υψηλότερη τιμή μεταξύ των MLP πειραμάτων. Παράλληλα, το ROI έγινε θετικό και έφτασε στο 4.43%, ένδειξη ότι το συγκεκριμένο μοντέλο δεν βελτίωσε μόνο την προβλεπτική του ικανότητα, αλλά απέκτησε και πρακτική στοιχηματική αξία. Συνεπώς, το πείραμα αυτό δείχνει ότι η πλουσιότερη αναπαράσταση εισόδου συνέβαλε καθοριστικά στη συνολική απόδοση του MLP, καθιστώντας τα 47 χαρακτηριστικά την πιο αποδοτική επιλογή από τις εξεταζόμενες περιπτώσεις.

Συνολικά, όπως φαίνεται στις Εικόνες 4.1–4.20, τα αποτελέσματα των MLP πειραμάτων δείχνουν ότι η μορφή της εισόδου επηρεάζει σημαντικά την απόδοση του μοντέλου. Η αρχική εκδοχή με 26 χαρακτηριστικά λειτούργησε ως βασικό σημείο αναφοράς, αλλά παρουσίασε περιορισμένη ακρίβεια, όπως φαίνεται στην Εικόνα 4.3, και αρνητικό ROI, όπως φαίνεται στην Εικόνα 4.5. Η αύξηση των χαρακτηριστικών σε 34 οδήγησε σε μικρή βελτίωση της ακρίβειας, όπως παρουσιάζεται στις Εικόνες 4.8 και 4.13 για τις δύο αντίστοιχες προσπάθειες, χωρίς όμως να προσφέρει σταθερά θετική στοιχηματική απόδοση, όπως φαίνεται στις Εικόνες 4.10 και 4.15.

Η πιο ουσιαστική βελτίωση εμφανίστηκε στην εκδοχή με 47 χαρακτηριστικά, όπου το μοντέλο αξιοποίησε πλουσιότερη προαγωνιστική πληροφορία για τις δύο ομάδες και τον τρέχοντα αγώνα. Σε αυτή την περίπτωση καταγράφηκε η καλύτερη επίδοση μεταξύ των MLP μοντέλων, με Test Accuracy 57.84%, όπως φαίνεται στην Εικόνα 4.18, και θετικό ROI 4.43%, όπως φαίνεται στην Εικόνα 4.20. Το αποτέλεσμα αυτό δείχνει ότι η προσθήκη πιο ολοκληρωμένων χαρακτηριστικών μπορεί να βελτιώσει τόσο την προβλεπτική όσο και την πρακτική απόδοση του MLP.

Παρόλα αυτά, η συνολική εικόνα των MLP δείχνει ότι, αν και αποτελούν χρήσιμο και σταθερό baseline, η απόδοσή τους παραμένει περιορισμένη για ένα τόσο δύσκολο πρόβλημα όπως η πρόβλεψη Over/Under 2.5. Η επίτευξη θετικού ROI μόνο στην πλουσιότερη αναπαράσταση υποδηλώνει ότι η ποιότητα και η οργάνωση της προαγωνιστικής πληροφορίας είναι καθοριστικές για την τελική απόδοση.

4.2 Αποτελέσματα RBF

4.2.1 RBF με 26 χαρακτηριστικά εισόδου

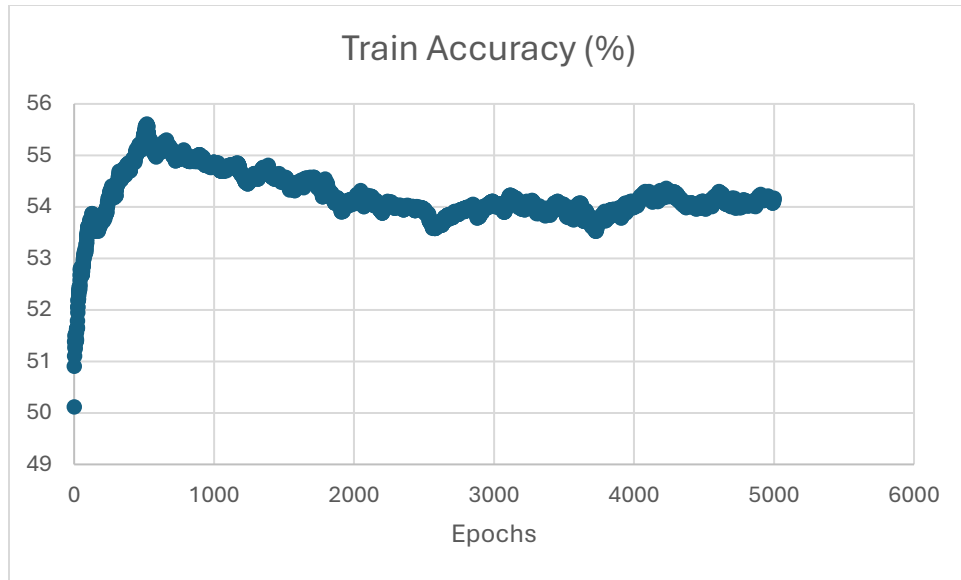
Στο πρώτο πείραμα με Δίκτυα Συναρτήσεων Ακτινικής Βάσης (RBF), χρησιμοποιήθηκε σύνολο δεδομένων με 26 χαρακτηριστικά εισόδου.

Η αρχιτεκτονική του δικτύου βασίστηκε στη χρήση 22 κέντρων (centers), τα οποία αποτελούν τον βασικό μηχανισμό μοντελοποίησης της μη γραμμικότητας στα RBF δίκτυα.

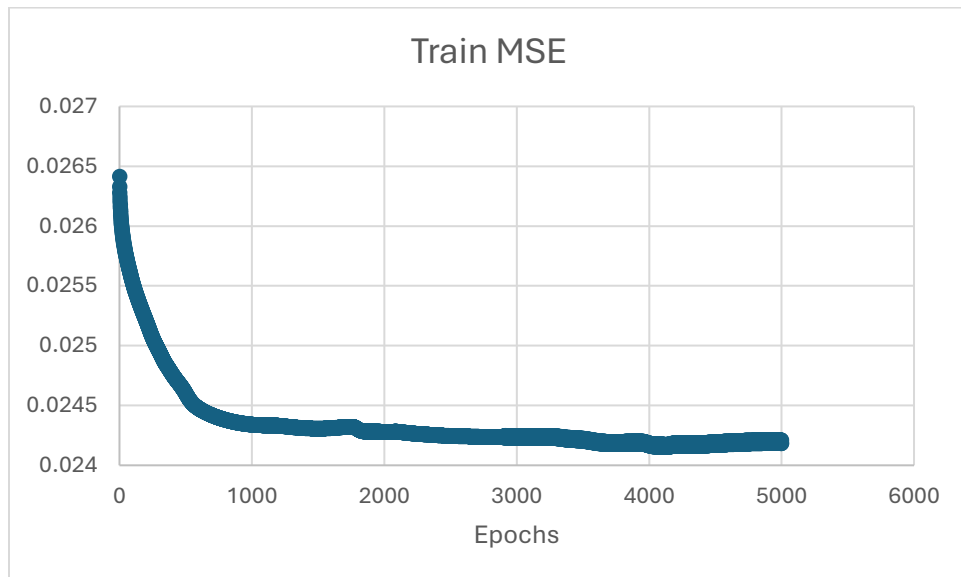
Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν ξεχωριστοί ρυθμοί μάθησης για τα επιμέρους στοιχεία του δικτύου. Συγκεκριμένα, ο ρυθμός μάθησης για τα κέντρα, τα βάρη και το bias, καθώς και για τις παραμέτρους διασποράς (sigma), ορίστηκε ίσος με 0.03. Η εκπαίδευση πραγματοποιήθηκε για 5000 εποχές, ενώ τα δεδομένα εκπαίδευσης καλύπτουν τη χρονική περίοδο από τη σεζόν 2009/10 έως και τη σεζόν 2023/24. Η αξιολόγηση του μοντέλου πραγματοποιήθηκε στη σεζόν 2024/25, διατηρώντας τη χρονική συνέπεια της διαδικασίας.

Συνοπτικά, τα βασικά χαρακτηριστικά του μοντέλου είναι τα εξής:

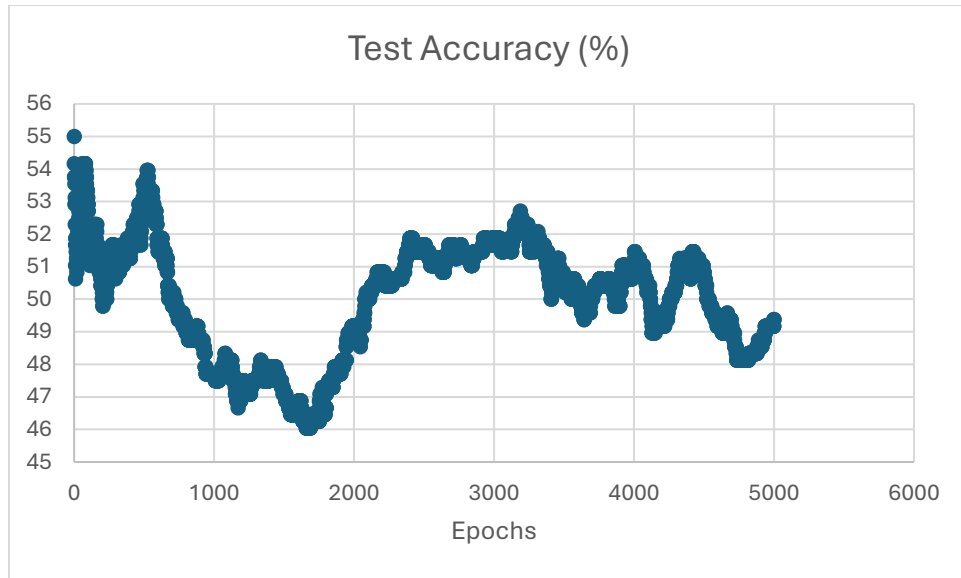
- Διαστάσεις εισόδου: 26
- Αριθμός κέντρων: 22
- Ρυθμός μάθησης κέντρων: 0.03
- Ρυθμός μάθησης βαρών και bias: 0.03
- Ρυθμός μάθησης sigma: 0.03
- Εποχές εκπαίδευσης: 5000
- Δεδομένα εκπαίδευσης: 2009/10 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



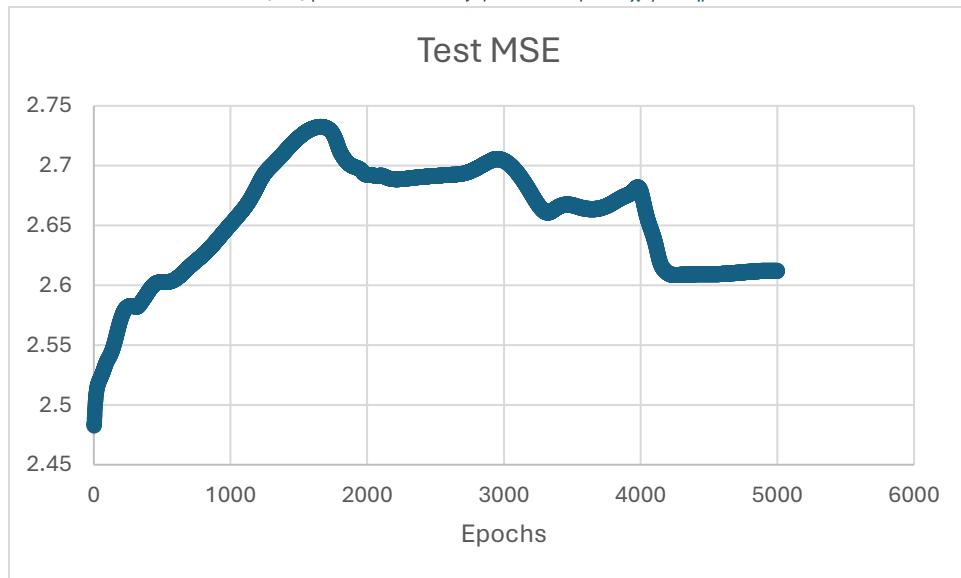
Εικόνα 4.21: Εξέλιξη του Train Accuracy για το RBF με 26 χαρακτηριστικά εισόδου.



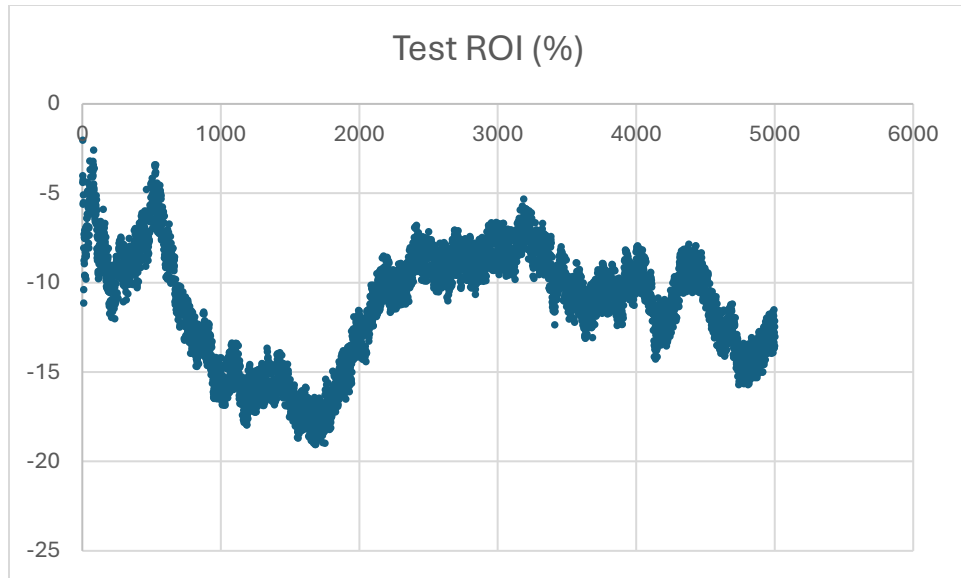
Εικόνα 4.22: Εξέλιξη του Train MSE για το RBF με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.23: Εξέλιξη του Test Accuracy για το RBF με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.24: Εξέλιξη του Test MSE για το RBF με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.25: Εξέλιξη του Test ROI για το RBF με 26 χαρακτηριστικά εισόδου.

Όπως φαίνεται στις Εικόνες 4.21–4.25, από τα αποτελέσματα του RBF με 26 χαρακτηριστικά εισόδου παρατηρείται ότι η απόδοση του μοντέλου παρουσιάζει έντονες διακυμάνσεις κατά τη διάρκεια της εκπαίδευσης. Η καλύτερη τιμή Test Accuracy φτάνει το 54.87%, όπως φαίνεται στην Εικόνα 4.23, όμως η ακρίβεια δεν παραμένει σταθερή και σε αρκετά σημεία υποχωρεί κάτω από το 50%. Παράλληλα, το Test MSE αυξάνεται σημαντικά στις πρώτες εποχές και παραμένει σε σχετικά υψηλά επίπεδα, όπως φαίνεται στην Εικόνα 4.24, γεγονός που δείχνει ότι οι αριθμητικές έξοδοι του μοντέλου δεν ευθυγραμμίζονται σταθερά με τις πραγματικές κλάσεις.

Η στοιχηματική απόδοση παραμένει αρνητική, με καλύτερο ROI -2.95%, όπως φαίνεται στην Εικόνα 4.25. Αυτό δείχνει ότι, παρότι το μοντέλο καταφέρνει σε ορισμένες φάσεις να ξεπεράσει οριακά την τυχαία πρόβλεψη, η απόδοσή του δεν είναι αρκετά σταθερή ώστε να οδηγήσει σε κερδοφόρα στρατηγική. Συνεπώς, η συγκεκριμένη εκδοχή του RBF λειτουργεί ως αρχικό σημείο αναφοράς, αλλά η περιορισμένη είσοδος των 26 χαρακτηριστικών και οι διακυμάνσεις της εκπαίδευσης περιορίζουν την πρακτική της αξία.

4.2.2 RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1

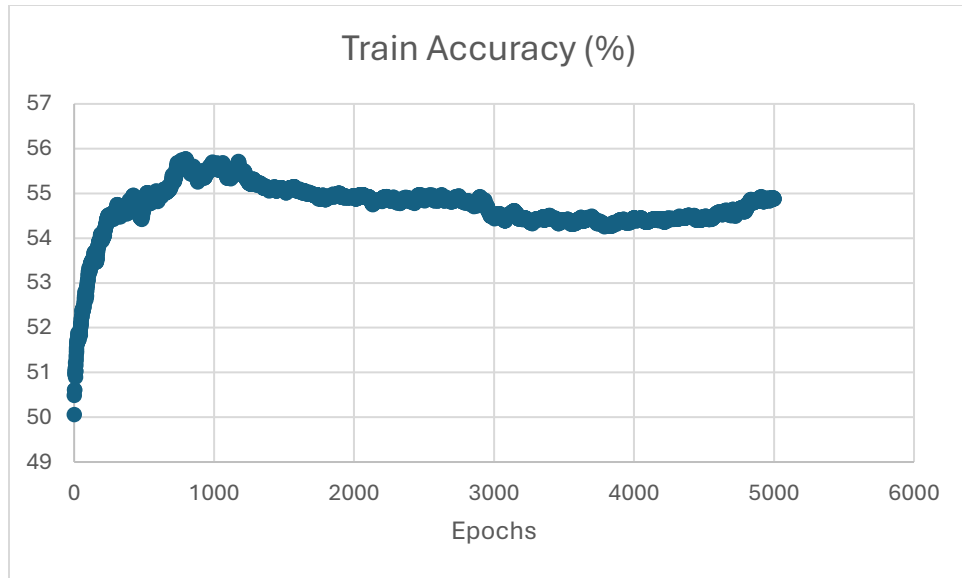
Στο επόμενο πείραμα με Δίκτυα Συναρτήσεων Ακτινικής Βάσης (RBF), χρησιμοποιήθηκε αναπαράσταση εισόδου με 34 χαρακτηριστικά.

Η αρχιτεκτονική του δικτύου παρέμεινε ίδια ως προς τον αριθμό των κέντρων, ο οποίος ορίστηκε σε 22. Η επιλογή αυτή επιτρέπει τη σύγκριση της επίδρασης των επιπλέον χαρακτηριστικών χωρίς να μεταβάλλεται η πολυπλοκότητα του ίδιου του μοντέλου. Όπως και στο προηγούμενο πείραμα, η μη γραμμικότητα προκύπτει μέσω των τοπικών συναρτήσεων βάσης, οι οποίες ενεργοποιούνται ανάλογα με την απόσταση των δεδομένων από τα αντίστοιχα κέντρα.

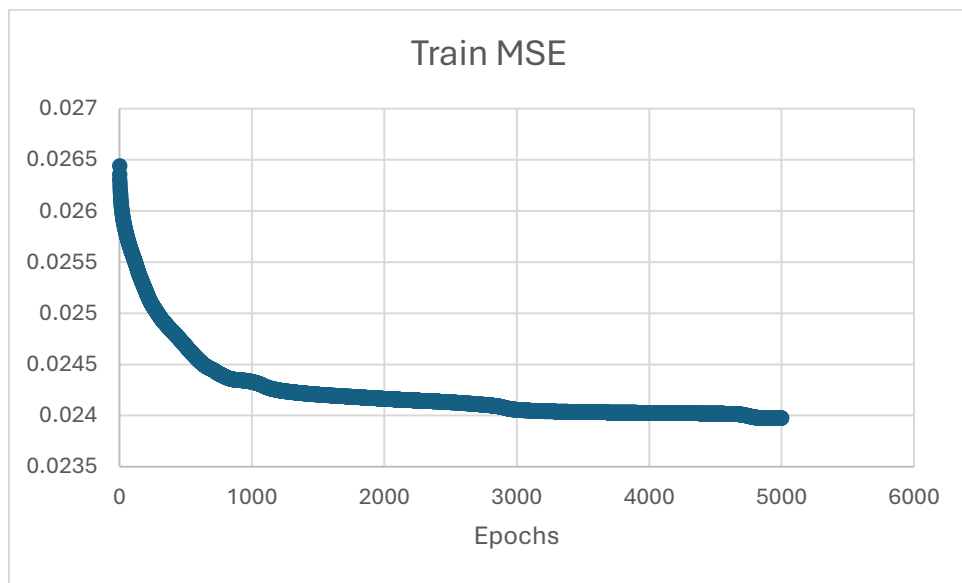
Για την εκπαίδευση του μοντέλου χρησιμοποιήθηκαν ίσοι ρυθμοί μάθησης για όλα τα βασικά στοιχεία του δικτύου. Συγκεκριμένα, ο ρυθμός μάθησης για τα κέντρα, τα βάρη και το bias, καθώς και για την παράμετρο διασποράς (sigma), ορίστηκε ίσος με 0.03. Η εκπαίδευση πραγματοποιήθηκε για 5000 εποχές, ενώ τα δεδομένα εκπαίδευσης καλύπτουν τη χρονική περίοδο από τη σεζόν 2009/10 έως και τη σεζόν 2023/24. Η αξιολόγηση πραγματοποιήθηκε στη σεζόν 2024/25, διατηρώντας τη χρονική συνέπεια της μεθοδολογίας.

