

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή στα Πληροφοριακά και Επικοινωνιακά Συστήματα



**Πρόβλεψη του Νικητή του Champions League 2014 - 2015, με τη
Χρησιμοποίηση του Αλγορίθμου Τεχνητής Νοημοσύνης Κατά
Πλάτος (Breadth First Search)**

Αντώνιος Χαλιάπας

**Επιβλέπων Καθηγητής
Δημήτριος Κυριάκου**

Μάιος 2015

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

**Πρόβλεψη του Νικητή του Champions League 2014 - 2015, με τη
Χρησιμοποίηση του Αλγορίθμου Τεχνητής Νοημοσύνης Κατά
Πλάτος (Breadth First Search)**

Αντώνιος Χαλιάπας

**Επιβλέπων Καθηγητής
Δημήτριος Κυριάκου**

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Μάιος 2015

Περίληψη

Το αντικείμενο της παρούσας μεταπτυχιακής διατριβής είναι η ανάλυση στατιστικών αγωνιστικής απόδοσης, για δέκα έξι (16) ομάδες του Champions League με σκοπό την πρόβλεψη του τελικού νικητή της διοργάνωσης. Το πρώτο κομμάτι της εργασίας αφορά την συλλογή των πληροφοριών σχετικά με την αγωνιστική παρουσία των ομάδων (αμυντική, επιθετική, προϊστορία) τόσο στα εγχώρια πρωταθλήματα όσο και στην ευρωπαϊκή διασυλλογική διοργάνωση. Τα δεδομένα συλλέγονται από ευρέως γνωστές ιστοσελίδες, που σκοπό έχουν την καταγραφή και την παρουσίαση στατιστικών ποδοσφαιρικών αγώνων.

Στο δεύτερο κομμάτι της μεταπτυχιακής διατριβής, επιλέγεται ο αλγόριθμος αναζήτησης τεχνητής νοημοσύνης Κατά Πλάτος, πάνω στον οποίο θα στηριχτεί η υλοποίηση της εφαρμογής πρόβλεψης. Στη συνέχεια, καταχωρούνται τα συλλεγόμενα στατιστικά σε βάση δεδομένων, που είναι απαραίτητη για την επικοινωνία της εφαρμογής με τα δεδομένα για την περαιτέρω επεξεργασία τους. Στην πορεία, πραγματοποιείται καθορισμός βαρύτητας των στατιστικών για να οριστεί η σειρά επεξεργασία τους. Η συγκεκριμένη διαδικασία επιτυγχάνεται με την καταγραφή της συχνότητας εμφάνισης των δεδομένων, στους αγώνες που διεξάγονται κατά την α' φάση των ομίλων της διοργάνωσης. Η εφαρμογή υλοποιείται βάσει αυτής της λογικής, με τη δημιουργία επιπέδων σύγκρισης των δεδομένων αγωνιστικής απόδοσης και της εξαγωγή ενός ποσοστού πρόκρισης για τις ανταγωνιζόμενες ομάδες.

Στη συνέχεια, παρουσιάζονται τα αποτελέσματα της εφαρμογής για τις διάφορες φάσεις του Champions League (β' φάση, προημιτελικά, ημιτελικά, τελικός), πραγματοποιείται μελέτη των πραγματικών αποτελεσμάτων των αγώνων για την αναπροσαρμογή της εφαρμογής, με απώτερο σκοπό την εξαγωγή εγκυρότερων αποτελεσμάτων. Τέλος, γίνεται μία προσπάθεια παρουσίασης προτάσεων για βελτίωση και μελλοντική χρήση της προκειμένης εφαρμογής πρόβλεψης.

Στις σελίδες που ακολουθούν, θα γίνει αναλυτική παρουσίαση των προβλημάτων της συγκεκριμένης μεταπτυχιακής διατριβής και ποιοί μέθοδοι χρησιμοποιήθηκαν για την επίτευξη των στόχων της.

Summary

The object of this dissertation is the statistic analysis of the performance of 16 football teams participating in the Champions League, in order to predict the winner of the tournament. The first part, refers to the accumulation of information about the performance of the football clubs (defense, attack, history), both in inner state championships and European tournament. Data is gathered from well known web sites, whose purpose is to register and present statistics of football matches.

In the second part, the Artificial Intelligence's searching algorithm Breadth First Search is chosen. The implementation of the predicting application will be based on this algorithm and the assembled data is recorded in a data base. This fact is the essential for the communication between the application and the data, in order to be analyzed further more. The importance of the statistics is defined so as to determine the weight of the process of each one. The current procedure is being fulfilled using the frequency of appearance of the data at matches being held during the group phase of the tournament. The application is being implemented using the logic of creating different levels of comparison between various data, concerning teams performance. As a result, the possibility of qualification of the football club is deducted.

Afterwards, the results of the application are presented for different phases of the Champions League (second phase, quarterfinals, semi finals, final). In addition, the results of real matches are examined in order to improve the application and deduct valid results. Finally, an effort is made to present proposals of improvement and future use of this predicting application.

In the following pages, will take place a detailed presentation of the problems that appeared during the realization of the current dissertation and the methods that were used in order to succeed its purpose.

Ευχαριστίες

Καταρχάς θα ήθελα να εκφράσω τις αληθινές μου ευχαριστίες στον επιβλέπων καθηγητή της μεταπτυχιακής μου διατριβής, κ Κυριάκου Δημήτριου, για την εμπιστοσύνη που μου έδειξε κατά την ανάθεση της παρούσας μεταπτυχιακής διατριβής αλλά και για την αμέριστη συμπαράσταση και την άρτια συνεργασία καθ' όλη τη διάρκεια της εκπόνησης.

Τέλος δε θα μπορούσα παρά να εκφράσω ένα μεγάλο ευχαριστώ στην οικογένειά μου και ιδιαίτερα στην αδερφή μου Κατερίνα για την ηθική υποστήριξη που μου παρείχαν.

Περιεχόμενα

1	Εισαγωγή	9
1.1	Τεχνητή Νοημοσύνη	9
1.2	Η Ιστορία της Τεχνητής Νοημοσύνης	10
1.3	Περιγραφή του Προβλήματος	12
1.4	Δομή Μεταπτυχιακής Διατριβής	14
2	Βιβλιογραφική Ανασκόπηση	16
2.1	Αλγόριθμοι ΤΝ και Τρόπος Επίλυσης Προβλημάτων	16
2.1.1	Βασικές Έννοιες	16
2.1.2	Αλγόριθμοι Αναζήτησης	18
2.1.3	Αλγόριθμοι Τυφλής Αναζήτησης	24
2.1.4	Αλγόριθμοι Ευριστικής Αναζήτησης	31
2.2	Ιστοσελίδες Ποδοσφαιρικών Στατιστικών	37
2.2.1	Η Ιστοσελίδα UEFA.com	38
2.2.2	Η Ιστοσελίδα WhoScored.com	39
2.2.3	Η Ιστοσελίδα Squawka	41
2.2.4	Η Ιστοσελίδα FourFourTwo	43
2.3	Τεχνολογία Συλλογής Στατιστικών Ποδοσφαίρου	45
2.4	Υπολογισμός Στοιχηματικών Αποδόσεων	46
2.5	Το Football Manager	47
2.6	Σύστημα Διεξαγωγής του Champions League	48
3	Μεθοδολογία της Έρευνας	50
3.1	Σκοπός και Προσδοκώμενα Αποτελέσματα	50
3.2	Ερευνητικά Ερωτήματα	51
3.3	Μεθοδολογία	52
3.3.1	Η Επιλογή του Αλγορίθμου Τεχνητής Νοημοσύνης	52
3.3.2	Μέθοδος Προσέγγισης	53
4	Σχεδιασμός και Υλοποίηση	55
4.1	Εισαγωγή	55
4.2	Περιγραφή του Προβλήματος	56

4.3	Προσέγγιση του Προβλήματος	57
4.3.1	Η Επιλογή Εργαλείων	57
4.3.2	Υπόθεση Γράφου	58
4.3.3	Επιλογή Στατιστικών	61
4.3.4	Καταγραφή Συχνότητας Εμφάνισης Στατιστικών	69
4.3.5	Συνδυασμός Στατιστικών	71
4.3.6	Εισαγωγή Προσωπικών Στατιστικών	76
4.4	Περιγραφή Βάσης Δεδομένων	78
4.4.1	Πίνακας Teams	80
4.4.2	Πίνακας Defensive	80
4.4.3	Πίνακας Offensive	81
4.4.4	Πίνακας Overview	82
4.4.5	Πίνακας History	83
4.5	Ανάλυση και Παρουσίαση του Κώδικα	84
4.5.1	Interface Helper	84
4.5.2	Κλάση Comparison	88
4.5.3	Interfaces Συγκρίσεων	89
4.5.4	Κλάση Main	93
5	Αξιολόγηση	95
5.1	B' Φάση - Φάση των 16	96
5.1.1	Σενάρια B' Φάσης	97
5.2	Προημιτελική Φάση	103
5.2.1	Σενάρια Προημιτελικής Φάσης	104
5.3	Ημιτελική Φάση	106
5.3.1	Σενάριο Ημιτελικής Φάσης	107
5.4	Τελική Φάση	108
6	Συμπεράσματα και Μελλοντικές Προεκτάσεις	109
6.1	Ανακεφαλαίωση	109
6.2	Μελλοντική Χρήση	110
6.3	Επίλογος	112
	Βιβλιογραφία	113

A	Κώδικας και Βάση Δεδομένων	A-1
A.1	Πηγαίος Κώδικας Εφαρμογής	A-2
A.2	Βάση Δεδομένων	A-2

Κεφάλαιο 1

Εισαγωγή

1.1 Τεχνητή Νοημοσύνη

Η εμφάνιση των ηλεκτρονικών υπολογιστών και η εξέλιξη τους με ταχύτατους ρυθμούς ενισχύουν τις προσδοκίες του ανθρώπου για την εύρεση απαντήσεων και λύσεων στα πιο σύνθετα προβλήματα διαφόρων επιστημονικών πεδίων. Οι παράγοντες που ενισχύουν τις προσδοκίες αυτές είναι: α) Η εξελιγμένες ικανότητες των υπολογιστών έχουν οδηγήσει στον καθορισμό εκ νέου ορισμού αναφορικά με το "μυαλό" ("mind"), συσχετιζόμενου με την έννοια της "νοημοσύνης" ("intelligence") καθώς και του όρου "μηχανή" ("machine"), β) Οι υπολογιστές σαν εργαλείο έχουν συνδράμει αποφασιστικά σε όλους τους επιστημονικούς τομείς, γ) Η ικανότητα κατασκευής υπερσύγχρονων

υπολογιστών με πολύ υψηλές επιδόσεις και τη δυνατότητα ενσωμάτωσης διανοητικών χαρακτηριστικών, οι οποίοι αποδεικνύουν την τεχνική τους καταλληλότητα [01, 06].

Οι εξελίξεις στον τομέα των υπολογιστών και της πληροφορικής, τα μειονεκτήματα κλασικών μεθόδων επίλυσης προβλημάτων και η επιθυμία χρησιμοποίησης μη συμβολικών μεθόδων για την επίλυση των προβλημάτων συνετέλεσαν στην ανάπτυξη της Τεχνητής Νοημοσύνης (ΤΝ, Artificial Intelligence - AI), η οποία αποτελεί ένα αναπτυσσόμενο και πολλά υποσχόμενο επιστημονικό πεδίο και προσελκύει το ενδιαφέρον πολλών ερευνητών.

Η Τεχνητή Νοημοσύνη είναι κλάδος της Επιστήμης των Υπολογιστών που ασχολείται με την συμπεριφορά, την συλλογιστική και τις διεργασίες σκέψης. Ο όρος Τεχνητή Νοημοσύνη αναφέρεται στον κλάδο της επιστήμης υπολογιστών ο οποίος ασχολείται με τη σχεδίαση και την υλοποίηση υπολογιστικών συστημάτων που μιμούνται στοιχεία της ανθρώπινης συμπεριφοράς τα οποία καταδεικνύουν έστω και στοιχειώδη ευφυΐα: μάθηση, προσαρμοστικότητα, εξαγωγή συμπερασμάτων, επίλυση προβλημάτων κλπ. Ο Τζον Μακάρθι όρισε τον τομέα αυτόν ως «επιστήμη και μεθοδολογία της δημιουργίας μηχανών με νόηση» [06].

Η ΤΝ διαιρείται στην Συμβολική Τεχνητή Νοημοσύνη, η οποία επιχειρεί να εξομοιώσει την ανθρώπινη νοημοσύνη αλγοριθμικά χρησιμοποιώντας σύμβολα και λογικούς κανόνες και στην Υπολογιστική Νοημοσύνη, η οποία προσπαθεί να αναπαράγει την ανθρώπινη νοημοσύνη χρησιμοποιώντας μαθηματικές μεθόδους που προσομοιώνουν πραγματικές βιολογικές διαδικασίες [03].

Η ΤΝ αποτελεί σημείο τομής μεταξύ πολλών πεδίων όπως της επιστήμης υπολογιστών, της φιλοσοφίας, των μαθηματικών, της ψυχολογίας, της νευρολογίας, της γλωσσολογίας και της επιστήμης μηχανικών ενώ οι κατηγορίες υλοποίησής της είναι η απόδειξη θεωρημάτων, η επεξεργασία φυσικής γλώσσας, η τεχνητή όραση, η μηχανική μάθηση, τα συστήματα γνώσης κ.α. Όλα αυτά στόχο έχουν τη σύνθεση ευφυούς συμπεριφοράς, με στοιχεία συλλογιστικής, μάθησης και προσαρμογής στο περιβάλλον, ενώ συνήθως εφαρμόζεται σε μηχανές ή υπολογιστές ειδικής κατασκευής

1.2 Η ιστορία της Τεχνητής Νοημοσύνης

Με την ανακάλυψη των σύγχρονων υπολογιστών, έγινε δυνατή η ανάπτυξη προγραμμάτων που εκτελούσαν δύσκολες νοητικές διεργασίες. Από αυτά τα προγράμματα κατασκευάστηκαν εργαλεία που έχουν εφαρμογή σε ένα ευρύ πλήθος προβλημάτων της καθημερινότητας.

Κατά τη δεκαετία του 1940 παρατηρείται η εμφάνιση της πρώτης μαθηματικής περιγραφής τεχνητού νευρωνικού δικτύου, το οποίο παρουσίαζε περιορισμένες δυνατότητες επίλυσης αριθμητικών προβλημάτων.

Το 1950 ο μαθηματικός Άλαν Τούρινγκ, πατέρας της θεωρίας υπολογισμού και της τεχνητής νοημοσύνης, πρότεινε τη δοκιμή Τούρινγκ, μία απλή δοκιμασία που θα μπορούσε να εξακριβώσει την ευφυή συμπεριφορά των μηχανών [06]. Η τεχνητή νοημοσύνη θεμελιώθηκε τυπικά ως όρο και πεδίο στη συνάντηση ορισμένων Αμερικανών επιστημόνων του τομέα το 1956. Όσοι παρακολούθησαν το συνέδριο, ήταν αυτοί που θα αποτελούσαν, στην πορεία τους πρωταγωνιστές για την έρευνα και την ανάπτυξη της ΤΝ για πολλές δεκαετίες. Τη χρονιά αυτή παρουσιάστηκε για πρώτη φορά και το Logic Theorist, ένα πρόγραμμα που θα "σκέφτονταν μη-αριθμητικά", το οποίο στηριζόταν σε συμπερασματικούς κανόνες τυπικής λογικής και σε ευρετικούς αλγορίθμους αναζήτησης για να αποδεικνύει μαθηματικά θεωρήματα [14]. Επόμενοι σημαντικοί σταθμοί ήταν η ανάπτυξη της υψηλού επιπέδου γλώσσας προγραμματισμού LISP το 1958 από τον Τζον Μακάρθι, δηλαδή της πρώτης γλώσσας συναρτησιακού προγραμματισμού η οποία προσανατολίζεται στο χειρισμό λιστών από σύμβολα.

Στις αρχές της δεκαετίας του '60 παρατηρείται η εμφάνιση των γενετικών αλγορίθμων και η παρουσίαση του βελτιωμένου νευρωνικού δικτύου perceptron από τον Ρόσενμπλατ [06]. Στα τέλη της δεκαετίας του '60 όμως άρχισε μία περίοδος στασιμότητας για την ΤΝ, μία εποχή υποχρηματοδότησης των ερευνητικών προγραμμάτων καθώς όλα τα μέχρι τότε εργαλεία του χώρου ήταν κατάλληλα μόνο για την επίλυση εξαιρετικά απλών προβλημάτων.

Στα μέσα του '70 ωστόσο προέκυψε μία αναθέρμανση του ενδιαφέροντος για τον τομέα λόγω των εμπορικών εφαρμογών που απέκτησαν τα έμπειρα συστήματα, μηχανές ΤΝ με αποθηκευμένη γνώση για έναν εξειδικευμένο τομέα και δυνατότητα ταχείας εξαγωγής λογικών συμπερασμάτων. Παράλληλα έκανε την εμφάνισή της η γλώσσα λογικού προγραμματισμού Prolog η οποία έδωσε νέα ώθηση στη συμβολική ΤΝ, ενώ στις αρχές της δεκαετίας του '80

άρχισαν να υλοποιούνται πολύ πιο ισχυρά και με περισσότερες εφαρμογές νευρωνικά δίκτυα, όπως τα πολυεπίπεδα perceptron και τα δίκτυα Hopfield, τα οποία χρησιμοποιήθηκαν ευρέως σε εφαρμογές των υπολογιστών και της ψυχολογίας [06]. Ταυτόχρονα οι γενετικοί αλγόριθμοι και άλλες συναφείς μεθοδολογίες αναπτύσσονταν πλέον από κοινού, κάτω από την ομπρέλα του εξελικτικού υπολογισμού.

Κατά τη δεκαετία του '90, με την αυξανόμενη σημασία και μετάδοση του Internet, γνώρισαν ανάπτυξη οι ευφυείς πράκτορες, τα συστήματα αισθητήρων, το αυτόνομο λογισμικό TN τοποθετημένο σε κάποιο περιβάλλον με το οποίο αλληλεπιδρά, τα οποία βρήκαν μεγάλο πεδίο εφαρμογών τόσο στον οικονομικό όσο και στον βιομηχανικό τομέα. Οι πράκτορες έχουν στόχο συνήθως στη συλλογή ή ανάλυση τεράστιων συνόλων δεδομένων ή στην αυτοματοποίηση επαναλαμβανόμενων διαδικασιών, στην παροχή βοήθειας στους χρήστες τους, ενώ συνοψίζουν όλες τις γνωστές μεθοδολογίες TN στους τρόπους κατασκευής και λειτουργίας τους. Έτσι σήμερα, η TN ορίζεται ως η επιστήμη που μελετά τη σχεδίαση και υλοποίηση ευφύων πρακτόρων. Επίσης τη δεκαετία του '90 η TN, κυρίως η μηχανική μάθηση (machine learning) και η εξόρυξη γνώσης (data mining), άρχισε να επηρεάζεται πολύ από τη θεωρία πιθανοτήτων και τη στατιστική. Τα μπεϋζιανά δίκτυα (bayes networks) είναι η εστίαση αυτής της νέας μετακίνησης που παρέχει τις συνδέσεις με τα πιο σχολαστικά θέματα της στατιστικής και της επιστήμης μηχανικών, όπως τα πρότυπα Markov και τα φίλτρα Kalman. Αυτή η νέα πιθανοκρατική προσέγγιση έχει αυστηρά υποσυμβολικό χαρακτήρα, όπως και οι τρεις μεθοδολογίες οι οποίες κατηγοριοποιούνται κάτω από την ετικέτα της υπολογιστικής νοημοσύνης: τα νευρωνικά δίκτυα, ο εξελικτικός υπολογισμός και η ασαφής λογική [01].

1.3 Περιγραφή του Προβλήματος

Το άθλημα του ποδοσφαίρου έχει αποκτήσει τεράστια δημοτικότητα τα τελευταία χρόνια σε παγκόσμιο επίπεδο, με όλο και περισσότερους ανθρώπους να ασχολούνται με αυτό. Ένας τομέας, που αναπτύσσεται παράλληλα με αυτό είναι και η στατιστική ανάλυση αριθμητικών δεδομένων αγωνιστικής απόδοσης διαφόρων ποδοσφαιρικών συλλόγων, για την ανάλυση της απόδοσης τους, την πρόβλεψη τόσο των μελλοντικών

τους επιδόσεων όσο και των αποτελεσμάτων τους σε διάφορους αγώνες που συμμετέχουν.

Οι αλγόριθμοι αναζήτησης αναζητούν τη λύση σε ένα πρόβλημα και αφορούν τον σχεδιασμό κατάλληλων ενεργειών με στόχο την άφιξη ενός συστήματος σε μία αποδεκτή τελική κατάσταση, εκκινώντας από κάποια αρχική κατάσταση που έχει καθοριστεί. Οι αλγόριθμοι λαμβάνουν ως είσοδο το πρόβλημα που έχει δοθεί και επιστρέφουν ως έξοδο μία λύση σε αυτό, αφού αξιολογήσουν πρώτα μία ομάδα υποψηφίων λύσεων.

Στην παρούσα μεταπτυχιακή διατριβή θα γίνει μελέτη διαφόρων αλγορίθμων αναζήτησης Τεχνητής Νοημοσύνης, με την επιλογή ενός από αυτούς για την ανάλυση στατιστικών αγωνιστικής απόδοσης διαφόρων ποδοσφαιρικών συλλόγων για την πρόβλεψη του νικητή του Champions League. Ονομαστικά οι αλγόριθμοι που θα περιγραφούν είναι:

- Αλγόριθμος κατά Πλάτος (Breadth First Search)
- Αλγόριθμος πρώτα σε Βάθος (Depth First Search)
- Άπληστη αναζήτηση πρώτα στο Καλύτερο (Greedy Best First Search)
- Αναζήτηση A^* (A^*)

Θα συλλεχθούν ομαδικά στατιστικά για τις ομάδες που έχουν προκριθεί στη β' φάση της διοργάνωσης του Champions League. Στη συνέχεια, θα γίνει επιλογή του αλγορίθμου Κατά Πλάτος που σκοπό έχει, μέσω της σύγκρισης των καταγεγραμμένων στατιστικών σε επίπεδα, την έγκυρη εξαγωγή αποτελέσματος για τον τελικό νικητή. Δεν θα έχουν όλα τα στατιστικά την ίδια προτεραιότητα αλλά θα διαχωρίζονται σε διάφορα επίπεδα ανάλογα με τη βαρύτητα που θα τους δοθεί. Έτσι, η εφαρμογή θα τα αναλύει και θα τα συγκρίνει με τη σειρά προτεραιότητας τους, θα πραγματοποιεί κάποιους υπολογισμούς μεταξύ των δοθέντων αριθμητικών δεδομένων και θα εξάγει το τελικό αποτέλεσμα. Επίσης, εκτός από τον τελικό νικητή, η εφαρμογή θα κάνει τις προβλέψεις για τους νικητές όλων των ζευγαριών που θα προκύψουν στη διοργάνωση. Θα πραγματοποιηθούν οι αγώνες και ανάλογα με τις αποκλίσεις που θα παρουσιαστούν στα αποτελέσματα τους θα επιχειρηθούν στη συνέχεια να γίνουν προσαρμογές στην εφαρμογή, με την αλλαγή σειράς εξέτασης και συσχετισμών μεταξύ των στατιστικών, έτσι ώστε να επιτευχθεί αποτελεσματικότερη και εγκυρότερη εξαγωγή τελικού αποτελέσματος.

Στο πρόβλημα αυτό, θα ήταν επιθυμητό τόσο ένα μεγάλο ποσοστό πρόβλεψης για τις ομάδες που θα προκριθούν στα διάφορα ζευγάρια που θα προκύψουν, όσο και η απεικόνιση ρεαλιστικών ποσοστιαίων διαφορών για τα διάφορα παιχνίδια, αλλά και η έγκυρη τελική πρόβλεψη του νικητή της διοργάνωσης.

1.4 Δομή Μεταπτυχιακής Διατριβής

Στο κεφάλαιο 2 γίνεται μία επισκόπηση για τους βασικούς αλγορίθμους αναζήτησης, με τους οποίους θα ασχοληθούμε, όπως επίσης παρουσιάζονται κάποιες βασικές έννοιες που διέπουν τους συγκεκριμένους αλγορίθμους. Επίσης, πραγματοποιείται παρουσίαση ιστοσελίδων που περιέχουν στατιστικά ποδοσφαιρικών ομάδων, και περιγράφεται σχετική τεχνολογία που χρησιμοποιείται για την καταγραφή στατιστικών σε ποδοσφαιρικούς αγώνες. Τέλος, γίνεται επισκόπηση της έκτασης που έχει πάρει στη σύγχρονη εποχή το βίντεο - παιχνίδι, Football Manager, όπως επιπροσθέτως παρουσιάζεται το σύστημα τέλεσης αγώνων της ποδοσφαιρικής διοργάνωσης Champions League.

Στο κεφάλαιο 3 παρουσιάζονται τα βασικά ερευνητικά ερωτήματα που προκύπτουν από τη συγκεκριμένη μεταπτυχιακή διατριβή, όπως επίσης και η μεθοδολογία που θα ακολουθηθεί για την υλοποίηση της εφαρμογής πρόβλεψης, δηλαδή της επαναληπτικής μεθόδου αναζήτησης (Iterative Searching) αλλά και του αλγορίθμου αναζήτησης Κατά Πλάτος (Breadth - First - Search).

Στο κεφάλαιο 4 πραγματοποιείται η ανάλυση και η σχεδίαση του προβλήματος πρόβλεψης του νικητή της διοργάνωσης του Champions League. Συγκεκριμένα, παρουσιάζονται τα εργαλεία υλοποίησης δηλαδή, η γλώσσα προγραμματισμού εφαρμογής Java, η σύνθεση και το εργαλείο δημιουργίας της βάσης δεδομένων, καθώς και η μορφή του γράφου υλοποίησης του προβλήματος. Επιπροσθέτως, πραγματοποιείται η επιλογή των στατιστικών που θα συνθέσουν τη βάση δεδομένων, ο καθορισμός της βαρύτητάς τους αλλά και ο συνδυασμός μεταξύ τους. Τέλος, γίνεται εισαγωγή προσωπικών στατιστικών στην υπάρχουσα βάση δεδομένων.

Στη συνέχεια, στο κεφάλαιο 5, γίνεται ανάλυση των αποτελεσμάτων της εφαρμογής πρόβλεψης. Επίσης, πραγματοποιείται συγκριτική αξιολόγηση των αποτελεσμάτων σε διάφορες φάσεις της διοργάνωσης του Champions League αλλά και με διάφορους συνδυασμούς των στατιστικών που χρησιμοποιούνται στη βάση δεδομένων.

Τέλος, στο κεφάλαιο 6, παρουσιάζονται τα συμπεράσματα σχετικά με την εφαρμογή πρόβλεψης που υλοποιήθηκε και γίνεται προσπάθεια απάντησης ερωτημάτων, για τη μελλοντική χρήση της εφαρμογής σε μελλοντικά αθλητικά γεγονότα.

Κεφάλαιο 2

Βιβλιογραφική Ανασκόπηση

2.1 Αλγόριθμοι ΤΝ και Τρόπος Επίλυσης Προβλημάτων

2.1.1 Βασικές Έννοιες

Στην παρούσα μεταπτυχιακή διατριβή θα χρησιμοποιήσουμε έναν αλγόριθμο αναζήτησης τεχνητής νοημοσύνης. Πιο συγκεκριμένα, η διαδικασία πρόβλεψης αναφέρεται ως πρόβλημα αναζήτησης (searching problem) στην ΤΝ, ως εκ τούτου θα γίνει χρήση τέτοιου αλγορίθμου.

Σε αυτό το σημείο θα ήταν σκόπιμο να αναφερθούμε στη λειτουργία και στα χαρακτηριστικά των συγκεκριμένων αλγορίθμων. Αρχικά, θα εστιάσουμε την προσοχή μας, σε ορισμένες βασικές έννοιες που λαμβάνουν χώρα στη διαδικασία της αναζήτησης.

- **Πρόβλημα:** Ένα πρόβλημα ορίζεται ως η τετράδα $P = (I, G, T, S)$ [13]
 - ♦ όπου I είναι η αρχική κατάσταση, $I \in S$
 - ♦ G είναι το σύνολο των τελικών καταστάσεων, $G \subseteq S$
 - ♦ T είναι ένα σύνολο τελεστών μετάβασης, $T: S \rightarrow S$
 - ♦ S είναι ο χώρος καταστάσεων.
- **Αρχική κατάσταση:** Η αρχική κατάσταση (initial state) εκφράζει τα δεδομένα του προς εξέταση προβλήματος.
- **Τελική κατάσταση:** Η τελική κατάσταση (final ή goal state) εκφράζει τα ζητούμενα του προς εξέταση προβλήματος.
- **Αναζήτηση:** Η λύση σε ένα πρόβλημα δίνεται με την εφαρμογή ενός αλγορίθμου αναζήτησης. Κάθε αλγόριθμος έχει τέτοια χαρακτηριστικά που το κάνουν ικανό σε κάποιο πρόβλημα να αναζητήσει πιο αποδοτικά τη λύση στο χώρο αναζήτησης από κάποιο άλλο.
- **Τελεστές μετάβασης:** Οι καταστάσεις ενός κόσμου συνδέονται μεταξύ τους. Τελεστές μετάβασης (transition operators) ή ενέργειες (actions) είναι μία αντιστοίχιση μίας κατάστασης του προβλήματος σε νέες καταστάσεις.

- **Περιγραφή Προβλήματος:** Η περιγραφή του χώρου καταστάσεων, ορισμός τελεστών μετάβασης, περιγραφή της αρχικής κατάστασης και των τελικών καταστάσεων. Η περιγραφή ενός προβλήματος μπορεί να γίνει με δύο βασικούς τρόπους:
 - ◆ Περιγραφή με χώρο καταστάσεων (State Space). Κατάσταση ενός προβλήματος είναι ένα στιγμιότυπο (instance) μίας συγκεκριμένης στιγμής της εξέλιξης του προς εξέτασης προβλήματος. Χώρος καταστάσεων ονομάζεται το σύνολο όλων των έγκυρων καταστάσεων.
 - ◆ Περιγραφή με Αναγωγή (Reduction).
- **Λύση:** Λύση (Solution) σε ένα πρόβλημα (ID G, T, S), είναι μία ακολουθία από τελεστές μετάβασης t_1, t_2, \dots, t_n με την ιδιότητα $g = t_n(\dots(t_2(t_1(I))))$, όπου $g \in G$. Δηλαδή λύση είναι μία σειρά ενεργειών που οδηγεί από μία αρχική κατάσταση σε τελική κατάσταση ή στόχου [13].