Συνοπτικά, τα βασικά χαρακτηριστικά του μοντέλου είναι τα εξής:

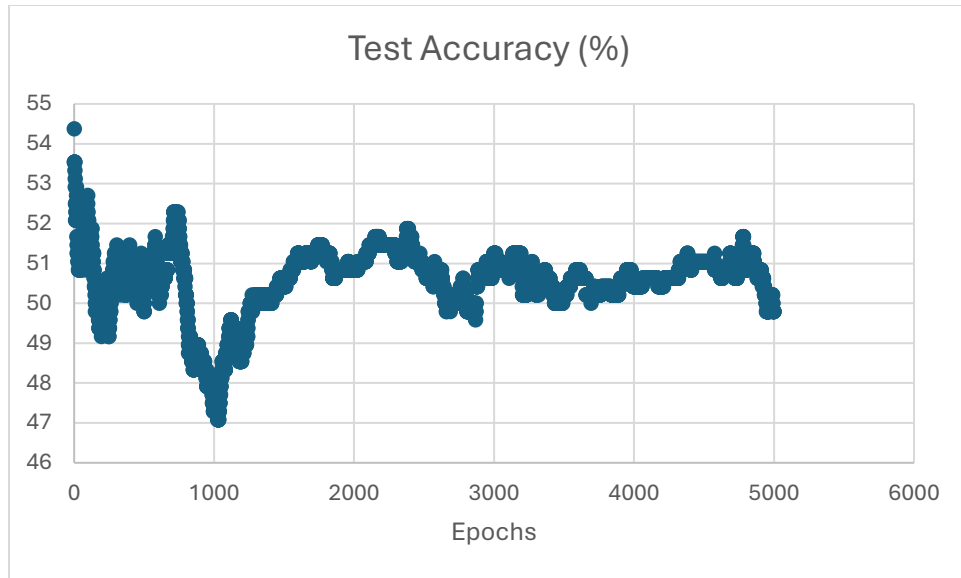
- Διαστάσεις εισόδου: 34
- Αριθμός κέντρων: 22
- Ρυθμός μάθησης κέντρων: 0.03
- Ρυθμός μάθησης βαρών και bias: 0.03
- Ρυθμός μάθησης sigma: 0.03
- Εποχές εκπαίδευσης: 5000
- Δεδομένα εκπαίδευσης: 2009/10 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



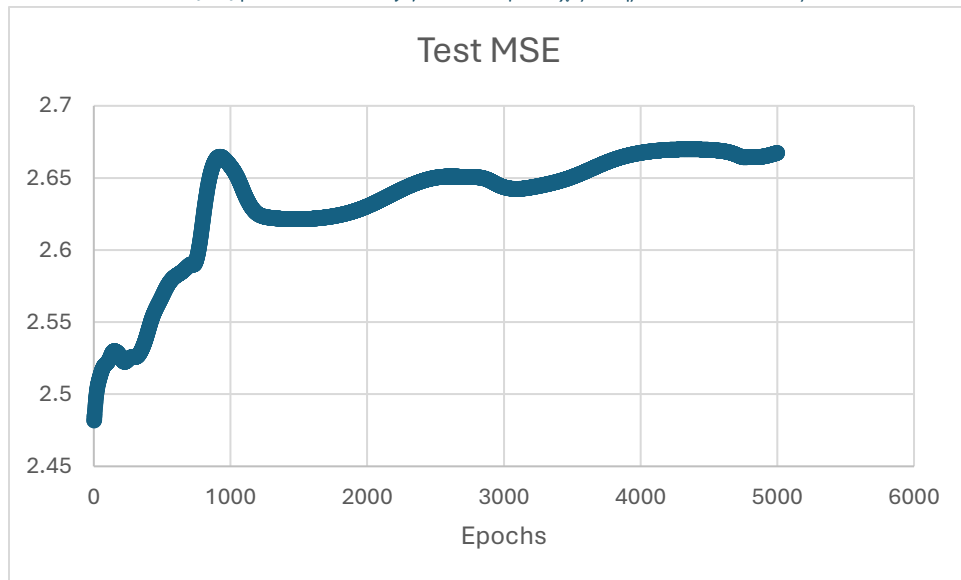
Εικόνα 4.26: Εξέλιξη του Train Accuracy για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



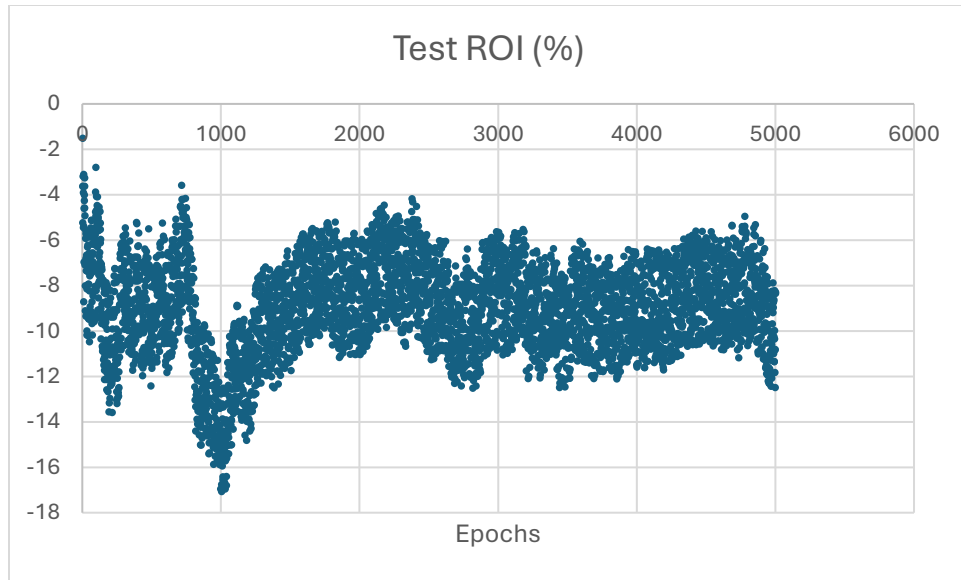
Εικόνα 4.27: Εξέλιξη του Train MSE για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



Εικόνα 4.28: Εξέλιξη του Test Accuracy για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



Εικόνα 4.29: Εξέλιξη του Test MSE για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.



Εικόνα 4.30: Εξέλιξη του Test ROI για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 1.

Όπως φαίνεται στις Εικόνες 4.26–4.30, στην πρώτη προσπάθεια του RBF με 34 χαρακτηριστικά εισόδου, η αύξηση της πληροφορίας σε σχέση με τα 26 χαρακτηριστικά δεν οδήγησε σε ουσιαστική βελτίωση της προβλεπτικής απόδοσης. Η καλύτερη τιμή Test Accuracy έφτασε το 54.41%, όπως φαίνεται στην Εικόνα 4.28, δηλαδή παρέμεινε πολύ κοντά στην προηγούμενη εκδοχή. Παρότι το μοντέλο παρουσιάζει περιόδους σχετικής σταθεροποίησης, η ακρίβεια εμφανίζει διακυμάνσεις και δεν διατηρείται σε υψηλότερα επίπεδα κατά τη διάρκεια της εκπαίδευσης.

Παράλληλα, το Test MSE αυξάνεται σταδιακά και παραμένει σχετικά υψηλό, όπως φαίνεται στην Εικόνα 4.29, γεγονός που δείχνει ότι οι αριθμητικές έξοδοι του μοντέλου δεν βελτιώνονται αντίστοιχα με την ταξινομητική του απόδοση. Το ROI παραμένει αρνητικό, με καλύτερη τιμή -1.04%, όπως φαίνεται στην Εικόνα 4.30, αν και είναι ελαφρώς βελτιωμένο σε σχέση με το RBF των 26 χαρακτηριστικών. Συνεπώς, η προσθήκη χαρακτηριστικών βελτίωσε οριακά τη στοιχηματική απόδοση, αλλά δεν ήταν αρκετή για να καταστήσει το μοντέλο κερδοφόρο.

4.2.3 RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2

Στο επόμενο πείραμα με Δίκτυα Συναρτήσεων Ακτινικής Βάσης (RBF), επιχειρήθηκε η βελτίωση της απόδοσης μέσω της πολυπλοκότητας του μοντέλου. Συγκεκριμένα, χρησιμοποιήθηκε αναπαράσταση εισόδου με 34 χαρακτηριστικά, ίδια με αυτά που παρουσιάστηκαν στο προηγούμενο πείραμα, ενώ αυξήθηκε σημαντικά ο αριθμός των κέντρων από 22 σε 65.

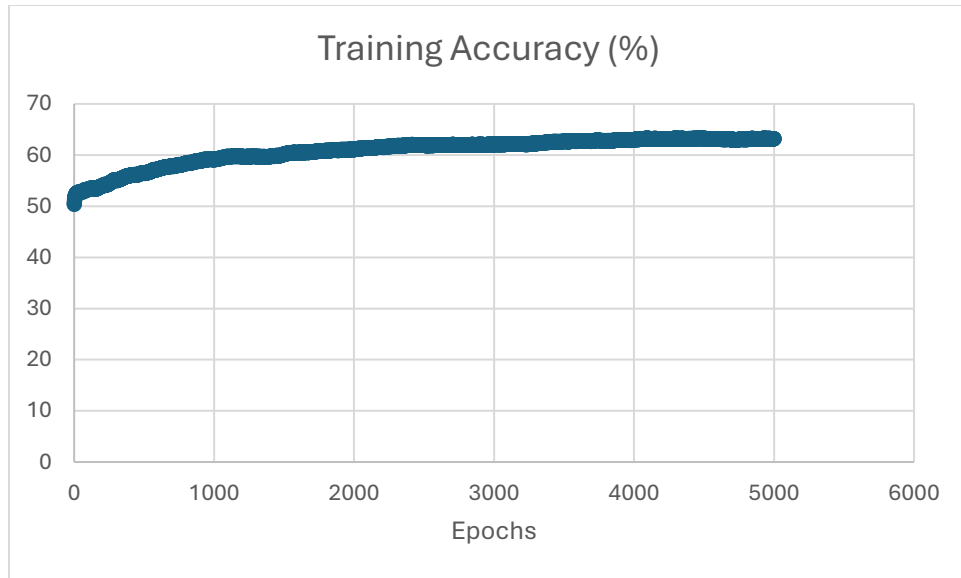
Η αύξηση του αριθμού των κέντρων επιτρέπει στο μοντέλο να καλύπτει μεγαλύτερο μέρος του χώρου χαρακτηριστικών και να αποτυπώνει πιο λεπτομερείς τοπικές δομές στα δεδομένα.

Παράλληλα, για να διατηρηθεί η σταθερότητα της εκπαίδευσης, μειώθηκε ο ρυθμός μάθησης σε 0.005 για όλα τα επιμέρους στοιχεία του δικτύου (κέντρα, βάρη, bias και sigma). Η επιλογή αυτή είναι σημαντική, καθώς σε πιο πολύπλοκα μοντέλα απαιτούνται μικρότερα βήματα ενημέρωσης ώστε να αποφεύγονται φαινόμενα αστάθειας.

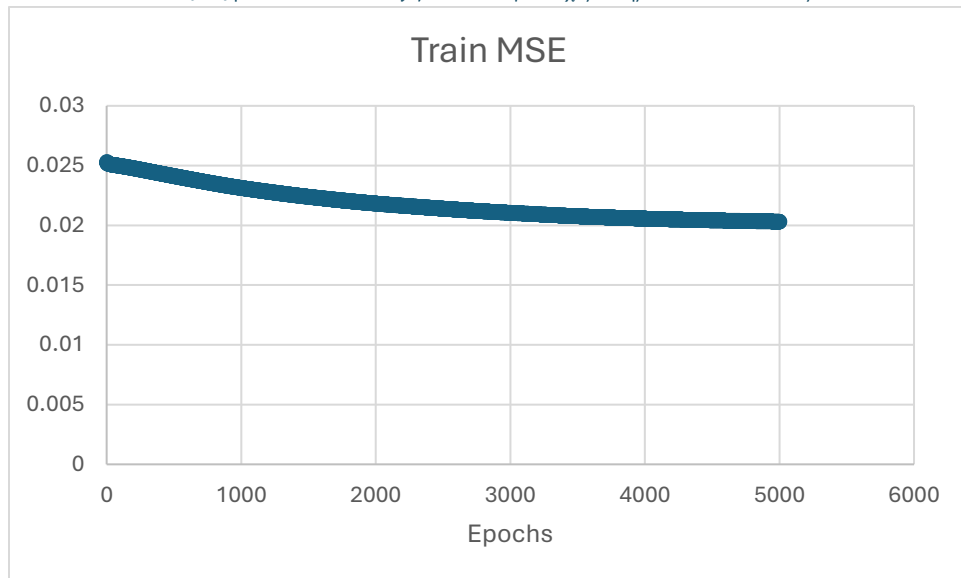
Επιπλέον, διευρύνθηκε το σύνολο εκπαίδευσης, το οποίο πλέον καλύπτει τη χρονική περίοδο από τη σεζόν 2005/06 έως και τη σεζόν 2023/24, προσφέροντας στο μοντέλο μεγαλύτερη ιστορική πληροφορία. Η εκπαίδευση πραγματοποιήθηκε για 5000 εποχές, ενώ η αξιολόγηση έγινε στη σεζόν 2024/25, διατηρώντας τη χρονική συνέπεια της μεθοδολογίας.

Συνοπτικά, τα βασικά χαρακτηριστικά του μοντέλου είναι τα εξής:

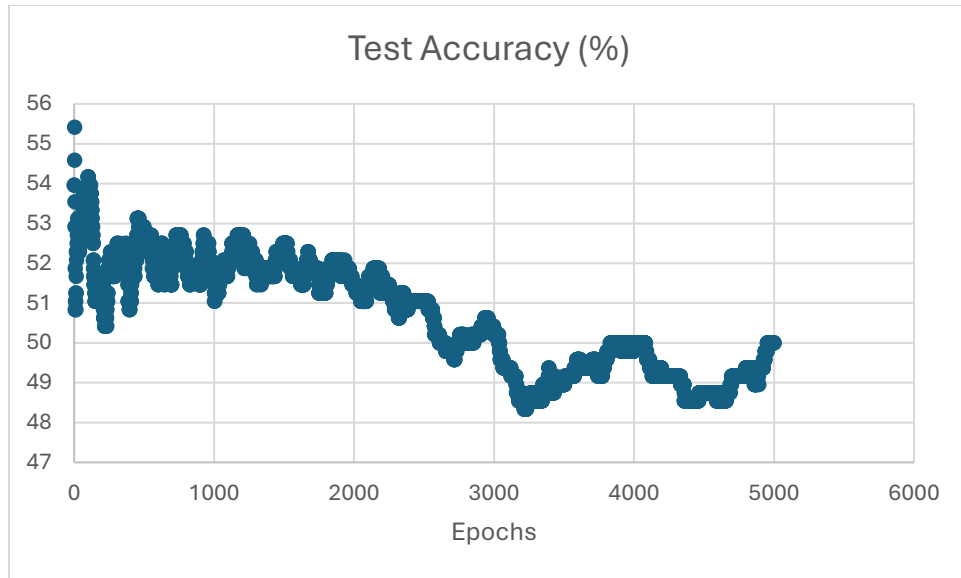
- Διαστάσεις εισόδου: 34
- Αριθμός κέντρων: 65
- Ρυθμός μάθησης κέντρων: 0.005
- Ρυθμός μάθησης βαρών και bias: 0.005
- Ρυθμός μάθησης sigma: 0.005
- Εποχές εκπαίδευσης: 500
- Δεδομένα εκπαίδευσης: 2005/06 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



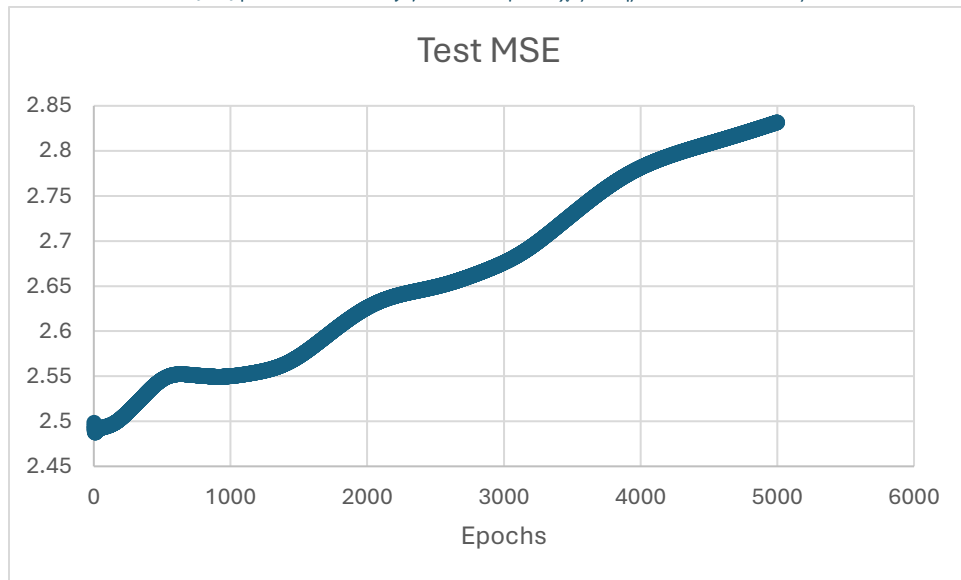
Εικόνα 4.31: Εξέλιξη του Train Accuracy για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



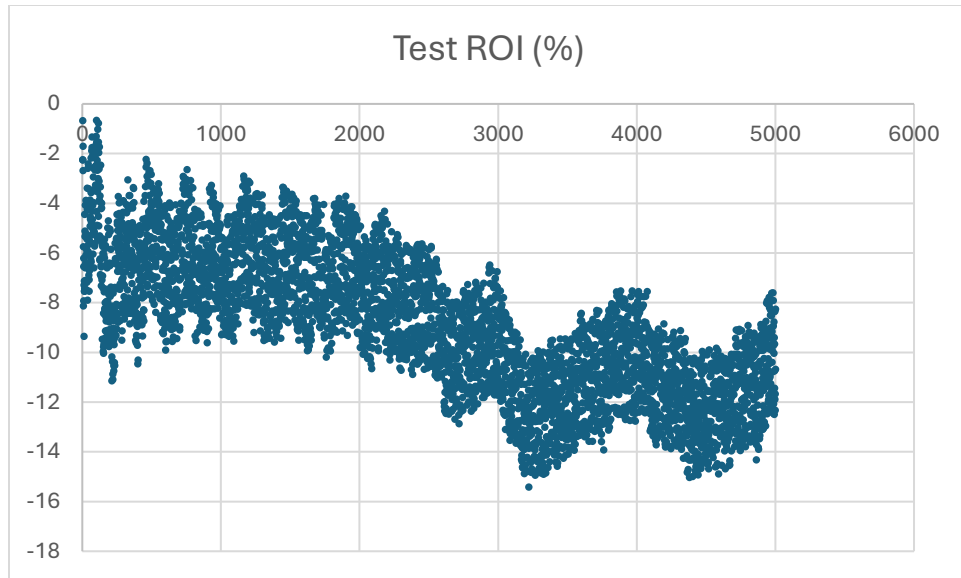
Εικόνα 4.32: Εξέλιξη του Train MSE για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



Εικόνα 4.33: Εξέλιξη του Test Accuracy για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



Εικόνα 4.34: Εξέλιξη του Test MSE για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.



Εικόνα 4.35: Εξέλιξη του Test ROI για το RBF με 34 χαρακτηριστικά εισόδου – Προσπάθεια 2.