2.1.2 Αλγόριθμοι Αναζήτησης

Οι αλγόριθμοι αναζήτησης αναζητούν τη λύση σε ένα πρόβλημα και αφορούν τον σχεδιασμό κατάλληλων ενεργειών με στόχο την άφιξη ενός συστήματος σε μία αποδεκτή τελική κατάσταση, εκκινώντας από κάποια αρχική κατάσταση που έχει καθοριστεί. Οι αλγόριθμοι λαμβάνουν ως είσοδο το πρόβλημα που έχει δοθεί και επιστρέφουν ως έξοδο μία λύση σε αυτό, αφού αξιολογήσουν πρώτα μία ομάδα υποψηφίων λύσεων. Πρόκειται για ένα θεμελιώδες γνωστικό πεδίο της Τεχνητής Νοημοσύνης το οποίο γνώρισε μεγάλη ανάπτυξη από τη δεκαετία του 1950. Αποτέλεσε μία προσπάθεια αλγοριθμικής εξομοίωσης της διαδικασίας της σκέψης, ενώ με τον καιρό ενσωμάτωσε μεθοδολογίες από τη θεωρία βελτιστοποίησης και τη θεωρία γράφων [13]. Ο λόγος για την ύπαρξη τόσων αλγορίθμων είναι ότι κάθε ένας έχει διαφορετική πολυπλοκότητα (complexity) που τον καθιστά περισσότερο ή λιγότερο αποδοτικό (efficient) σε απαίτηση μνήμης ή χρόνο εκτέλεσης από άλλους αλγόριθμους κατά τη διάρκεια της αναζήτησης λύσεων σε συγκεκριμένη κατηγορία προβλημάτων.

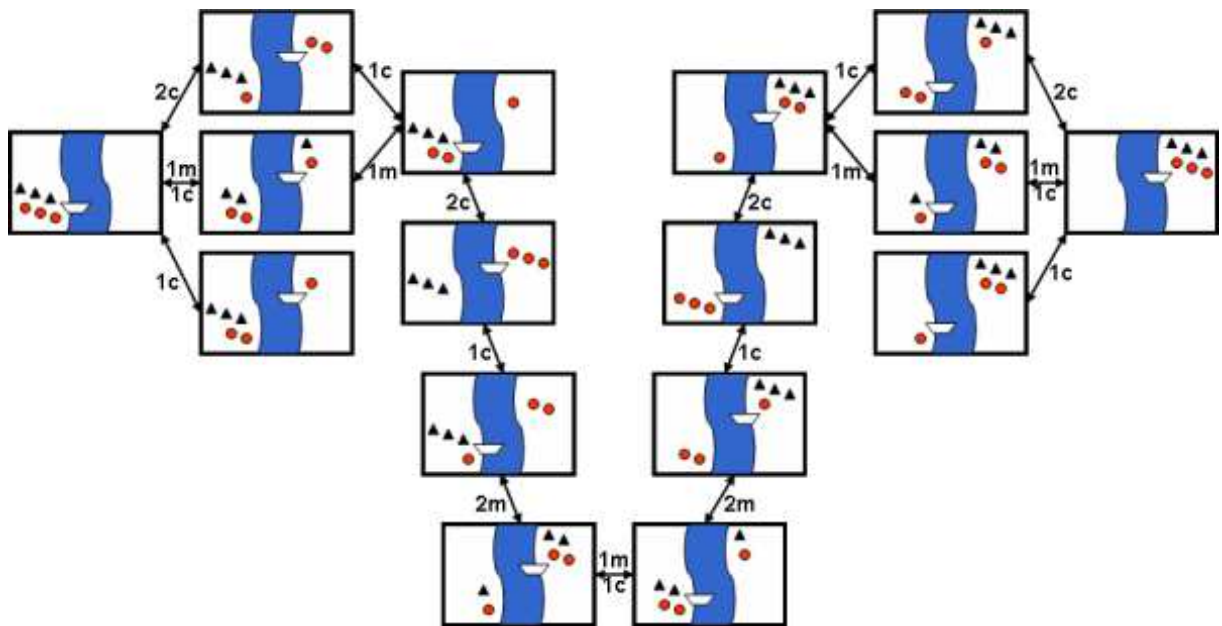
Παρακάτω παρουσιάζονται οι αλγόριθμοι αναζήτησης οι οποίοι χωρίζονται σε τρεις κατηγορίες: Τυφλής Αναζήτησης, Ευριστικής Αναζήτησης και τα Παιχνίδια Δύο (2) Ατόμων. Συγκεκριμένα έχουμε:

ΤΥΦΛΗΣ ΑΝΑΖΗΤΗΣΗΣ		
Όνομα Αλγορίθμου	Συντομογραφία	Ελληνική Ορολογία
Depth - First - Search	DFS	Αναζήτηση Πρώτα σε Βάθος
Breadth - First - Search	BFS	Αναζήτηση Κατά Πλάτος
Iterative Deepening	ID	Επαναληπτική Εκβάθυνση
Bi - directional Search	BiS	Αναζήτηση Διπλής Κατεύθυνσης
Branch and Bound	B&B	Επέκταση και Οριοθέτηση
ΕΥΡΙΣΤΙΚΗΣ ΑΝΑΖΗΤΗΣΗΣ		
Hill Climbing	HC	Αναρρίχηση Λόφων
Best - First - Search	BestFS	Αναζήτηση Πρώτα στο Καλύτερο
A* (A-star)	A*	A* (Άλφα Άστρο)
ΠΑΙΧΝΙΔΙΑ 2 ΑΤΟΜΩΝ		
Minimax	Minimax	Αναζήτηση Μεγίστου - Ελαχίστου
Alpha - Beta	AB	Άλφα - Βήτα

Πίνακας 2.1: Η συνοπτική παρουσίαση των αλγορίθμων αναζήτησης και αντίστοιχες κατηγορίες στις οποίες ανήκουν.

Χώρος Αναζήτησης

Ένα πρόβλημα ορίστηκε παραπάνω ως τετράδα (I, G, T, S) , όπου I είναι η αρχική κατάσταση, G είναι το σύνολο των τελικών καταστάσεων, T είναι το σύνολο των τελεστών μετάβασης και S είναι ο χώρος καταστάσεων. Ο αλγόριθμος αναζήτησης σκοπό έχει να προσπαθήσει να βρει τη λύση μέσα στο χώρο καταστάσεων. Δοθέντος ενός προβλήματος (I, G, T, S) , χώρος αναζήτησης (search space) SP είναι το σύνολο όλων των καταστάσεων που είναι προσβάσιμες από την αρχική κατάσταση. Μία κατάσταση s ονομάζεται προσβάσιμη (accessible) αν υπάρχει μία ακολουθία τελεστών μετάβασης t_1, t_2, \dots, t_n τέτοια ώστε $s = t_n(\dots(t_2(t_1(I))))$. Ο χώρος αναζήτησης είναι υποσύνολο του χώρου καταστάσεων, $SP \subseteq S$ [03].



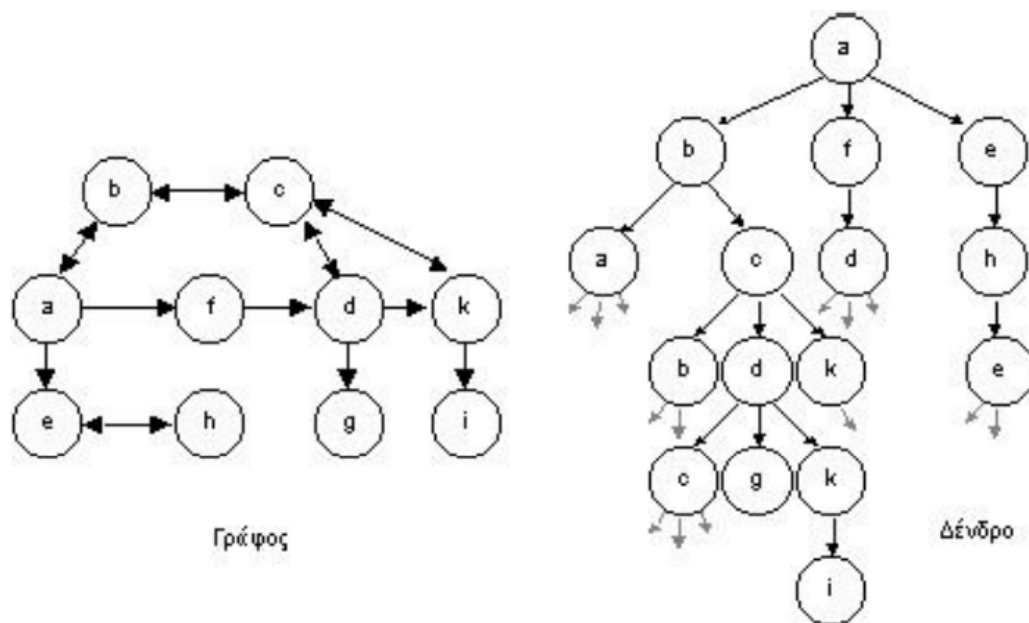
Σχήμα 2.2: Ο χώρος αναζήτησης στο πρόβλημα διάσχισης των ιεραποστόλων και των κανίβαλων.

Ο χώρος αναζήτησης μπορεί να αναπαρασταθεί με γράφους. Πολλές φορές είναι εφικτό να μετατραπεί ο γράφος σε δένδρο αναζήτησης (search tree), το οποίο όμως μπορεί να έχει μονοπάτια απείρου μήκους. Ο παρακάτω πίνακας παρουσιάζει την αντιστοιχία εννοιών δομής δένδρου με το χώρο αναζήτησης ενός προβλήματος.

Τμήμα Δένδρου	Αναπαράσταση
Κόμβος (Node)	Κατάσταση
Ρίζα (Root)	Αρχική Κατάσταση
Αρχική Κατάσταση	Τελική Κατάσταση, δηλαδή κατάσταση στην οποία δεν μπορεί να εφαρμοστεί κανένας τελεστής μετάβασης.
Κλαδί (Branch)	Τελεστής Μετάβασης που μετατρέπει μία κατάσταση - Γονέα (Parent State) σε μία άλλη κατάσταση - Παιδί (Child State).
Λύση (Solution)	Μονοπάτι (Path) που ενώνει την αρχική με μία τελική κατάσταση.
Επέκταση (Expansion)	Η διαδικασία παραγωγής όλων των καταστάσεων - παιδιών ενός κόμβου.
(Branching Factor)	Ο αριθμός των καταστάσεων - Παιδιών που προκύπτουν από την επέκταση μίας κατάστασης.

Πίνακας 2.3: Η συνοπτική παρουσίαση της αντιστοιχίας μίας δομής δένδρου με το χώρο αναζήτησης ενός προβλήματος.

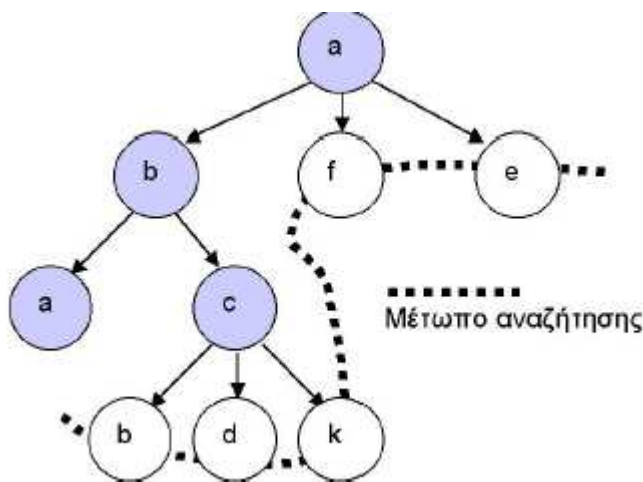
Στο παρακάτω σχήμα απεικονίζεται ένας χώρος αναζήτησης και το αντίστοιχο δένδρο. Οι κόμβοι α μέχρι κ είναι καταστάσεις. Η ρίζα α είναι η αρχική κατάσταση. Τα φύλλα g και i είναι αντίστοιχα η τελική κατάσταση και ένα αδιέξοδο. Οι συνδέσεις μεταξύ κόμβων είναι τα κλαδιά του δένδρου. Μία λύση στο πρόβλημα αναζήτησης αποτελούν τα κλαδιά που συνδέουν με τη σειρά τους κόμβους $a \rightarrow b \rightarrow c \rightarrow d \rightarrow g$. Ο παράγοντας διακλάδωσης (branching factor) εκφράζει τον αριθμό των καταστάσεων που προκύπτουν από μία άλλη κατάσταση. Για παράδειγμα, ο παράγοντας διακλάδωσης είναι 3 για τον κόμβο α, 2 για τον κόμβο f, 1 για τον e, κ.ο.κ. Όπως προαναφέρθηκε, το αντίστοιχο δένδρο μπορεί να έχει και ακολουθία κλαδιών απείρου μήκους, όπως φαίνεται από στο παρακάτω σχήμα.



Σχήμα 2.4: Αναπαράσταση γράφου, και η αντιστοίχησή του σε δένδρο

Το πρόβλημα που παρουσιάζεται στους χώρους αναζήτησης πραγματικών προβλημάτων είναι ο γρήγορος ρυθμός με τον οποίον αναπτύσσεται το δένδρο. Το φαινόμενο αυτό της εκθετικής αύξησης του αριθμού των κόμβων του δένδρου ονομάζεται συνδυαστική έκρηξη (combinatorial explosion) [06]. Λόγω της συνδυαστικής έκρηξης αυξάνονται εκθετικά και οι απαιτήσεις ενός αλγορίθμου σε μνήμη και χρόνο, με αποτέλεσμα να είναι πρακτικά αδύνατον να βρεθεί λύση σε πραγματικό χρόνο.

Οι αλγόριθμοι αναζήτησης, παρόλες τις διαφορές τους χρησιμοποιούν κάποιες κοινές δομές δεδομένων όπως είναι το μέτωπο αναζήτησης (frontier), το οποίο είναι το διατεταγμένο σύνολο των καταστάσεων που ο αλγόριθμος έχει ήδη επισκεφτεί, αλλά δεν έχουν ακόμη επεκταθεί ώσπου να βρεθεί η τελική κατάσταση [13]. Επίσης, παρατηρείται το κλειστό σύνολο (closed set) ενός αλγορίθμου αναζήτησης που είναι το σύνολο όλων των καταστάσεων που έχουν ήδη επεκταθεί από τον αλγόριθμο [13]. Στο παρακάτω σχήμα παρουσιάζεται το μέτωπο αναζήτησης, σε ένα πρόβλημα αναζήτησης γράφου.



Σχήμα 2.5: Το μέτωπο αναζήτησης σε ένα πρόβλημα αναζήτησης γράφου

Χαρακτηριστικά Αλγορίθμων

Ένας αλγόριθμος είναι μία αυστηρά καθορισμένη ακολουθία βημάτων-εντολών, η οποία επιδιώκει να λύσει ένα πρόβλημα. Μετά την εφαρμογή κάποιου αλγορίθμου στο χώρο αναζήτησης ενός προβλήματος $P=(I, G, T, S)$ προκύπτει το λυμένο πρόβλημα (solved problem), το οποίο ορίζεται ως μία τετράδα $P_s=(V, A, F, G_s)$ όπου:

- V είναι το σύνολο των καταστάσεων που εξέτασε ο αλγόριθμος αναζήτησης,
- A είναι ο αλγόριθμος που χρησιμοποιήθηκε,
- F είναι το σύνολο των λύσεων που βρέθηκαν, και
- G_s είναι το σύνολο των τελικών καταστάσεων που εξετάστηκαν.

Όταν το σύνολο των καταστάσεων που εξετάζει ο αλγόριθμος για να βρει τις απαιτούμενες λύσεις είναι ίσο με το χώρο αναζήτησης, δηλαδή $V=SP$, τότε ο αλγόριθμος ονομάζεται εξαντλητικός (exhaustive) [13]. Ένας αλγόριθμος δε λύνει πάντα κάποιο πρόβλημα, έστω και αν υπάρχει κάποια λύση. Τότε τα σύνολα GS και F είναι κενά. Ένας αλγόριθμος αναζήτησης ονομάζεται πλήρης (complete) αν εγγυάται ότι θα βρει μία λύση για οποιαδήποτε τελική κατάσταση, αν η λύση υπάρχει. Σε αντίθετη περίπτωση, ο αλγόριθμος ονομάζεται ατελής (incomplete). Ωστόσο για ορισμένες περιπτώσεις είναι σίγουρο πως αν ο αλγόριθμος δε βρει λύση, τότε οπωσδήποτε δεν υπάρχει λύση στο πρόβλημα, όπως για παράδειγμα στην περίπτωση που ένας αλγόριθμος είναι εξαντλητικός. Μερικά προβλήματα έχουν διατεταγμένο το σύνολο των τελικών καταστάσεων, σύμφωνα με τη σημαντικότητα - αξία της κάθε τελικής κατάστασης. Η μετάβαση από μία κατάσταση σε μία άλλη έχει ένα κόστος, μία λύση ονομάζεται βέλτιστη (optimal) αν οδηγεί στην καλύτερη - με το χαμηλότερο κόστος, σύμφωνα με τη διάταξη, τελική κατάσταση. Όταν δεν υπάρχει διάταξη, μία λύση ονομάζεται βέλτιστη αν είναι η συντομότερη (shortest), δηλαδή αν περιέχει το μικρότερο αριθμό τελεστών μετάβασης που οδηγούν σε κάποια τελική κατάσταση. Διαφορετικοί αλγόριθμοι βρίσκουν λύσεις διαφορετικής ποιότητας. Ένας αλγόριθμος αναζήτησης καλείται αποδεκτός (admissible) αν εγγυάται ότι θα βρει τη βέλτιστη λύση, αν μια τέτοια λύση υπάρχει.

Κριτήρια Επιλογής Αλγορίθμων Αναζήτησης

Η επιλογή του κατάλληλου αλγορίθμου, για τη λύση ενός προβλήματος, γίνεται βάση κάποιων κριτηρίων. Η επιλογή συνήθως εξαρτάται από τη φύση του προβλήματος και σε μεγάλο βαθμό από τους περιορισμούς που παρουσιάζονται. Παρακάτω παρουσιάζονται συνοπτικά τα κριτήρια επιλογής:

- Αριθμός καταστάσεων που επισκέπτεται
- Δυνατότητα εύρεσης λύσεων, εφόσον αυτές υπάρχουν
- Αριθμός και ποιότητα λύσεων
- Αποδοτικότητα σε χρόνο
- Αποδοτικότητα σε χώρο μνήμης
- Πολυπλοκότητα σχετικά με την υλοποίηση του

Οι αλγόριθμοι αναζήτησης, όπως σημειώθηκε παραπάνω, χωρίζονται σε τρεις (3) μεγάλες κατηγορίες της Τυφλής Αναζήτησης, Ευριστικής Αναζήτησης και τα Παιχνίδια Δύο (2) Ατόμων. Κάθε κατηγορία παρουσιάζει διάφορα πλεονεκτήματα και μειονεκτήματα, καθώς και καταλληλότητα υλοποίησης σε διάφορης φύσης προβλήματα. Στις επόμενες ενότητες θα γίνει αναλυτική παρουσίαση των σημαντικότερων αλγορίθμων αναζήτησης. Συγκεκριμένα, θα γίνει περιγραφή δύο (2) αλγορίθμων τυφλής αναζήτησης, Πρώτα σε Βάθος (DFS) και Πρώτα σε Πλάτος (BFS) καθώς και δύο (2) αλγορίθμων ευριστικής αναζήτησης, Άπληστη Αναζήτηση Πρώτα στο Καλύτερο (GBFS) και Άλφα Άστρο (A^*)

2.1.3 Αλγόριθμοι Τυφλής Αναζήτησης

Οι αλγόριθμοι τυφλής αναζήτησης (blind search algorithms) εφαρμόζονται σε προβλήματα που δεν υπάρχει πληροφορία για το χώρο αναζήτησης. Έτσι, οι αλγόριθμοι αυτοί αντιμετωπίζουν ακριβώς με τον ίδιο τρόπο οποιοδήποτε πρόβλημα καλούνται να λύσουν. Για τους αλγόριθμους τυφλής αναζήτησης, είναι αδιάφορο το τι απεικονίζει η κάθε κατάσταση και δεν μπορεί να γίνει διάκριση ανάμεσα στην κατάσταση στόχου και στις υπόλοιπες καταστάσεις. Σημασία έχει η χρονική αλληλουχία μεταξύ των προς επέκταση ενεργειών και καταστάσεων.

Αναζήτηση Κατά Πλάτος (Breadth - First - Search)

Ο αλγόριθμος αναζήτησης Κατά Πλάτος (Breadth - First - Search, BFS) εξετάζει πρώτα όλες τις καταστάσεις που βρίσκονται στο ίδιο βάθος και μετά συνεχίζει την επέκταση των καταστάσεων στο επόμενο επίπεδο. Ο αλγόριθμος Κατά Πλάτος είναι μία απλή στρατηγική, στην οποία πρώτα επεκτείνεται η ρίζα του δένδρου, στη συνέχεια επεκτείνονται οι απόγονοι της ρίζας, στην πορεία οι απόγονοι των απογόνων κ.ο.κ. Όλοι οι κόμβοι επεκτείνονται σε δεδομένο βάθος του δένδρου αναζήτησης, πριν οποιοσδήποτε κόμβους του επόμενου επιπέδου επεκταθεί [01]. Η περιγραφή του αλγορίθμου είναι:

1. Βάλε την αρχική κατάσταση στο μέτωπο αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλε την πρώτη κατάσταση από το μέτωπο αναζήτησης.

4. Αν η κατάσταση βρίσκεται στο κλειστό σύνολο τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι η τελική τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλη λύση πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε τους τελεστές μετάβασης για να βρεις τις καταστάσεις - παιδιά.
8. Βάλε τις καταστάσεις - παιδιά στο τέλος του μετώπου αναζήτησης.
9. Βάλε την κατάσταση - γονέα στο κλειστό σύνολο.
10. Πήγαινε στο βήμα 2.

Ο αλγόριθμος Κατά Πλάτος (BFS) είναι ένα στιγμιότυπο του γενικότερου κανόνα αναζήτησης σε γράφο κατά τον οποίο ο πιο "ρηχός" κόμβος που δεν έχει επεκταθεί, επιλέγεται για επέκταση. Αυτό επιτυγχάνεται με τη χρήση της δομής δεδομένων Ουράς (First In First Out Queue), για την παρουσίαση του χώρου αναζήτησης (frontier). Έτσι, οι νέοι κόμβοι τοποθετούνται στο πίσω μέρος της ουράς (tail) και οι παλιοί κόμβοι, οι οποίοι βρίσκονται σε πιο αρχικό επίπεδο από τους νέους κόμβους, κινούνται προς την κεφαλή (head) της ουράς και επεκτείνονται πρώτοι. Ο γενικός κανόνας του αλγορίθμου αναζήτησης Κατά Πλάτος (BFS) είναι ότι, ποτέ δεν επεκτείνεται μία κατάσταση αν δεν επεκταθούν πρώτα όλες οι καταστάσεις που βρίσκονται σε μικρότερο βάθος. Παρακάτω παρουσιάζεται και ο ψευδοκώδικας (pseudo code) του αλγορίθμου [13]:

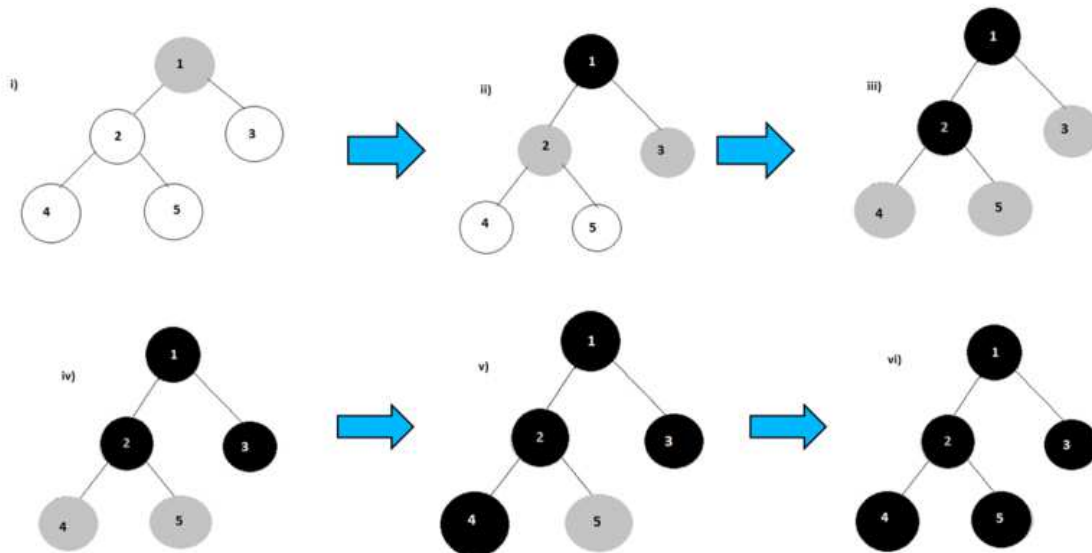
```

algorithm bfs (InitialState, FinalStates)
begin
  Closed ← ∅;
  Frontier ← <InitialState>;
  CurrentState ← First (Frontier);
  while CurrentState ∉ FinalStates do
    Frontier ← delete (CurrentState, Frontier);
    if CurrentState ∉ ClosedSet
    begin
      ChildrenStates ← Expand (CurrentState);
      Frontier ← Frontier ^ ChildrenStates;
      Closed ← Closed ∪ {CurrentState};
    end;
    if Frontier = ∅ then exit;
    CurrentState ← First (Frontier);
  endwhile;
  return success;
end.

```

Σχήμα 2.6: Ο ψευδοκώδικας του αλγορίθμου τυφλής αναζήτησης Πρώτα Σε Πλάτος (BFS) [13]

Το μεγάλο πλεονέκτημα του αλγορίθμου τυφλής αναζήτησης Κατά Πλάτος (BFS) είναι ότι μπορεί να χαρακτηριστεί πλήρης (complete), αφού αν ο πιο "ρηχός" κόμβος στόχος (goal) βρίσκεται σε κάποιο πεπερασμένο βάθος d , τότε ο αλγόριθμος θα βρει την λύση, εφόσον αυτή υπάρχει. Επίσης, μπορεί να χαρακτηριστεί σαν βέλτιστος (optimal), δηλαδή βρίσκει τη λύση με το μικρότερο μήκος, εφόσον όλοι οι τελεστές μετάβασης (ενέργειες - actions) έχουν το ίδιο κόστος [06]. Στο παρακάτω σχήμα, παρουσιάζεται η σειρά διάσχισης σε ένα απλό γράφο αναζήτησης, σύμφωνα με τον αλγόριθμο τυφλής αναζήτησης Κατά Πλάτος (BFS):



Σχήμα 2.7: Η εφαρμογή του αλγορίθμου Πρώτα σε Πλάτος σε ένα γράφο αναζήτησης

Το μεγαλύτερο μειονέκτημα του συγκεκριμένου αλγορίθμου είναι οι απαιτήσεις του τόσο στο χρόνο υλοποίησης του, όσο και στις απαιτήσεις που έχει σε μνήμη αποθήκευσης. Συγκεκριμένα, η ρίζα του γράφου επεκτείνεται σε b κόμβους στο πρώτο επίπεδο, ο καθένας από τους οποίους επεκτείνονται με τη σειρά τους σε b κόμβους κοκ. Αν η λύση βρίσκεται σε βάθος d , τότε στην χειρότερη περίπτωση ο συνολικός αριθμός των κόμβων που θα εξεταστούν θα είναι $O(b^d)$ [13], ισοδυναμεί με την πολυπλοκότητα του χρόνου (time complexity). Σχετικά με την πολυπλοκότητα χώρου (space complexity), αποθηκεύονται όλοι οι κόμβοι που έχουν ανακαλυφθεί στην μνήμη, οπότε απαιτούνται μεγάλες ποσότητες μνήμης αποθήκευσης και η συγκεκριμένη πολυπλοκότητα βρίσκεται πάντα σε συνάρτηση με τον παράγοντα b της πολυπλοκότητας χρόνου. Έτσι, η πολυπλοκότητα χώρου ισοδυναμεί με τον αριθμό των κόμβων που βρίσκονται στο χώρο αναζήτησης, δηλαδή είναι ίση $O(b^d)$ [13]. Τέλος, σε ένα μεγάλο πρόβλημα, όταν δηλαδή το δένδρο αναζήτησης έχει μεγάλο πλάτος και μεγαλώνει το μέτωπο αναζήτησης πολύ σε μέγεθος, τότε ο αλγόριθμος Κατά Πλάτος (BFS) έχει πρόβλημα στην ανάπτυξη του δένδρου αναζήτησης, ενώ παρουσιάζει εκθετική πολυπλοκότητα (exponential complexity) [01], έχει δηλαδή μεγάλες απαιτήσεις σε μνήμη και χρειάζεται μεγάλο χρόνο εκτέλεσης.

Αναζήτηση Πρώτα Σε Βάθος (Depth - First - Search)

Όπως φανερώνει η ονομασία του, ο αλγόριθμος τυφλής αναζήτησης Πρώτα σε Βάθος (Depth - First - Search, DFS), επεκτείνει πάντα τον "βαθύτερο" κόμβο του μετώπου αναζήτησης. Η αναζήτηση εξελίσσεται κατευθείαν στο βαθύτερο επίπεδο του δένδρου αναζήτησης, όπου οι κόμβοι δεν έχουν απογόνους. Καθώς αυτοί οι κόμβοι επεκτείνονται, διαγράφονται από το μέτωπο αναζήτησης, οπότε η αναζήτηση επιστρέφει στον επόμενο "βαθύτερο κόμβο", ο οποίος έχει ανεξερεύνητους απογόνους [01]. Στην περίπτωση που υπάρχουν περισσότεροι από έναν κόμβοι (καταστάσεις) στο ίδιο βάθος, τότε ο αλγόριθμος Πρώτα σε Βάθος (DFS) επιλέγει έναν από αυτούς και για ευκολία επιλέγεται ο αριστερότερος. Η περιγραφή του αλγορίθμου είναι:

1. Βάλε την αρχική κατάσταση στο μέτωπο αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι κενό τότε σταμάτησε.
3. Βγάλε την πρώτη κατάσταση από το μέτωπο αναζήτησης.
4. Αν η κατάσταση ανήκει στο κλειστό σύνολο τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι μία από τις τελικές, τότε ανέφερε τη λύση.
6. Αν θέλεις και άλλες λύσεις πήγαινε στο βήμα 2. Αλλιώς σταμάτησε.
7. Εφάρμοσε του τελεστές μετάβασης για να βρεις τις καταστάσεις - παιδιά.
8. Βάλε τις καταστάσεις - παιδιά στην αρχή του μετώπου αναζήτησης.
9. Βάλε την κατάσταση - γονέα στο κλειστό σύνολο.
10. Πήγαινε στο βήμα 2.