Όπως φαίνεται στις Εικόνες 4.31–4.35, στη δεύτερη προσπάθεια με 34 χαρακτηριστικά εισόδου, το RBF δίκτυο χρησιμοποιεί μεγαλύτερο αριθμό κέντρων και χαμηλότερο ρυθμό μάθησης. Η καλύτερη τιμή Test Accuracy φτάνει το 55.38%, όπως φαίνεται στην Εικόνα 4.33, παρουσιάζοντας μικρή βελτίωση σε σχέση με την προηγούμενη προσπάθεια. Ωστόσο, η καμπύλη της ακρίβειας δεν παραμένει σταθερά ανοδική· αντίθετα, μετά τις αρχικές εποχές παρουσιάζει σταδιακή υποχώρηση, γεγονός που δείχνει ότι η αύξηση των κέντρων δεν οδήγησε σε σταθερότερη γενίκευση.

Παράλληλα, το Test MSE αυξάνεται αισθητά κατά τη διάρκεια της εκπαίδευσης, όπως φαίνεται στην Εικόνα 4.34, κάτι που υποδηλώνει ότι οι αριθμητικές έξοδοι του μοντέλου απομακρύνονται σταδιακά από τις πραγματικές τιμές-στόχους. Η συμπεριφορά αυτή δείχνει ότι, παρότι το μοντέλο επιτυγχάνει στιγμιαία καλύτερη ακρίβεια, η συνολική ποιότητα των προβλέψεών του δεν βελτιώνεται με τον ίδιο τρόπο. Το ROI παραμένει αρνητικό, με καλύτερη τιμή -0.57%, όπως φαίνεται στην Εικόνα 4.35, αν και βρίσκεται πιο κοντά στο μηδέν σε σχέση με τις προηγούμενες RBF εκδοχές. Συνεπώς, η αύξηση των κέντρων και η μείωση του ρυθμού μάθησης οδήγησαν σε οριακή βελτίωση, αλλά όχι σε σταθερά κερδοφόρα απόδοση.

4.2.4 RBF με 47 χαρακτηριστικά εισόδου

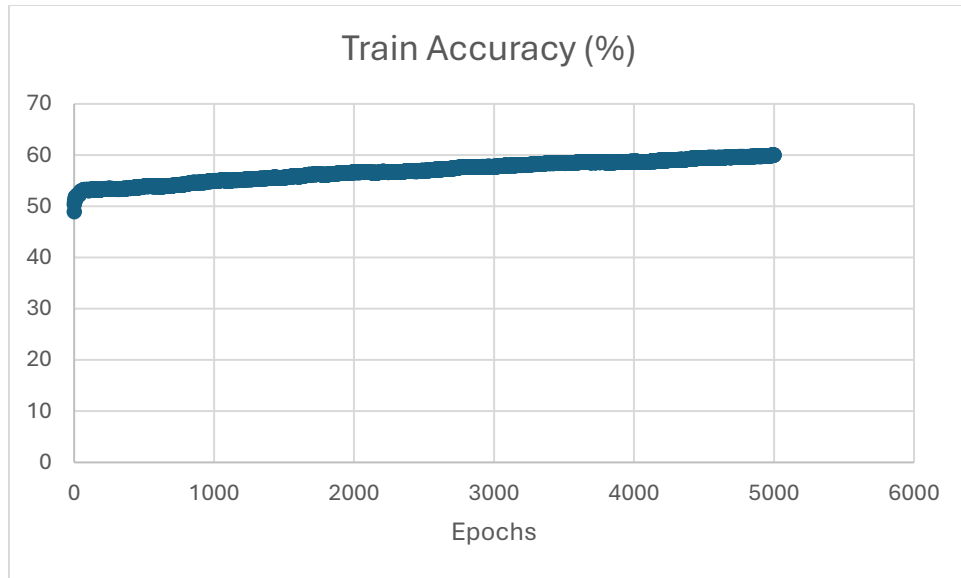
Στο επόμενο πείραμα με Δίκτυα Συναρτήσεων Ακτινικής Βάσης (RBF), χρησιμοποιήθηκε η πιο πλούσια αναπαράσταση εισόδου της εργασίας, με 47 χαρακτηριστικά.

Η αρχιτεκτονική του μοντέλου προσαρμόστηκε ανάλογα με την αυξημένη πολυπλοκότητα της εισόδου. Συγκεκριμένα, ο αριθμός των κέντρων αυξήθηκε σε 75, επιτρέποντας στο δίκτυο να καλύψει μεγαλύτερο και πιο σύνθετο χώρο χαρακτηριστικών. Παράλληλα, οι ρυθμοί μάθησης για όλα τα επιμέρους στοιχεία του μοντέλου (κέντρα, βάρη, bias και sigma) μειώθηκαν σε 0.001, ώστε να διασφαλιστεί η σταθερότητα της εκπαίδευσης σε ένα τόσο υψηλής διάστασης πρόβλημα.

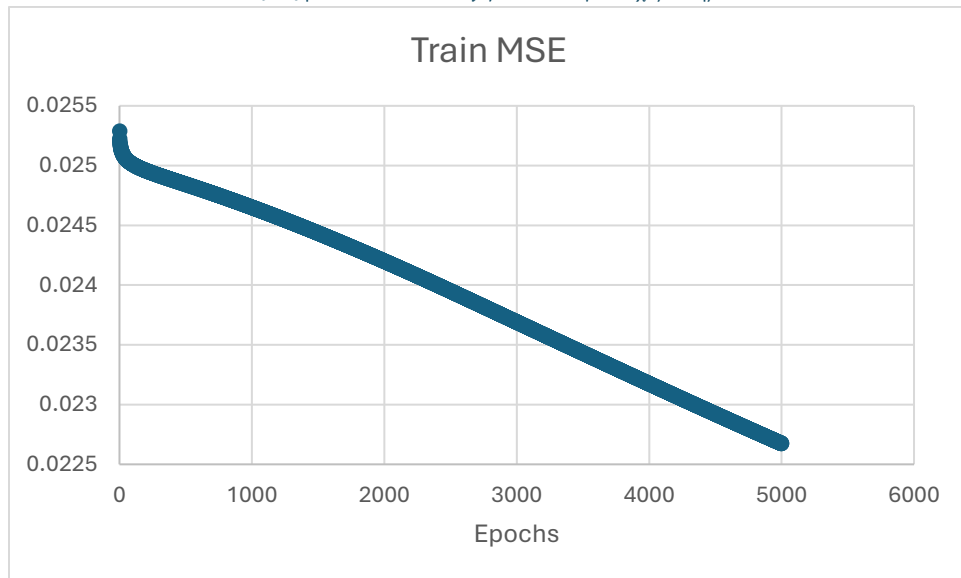
Η εκπαίδευση πραγματοποιήθηκε για 5000 εποχές, ενώ το σύνολο εκπαίδευσης καλύπτει τη χρονική περίοδο από τη σεζόν 2005/06 έως και τη σεζόν 2023/24. Η αξιολόγηση πραγματοποιήθηκε στη σεζόν 2024/25, διατηρώντας τη χρονική συνέπεια της διαδικασίας και αποφεύγοντας οποιαδήποτε μορφή leakage.

Συνοπτικά, τα βασικά χαρακτηριστικά του μοντέλου είναι τα εξής:

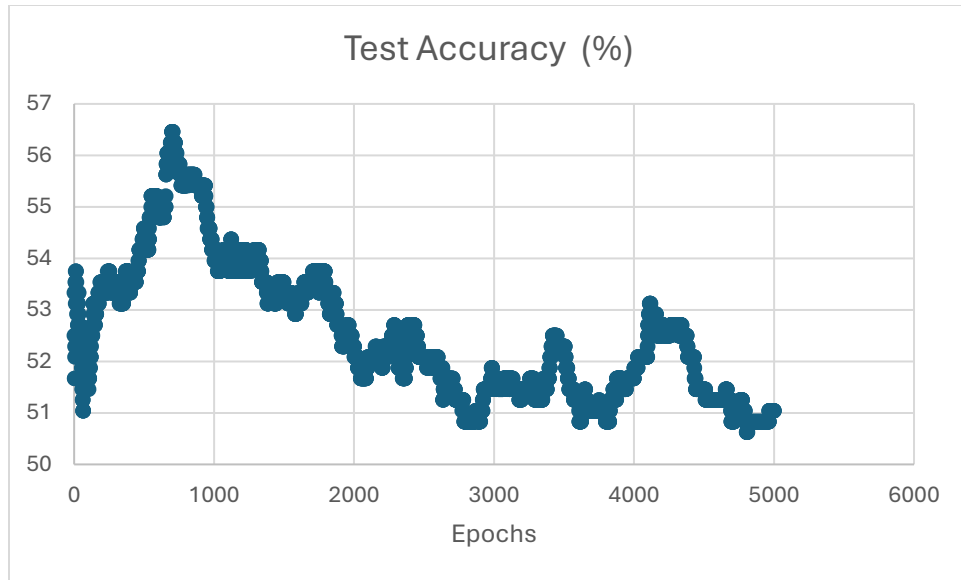
- Διαστάσεις εισόδου: 47
- Αριθμός κέντρων: 75
- Ρυθμός μάθησης κέντρων: 0.001
- Ρυθμός μάθησης βαρών και bias: 0.001
- Ρυθμός μάθησης sigma: 0.001
- Εποχές εκπαίδευσης: 5000
- Δεδομένα εκπαίδευσης: 2005/06 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



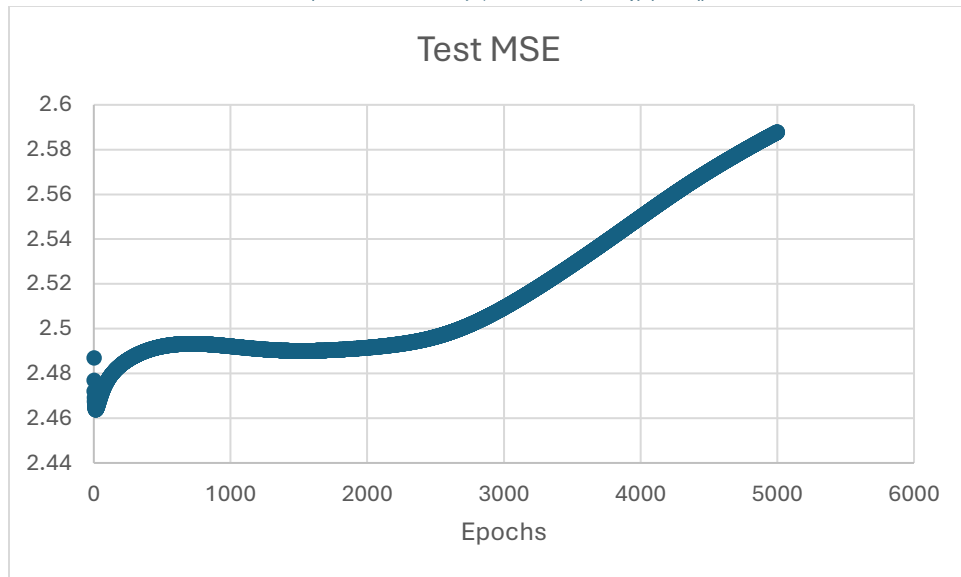
Εικόνα 4.36: Εξέλιξη του Train Accuracy για το RBF με 47 χαρακτηριστικά εισόδου.



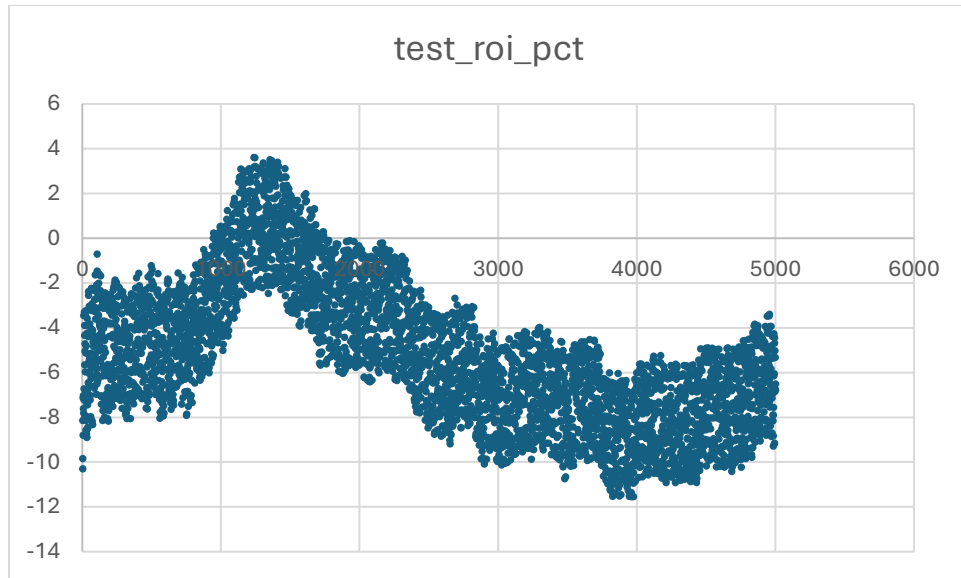
Εικόνα 4.37: Εξέλιξη του Train MSE για το RBF με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.38: Εξέλιξη του Test Accuracy για το RBF με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.39: Εξέλιξη του Test MSE για το RBF με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.40: Εξέλιξη του Test ROI για το RBF με 47 χαρακτηριστικά εισόδου.

Όπως φαίνεται στις Εικόνες 4.36–4.40, στην εκδοχή του RBF με 47 χαρακτηριστικά εισόδου παρατηρείται η καλύτερη συμπεριφορά μεταξύ των RBF πειραμάτων ως προς τη στοιχηματική απόδοση. Η ακρίβεια στα δεδομένα ελέγχου αυξάνεται έντονα στις πρώτες εποχές και φτάνει στη μέγιστη τιμή της σχετικά νωρίς, όπως φαίνεται στην Εικόνα 4.38, πριν αρχίσει σταδιακά να υποχωρεί. Αυτό δείχνει ότι το μοντέλο αρχικά αξιοποιεί αποτελεσματικά την πλουσιότερη προαγωνιστική πληροφορία, αλλά η παρατεταμένη εκπαίδευση δεν οδηγεί σε περαιτέρω βελτίωση της γενίκευσης.

Παράλληλα, το Test MSE παρουσιάζει σταδιακή αύξηση κατά τη διάρκεια της εκπαίδευσης, όπως φαίνεται στην Εικόνα 4.39, γεγονός που δείχνει ότι οι αριθμητικές έξοδοι του μοντέλου απομακρύνονται προοδευτικά από τις πραγματικές τιμές-στόχους. Επομένως, παρότι το μοντέλο επιτυγχάνει καλύτερες στιγμιαίες προβλέψεις σε σχέση με τις προηγούμενες RBF εκδοχές, η ποιότητα των εκτιμήσεων δεν βελτιώνεται σταθερά όσο συνεχίζεται η εκπαίδευση.

Το σημαντικότερο στοιχείο είναι ότι το ROI γίνεται θετικό σε συγκεκριμένο χρονικό διάστημα της εκπαίδευσης, φτάνοντας περίπου στο 3.86%, όπως φαίνεται στην Εικόνα 4.40. Αυτό δείχνει ότι η πλουσιότερη αναπαράσταση των 47 χαρακτηριστικών μπορεί να οδηγήσει σε πρακτικά αξιοποιήσιμες προβλέψεις, έστω και όχι με απόλυτη σταθερότητα σε όλη τη διάρκεια της εκπαίδευσης. Συνεπώς, το RBF με 47 χαρακτηριστικά αποτελεί την ισχυρότερη εκδοχή των RBF πειραμάτων, αλλά η αύξηση του Test MSE υποδηλώνει ότι απαιτείται προσεκτική επιλογή του κατάλληλου σημείου εκπαίδευσης.

Συνολικά, όπως φαίνεται στις Εικόνες 4.21–4.40, τα αποτελέσματα των RBF πειραμάτων δείχνουν ότι η απόδοση του μοντέλου επηρεάζεται τόσο από το πλήθος των χαρακτηριστικών εισόδου όσο και από τον αριθμό των κέντρων. Στις πρώτες εκδοχές, με 26 και 34 χαρακτηριστικά, η ακρίβεια στα δεδομένα ελέγχου κινήθηκε σε μέτρια επίπεδα, όπως φαίνεται στις Εικόνες 4.23, 4.28 και 4.33, ενώ το ROI παρέμεινε αρνητικό, όπως φαίνεται στις Εικόνες 4.25, 4.30 και 4.35. Παρότι σε ορισμένες προσπάθειες παρατηρήθηκε μικρή βελτίωση, η αύξηση των χαρακτηριστικών από μόνη της δεν ήταν αρκετή για να οδηγήσει σε σταθερά κερδοφόρα αποτελέσματα.

Η καλύτερη συμπεριφορά μεταξύ των RBF μοντέλων εμφανίστηκε στην εκδοχή με 47 χαρακτηριστικά εισόδου. Σε αυτή την περίπτωση, η πλουσιότερη προαγωνιστική αναπαράσταση, σε συνδυασμό με μεγαλύτερο αριθμό κέντρων και χαμηλότερο ρυθμό μάθησης, οδήγησε σε θετικό ROI περίπου 3.86%, όπως φαίνεται στην Εικόνα 4.40. Το αποτέλεσμα αυτό δείχνει ότι το RBF μπορεί να αξιοποιήσει μέρος της πρόσθετης πληροφορίας όταν η είσοδος είναι πιο ολοκληρωμένη και η εκπαίδευση γίνεται πιο συντηρητικά.

Ωστόσο, η ανάλυση του Test MSE δείχνει ότι τα RBF μοντέλα δεν παρουσιάζουν πάντα σταθερή βελτίωση ως προς την ποιότητα των αριθμητικών εκτιμήσεών τους. Σε αρκετές περιπτώσεις το MSE αυξάνεται κατά τη διάρκεια της εκπαίδευσης, όπως φαίνεται στις Εικόνες 4.24, 4.29, 4.34 και 4.39, ακόμη και όταν η ακρίβεια ή το ROI βελτιώνονται προσωρινά. Αυτό σημαίνει ότι το μοντέλο μπορεί να πετυχαίνει ορισμένες σωστές ταξινομήσεις, αλλά οι προβλέψεις του δεν γίνονται απαραίτητα πιο καλά ευθυγραμμισμένες με τις πραγματικές τιμές-στόχους.

Επομένως, τα RBF δίκτυα αποτελούν χρήσιμη baseline προσέγγιση για το πρόβλημα Over/Under 2.5, αλλά η απόδοσή τους παραμένει ευαίσθητη στην επιλογή χαρακτηριστικών, κέντρων και ρυθμών μάθησης. Η θετική απόδοση της εκδοχής των 47 χαρακτηριστικών δείχνει ότι υπάρχει αξιοποιήσιμη πληροφορία στην εμπλουτισμένη προαγωνιστική αναπαράσταση, όμως η αστάθεια του Test MSE υποδηλώνει ότι απαιτείται προσεκτική παρακολούθηση της εκπαίδευσης και επιλογή του κατάλληλου σημείου τερματισμού.

4.3 Transformer

4.3.1 Transformer με 26 χαρακτηριστικά εισόδου

Στο επόμενο πείραμα χρησιμοποιείται αρχιτεκτονική Transformer Encoder για την πρόβλεψη του Over/Under 2.5, βασισμένη σε ακολουθιακή αναπαράσταση προηγούμενων αγώνων. Σε αυτή την εκδοχή, η είσοδος του μοντέλου περιορίζεται σε 26 χαρακτηριστικά ανά χρονικό βήμα.

Η χρήση μικρότερου πλήθους χαρακτηριστικών επιτρέπει την αξιολόγηση του κατά πόσο ένα Transformer μπορεί να αξιοποιήσει τη χρονική πληροφορία ακόμη και όταν η αναπαράσταση εισόδου είναι λιγότερο πλούσια. Με τον τρόπο αυτό, το πείραμα λειτουργεί ως σημείο σύγκρισης με πιο εκτεταμένες αναπαραστάσεις, όπως η εκδοχή των 47 χαρακτηριστικών.

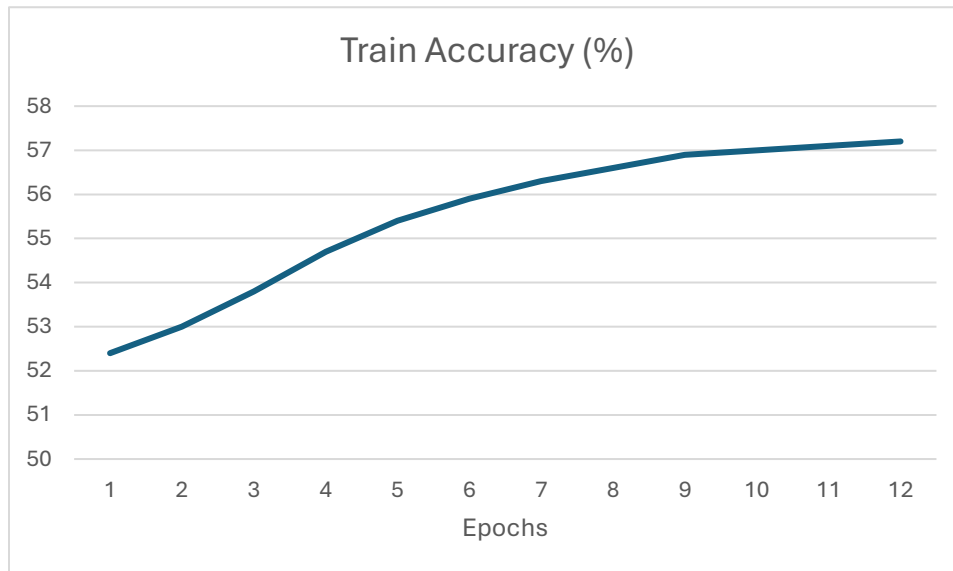
Η είσοδος αποτελείται από ακολουθίες μήκους $k = 10$, όπου κάθε στοιχείο της ακολουθίας περιγράφει έναν προηγούμενο αγώνα από την οπτική της ομάδας. Η πληροφορία της ακολουθίας ενσωματώνεται μέσω μηχανισμών προσοχής και συνοψίζεται μέσω pooling τύπου CLS πριν την τελική πρόβλεψη.

Η αρχιτεκτονική και η διαδικασία εκπαίδευσης παραμένουν ίδιες με αυτές που χρησιμοποιήθηκαν στα επόμενα Transformer πειράματα, ώστε η σύγκριση να εστιάζει αποκλειστικά στην επίδραση της αναπαράστασης εισόδου και όχι σε αλλαγές υπερπαραμέτρων.

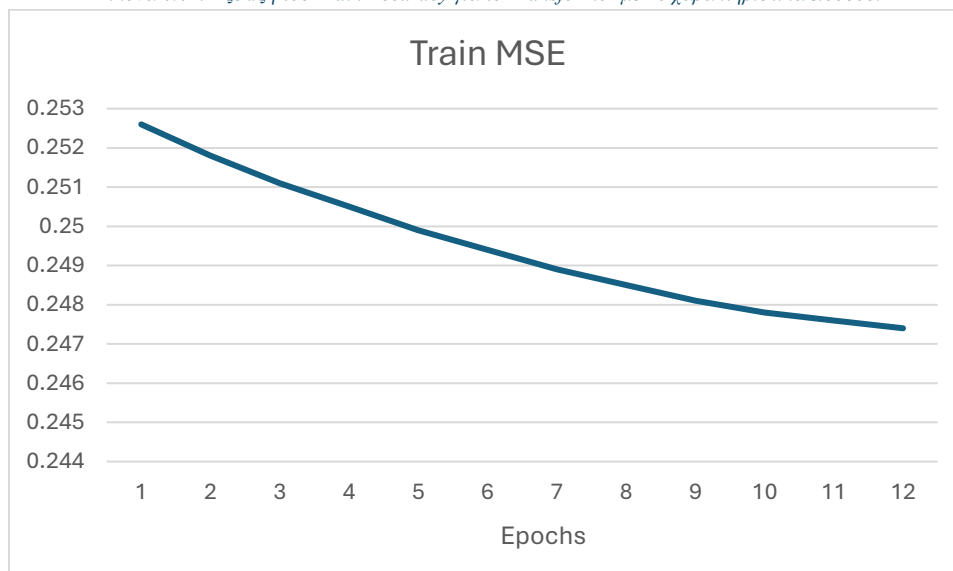
Συνοπτικά, τα βασικά χαρακτηριστικά του μοντέλου είναι τα εξής:

- Διαστάσεις εισόδου: 26
- Παράθυρο υπολογισμού rolling χαρακτηριστικών (k): 10 προηγούμενοι αγώνες
- Batch size: 64
- Εποχές εκπαίδευσης: 12
- Patience: 8
- Διάσταση αναπαράστασης (d_{model}): 64
- Αριθμός κεφαλών προσοχής: 2
- Αριθμός επιπέδων Transformer: 1
- Dropout: 0.2
- Τύπος pooling: cls
- Ρυθμός μάθησης: 0.001
- Decision threshold: 0.5

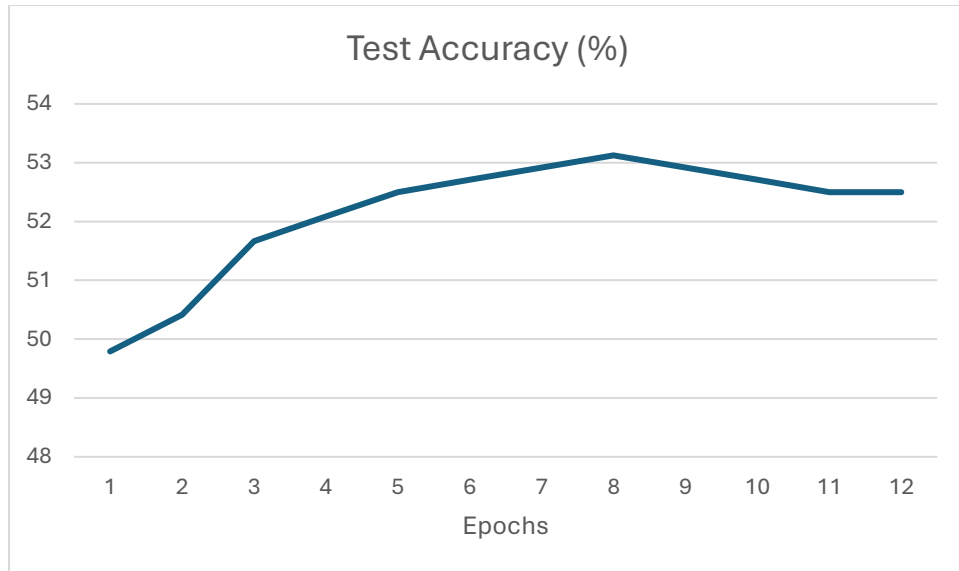
- Threshold strategy: fixed
- Δεδομένα εκπαίδευσης: 2005/06 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



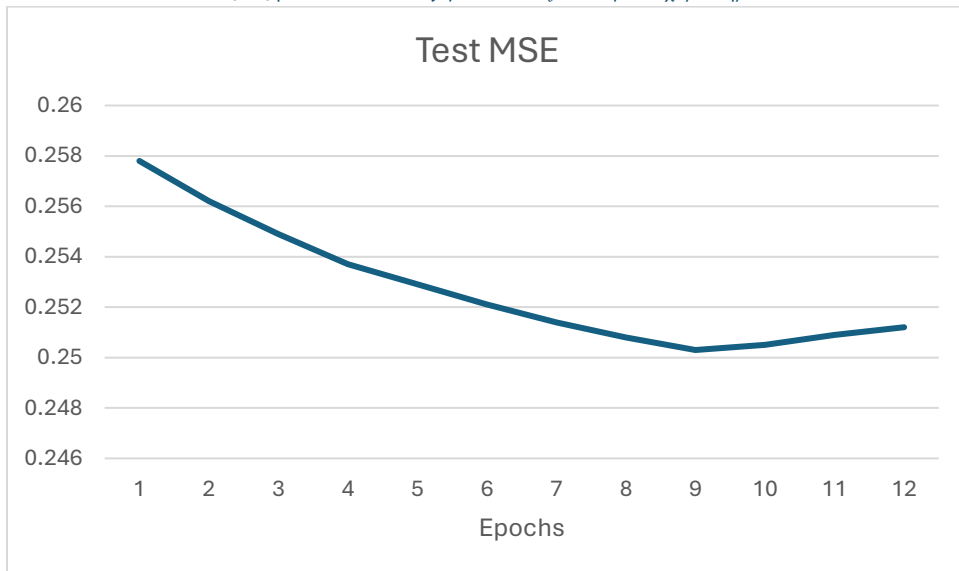
Εικόνα 4.41: Εξέλιξη του Train Accuracy για το Transformer με 26 χαρακτηριστικά εισόδου.



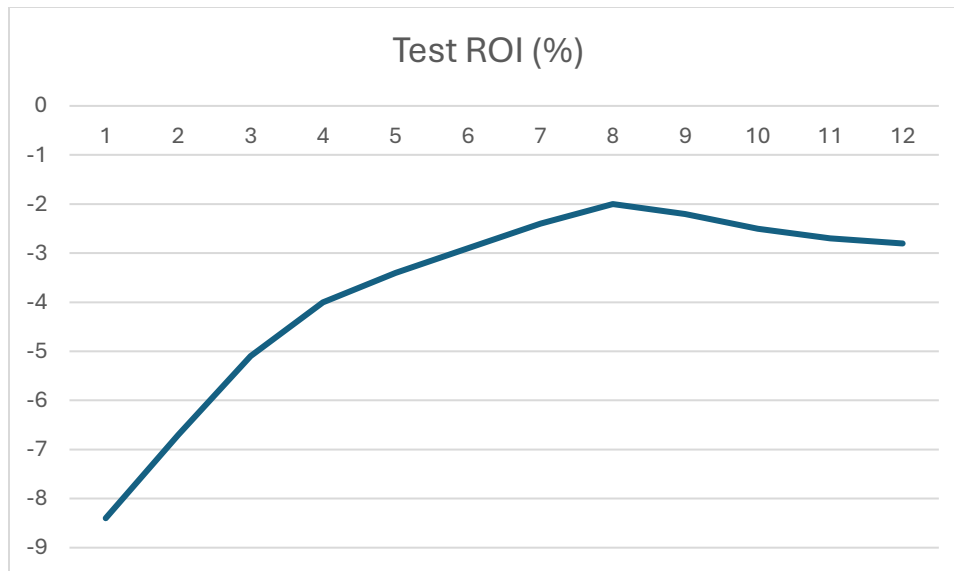
Εικόνα 4.42: Εξέλιξη του Train MSE για το Transformer με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.43: Εξέλιξη του Test Accuracy για το Transformer με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.44: Εξέλιξη του Test MSE για το Transformer με 26 χαρακτηριστικά εισόδου.



Εικόνα 4.45: Εξέλιξη του Test ROI για το Transformer με 26 χαρακτηριστικά εισόδου.

Όπως φαίνεται στις Εικόνες 4.41–4.45, από τα αποτελέσματα του Transformer με 26 χαρακτηριστικά εισόδου παρατηρείται ότι η ακρίβεια στα δεδομένα ελέγχου αυξάνεται σταδιακά στις πρώτες εποχές και φτάνει στη μέγιστη τιμή της περίπου στην 8η εποχή. Η καλύτερη τιμή Test Accuracy είναι 53.36%, όπως φαίνεται στην Εικόνα 4.43, ενώ στη συνέχεια η απόδοση παρουσιάζει μικρή υποχώρηση. Το Test MSE μειώνεται κατά τις πρώτες εποχές, όπως φαίνεται στην Εικόνα 4.44, γεγονός που δείχνει αρχική βελτίωση της προσαρμογής του μοντέλου, αλλά προς το τέλος της εκπαίδευσης εμφανίζει μικρή αύξηση.

Το ROI βελτιώνεται σημαντικά σε σχέση με τις πρώτες εποχές, όμως παραμένει αρνητικό, με καλύτερη τιμή -1.46%, όπως φαίνεται στην Εικόνα 4.45. Αυτό δείχνει ότι το μοντέλο καταφέρνει να αξιοποιήσει μέρος της πληροφορίας των 26 χαρακτηριστικών, αλλά όχι σε βαθμό που να παράγει κερδοφόρα στοιχηματική στρατηγική. Συνεπώς, η συγκεκριμένη εκδοχή του Transformer λειτουργεί ως αρχικό σημείο αναφοράς, αλλά η περιορισμένη μορφή εισόδου δεν επαρκεί για ουσιαστική υπεροχή στο πρόβλημα Over/Under 2.5.

4.3.2 Transformer με 34 χαρακτηριστικά εισόδου

Στο επόμενο πείραμα εφαρμόζεται η ίδια αρχιτεκτονική Transformer Encoder για την πρόβλεψη του Over/Under 2.5, αυτή τη φορά με είσοδο 34 χαρακτηριστικών ανά χρονικό βήμα.

Η χρήση 34 χαρακτηριστικών επιτρέπει την ενσωμάτωση περισσότερης πληροφορίας σχετικά με επιθετικές και αμυντικές τάσεις, πρόσφατη φόρμα και venue-aware συμπεριφορά, χωρίς όμως να φτάνει στο επίπεδο των 47 χαρακτηριστικών. Με τον τρόπο αυτό, το συγκεκριμένο πείραμα λειτουργεί ως ενδιάμεσο στάδιο, επιτρέποντας τη διερεύνηση του πώς η αύξηση της πληροφορίας εισόδου επηρεάζει την απόδοση του Transformer.

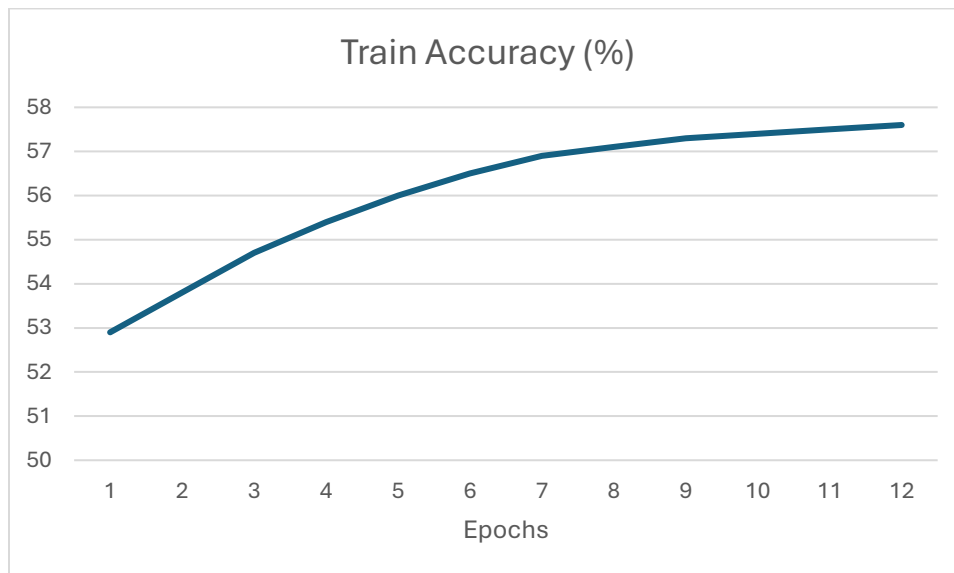
Όπως και στην προηγούμενη περίπτωση, η είσοδος αποτελείται από ακολουθίες μήκους $k = 10$, οι οποίες περιγράφουν τους προηγούμενους αγώνες των ομάδων. Η χρονική δομή της ακολουθίας αξιοποιείται μέσω μηχανισμών self-attention, ενώ η τελική αναπαράσταση εξάγεται μέσω pooling τύπου CLS.

Η αρχιτεκτονική και οι υπερπαράμετροι παραμένουν ίδιες με τα υπόλοιπα Transformer πειράματα, ώστε να εξασφαλίζεται δίκαιη σύγκριση μεταξύ διαφορετικών αναπαραστάσεων εισόδου.

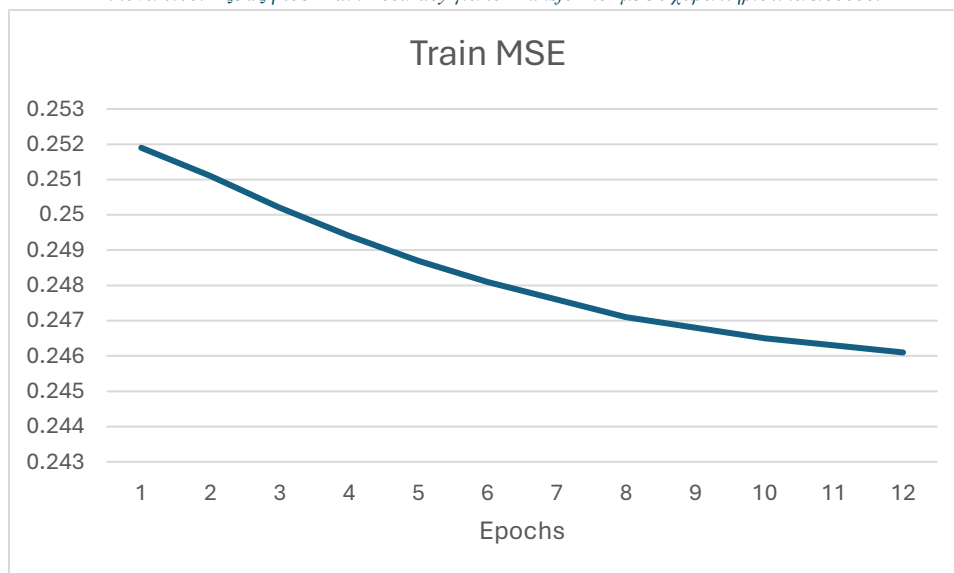
Συνοπτικά, τα βασικά χαρακτηριστικά του μοντέλου είναι τα εξής:

- Διαστάσεις εισόδου: 34
- Παράθυρο υπολογισμού rolling χαρακτηριστικών (k): 10 προηγούμενοι αγώνες
- Batch size: 64
- Εποχές εκπαίδευσης: 12
- Patience: 8
- Διάσταση αναπαράστασης (d_{model}): 64
- Αριθμός κεφαλών προσοχής: 2
- Αριθμός επιπέδων Transformer: 1
- Dropout: 0.2
- Τύπος pooling: cls
- Ρυθμός μάθησης: 0.001
- Decision threshold: 0.5

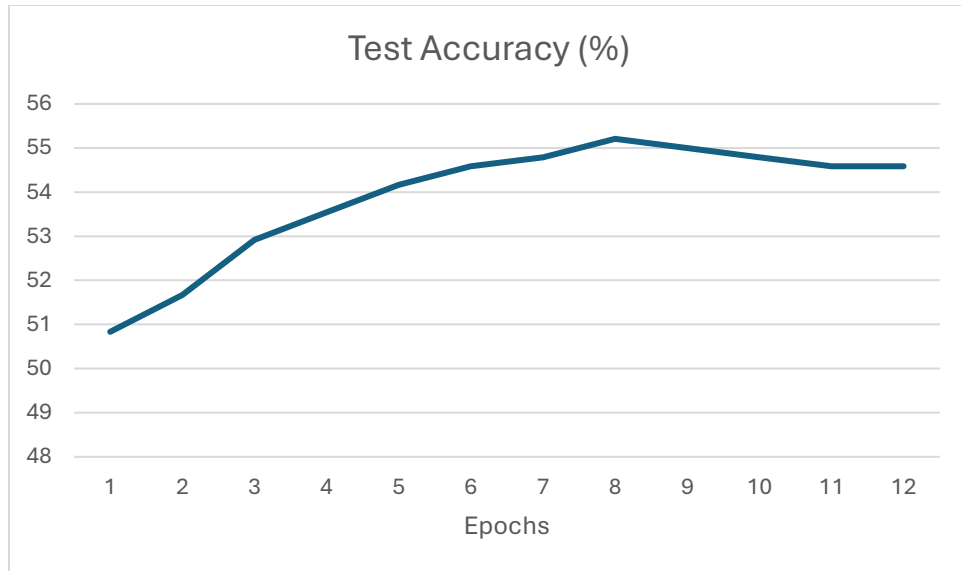
- Threshold strategy: fixed
- Δεδομένα εκπαίδευσης: 2005/06 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



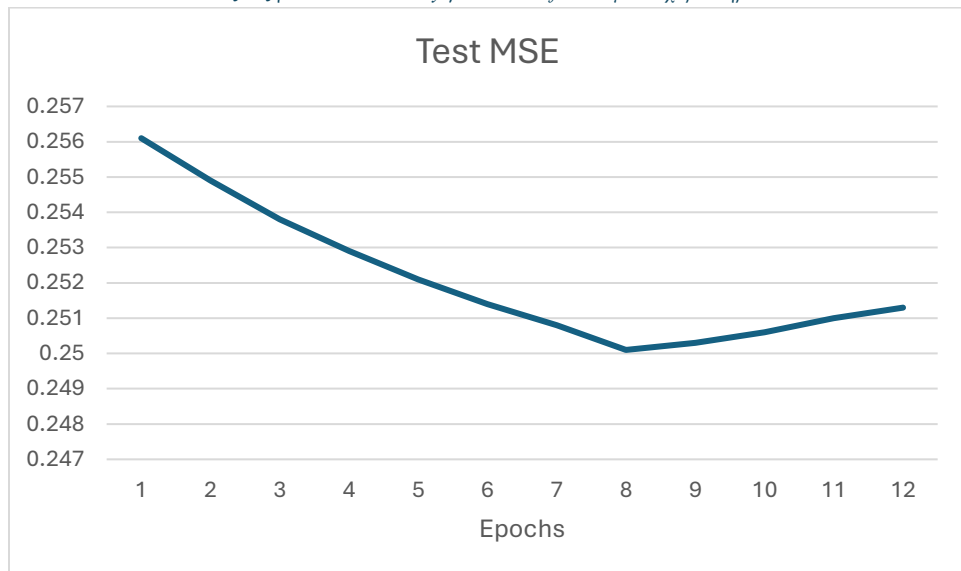
Εικόνα 4.46: Εξέλιξη του Train Accuracy για το Transformer με 34 χαρακτηριστικά εισόδου.