Η διαφορά του αλγορίθμου Πρώτα σε Βάθος (DFS) από τον αλγόριθμο Πρώτα σε Πλάτος (BFS) εντοπίζεται στον τρόπο υλοποίησης του μετώπου αναζήτησης. Στον αλγόριθμο Πρώτα σε Βάθος (DFS) ο πιο "βαθύς" κόμβος που δεν έχει επεκταθεί, επιλέγεται για επέκταση. Αυτό επιτυγχάνεται με τη χρήση της δομής δεδομένων Στοίβας (Last In First Out Stack), για την παρουσίαση του χώρου αναζήτησης (frontier). Έτσι, ο πιο πρόσφατος παραγόμενος κόμβος επιλέγεται να είναι επόμενος προς επέκταση. Αυτός πρέπει να είναι ο βαθύτερος κόμβος που δεν έχει επεκταθεί, διότι βρίσκεται σε μεγαλύτερο βάθος από τον γονέα του - ο οποίος ήταν με τη σειρά του ο "βαθύτερος κόμβος" όταν επιλέχθηκε. Η εξέταση αμέσως προηγούμενων χρονικά καταστάσεων ονομάζεται χρονική οπισθοδρόμηση (chronological backtracking) [13]. Αν όλες οι πιθανές επιλογές από ένα συγκεκριμένο σημείο έχουν ήδη εξεταστεί, η οπισθοδρόμηση συνεχίζεται στο αμέσως πιο επάνω σημείο κοκ. Παρακάτω παρουσιάζεται και ο ψευδοκώδικας (pseudo code) του αλγορίθμου [13]:

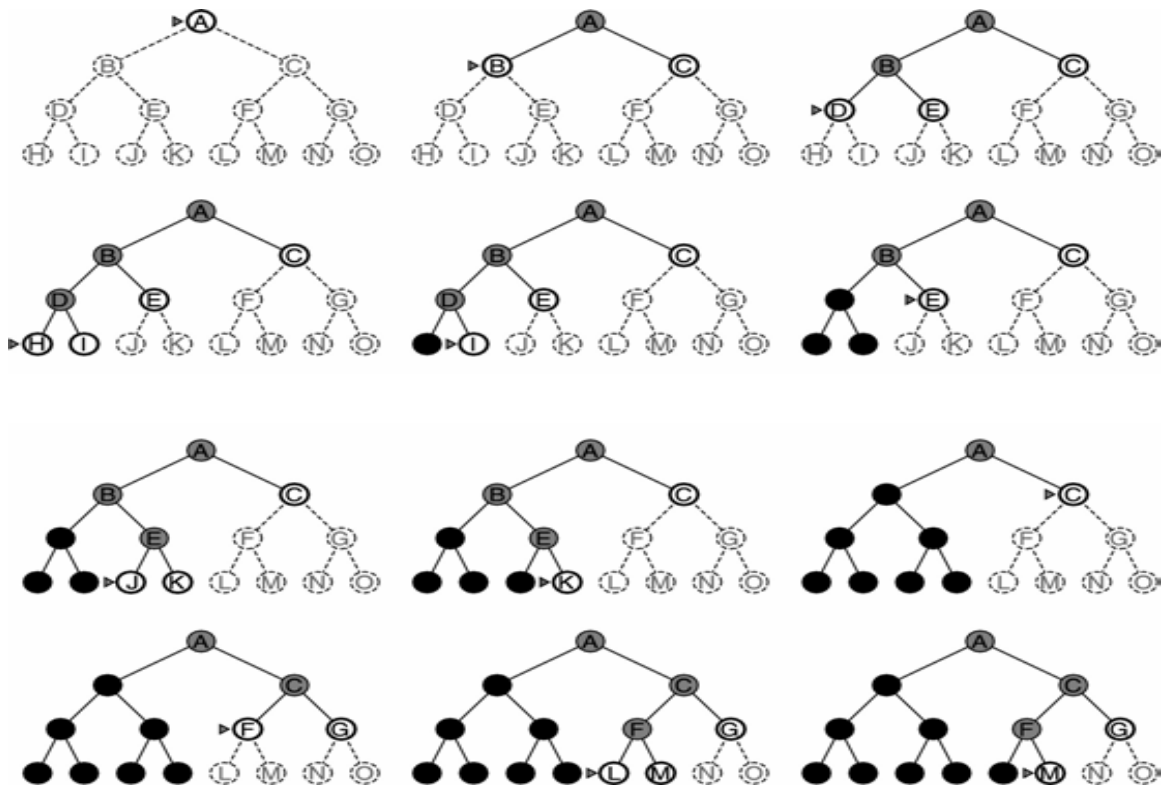
```

algorithm dfs(InitialState, FinalStates)
begin
  Closed $\leftarrow\emptyset$ ;
  Frontier $\leftarrow\langle$ InitialState $\rangle$ ;
  CurrentState $\leftarrow$ First(Frontier);
  while CurrentState  $\notin$  FinalStates do
    Frontier $\leftarrow$ delete(CurrentState, Frontier);
    if CurrentState  $\notin$  ClosedSet then
      begin
        ChildrenStates  $\leftarrow$ Expand(CurrentState);
        Frontier $\leftarrow$ ChildrenStates  $\wedge$  Frontier;
        Closed $\leftarrow$ Closed $\cup$ {CurrentState};
      end;
    if Frontier= $\emptyset$  then exit;
    CurrentState $\leftarrow$ First(Frontier);
  endwhile;
return success;
end.

```

Σχήμα 2.8: Ο ψευδοκώδικας του αλγορίθμου τυφλής αναζήτησης Πρώτα σε Βάθος (DFS) [13]

Οι ιδιότητες του συγκεκριμένου αλγορίθμου εξαρτιόνται σε μεγάλο βαθμό από το αν η αναζήτηση πραγματοποιείται σε γράφο ή σε δένδρο. Η αναζήτηση σε γράφο, αποφεύγει τις επαναλαμβανόμενες καταστάσεις, οπότε μπορεί να χαρακτηριστεί ως πλήρης για πεπερασμένο αριθμό καταστάσεων, διότι επεκτείνονται όλοι οι κόμβοι και χρησιμοποιείται το κλειστό σύνολο. Από την άλλη πλευρά, η αναζήτηση σε δένδρο δεν είναι πλήρης, διότι καταλήγει να ελέγχει νέες καταστάσεις ενάντια σε εκείνες που βρίσκονται στο μονοπάτι, από τη ρίζα μέχρι τον τρέχοντα κόμβο, οπότε μπορεί να μπλεχτεί σε κλαδιά μεγάλου μήκους ή σε ατέρμονα κλαδιά. Επιπλέον, και οι δύο αναζητήσεις αποτυγχάνουν να βρουν λύση, όταν παρατηρείται η ύπαρξη άπειρου χώρου καταστάσεων. Για τους παραπάνω παρόμοιους λόγους, θεωρείται μη βέλτιστος (non optimal), δεν εγγυάται δηλαδή ότι η πρώτη λύση που θα βρεθεί είναι η βέλτιστη (μονοπάτι με το μικρότερο μήκος ή με το μικρότερο κόστος). Στο παρακάτω σχήμα παρουσιάζεται η σειρά με την οποία επισκέπτεται τους κόμβους του γράφου αναζήτησης, ο αλγόριθμος Πρώτα σε Βάθος (Depth First Search):



Σχήμα 2.9: Η εφαρμογή του αλγορίθμου Πρώτα σε Βάθος σε ένα γράφο αναζήτησης

Η πολυπλοκότητα χρόνου του αλγορίθμου Πρώτα σε Βάθος σε γράφο εξαρτάται από το μέγεθος του χώρου καταστάσεων. Αντίστοιχα, ο αλγόριθμος Πρώτα σε Βάθος σε ένα δένδρο αναζήτησης παρουσιάζει πολυπλοκότητα χρόνου $O(b^m)$ [06], όπου m είναι το μέγιστο βάθος σε κάθε κόμβο, Αυτό μπορεί να σημαίνει ότι είναι μεγαλύτερο από το μέγεθος του χώρου καταστάσεων. Το μεγάλο πλεονέκτημα του συγκεκριμένου αλγορίθμου είναι η πολυπλοκότητα χώρου. Το πλεονέκτημα αυτό δεν εμφανίζεται σε μία αναζήτηση γράφου, αλλά στην δενδρική αναζήτηση. Έχει μικρές απαιτήσεις σε χώρο διότι το μέτωπο αναζήτησης δεν μεγαλώνει πάρα πολύ. Χρειάζεται να αποθηκεύσει μόνο μία απλή διαδρομή από τον κόμβο ρίζα στον κόμβο φύλλο. Όταν ένας κόμβος έχει επεκταθεί μία φορά, μπορεί να αφαιρεθεί από τη μνήμη όταν όλοι οι απόγονοι του έχουν εξερευνηθεί. Για μία κατάσταση χώρου με βαθμό διακλάδωσης b και μέγιστο βάθος m , ο αλγόριθμος Πρώτα σε Βάθος απαιτεί χώρο αποθήκευσης $O(bm)$ [06]. Ανάλογα η αναζήτηση Πρώτα σε Βάθος με οπισθοδρόμηση θα παράγει έναν απόγονο την φορά, οπότε απαιτεί μνήμη που ισοδυναμεί με $O(m)$. Αυτή είναι η βασικότερη αιτία, που καθιστά τον αλγόριθμο Πρώτα σε Βάθος (DFS) ως μία από τις βασικότερες υλοποιήσεις προβλημάτων αναζήτησης, σε διάφορους τομείς της Τεχνητής Νοημοσύνης (AI), όπως είναι η ικανοποίηση των περιορισμών, η προτασιακή λογική και ο λογικός προγραμματισμός.

2.1.4 Αλγόριθμοι Ευριστικής Αναζήτησης

Οι αλγόριθμοι τυφλής αναζήτησης προχωρούν κατά πλάτος ή κατά βάθος χωρίς να έχουν καμία ένδειξη για το αν πλησιάζουν στην τελική κατάσταση. Επομένως, η τυφλή αναζήτηση δεν επαρκεί για προβλήματα συνδυαστικής έκρηξης, όπου δηλαδή ο χώρος αναζήτησης αυξάνεται ραγδαία και κάνει πρακτικά δύσκολη την εύρεση λύσης. Σε τέτοιες περιπτώσεις χρησιμοποιούνται οι αλγόριθμοι ευριστικής αναζήτησης, όπου σκοπός είναι να μειωθεί ο χρόνος αναζήτησης, να μειωθεί δηλαδή ο αριθμός των καταστάσεων που θα εξετάσει ο αλγόριθμος. Για να επιτευχθεί αυτό απαιτείται η εύρεση πληροφορίας για την αξιολόγηση των καταστάσεων, που θα οδηγήσει γρηγορότερα στη λύση ή θα οδηγήσει σε κλάδεμα (pruning) των καταστάσεων που δεν οδηγούν πουθενά. Στις προκειμένες περιπτώσεις, γίνεται χρήση αλγορίθμων που εκμεταλλεύονται τους ευρετικούς μηχανισμούς (heuristic), δηλαδή τις στρατηγικές που βασίζονται στη γνώση για το συγκεκριμένο πρόβλημα, αξιολογούν τις διάφορες καταστάσεις και χρησιμοποιούνται σαν βοήθημα στη γρήγορη επίλυσή του [03]. Ο ευριστικός μηχανισμός υλοποιείται με ευριστική συνάρτηση (heuristic function), έχει πεδίο ορισμού το σύνολο των καταστάσεων ενός προβλήματος και πεδίο τιμών που αντιστοιχεί σε αυτές. Η ευριστική συνάρτηση $h(n)$ είναι το εκτιμώμενο κόστος της πιο "φτηνής" διαδρομής από την κατάσταση στο τρέχοντα κόμβο n προς την τελική κατάσταση (goal state). Συνεπώς, ο ευρετικός μηχανισμός αξιολογεί τις ενδιάμεσες καταστάσεις ως προς την εκτιμώμενη απόστασή τους από μία τελική κατάσταση, επεκτείνουν πρώτα αυτές με τη βέλτιστη ευρετική τιμή (heuristic value) [06]. Η ευρετική τιμή δεν είναι η πραγματική τιμή της απόστασης από μία τελική κατάσταση, αλλά μία εκτίμηση (estimation) που πολλές φορές μπορεί να είναι και λανθασμένη. Στην πλειονότητα των περιπτώσεων πάντως οι ευρετικές στρατηγικές οδηγούν σε πολύ καλά αποτελέσματα (ανάλογα τη φύση του προβλήματος), προσαρμόζονται στις ανάγκες της αναζήτησης και χρησιμοποιούνται για την προσομοίωση της ανθρώπινης σκέψης και συμπεριφοράς.

Άπληστη Αναζήτηση Πρώτα Στο Καλύτερο (Greedy - Best - First - Search)

Η Άπληστη Αναζήτηση Πρώτα Στο Καλύτερο (Greedy - Best - First - Search, GBFS), κρατάει όλες τις καταστάσεις στο μέτωπο αναζήτησης και προσπαθεί να επεκτείνει τον κόμβο που βρίσκεται πιο κοντά στην τελική κατάσταση, οδηγεί δηλαδή στην γρηγορότερη εύρεση λύσης. Αξιολογεί τους κόμβους του γράφου ή του δένδρου αναζήτησης, κάνοντας χρήση της ευριστικής συνάρτησης όπου $f(n) = h(n)$ [03]. Ο αλγόριθμος καλείται "άπληστος", διότι προσπαθεί πάντα

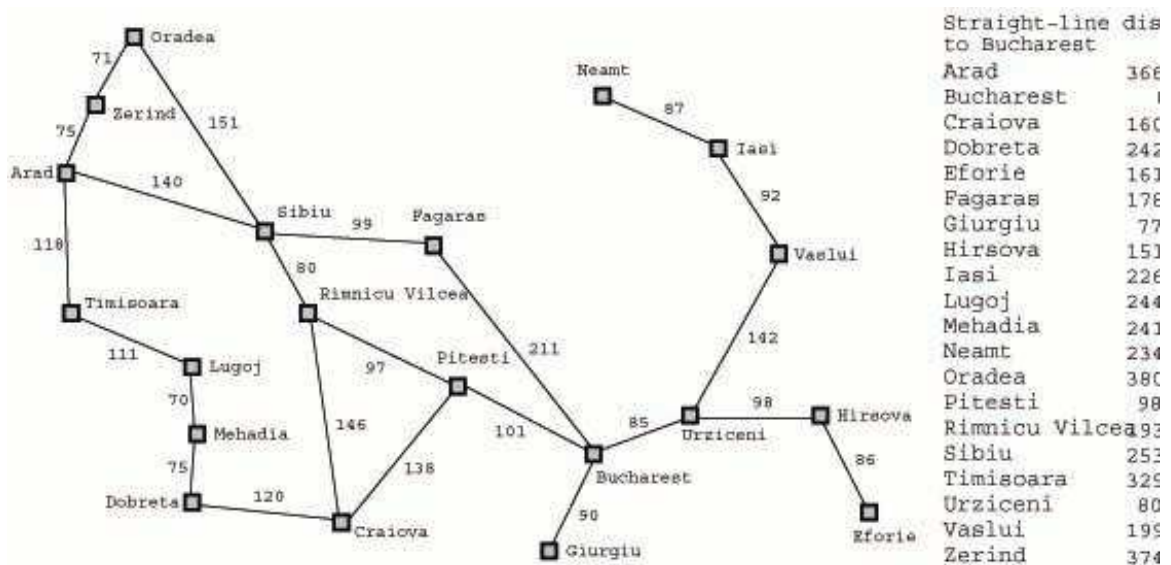
να μειώσει όσο το δυνατόν περισσότερο το κόστος που απομένει, μέχρι να φτάσουμε στην κατάσταση στόχος. Η περιγραφή και ο ψευδοκώδικας του είναι [13]:

1. Βάλε την αρχική κατάσταση στο μέτωπο αναζήτησης.
2. Αν το μέτωπο αναζήτησης είναι κενό τότε σταμάτησε.
3. Πάρε την πρώτη σε σειρά κατάσταση από το μέτωπο αναζήτησης.
4. Αν η κατάσταση είναι μέλος του κλειστού συνόλου τότε πήγαινε στο βήμα 2.
5. Αν η κατάσταση είναι μία τελική τότε ανέφερε τη λύση και σταμάτα.
6. Εφάρμοσε τους τελεστές μεταφοράς για την παραγωγή των καταστάσεων - παιδιά.
7. Εφάρμοσε την ευριστική συνάρτηση σε κάθε παιδί.
8. Βάλε τις καταστάσεις - παιδιά στο μέτωπο αναζήτησης.
9. Αναδιάταξε το μέτωπο αναζήτησης, έτσι ώστε η κατάσταση με την καλύτερη ευριστική τιμή να είναι πρώτη.
10. Βάλε την κατάσταση - γονέα στο κλειστό σύνολο.
11. Πήγαινε στο βήμα 2.

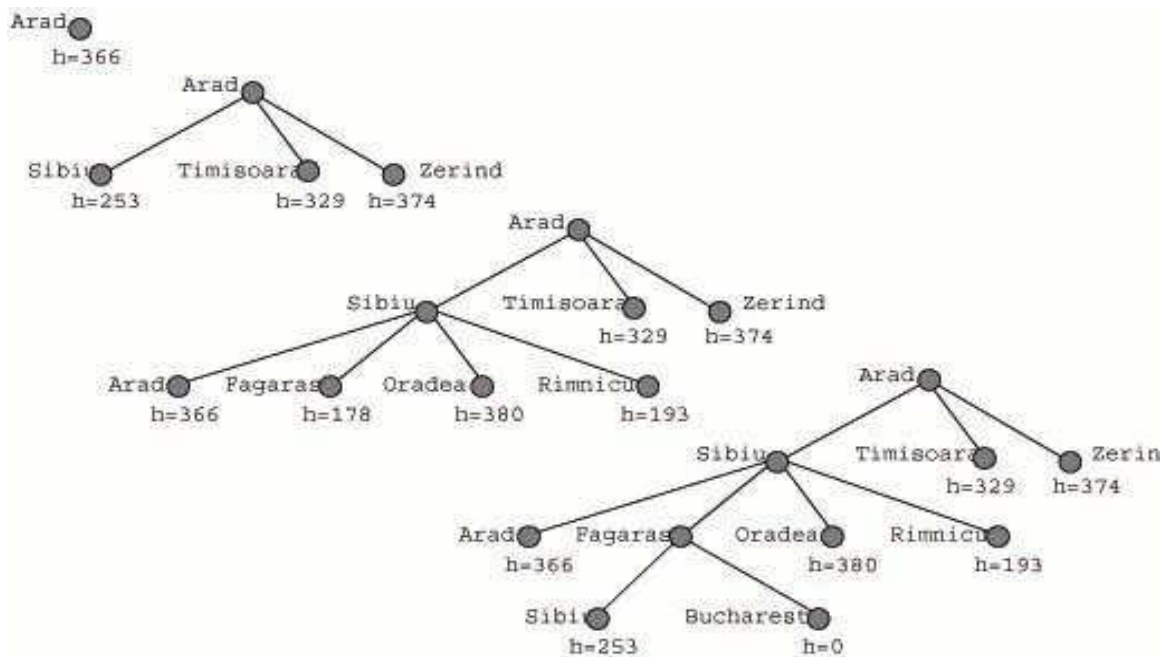
```
algorithm bestfs (InitialState, FinalStates)
begin
  Closed ← ∅;
  EvaluatedInitialState ← Heuristic (<InitialState>)
  Frontier ← <EvaluatedInitialState>;
  CurrentState ← best (Frontier);
  while CurrentState ∉ FinalStates do
    Frontier ← delete (CurrentState, Frontier);
    if CurrentState ∉ ClosedSet then
      begin
        Children ← Expand (CurrentState);
        EvaluatedChildren ← Heuristic (Children);
        Frontier ← Frontier ^ EvaluatedChildren;
        Closed ← Closed ∪ {CurrentState};
      end;
    if Frontier = ∅ then return fail;
    CurrentState ← best (Frontier);
  endwhile;
  return success;
end.
```

Σχήμα 2.10: Ο ψευδοκώδικας του αλγορίθμου άπληστης αναζήτησης Πρώτα στο Καλύτερο (GBFS) [13]

Το μεγαλύτερο πλεονέκτημα του αλγορίθμου άπληστης αναζήτησης Πρώτα στο Καλύτερο (GBFS), είναι ότι προσπαθεί να δώσει μία γρήγορη λύση σε κάποιο πρόβλημα. Η επιτυχία της συγκεκριμένης επιλογής, εξαρτάται σε μεγάλο βαθμό από την εγκυρότητα του ευριστικού μηχανισμού. Ο συγκεκριμένος αλγόριθμος είναι πλήρης σε αναζήτηση γράφου για πεπερασμένο αριθμό καταστάσεων, ενώ χαρακτηρίζεται μη πλήρης σε περιπτώσεις δενδρικής αναζήτησης, τόσο για άπειρο όσο και για πεπερασμένο μέγεθος χώρου καταστάσεων. Στο παρακάτω σχήμα παρουσιάζεται η εφαρμογή του αλγορίθμου Πρώτα στο Καλύτερο (GBFS), στο πρόβλημα αναζήτησης διαδρομής για τη χώρα της Ρουμανίας. Γίνεται χρήση της ευριστικής ευθείας γραμμής (straight line distance), η οποία καλείται h_{SLD} . Η αρχική κατάσταση είναι η πόλη Αράντ (Arad) και η τελική κατάσταση η πόλη του Βουκουρεστίου (Bucharest).



Σχήμα 2.11: Η παρουσίαση των αποστάσεων από κάθε πόλη προς τον Βουκουρέστι (τελικός στόχος)



Σχήμα 2.12: Η εφαρμογή της άπληστης αναζήτησης Πρώτα στο Καλύτερο (GBFS), για το πρόβλημα της εύρεσης της καλύτερης διαδρομής

Παρόλα αυτά στο συγκεκριμένο αλγόριθμο, το μέτωπο αναζήτησης μεγαλώνει με υψηλό ρυθμό και μαζί του και ο χώρος που χρειάζεται για την αποθήκευσή του, καθώς και ο χρόνος για την επεξεργασία των στοιχείων του. Ο αλγόριθμος της άπληστης αναζήτησης Πρώτα στο Καλύτερο (GBFS) δεν εγγυάται ότι η λύση που θα βρει θα είναι βέλτιστη, ενώ στην χειρότερη περίπτωση η πολυπλοκότητα χρόνου και χώρου είναι ίση $O(b^m)$ [06], όπου m είναι το μέγιστο βάθος του χώρου αναζήτησης. Ωστόσο, η επιλογή μίας καλής και ποιοτικής ευριστικής συνάρτησης μπορεί να μειώσει αισθητά την πολυπλοκότητα του προβλήματος.

Αναζήτηση Άλφα Άστρο (A*)

Ο A* είναι ο πιο γνωστός Πρώτα στο Καλύτερο αλγόριθμος και από τους πιο ευρέως χρησιμοποιούμενους στην Τεχνητή Νοημοσύνη, για προβλήματα αναζήτησης μονοπατιού και διάσχισης γράφου. Συνδυάζει τις λειτουργίες της άπληστης αναζήτησης και αυτής με βάση το κόστος, ενώ επιλύει προβλήματα επιλογής καταστάσεων με την ίδια τιμή στην ευριστική τους συνάρτηση, καθώς και για τον περιορισμό των βημάτων προς τον τελικό στόχο του προβλήματος. Γι' αυτό το λόγο λαμβάνεται υπόψη όχι μόνο η απόσταση του κόμβου από την τελική κατάσταση αλλά και η απόσταση του κόμβου από την αρχική κατάσταση. Πραγματοποιεί

μία αξιολόγηση των κόμβων με βάση την ευριστική συνάρτηση $f(n) = g(n) + h(n)$ [03]. Η συνάρτηση $g(n)$ αντιπροσωπεύει την απόσταση του κόμβου n από την αρχική κατάσταση, η οποία είναι πραγματική και γνωστή και η $h(n)$ δείχνει την εκτίμηση της απόστασης του n από την τελική κατάσταση μέσω μίας ευριστικής συνάρτησης, όπως ακριβώς στον GBFS. Η περιγραφή και ο ψευδοκώδικας του είναι [13]:

1. Δημιούργησε ουρά για τα δεδομένα προς επεξεργασία. Αρχικά έχει την ρίζα.
2. Αν η ουρά είναι κενή σταμάτησε.
3. Βρες το στοιχείο της ουράς με το μικρότερο κόστος.
4. Εκχώρησε το στοιχείο σε προσωρινή μεταβλητή.
5. Σβήσε από την ουρά το στοιχείο με το μικρότερο κόστος.
6. Αν είναι η προσωρινή μεταβλητή σταμάτα.
7. Αλλιώς δημιούργησε τα παιδιά της προσωρινής μεταβλητής και τοποθέτησε τα στην αρχή της ουράς.
8. Πήγαινε στο βήμα 2.

```
// A*
initialize the open list
initialize the closed list
put the starting node on the open list (you can leave its f at zero)

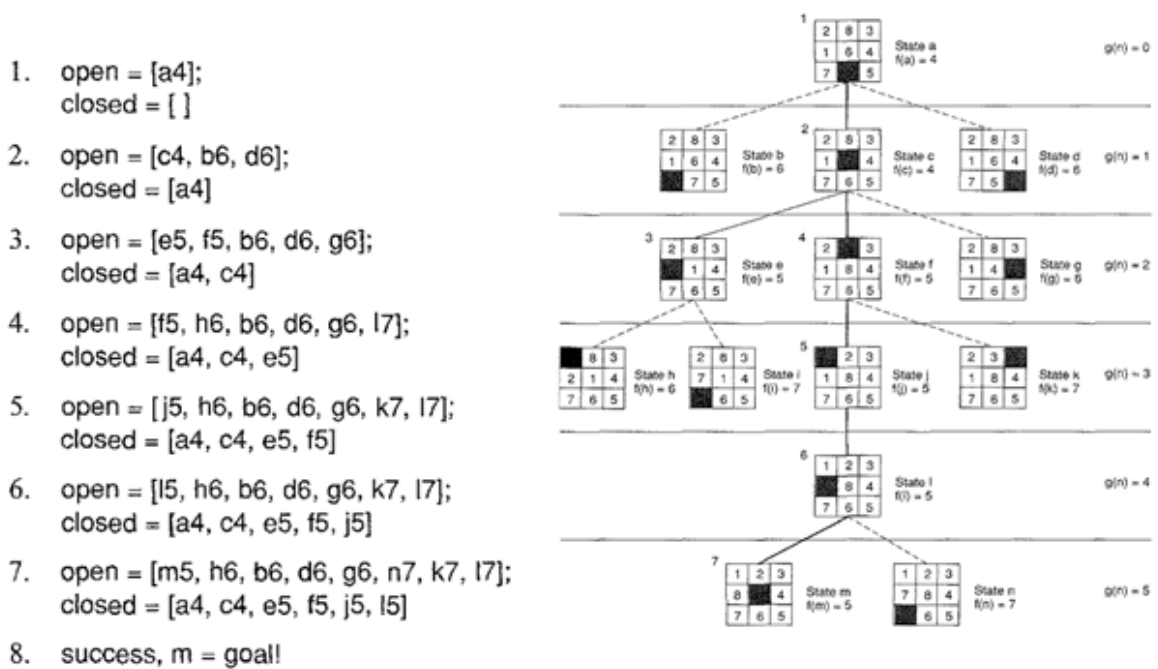
while the open list is not empty
  find the node with the least f on the open list, call it "q"
  pop q off the open list
  generate q's 8 successors and set their parents to q
  for each successor
    if successor is the goal, stop the search
    successor.g = q.g + distance between successor and q
    successor.h = distance from goal to successor
    successor.f = successor.g + successor.h

    if a node with the same position as successor is in the OPEN list \
      which has a lower f than successor, skip this successor
    if a node with the same position as successor is in the CLOSED list \
      which has a lower f than successor, skip this successor
    otherwise, add the node to the open list
  end
  push q on the closed list
end
```

Σχήμα 2.13: Ο ψευδοκώδικας του αλγορίθμου A* [13]

Στον A* αν για κάθε κατάσταση η τιμή $h(n)$ είναι μικρότερη ή το πολύ ίση με την πραγματική απόσταση του n από την τελική κατάσταση, τότε ο A* βρίσκει πάντα την βέλτιστη λύση. Στην περίπτωση αυτή, ο ευριστικός μηχανισμός ονομάζεται αποδεκτός

(admissible) και ικανοποιεί το κριτήριο της αποδοχής [13]. Γενικά, για την εύρεση της βέλτιστης λύσης πρέπει η ευριστική συνάρτηση $h(n)$ να είναι συνεπής (consistent). Η $h(n)$ ονομάζεται συνεπής αν και μόνο αν για όλους τους κόμβους n, n' , όπου ο n' είναι απόγονος του n με μια πράξη a , το εκτιμώμενο κόστος για να φτάσουμε το στόχο από τον κόμβο n δεν είναι μεγαλύτερο από το κόστος να φτάσουμε το n' συν το εκτιμώμενο κόστος να φτάσουμε το στόχο από τον κόμβο n' δηλαδή, $h(n) \leq c(n,a,n') + h(n')$ [03]. Η αναζήτηση A^* σε δένδρο είναι βέλτιστη αν η $h(n)$ είναι αποδεκτή (admissible) και στην αναζήτηση γράφου είναι βέλτιστη αν η $h(n)$ είναι συνεπής (consistent). Ο A^* επίσης είναι πλήρης δηλαδή, θα επισκεφτεί τους λιγότερους κόμβους, θα τελειώσει και θα βρει τη βέλτιστη λύση ακόμα και σε άπειρους χώρους αναζήτησης, αν υπάρχει βέλτιστη λύση και όλα τα κόστη είναι θετικά. Αυτό σημαίνει ότι θα επεκτείνει όλους τους κόμβους με $f(n) < C^*$ (C^* είναι το κόστος της βέλτιστης λύσης) ή μπορεί να επισκεφτεί κάποιους από τους κόμβους για τους οποίους ισχύει $f(n) = C^*$ με σκοπό να βρει την τελική λύση. Σε καμία των περιπτώσεων δεν επισκέπτεται κόμβους με κόστος $f(n) > C^*$. Στο παρακάτω σχήμα παρουσιάζεται μία απλή υλοποίηση του αλγορίθμου A^* :



Σχήμα 2.14: Η υλοποίηση του αλγορίθμου A^* σε ένα παιχνίδι 8 - puzzle

Παρόλα τα πλεονεκτήματα η αναζήτηση A^* δεν σημαίνει ότι είναι η απάντηση για όλα τα προβλήματα αναζήτησης, διότι σε ορισμένα ο αριθμός των κόμβων μέσα στο χώρο

αναζήτησης εξακολουθεί να αυξάνεται εκθετικά με το μήκος της λύσης. Έτσι, η πολυπλοκότητα του χρόνου είναι εκθετική εκτός και αν το σφάλμα της ευριστικής συνάρτησης δεν αυξάνεται γρηγορότερα από το λογάριθμο του πραγματικού κόστους διαδρομής. Για όλους σχεδόν τους ευρεστικούς μηχανισμούς που χρησιμοποιούνται στην πράξη, το σφάλμα είναι τουλάχιστον ανάλογο με το κόστος διαδρομής και η εκθετική αύξηση που προκύπτει τελικά ξεπερνά τις δυνατότητες οποιουδήποτε υπολογιστή. Επιπλέον, η πολυπλοκότητα χώρου του αλγορίθμου A^* ισοδυναμεί με $O(b^d)$ αφού, διατηρεί στην μνήμη όλους τους κόμβους που παράγονται, εξαντλεί πολύ συχνά τον χώρο που έχουμε στην διάθεση μας [06]. Για αυτόν τον λόγο παρουσιάζει αδυναμίες σε προβλήματα μεγάλης κλίμακας. Σαν αποτέλεσμα, έχουν επινοηθεί νέοι αλγόριθμοι που ξεπερνούν το πρόβλημα του χώρου χωρίς να θυσιάσουν τη βέλτιστη συμπεριφορά ή την πληρότητα, με μικρό κόστος σε χρόνο εκτέλεσης.

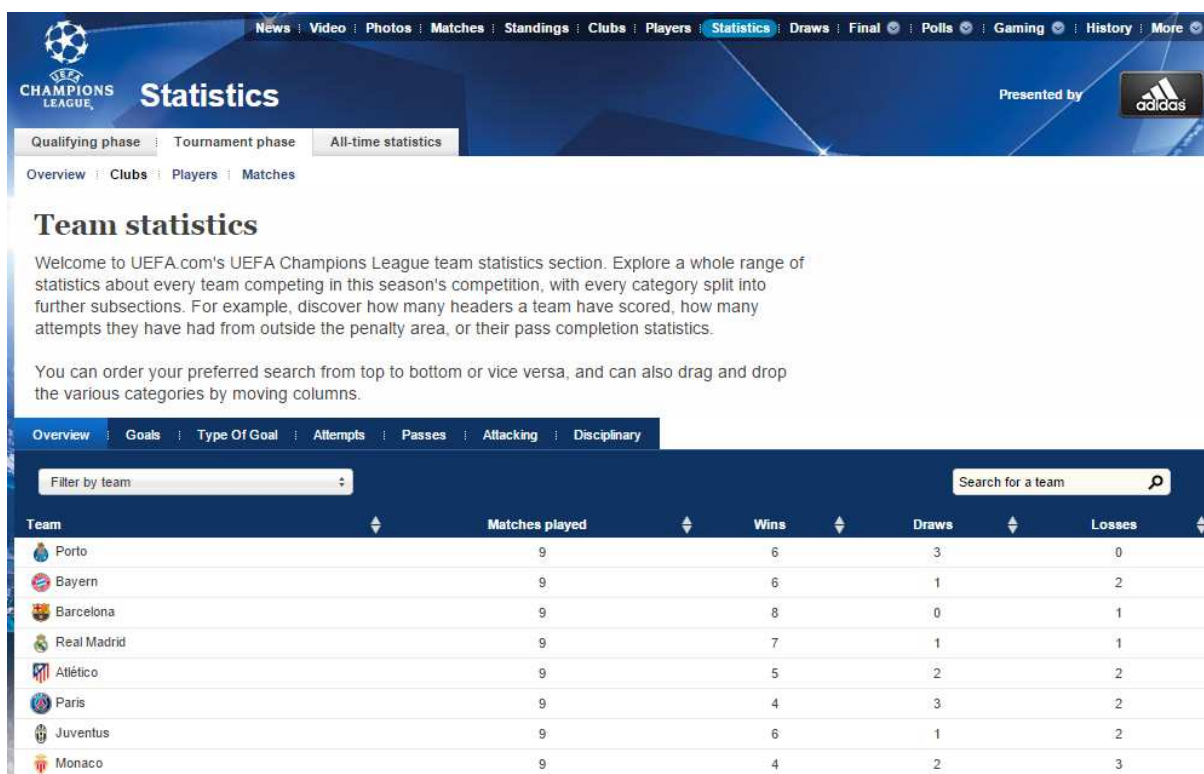
Στην πορεία της συγκεκριμένης μεταπτυχιακής διατριβής εκπονήθηκε άσκηση κατανόησης της λειτουργίας των αλγορίθμων Τεχνητής Νοημοσύνης, που αναλύθηκαν παραπάνω δηλαδή, της αναζήτησης Πρώτα σε Πλάτος, Πρώτα σε Βάθος, Άπληστης Πρώτα στο Καλύτερο και του A^* . Συγκεκριμένα, δόθηκε γράφος και τα αντίστοιχα κόστη αποστάσεων μεταξύ των κόμβων του γράφου. Στη συνέχεια, ζητήθηκε να αποτυπωθεί η σειρά επίσκεψης των κόμβων για κάθε αλγόριθμο ξεχωριστά. Τέλος, έγινε η παρουσίαση των συγκεκριμένων αποτελεσμάτων.

2.2 Ιστοσελίδες Ποδοσφαιρικών Στατιστικών

Για τις ανάγκες της μεταπτυχιακής διατριβής και την επίτευξη του στόχου πρόβλεψης του νικητή του Champions League, απαιτείται η εύρεση και η συλλογή στατιστικών δεδομένων που αφορούν τους κορυφαίους ποδοσφαιρικούς συλλόγους, οι οποίοι αγωνίζονται στη συγκεκριμένη Ευρωπαϊκή Διασυλλογική Διοργάνωση. Για την επίτευξη της ανάγκης αυτής, συμβάλλει σημαντικά η ύπαρξη στο Διαδίκτυο, πλειάδας ιστοσελίδων που αναλύουν και επεξεργάζονται αριθμητικά δεδομένα αγωνιστικής απόδοσης διαφόρων ποδοσφαιρικών ομάδων. Παρακάτω, παρουσιάζονται οι ιστοσελίδες που χρησιμοποιήθηκαν για την συλλογή, την επεξεργασία και την ανάλυση των δεδομένων, στα οποία βασίστηκε η περαιτέρω εξαγωγή αποτελεσμάτων της μεταπτυχιακής διατριβής.

2.2.1 Η Ιστοσελίδα UEFA.com

Η συγκεκριμένη ιστοσελίδα ανήκει στη διοργανώτρια αρχή του Champions League, την UEFA [11]. Μπαίνοντας ο χρήστης μπορεί να παρακολουθήσει διάφορες πληροφορίες και να επιλέξει την ενημέρωση του από ένα μεγάλο πλήθος συνδέσμων. Συγκεκριμένα, έχει τη δυνατότητα να ενημερωθεί για τα τελευταία νέα των ομάδων που αγωνίζονται στις ποδοσφαιρικές διοργανώσεις της UEFA, να διαβάσει συνεντεύξεις των πρωταγωνιστών (παιχτών - προπονητών), να παρακολουθήσει την πορεία των αγώνων που βρίσκονται σε εξέλιξη, να δει κάποια βίντεο αγώνων που έχουν διεξαχθεί, να μάθει για τα τελευταία νέα που αφορούν την διοργανώτρια αρχή κ.α. Για τις ανάγκες της μεταπτυχιακής διατριβής, είναι σκόπιμο να βρεθεί το τμήμα που περιέχει τα στατιστικά των ποδοσφαιρικών ομάδων που αγωνίζονται στο Champions League.



Team statistics

Welcome to UEFA.com's UEFA Champions League team statistics section. Explore a whole range of statistics about every team competing in this season's competition, with every category split into further subsections. For example, discover how many headers a team have scored, how many attempts they have had from outside the penalty area, or their pass completion statistics.

You can order your preferred search from top to bottom or vice versa, and can also drag and drop the various categories by moving columns.

Team	Matches played	Wins	Draws	Losses
Porto	9	6	3	0
Bayern	9	6	1	2
Barcelona	9	8	0	1
Real Madrid	9	7	1	1
Atlético	9	5	2	2
Paris	9	4	3	2
Juventus	9	6	1	2
Monaco	9	4	2	3

Σχήμα 2.15: Το τμήμα της ιστοσελίδας της UEFA, όπου περιέχει τα ομαδικά στατιστικά

Στο τμήμα των στατιστικών, ο χρήστης έχει την δυνατότητα να επιλέξει ανάμεσα σε δεδομένα που σχετίζονται με γενικά χαρακτηριστικά (Overview), ομαδικά (Clubs), ατομικά (Players), αυτά που αναφέρονται στα αποτελέσματα των αγώνων (Matches) και εκείνα που έχουν σχέση με ιστορικά δεδομένα αγώνων προηγούμενων ετών (All-time statistics). Στα γενικά χαρακτηριστικά, παρουσιάζονται οι ποδοσφαιριστές που

προηγούνται στις κυριότερες στατιστικές κατηγορίες. Στα ομαδικά δεδομένα, περιγράφονται οι συνολικές αποδόσεις των ομάδων που συμμετέχουν στην διοργάνωση, όπως τα επιθετικά τους επιτεύγματα (Goals, Type Of Goal, Attempts, Passes, Attacking), καθώς και η αμυντική τους συμπεριφορά (Disciplinary). Στα ατομικά στατιστικά παρουσιάζονται ακριβώς οι ίδιες κατηγορίες με τα ομαδικά, που αφορούν όμως τις επιδόσεις του κάθε ποδοσφαιριστή ξεχωριστά. Στο τμήμα των αγωνιστικών πληροφοριών παρατίθενται οι βαθμολογίες των ομάδων της διοργάνωσης, τα αποτελέσματα των αγώνων που έχουν διεξαχθεί, αλλά και ποιοι θα είναι οι μελλοντικοί αγώνες. Τέλος, στα ιστορικά δεδομένα περιγράφονται αποτελέσματα προηγούμενων ετών, κατάταξη ομάδων, γενικά χαρακτηριστικά των ομάδων που έχουν συμμετάσχει στο Champions League. Η προκειμένη ιστοσελίδα [11] αποτελεί μία πολύ καλή πηγή πληροφόρησης για τα δρώμενα στο χώρο του Ευρωπαϊκού ποδοσφαίρου και αποτελεί σημαντικό εργαλείο για την εύρεση και συλλογή στατιστικών ποδοσφαιρικών δεδομένων, για τη περαιτέρω επεξεργασία τους.

2.2.2 Η Ιστοσελίδα WhoScored.com

Η επόμενη ιστοσελίδα είναι η WhoScored.com [12], η οποία είναι δημιουργία αναλυτών ποδοσφαίρου και κατασκευαστών λογισμικού με υπόβαθρο και γνώση στο συγκεκριμένο άθλημα. Αποτελεί μοναδική ιστοσελίδα [12] στο είδος της και ειδικεύεται στην εις βάθος ανάλυση και λεπτομερή ανάλυση ποδοσφαιρικών δεδομένων. Ο χρήστης έχει τη δυνατότητα να παρακολουθήσει ζωντανή εξέλιξη αγώνων, να βρει και να μελετήσει μοναδικά στατιστικά των ευρωπαϊκών διοργανώσεων (Champions League, Europa League), των πέντε (5) κορυφαίων Ευρωπαϊκών πρωταθλημάτων (Αγγλικό, Γαλλικό, Γερμανικό, Ισπανικό, Ιταλικό) καθώς και άλλων υποδεέστερων διοργανώσεων, να παρακολουθήσει τις εβδομαδιαίες βαθμολογίες παιχτών και ομάδων, να μελετήσει πληροφορίες για τα παιχνίδια που θα διεξαχθούν (τραυματισμοί παιχτών, προβλέψεις ειδικών κα) και να διαβάσει άρθρα έγκριτων αναλυτών ποδοσφαίρου. Για τις ανάγκες μας μπορούμε να πλοηγηθούμε σε διάφορες ενδιαφέρουσες κατηγορίες της συγκεκριμένης ιστοσελίδας [12].

WhoScored.com
Revolutionising Football Statistics

[Home](#) [LIVE SCORES](#) [STATISTICS](#) [PREVIEWS](#) [ARTICLES](#) [INFOGRAPHICS](#) [VIDEOS](#) [NEWS](#) [Sign In](#) [Sign Up](#)

Browse: [My Favourites](#) [Detailed Tournaments](#) [All Leagues & Cups](#) [International Cups](#)

[Premier League](#) [Serie A](#) [La Liga](#) [Bundesliga](#) [Ligue 1](#)
[Eredivisie](#) [Premier League](#) [Brasileirão](#) [Major League Soccer](#) [Super Lig](#)
[Championship](#) [UEFA Champions League](#) [UEFA Europa League](#) [Africa Cup of Nations](#)

Europe » [UEFA Champions League](#) » 2014/2015 » [UEFA Champions League Group Stages](#)

UEFA Champions League Group Stages

Summary [Fixtures](#) [Team Statistics](#) [Player Statistics](#) [News](#) [Add to Favourites](#)

UEFA Champions League Group Stages Team Statistics

Summary [Defensive](#) [Offensive](#) [Detailed](#)

View: Overall Home Away

R Team	Shots pg	Discipline	Possession%	PassSuccess%	AerialsWon	Rating
1 Real Madrid	20.5	9 0	57.0	88.9	9.2	7.72
2 Chelsea	17.2	8 0	56.4	86.5	14.2	7.66

Wenger 'confident' of Walcott stay
2 hours ago - BBC

Watch the new UEFA Women's Football Show
4 hours ago - UEFA

Rafa the gaffer: Benitez's coaching creed
5 hours ago - UEFA

Porto v Bayern: five things we learned
5 hours ago - UEFA

Luis Suárez: another dimension for Barcelona
5 hours ago - UEFA

Terry 'should still play for England'
6 hours ago - BBC

UEFA Champions League Group Stages Situational Statistics

Goal Types [Pass Types](#) [Card Situations](#)

View: Overall Home Away

R Team	Open Play	Counter Attack	Set Piece	Penalty	Own Goal
1 Barcelona	14	0	1	0	0
2 Atletico Madrid	12	0	2	0	0
3 FC Porto	12	1	2	1	0

UEFA Champions League Group Stages Positional Statistics

Attack Sides [Shot Directions](#) [Shot Zones](#) [Action Zones](#)

View: Overall Home Away

R Team	Left Side	Middle of the pitch	Right Side
1 Zenit St. Petersburg	42%	20%	38%
2 Galatasaray	40%	26%	33%
3 Chelsea	40%	28%	32%
4 Real Madrid	40%	32%	28%
5 Olympiakos	39%	27%	34%

Σχήμα 2.16: Η ιστοσελίδα WhoScored.com και οι διάφορες κατηγορίες στατιστικών

Συγκεκριμένα, ο επισκέπτης έχει τη δυνατότητα να δει την βαθμολογία της διοργάνωσης της επιλογής του, τα αποτελέσματα των αγώνων, τα συγκεντρωτικά ομαδικά χαρακτηριστικά (Team Statistics) που έχουν σχέση με την επιθετική και την αμυντική αγωνιστική συγκομιδή (Summary, Offensive, Defensive, Detailed), τις διάφορες καταστάσεις που εξελίσσονται κατά τη διάρκεια ενός αγώνα (Goal Types, Pass Types, Card Situations), όπως επίσης και δεδομένα κατοχής μπάλας (Attack Sides, Shot Directions, Shot Zones, Action Zones). Επιπλέον, ο ενδιαφερόμενος χρήστης έχει την ευχέρεια να επεξεργαστεί δεδομένα ατομικών επιδόσεων (Player Statistics) τα οποία αναλύονται στις αντίστοιχες κατηγορίες των ομαδικών στατιστικών που περιγράφηκαν παραπάνω. Η συγκεκριμένη ιστοσελίδα [12] αποτελεί μία νέα και πλήρης πηγή πληροφόρησης στο χώρο του ποδοσφαίρου και αποτελεί σημαντικό εργαλείο για την ολοκληρωμένη ανάλυση και επεξεργασία ποδοσφαιρικών στατιστικών.

2.2.3 Η Ιστοσελίδα Squawka

Η διαδικτυακή εφαρμογή Squawka.com [07] αναγνωρίζει τον παγκόσμιο αντίκτυπο του ποδοσφαίρου στο φίλαθλο κοινό και επιτυγχάνει να προσφέρει στατιστικά δεδομένα σε πραγματικό χρόνο, από διάφορους ποδοσφαιρικούς αγώνες. Ο επισκέπτης μπορεί να διαβάσει τα νέα των ομάδων που συμμετέχουν σε κορυφαία πρωταθλήματα, να παρακολουθήσει τα αποτελέσματά τους και το πρόγραμμα των επόμενων αγώνων τους. Μπορεί επίσης, να συλλέξει διάφορα μοναδικά στατιστικά που έχουν σχέση με την αγωνιστική απόδοση πολλών ποδοσφαιρικών συλλόγων. Για την συλλογή στατιστικών δεδομένων επιλέγουμε την κατηγορία κέντρο στατιστικών (STATS CENTRE).

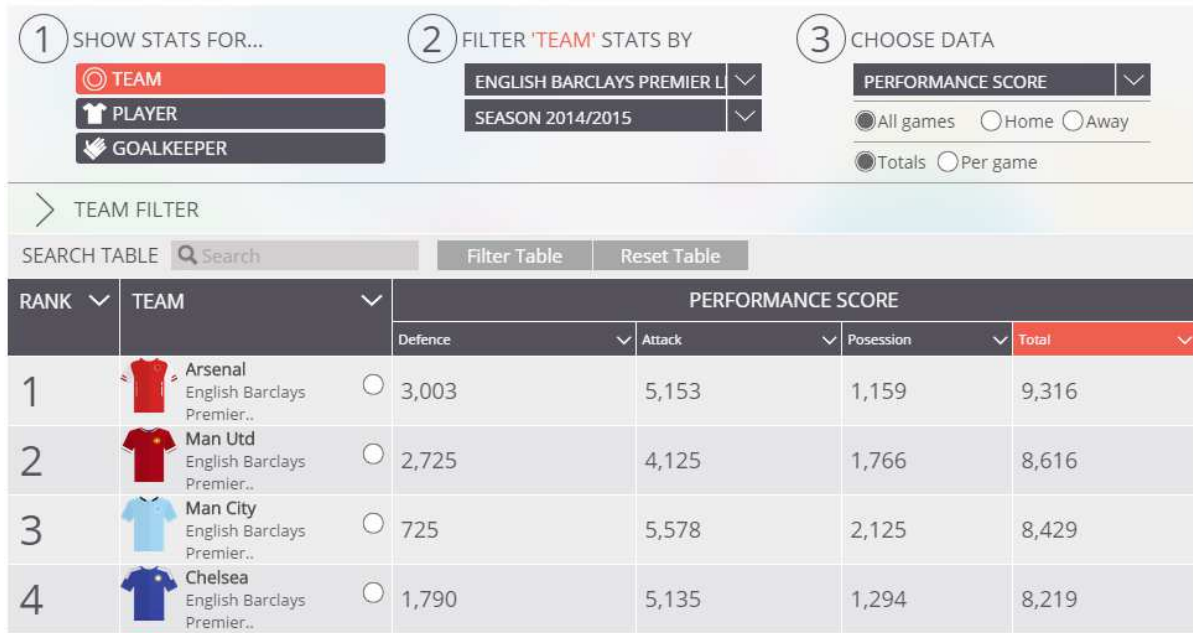


Σχήμα 2.17: Ο πίνακας συγκρίσεων μεταξύ ομάδων της ιστοσελίδας Squawka

Ο χρήστης έχει τη δυνατότητα χρησιμοποιώντας το μενού πίνακα συγκρίσεων (Comparison Matrix) να επιλέξει είτε τους ποδοσφαιριστές που επιθυμεί, είτε τις ομάδες που θέλει και να συγκρίνει τόσο τους μέσους όρους όσο και τα συγκεντρωτικά στατιστικά τους. Στο παραπάνω σχήμα φαίνεται η σύγκριση μεταξύ κάποιων ομάδων που συμμετέχουν στο Champions League. Επίσης, επιλέγοντας ο επισκέπτης την κατηγορία στατιστικά ποδοσφαίρου (Football Stats) μπορεί να μελετήσει τους κορυφαίους ποδοσφαιριστές και τις καλύτερες ομάδες από τη διοργάνωση που έχει προτιμήσει.

TEAM RANKINGS

Share with your friends [Tweet](#) [Share](#) [g+1](#) [Pin it](#)



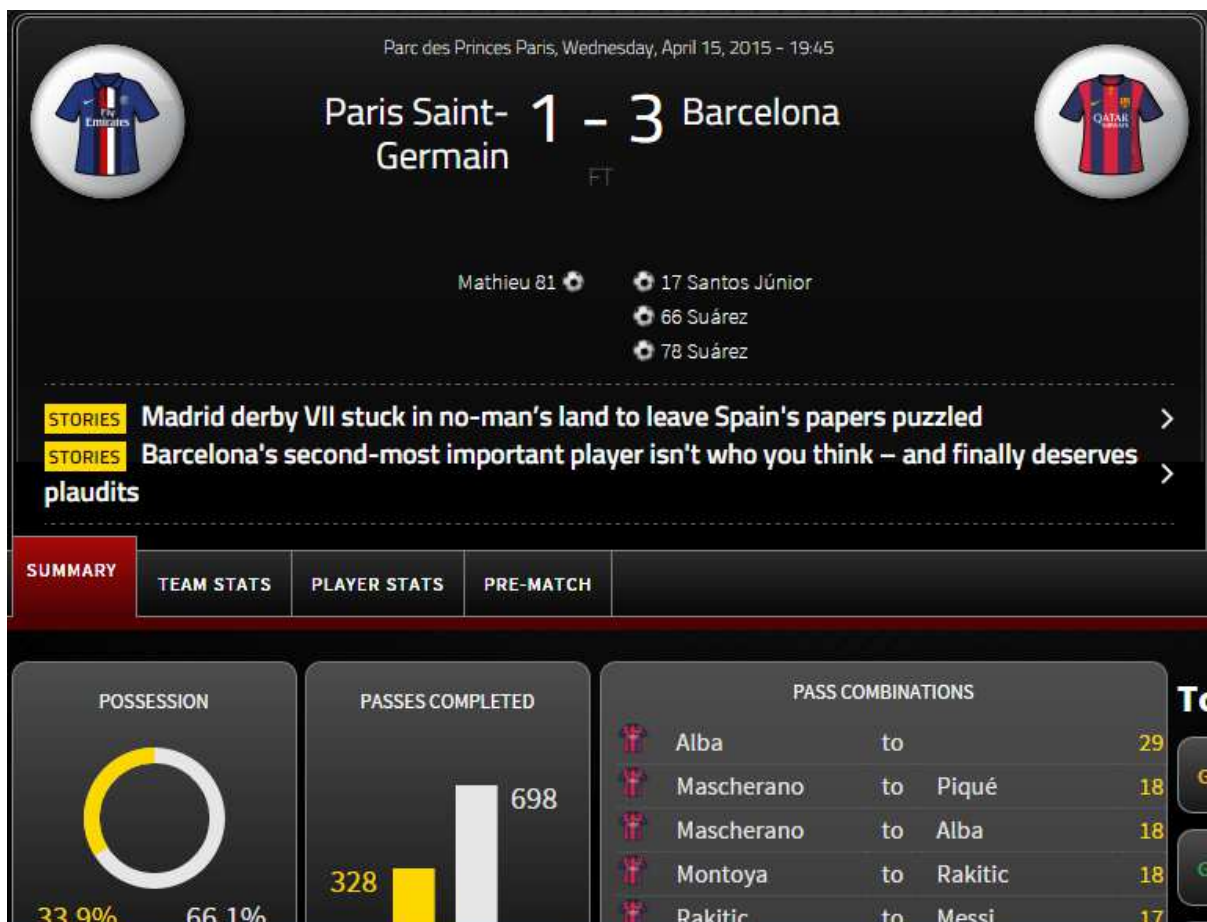
Σχήμα 2.18: Ομαδικά στατιστικά για τις ομάδες που αγωνίζονται στο αγγλικό πρωτάθλημα

Τέλος, υπάρχει δυνατότητα ο ενδιαφερόμενος να μελετήσει τόσο τα ομαδικά όσο και τα ατομικά συγκεντρωτικά στατιστικά. Αυτό που πρέπει να κάνει είναι να επιλέξει το μενού ομαδικές ή ατομικές βαθμολογίες (Team Rankings ή Player Rankings). Μέσα από την κατηγορία ο χρήστης έχει την ευχέρεια να επιλέξει ομάδες, παίχτες, πρωτάθλημα, αγωνιστική περίοδος και διάφορα στατιστικά αγωνιστικής απόδοσης (PERFORMANCE SCORE, SHOT ACCURACY, CHANCES CREATED, DEFENSIVE ACTIONS κα). Το Squawka.com [07] αποτελεί μία πρωτοποριακή ιστοσελίδα, προσφέροντας εργαλεία που συντελούν στην καλύτερη και στην πιο ολοκληρωμένη ανάλυση τόσο ομαδικών όσο και ατομικών στατιστικών δεδομένων.

2.2.4 Η Ιστοσελίδα FourFourTwo

Η ιστοσελίδα FourFourTwo [02] μετράει δύο δεκαετίες στο χώρο της ποδοσφαιρικής ενημέρωσης. Αποτελεί ένα ηλεκτρονικό περιοδικό ποδοσφαίρου, μέσα από το οποίο ο επισκέπτης έχει την ευκαιρία να διαβάσει τα νέα των ομάδων, στοιχεία και πληροφορίες για τους ποδοσφαιρικούς αγώνες που έπονται του προγράμματος, αναλύσεις αγώνων που έχουν διεξαχθεί, συνεντεύξεις και απόψεις τόσο προπονητών όσο και αθλητών ποδοσφαίρου. Για την

εύρεση αναλυτικών ποδοσφαιρικών δεδομένων αγωνιστικής απόδοσης, επιλέγουμε την κατηγορία της στατιστικής ζώνης (STATS ZONE), μέσα από την οποία γίνεται εφικτή η μελέτη της αγωνιστικής παρουσίας των ποδοσφαιρικών συλλόγων.



Σχήμα 2.19: Η ιστοσελίδα FourFourTwo και τα στατιστικά για ένα παιχνίδι του Champions League

Ειδικότερα, ο χρήστης μπορεί να επιλέξει τον αγώνα που θέλει να μελετήσει και να πλοηγηθεί στις διάφορες κατηγορίες δεδομένων αγωνιστικής απόδοσης. Τα στατιστικά που παρουσιάζονται έχουν σχέση με τα στοιχεία κάθε αγώνα ξεχωριστά και όχι με συγκεντρωτικά δεδομένα αγωνιστικής παρουσίας. Ο επισκέπτης μπορεί να επιλέξει ανάμεσα σε τρεις (3) κατηγορίες στατιστικών. Συγκεκριμένα, η κατηγορία γενικά (SUMMARY) περιέχει κάποια συγκεντρωτικά αναλυτικά στοιχεία των ομάδων που αγωνίστηκαν και των παιχτών που συμμετείχαν. Η κατηγορία ομαδικά στατιστικά (TEAM STATS) περιλαμβάνει υποκατηγορίες που δείχνουν και συγκεντρώνουν τόσο την επιθετική (SHOTS, PASSES, ATTACK) όσο και την αμυντική (DEFENCE, FOULS) παρουσία και απόδοση των ποδοσφαιρικών ομάδων. Επιλέγοντας

την υποκατηγορία ατομικά στατιστικά (PLAYER STATS), και πατώντας πάνω στο παίχτη προτίμησης, ο χρήστης μπορεί να μελετήσει στοιχεία για την αγωνιστική προσφορά του ποδοσφαιριστή στον συγκεκριμένο αγώνα. Η ιστοσελίδα FourFourTwo [02], αποτελεί κάτι διαφορετικό στο χώρο της ποδοσφαιρικής στατιστικής ανάλυσης. Ο λόγος είναι ότι, σε αντίθεση με τις περισσότερες ιστοσελίδες που παρουσιάζουν αθροιστικά στατιστικά των ποδοσφαιρικών ομάδων, η συγκεκριμένη παρουσιάζει τα ανεξάρτητα δεδομένα των ομάδων για κάθε αγώνα που συμμετέχουν. Τέλος,, μέσω της FourFourTwo [02] ο επισκέπτης μπορεί να αποκτήσει μία σφαιρικότερη και πληρέστερη εικόνα για τις αγωνιστικές επιδόσεις της εξεταζόμενης ομάδας.

Συμπερασματικά, τα τελευταία χρόνια παρατηρείται η δημιουργία μεγάλου αριθμού ιστοσελίδων που αποκλειστικό στόχο έχουν την πληρέστερη συλλογή και ολοκληρωμένη ανάλυση και την επιστημονική επεξεργασία στατιστικών αγωνιστικής απόδοσης. Το γεγονός αυτό αποδεικνύει την τεράστια αύξηση της εμπορικότητας του αθλήματος του ποδοσφαίρου και του ενδιαφέροντος του φίλαθλου κοινού για τις παγκόσμιες ποδοσφαιρικές εξελίξεις. Παραπάνω, παρουσιάστηκαν τέσσερις (4) από τις κυριότερες σελίδες παρουσίας ποδοσφαιρικών δεδομένων, που αποστολή τους είναι να αποτελέσουν βασικό εργαλείο στα χέρια του κάθε ενδιαφερόμενου χρήστη για την προσωπική ενημέρωση του ή να συμβάλλουν στην επαγγελματική του εξέλιξη στο χώρο αθλητικών στατιστικών (sport analytics).

2.3 Τεχνολογία Συλλογής Στατιστικών Ποδοσφαίρου

Η εύρεση εκατοντάδων στατιστικών ποδοσφαίρου οφείλεται σε μεγάλο βαθμό στην τεράστια εξέλιξη της τεχνολογίας, σε συνδυασμό πάντα με την αλλαγή στον τρόπο τηλεοπτικής κάλυψης των ποδοσφαιρικών αγώνων. Η UEFA συγκεντρώνει τα αγωνιστικά δεδομένα μέσω του συστήματος Matrics, το οποίο έχει κατασκευαστεί από την Ιταλική εταιρεία Deltatre, έχει χρησιμοποιηθεί σε περισσότερους από δύο χιλιάδες (2000) αγώνες του Champions League και εφαρμόζει ένα συνδυασμό τεχνολογιών [10]. Το συγκεκριμένο σύστημα χρησιμοποιεί δύο (2) σετ οχτώ (8) καμερών υψηλής ευκρίνειας, οι οποίες είναι τοποθετημένες σε διάφορα σημεία του αγωνιστικού χώρου και χρησιμοποιεί τεχνολογία ανίχνευσης εικόνων (image recognition) για την σάρωση των κινήσεων των είκοσι δύο (2) ποδοσφαιριστών, των τριών (3) διαιτητών, της μπάλας, του σχηματισμού της κάθε ομάδας και των κινήσεων του κάθε προπονητή [05]. Έτσι, επιτυγχάνεται η παραγωγή δείκτη παρουσίας κάθε παίχτη (performance index), η οποία

αξιολογείται και εξάγεται με τη χρήση μίας μαθηματικής φόρμουλας. Η εξαγωγή στατιστικών δεδομένων μέσω της χρήσης οπτικών μέσων ανίχνευσης κινήσεων εντός γηπέδου, οφείλεται σε μεγάλο βαθμό και στην άδεια των ποδοσφαιριστών να τοποθετηθεί μία τεχνολογία σάρωσης στα παπούτσια τους. Στην κορυφή των καμερών έχουν υλοποιηθεί αλγόριθμοι, οι οποίοι έχουν τη δυνατότητα να παράγουν δεδομένα κάθε δευτερόλεπτο, να τα επεξεργάζονται και να τα αξιολογούν, έτσι ώστε να τα εμφανίζουν σαν στατιστικά στοιχεία (περίπου 350) στους τηλεθεατές. Έτσι, παράγονται γραφικό περιεχόμενο με τα καταγεγραμμένα δεδομένα, στατιστικά πρώτου επιπέδου (goals, assists, attempts, κα), καθώς και συνθετότερα (ball possession, distance covered, passing distribution κα) [05]. Η ύπαρξη τέτοιων τεχνολογιών συμβάλλει στην βαθύτερη και αναλυτικότερη τηλεοπτική κάλυψη των αγώνων ποδοσφαίρου και επιπλέον βοηθάει στην αποτελεσματικότερη περιγραφή τόσο της ομαδικής όσο και της ατομικής αγωνιστικής παρουσίας και απόδοσης.