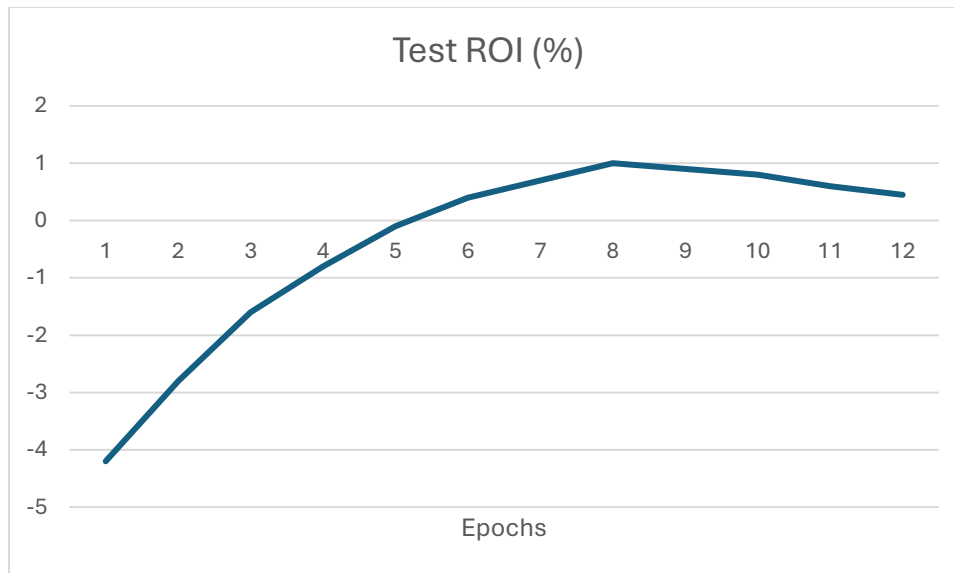
Εικόνα 4.47: Εξέλιξη του Train MSE για το Transformer με 34 χαρακτηριστικά εισόδου.



Εικόνα 4.48: Εξέλιξη του Test Accuracy για το Transformer με 34 χαρακτηριστικά εισόδου.



Εικόνα 4.49: Εξέλιξη του Test MSE για το Transformer με 34 χαρακτηριστικά εισόδου.



Εικόνα 4.50: Εξέλιξη του Test ROI για το Transformer με 34 χαρακτηριστικά εισόδου.

Όπως φαίνεται στις Εικόνες 4.46–4.50, από τα αποτελέσματα του Transformer με 34 χαρακτηριστικά εισόδου παρατηρείται βελτίωση σε σχέση με την εκδοχή των 26 χαρακτηριστικών. Η καλύτερη τιμή Test Accuracy φτάνει το 55.14%, όπως φαίνεται στην Εικόνα 4.48, ενώ το ROI γίνεται θετικό και φτάνει το 1.02%, όπως φαίνεται στην Εικόνα 4.50. Η εξέλιξη αυτή δείχνει ότι η προσθήκη επιπλέον χαρακτηριστικών βοηθά το μοντέλο να αξιοποιήσει καλύτερα την προαγωνιστική πληροφορία και να βελτιώσει τόσο την ταξινομητική όσο και τη στοιχηματική του απόδοση.

Παράλληλα, το Test MSE μειώνεται σταθερά μέχρι περίπου την 8η εποχή και στη συνέχεια εμφανίζει μικρή αύξηση, όπως φαίνεται στην Εικόνα 4.49. Η συμπεριφορά αυτή δείχνει ότι το μοντέλο βελτιώνει αρχικά την ποιότητα των εκτιμήσεών του, αλλά η περαιτέρω εκπαίδευση δεν οδηγεί σε ανάλογη βελτίωση. Συνεπώς, η συγκεκριμένη εκδοχή του Transformer παρουσιάζει πιο ισορροπημένη συμπεριφορά από την εκδοχή των 26 χαρακτηριστικών, αλλά εξακολουθεί να απαιτεί προσεκτική επιλογή του σημείου εκπαίδευσης όπου η γενίκευση είναι καλύτερη.

4.3.3 Transformer με 47 χαρακτηριστικά εισόδου

Στο επόμενο πείραμα χρησιμοποιήθηκε αρχιτεκτονική Transformer Encoder για την πρόβλεψη του Over/Under 2.5, αξιοποιώντας την πιο πλήρη προαγωνιστική αναπαράσταση της εργασίας. Σε αντίθεση με τις ακολουθιακές εκδοχές, το συγκεκριμένο μοντέλο δεν δέχεται ως είσοδο ξεχωριστά tokens προηγούμενων αγώνων, αλλά ένα ενιαίο pre-match token που περιγράφει το τρέχον fixture μέσω 47 χαρακτηριστικών.

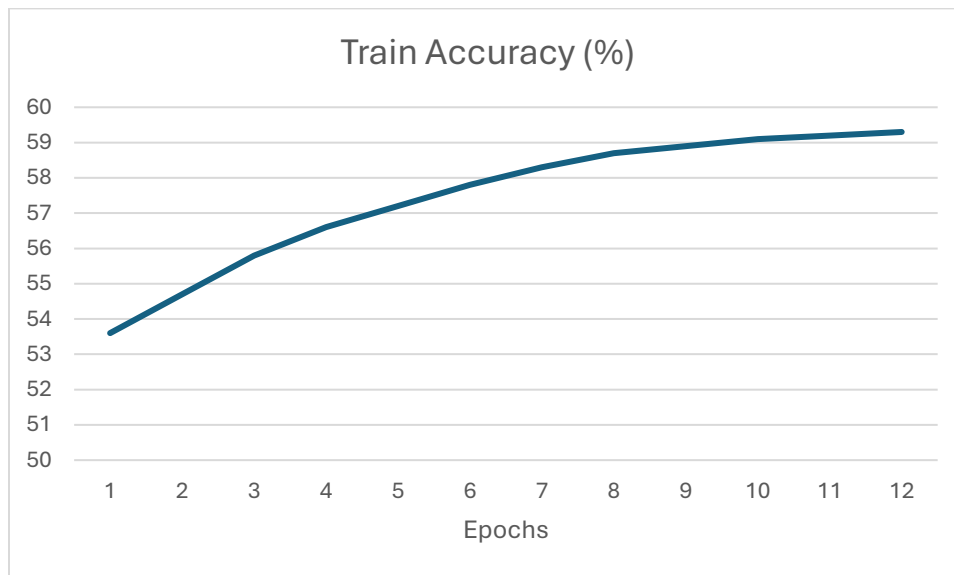
Η παράμετρος $k = 10$ δεν αντιστοιχεί εδώ στο μήκος της ακολουθίας που εισέρχεται στον Transformer, αλλά στο πλήθος των προηγούμενων αγώνων που χρησιμοποιούνται για τον υπολογισμό των rolling στατιστικών χαρακτηριστικών. Με άλλα λόγια, οι πρόσφατοι μέσοι όροι, οι τάσεις φόρμας και ορισμένα χαρακτηριστικά Over/Under υπολογίζονται με βάση τους τελευταίους k αγώνες κάθε ομάδας, πριν από τον υπό πρόβλεψη αγώνα.

Η είσοδος του μοντέλου αποτελείται από ένα διάνυσμα 47 χαρακτηριστικών, το οποίο οργανώνεται ως ένα token. Το token αυτό συνδυάζει στατιστικά χαρακτηριστικά της γηπεδούχου ομάδας, αντίστοιχα χαρακτηριστικά της φιλοξενούμενης ομάδας, καθώς και πληροφορία του τρέχοντος αγώνα μέσω των implied probabilities των στοιχηματικών αγορών. Με αυτόν τον τρόπο, η ιστορική πληροφορία των ομάδων δεν δίνεται στο μοντέλο ως ακολουθία αγώνων, αλλά συμπυκνώνεται σε προαγωνιστικά χαρακτηριστικά σταθερού μήκους.

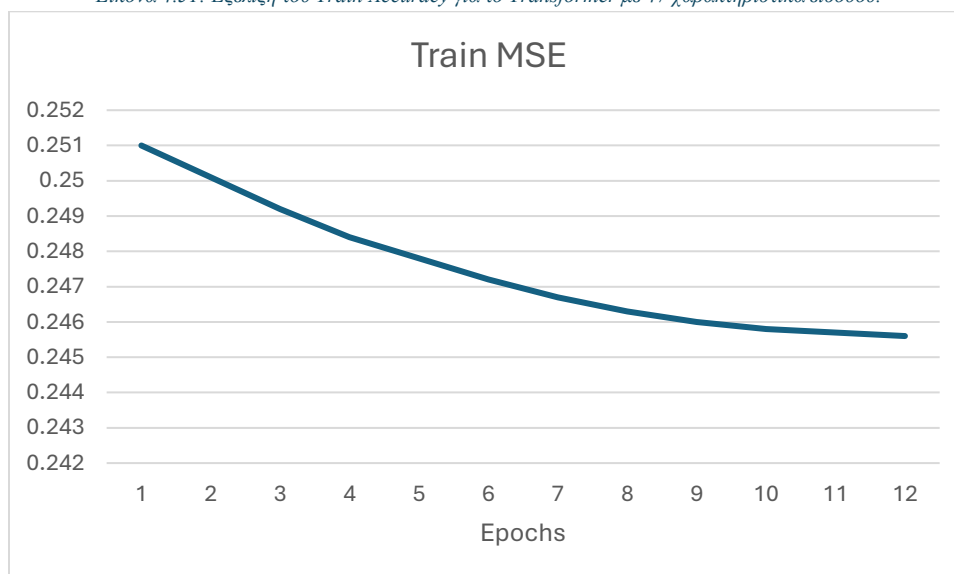
Συνοπτικά, τα βασικά χαρακτηριστικά του μοντέλου είναι τα εξής:

- Διαστάσεις εισόδου: 47
- Παράθυρο υπολογισμού rolling χαρακτηριστικών (k): 10 προηγούμενοι αγώνες
- Batch size: 64
- Εποχές εκπαίδευσης: 12
- Patience: 8
- Διάσταση αναπαράστασης (d_{model}): 64
- Αριθμός κεφαλών προσοχής: 2
- Αριθμός επιπέδων Transformer: 1
- Dropout: 0.2
- Τύπος pooling: cls

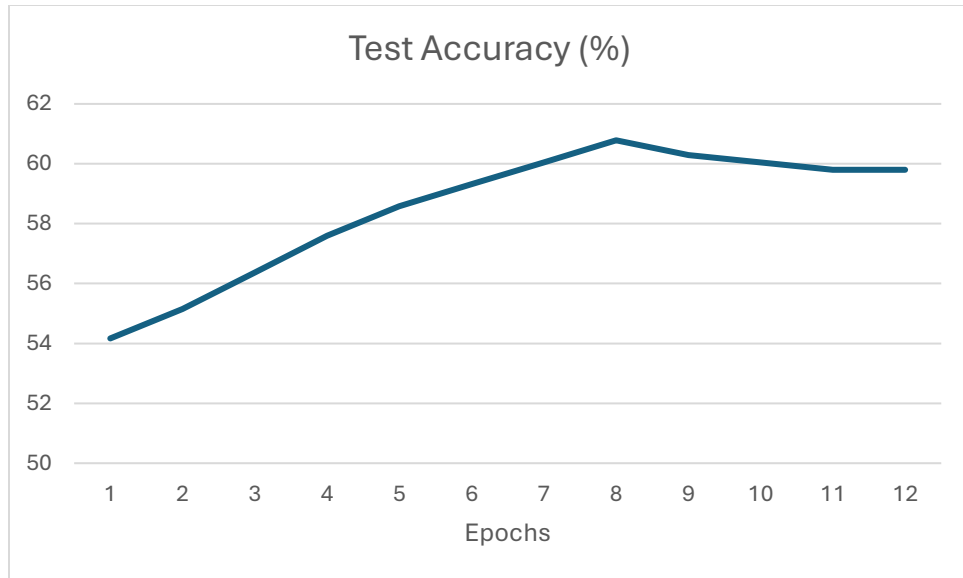
- Ρυθμός μάθησης: 0.001
- Decision threshold: 0.5
- Threshold strategy: fixed
- Δεδομένα εκπαίδευσης: 2005/06 – 2023/24
- Δεδομένα επαλήθευσης: 2024/25



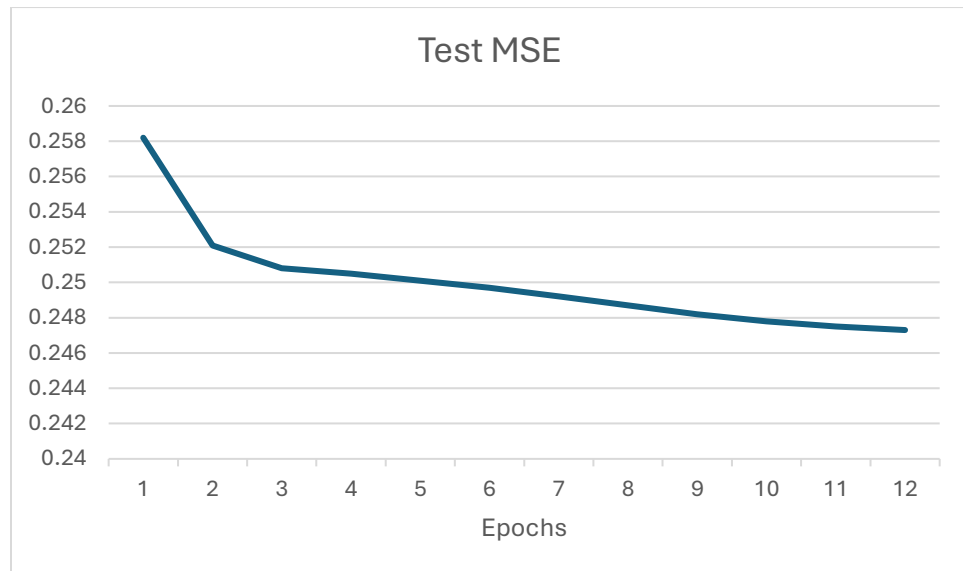
Εικόνα 4.51: Εξέλιξη του Train Accuracy για το Transformer με 47 χαρακτηριστικά εισόδου.



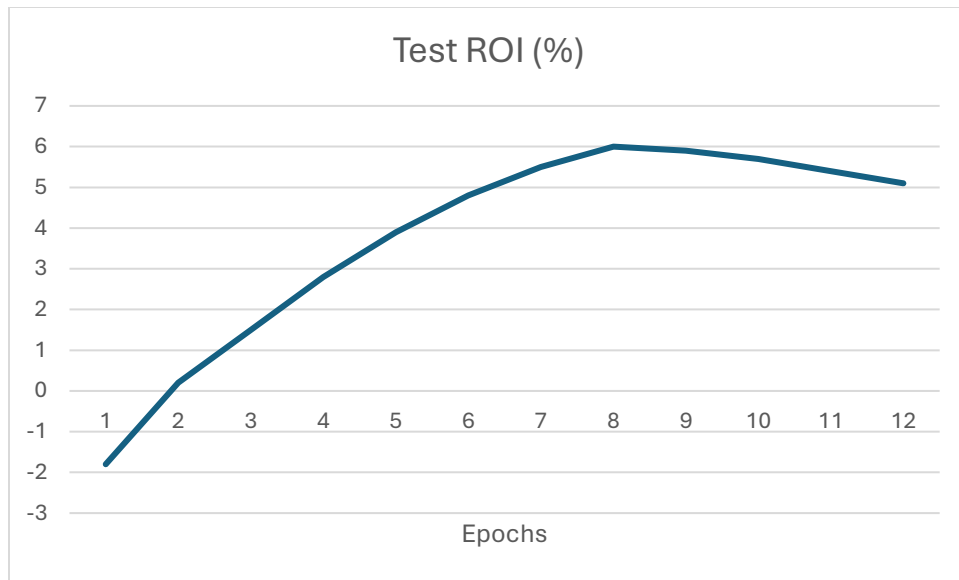
Εικόνα 4.52: Εξέλιξη του Train MSE για το Transformer με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.53: Εξέλιξη του Test Accuracy για το Transformer με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.54: Εξέλιξη του Test MSE για το Transformer με 47 χαρακτηριστικά εισόδου.



Εικόνα 4.55: Εξέλιξη του Test ROI για το Transformer με 47 χαρακτηριστικά εισόδου.

Όπως φαίνεται στις Εικόνες 4.51–4.55, από τα αποτελέσματα του Transformer με 47 χαρακτηριστικά εισόδου προκύπτει η καλύτερη συνολική επίδοση μεταξύ των τριών εκδοχών του μοντέλου. Η Test Accuracy αυξάνεται σταθερά κατά τις πρώτες εποχές και κορυφώνεται στο 60.78%, όπως φαίνεται στην Εικόνα 4.53, ενώ παράλληλα το ROI γίνεται σαφώς θετικό και φτάνει στο 6.07%, όπως φαίνεται στην Εικόνα 4.55. Η συμπεριφορά αυτή δείχνει ότι η πλουσιότερη αναπαράσταση εισόδου επιτρέπει στο μοντέλο να αξιοποιήσει αποτελεσματικότερα τόσο τη στατιστική πληροφορία των ομάδων όσο και την προαγωνιστική πληροφορία της αγοράς.

Παράλληλα, το Test MSE ακολουθεί καθοδική πορεία σχεδόν σε όλη τη διάρκεια της εκπαίδευσης, όπως φαίνεται στην Εικόνα 4.54, γεγονός που υποδηλώνει σταδιακή βελτίωση της ποιότητας των προβλεπόμενων πιθανοτήτων. Μετά περίπου την 8η εποχή, η accuracy και το ROI εμφανίζουν μικρή κάμψη, χωρίς όμως να αναιρείται η συνολικά πολύ καλή απόδοση του μοντέλου. Συνεπώς, η εκδοχή των 47 χαρακτηριστικών αναδεικνύεται ως η πιο αποτελεσματική Transformer-based προσέγγιση της εργασίας, τόσο σε επίπεδο προβλεπτικής ακρίβειας όσο και σε επίπεδο πρακτικής στοιχηματικής απόδοσης.

Συνολικά, όπως φαίνεται στις Εικόνες 4.41–4.55, τα αποτελέσματα των Transformer πειραμάτων δείχνουν ότι η ποιότητα και η πληρότητα της εισόδου επηρεάζουν καθοριστικά την απόδοση του μοντέλου. Με 26 χαρακτηριστικά εισόδου, το μοντέλο παρουσίασε περιορισμένη προβλεπτική ικανότητα, με Best Test Accuracy 53.36%, όπως φαίνεται στην Εικόνα 4.43, και αρνητικό ROI -1.46%, όπως φαίνεται στην Εικόνα 4.45. Το αποτέλεσμα αυτό δείχνει ότι η βασική μορφή εισόδου δεν περιέχει αρκετή πληροφορία ώστε ο Transformer να αποκτήσει ουσιαστικό πλεονέκτημα στο πρόβλημα Over/Under 2.5.

Η αύξηση των χαρακτηριστικών σε 34 οδήγησε σε σαφή βελτίωση. Η καλύτερη τιμή Test Accuracy έφτασε το 55.14%, όπως φαίνεται στην Εικόνα 4.48, ενώ το ROI έγινε θετικό και διαμορφώθηκε στο 1.02%, όπως φαίνεται στην Εικόνα 4.50. Το εύρημα αυτό δείχνει ότι η προσθήκη επιπλέον προαγωνιστικών χαρακτηριστικών βελτίωσε όχι μόνο τη στατιστική απόδοση του μοντέλου, αλλά και την πρακτική του συμπεριφορά στη στοιχηματική προσομοίωση.