2.4 Υπολογισμός Στοιχηματικών Αποδόσεων

Η αισθητή αύξηση της εμπορικότητας του ποδοσφαίρου έχει συμβάλλει και στην παράλληλη έξαρση και διάδοση άλλων δραστηριοτήτων σχετικών με το συγκεκριμένο άθλημα, όπως είναι το στοίχημα. Στη σημερινή εποχή, παρατηρείται η ύπαρξη μεγάλου αριθμού στοιχηματικών εταιριών, οι οποίες προσφέρουν στους πελάτες τους την ευκαιρία στοιχηματισμού σε διάφορα γεγονότα ποδοσφαιρικών αγώνων ανά τον κόσμο. Η στοιχηματική απόδοση που εξάγεται από την εκάστοτε εταιρία στηρίζεται στην πιθανότητα ένα γεγονός να συμβεί. Χρησιμοποιούνται διάφορα περιγραφικά στατιστικά (descriptive statistics), όπως ο αριθμητικός μέσος (mean) οι σχετικές και απόλυτες αποκλίσεις (relative and absolute deviation - residuals) [08]. Ο υπολογισμός στηρίζεται σε μεγάλο βαθμό στην ύπαρξη δεδομένων για κάθε αγώνα, δηλαδή όσο περισσότερα ιστορικά δεδομένα υπάρχουν τόσο ακριβότερη πρόβλεψη προκύπτει. Έτσι, γίνεται προσπάθεια ανάλυσης των αποτελεσμάτων προηγούμενων αγώνων των ομάδων (περίπου 20 με 25 αγώνες) και παράλληλα πραγματοποιείται επεξεργασία της προϊστορίας μεταξύ των δύο (2) αντιπάλων για τα τελευταία δέκα (10) χρόνια [08]. Το άθροισμα των πιθανοτήτων για κάθε σημείο του αγώνα (Νίκη Γηπεδούχου - Ισπαλία - Νίκη Φιλοξενούμενου), θα έπρεπε να ισούται με 100%. Παρόλα αυτά είναι μεγαλύτερο από το συγκεκριμένο αριθμό και η διαφορά αυτή αντιστοιχεί στο περιθώριο κέρδους των στοιχηματικών εταιριών, το οποίο κυμαίνεται από 4% έως 7% και ονομάζεται overground. Έτσι, αν η νίκη του γηπεδούχου έχει πιθανότητα 48,16% τότε η απόδοση θα είναι $1 / 0,4816 = 2,08$. Συμπερασματικά, ο υπολογισμός των στοιχηματικών

αποδόσεων είναι μία σύνθετη διαδικασία, η οποία περιλαμβάνει την εφαρμογή τόσο στατιστικών όσο και μαθηματικών μεθόδων και παράλληλα απαιτεί την μελέτη και γνώση εξειδικευμένων ποδοσφαιρικών πληροφοριών.

2.5 Το Football Manager

Η παγκόσμια αύξηση του ενδιαφέροντος για το άθλημα του ποδοσφαίρου έχει επηρεάσει σε μεγάλο βαθμό και τη βιομηχανία των video games. Έτσι, παρατηρείται τα τελευταία χρόνια η προσπάθεια από πολλές εταιρίες λογισμικού για τη δημιουργία πολλών παιχνιδιών σχετικών με το συγκεκριμένο άθλημα. Ένα από τα εμπορικότερα είναι το Football Manager, δημιουργήθηκε το 1992 και δίνει την ευκαιρία στον παίχτη να γίνει ο προπονητής και διαχειριστής μίας ομάδας, έχοντας την ευκαιρία να προπονεί τους ποδοσφαιριστές, να κάνει μεταγραφές παιχτών, να επιλέγει σχηματισμούς - συστήματα των ομάδων στους διάφορους αγώνες. Το συγκεκριμένο παιχνίδι είναι στην πραγματικότητα μία προσομοίωση των συνθηκών λειτουργίας ενός ποδοσφαιρικού συλλόγου. Το Football Manger περιλαμβάνει μία τεράστια βάση δεδομένων 550000 παιχτών με 200 - 250 επιμέρους χαρακτηριστικά για τον καθένα και αντίστοιχα 150 για τους προπονητές και τους συνεργάτες τους. Για την εξεύρεση των παραπάνω πληροφοριών εργάζονται περίπου 2000 ανιχνευτές ταλέντων (scouts) σε περισσότερες από 50 χώρες, οι οποίοι παρακολουθούν ομάδες και παίχτες σε όλο τον κόσμο, συμπεριλαμβανομένης της πρώτης ομάδας, τις ρεζέρβες και την ομάδα νέων. Οι συγκεκριμένοι ενημερώνονται από οδηγούς 150 σελίδων για το πως πρέπει να αξιολογούν και να βαθμολογούν ένα ποδοσφαιριστή [09]. Όλα αυτά, αποδεικνύουν την αξιοπιστία των στοιχείων του παιχνιδιού και παράλληλα το σημαντικό ρόλο που μπορεί να διαδραματίσει. Πολλοί ποδοσφαιρικοί σύλλογοι, χρησιμοποιούν το Football Manager για την εξεύρεση ελπιδοφόρων ποδοσφαιριστών, οι οποίοι μπορεί να έχουν μεγάλη συμβολή στο σύλλογο με μικρό χρηματικό κόστος αντίστοιχα.

Το Football Manager έχει κατασκευαστεί με τεχνικές τεχνητής νοημοσύνης και μηχανικής μάθησης (νευρωνικά δίκτυα - neural networks), ενώ χρησιμοποιεί μοντέλα πιθανοτήτων (probability models) για να γίνεται το παιχνίδι πιο ρεαλιστικό. Ο πηγαίος κώδικας του είναι μια σειρά από μαθηματικές εξισώσεις και συμπεριλαμβάνει μεγάλο όγκο στατιστικών δεδομένων. Η μηχανή του παιχνιδιού (match engine) κάνει χρήση 3D γραφικών, οι υπολογισμοί συντελούνται κάθε 1/8 του δευτερολέπτου ενώ το αποτέλεσμα του αγώνα έχει υπολογιστεί πριν την έναρξη του [09]. Κατά τη διάρκεια του αγώνα, αν γίνουν αλλαγές από την πλευρά του παίχτη (αλλαγή

συστήματος ομάδας, αλλαγή παίχτη κα), τότε πραγματοποιείται νέος υπολογισμός του αποτελέσματος με τα νέα δεδομένα. Για την τελική έκβαση του αγώνα λαμβάνονται υπόψη κάθε απόφαση, ενέργεια του παίχτη προς τους άλλους, έτσι ώστε να γίνεται η καλύτερη πρόβλεψη σε συνδυασμό με τα ατομικά του χαρακτηριστικά. Συμπερασματικά, το Football Manager με τα εμπειριστατωμένα δεδομένα του αποτελεί μία πραγματικότητα και μετατρέπεται σε ένα χρήσιμο εργαλείο στα χέρια κάθε προπονητή ή μάνατζερ ποδοσφαίρου.

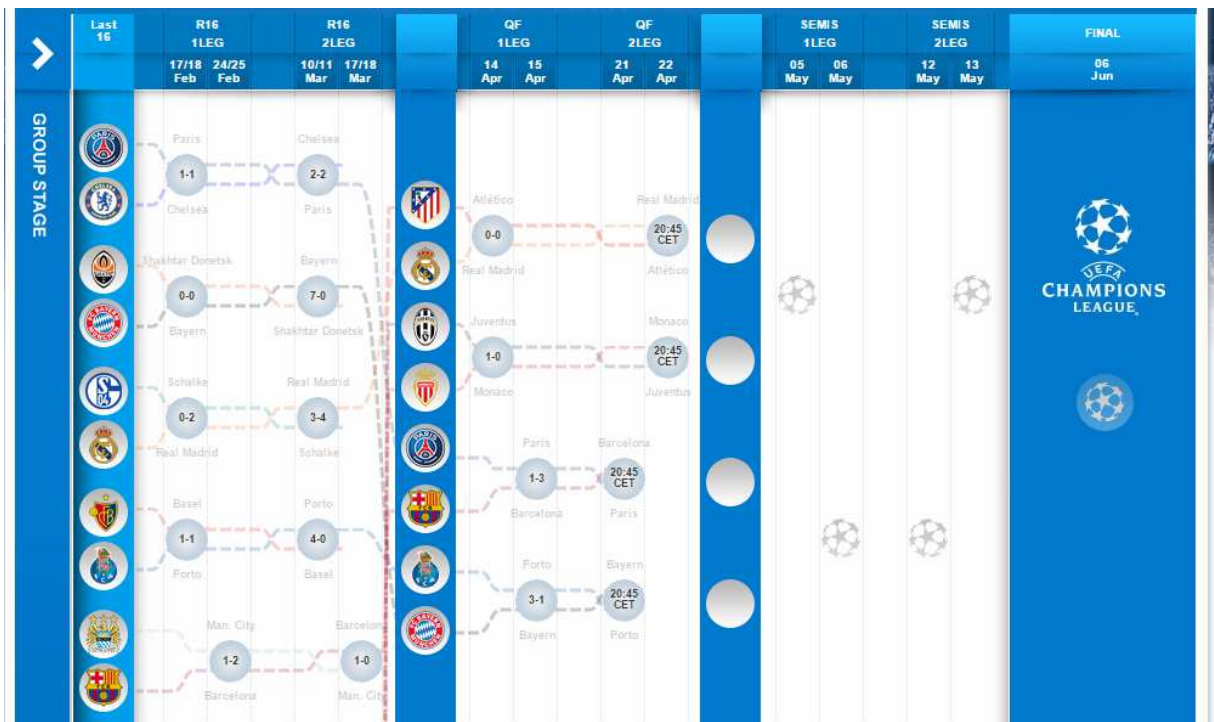
2.6 Σύστημα Διεξαγωγής του Champions League

Η διοργάνωση του Champions League αποτελείται από δύο (2) φάσεις, την φάση των ομίλων και τη knockout. Στη φάση των ομίλων δημιουργούνται οχτώ (8) όμιλοι των τεσσάρων (4) ομάδων, προκρίνονται στην επόμενη φάση οι δύο (2) πρώτες ομάδες κάθε ομίλου, ενώ οι αγώνες διεξάγονται στο διάστημα 16 - 17 Σεπτεμβρίου με 9 - 10 Δεκεμβρίου.



Σχήμα 2.20: Το αποτελέσματα των δύο πρώτων ομίλων και ποιες ομάδες προκρίθηκαν

Στη συνέχεια διεξάγεται η Knockout φάση των αγώνων, οι οποίοι προκύπτουν κατόπιν κλήρωσης. Στη φάση των 16 οι πρώτοι των ομίλων κληρώνονται με τους αντίστοιχους δεύτερους, ενώ δεν επιτρέπεται να αγωνιστούν μεταξύ τους ομάδες από την ίδια χώρα. Η παρούσα φάση διεξάγεται στο διάστημα 17 - 18 Φεβρουαρίου με 17 - 18 Μαρτίου. Οι νικητές των αγώνων προκρίνονται στην φάση των 8 (Προημιτελικά), όπου όλες οι ομάδες έχουν τις ίδιες πιθανότητες να κληρωθούν μεταξύ τους. Στη φάση αυτή το πρώτο παιχνίδι πραγματοποιείται στις 14-15 Απριλίου και το δεύτερο στις 21 - 22 Απριλίου.



Σχήμα 2.21: Το γράφημα με τη knockout φάση της διοργάνωσης του Champions League

Οι νικητές της προημιτελικής φάσης προκρίνονται στην φάση των 4 (Ημιτελικά), η οποία πραγματοποιείται στις 5 - 6 Μαΐου το πρώτο παιχνίδι και στις 12 - 13 Μαΐου το δεύτερο. Οι νικητές συμμετέχουν στον τελικό αγώνα (ένα παιχνίδι) που θα διεξαχθεί στις 6 Ιουνίου, όπου προκύπτει και ο πρωταθλητής της διοργάνωσης.

Κεφάλαιο 3

Μεθοδολογία της Έρευνας

3.1 Σκοπός και Προσδοκώμενα Αποτελέσματα

Σκοπός της παρούσας μεταπτυχιακής διατριβής είναι η συλλογή και η ολοκληρωμένη ανάλυση στατιστικών δεδομένων αγωνιστικής απόδοσης των δέκα έξι (16) κορυφαίων ευρωπαϊκών ποδοσφαιρικών συλλόγων, που αγωνίζονται στην διοργάνωση του Champions League την περίοδο 2104 - 2015 και η πρόβλεψη του πρωταθλητή της συγκεκριμένης διοργάνωσης. Συγκεκριμένα, θα υλοποιηθεί εφαρμογή η οποία θα χρησιμοποιεί και θα εφαρμόζει τον αλγόριθμο αναζήτησης Τεχνητής Νοημοσύνης Κατά Πλάτος (BFS), για την στατιστική ανάλυση

των αριθμητικών δεδομένων και την τελική πρόβλεψη του τελικού νικητή. Κατά συνέπεια με την συγκεκριμένη μεταπτυχιακή διατριβή διερευνάται:

1. Η ύπαρξη μεγάλου αριθμού στατιστικών ποδοσφαίρου σε υπάρχουσες βάσεις δεδομένων και η ολοκληρωμένη ανάλυση τους για την εξαγωγή ικανοποιητικού αποτελέσματος πρόβλεψης του τελικού νικητή,
2. Η επίδραση της προσθήκης νέων προσωπικών στατιστικών στο υπάρχον αποτέλεσμα και η εξαγωγή εγκυρότερης πρόβλεψης.

Κίνητρο για την συγκεκριμένη μεταπτυχιακή διατριβή ήταν η έλλειψη ικανών εργαλείων ανάλυσης αθλητικών στατιστικών δεδομένων και πρόβλεψης αποτελεσμάτων διαφόρων αθλητικών γεγονότων και ειδικά του διασημότερου αθλήματος παγκοσμίως, του ποδοσφαίρου. Έτσι, φιλοδοξία είναι ο σχεδιασμός και η υλοποίηση μίας εφαρμογής στατιστικής ανάλυσης, η οποία θα βασίζει την εύρεση του αποτελέσματος σε επιστημονικά δεδομένα και όχι στον παράγοντα της τύχης και να αποτελέσει πρότυπο σχεδίασης και υλοποίησης άλλων παρόμοιων, ίσως και πιο σύνθετων εφαρμογών. Προσδοκείται η κάλυψη του κενού που υπάρχει στη συσχέτιση χρήσης αλγορίθμων τεχνητής νοημοσύνης και στην επεξεργασία στατιστικών δεδομένων, με στόχο την επαρκή πρόβλεψη της έκβασης των αποτελεσμάτων διαφόρων αθλητικών διοργανώσεων και η δημιουργία ενός καινοτόμου εργαλείου, το οποίο θα αποτελεί τη βάση για μελλοντική χρήση σε παρόμοια πιθανά σενάρια πρόβλεψης αποτελεσμάτων διαφόρων αθλητικών διοργανώσεων.

3.2 Ερευνητικά Ερωτήματα

Με βάση τους παραπάνω στόχους προκύπτουν τα εξής ερευνητικά ερωτήματα:

1. Με τα υπάρχοντα ποδοσφαιρικά στατιστικά στοιχεία μπορεί η εφαρμογή να εξάγει έγκυρο και αποδεκτό τελικό αποτέλεσμα;

2. Με την εισαγωγή νέων στατιστικών στοιχείων που θα προτείνει ο ερευνητής, μπορεί η εφαρμογή να εξαγει έγκυρο και αποδεκτό αποτέλεσμα;

Με βάση τα παραπάνω ερευνητικά ερωτήματα θα επιχειρηθεί προσπάθεια για τη δημιουργία ικανού εργαλείου για την πρόβλεψη του νικητή μίας ποδοσφαιρικής διοργάνωσης. Η ύπαρξη μεγάλου όγκου ποδοσφαιρικών στατιστικών αγωνιστικής απόδοσης στο διαδίκτυο, δεν εξασφαλίζει την εύκολη και την πλήρη πρόβλεψη των νικητών διαφόρων ποδοσφαιρικών διοργανώσεων. Η υλοποίηση εφαρμογής πρόβλεψης θα συνεισφέρει στην κάλυψη του κενού σχετικά με την έλλειψη αντίστοιχων εργαλείων αλλά θα συμβάλει και στη δημιουργία μίας κατεύθυνσης για τη βελτίωση υπάρχουσών εφαρμογών και στην υλοποίηση παρόμοιων και πιο εξελιγμένων εργαλείων πρόβλεψης. Επιπλέον, η εισαγωγή προσωπικών στατιστικών που δεν είναι καταχωρημένα σε κάποια βάση δεδομένων δίνει την ελπίδα για την εξέλιξη της παρούσας εφαρμογής και για την εξαγωγή πληρέστερου και εγκυρότερου αποτελεσματος. Τα αποτελέσματα, της παρούσας μεταπτυχιακής διατριβής, είναι η παροχή πληροφοριών για την υλοποίηση ικανού εργαλείου πρόβλεψης του νικητή της διοργάνωσης του Champions League και η δυνατότητα παρουσίασης και αξιοποίησης κάποιων τεχνικών ανάλυσης στατιστικών γενικότερα, έτσι ώστε να συνεισφέρουν στην εξέλιξη και στον εμπλουτισμό του πεδίου των προβλέψεων ποδοσφαιρικών γεγονότων.

3.3 Μεθοδολογία

3.3.1 Η Επιλογή του Αλγορίθμου Τεχνητής Νοημοσύνης

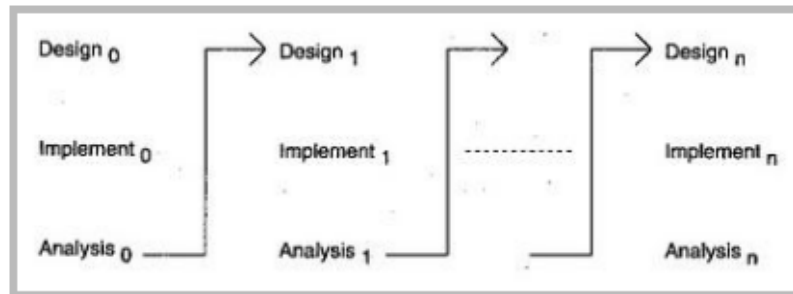
Ο σχεδιασμός και η υλοποίηση της εφαρμογής πρόβλεψης προϋποθέτει τη χρήση ενός αλγορίθμου αναζήτησης Τεχνητής Νοημοσύνης. Για τις ανάγκες της μεταπτυχιακής διατριβής επιλέχτηκε ο αλγόριθμος τυφλής αναζήτησης Κατά Πλάτος (BFS). Θεωρείται αρκετά απλός στην υλοποίηση και ταιριάζει στα περισσότερα προβλήματα αναζήτησης. Στη περίπτωση του συγκεκριμένου προβλήματος θα χρησιμοποιηθεί γράφος που θα αναπαριστά όλες τις αγωνιζόμενες ομάδες και τα αντίστοιχα στατιστικά τους δεδομένα με κόμβους ενώ τα κόστη από κόμβο σε κόμβο θα είναι μηδενικά. Ο αλγόριθμος Κατά Πλάτος θεωρείται ότι είναι ο ιδανικότερος και χαρακτηρίζεται βέλτιστος για αναζήτηση σε γράφους όπου παρατηρούνται μηδενικά κόστη για ενέργειες μεταξύ κόμβων [06]. Επιπλέον, για τη συγκεκριμένη εφαρμογή

απαιτείται η ανάλυση και η εξέταση όλων των στατιστικών που θα συλλεχθούν, οπότε ο συγκεκριμένος αλγόριθμος είναι η καλύτερη λύση γιατί εξετάζει όλους τους κόμβους σε ένα επίπεδο γράφου και μετά επεκτείνεται στους κόμβους των επόμενων επιπέδων, μέχρι να βρει τη λύση, που στη προκειμένη περίπτωση είναι η εξεύρεση του τελικού νικητή. Επίσης, στις παρούσες συνθήκες παρουσιάζεται ένας γράφος πεπερασμένου αριθμού κόμβων (όσες είναι οι ομάδες που συμμετέχουν), οπότε ο αλγόριθμος Κατά Πλάτος παρόλο που πρέπει να εξετάσει όλους τους κόμβους, δεν θα έχει μεγάλες απαιτήσεις τόσο σε χρόνο εκτέλεσης (πολυπλοκότητα χρόνου) όσο και σε χώρο αποθήκευσης (πολυπλοκότητα χώρου).

3.3.2 Μέθοδος Προσέγγισης

Η μεθοδολογία ανάπτυξης λογισμικού ή υπολογιστικών συστημάτων, είναι ένα πλαίσιο το οποίο χρησιμοποιείται για το σχεδιασμό, την υλοποίηση και τον έλεγχο της διαδικασίας ανάπτυξης εφαρμογών. Μία πολύ σημαντική μεθοδολογία ανάπτυξης λογισμικού είναι και η επαναληπτική μέθοδος (iterative development). Η επαναληπτική ανάπτυξη είναι ο τρόπος να χωρίζεται η διαδικασία κατασκευής λογισμικού σε μικρότερα τμήματα, ο χαρακτηριστικός πηγαίος κώδικας σχεδιάζεται, αναπτύσσεται και ελέγχεται σε συνεχόμενους κύκλους. Δεν επιχειρεί να ξεκινήσει μία εφαρμογή με όλες τις προσδιοριζόμενες απαιτήσεις, αλλά η ανάπτυξη του συστήματος αρχίζει με τον προσδιορισμό και την εφαρμογή ενός τμήματος του λογισμικού. Σε κάθε επανάληψη, πρόσθετα χαρακτηριστικά μπορούν να σχεδιαστούν, να υλοποιηθούν και να ελεγχθούν, μέχρι να επιτευχθεί η πλήρης λειτουργικότητα του λογισμικού [04]. Η διαδικασία λαμβάνει υπόψη ότι στην πραγματικότητα δεν λειτουργεί η εφαρμογή ιδανικά από το πρώτο στάδιο της υλοποίησής της. Ο στόχος της επαναληπτικής διαδικασίας είναι να επιτρέπει μεγαλύτερη ευελιξία στις διαδικασίες αλλαγών και να αποτρέπει την εμφάνιση απρόβλεπτων προβλημάτων κατά τη λειτουργία εφαρμογής

Diagram of Iterative model:



Σχήμα 3.1: Το διάγραμμα της επαναληπτικής διαδικασίας

Για να επιτευχθεί ο σκοπός της μεταπτυχιακής διατριβής, δηλαδή η δημιουργία εφαρμογής για την έγκυρη πρόβλεψη του νικητή του Champions League, επιλέχθηκε η παραπάνω επαναληπτική μέθοδος ανάπτυξη λογισμικού, η οποία είναι ιδιαίτερα αποτελεσματική και ταιριάζει στα χαρακτηριστικά του συγκεκριμένου προβλήματος. Η εφαρμογή μέσα από την υλοποίηση του αλγορίθμου Κατά Πλάτος και για την έγκυρη εξαγωγή τελικού αποτελέσματος, σκοπό έχει την ανάλυση των καταγεγραμμένων στατιστικών σε επίπεδα. Δεν θα έχουν όλα τα στατιστικά την ίδια προτεραιότητα αλλά θα διαχωρίζονται σε διάφορα επίπεδα ανάλογα με τη βαρύτητα που τους έχει δοθεί. Έτσι, η εφαρμογή θα τα αναλύει με τη σειρά προτεραιότητας τους, θα πραγματοποιεί κάποιους υπολογισμούς μεταξύ των δοθέντων αριθμητικών δεδομένων και θα εξάγει το τελικό αποτέλεσμα. Επίσης, εκτός από τον τελικό νικητή, η εφαρμογή θα κάνει τις προβλέψεις για τους νικητές όλων των ζευγαριών που θα προκύψουν στη διοργάνωση. Θα πραγματοποιηθούν οι αγώνες και ανάλογα με τις αποκλίσεις που θα παρουσιαστούν στα αποτελέσματα τους θα επιχειρηθούν στη συνέχεια να γίνουν προσαρμογές στην εφαρμογή, με την αλλαγή σειράς εξέτασης και συσχετισμών μεταξύ των στατιστικών, έτσι ώστε να επιτευχθεί αποτελεσματικότερη και εγκυρότερη εξαγωγή τελικού αποτελέσματος.

Κεφάλαιο 4

Σχεδιασμός και Υλοποίηση

4.1 Εισαγωγή

Στα προηγούμενα κεφάλαια έγινε περιγραφή των βασικών εννοιών των αλγορίθμων αναζήτησης Τεχνητής Νοημοσύνης. Έγινε συνοπτική ανάλυση των κυριότερων αλγορίθμων τυφλής και ευριστικής αναζήτησης και πως προσεγγίζουν τα προβλήματα αναζήτησης, Τέλος, πραγματοποιήθηκε παρουσίαση της μεθοδολογίας και ποιός αλγόριθμος Τεχνητής Νοημοσύνης επιλέχθηκε για την υλοποίηση της εφαρμογής.

Στο κεφάλαιο αυτό, θα γίνει αναλυτική περιγραφή του προβλήματος που θα επιλυθεί. Θα αναλυθεί ο τρόπος με τον οποίο, θα χρησιμοποιηθεί στην πράξη ο επιλεγόμενος αλγόριθμος

αναζήτησης. Τέλος, θα δοθεί προσοχή στην εφαρμογή που υλοποιήθηκε στα πλαίσια της μεταπτυχιακής διατριβής.

4.2 Περιγραφή του Προβλήματος

Το αντικείμενο της παρούσας μεταπτυχιακής διατριβής είναι η ανάπτυξη μίας εφαρμογής η οποία θα υλοποιεί στατιστική ανάλυση και θα προβλέπει τον τελικό νικητή του Champions League. Το πρόβλημα αναλύεται στα παρακάτω τέσσερα (4) υποπροβλήματα:

1. Συλλογή στατιστικών αγωνιστικής απόδοσης των δέκα έξι (16) ομάδων που έχουν προκριθεί στη δεύτερη φάση της διοργάνωσης του Champions League, από τις ιστοσελίδες που περιγράφηκαν στο Κεφάλαιο 2. Τα στατιστικά θα περιλαμβάνουν στοιχεία για την απόδοση της ομάδας τόσο στο εγχώριο πρωτάθλημα της όσο και στη διοργάνωση του Champions League. Τα δεδομένα αυτά αποθηκεύονται σε μία βάση δεδομένων.
2. Παρατήρηση της εξέλιξης των αγώνων του πρώτου γύρου της διοργάνωσης καθώς και των στατιστικών των ομάδων που συμμετείχαν σε αυτόν, στη συνέχεια γίνεται καταγραφή των συχνοτήτων εμφάνισης των δεδομένων που έχουν καταχωρηθεί στη βάση δεδομένων. Με βάση τις καταγεγραμμένες συχνότητες, θα δοθεί προτεραιότητα και βαρύτητα στα δεδομένα της βάσης.
3. Υλοποίηση εφαρμογής, η οποία θα επικοινωνεί με τη βάση δεδομένων, θα έχει πρόσβαση στους πίνακες και τα δεδομένα της. Στη συνέχεια θα συγκρίνει τα στατιστικά των ομάδων που είναι αντίπαλες σε κάθε αγωνιστική φάση της διοργάνωσης βάσει της βαρύτητας που έχουν πάρει και στο τέλος θα τα αθροίζει και θα εξάγει τις πιθανότητες πρόκρισης για κάθε ομάδα ξεχωριστά.
4. Εισαγωγή στη βάση δεδομένων ενός ή περισσότερων προσωπικών στατιστικών για την αναπροσαρμογή της σειράς των συγκρίσεων των δεδομένων αγωνιστικής απόδοσης για την εξαγωγή εγκυρότερου αποτελέσματος.

Όπως γίνεται αντιληπτό, ότι έχει αναφερθεί μέχρι στιγμής αποτελεί βάση για την ανάπτυξη της εφαρμογής πρόβλεψης. Με το συσχετισμό της εφαρμογής με τον αλγόριθμο Κατά Πλάτος θα επιχειρηθεί η πληρέστερη πρόβλεψη του τελικού νικητή. Τα στατιστικά της βάσης δεδομένων θα συγκρίνονται σειριακά ανά επίπεδο προτεραιότητας - η ομάδα που υπερτερεί σε ένα στατιστικό θα κερδίζει κάποιους πόντους, η χαμένη δεν θα παίρνει κάτι - αν έχουν ακριβώς την ίδια τιμή σε ένα δεδομένο θα παίρνουν τους ίδιους πόντους. Η εφαρμογή, στη συνέχεια θα αποθηκεύει ένα συνολικό άθροισμα πόντων για κάθε ομάδα, στη πορεία η σύγκριση θα οδηγείται στο αμέσως επόμενο επίπεδο στατιστικών όπου θα ακολουθείται η ίδια ακριβώς διαδικασία κοκ, μέχρι να φτάσει η εφαρμογή στο τελευταίο επίπεδο προτεραιότητας, όπου τελειώνει το στάδιο της σύγκρισης. Στο τέλος, τα αποθηκευμένα αθροίσματα των πόντων των δύο ομάδων εξάγονται σε μορφή ποσοστού επί της εκατό και παρουσιάζεται το αποτέλεσμα με τον τελικό νικητή.

4.3 Προσέγγιση του Προβλήματος

Προτού ακολουθήσει η συνέχεια με την περιγραφή και την ανάλυση της εφαρμογής, θα αναφερθούν ορισμένα στοιχεία τα οποία πρέπει να ληφθούν υπόψη προτού ξεκινήσει η ανάπτυξη της.

4.3.1 Επιλογή Εργαλείων

Μία σημαντική διαδικασία για την υλοποίηση της εφαρμογής ήταν η επιλογή των απαραίτητων εργαλείων. Τα εργαλεία που επιλέχθηκαν να χρησιμοποιηθούν είναι τα εξής:

1. Ένα σύστημα διαχείρισης βάσης δεδομένων (ΣΔΒΜ) το οποίο επιτρέπει τη δημιουργία μία βάση δεδομένων που είναι απαραίτητη για την αποθήκευση των απαραίτητων στατιστικών που περιγράφηκαν παραπάνω. Χρησιμοποιήθηκε η MySQL που είναι ένα ΣΔΒΜ ανοιχτού κώδικα το οποίο παρέχεται δωρεάν και είναι γνωστό για τη σταθερότητα και την ταχύτητά του. Το εργαλείο που χρησιμοποιήθηκε για τη δημιουργία της βάσης δεδομένων είναι το SQL Server Management Studio το οποίο είναι ένα σύστημα διαχείρισης σχεσιακής βάσης δεδομένων και αναπτύχθηκε από την εταιρία Microsoft. Είναι ένα λογισμικό το οποίο επιτρέπει την άμεση καταχώρηση και ανάκτηση

δεδομένων και βοηθάει στην εύκολη επικοινωνία με άλλες εξωτερικές εφαρμογές. Η έκδοση που χρησιμοποιήθηκε είναι η SQL Server Management Studio 2008.

2. Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίηση της εφαρμογής είναι η Java η οποία αναπτύχθηκε από την εταιρία Sun Microsystems και αποτελεί μία ευρέως χρησιμοποιημένη γλώσσα προγραμματισμού στο τομέα της ανάπτυξης λογισμικού. Είναι μία γλώσσα γενικού σκοπού, αντικειμενοστραφής, φορητή σε επίπεδο μεταγλωττισμένου κώδικα, γλώσσα υψηλού επιπέδου και δυναμική επιτρέποντας την εύκολη προσαρμογή σε διαφορετικά περιβάλλοντα και τη επικοινωνία μεταξύ ετερογενών συστημάτων.
3. Για τη σύνδεση της εφαρμογής με τη MySql χρησιμοποιήθηκε το interface JDBC το οποίο προσφέρεται από το jdk της Java, και το Driver `SQLServerDriver` το οποίο υλοποιεί τη διασύνδεση JDBC.
4. Η συγγραφή του κώδικα έγινε στο Eclipse που είναι ένα IDE (integrated development environment) το οποίο παρέχεται δωρεάν από την εταιρία Eclipse Foundation. Είναι εύκολα επεκτάσιμο για τη παραμετροποίηση του περιβάλλοντος εργασίας, ενώ έχει δημιουργηθεί κυρίως για τη συγγραφή κώδικα Java.

4.3.2 Υπόθεση Γράφου

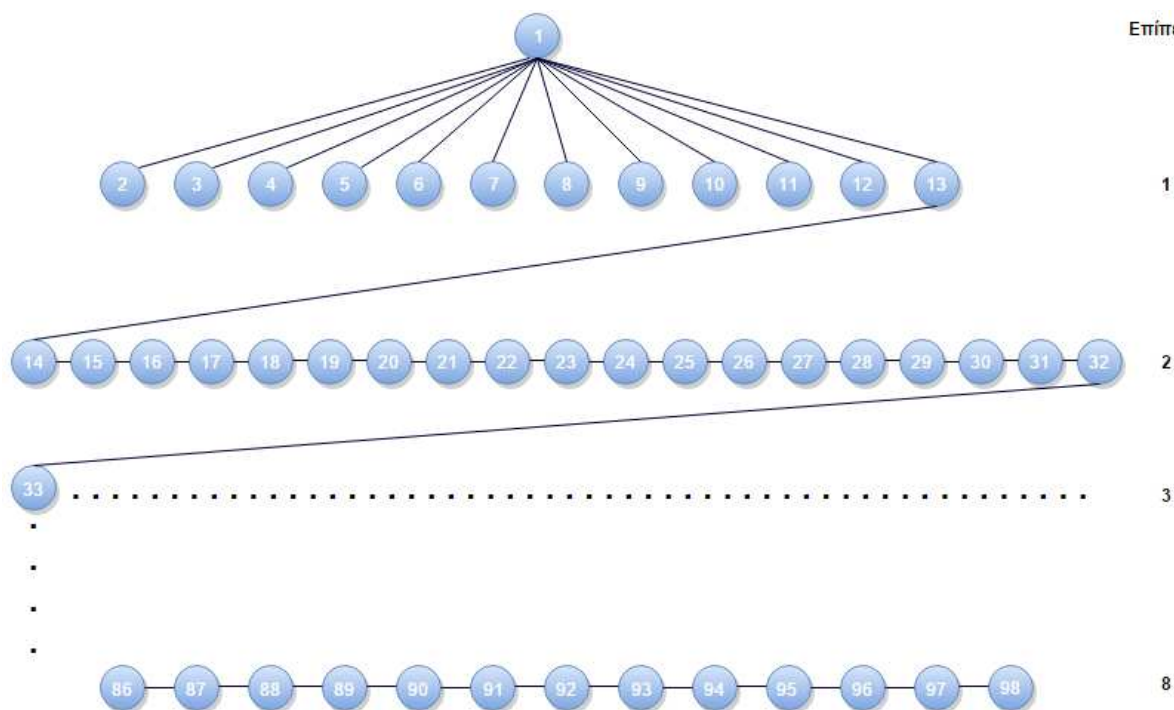
Ο όρος μονός αποκλεισμός από τουρνουά (single elimination ή knockout) χρησιμοποιείται για να προσδιορίσει τα τουρνουά, όπου ο χαμένος από ένα ζευγάρι ομάδων δεν έχει δικαίωμα να συμμετάσχει στη συνέχεια της διοργάνωσης. Υπάρχουν περιπτώσεις όπου η χαμένη ομάδα δεν αποκλείεται εξολοκλήρου από τη συνέχεια της διοργάνωσης και αγωνίζεται πχ για την τρίτη θέση του τουρνουά. Στην περίπτωση της διοργάνωσης του Champions League οι ομάδες που θα χάσουν αποκλείονται από τη συνέχεια των αγώνων γι'αυτό και θα γίνει χρήση του γραφήματος single elimination bracket, όπου τα ζευγάρια των ομάδων καταρτίζονται βάσει κλήρωσης. Παρακάτω παρουσιάζεται το γράφημα του Champions League με τα ζευγάρια της δεύτερης φάσης των 16, της προημιτελικής και της ημιτελικής φάσης:

Champions League



Σχήμα 4.1: Το single elimination bracket για την διοργάνωση του Champions League

Στο γράφημα που επιλέχθηκε ο κάθε κόμβος αντιστοιχεί σε ένα ζευγάρι ομάδων της διοργάνωσης, και μετά επεκτείνεται σε έναν ή περισσότερους κόμβους που αντιστοιχούν στα διάφορα στατιστικά που ανήκουν στα επίπεδα βαρύτητας που έχουν καθοριστεί. Παρακάτω παρουσιάζεται ο γράφος εκτέλεσης του αλγορίθμου κατά Πλάτος:



Κόμβος 1:	Zευγάρι Ομάδων	Κόμβος 24:	Rating
Κόμβος 2:	Totals_Goals_Against	Κόμβος 25:	First_Game_16
Κόμβος 3:	Total_Shots	Κόμβος 26:	First_Game_Quart
Κόμβος 4:	Shots_Per_Game	Κόμβος 27:	Draws
Κόμβος 5:	Total_Goals_For	Κόμβος 28:	Loses
Κόμβος 6:	LongBalls_Per_Game	Κόμβος 29:	Goals_Against
Κόμβος 7:	Left_Side_Attack	Κόμβος 30:	Goals_Difference
Κόμβος 8:	Wins	Κόμβος 31:	Points
Κόμβος 9:	Wins_Percent	Κόμβος 32:	Second_Time_Winner
Κόμβος 10:	First_Game_Sem	Κόμβος 33:	In_6_Yards_Box_Attempts_Against
Κόμβος 11:	Position	Κόμβος 86:	Other_Way_Goals_Against
Κόμβος 12:	Wins	Κόμβος 87:	Yellow_Cards
Κόμβος 13:	Goals_For	Κόμβος 88:	Foul_Card
Κόμβος 14:	Goal_Difference	Κόμβος 89:	Dive_Foul_Card
Κόμβος 15:	Clean_Sheets	Κόμβος 90:	Unprofessional_Foul_Card
Κόμβος 16:	Inside_Area_Goals_Against	Κόμβος 91:	Dive_Foul_Card
Κόμβος 17:	Defensive_Distance	Κόμβος 92:	Other_Card
Κόμβος 18:	Assists	Κόμβος 93:	Othe_Way_Goals_For
Κόμβος 19:	Attempts_On_Target	Κόμβος 94:	Penalty_Goals_For
Κόμβος 20:	Left_Foot_Goals_For	Κόμβος 95:	Own_Goals_For

Κόμβος 21: Offensive_Distance

Κόμβος 96: Action_Zone_Opposition_Half

Κόμβος 22: Loses

Κόμβος 97: Action_Zone_Own_Half

Κόμβος 23: Loses_Percent

Κόμβος 98: Action_Zone_Middle

Σχήμα 4.2: Ο γράφος εκτέλεσης του αλγορίθμου κατά Πλάτος και οι επεξηγήσεις των κόμβων

4.3.3 Επιλογή Στατιστικών

Τα στατιστικά που συγκεντρώθηκαν, καταγράφηκαν και συλλέχθηκαν από τις ιστοσελίδες που παρουσιάστηκαν στο Κεφάλαιο 2. Υπάρχει μία μεγάλη πλειάδα δεδομένων αγωνιστικής απόδοσης τα οποία χωρίζονται σε τέσσερις (4) κατηγορίες: αμυντικά (defensive), επιθετικά (offensive), γενικά (overview), ιστορικά (history). Παρακάτω περιγράφονται συνοπτικά ανά κατηγορία:

Αμυντικά Στατιστικά

Totals_Goals_Against:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που δέχεται η ομάδα.
Average_Conceded:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των γκολ που δέχεται η ομάδα ανά παιχνίδι.
Goal_Difference:	Είναι τύπου int και αναφέρεται στη διαφορά τερμάτων (γκολ που πετυχαίνει - γκολ που δέχεται) της ομάδας.
Clean_Sheets:	Είναι τύπου int και αναφέρεται στον αριθμό των παιχνιδιών που η ομάδα δεν έχει δεχτεί γκολ.
Clean_Sheets_Percent:	Είναι τύπου decimal και αναφέρεται σε τι ποσοστό παιχνιδιών η ομάδα δεν δέχεται γκολ.
Inside_Area_Goals_Against:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που δέχεται η ομάδα μέσα από την περιοχή.
Outside_Area_Goals_Against:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που δέχεται η ομάδα έξω από την περιοχή.
In_6_Yards_Box_Attempts_Against:	Είναι τύπου decimal και αναφέρεται στις προσπάθειες που δέχεται η ομάδα μέσα από τη

	μικρή περιοχή ανά παιχνίδι.
In_18_Yards_Box_Attempts_Against:	Είναι τύπου decimal και αναφέρεται στις προσπάθειες που δέχεται η ομάδα μέσα από τη μεγάλη περιοχή ανά παιχνίδι.
Outside_Of_Box_Attempts_Against:	Είναι τύπου decimal και αναφέρεται στις προσπάθειες που δέχεται η ομάδα έξω από τη μεγάλη περιοχή ανά παιχνίδι.
Total_Shots:	Είναι τύπου int και αναφέρεται στον αριθμό των σουτ που δέχθηκε μία ομάδα.
Shots_Per_Game:	Είναι τύπου decimal και αναφέρεται στα σουτ που δέχεται η ομάδα ανά παιχνίδι.
Right_Foot_Goals_Against:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που δέχεται η ομάδα με το δεξί πόδι.
Left_Foot_Goals_Against:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που δέχεται η ομάδα με το αριστερό πόδι.
Right_Side_Attempts_Against:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που δέχεται η ομάδα από την δεξιά πλευρά.
Left_Side_Attempts_Against:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που δέχεται η ομάδα από την αριστερή πλευρά.
Middle_Side_Attempts_Against:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που δέχεται η ομάδα από τον άξονα (κέντρο) του γηπέδου.
Aerials_Won:	Είναι τύπου decimal και αναφέρεται στις κεφαλές που κερδίζει η ομάδα έναντι του αντιπάλου ανά παιχνίδι.
Interceptions:	Είναι τύπου int και αναφέρεται στο πόσες φορές αποτράπηκε η πάσα της αντίπαλης ομάδας.
Interceptions_Per_Game:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των φορών που αποτρέπεται η πάσα της αντίπαλης ομάδας ανά αγώνα.
Defensive_Errors:	Είναι τύπου int και αναφέρεται στα αμυντικά λάθη της ομάδας που οδηγούν σε σουτ των

	αντιπάλων.
Errors_Leading_To_Goal:	Είναι τύπου int και αναφέρεται στα αμυντικά λάθη της ομάδας που οδηγούν σε γκολ των αντιπάλων.
Total_Corners_Against:	Είναι τύπου int και αναφέρεται στο σύνολο των κόρνερ που παραχωρεί η ομάδα.
Average_Corners_Against:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των κόρνερ που παραχωρεί η ομάδα ανά παιχνίδι.
Red_Cards:	Είναι τύπου int και αναφέρεται στο σύνολο των κόκκινων καρτών που έχει δεχτεί η ομάδα.
Blocked_Shots:	Είναι τύπου int και αναφέρεται στο σύνολο των σουτ που έχει σταματήσει η ομάδα.
Clearances:	Είναι τύπου int και αναφέρεται στο σύνολο των αμυντικών ενεργειών της ομάδας.
Tackles_Per_Game:	Είναι τύπου int και αναφέρεται στο σύνολο των αμυντικών παρεμβάσεων της ομάδας.
Offsides:	Είναι τύπου int και αναφέρεται στο σύνολο των Offsides που βγαίνει ο αντίπαλος.
Offsides_Per_Game:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των Offsides που βγαίνει ο αντίπαλος ανά παιχνίδι.
Fouls_Committed:	Είναι τύπου int και αναφέρεται στο σύνολο των φάουλ που έχει υποπέσει η ομάδα.
Fouls_Per_Game:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των φάουλ που κάνει η ομάδα ανά παιχνίδι.
Header_Goals_Against:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που δέχεται η ομάδα με κεφαλιά.
Other_Way_Goals_Against:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που δέχεται η ομάδα με άλλο τρόπο.
Yellow_Cards:	Είναι τύπου int και αναφέρεται στο σύνολο των κίτρινων καρτών που έχει δεχτεί η ομάδα.
Foul_Card:	Είναι τύπου int και αναφέρεται στο σύνολο των καρτών που έχει δεχτεί η ομάδα από φάουλ.

Unprofessional_Foul_Card:	Είναι τύπου int και αναφέρεται στο σύνολο των καρτών που έχει δεχτεί η ομάδα από αντιαθλητικό φάουλ.
Dive_Foul_Card:	Είναι τύπου int και αναφέρεται στο σύνολο των καρτών που έχει δεχτεί η ομάδα από βουτιά.
Other_Card:	Είναι τύπου int και αναφέρεται στο σύνολο των καρτών που έχει δεχτεί η ομάδα για άλλο λόγο.

Πίνακας 4.3: Ο πίνακας με τα καταγεγραμμένα αμυντικά στατιστικά

Επιθετικά Στατιστικά

Total_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που πετυχαίνει η ομάδα.
Average_Scored_For:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των γκολ που πετυχαίνει η ομάδα ανά παιχνίδι.
Assists:	Είναι τύπου int και αναφέρεται στο σύνολο των πασών για γκολ.
Key_Passes:	Είναι τύπου int και αναφέρεται στο σύνολο των πασών κλειδιών για ευκαιρία.
Open_Play_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ στο ανοιχτό γήπεδο.
Inside_Area_Goals_For:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που πετυχαίνει η ομάδα μέσα από την περιοχή.
Outside_Area_Goals_For:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που πετυχαίνει η ομάδα έξω από την περιοχή.
Fail_To_Score_Percent:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των παιχνιδιών που η ομάδα δεν κατάφερε να σκοράρει.
LongBalls_Per_Game:	Είναι τύπου int και αναφέρεται στο μέσο όρο των μακρινών πασών ανά παιχνίδι.
In_6_Yards_Box_Attempts_For:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που κάνει η ομάδα μέσα από τη μικρή περιοχή ανά παιχνίδι.

In_18_Yards_Box_Attempts_For:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που κάνει η ομάδα μέσα από τη μεγάλη περιοχή ανά παιχνίδι.
Outside_Of_Box_Attempts_For:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που κάνει η ομάδα έξω από τη μεγάλη περιοχή ανά παιχνίδι.
Right_Foot_Goals_For:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που πετυχαίνει η ομάδα με το δεξί πόδι.
Left_Foot_Goals_For:	Είναι τύπου int και αναφέρεται στον αριθμό των γκολ που πετυχαίνει η ομάδα με το αριστερό πόδι.
ShortPasses_Per_Game:	Είναι τύπου int και αναφέρεται στο μέσο όρο των κοντινών πασών ανά παιχνίδι.
Total_Attempts:	Είναι τύπου int και αναφέρεται στον αριθμό των σουτ που έχει επιχειρήσει η ομάδα.
Average_Attempts_Per_Game:	Είναι τύπου decimal και αναφέρεται στα σουτ που κάνει η ομάδα ανά παιχνίδι.
Attempts_On_Target:	Είναι τύπου int και αναφέρεται στον σύνολο των σουτ που έχει επιχειρήσει η ομάδα στην εστία.
Attempts_Off_Target:	Είναι τύπου int και αναφέρεται στον σύνολο των σουτ που έχει επιχειρήσει η ομάδα εκτός εστίας.
Left_Side_Attempts_For:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που επιχειρεί η ομάδα από την αριστερή πλευρά.
Left_Side_Attack:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των επιθετικών ενεργειών που επιχειρεί η ομάδα από την αριστερή πλευρά.
Right_Side_Attempts_For:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που επιχειρεί η ομάδα από την δεξιά πλευρά.
Right_Side_Attack:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των επιθετικών ενεργειών που επιχειρεί η ομάδα από την δεξιά πλευρά.
Middle_Side_Attempts_For:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των προσπαθειών που επιχειρεί η ομάδα από τον

	άξονα.
Middle_Of_The_Pitch_Attack:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των επιθετικών ενεργειών που επιχειρεί η ομάδα από τον άξονα.
Passes_Attempted:	Είναι τύπου int και αναφέρεται στον σύνολο των πασών που έχει κάνει η ομάδα.
Passes_Completed:	Είναι τύπου int και αναφέρεται στον σύνολο των πασών που ολοκληρώθηκαν από την ομάδα.
Passes_Completion:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των επιτυχημένων πασών.
Ball_Possession:	Είναι τύπου decimal και αναφέρεται στο ποσοστό κατοχής μπάλας.
ThroughBall_Per_Game:	Είναι τύπου int και αναφέρεται στο μέσο όρο των πασών στην άμυνα του αντιπάλου ανά παιχνίδι.
Total_Crosses:	Είναι τύπου int και αναφέρεται στο σύνολο των πασών προς μία κεντρική επιθετική γραμμή.
Crosses_Per_Game:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των πασών προς μία κεντρική επιθετική γραμμή ανά παιχνίδι.
Dribbles_Per_Game:	Είναι τύπου int και αναφέρεται στο μέσο των περασμάτων με την μπάλα ανά παιχνίδι.
Fouls_Suffered:	Είναι τύπου int και αναφέρεται στο σύνολο των φάουλ που κερδίζει η ομάδα.
Set_Piece_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που έχει πετύχει η ομάδα από στημένες φάσεις.
Header_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που έχει πετύχει η ομάδα με κεφαλιές.
Counter_Attack_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που έχει πετύχει η ομάδα με αντεπιθέσεις.
Penalty_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που έχει πετύχει η ομάδα με πέναλτι.
Own_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που έχει πετύχει η ομάδα με αυτογκόλ.
Othe_Way_Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των

	γκολ που έχει πετύχει η ομάδα με άλλο τρόπο
Total_Corners_For:	Είναι τύπου int και αναφέρεται στο σύνολο των κόρνερ που έχει κερδίσει η ομάδα.
Average_Corners_For:	Είναι τύπου decimal και αναφέρεται στο μέσο όρο των κόρνερ που κερδίζει η ομάδα ανά παιχνίδι.
Action_Zone_Opposition_Half:	Είναι τύπου decimal και αναφέρεται στο ποσοστό που η κατοχή της μπάλας είναι στην περιοχή του αντιπάλου.
Action_Zone_Own_Half:	Είναι τύπου decimal και αναφέρεται στο ποσοστό που η κατοχή της μπάλας είναι στην περιοχή της ομάδας.
Action_Zone_Middle:	Είναι τύπου decimal και αναφέρεται στο ποσοστό που η κατοχή της μπάλας είναι στο κέντρο του γηπέδου.

Πίνακας 4.4: Ο πίνακας με τα καταγεγραμμένα επιθετικά στατιστικά

Γενικά Στατιστικά

Wins:	Είναι τύπου int και αναφέρεται στο σύνολο των νικών της ομάδας.
Wins_Percent:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των νικών της ομάδας προς το σύνολο των παιχνιδιών.
Loses:	Είναι τύπου int και αναφέρεται στο σύνολο των ηττών της ομάδας.
Loses_Percent:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των ηττών της ομάδας προς το σύνολο των παιχνιδιών.
Draws:	Είναι τύπου int και αναφέρεται στο σύνολο των ισοπαλιών της ομάδας.
Draws_Percent:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των ισοπαλιών της ομάδας προς το σύνολο των παιχνιδιών.
Rating:	Είναι τύπου decimal και αναφέρεται στη βαθμολογία της ομάδας.
Matches_Played:	Είναι τύπου int και αναφέρεται στο σύνολο των παιχνιδιών

	της ομάδας.
Points:	Είναι τύπου int και αναφέρεται στο σύνολο των πόντων που έχει κερδίσει η ομάδα.
Points_Per_Game:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των πόντων που κερδίζει η ομάδα ανά παιχνίδι.
Possible_Points_Percent:	Είναι τύπου decimal και αναφέρεται στο ποσοστό των πόντων που κερδίζει η ομάδα προς το σύνολο των πιθανών πόντων.

Πίνακας 4.5: Ο πίνακας με τα καταγεγραμμένα γενικά στατιστικά

Ιστορικά Στατιστικά

Position:	Είναι τύπου int και αναφέρεται στη θέση της ομάδας.
Wins:	Είναι τύπου int και αναφέρεται στο σύνολο των νικών της ομάδας στην ιστορία της διοργάνωσης.
Draws:	Είναι τύπου int και αναφέρεται στο σύνολο των ισοπαλιών της ομάδας στην ιστορία της διοργάνωσης.
Loses:	Είναι τύπου int και αναφέρεται στο σύνολο των ηττών της ομάδας στην ιστορία της διοργάνωσης.
Goals_For:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ της ομάδας.
Goals_Against:	Είναι τύπου int και αναφέρεται στο σύνολο των γκολ που έχει δεχτεί η ομάδα.
Goals_Difference:	Είναι τύπου int και αναφέρεται στη διαφορά τερμάτων της ομάδας.
Points:	Είναι τύπου int και αναφέρεται στο σύνολο των πόντων της ομάδας.
Participations:	Είναι τύπου int και αναφέρεται στο σύνολο των συμμετοχών της ομάδας στη διοργάνωση.

Games_Played:	Είναι τύπου int και αναφέρεται στο σύνολο των παιχνιδιών της ομάδας.
Participations_In_Knockout_Phase:	Είναι τύπου int και αναφέρεται στο σύνολο των συμμετοχών της ομάδας στη knockout φάση.
Games_In_The_Knockout_Phase:	Είναι τύπου int και αναφέρεται στο σύνολο των παιχνιδιών της ομάδας στη knockout φάση.
Successive_Appearances_In_The_KnockOut_Phase:	Είναι τύπου int και αναφέρεται στο σύνολο των επιτυχημένων παρουσιών στη knockout φάση.
Appearances_In_The_Quarter_Finals:	Είναι τύπου int και αναφέρεται στον αριθμό των συμμετοχών στα προημιτελικά.
Games_In_The_Quarter_Finals:	Είναι τύπου int και αναφέρεται στο σύνολο των παιχνιδιών στα προημιτελικά.
Successive_Appearances_In_The_Quarter_Finals:	Είναι τύπου int και αναφέρεται στον αριθμό των επιτυχημένων συμμετοχών στα προημιτελικά.
Appearances_In_The_Semi_Finals:	Είναι τύπου int και αναφέρεται στον αριθμό των συμμετοχών στα ημιτελικά.
Games_In_The_Semi_Finals:	Είναι τύπου int και αναφέρεται στο σύνολο των παιχνιδιών στα ημιτελικά.
Final_Appearances:	Είναι τύπου int και αναφέρεται στον αριθμό των συμμετοχών στον τελικό.
Titles:	Είναι τύπου int και αναφέρεται στο σύνολο των τίτλων της ομάδας.
Final_Loses:	Είναι τύπου int και αναφέρεται στον αριθμό των ηττών στον τελικό.

Πίνακας 4.6: Ο πίνακας με τα καταγεγραμμένα ιστορικά στατιστικά

4.3.4 Καταγραφή Συχνότητας Εμφάνισης Στατιστικών

Για τον καθορισμό της βαρύτητας των στατιστικών πραγματοποιήθηκε καταγραφή των συχνοτήτων εμφάνισης τους από την ιστοσελίδα της UEFA, για τα παιχνίδια των ομίλων, μόνο

όμως για τις ομάδες που προκρίθηκαν στην φάση των δέκα έξι (16). Η συχνότητα επιτυχημένης εμφάνισης καθορίζεται από τη νίκη μίας ομάδας. Αν η ομάδα που κέρδιζε είχε και μεγαλύτερη τιμή για το εξεταζόμενο στατιστικό τότε αυξάνονταν η συχνότητα εμφάνισης του. Βέβαια, υπάρχουν περιπτώσεις όπου η τιμή του στατιστικού πρέπει να είναι μικρότερη για να αυξηθεί η συχνότητά του (κυρίως σε αμυντικά χαρακτηριστικά). Έτσι, από σύνολο ενενήντα έξι (96) παιχνιδιών στη φάση των ομίλων η καταγραφή έγινε για τους ογδόντα (80) αγώνες. Συνοπτικά παρουσιάζονται παρακάτω τα ποσοστά εμφάνισης των αμυντικών και επιθετικών στατιστικών:

Totals_Goals_Against: 80%	Errors_Leading_To_Goal: 48%
Average_Conceded: 80%	Total_Corners_Against: 48%
Goal_Difference: 70%	Average_Corners_Against: 48%
Inside_Area_Goals_Against: 69%	Red_Cards: 47%
In_6_Yards_Box_Attempts_Against: 65%	Blocked_Shots: 46%
In_18_Yards_Box_Attempts_Against: 65%	Foul_Card: 44%
Right_Foot_Goals_Against: 64%	Clearances: 43%
Clean_Sheets: 63%	Tackles_Per_Game: 43%
Clean_Sheets_Percent: 63%	Header_Goals_Against: 41%
Total_Shots: 60%	Outside_Area_Goals_Against: 41%
Shots_Per_Game: 60%	Yellow_Cards: 39%
Left_Foot_Goals_Against: 60%	Other_Way_Goals_Against: 37%
Left_Side_Attempts_Against: 57%	Offsides: 35%
Right_Side_Attempts_Against: 56%	Offsides_Per_Game: 35%
Aerials_Won: 55%	Fouls_Committed: 32%
Middle_Side_Attempts_Against: 54%	Fouls_Per_Game: 32%
Outside_Of_Box_Attempts_Against: 53%	Unprofessional_Foul_Card: 29%
Interceptions: 50%	Other_Card: 27%
Interceptions_Per_Game: 50%	Dive_Foul_Card: 25%
Defensive_Errors: 48%	

Πίνακας 4.7: Ο πίνακας με τις συχνότητες εμφάνισης των αμυντικών στατιστικών

Total_Goals_For: 80%	Passes_Completed: 54%
Average_Scored_For: 80%	Middle_Side_Attempts_For: 54%
Assists: 71%	Middle_Of_The_Pitch_Attack: 54%
Open_Play_Goals_For: 71%	Ball_Possession: 54%
Inside_Area_Goals_For: 69%	Outside_Of_Box_Attempts_For: 53%
LongBalls_Per_Game: 65%	ThroughBall_Per_Game: 51%
In_6_Yards_Box_Attempts_For: 65%	Total_Crosses: 50%
In_18_Yards_Box_Attempts_For: 65%	Crosses_Per_Game: 50%
Right_Foot_Goals_For: 64%	Fouls_Suffered: 50%
Fail_To_Score_Percent: 63%	Total_Corners_For: 48%
Key_Passes: 62%	Average_Corners_For: 48%
Left_Foot_Goals_For: 62%	Attempts_Off_Target: 46%
ShortPasses_Per_Game: 61%	Set_Piece_Goals_For: 45%
Total_Attempts: 61%	Header_Goals_For: 44%
Average_Attempts_Per_Game: 61%	Counter_Attack_Goals_For: 44%
Attempts_On_Target: 60%	Outside_Area_Goals_For: 42%
Left_Side_Attempts_For: 60%	Penalty_Goals_For: 38%
Left_Side_Attack: 60%	Action_Zone_Opposition_Half: 35%
Right_Side_Attempts_For: 56%	Othe_Way_Goals_For: 33%
Right_Side_Attack: 56%	Own_Goals_For: 32%
Passes_Completion: 55%	Action_Zone_Own_Half: 31%
Passes_Attempted: 54%	Action_Zone_Middle: 29%
Dribbles_Per_Game: 50%	

Πίνακας 4.8: Ο πίνακας με τις συχνότητες εμφάνισης των επιθετικών στατιστικών

4.3.5 Συνδυασμός Στατιστικών

Τα στατιστικά θα χωριστούν σε επίπεδα σύγκρισης ανάλογα με το ποσοστό εμφάνισής τους. Συγκεκριμένα, τα στατιστικά που παρουσιάζουν συχνότητα εμφάνισης από 60% και πάνω θα ανήκουν στα πρώτα επίπεδα σύγκρισης και θα συγκρίνονται ανεξάρτητα το ένα από το άλλο. Έτσι, τα αμυντικά στατιστικά που ανήκουν σε αυτήν την κατηγορία είναι:

Totals_Goals_Against	Right_Foot_Goals_Against
Average_Conceded	Clean_Sheets
Goal_Difference	Clean_Sheets_Percent
Inside_Area_Goals_Against	Total_Shots
In_6_Yards_Box_Attempts_Against	Shots_Per_Game
In_18_Yards_Box_Attempts_Against	Left_Foot_Goals_Against

Πίνακας 4.9: Ο πίνακας με τα αμυντικά στατιστικά με ποσοστό εμφάνισης από 60% και πάνω

Αντίστοιχα τα επιθετικά στατιστικά που ανήκουν σε αυτήν την κατηγορία είναι:

Total_Goals_For	Fail_To_Score_Percent
Average_Scored_For	Key_Passes
Assists	Left_Foot_Goals_For
Open_Play_Goals_For	ShortPasses_Per_Game
Inside_Area_Goals_For	Total_Attempts
LongBalls_Per_Game	Average_Attempts_Per_Game
In_6_Yards_Box_Attempts_For	Attempts_On_Target
In_18_Yards_Box_Attempts_For	Left_Side_Attempts_For
Right_Foot_Goals_For	Left_Side_Attack

Πίνακας 4.10: Ο πίνακας με τα επιθετικά στατιστικά με ποσοστό εμφάνισης από 60% και πάνω

Τα παραπάνω στατιστικά τοποθετήθηκαν στα τέσσερα (4) πρώτα επίπεδα σύγκρισης που είναι τα παρακάτω:

Επίπεδα	Αμυντικά	Επιθετικά
Επίπεδο 1	Totals_Goals_Against, Total_Shots, Shots_Per_Game	Total_Goals_For, LongBalls_Per_Game, Left_Side_Attack
Επίπεδο 2	Goal_Difference, Clean_Sheets,	Assists, Attempts_On_Target,

	Inside_Area_Goals_Against	Left_Foot_Goals_For
Επίπεδο 3	In_6_Yards_Box_Attempts_Against, In_18_Yards_Box_Attempts_Against, Left_Foot_Goals_Against, Average_Conceded	Average_Scored_For, Open_Play_Goals_For, Inside_Area_Goals_For, Right_Foot_Goals_For, In_6_Yards_Box_Attempts_For, In_18_Yards_Box_Attempts_For, ShortPasses_Per_Game, Left_Side_Attempts_For
Επίπεδο 4	Clean_Sheets_Percent, Right_Foot_Goals_For	Fail_To_Score_Percent, Key_Passes, Total_Attempts, Average_Attempts_Per_Game,

Πίνακας 4.11: Ο πίνακας με τα στατιστικά στα πρώτα τέσσερα επίπεδα σύγκρισης

Για τα υπόλοιπα στατιστικά με συχνότητα εμφάνισης κάτω από 60%, αναζητήθηκαν κάποιοι συσχετισμοί - μοτίβα (patterns), όπου συνδυάζονται κάποια από αυτά για να αυξηθεί η συχνότητα εμφάνισής τους. Για της ανάγκες της εφαρμογής δημιουργήθηκαν κάποιες τετράδες στατιστικών. Παρακάτω φαίνονται οι συσχετισμοί, τα ποσοστά εμφάνισης και τα επίπεδα στα οποία ανήκουν:

Επίπεδα	Αμυντικά
Επίπεδο 5	Right_Side_Attempts_Against - Middle_Side_Attempts_Against - Outside_Of_Box_Attempts_Against - Defensive_Errors: 77%
Επίπεδο 5	Left_Side_Attempts_Against - Total_Corners_Against - Average_Corners_Against - Errors_Leading_To_Goal: 75%
Επίπεδο 6	Aerials_Won - Interceptions - Interceptions_Per_Game - Clearances:70%
Επίπεδο 7	Blocked_Shots - Tackles_Per_Game - Offsides - Offsides_Per_Game: 64%
Επίπεδο 7	Red_Cards - Outside_Area_Goals_Against - Fouls_Committed - Fouls_Per_Game: 62%
Επίπεδα	Επιθετικά
Επίπεδο 5	Passes_Completion_Rate - Right_Side_Attempts_For -

Επίπεδο 5	Total_Crosses - Middle_Side_Attempts_For: 78%
Επίπεδο 6	Ball_Possession - Fouls_Suffered - Attempts_Off_Target - Total_Corners_For: 76%
Επίπεδο 6	Passes_Attempted - Passes_Completed - Crosses_Per_Game - Average_Corners_For: 71%
Επίπεδο 7	Right_Side_Attack - Middle_Of_The_Pitch_Attack - ThroughBall_Per_Game - Dribbles_Per_Game: 69%
Επίπεδο 7	Outside_Of_Box_Attempts_For - Set_Piece_Goals_For - Outside_Area_Goals_For - Counter_Attack_Goals_For: 63%

Πίνακας 4.12: Ο πίνακας με τα μοτίβα στατιστικών από τα επίπεδα σύγκρισης 5 με 7

Υπήρχαν βέβαια και κάποιοι συσχετισμοί στατιστικών που είχαν ποσοστό εμφάνισης κάτω από 60%, οπότε αποφασίστηκε να προστεθούν στο όγδοο και τελευταίο επίπεδο σύγκρισης και θα είναι ανεξάρτητα το ένα από το άλλο. Συνοπτικά παρουσιάζονται:

Επίπεδα	Αμυντικά	Επιθετικά
Επίπεδο 8	Other_Way_Goals_Against, Yellow_Cards, Foul_Card, Dive_Foul_Card, Unprofessional_Foul_Card, Dive_Foul_Card, Other_Card	Othe_Way_Goals_For, Penalty_Goals_For, Own_Goals_For, Action_Zone_Opposition_Half, Action_Zone_Own_Half, Action_Zone_Middle

Πίνακας 4.13: Ο πίνακας με τα στατιστικά στο όγδοο επίπεδο σύγκρισης

Για τις υπόλοιπες δύο (2) κατηγορίες στατιστικών, τα γενικά και τα ιστορικά αποφασίστηκε η ανεξάρτητη σύγκριση των στατιστικών και ο διαχωρισμός τους στα παρακάτω επίπεδα σύγκρισης:

Επίπεδα	Γενικά	Ιστορικά
Επίπεδο 1	Wins, Wins_Percent	Position, Wins, Goals_For
Επίπεδο 2	Loses, Loses_Percent, Rating	Draws, Loses, Goals_Against, Goals_Difference, Points
Επίπεδο 3	Draws, Draws_Percent,	Participations, Titles, Games_Played

Matches_Played		
Επίπεδο 4	Points, Points_Per_Game, Possible_Points_Percent	Appearances_In_The_Quarter_Finals, Games_In_The_Quarter_Finals, Successive_Appearances_In_The_Quarter_Finals, Appearances_In_The_Semi_Finals, Games_In_The_Semi_Finals, Final_Appearances, Titles, Final_Loses
Επίπεδο 5		Participations_In_Knockout_Phase, Games_In_The_Knockout_Phase, Successive_Appearances_In_The_KnockOut_Phase

Πίνακας 4.14: Ο πίνακας με τα γενικά και ιστορικά στατιστικά και τα αντίστοιχα επίπεδα σύγκρισης

Οι παραπάνω κατηγορίες στατιστικών (αμυντικά, επιθετικά, γενικά, ιστορικά) θα έχουν ένα ποσοστό βαρύτητας και τα αντίστοιχα δεδομένα της κάθε κατηγορίας θα έχουν μία αντίστοιχη βαρύτητα ανάλογα με το επίπεδο σύγκρισης στο οποίο βρίσκονται. Συνοπτικά, έχουμε:

Επίπεδα	Αμυντικά:	Επιθετικά:	Γενικά:	Ιστορικά:
	45%	35%	15%	5%
Επίπεδο 1	22%	22%	35%	30%
Επίπεδο 2	19%	19%	30%	25%
Επίπεδο 3	16%	16%	20%	20%
Επίπεδο 4	14%	14%	15%	15%
Επίπεδο 5	12%	12%		10%
Επίπεδο 6	9%	9%		
Επίπεδο 7	5%	5%		
Επίπεδο 8	3%	3%		
Σύνολο Επιπέδων	100%	100%	100%	100%

Πίνακας 4.15: Ο πίνακας τα ποσοστά των κατηγοριών και τα αντίστοιχα των επιπέδων τους

4.3.6 Εισαγωγή Προσωπικών Στατιστικών

Για τις ανάγκες της μεταπτυχιακής διατριβής ζητήθηκε η εισαγωγή ενός ή περισσότερων προσωπικών στατιστικών για την εξαγωγή εγκυρότερου αποτελέσματος. Συγκεκριμένα επιλέχθηκαν στατιστικά και για τις τέσσερις κατηγορίες που παρουσιάστηκαν παραπάνω.

Για την κατηγορία των αμυντικών στατιστικών επιλέχθηκαν:

Goals_Conceded_Per_Shoot:	Είναι τύπου decimal και αναφέρεται στα γκολ που δέχεται η ομάδα ανά προσπάθεια (σουτ) του αντιπάλου.
Def_Distance_Covered:	Είναι τύπου decimal και αναφέρεται και αναφέρεται στην απόσταση (χιλιόμετρα) που έχουν διανύσει οι τέσσερις (4) κορυφαίοι αμυντικοί της ομάδας, στους αγώνες του Champions League.
Avg_Def_Distance_Covered:	Είναι τύπου decimal και αναφέρεται και αναφέρεται στην μέση απόσταση (χιλιόμετρα) που διανύουν οι τέσσερις (4) κορυφαίοι αμυντικοί της ομάδας, ανά αγώνα του Champions League.

Πίνακας 4.16: Ο πίνακας με τα προσωπικά αμυντικά στατιστικά

Για την κατηγορία των επιθετικών στατιστικών επιλέχθηκαν:

Goal_Per_Attempt:	Είναι τύπου decimal και αναφέρεται στα γκολ που πετυχαίνει η ομάδα ανά προσπάθεια (σουτ).
Goal_Per_Attempt_On_Target:	Είναι τύπου decimal και αναφέρεται στα γκολ που πετυχαίνει η ομάδα ανά προσπάθεια (σουτ) στην εστία.
Off_Distance_Covered:	Είναι τύπου decimal και αναφέρεται και αναφέρεται στην απόσταση (χιλιόμετρα)

που έχουν διανύσει οι πέντε (5) κορυφαίοι μέσο-επιθετικοί της ομάδας, στους αγώνες του Champions League.

Avg_Off_Distance_Covered:

Είναι τύπου decimal και αναφέρεται και αναφέρεται στην μέση απόσταση (χιλιόμετρα) που διανύουν οι πέντε (5) κορυφαίοι μέσο-επιθετικοί της ομάδας, ανά αγώνα του Champions League.

Πίνακας 4.17: Ο πίνακας με τα προσωπικά επιθετικά στατιστικά

Για την κατηγορία των γενικών στατιστικών επιλέχθηκαν:

Second_Game_16:

Είναι τύπου int και αναφέρεται στο πλεονέκτημα της ομάδας που αγωνίζεται στην έδρα της το δεύτερο παιχνίδι της β' φάσης.

Second_Game_Quart:

Είναι τύπου int και αναφέρεται στο πλεονέκτημα της ομάδας που αγωνίζεται στην έδρα της το δεύτερο παιχνίδι των προημιτελικών.

First_Game_Sem:

Είναι τύπου int και αναφέρεται στο πλεονέκτημα της ομάδας που αγωνίζεται στην έδρα της στο πρώτο παιχνίδι των ημιτελικών.

Πίνακας 4.18: Ο πίνακας με τα προσωπικά γενικά στατιστικά

Για την κατηγορία των ιστορικών στατιστικών επιλέχθηκαν:

Second_Time_Winner:

Είναι τύπου int και αναφέρεται στο γεγονός ότι η ομάδα δεν μπορεί να κερδίσει για δεύτερη συνεχόμενη χρονιά το Champions League.

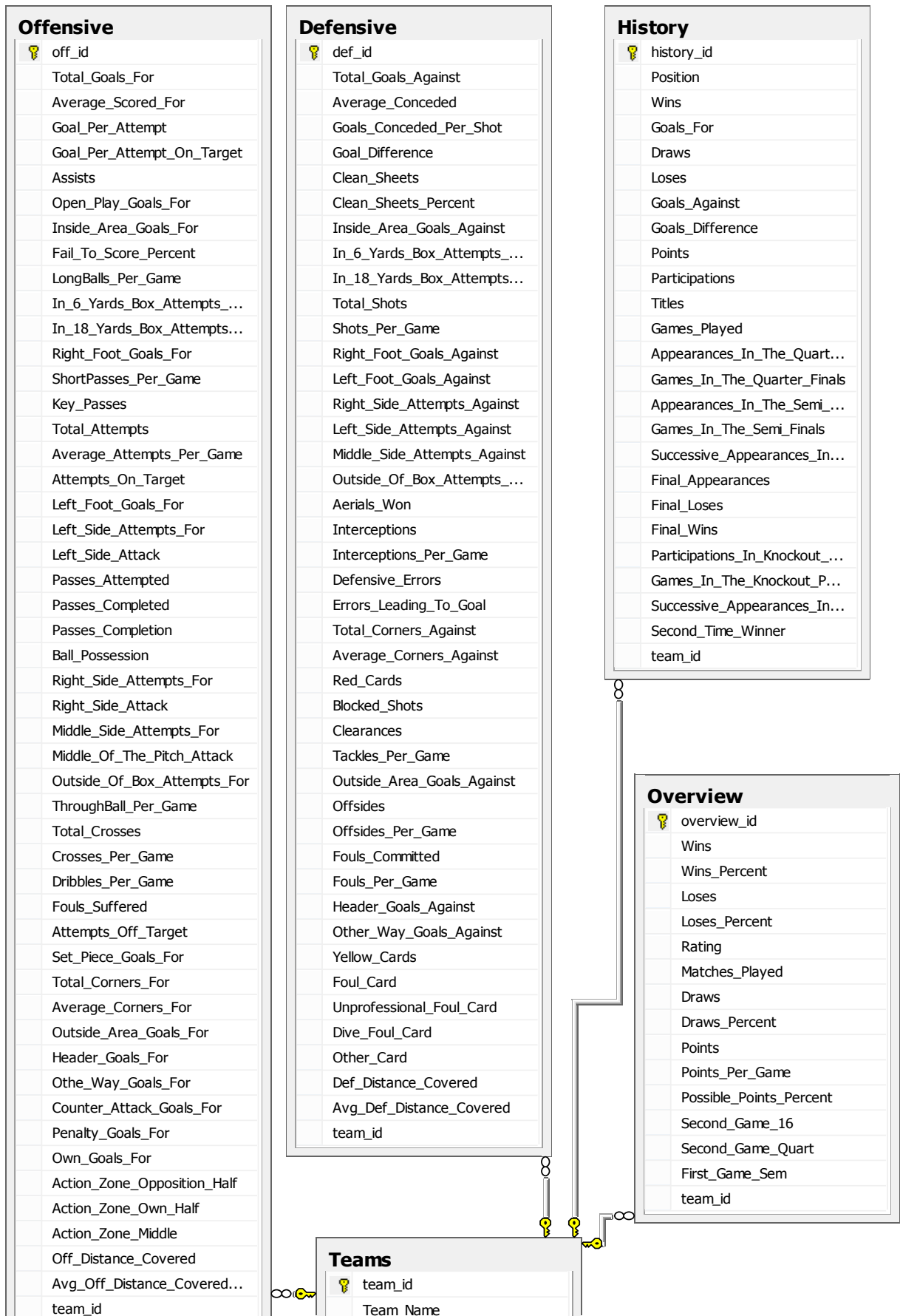
Πίνακας 4.19: Ο πίνακας με τα προσωπικά ιστορικά στατιστικά

Στη συνέχεια, επιλέχθηκαν και τα επίπεδα σύγκρισης στα οποία θα τοποθετηθούν τα προσωπικά στατιστικά. Συγκεκριμένα, τα αμυντικά δεδομένα `Def_Distance_Covered` και `Avg_Def_Distance_Covered` τοποθετήθηκαν στο δεύτερο (2) και τρίτο (3) επίπεδο σύγκρισης αντίστοιχα, ενώ το `Goals_Conceded_Per_Shoot` τοποθετήθηκε στο τέταρτο (4). Για τα επιθετικά στατιστικά `Off_Distance_Covered`, `Avg_Off_Distance_Covered` τοποθετήθηκαν στο δεύτερο (2) και τρίτο (3) επίπεδο σύγκρισης αντίστοιχα, ενώ τα `Goal_Per_Attempt`, `Goal_Per_Attempt_On_Target` τοποθετήθηκαν στο τέταρτο (4) επίπεδο. Στα γενικά χαρακτηριστικά το `First_Game_Sem` τοποθετήθηκε στο πρώτο (1) επίπεδο σύγκρισης και οι δύο άλλες προσωπικές επιλογές δηλαδή, `Second_Game_16`, `Second_Game_Quart`, τοποθετήθηκαν στο δεύτερο (2) επίπεδο σύγκρισης. Τέλος, για τα ιστορικά δεδομένα το προσωπικό στατιστικό `Second_Time_Winner` βρίσκεται στο δεύτερο (2) επίπεδο.

4.4 Περιγραφή Βάσης Δεδομένων

Όπως αναφέρθηκε παραπάνω για τη δημιουργία της βάσης δεδομένων χρησιμοποιήθηκε το εργαλείο Microsoft SQL Server Management Studio. Η βάση δεδομένων αποτελείται από πέντε (5) πίνακες τον `Defensive`, τον `Offensive`, τον `Overview`, τον `History` και τον `Teams`. Το backup της βάσης δεδομένων `FootballStats`, με τους πίνακες και τα καταχωρημένα στατιστικά περιέχεται στο παράρτημα Α.2, αποθηκευμένα στο σχετικό `cd` που συνοδεύει το κείμενο της μεταπτυχιακής διατριβής.


Παρακάτω παρουσιάζεται το διάγραμμα της βάσης δεδομένων και οι συσχετισμοί μεταξύ των πινάκων που την αποτελούν και στη συνέχεια περιγράφονται οι πίνακες που αποτελούν τη βάση δεδομένων της εφαρμογής.



Σχήμα 4.20: Το διάγραμμα της Βάσης Δεδομένων

4.4.1 Πίνακας Teams

Είναι ο πίνακας που αποθηκεύει τα ονόματα των ομάδων που συμμετέχουν στη δεύτερη φάση του Champions League. Έχει πρωτεύον κλειδί το πεδίο team_id, το οποίο είναι auto increment παίρνοντας αυτόματα αυξανόμενες τιμές και έχει ακόμα ένα ακόμη πεδίο το Team_Name το οποίο είναι τύπου nchar και περιγράφει τα ονόματα των ομάδων.

	Column Name	Data Type	Allow Nulls
	team_id	int	<input type="checkbox"/>
	Team_Name	nchar(100)	<input type="checkbox"/>
			<input type="checkbox"/>

Σχήμα 4.21: Ο πίνακας Teams της Βάσης Δεδομένων

4.4.2 Πίνακας Defensive

Είναι ο πίνακας που αποθηκεύει τα αμυντικά στατιστικά των ομάδων που περιγράφηκαν παραπάνω. Τα αριθμητικά αμυντικά δεδομένα όπως αναφέρθηκε και παραπάνω είναι το άθροισμα των τιμών κάθε στατιστικού από την παρουσία της ομάδας στο εγχώριο πρωτάθλημα και στο Champions League. Ο πίνακας έχει πρωτεύον κλειδί το def_id το οποίο είναι auto increment παίρνοντας αυτόματα αυξανόμενες τιμές και δευτερεύον κλειδί έχει το πεδίο team_id το οποίο χρησιμοποιείται για το συσχετισμό (relationship) και την σύνδεση (union) του πίνακα Defensive με τον πίνακα Teams.

	Column Name	Data Type	Allow Nulls
🔑	def_id	int	<input type="checkbox"/>
	Total_Goals_Against	int	<input type="checkbox"/>
	Average_Conceded	decimal(4, 2)	<input type="checkbox"/>
	Goals_Conceded_Per...	decimal(3, 2)	<input type="checkbox"/>
	Goal_Difference	int	<input type="checkbox"/>
	Clean_Sheets	int	<input type="checkbox"/>
	Clean_Sheets_Percent	decimal(3, 2)	<input type="checkbox"/>
	Inside_Area_Goals_A...	int	<input type="checkbox"/>
	In_6_Yards_Box_Atte...	decimal(3, 2)	<input type="checkbox"/>
	In_18_Yards_Box_Att...	decimal(3, 2)	<input type="checkbox"/>
	Total_Shots	int	<input type="checkbox"/>
	Shots_Per_Game	decimal(3, 1)	<input type="checkbox"/>
	Right_Foot_Goals_Ag...	int	<input type="checkbox"/>
	Left_Foot_Goals_Agai...	int	<input type="checkbox"/>
	Right_Side_Attempts_...	decimal(3, 2)	<input type="checkbox"/>

Σχήμα 4.22: Ο πίνακας Defensive της Βάσης Δεδομένων

4.4.3 Πίνακας Offensive


Είναι ο πίνακας που αποθηκεύει τα επιθετικά στατιστικά των ομάδων που περιγράφηκαν παραπάνω. Τα αριθμητικά αμυντικά δεδομένα όπως αναφέρθηκε και παραπάνω είναι το άθροισμα των τιμών κάθε στατιστικού από την παρουσία της ομάδας στο εγχώριο πρωτάθλημα και στο Champions League. Ο πίνακας έχει πρωτεύον κλειδί το off_id το οποίο είναι auto increment παίρνοντας αυτόματα αυξανόμενες τιμές και δευτερεύον κλειδί έχει το πεδίο team_id το οποίο χρησιμοποιείται για το συσχετισμό (relationship) και την σύνδεση (union) του πίνακα Defensive με τον πίνακα Teams.

	Column Name	Data Type	Allow Nulls
🔑	off_id	int	<input type="checkbox"/>
	Total_Goals_For	int	<input type="checkbox"/>
	Average_Scored_For	decimal(4, 2)	<input type="checkbox"/>
	Goal_Per_Attempt	decimal(3, 2)	<input type="checkbox"/>
	Goal_Per_Attempt_O...	decimal(3, 2)	<input type="checkbox"/>
	Assists	int	<input type="checkbox"/>
	Open_Play_Goals_For	int	<input type="checkbox"/>
	Inside_Area_Goals_For	int	<input type="checkbox"/>
	Fail_To_Score_Percent	decimal(3, 2)	<input type="checkbox"/>
	LongBalls_Per_Game	int	<input type="checkbox"/>
	In_6_Yards_Box_Atte...	decimal(3, 2)	<input type="checkbox"/>
	In_18_Yards_Box_Att...	decimal(3, 2)	<input type="checkbox"/>
	Right_Foot_Goals_For	int	<input type="checkbox"/>
	ShortPasses_Per_Game	int	<input type="checkbox"/>
	Key_Passes	int	<input type="checkbox"/>

Σχήμα 4.23: Ο πίνακας Offensive της Βάσης Δεδομένων

4.4.4 Πίνακας Overview


Είναι ο πίνακας που αποθηκεύει τα γενικά στατιστικά των ομάδων που περιγράφηκαν παραπάνω. Τα αριθμητικά αμυντικά δεδομένα όπως αναφέρθηκε και παραπάνω είναι το άθροισμα των τιμών κάθε στατιστικού από την παρουσία της ομάδας στο εγχώριο πρωτάθλημα και στο Champions League. Ο πίνακας έχει πρωτεύον κλειδί το overview_id το οποίο είναι auto increment παίρνοντας αυτόματα αυξανόμενες τιμές και δευτερεύον κλειδί έχει το πεδίο team_id το οποίο χρησιμοποιείται για το συσχετισμό (relationship) και την σύνδεση (union) του πίνακα Defensive με τον πίνακα Teams.

 overview_id	int	<input type="checkbox"/>
Wins	int	<input type="checkbox"/>
Wins_Percent	decimal(3, 2)	<input type="checkbox"/>
Loses	int	<input type="checkbox"/>
Loses_Percent	decimal(3, 2)	<input type="checkbox"/>
Rating	decimal(4, 2)	<input type="checkbox"/>
Matches_Played	int	<input type="checkbox"/>
Draws	int	<input type="checkbox"/>
Draws_Percent	decimal(3, 2)	<input type="checkbox"/>
Points	int	<input type="checkbox"/>
Points_Per_Game	decimal(3, 2)	<input type="checkbox"/>
Possible_Points_Percent	decimal(3, 2)	<input type="checkbox"/>
Second_Game_16	int	<input type="checkbox"/>
Second_Game_Quart	int	<input type="checkbox"/>
team_id	int	<input type="checkbox"/>

Σχήμα 4.24: Ο πίνακας Overview της Βάσης Δεδομένων

4.4.5 Πίνακας History

Είναι ο πίνακας που αποθηκεύει τα ιστορικά στατιστικά των ομάδων που περιγράφηκαν παραπάνω. Έχει πρωτεύον κλειδί το history_id το οποίο είναι auto increment παίρνοντας αυτόματα αυξανόμενες τιμές και δευτερεύον κλειδί έχει το πεδίο team_id το οποίο χρησιμοποιείται για το συσχετισμό (relationship) και την σύνδεση (union) του πίνακα Defensive με τον πίνακα Teams.

 history_id	int	<input type="checkbox"/>
Position	int	<input type="checkbox"/>
Wins	int	<input type="checkbox"/>
Goals_For	int	<input type="checkbox"/>
Draws	int	<input type="checkbox"/>
Loses	int	<input type="checkbox"/>
Goals_Against	int	<input type="checkbox"/>
Goals_Difference	int	<input type="checkbox"/>
Points	int	<input type="checkbox"/>
Participations	int	<input type="checkbox"/>
Titles	int	<input type="checkbox"/>
Games_Played	int	<input type="checkbox"/>
Appearances_In_The_Quarter_Finals	int	<input type="checkbox"/>
Games_In_The_Quarter_Finals	int	<input type="checkbox"/>
Appearances_In_The_Semi_Finals	int	<input type="checkbox"/>

Σχήμα 4.25: Ο πίνακας History της Βάσης Δεδομένων

4.5 Ανάλυση και Παρουσίαση του Κώδικα

Έχοντας περιγράψει σε θεωρητικό επίπεδο την εφαρμογή, είναι η στιγμή να γίνει η παρουσίαση το πως υλοποιούνται τα παραπάνω στην πράξη. Η υλοποίηση της εφαρμογής έγινε στη γλώσσα προγραμματισμού Java. Οι λόγοι που χρησιμοποιήθηκε η συγκεκριμένη γλώσσα είναι η διαλειτουργικότητα, όπου έχει την ιδιότητα εφαρμογές να μπορούν να εκτελούνται σε μηχανές διαφορετικής αρχιτεκτονικής αλλά και με διαφορετικό λειτουργικό σύστημα, κατά τον ίδιο τρόπο. Επιπλέον, η Java είναι μία αντικειμενοστραφής γλώσσα προγραμματισμού, οπότε ταιριάζει με την προσέγγιση για την ανάπτυξη της εφαρμογής. Ο πηγαίος κώδικας της εφαρμογής πρόβλεψης Prediction, περιέχεται στο παράρτημα A.1, αποθηκευμένος στο σχετικό cd που συνοδεύει το κείμενο της μεταπτυχιακής διατριβής. Στις γραμμές που ακολουθούν, θα αναλυθεί ο κώδικας της εφαρμογής.

4.5.1 Interface Helper

Το interface `Helper` αποτελείται από κάποιες μεθόδους που βοηθούν στο να είναι καλύτερα δομημένος ο κώδικας και καλούνται από άλλη κλάση για την εξαγωγή του αποτελεσματος. Οι μέθοδοι είναι:

- `public static void add_stats(ArrayList<Float> stats, ResultSet rs, int numberOfColumns)`: Η συγκεκριμένη βοηθητική μέθοδος χρησιμοποιείται για την εισαγωγή των στατιστικών στην παράμετρο `Collection stats`. Συγκεκριμένα, διαβάζει την παράμετρο `ResultSet rs` για όλες τις στήλες του και αποθηκεύει ένα - ένα τα στατιστικά στο `Collection stats`.

```
public static void add_stats(ArrayList<Float> stats, ResultSet rs, int numberOfColumns){
    try {
        while (rs.next()) {
            for (int i = 1; i <= numberOfColumns; i++){
                stats.add(rs.getFloat(i));
            }
        }rs.close();
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}
```

Σχήμα 4.26: Η μέθοδος `add_stats`

- `public static void display_names(ArrayList<String> names, Statement stmt, String query)`: Η συγκεκριμένη βοηθητική μέθοδος χρησιμοποιείται για την εισαγωγή των ονομάτων των ομάδων στην παράμετρο `Collection names`. Συγκεκριμένα, εκτελείται από την παράμετρο `stmt` το `query` από το οποίο διαβάζονται και εξάγονται τα δύο ονόματα των ομάδων και αποθηκεύονται στο `Collection names`.

```

public static void display_names(ArrayList<String> names, Statement stmt, String query){
    try {
        ResultSet rs = stmt.executeQuery(query);
        while (rs.next()) {
            names.add(rs.getString(1).trim());
        }
    } catch (SQLException e) {
        System.out.println(e.getMessage());
    }
}

```

Σχήμα 4.27: Η μέθοδος display_names

- `public static Float FirstTeamResult(ArrayList<Float> stats)`: Η συγκεκριμένη βοηθητική μέθοδος επιστρέφει τους συνολικούς πόντους της πρώτης ομάδας. Οι πόντοι προέρχονται από τους το άθροισμα των κερδισμένων πόντων των οχτώ (8) επιπέδων σύγκρισης. Το αποτέλεσμα αποθηκεύεται σε μία μεταβλητή float η οποία είναι το άθροισμα των πόντων από το πρώτο στοιχείο του Collection που επιστρέφουν οι μέθοδοι των οχτώ επιπέδων σύγκρισης.

```

public static Float FirstTeamResult(ArrayList<Float> stats){
    float total1 = FirstLevel.FirstLevelComparison(stats).get(0)
        + SecondLevel.SecondLevelComparison(stats).get(0)
        + ThirdLevel.ThirdLevelComparison(stats).get(0)
        + FourthLevel.FourthLevelComparison(stats).get(0)
        + FifthLevel.FifthLevelComparison(stats).get(0)
        + SixthLevel.SixthLevelComparison(stats).get(0)
        + SeventhLevel.SeventhLevelComparison(stats).get(0)
        + EighthLevel.EighthLevelComparison(stats).get(0);

    return total1;
}

```

Σχήμα 4.28: Η μέθοδος FirstTeamResult

- `public static Float SecondTeamResult(ArrayList<Float> stats)`: Η συγκεκριμένη βοηθητική μέθοδος επιστρέφει τους συνολικούς πόντους της δεύτερης ομάδας. Οι πόντοι προέρχονται

από τους το άθροισμα των κερδισμένων πόντων των οχτώ (8) επιπέδων σύγκρισης. Το αποτέλεσμα αποθηκεύεται σε μία μεταβλητή float η οποία είναι το άθροισμα των πόντων από το δεύτερο στοιχείο του Collection που επιστρέφουν οι μέθοδοι των οχτώ επιπέδων σύγκρισης.

```
public static Float SecondTeamResult(ArrayList<Float> stats){  
    float total2 = FirstLevel.FirstLevelComparison(stats).get(1)  
        + SecondLevel.SecondLevelComparison(stats).get(1)  
        + ThirdLevel.ThirdLevelComparison(stats).get(1)  
        + FourthLevel.FourthLevelComparison(stats).get(1)  
        + FifthLevel.FifthLevelComparison(stats).get(1)  
        + SixthLevel.SixthLevelComparison(stats).get(1)  
        + SeventhLevel.SeventhLevelComparison(stats).get(1)  
        + EighthLevel.EighthLevelComparison(stats).get(1);  
    return total2;  
}
```

Σχήμα 4.29: Η μέθοδος SecondTeamResult

- `public static void Result(ArrayList<Float> stats, ArrayList<String> names, int id1, int id2):` Η συγκεκριμένη μέθοδος αποθηκεύει σε δύο προσωρινές μεταβλητές τα αποτελέσματα των δύο ομάδων. Στη συνέχεια τα μετατρέπει σε ποσοστά επί τις εκατό διαιρώντας το σκορ της μίας ομάδας με το άθροισμα των πόντων και των δύο ομάδων και πολλαπλασιάζοντας το με το 100. Στη συνέχεια, συγκρίνει τα δύο σκορ των ομάδων και βγάζει το τελικό μήνυμα και αποτέλεσμα.

```

public static void Result(ArrayList<Float> stats, ArrayList<String> names, int id1, int id2){

    float temp1 = Helper.FirstTeamResult(stats);
    float temp2 = Helper.SecondTeamResult(stats);

    float total1 = Math.round(((temp1/(float)(temp1+temp2))*100));
    float total2 = Math.round(((temp2/(float)(temp1+temp2))*100));

    if (total1 > total2){

        System.out.println("The team " + id1 + " (" + names.get(0) + ") "
            + " wins team " + id2 + " (" + names.get(1) + ") " + (int)total1 + "% - "
            + (int)total2 + "%" + "\n-----\t-----\t-----");

    }else if (total1 < total2){

        System.out.println("The team " + id2 + " (" + names.get(1) + ") "
            + " wins team " + id1 + " (" + names.get(0) + ") " + (int)total2 + "% - "
            + (int)total1 + "%" + "\n-----\t-----\t-----");

    }
    else{

        System.out.println("THIS IS A DRAW: The team " + id1 + " (" + names.get(0) + ") "
            + " has exact same chances with team " + id2 + " (" + names.get(1) + ") " + (int)total1
            + (int)total2 + "%" + "\n-----\t-----\t-----");

    }

}
}

```

Σχήμα 4.30: Η μέθοδος Result

4.5.2 Κλάση Comparison

Η κλάση Comparison χρησιμοποιείται για να εκτελεί τις συγκρίσεις μεταξύ των δύο ομάδων. Δηλώνονται δύο μεταβλητές μέλη, οι οποίες αντιστοιχούν στα δύο id των ομάδων που είναι αντίπαλες. Αποτελείται από την:

- `public void BFSComparison(String jdbcUrl, String query1, String query2)`: Η συγκεκριμένη μέθοδος αρχικά, δημιουργεί ένα Connection με το jdbcUrl για τη σύνδεση με τη βάση δεδομένων. Στη συνέχεια δημιουργεί ένα Statement μέσω του Connection που δημιουργήθηκε προηγουμένως. Κατόπιν, δημιουργείται μία μεταβλητή results η οποία ισούται με την εκτέλεση το Statement για να διαβάσει το query1. Στην πορεία διαβάζοντας τη μεταβλητή results, δημιουργείται ένα ResultSet στο οποίο αποθηκεύονται τα αποτελέσματα της results και υπολογίζονται οι στήλες του πίνακα από ένα ResultMetaData. Στη συνέχεια καλείται η μέθοδος add_stats του interface Helper για να εισάγει τα στατιστικά στο Collection stats. Στην πορεία καλείται η μέθοδος display_names του interface Helper για

την εμφάνιση των ονομάτων των δύο ομάδων και τέλος καλείται η μέθοδος Result του interface Helper για την εμφάνιση του αποτελέσματος μεταξύ των δύο ομάδων.

```
public void BFSComparison(String jdbcUrl, String query1, String query2){  
    try {  
        Connection con = DriverManager.getConnection(jdbcUrl);  
        Statement stmt = con.createStatement();  
        boolean results = stmt.execute(query1);  
  
        // Collection για την αποθήκευση των στατιστικών  
        ArrayList<Float> stats = new ArrayList<Float>();  
        // Collection για την αποθήκευση των ομαδικών ονομάτων  
        ArrayList<String> names = new ArrayList<String>();  
  
        int numberOfColumns = 0;  
  
        do {  
            if(results) {  
                ResultSet rs = stmt.getResultSet();  
                ResultSetMetaData rsMetaData = rs.getMetaData();  
                numberOfColumns = rsMetaData.getColumnCount(); //Υπολογισμός των στηλών του  
  
                // Κλήση μεθόδου για την εισαγωγή των στατιστικών στο ArrayList  
                Helper.add_stats(stats, rs, numberOfColumns);  
            }  
  
            results = stmt.getMoreResults();  
        }  
    }  
}
```

Σχήμα 4.31: Η μέθοδος BFSComparison της κλάσης Comparison

4.5.3 Interfaces Συγκρίσεων

Τα οχτώ (8) interfaces συγκρίσεων χρησιμοποιούνται για την σύγκριση των δεδομένων και την εξαγωγή αποτελεσματος για κάθε επίπεδο.