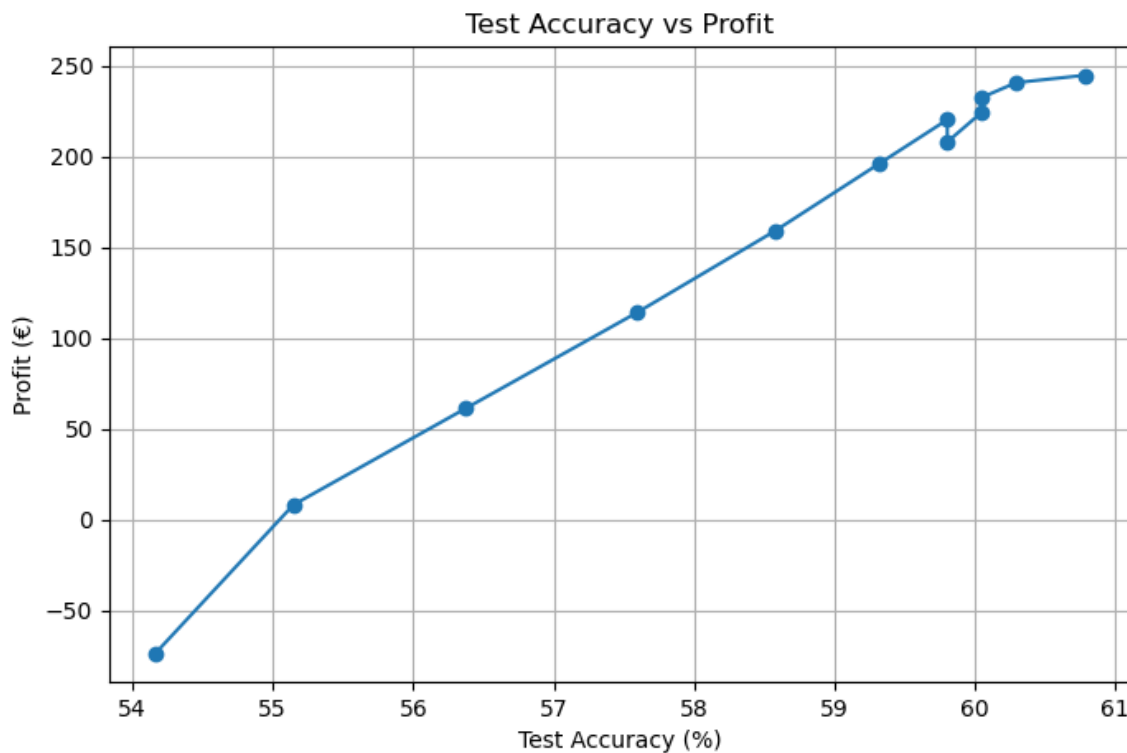
Η ισχυρότερη επίδοση καταγράφηκε στην εκδοχή με 47 χαρακτηριστικά εισόδου. Σε αυτή την περίπτωση, η ενιαία προαγωνιστική αναπαράσταση περιλαμβάνει στατιστικά χαρακτηριστικά για τη γηπεδούχο και τη φιλοξενούμενη ομάδα, καθώς και πληροφορία του τρέχοντος αγώνα μέσω των implied probabilities των στοιχηματικών αγορών. Το μοντέλο πέτυχε Best Test Accuracy 60.78%, όπως φαίνεται στην Εικόνα 4.53, και ROI 6.07%, όπως φαίνεται στην Εικόνα 4.55, αποτελώντας την καλύτερη συνολική επίδοση της παρούσας εργασίας.

Η σύγκριση των τριών εκδοχών δείχνει ότι ο Transformer αξιοποιεί πιο αποτελεσματικά την πληροφορία όταν αυτή είναι οργανωμένη σε πλουσιότερη και πιο ολοκληρωμένη προαγωνιστική αναπαράσταση. Η βελτίωση από τα 26 στα 34 και τελικά στα 47 χαρακτηριστικά δεν είναι μόνο αριθμητική, αλλά αφορά και την ποιότητα της πληροφορίας που παρέχεται στο μοντέλο. Ιδιαίτερα η ενσωμάτωση των πιθανοτήτων της αγοράς και των χαρακτηριστικών του τρέχοντος αγώνα φαίνεται να ενισχύει σημαντικά την προβλεπτική ικανότητα.

Επιπλέον, η συμπεριφορά του Test MSE στις Transformer εκδοχές δείχνει πιο σταθερή εικόνα σε σχέση με τα RBF πειράματα. Ειδικά στην εκδοχή των 47 χαρακτηριστικών, το Test MSE ακολουθεί καθοδική πορεία κατά τη διάρκεια της εκπαίδευσης, όπως φαίνεται στην Εικόνα 4.54, γεγονός που υποδηλώνει ότι οι προβλεπόμενες πιθανότητες βελτιώνονται σταδιακά και δεν αυξάνεται απλώς η τελική ακρίβεια ταξινόμησης.

Συνεπώς, τα αποτελέσματα των Transformer πειραμάτων αναδεικνύουν ότι η αρχιτεκτονική Transformer μπορεί να αποδώσει αποτελεσματικά στο πρόβλημα Over/Under 2.5, όταν τροφοδοτείται με επαρκώς πλούσια και χρονικά ορθή προαγωνιστική πληροφορία. Το σημαντικότερο συμπέρασμα δεν είναι απλώς ότι ο Transformer υπερτερεί των απλούστερων μοντέλων, αλλά ότι η επιτυχία του εξαρτάται άμεσα από την ποιότητα της αναπαράστασης εισόδου.

4.4 Σχέση Ακρίβειας και Στοιχηματικού Κέρδους



Εικόνα 4.56: Σχέση μεταξύ test accuracy και συνολικού στοιχηματικού κέρδους στα πειράματα της εργασίας.

Όπως φαίνεται στην Εικόνα 4.56, η ανάλυση του διαγράμματος αναδεικνύει μια σαφή θετική σχέση μεταξύ της ακρίβειας στα δεδομένα ελέγχου και του συνολικού στοιχηματικού κέρδους. Όσο αυξάνεται το Test Accuracy των μοντέλων, τόσο βελτιώνεται και το τελικό οικονομικό αποτέλεσμα της στρατηγικής πονταρίσματος. Στις χαμηλότερες τιμές ακρίβειας το profit παραμένει αρνητικό ή οριακά θετικό, ενώ σε υψηλότερες τιμές ακρίβειας παρατηρείται σημαντική αύξηση του συνολικού κέρδους.

Η συμπεριφορά αυτή είναι αναμενόμενη, καθώς στην αγορά Over/Under 2.5 κάθε σωστή πρόβλεψη επιστρέφει ποσό ανάλογο με την αντίστοιχη απόδοση, ενώ κάθε λανθασμένη πρόβλεψη οδηγεί σε απώλεια του πονταρίσματος. Επομένως, όταν η ακρίβεια του μοντέλου ξεπερνά το σημείο ισορροπίας που απαιτείται από τις διαθέσιμες αποδόσεις, οι σωστές προβλέψεις αρχίζουν να υπερκαλύπτουν τις απώλειες και το συνολικό αποτέλεσμα γίνεται θετικό.

Στο συγκεκριμένο διάγραμμα φαίνεται ότι η αύξηση της ακρίβειας από περίπου 54% σε πάνω από 60% συνοδεύεται από σημαντική μεταβολή του profit, από αρνητικές τιμές σε κέρδος άνω των 200€. Το εύρημα αυτό δείχνει ότι ακόμη και φαινομενικά μικρές βελτιώσεις στην ακρίβεια μπορούν να έχουν μεγάλη πρακτική σημασία όταν μεταφράζονται σε στοιχηματικές αποφάσεις.

Παράλληλα, το διάγραμμα δείχνει ότι η ακρίβεια δεν πρέπει να αξιολογείται απομονωμένα. Η τελική κερδοφορία εξαρτάται όχι μόνο από το ποσοστό σωστών προβλέψεων, αλλά και από τις αποδόσεις των αγώνων στους οποίους το μοντέλο προβλέπει σωστά ή λανθασμένα. Για τον λόγο αυτό, η παρούσα εργασία αξιολογεί τα μοντέλα τόσο με κλασικές μετρικές ταξινόμησης όσο και με χρηματοοικονομικούς δείκτες, όπως profit και ROI.

Συνολικά, η σχέση Test Accuracy και Profit επιβεβαιώνει ότι η βελτίωση της προβλεπτικής απόδοσης έχει άμεση πρακτική αξία, όταν είναι αρκετά ισχυρή ώστε να ξεπερνά το περιθώριο των bookmakers. Στο πρόβλημα Over/Under 2.5, μικρές διαφορές στην ακρίβεια μπορούν να καθορίσουν αν μια στρατηγική θα είναι ζημιογόνα ή κερδοφόρα.

4.5 Στοιχηματικό κέρδος με πραγματικές αποδόσεις

Μετά την ολοκλήρωση της εκπαίδευσης και αξιολόγησης των μοντέλων στα ιστορικά δεδομένα, πραγματοποιήθηκε ένας επιπλέον έλεγχος σε πραγματικές συνθήκες στοιχηματισμού. Για τον σκοπό αυτό χρησιμοποιήθηκαν οι αγώνες της τελευταίας αγωνιστικής της Championship για τη σεζόν 2024/25. Κατά τη στιγμή παραγωγής των προβλέψεων, οι αγώνες δεν είχαν ακόμη διεξαχθεί, επομένως οι αποδόσεις που χρησιμοποιήθηκαν αντιστοιχούσαν σε πραγματικές pre-match αποδόσεις της αγοράς.

Για κάθε αγώνα θεωρήθηκε σταθερό ποντάρισμα ύψους 1€. Όταν η πρόβλεψη του μοντέλου ήταν σωστή, η επιστροφή υπολογιζόταν με βάση την αντίστοιχη απόδοση της αγοράς Over/Under 2.5. Σε περίπτωση λανθασμένης πρόβλεψης, η επιστροφή θεωρούνταν μηδενική και το ποντάρισμα χανόταν.

Ένα πιθανό σενάριο προβλέψεων για την τελευταία αγωνιστική παρουσιάζεται στην Εικόνα 4.57. Το σενάριο αυτό δεν χρησιμοποιείται ως κύριο μέτρο αξιολόγησης του μοντέλου, καθώς αφορά περιορισμένο αριθμό αγώνων, αλλά λειτουργεί ως πρακτική ένδειξη του τρόπου με τον οποίο οι προβλέψεις μπορούν να μεταφραστούν σε στοιχηματικές αποφάσεις με πραγματικές αποδόσεις.

Αγώνας	Απόδοση Under	Απόδοση Over	Πρόβλεψη Μοντέλου	Επιστροφή
Bristol City – Preston	1.88	1.98	Over 2.5	1.98
Burnley – Millwall	1.80	2.00	Over 2.5	2.00
Coventry – Middlesbrough	2.38	1.57	Under 2.5	2.38
Derby – Stoke City	1.53	2.50	Under 2.5	1.53
Norwich – Cardiff	2.38	1.57	Over 2.5	1.57
Plymouth – Leeds	2.63	1.50	Over 2.5	1.50

Αγώνας	Απόδοση Under	Απόδοση Over	Πρόβλεψη Μοντέλου	Επιστροφή
Portsmouth – Hull City	1.90	1.95	Under 2.5	1.90
Sheffield Utd – Blackburn	1.83	2.03	Over 2.5	0
Sunderland – QPR	1.93	1.93	Under 2.5	1.93
Swansea – Oxford Utd	1.90	1.95	Under 2.5	0
Watford – Sheffield Wed	2.20	1.67	Over 2.5	0
West Brom – Luton	1.80	2.05	Under 2.5	0

Εικόνα 4.57: Ενδεικτικό σενάριο στοιχηματικών προβλέψεων για την 46η αγωνιστική της Championship (σεζόν 2024/25) με χρήση πραγματικών pre-match αποδόσεων.

Με βάση το συγκεκριμένο σενάριο, το συνολικό ποσό στοιχηματισμού ανέρχεται σε 12€, καθώς πραγματοποιείται ποντάρισμα 1€ σε κάθε έναν από τους 12 αγώνες. Οι συνολικές επιστροφές ανέρχονται σε 14.79€, οδηγώντας σε καθαρό κέρδος 2.79€. Επομένως, το ROI της συγκεκριμένης προσομοίωσης υπολογίζεται σε 23.25%.

Το αποτέλεσμα αυτό δείχνει ότι, σε ένα μικρό και συγκεκριμένο σύνολο πραγματικών αγώνων, οι προβλέψεις του μοντέλου μπορούν να οδηγήσουν σε θετική στοιχηματική απόδοση. Ωστόσο, το συμπέρασμα αυτό πρέπει να ερμηνευθεί με προσοχή, καθώς η τελευταία αγωνιστική περιλαμβάνει μικρό αριθμό παρατηρήσεων και δεν αρκεί από μόνη της για γενίκευση. Παρόλα αυτά, η προσομοίωση έχει πρακτική αξία, επειδή δείχνει πώς το μοντέλο μπορεί να εφαρμοστεί σε πραγματικές pre-match συνθήκες και πώς οι προβλέψεις του συνδέονται με οικονομικούς δείκτες όπως profit και ROI.

Κεφάλαιο 5

Συμπεράσματα και Μελλοντική Εργασία

5.1 Συμπεράσματα

5.2 Μελλοντική Εργασία

5.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία ασχολήθηκε με την πρόβλεψη του αν ένας ποδοσφαιρικός αγώνας της αγγλικής Championship θα λήξει με Over ή Under 2.5 συνολικά γκολ. Το πρόβλημα διατυπώθηκε ως δυαδική ταξινόμηση, όπου η κλάση Over αντιστοιχεί σε αγώνες με περισσότερα από 2.5 γκολ, ενώ η κλάση Under αντιστοιχεί σε αγώνες με λιγότερα ή ίσα με 2.5 γκολ. Η επιλογή του συγκεκριμένου προβλήματος έχει ιδιαίτερο ενδιαφέρον, καθώς η αγορά Over/Under 2.5 είναι μία από τις πιο διαδεδομένες στοιχηματικές αγορές, αλλά ταυτόχρονα παρουσιάζει υψηλή αβεβαιότητα και είναι δύσκολη στην πρόβλεψη.

Κεντρικός στόχος της εργασίας ήταν η ανάπτυξη, εκπαίδευση και αξιολόγηση μοντέλων μηχανικής μάθησης και νευρωνικών δικτύων για το συγκεκριμένο πρόβλημα.

Χρησιμοποιήθηκαν μοντέλα MLP, RBF και Transformer, με διαφορετικές μορφές εισόδου 26, 34 και 47 χαρακτηριστικών. Με αυτόν τον τρόπο, η εργασία δεν περιορίστηκε στην αξιολόγηση ενός μόνο μοντέλου, αλλά εξέτασε πώς επηρεάζεται η απόδοση από την αρχιτεκτονική και κυρίως από την ποιότητα της προαγωνιστικής αναπαράστασης.

Ένα από τα σημαντικότερα συμπεράσματα της εργασίας είναι ότι τα χαρακτηριστικά εισόδου επηρεάζουν καθοριστικά την τελική απόδοση. Η μετάβαση από τις απλούστερες αναπαραστάσεις των 26 χαρακτηριστικών στις πιο πλήρεις αναπαραστάσεις των 47 χαρακτηριστικών οδήγησε σε σαφή βελτίωση στα περισσότερα μοντέλα. Στα MLP, η καλύτερη επίδοση καταγράφηκε στην εκδοχή των 47 χαρακτηριστικών, με Test Accuracy 57.84%, όπως φαίνεται στην Εικόνα 4.18, και θετικό ROI 4.43%, όπως φαίνεται στην Εικόνα 4.20. Αντίστοιχα, στα RBF, η καλύτερη συμπεριφορά εμφανίστηκε επίσης στην εκδοχή των 47 χαρακτηριστικών, όπου το ROI έγινε θετικό και έφτασε περίπου στο 3.86%, όπως φαίνεται στην Εικόνα 4.40. Το γεγονός αυτό δείχνει ότι η αύξηση της πληροφορίας εισόδου δεν είναι απλώς τεχνική λεπτομέρεια, αλλά βασικός παράγοντας της συνολικής απόδοσης.

Το ισχυρότερο αποτέλεσμα της εργασίας προέκυψε από το Transformer μοντέλο με 47 χαρακτηριστικά εισόδου. Η συγκεκριμένη εκδοχή πέτυχε Best Test Accuracy 60.78%, όπως φαίνεται στην Εικόνα 4.53, και ROI 6.07%, όπως φαίνεται στην Εικόνα 4.55, αποτελώντας την καλύτερη συνολική επίδοση μεταξύ όλων των πειραμάτων. Η επίδοση αυτή δείχνει ότι ο Transformer μπορεί να αξιοποιήσει αποτελεσματικά μια πλούσια προαγωνιστική αναπαράσταση, η οποία συνδυάζει στατιστικά χαρακτηριστικά των δύο ομάδων και πληροφορία

της αγοράς μέσω implied probabilities. Το σημαντικότερο συμπέρασμα, επομένως, δεν είναι απλώς ότι ο Transformer απέδωσε καλύτερα, αλλά ότι απέδωσε καλύτερα όταν τροφοδοτήθηκε με πιο ολοκληρωμένα και κατάλληλα οργανωμένα χαρακτηριστικά.

Από τη σύγκριση των Transformer εκδοχών φαίνεται καθαρά η επίδραση της εισόδου. Με 26 χαρακτηριστικά, το μοντέλο πέτυχε Test Accuracy 53.36%, όπως φαίνεται στην Εικόνα 4.43, και ROI -1.46%, όπως φαίνεται στην Εικόνα 4.45. Με 34 χαρακτηριστικά, η ακρίβεια αυξήθηκε σε 55.14%, όπως φαίνεται στην Εικόνα 4.48, και το ROI έγινε οριακά θετικό, φτάνοντας το 1.02%, όπως φαίνεται στην Εικόνα 4.50. Τέλος, με 47 χαρακτηριστικά, η ακρίβεια αυξήθηκε στο 60.78%, όπως φαίνεται στην Εικόνα 4.53, και το ROI στο 6.07%, όπως φαίνεται στην Εικόνα 4.55. Η σταδιακή αυτή βελτίωση δείχνει ότι η ποιότητα και η πληρότητα της προαγωνιστικής πληροφορίας είχαν άμεση επίδραση τόσο στην προβλεπτική όσο και στη χρηματοοικονομική απόδοση.

Ιδιαίτερα σημαντική ήταν και η σχέση μεταξύ ακρίβειας και στοιχηματικού κέρδους. Από την ανάλυση των αποτελεσμάτων προέκυψε ότι όσο αυξάνεται το Test Accuracy, τόσο βελτιώνεται και το συνολικό profit, όπως φαίνεται στην Εικόνα 4.56. Στα χαμηλότερα επίπεδα ακρίβειας, περίπου κοντά στο 54%, το οικονομικό αποτέλεσμα ήταν αρνητικό. Αντίθετα, όταν η ακρίβεια ξεπέρασε το 60%, το profit αυξήθηκε σημαντικά και έφτασε σε επίπεδα άνω των 200€. Η παρατήρηση αυτή δείχνει ότι ακόμη και μικρές ποσοστιαίες βελτιώσεις στην ακρίβεια μπορούν να έχουν μεγάλη πρακτική σημασία σε στοιχηματικό περιβάλλον, ιδιαίτερα όταν το μοντέλο ξεπερνά το σημείο ισορροπίας που απαιτείται από τις αποδόσεις.

Παράλληλα, η εργασία ανέδειξε ότι η ακρίβεια από μόνη της δεν είναι επαρκής για την αξιολόγηση ενός μοντέλου με στοιχηματική εφαρμογή. Ένα μοντέλο μπορεί να παρουσιάζει σχετικά καλή ταξινομητική απόδοση, αλλά να μην είναι κερδοφόρο, εάν οι σωστές προβλέψεις του δεν συνοδεύονται από κατάλληλες αποδόσεις. Για τον λόγο αυτό, η αξιολόγηση με profit και ROI ήταν απαραίτητη. Η συνδυασμένη ανάλυση στατιστικών και χρηματοοικονομικών δεικτών προσφέρει πιο ρεαλιστική εικόνα της πρακτικής αξίας των μοντέλων.

Η προσομοίωση με πραγματικές pre-match αποδόσεις στην τελευταία αγωνιστική της Championship 2024/25 έδωσε επίσης ένα πρακτικό παράδειγμα εφαρμογής του μοντέλου, όπως παρουσιάζεται στην Εικόνα 4.57. Σε 12 αγώνες, με σταθερό ποντάρισμα 1€ ανά αγώνα, οι συνολικές επιστροφές έφτασαν τα 14.79€, με καθαρό κέρδος 2.79€ και ROI 23.25%. Παρότι το

δείγμα αυτό είναι μικρό και δεν μπορεί να χρησιμοποιηθεί για γενίκευση, δείχνει πώς οι προβλέψεις του μοντέλου μπορούν να μετατραπούν σε πραγματικές στοιχηματικές αποφάσεις και να αξιολογηθούν με πρακτικούς οικονομικούς δείκτες.

Ένα ακόμη βασικό μεθοδολογικό συμπέρασμα αφορά την αποφυγή διαρροής πληροφορίας από το μέλλον. Όλα τα δείγματα κατασκευάστηκαν με βάση πληροφορία διαθέσιμη πριν από την έναρξη του αγώνα, ενώ η αξιολόγηση πραγματοποιήθηκε με χρονολογική συνέπεια, χρησιμοποιώντας μεταγενέστερη σεζόν ως σύνολο επαλήθευσης. Η επιλογή αυτή είναι ιδιαίτερα σημαντική σε προβλήματα ποδοσφαιρικής πρόβλεψης, καθώς οποιαδήποτε χρήση μελλοντικής πληροφορίας θα μπορούσε να οδηγήσει σε τεχνητά υψηλή απόδοση. Επομένως, η αυστηρή προαγωνιστική και χρονολογικά ορθή διαδικασία αποτελεί βασικό πλεονέκτημα της παρούσας εργασίας.

Συνολικά, τα αποτελέσματα δείχνουν ότι η πρόβλεψη Over/Under 2.5 παραμένει ένα ιδιαίτερα απαιτητικό πρόβλημα. Οι στοιχηματικές αγορές ενσωματώνουν ήδη μεγάλο μέρος της διαθέσιμης πληροφορίας στις αποδόσεις, γεγονός που περιορίζει το περιθώριο εύρεσης σταθερού πλεονεκτήματος. Παρόλα αυτά, η εργασία έδειξε ότι με σωστή αναπαράσταση εισόδου, προσεκτική προεπεξεργασία και κατάλληλο μοντέλο, είναι δυνατό να επιτευχθεί προβλεπτική απόδοση υψηλότερη από τις απλούστερες προσεγγίσεις και, σε ορισμένες περιπτώσεις, θετικό ROI.

5.2 Μελλοντική Εργασία

Ως προς τη μελλοντική εργασία, μία σημαντική κατεύθυνση είναι η χρήση τεχνικών επεξηγήσιμης τεχνητής νοημοσύνης, όπως οι SHAP values. Η μέθοδος αυτή μπορεί να βοηθήσει στην κατανόηση του ποια χαρακτηριστικά επηρεάζουν περισσότερο την πρόβλεψη του μοντέλου. Αυτό είναι ιδιαίτερα σημαντικό στην παρούσα εργασία, καθώς τα αποτελέσματα έδειξαν ότι η επιλογή και η οργάνωση των χαρακτηριστικών εισόδου παίζουν καθοριστικό ρόλο. Μέσω SHAP ανάλυσης θα μπορούσε να εξεταστεί, για παράδειγμα, αν οι bookmaker probabilities, τα πρόσφατα Over/Under rates, οι δείκτες φόρμας ή τα venue-aware χαρακτηριστικά έχουν τη μεγαλύτερη συμβολή στις τελικές προβλέψεις.

Μια δεύτερη σημαντική κατεύθυνση αφορά τη δοκιμή του μοντέλου σε περισσότερα πρωταθλήματα. Η παρούσα εργασία επικεντρώθηκε στην αγγλική Championship, ώστε να υπάρχει ομοιογένεια και καθαρό πειραματικό πλαίσιο. Ωστόσο, διαφορετικά πρωταθλήματα έχουν διαφορετικό αγωνιστικό ρυθμό, διαφορετικές σκοραριστικές τάσεις, διαφορετικό βαθμό ισορροπίας μεταξύ ομάδων και διαφορετική αποδοτικότητα στις στοιχηματικές αγορές. Επομένως, ένα μοντέλο με τα ίδια χαρακτηριστικά δεν είναι απαραίτητα το βέλτιστο για όλα τα πρωταθλήματα. Μελλοντικά, θα μπορούσε να εξεταστεί πόσο πρέπει να προσαρμόζεται η αναπαράσταση εισόδου και η αρχιτεκτονική του μοντέλου ανάλογα με το πρωτάθλημα.

Μια τρίτη πιθανή επέκταση είναι η ενσωμάτωση επιπλέον ποδοσφαιρικών χαρακτηριστικών υψηλότερης ποιότητας. Ένα χαρακτηριστικό παράδειγμα είναι το Expected Goals (xG), το οποίο αποτυπώνει την ποιότητα των ευκαιριών που δημιουργεί ή δέχεται μια ομάδα. Σε αντίθεση με τα απλά γκολ, τα οποία μπορεί να επηρεάζονται έντονα από τύχη ή αποτελεσματικότητα, το xG προσφέρει πιο σταθερή εικόνα της επιθετικής και αμυντικής απόδοσης. Η χρήση rolling xG υπέρ και κατά, xG difference και xG per shot θα μπορούσε να βελτιώσει περαιτέρω την ποιότητα της προαγωνιστικής αναπαράστασης.

Επιπλέον, θα μπορούσε να εξεταστεί η χρήση πιο στοχευμένων στοιχηματικών στρατηγικών. Στην παρούσα εργασία, η προσομοίωση βασίστηκε κυρίως σε ποντάρισμα σε όλους τους αγώνες. Μελλοντικά, θα μπορούσε να εφαρμοστεί value betting, όπου το μοντέλο θα ποντάρει μόνο όταν η εκτιμώμενη πιθανότητα του Over ή Under είναι μεγαλύτερη από την implied probability του bookmaker κατά ένα συγκεκριμένο περιθώριο. Με αυτόν τον τρόπο, το μοντέλο δεν θα αξιολογείται μόνο ως ταξινομητής, αλλά και ως εργαλείο εντοπισμού ευκαιριών με θετική αναμενόμενη αξία.

Μια ακόμη κατεύθυνση αφορά τη βελτίωση της βαθμονόμησης των πιθανοτήτων. Για στοιχηματικές εφαρμογές, δεν αρκεί το μοντέλο να προβλέπει σωστά την κλάση· είναι σημαντικό οι πιθανότητες που παράγει να αντανakλούν όσο το δυνατόν καλύτερα τις πραγματικές πιθανότητες εμφάνισης Over ή Under. Τεχνικές όπως Platt scaling, isotonic regression ή temperature scaling θα μπορούσαν να χρησιμοποιηθούν ώστε οι πιθανότητες του μοντέλου να γίνουν πιο αξιόπιστες και κατάλληλες για value betting στρατηγικές.

Τέλος, μελλοντικά θα μπορούσαν να εξεταστούν διαφορετικές παραλλαγές της Transformer αρχιτεκτονικής, μεγαλύτερα ιστορικά παράθυρα, διαφορετικές στρατηγικές pooling και συνδυασμός με άλλα μοντέλα. Παρότι η καλύτερη επίδοση της παρούσας εργασίας προήλθε από Transformer με 47 χαρακτηριστικά εισόδου, παραμένει ανοιχτό το ερώτημα αν διαφορετικές αρχιτεκτονικές ή καλύτερα προσαρμοσμένες αναπαραστάσεις μπορούν να βελτιώσουν περαιτέρω την απόδοση.

Συμπερασματικά, η εργασία έδειξε ότι η χρήση Transformer-based μοντέλων για την πρόβλεψη Over/Under 2.5 είναι τεχνικά εφικτή και μεθοδολογικά ενδιαφέρουσα. Το καλύτερο αποτέλεσμα, με Test Accuracy 60.78% και ROI 6.07%, δείχνει ότι υπάρχει αξιοποιήσιμο προβλεπτικό σήμα στα δεδομένα. Ωστόσο, η σταθερή αξιοποίηση αυτού του σήματος απαιτεί προσεκτική επιλογή χαρακτηριστικών, αυστηρή αποφυγή leakage, σωστή βαθμονόμηση πιθανοτήτων και πιο στοχευμένες στρατηγικές στοιχηματικής εφαρμογής.

References

- [1] R. Baboota and H. Kaur, “Predictive analysis and modelling football results using machine learning approach for English Premier League,” *International Journal of Forecasting*, vol. 35, no. 2, pp. 741–755, 2019. doi: 10.1016/j.ijforecast.2018.01.003.
- [2] D. Berrar, P. Lopes, J. Davis, and W. Dubitzky, “The 2017 Soccer Prediction Challenge,” *Machine Learning*, vol. 108, pp. 49–75, 2019.
- [3] C. M. Bishop and H. Bishop, *Deep Learning: Foundations and Concepts*. Cham, Switzerland: Springer, 2024. doi: 10.1007/978-3-031-45468-4.
- [4] R. P. Bunker and F. Thabtah, “A machine learning framework for sport result prediction,” *Applied Computing and Informatics*, vol. 15, no. 1, pp. 27–33, 2019. doi: 10.1016/j.aci.2017.09.005.
- [5] A. C. Constantinou, “Dolores: A model that predicts football match outcomes from all over the world,” *Machine Learning*, vol. 108, pp. 49–75, 2019. doi: 10.1007/s10994-018-5703-7.
- [6] A. C. Constantinou and N. E. Fenton, “Determining the level of ability of football teams by dynamic ratings based on the relative discrepancies in scores between adversaries,” *Journal of Quantitative Analysis in Sports*, vol. 9, no. 1, pp. 37–50, 2013. doi: 10.1515/jqas-2012-0036.
- [7] I. B. da Costa, L. B. Marinho, and C. E. S. Pires, “Forecasting football results and exploiting betting markets: The case of ‘both teams to score’,” *International Journal of Forecasting*, vol. 38, no. 3, pp. 895–909, 2022. doi: 10.1016/j.ijforecast.2021.06.008.
- [8] M. J. Dixon and S. G. Coles, “Modelling association football scores and inefficiencies in the football betting market,” *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, vol. 46, no. 2, pp. 265–280, 1997. doi: 10.1111/1467-9876.00065.
- [9] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in *International Conference on Learning Representations*, 2021.
- [10] Football-Data.co.uk, “Football results, statistics and soccer betting odds data.” [Online]. Available: <https://www.football-data.co.uk/>. Accessed: May 22, 2026.
- [11] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.
- [12] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, “On calibration of modern neural networks,” in *Proceedings of the 34th International Conference on Machine Learning*, vol. 70,

pp. 1321–1330, 2017.

[13] B. Holmes and I. G. McHale, “Forecasting football match results using a player rating based model,” *International Journal of Forecasting*, vol. 40, no. 1, pp. 302–312, 2024. doi:

10.1016/j.ijforecast.2023.03.002.

[14] O. Hubáček, G. Šourek, and F. Železný, “Learning to predict soccer results from relational data with gradient boosted trees,” *Machine Learning*, vol. 108, pp. 29–47, 2019. doi:

10.1007/s10994-018-5704-6.

[15] L. M. Hvattum and H. Arntzen, “Using ELO ratings for match result prediction in association football,” *International Journal of Forecasting*, vol. 26, no. 3, pp. 460–470, 2010. doi:

10.1016/j.ijforecast.2009.10.002.

[16] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations*, 2015.

[17] M. J. Maher, “Modelling association football scores,” *Statistica Neerlandica*, vol. 36, no. 3, pp. 109–118, 1982. doi: 10.1111/j.1467-9574.1982.tb00782.x.

[18] M.-C. Malamatinos, E. Vrochidou, and G. A. Papakostas, “On predicting soccer outcomes in the Greek League using machine learning,” *Computers*, vol. 11, no. 9, article 133, 2022. doi:

10.3390/computers11090133.

[19] R. G. Martins, A. S. Martins, L. A. Neves, L. V. Lima, E. L. Flores, and M. Z. do Nascimento, “Exploring polynomial classifier to predict match results in football championships,” *Expert Systems with Applications*, vol. 83, pp. 79–93, 2017. doi:

10.1016/j.eswa.2017.04.040.

[20] J. Mead, A. O’Hare, and P. McMenemy, “Expected goals in football: Improving model performance and demonstrating value,” *PLOS ONE*, vol. 18, no. 4, article e0282295, 2023. doi:

10.1371/journal.pone.0282295.

[21] A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[22] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[23] A. Vaswani et al., “Attention is all you need,” in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[24] E. Wheatcroft, “A profitable model for predicting the over/under market in football,”

International Journal of Forecasting, vol. 36, no. 3, pp. 916–932, 2020. doi: 10.1016/j.ijforecast.2019.11.001.

[25] A. Wong, E. Li, H. Le, G. Bhangu, and S. Bhatia, “A predictive analytics framework for forecasting soccer match outcomes using machine learning models,” *Decision Analytics Journal*, vol. 14, article 100537, 2025. doi: 10.1016/j.dajour.2024.100537.

[26] C. Yeung, R. Bunker, R. Umemoto, and K. Fujii, “Evaluating soccer match prediction models: A deep learning approach and feature optimization for gradient-boosted trees,” *Machine Learning*, vol. 113, pp. 7541–7564, 2024. doi: 10.1007/s10994-024-06608-w.

Appendix

src/transformer_single_hybrid/preprocessing_single_hybrid.py

```
from pathlib import Path
```

```
import numpy as np
```

```
import pandas as pd
```

```
RAW_DATA_DIR = Path(__file__).resolve().parent.parent.parent / "data" / "raw"
```

```
DEFAULT_K = 20
```

```
DEFAULT_LEAGUE_FILTER = None
```

```
def _to_float(x):
```

```
    try:
```

```
        if pd.isna(x):
```

```
            return 0.0
```

```
        return float(x)
```

```
    except Exception:
```

```
        return 0.0
```

```
def _imp_prob(odd):
```

```
    odd = _to_float(odd)
```

```
    return (1.0 / odd) if odd > 1e-9 else 0.0
```

```
def _norm3(p1, p2, p3):
```



```

total = p1 + p2 + p3
if total <= 1e-12:
    return 0.0, 0.0, 0.0
return p1 / total, p2 / total, p3 / total

```

```

def _safe_mean(values):
    return float(np.mean(values)) if values else 0.0

```

```

def parse_match_dates(date_series):
    try:
        return pd.to_datetime(date_series, format="mixed", dayfirst=True, errors="coerce")
    except TypeError:
        return pd.to_datetime(date_series, dayfirst=True, errors="coerce")

```

```

def extract_ou_odds(row, available_cols):
    priority_pairs = [
        ("B365>2.5", "B365<2.5"),
        ("BbAv>2.5", "BbAv<2.5"),
        ("Avg>2.5", "Avg<2.5"),
        ("GB>2.5", "GB<2.5"),
    ]

```

```

for over_col, under_col in priority_pairs:
    if over_col in available_cols and under_col in available_cols:
        over_odd = _to_float(row.get(over_col, np.nan))
        under_odd = _to_float(row.get(under_col, np.nan))
        if over_odd > 1.01 and under_odd > 1.01:
            return over_odd, under_odd

```

```
return 0.0, 0.0
```

```
def extract_ou_probs(row, available_cols):  
    priority_pairs = [  
        ("B365>2.5", "B365<2.5"),  
        ("BbAv>2.5", "BbAv<2.5"),  
        ("Avg>2.5", "Avg<2.5"),  
        ("GB>2.5", "GB<2.5"),  
    ]  
  
    ou_over = 0.0  
    ou_under = 0.0  
  
    for over_col, under_col in priority_pairs:  
        if over_col in available_cols and under_col in available_cols:  
            ou_over = row.get(over_col, 0.0)  
            ou_under = row.get(under_col, 0.0)  
            break  
  
    p_over = _imp_prob(ou_over)  
    p_under = _imp_prob(ou_under)  
    total = p_over + p_under  
  
    if total > 0:  
        p_over /= total  
        p_under /= total  
    else:  
        p_over, p_under = 0.0, 0.0
```

```
return p_over, p_under
```

```
def build_current_match_token(row, available_cols, team_is_home=True):
    b365h = row.get("B365H", 0) if "B365H" in available_cols else 0.0
    b365d = row.get("B365D", 0) if "B365D" in available_cols else 0.0
    b365a = row.get("B365A", 0) if "B365A" in available_cols else 0.0

    p_home, p_draw, p_away = _norm3(_imp_prob(b365h), _imp_prob(b365d),
    _imp_prob(b365a))
    p_team_win = p_home if team_is_home else p_away
    p_team_loss = p_away if team_is_home else p_home
    p_over, p_under = extract_ou_probs(row, available_cols)
    odd_over, odd_under = extract_ou_odds(row, available_cols)

    feats = [
        float(team_is_home),
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        0.0,
        p_team_win,
        p_draw,
        p_team_loss,
        p_over,
        p_under,
    ]
```

```
return np.array(feats, dtype=np.float32), odd_over, odd_under
```

```
def build_engineered_features(history, venue, max_history=DEFAULT_K):
```

```
    if not history:
```

```
        return [0.0] * 17
```

```
    venue_matches = [match for match in history if match["is_home"] == venue]
```

```
    if not venue_matches:
```

```
        venue_matches = history
```

```
    last3 = venue_matches[-3:]
```

```
    last6 = history[-6:]
```

```
    avg_gf = _safe_mean([match["gf"] for match in venue_matches])
```

```
    pct_scored = _safe_mean([match["scored"] for match in venue_matches])
```

```
    avg_ga = _safe_mean([match["ga"] for match in venue_matches])
```

```
    pct_conceded = _safe_mean([match["conceded"] for match in venue_matches])
```

```
    pct_over = _safe_mean([match["over25"] for match in venue_matches])
```

```
    avg_gf_l3 = _safe_mean([match["gf"] for match in last3])
```

```
    pct_scored_l3 = _safe_mean([match["scored"] for match in last3])
```

```
    avg_ga_l3 = _safe_mean([match["ga"] for match in last3])
```

```
    pct_conceded_l3 = _safe_mean([match["conceded"] for match in last3])
```

```
    pct_over_l3 = _safe_mean([match["over25"] for match in last3])
```

```
    avg_gf_all = _safe_mean([match["gf"] for match in history])
```

```
    avg_ga_all = _safe_mean([match["ga"] for match in history])
```

```
    pct_over_all = _safe_mean([match["over25"] for match in history])
```

```
    pct_over_l6 = _safe_mean([match["over25"] for match in last6])
```

```
streak_over = 0.0
streak_under = 0.0
```

```
for match in reversed(history):
    if match["over25"] == 1:
        streak_over += 1.0
    else:
        break
```

```
for match in reversed(history):
    if match["over25"] == 0:
        streak_under += 1.0
    else:
        break
```

```
history_len = min(len(history), max_history) / float(max_history)
```

```
return [
    avg_gf,
    pct_scored,
    avg_ga,
    pct_conceded,
    pct_over,
    avg_gf_l3,
    pct_scored_l3,
    avg_ga_l3,
    pct_conceded_l3,
    pct_over_l3,
    avg_gf_all,
    avg_ga_all,
    pct_over_all,
```

```

    pct_over_l6,
    streak_over,
    streak_under,
    history_len,
]

```

```

def build_match_dict(row, team_is_home):

```

```

    hg = int(_to_float(row.get("FTHG", 0)))
    ag = int(_to_float(row.get("FTAG", 0)))

```

```

    gf = hg if team_is_home else ag
    ga = ag if team_is_home else hg
    total = gf + ga

```

```

    return {
        "gf": gf,
        "ga": ga,
        "over25": 1 if total >= 3 else 0,
        "scored": 1 if gf > 0 else 0,
        "conceded": 1 if ga > 0 else 0,
        "is_home": 1 if team_is_home else 0,
    }

```

```

def preprocess_single_hybrid(k=DEFAULT_K, league_filter=DEFAULT_LEAGUE_FILTER):

```

```

    files = sorted(file_name for file_name in RAW_DATA_DIR.iterdir() if
file_name.suffix.lower() == ".csv")
    all_rows = []

```

```

    for file_path in files:

```

```

df = pd.read_csv(file_path, engine="python", encoding="latin1", on_bad_lines="skip")

if league_filter is not None and "Div" in df.columns:
    df = df[df["Div"] == league_filter]

if "Date" in df.columns:
    df["Date"] = parse_match_dates(df["Date"])

df = df.dropna(subset=["HomeTeam", "AwayTeam", "FTHG", "FTAG"])
all_rows.append(df)

df_all = pd.concat(all_rows, ignore_index=True)
if "Date" in df_all.columns:
    df_all = df_all.sort_values("Date", kind="stable").reset_index(drop=True)

teams = sorted(pd.unique(df_all[["HomeTeam", "AwayTeam"]].values.ravel("K")))
team2id = {team: idx for idx, team in enumerate(teams)}
available_cols = set(df_all.columns)
histories = {team: [] for team in teams}

x_rows = []
team_ids = []
opp_ids = []
is_home = []
y_rows = []
odds_over = []
odds_under = []

for _, row in df_all.iterrows():
    home = row["HomeTeam"]
    away = row["AwayTeam"]

```

```

if len(histories[home]) >= k and len(histories[away]) >= k:
    home_features = build_engineered_features(histories[home][-k:], venue=1,
max_history=k)
    away_features = build_engineered_features(histories[away][-k:], venue=0,
max_history=k)
    current_features, odd_over, odd_under = build_current_match_token(row,
available_cols, team_is_home=True)

    hybrid_features = np.array(home_features + away_features + list(current_features),
dtype=np.float32)
    x_rows.append(hybrid_features[np.newaxis, :])

    home_id = team2id[home]
    away_id = team2id[away]
    team_ids.append(np.array([home_id], dtype=np.int64))
    opp_ids.append(np.array([away_id], dtype=np.int64))
    is_home.append(np.array([1], dtype=np.int64))

    hg = int(_to_float(row["FTHG"]))
    ag = int(_to_float(row["FTAG"]))
    y_rows.append(1.0 if (hg + ag) >= 3 else 0.0)
    odds_over.append(odd_over)
    odds_under.append(odd_under)

    histories[home].append(build_match_dict(row, True))
    histories[away].append(build_match_dict(row, False))

x = np.array(x_rows, dtype=np.float32)
team_ids = np.array(team_ids, dtype=np.int64)
opp_ids = np.array(opp_ids, dtype=np.int64)

```



```

is_home = np.array(is_home, dtype=np.int64)
y = np.array(y_rows, dtype=np.float32)
odds_over = np.array(odds_over, dtype=np.float32)
odds_under = np.array(odds_under, dtype=np.float32)

meta = {
    "representation": "single_hybrid",
    "num_teams": len(teams),
    "d_feat": int(x.shape[-1]),
    "seq_len": int(x.shape[1]),
    "k": int(k),
    "num_samples": int(len(x)),
}
return x, team_ids, opp_ids, is_home, y, odds_over, odds_under, meta

```

src/transformer_single_hybrid/dataset_single_hybrid.py

```

import torch
from torch.utils.data import Dataset

class FootballSingleHybridDataset(Dataset):
    def __init__(self, x, team_ids, opp_ids, is_home, y, odds_over, odds_under):
        self.x = torch.tensor(x, dtype=torch.float32)
        self.team_ids = torch.tensor(team_ids, dtype=torch.long)
        self.opp_ids = torch.tensor(opp_ids, dtype=torch.long)
        self.is_home = torch.tensor(is_home, dtype=torch.long)
        self.y = torch.tensor(y, dtype=torch.float32)
        self.odds_over = torch.tensor(odds_over, dtype=torch.float32)
        self.odds_under = torch.tensor(odds_under, dtype=torch.float32)

```

```

def __len__(self):
    return len(self.y)

def __getitem__(self, idx):
    return (
        self.x[idx],
        self.team_ids[idx],
        self.opp_ids[idx],
        self.is_home[idx],
        self.y[idx],
        self.odds_over[idx],
        self.odds_under[idx],
    )

```

src/transformer_single_hybrid/model_single_hybrid_transformer.py

```

import numpy as np
import torch
import torch.nn as nn

```

```

class PositionalEncoding(nn.Module):
    def __init__(self, d_model, max_len=128):
        super().__init__()
        pe = torch.zeros(max_len, d_model)
        pos = torch.arange(0, max_len, dtype=torch.float32).unsqueeze(1)
        div = torch.exp(torch.arange(0, d_model, 2, dtype=torch.float32) * (-np.log(10000.0) /
d_model))
        pe[:, 0::2] = torch.sin(pos * div)

```

```

        pe[:, 1::2] = torch.cos(pos * div)
        self.register_buffer("pe", pe.unsqueeze(0))

    def forward(self, x):
        return x + self.pe[:, : x.size(1)]

class SingleHybridTransformerModel(nn.Module):
    def __init__(
        self,
        d_feat,
        num_teams,
        d_model=64,
        nhead=2,
        num_layers=1,
        dropout=0.2,
        pooling="cls",
    ):
        super().__init__()

        if pooling not in {"cls", "mean"}:
            raise ValueError("pooling must be 'cls' or 'mean'")
        if d_model % nhead != 0:
            raise ValueError("d_model must be divisible by nhead")

        self.pooling = pooling
        self.input_norm = nn.LayerNorm(d_feat)
        self.feats_proj = nn.Linear(d_feat, d_model)
        self.team_emb = nn.Embedding(num_teams, d_model)
        self.opponent_emb = nn.Embedding(num_teams, d_model)
        self.home_emb = nn.Embedding(2, d_model)

```

```

self.pos_enc = PositionalEncoding(d_model)
self.emb_dropout = nn.Dropout(dropout)

enc_layer = nn.TransformerEncoderLayer(
    d_model=d_model,
    nhead=nhead,
    dim_feedforward=d_model * 4,
    dropout=dropout,
    activation="gelu",
    batch_first=True,
    norm_first=True,
)
self.encoder = nn.TransformerEncoder(enc_layer, num_layers=num_layers)

if pooling == "cls":
    self.cls = nn.Parameter(torch.zeros(1, 1, d_model))
else:
    self.register_parameter("cls", None)

head_hidden = max(d_model // 2, 16)
self.head = nn.Sequential(
    nn.LayerNorm(d_model),
    nn.Linear(d_model, head_hidden),
    nn.GELU(),
    nn.Dropout(dropout),
    nn.Linear(head_hidden, 1),
)

def forward(self, x, team_ids, opp_ids, is_home):
    batch_size = x.size(0)

```

```

h = self.input_norm(x)
h = self.feat_proj(h)
h = h + self.team_emb(team_ids) + self.opp_emb(opp_ids) + self.home_emb(is_home)

if self.pooling == "cls":
    cls = self.cls.expand(batch_size, -1, -1)
    h = torch.cat([cls, h], dim=1)

h = self.pos_enc(h)
h = self.emb_dropout(h)
h = self.encoder(h)

if self.pooling == "cls":
    pooled = h[:, 0]
else:
    pooled = h.mean(dim=1)

return self.head(pooled).squeeze(-1)

```

src/transformer_single_hybrid/train_single_hybrid_transformer.py

```

import argparse
import csv
import json
import os
import random
from pathlib import Path

import numpy as np
import torch

```

```

import torch.nn as nn

from sklearn.metrics import accuracy_score, precision_recall_fscore_support, roc_auc_score
from torch.utils.data import DataLoader, Subset

from dataset_single_hybrid import FootballSingleHybridDataset
from model_single_hybrid_transformer import SingleHybridTransformerModel
from preprocessing_single_hybrid import preprocess_single_hybrid

os.environ["KMP_DUPLICATE_LIB_OK"] = "TRUE"
os.environ["OMP_NUM_THREADS"] = "1"

DEFAULT_OUTPUT_DIR = (
    Path(__file__).resolve().parent.parent.parent
    / "thesis_draft"
    / "baseline"
    / "transformer_single_hybrid"
)

def parse_args():
    parser = argparse.ArgumentParser(description="Train the best single_hybrid transformer in a
self-contained folder.")
    parser.add_argument("--experiment-name", type=str,
default="transformer_single_hybrid_best_attempt")
    parser.add_argument("--k", type=int, default=20)
    parser.add_argument("--train-ratio", type=float, default=0.975)
    parser.add_argument("--batch-size", type=int, default=64)
    parser.add_argument("--epochs", type=int, default=30)
    parser.add_argument("--patience", type=int, default=8)
    parser.add_argument("--seed", type=int, default=2)

```

```

parser.add_argument("--d-model", type=int, default=64)
parser.add_argument("--nhead", type=int, default=2)
parser.add_argument("--num-layers", type=int, default=1)
parser.add_argument("--dropout", type=float, default=0.2)
parser.add_argument("--pooling", choices=["cls", "mean"], default="cls")
parser.add_argument("--lr", type=float, default=1e-3)
parser.add_argument("--weight-decay", type=float, default=0.0)
parser.add_argument("--grad-clip", type=float, default=1.0)
parser.add_argument("--scheduler", choices=["none", "plateau"], default="none")
parser.add_argument("--selection-metric", choices=["auc", "accuracy", "f1"],
default="accuracy")
parser.add_argument("--decision-threshold", type=float, default=0.5)
parser.add_argument("--threshold-strategy", choices=["fixed", "train_accuracy", "train_f1"],
default="fixed")
parser.add_argument("--threshold-min", type=float, default=0.35)
parser.add_argument("--threshold-max", type=float, default=0.65)
parser.add_argument("--threshold-step", type=float, default=0.01)
parser.add_argument("--bet-edge-threshold", type=float, default=0.05)
parser.add_argument("--stake", type=float, default=10.0)
parser.add_argument("--print-every", type=int, default=1)
parser.add_argument("--output-dir", type=str, default=str(DEFAULT_OUTPUT_DIR))
return parser.parse_args()

```

```

def set_seed(seed):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    if torch.cuda.is_available():
        torch.cuda.manual_seed_all(seed)

```

```

def normalize_sequences(x, split_idx):
    train_x = x[:split_idx]
    mean = train_x.mean(axis=(0, 1), keepdims=True)
    std = train_x.std(axis=(0, 1), keepdims=True) + 1e-6
    return (x - mean) / std


def compute_metrics(labels, probs, threshold):
    preds = (probs >= threshold).astype(int)
    precision, recall, f1, _ = precision_recall_fscore_support(labels, preds, average="binary",
zero_division=0)
    metrics = {
        "accuracy": float(accuracy_score(labels, preds)),
        "precision": float(precision),
        "recall": float(recall),
        "f1": float(f1),
    }
    try:
        metrics["auc"] = float(roc_auc_score(labels, probs))
    except ValueError:
        metrics["auc"] = float("nan")
    return metrics


def find_best_threshold(labels, probs, args):
    if args.threshold_strategy == "fixed":
        return float(args.decision_threshold)

    thresholds = np.arange(args.threshold_min, args.threshold_max + 1e-9, args.threshold_step)
    best_threshold = float(args.decision_threshold)

```



```

best_score = float("-inf")

for threshold in thresholds:
    metrics = compute_metrics(labels, probs, threshold)
    score = metrics["accuracy"] if args.threshold_strategy == "train_accuracy" else
metrics["f1"]
    if score > best_score + 1e-12:
        best_score = score
        best_threshold = float(threshold)

return best_threshold


def simulate_betting(labels, probs, odds_over, odds_under, edge_threshold, decision_threshold,
stake):
    total_spent = 0.0
    total_return = 0.0
    bet_count = 0
    correct_bets = 0

    for label, prob_over, odd_over, odd_under in zip(labels, probs, odds_over, odds_under):
        if odd_over <= 1.01 or odd_under <= 1.01:
            continue

        book_prob_over = 1.0 / odd_over
        book_prob_under = 1.0 / odd_under
        margin_sum = book_prob_over + book_prob_under
        book_prob_over /= margin_sum
        book_prob_under /= margin_sum
        model_prob_under = 1.0 - prob_over

```

```

if prob_over >= decision_threshold and prob_over > book_prob_over + edge_threshold:
    total_spent += stake
    bet_count += 1
    if label == 1:
        total_return += stake * odd_over
        correct_bets += 1
    elif model_prob_under >= decision_threshold and model_prob_under > book_prob_under +
edge_threshold:
        total_spent += stake
        bet_count += 1
        if label == 0:
            total_return += stake * odd_under
            correct_bets += 1

profit = total_return - total_spent
roi = (profit / total_spent) * 100.0 if total_spent > 0 else 0.0
bet_accuracy = (correct_bets / bet_count) if bet_count > 0 else 0.0

return {
    "bet_count": int(bet_count),
    "bet_accuracy": float(bet_accuracy),
    "total_spent": float(total_spent),
    "total_return": float(total_return),
    "profit": float(profit),
    "roi": float(roi),
}

```

```

@torch.no_grad()
def evaluate(model, loader, criterion, device, args, decision_threshold):
    model.eval()

```

```

losses = []
probs = []
labels = []
odds_over = []
odds_under = []

for xb, tid, oid, hm, yb, odds_over_batch, odds_under_batch in loader:
    xb = xb.to(device)
    tid = tid.to(device)
    oid = oid.to(device)
    hm = hm.to(device)
    yb = yb.to(device)

    logits = model(xb, tid, oid, hm)
    loss = criterion(logits, yb)
    losses.append(loss.item())

    probs.extend(torch.sigmoid(logits).cpu().numpy())
    labels.extend(yb.cpu().numpy())
    odds_over.extend(odds_over_batch.numpy())
    odds_under.extend(odds_under_batch.numpy())

probs = np.array(probs)
labels = np.array(labels)
odds_over = np.array(odds_over)
odds_under = np.array(odds_under)

metrics = compute_metrics(labels, probs, decision_threshold)
metrics["loss"] = float(np.mean(losses)) if losses else 0.0
metrics["decision_threshold"] = float(decision_threshold)
metrics.update(

```

```

simulate_betting(
    labels=labels,
    probs=probs,
    odds_over=odds_over,
    odds_under=odds_under,
    edge_threshold=args.bet_edge_threshold,
    decision_threshold=decision_threshold,
    stake=args.stake,
)
)
return metrics

```

```
def main():
```

```

    args = parse_args()
    set_seed(args.seed)

```

```

    output_dir = Path(args.output_dir)
    output_dir.mkdir(parents=True, exist_ok=True)
    epoch_csv_path = output_dir / f'{args.experiment_name}_epoch_metrics.csv'
    summary_path = output_dir / f'{args.experiment_name}_summary.json'
    best_state_path = output_dir / f'{args.experiment_name}_best_state.pt'

```

```

    x, team_ids, opp_ids, is_home, y, odds_over, odds_under, meta =
preprocess_single_hybrid(k=args.k)
    split_idx = int(args.train_ratio * len(x))
    if split_idx <= 0 or split_idx >= len(x):
        raise ValueError("train_ratio produced an empty train or validation split")

    x = normalize_sequences(x, split_idx)
    dataset = FootballSingleHybridDataset(x, team_ids, opp_ids, is_home, y, odds_over,

```

```

odds_under)
train_dataset = Subset(dataset, range(0, split_idx))
val_dataset = Subset(dataset, range(split_idx, len(dataset)))

train_loader = DataLoader(train_dataset, batch_size=args.batch_size, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=args.batch_size, shuffle=False)

meta["train_samples"] = int(split_idx)
meta["val_samples"] = int(len(x) - split_idx)
meta["train_positive_rate"] = float(y[:split_idx].mean())
meta["val_positive_rate"] = float(y[split_idx:].mean())

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = SingleHybridTransformerModel(
    d_feat=x.shape[2],
    num_teams=int(team_ids.max()) + 1,
    d_model=args.d_model,
    nhead=args.nhead,
    num_layers=args.num_layers,
    dropout=args.dropout,
    pooling=args.pooling,
).to(device)

optimizer = torch.optim.AdamW(model.parameters(), lr=args.lr,
weight_decay=args.weight_decay)
criterion = nn.BCEWithLogitsLoss()

if args.scheduler == "plateau":
    scheduler = torch.optim.lr_scheduler.ReduceLROnPlateau(
        optimizer,
        mode="max",

```

```

        factor=0.5,
        patience=2,
        min_lr=1e-5,
    )
else:
    scheduler = None

best_score = float("-inf")
best_metrics = None
best_epoch = 0
best_accuracy_seen = float("-inf")
best_accuracy_epoch = 0
wait = 0
best_state = None

with epoch_csv_path.open("w", encoding="utf-8", newline="") as csv_handle:
    writer = csv.DictWriter(
        csv_handle,
        fieldnames=[
            "epoch",
            "lr",
            "decision_threshold",
            "train_loss",
            "train_accuracy",
            "train_precision",
            "train_recall",
            "train_f1",
            "train_auc",
            "val_loss",
            "val_accuracy",
            "val_precision",

```

```

        "val_recall",
        "val_f1",
        "val_auc",
        "bet_count",
        "bet_accuracy",
        "total_spent",
        "total_return",
        "profit",
        "roi",
    ],
)
writer.writeheader()

for epoch in range(1, args.epochs + 1):
    model.train()
    train_losses = []
    train_probs = []
    train_labels = []

    for xb, tid, oid, hm, yb, _, _ in train_loader:
        xb = xb.to(device)
        tid = tid.to(device)
        oid = oid.to(device)
        hm = hm.to(device)
        yb = yb.to(device)

        optimizer.zero_grad()
        logits = model(xb, tid, oid, hm)
        loss = criterion(logits, yb)
        loss.backward()
        if args.grad_clip > 0:

```

```

        torch.nn.utils.clip_grad_norm_(model.parameters(), args.grad_clip)
optimizer.step()

train_losses.append(loss.item())
train_probs.extend(torch.sigmoid(logits).detach().cpu().numpy())
train_labels.extend(yb.detach().cpu().numpy())

train_labels = np.array(train_labels)
train_probs = np.array(train_probs)
decision_threshold = find_best_threshold(train_labels, train_probs, args)

train_metrics = compute_metrics(train_labels, train_probs, decision_threshold)
train_metrics["loss"] = float(np.mean(train_losses)) if train_losses else 0.0
train_metrics["decision_threshold"] = float(decision_threshold)

val_metrics = evaluate(model, val_loader, criterion, device, args, decision_threshold)
score = val_metrics[args.selection_metric]

if val_metrics["accuracy"] > best_accuracy_seen:
    best_accuracy_seen = val_metrics["accuracy"]
    best_accuracy_epoch = epoch

if scheduler is not None and np.isfinite(score):
    scheduler.step(score)

writer.writerow(
    {
        "epoch": epoch,
        "lr": optimizer.param_groups[0]["lr"],
        "decision_threshold": decision_threshold,
        "train_loss": train_metrics["loss"],

```



```

        "train_accuracy": train_metrics["accuracy"],
        "train_precision": train_metrics["precision"],
        "train_recall": train_metrics["recall"],
        "train_f1": train_metrics["f1"],
        "train_auc": train_metrics["auc"],
        "val_loss": val_metrics["loss"],
        "val_accuracy": val_metrics["accuracy"],
        "val_precision": val_metrics["precision"],
        "val_recall": val_metrics["recall"],
        "val_f1": val_metrics["f1"],
        "val_auc": val_metrics["auc"],
        "bet_count": val_metrics["bet_count"],
        "bet_accuracy": val_metrics["bet_accuracy"],
        "total_spent": val_metrics["total_spent"],
        "total_return": val_metrics["total_return"],
        "profit": val_metrics["profit"],
        "roi": val_metrics["roi"],
    }
)

```

```

if epoch == 1 or epoch % args.print_every == 0 or epoch == args.epochs:

```

```

    print(
        f"Epoch {epoch:04d} | "
        f"Train Acc {train_metrics['accuracy'] * 100:.2f}% | "
        f"Val Acc {val_metrics['accuracy'] * 100:.2f}% | "
        f"Val AUC {val_metrics['auc']:.3f} | "
        f"Profit {val_metrics['profit']:.2f} | "
        f"ROI {val_metrics['roi']:.2f}%"
    )

```

```

if np.isfinite(score) and score > best_score + 1e-4:

```

```

        best_score = score
        best_metrics = dict(val_metrics)
        best_metrics["epoch"] = epoch
        best_metrics["train_accuracy"] = train_metrics["accuracy"]
        best_metrics["train_loss"] = train_metrics["loss"]
        best_state = {key: value.detach().cpu().clone() for key, value in
model.state_dict().items()}
        best_epoch = epoch
        wait = 0
    else:
        wait += 1
        if wait >= args.patience:
            print(f'Early stopping at epoch {epoch:04d}')
            break

if best_state is not None:
    torch.save(best_state, best_state_path)
else:
    best_state_path = None

summary = {
    "experiment_name": args.experiment_name,
    "config": vars(args),
    "meta": meta,
    "best_epoch": best_epoch,
    "best_accuracy_seen_pct": best_accuracy_seen * 100.0,
    "best_accuracy_epoch": best_accuracy_epoch,
    "best_metrics": best_metrics,
    "paths": {
        "epoch_metrics_csv": str(epoch_csv_path),
        "summary_json": str(summary_path),
    }
}

```

```

        "best_state_path": str(best_state_path) if best_state_path is not None else None,
    },
}

with summary_path.open("w", encoding="utf-8") as handle:
    json.dump(summary, handle, indent=2)

print("Best validation result:")
print(json.dumps(best_metrics, indent=2))
print(f'Best accuracy for this run: {best_accuracy_seen * 100:.2f}% (epoch
{best_accuracy_epoch})')
print("Epoch CSV written.")
print("Summary JSON written.")
if best_state_path is not None:
    print("Best state written.")

if __name__ == "__main__":
    main()

```