Interface FirstLevel

Είναι το πρώτο επίπεδο σύγκρισης και αποτελείται από τη μέθοδο:

- `public static ArrayList<Float> FirstLevelComparison(ArrayList<Float> stats):` Επιστρέφει ένα Collection με τα σκορ των δύο ομάδων από τις συγκρίσεις του πρώτου επιπέδου. Ορίζονται δύο μετρητές (counter1, counter2) αντίστοιχα για τις δύο συγκρινόμενες ομάδες και

αντίστοιχες μεταβλητές που κρατάνε τα συνολικούς πόντους της κάθε ομάδας για τις τέσσερις κατηγορίες στατιστικών (totalof1, totalof2, totaldef1, totaldef2, totalov1, totalov2, totalhst1, totalhst2). Στη συνέχεια διαβάζει τα αριθμητικά δεδομένα από το Collection stats και γίνεται η ανεξάρτητη σύγκριση των στατιστικών, για την ομάδα που κερδίζει προστίθεται μία μονάδα στον αντίστοιχο μετρητή της ενώ η ομάδα που χάνει δεν κερδίζει κάτι, αν υπάρχει ισοπαλία τότε παίρνουν από μισή μονάδα. Μόλις τελειώσουν οι συγκρίσεις των στατιστικών της κατηγορίας, υπολογίζεται το συνολικό αποτέλεσμα για κάθε ομάδα στη συγκεκριμένη κατηγορία, με τον πολλαπλασιασμό του σκορ με την βαρύτητα της κατηγορίας και την βαρύτητα των στατιστικών που ανήκουν στο επίπεδο αυτό. Στη συνέχεια μηδενίζονται οι μετρητές και προχωράει η σύγκριση στις επόμενες κατηγορίες. Μόλις τελειώσουν οι συγκρίσεις, εξάγεται ένα τελικό αποτέλεσμα για κάθε ομάδα, με τα αθροίσματα των σκορ και από τις τέσσερις κατηγορίες. Τέλος, αποθηκεύονται τα τελικά αποτελέσματα σε ένα Collection results.

Interfaces SecondLevel, ThirdLevel, FourthLevel

Τα επόμενα τρία (3) interfaces SecondLevel, ThirdLevel, FourthLevel περιέχουν τις αντίστοιχες τρεις (3) μεθόδους:

- `public static ArrayList<Float> SecondLevelComparison(ArrayList<Float> stats)`
- `public static ArrayList<Float> ThirdLevelComparison(ArrayList<Float> stats)`
- `public static ArrayList<Float> FourthLevelComparison(ArrayList<Float> stats)`

οι οποίες εκτελούν ακριβώς την ίδια διαδικασία με την παραπάνω μέθοδο που παρουσιάστηκε στο πρώτο επίπεδο σύγκρισης. Η μόνη διαφορά είναι ότι αλλάζουν τα ποσοστά βαρύτητας για κάθε κατηγορία και για τα στατιστικά που ανήκουν σε αυτές, τα οποία χρησιμοποιούνται για να υπολογιστεί το συνολικό αποτέλεσμα της κάθε ομάδας σε κάθε κατηγορία.

```

public static ArrayList<Float> FirstLevelComparison(ArrayList<Float> stats){

    float counter1 = 0, counter2 = 0;
    ArrayList<Float> results = new ArrayList<Float>();
    float totalof1, totalof2, totaldef1, totaldef2, totalov1, totalov2, totalhst1, totalhst2;

    // First Level - Defensive Table
    if (stats.get(1) < stats.get(1 + stats.size() / 2)){

        counter1 += 1;

    }else if (stats.get(1).equals(stats.get(1 + stats.size() / 2))){

        counter1 += 0.5;
        counter2 += 0.5;

    }else{

        counter2 += 1;

    }
}

```

Σχήμα 4.32: Η μέθοδος FirstLevelComparison του πρώτου επιπέδου σύγκρισης

Interface FifthLevel

Είναι το πέμπτο επίπεδο σύγκρισης και αποτελείται από τη μέθοδο:

- `public static ArrayList<Float> FifthLevelComparison(ArrayList<Float> stats):` Επιστρέφει ένα Collection με τα σκορ των δύο ομάδων από τις συγκρίσεις του πέμπτου επιπέδου. Ορίζονται δύο μετρητές (counter1, counter2) αντίστοιχα για τις δύο συγκρινόμενες ομάδες και αντίστοιχες μεταβλητές που κρατάνε τα συνολικούς πόντους της κάθε ομάδας για τις τέσσερις κατηγορίες στατιστικών (totalof1, totalof2, totaldef1, totaldef2, totalov1, totalov2, totalhst1, totalhst2). Για τις κατηγορίες των αμυντικών και επιθετικών χαρακτηριστικών γίνεται χρήση κάποιων συσχετισμών - μοτίβων για τα συγκρινόμενα στατιστικά. Συγκεκριμένα, επιλέχθηκαν τετράδες στατιστικών για την ταυτόχρονη σύγκρισή τους. Αν η ομάδα κερδίζει σε δύο στατιστικά από τα τέσσερα τότε απλά παίρνει δύο (2) πόντους, η χαμένη δεν κερδίζει κάποιον πόντο. Αν κερδίσει σε τρία από τα τέσσερα τότε παίρνει σαν μπόνους και μισό πόντο και αν κερδίσει και στα τέσσερα τότε κερδίζει ακόμα ένα επιπλέον πόντο. Συμπερασματικά, μπορεί μία ομάδα που έχει κερδίσει και στα τέσσερα στατιστικά των μοτίβων (patterns), να συγκεντρώσει μέχρι και πέντε και μισό (5,5) πόντους. Για τις υπόλοιπες δύο κατηγορίες (γενικά, ιστορικά), ισχύει η ανεξάρτητη σύγκριση όπου η ομάδα που κερδίζει παίρνει έναν πόντο, στην ισοπαλία οι ομάδες μοιράζονται από μισό πόντο και η

χαμένη δεν κερδίζει κάτι. Μόλις τελειώσουν οι συγκρίσεις των στατιστικών της κατηγορίας, υπολογίζεται το συνολικό αποτέλεσμα για κάθε ομάδα στη συγκεκριμένη κατηγορία, με τον πολλαπλασιασμό του σκορ με την βαρύτητα της κατηγορίας και την βαρύτητα των στατιστικών που ανήκουν στο επίπεδο αυτό. Στη συνέχεια μηδενίζονται οι μετρητές και προχωράει η σύγκριση στις επόμενες κατηγορίες. Μόλις τελειώσουν οι συγκρίσεις, εξάγεται ένα τελικό αποτέλεσμα για κάθε ομάδα, με τα αθροίσματα των σκορ και από τις τέσσερις κατηγορίες. Τέλος, αποθηκεύονται τα τελικά αποτελέσματα σε ένα Collection results.

```
public static ArrayList<Float> FifthLevelComparison(ArrayList<Float> stats){  
  
    float counter1 = 0, counter2 = 0;  
    ArrayList<Float> results = new ArrayList<Float>();  
    float totalof1, totalof2, totaldef1, totaldef2, totalhst1, totalhst2;  
  
    //Level 5 - Defensive Table  
  
    /*Συνδυασμός στατιστικών:  
    Left_Side_Attempts - Middle_Side_Attempts - Outside_Of_Box_Attempts - Aerials_Won*/  
    if (stats.get(15) < stats.get(15 + stats.size() / 2)){  
  
        counter1 += 1;  
  
        if (stats.get(16) > stats.get(16 + stats.size() / 2)){  
  
            counter1 += 1;  
  
            if (stats.get(17) < stats.get(17 + stats.size() / 2)){  
  
                counter1 += 1.5;  
  
                if (stats.get(18) > stats.get(18 + stats.size() / 2)){  
  
                    counter1 += 2;  
  
                }else if (stats.get(18).equals(stats.get(18 + stats.size() / 2))){
```

Σχήμα 4.33: Η μέθοδος FifthLevelComparison του πέμπτου επιπέδου σύγκρισης

Interfaces SixthLevel, SeventhLevel

Τα επόμενα δύο (2) interfaces SixthLevel, SeventhLevel περιέχουν τις αντίστοιχες δύο (2) μεθόδους:

- public static ArrayList<Float> SixthLevelComparison(ArrayList<Float> stats)

- `public static ArrayList<Float> SeventhLevelComparison(ArrayList<Float> stats)`

οι οποίες εκτελούν ακριβώς την ίδια διαδικασία με την παραπάνω μέθοδο που παρουσιάστηκε στο πέμπτο επίπεδο σύγκρισης. Η μόνη διαφορά είναι ότι αλλάζουν τα ποσοστά βαρύτητας για κάθε κατηγορία και για τα στατιστικά που ανήκουν σε αυτές, τα οποία χρησιμοποιούνται για να υπολογιστεί το συνολικό αποτέλεσμα της κάθε ομάδας σε κάθε κατηγορία.

Interface EighthLevel

Είναι το όγδοο και τελευταίο επίπεδο σύγκρισης και αποτελείται από τη μέθοδο:

- `public static ArrayList<Float> EighthLevelComparison(ArrayList<Float> stats)`: Επιστρέφει ένα Collection με τα σκορ των δύο ομάδων από τις συγκρίσεις του όγδοου επιπέδου. Είναι το τελευταίο επίπεδο σύγκρισης των αριθμητικών δεδομένων. Εδώ συγκρίνονται ανεξάρτητα τα αμυντικά και επιθετικά στατιστικά που είχαν χαμηλή συνδυαστική συχνότητα εμφάνισης (κάτω από 60%). Ορίζονται δύο μετρητές (`counter1`, `counter2`) αντίστοιχα για τις δύο συγκρινόμενες ομάδες και αντίστοιχες μεταβλητές που κρατάνε τα συνολικούς πόντους της κάθε ομάδας για τις δύο κατηγορίες στατιστικών (`totalof1`, `totalof2`, `totaldef1`, `totaldef2`). Στη συνέχεια διαβάζει τα αριθμητικά δεδομένα από το Collection `stats` και γίνεται η ανεξάρτητη σύγκριση των στατιστικών, για την ομάδα που κερδίζει προστίθεται μία μονάδα στον αντίστοιχο μετρητή της ενώ η ομάδα που χάνει δεν κερδίζει κάτι, αν υπάρχει ισοπαλία τότε παίρνουν από μισή μονάδα. Μόλις τελειώσουν οι συγκρίσεις των στατιστικών της κατηγορίας, υπολογίζεται το συνολικό αποτέλεσμα για κάθε ομάδα στη συγκεκριμένη κατηγορία, με τον πολλαπλασιασμό του σκορ με την βαρύτητα της κατηγορίας και την βαρύτητα των στατιστικών που ανήκουν στο επίπεδο αυτό. Στη συνέχεια μηδενίζονται οι μετρητές και προχωράει η σύγκριση στην επόμενη κατηγορία. Μόλις τελειώσουν οι συγκρίσεις, εξάγεται ένα τελικό αποτέλεσμα για κάθε ομάδα, με τα αθροίσματα των σκορ και από τις τέσσερις κατηγορίες. Τέλος, αποθηκεύονται τα τελικά αποτελέσματα σε ένα Collection `results`.

4.5.4 Κλάση Main

Η κλάση Main είναι κύρια της εφαρμογής, όπου εκτελείται το πρόγραμμα και εξάγονται τα τελικά αποτελέσματα. Για λόγους ευκολίας και καλύτερης δόμησης κώδικα και μεγάλης συχνότητας εμφάνισης, δηλώνονται στην αρχή κάποιες σταθερές αλφαριθμητικές μεταβλητές. Στη συνέχεια, πραγματοποιείται η σύνδεση με το jdbc αρχείο που είναι απαραίτητο για τη σύνδεση με τη βάση δεδομένων. Κατόπιν, ορίζεται το αλφαριθμητικό jdbcUrl που είναι απαραίτητο για τη σύνδεση με τη βάση. Στην πορεία, εκτελούνται οι φάσεις της διοργάνωσης σειριακά (16 - 8 - 4 - 2). Δημιουργούνται αντικείμενα - στιγμιότυπα της κλάσης Comparison και κάθε ένα από αυτά υποδηλώνει έναν αγώνα. Στη συνέχεια, το κάθε αντικείμενο εκτελεί την μέθοδο BFSComparison που δέχεται σαν ορίσματα το αλφαριθμητικό σύνδεσης με τη βάση δεδομένων, ένα string query με το οποίο επιλέγονται όλα τα στατιστικά από τους πίνακες της βάσης, για τις δύο ομάδες που τίθενται αντιμέτωπες βάσει των αυξόντων αριθμών τους (ids), και τοποθετούνται σε ένα Collection για ευκολότερη επεξεργασία τους και ακόμη ένα string query με το οποίο επιλέγονται τα ονόματα των δύο αντίπαλων ομάδων, βάσει των αυξόντων αριθμών τους (ids). Η ίδια διαδικασία ακολουθείται για όλες τις φάσεις της διοργάνωσης.

```
public class Main {

    private static final String DB_DRIVER = "com.microsoft.sqlserver.jdbc.SQLServerDriver";
    private static final String DB_SERVER= "jdbc:sqlserver://CHALIAPAS\\SQLEXPRESS:1433";
    private static final String DB_USER = "antonis";
    private static final String DB_PASSWORD = "1981";
    private static final String DB_NAME = "FootballStats";
    private static final String DB_QUERY1 = "SELECT df.*, o.*, ov.*, hs.* "
        + "FROM Defensive df INNER JOIN Offensive o ON df.team_id = o.team_id "
        + "JOIN Overview ov ON o.team_id = ov.team_id JOIN History hs ON ov.team_id = hs.tea
        + "WHERE hs.team_id = ";
    private static final String DB_QUERY2 = "SELECT Team_Name FROM Teams WHERE team_id =";

    public static void main(String[] args) {

        try {
            // Σύνδεση με jdbc αρχείο - απαραίτητο για τη σύνδεση με τη βάση
            Class.forName(DB_DRIVER);

            //Καθορισμός αλφαριθμητικού για τη σύνδεση με την βάση
            String jdbcUrl = DB_SERVER + ";user=" + DB_USER + ";password=" + DB_PASSWORD
                + ";databaseName=" + DB_NAME;

            //PHASE 16
            Comparison match1 = new Comparison(5,7);
            Comparison match2 = new Comparison(4,9);
            Comparison match3 = new Comparison(1,16);
        }
    }
}
```

Σχήμα 4.34: Η κλάση εκτέλεσης Main της εφαρμογής

Κεφάλαιο 5

Αξιολόγηση

Μετά την ολοκλήρωση της υλοποίησης της εφαρμογής, στο κεφάλαιο αυτό γίνεται παράθεση των αποτελεσμάτων της εκτέλεσης του εργαλείου πρόβλεψης. Συγκεκριμένα, θα πραγματοποιηθεί η παρουσίαση των αποτελεσμάτων της εφαρμογής σε διάφορες χρονικές περιόδους, για κάθε φάση της διοργάνωσης του Champions League και παράλληλα θα γίνεται σύγκριση με τα πραγματικά αποτελέσματα των αγώνων της διοργάνωσης. Στο στάδιο της συγγραφής της μεταπτυχιακής διατριβής, θα έχουν διεξαχθεί οι φάσεις των δέκα έξι (16) και των προημιτελικών (8), για τις υπόλοιπες δύο, ημιτελικά (4) και τελικός (2) θα πραγματοποιηθεί παρουσίαση της πρόβλεψης των συγκεκριμένων αγώνων και θα γίνει προσπάθεια εξαγωγής έγκυρης πρόβλεψης του νικητή της διοργάνωσης.

5.1 Β' Φάση - Φάση των 16

Η δεύτερη φάση της διοργάνωσης του Champions League διεξήχθη το διάστημα 17 - 18 Φεβρουαρίου με 17 - 18 Μαρτίου. Στη συγκεκριμένη φάση συμμετείχαν τέσσερις (4) ομάδες από τη Γερμανία (FC Bayern Munchen, FC Schalke 04, Borussia Dortmund, Bayer 04 Leverkusen), τρεις ομάδες (3) από την Αγγλία (Chelsea FC, Manchester City FC, Arsenal FC), τρεις (3) από την Ισπανία (Real Madrid CF, FC Barcelona, Club Atletico de Madrid), δύο (2) από την Γαλλία (Paris Saint Germain, AS Monaco FC), μία (1) από την Ιταλία (Juventus), μία (1) από την Ουκρανία (FC Sakhtar Donetsk), μία (1) από την Ελβετία (FC Basel 1893) και τέλος μία (1) από την Πορτογαλία (FC Porto). Τα ζευγάρια των ομάδων ήταν τα ακόλουθα:

Φάση των 16 (Β' Φάση)
Paris Saint Germain - Chelsea FC
FC Sakhtar Donetsk - FC Bayern Munchen
FC Schalke 04 - Real Madrid CF
FC Basel 1893 - FC Porto
Manchester City FC - FC Barcelona
Juventus - Borussia Dortmund
Arsenal FC - AS Monaco FC
Bayer 04 Leverkusen - Club Atletico de Madrid

Πίνακας 5.1: Τα ζευγάρια της β' φάσης

Μετά τη διεξαγωγή των αγώνων, τα αποτελέσματα και οι ομάδες που προκρίθηκαν ήταν:

Ζευγάρια	Αποτελέσματα	Πρόκριση
Paris Saint Germain - Chelsea FC	1-1, 2-2, (3-3)	Paris Saint Germain
FC Sakhtar Donetsk - FC Bayern Munchen	0-0, 0-7, (0-7)	FC Bayern
FC Schalke 04 - Real Madrid CF	0-2, 4-3, (4-5)	Real Madrid CF
FC Basel 1893 - FC Porto	1-1, 0-4, (1-5)	FC Porto

Manchester City FC - FC Barcelona	1-2, 0-1, (1-3)	FC Barcelona
Juventus - Borussia Dortmund	2-1, 3-0, (5-1)	Juventus
Arsenal FC - AS Monaco FC	1-3, 2-0, (3-3)	AS Monaco FC
Bayer 04 Leverkusen - Club Atletico de Madrid	1-0, 0-1 (2-3 Πέναλτι), (1-1)	Club Atletico de Madrid

Πίνακας 5.2: Ο πίνακας με την αποτελέσματα και τις ομάδες που προκρίθηκαν από τη β' φάση

Για τα ζευγάρια όπου οι ομάδες είχαν τα ίδια γκολ και στην άμυνα και στην επίθεση, προκρίνεται η ομάδα που έχει βάλει τα περισσότερα γκολ στο εκτός έδρα παιχνίδι (πχ στο ζευγάρι Paris Saint Germain - Chelsea FC, προκρίνεται η Paris Saint Germain διότι έχει σκοράρει δύο (2) γκολ εκτός έδρας). Για τα ζευγάρια που είχαν ακριβώς τα ίδια αποτελέσματα, η πρόκριση κρίνεται στην παράταση και αν δεν αλλάξει κάτι τότε στην διαδικασία των πέναλτι (πχ το ζευγάρι Bayer 04 Leverkusen - Club Atletico de Madrid, κρίθηκε στα πέναλτι, όπου πέρασε στην επόμενη φάση η Club Atletico de Madrid).

5.1.1 Σενάρια Β' Φάσης

Στην παρών στάδιο παρουσιάζονται τα διάφορα σενάρια αποτελεσμάτων της εφαρμογής για την β' φάση της διοργάνωσης.

Σενάριο 1

Για το σενάριο 1 που εκτελέστηκε στην αρχή της υλοποίησης, δεν χρησιμοποιήθηκαν κάποια μοτίβα συσχετισμών (patterns), δεν έγινε χρήση βαρύτητας προτεραιότητας για τους πίνακες της βάσης δεδομένων ενώ στην ισοπαλία των δεδομένων σύγκρισης οι ομάδες δεν κέρδιζαν κάποιους πόντους. Το μοναδικό κριτήριο που ίσχυε όταν μία ομάδα κέρδιζε, ήταν να παίρνει έναν πόντο. Με βάση αυτά το κριτήριο τα αποτελέσματα ήταν:

Ζευγάρια	Αποτελέσματα
Paris Saint Germain - Chelsea FC	36% - 64%
FC Sakhtar Donetsk - FC Bayern Munchen	26% - 74%
FC Schalke 04 - Real Madrid CF	22% - 78%

FC Basel 1893 - FC Porto	24% - 76%
Manchester City FC - FC Barcelona	33% - 67%
Juventus - Borussia Dortmund	68% - 32%
Arsenal FC - AS Monaco FC	63% - 37%
Bayer 04 Leverkusen - Club Atletico de Madrid	43% - 57%

Πίνακας 5.3: Ο πίνακας με την αποτελέσματα του σεναρίου 1 της β' φάσης

Τα αποτελέσματα του σεναρίου 1 προβλέπουν τις έξι (6) από τις οχτώ (8) ομάδες που προκρίνονται στην προημιτελική φάση της διοργάνωσης. Ειδικότερα, για τα τέσσερα (4) ζευγάρια (FC Sakhtar Donetsk - FC Bayern Munchen, FC Basel 1893 - FC Porto, Manchester City FC - FC Barcelona, Juventus - Borussia Dortmund) οι διαφορές των ποσοστών πρόκρισης αντικατοπτρίζουν και τα πραγματικά αποτελέσματα των αγώνων. Για τα άλλα δύο (2) ζευγάρια (FC Schalke 04 - Real Madrid CF, Bayer 04 Leverkusen - Club Atletico de Madrid), οι διαφορές των ποσοστών πρόκρισης είναι μεγαλύτερες και δεν αντικατοπτρίζουν τόσο πολύ τα πραγματικά αποτελέσματα. Τέλος, για τα ζευγάρια (Paris Saint Germain - Chelsea FC, Arsenal FC - AS Monaco FC) η εφαρμογή δεν προβλέπει σωστά την ομάδα που προκρίνεται.

Σενάριο 2

Για το σενάριο 2 που εκτελέστηκε στην συνέχεια της υλοποίησης, χρησιμοποιήθηκαν βαρύτερες προτεραιότητας για τους πίνακες της βάσης δεδομένων, η ομάδα που κέρδιζε σε ένα στατιστικό έπαιρνε έναν πόντο και στην ισοπαλία των δεδομένων σύγκρισης οι ομάδες κέρδιζαν από μισό πόντο. Με βάση αυτά τα κριτήρια τα αποτελέσματα ήταν:

Ζευγάρια	Αποτελέσματα
Paris Saint Germain - Chelsea FC	30% - 70%
FC Sakhtar Donetsk - FC Bayern Munchen	25% - 75%
FC Schalke 04 - Real Madrid CF	20% - 80%
FC Basel 1893 - FC Porto	20% - 80%
Manchester City FC - FC Barcelona	26% - 74%
Juventus - Borussia Dortmund	73% - 27%
Arsenal FC - AS Monaco FC	62% - 38%

Bayer 04 Leverkusen - Club Atletico de Madrid	35% - 65%
---	-----------

Πίνακας 5.4: Ο πίνακας με την αποτελέσματα του σεναρίου 2 της β' φάσης

Τα αποτελέσματα του σεναρίου 2 δείχνουν πάλι τις έξι (6) από τις οχτώ (8) ομάδες που προκρίνονται στην προημιτελική φάση της διοργάνωσης. Ειδικότερα, για τα τέσσερα ζευγάρια (FC Sakhtar Donetsk - FC Bayern Munchen, FC Basel 1893 - FC Porto, Manchester City FC - FC Barcelona, Juventus - Borussia Dortmund) οι διαφορές των ποσοστών πρόκρισης αντικατοπτρίζουν και τις πραγματικές διαφορές των αποτελεσμάτων των αγώνων. Για τα άλλα δύο (2) ζευγάρια (FC Schalke 04 - Real Madrid CF, Bayer 04 Leverkusen - Club Atletico de Madrid), οι διαφορές των ποσοστών πρόκρισης αυξήθηκαν σε σχέση με το πρώτο σενάριο και δεν αντικατοπτρίζουν τόσο πολύ τα πραγματικά αποτελέσματα. Τέλος, για τα ζευγάρια (Paris Saint Germain - Chelsea FC, Arsenal FC - AS Monaco FC) η εφαρμογή δεν προβλέπει σωστά την ομάδα που προκρίνεται.

Σενάριο 3

Για το σενάριο 3 που εκτελέστηκε στην συνέχεια της υλοποίησης, χρησιμοποιήθηκαν συσχετισμοί στατιστικών για τα δεδομένα με συχνότητα επιτυχίας κάτω από 60% (από το πέμπτο επίπεδο σύγκρισης και κάτω) ενώ ενημερώθηκε η βάση δεδομένων με στατιστικά των ομάδων μέχρι τις 8/2/2015. Με βάση αυτά τα κριτήρια τα αποτελέσματα ήταν:

Ζευγάρια	Αποτελέσματα
Paris Saint Germain - Chelsea FC	42% - 58%
FC Sakhtar Donetsk - FC Bayern Munchen	28% - 72%
FC Schalke 04 - Real Madrid CF	23% - 77%
FC Basel 1893 - FC Porto	23% - 77%
Manchester City FC - FC Barcelona	28% - 72%
Juventus - Borussia Dortmund	71% - 29%
Arsenal FC - AS Monaco FC	54% - 46%
Bayer 04 Leverkusen - Club Atletico de Madrid	46% - 54%

Πίνακας 5.5: Ο πίνακας με την αποτελέσματα του σεναρίου 3 της β' φάσης

Τα αποτελέσματα του σεναρίου 3 δείχνουν πάλι τις έξι (6) από τις οχτώ (8) ομάδες που προκρίνονται στην προημιτελική φάση της διοργάνωσης. Για τα τέσσερα ζευγάρια (FC Sakhtar Donetsk - FC Bayern Munchen, FC Basel 1893 - FC Porto, Manchester City FC - FC Barcelona, Juventus - Borussia Dortmund) οι διαφορές των ποσοστών πρόκρισης αντικατοπτρίζουν και τις πραγματικές διαφορές των αποτελεσμάτων των αγώνων. Για τα άλλα δύο (2) ζευγάρια (FC Schalke 04 - Real Madrid CF, Bayer 04 Leverkusen - Club Atletico de Madrid), οι διαφορές των ποσοστών πρόκρισης μειώθηκαν κατά ένα ποσοστό σε σχέση με το δεύτερο σενάριο και αρχίζουν να πλησιάζουν στα πραγματικά αποτελέσματα. Τέλος, για τα ζευγάρια (Paris Saint Germain - Chelsea FC, Arsenal FC - AS Monaco FC) η εφαρμογή παρόλο που δεν προβλέπει σωστά την ομάδα που προκρίνεται, μειώνεται αισθητά η διαφορά των ποσοστών μεταξύ των ομάδων και προσεγγίζει τα αληθινά αποτελέσματα.

Σενάριο 4

Για το σενάριο 4 που εκτελέστηκε στην συνέχεια της υλοποίησης, ενημερώθηκε η βάση δεδομένων με στατιστικά των ομάδων μέχρι τις 8/3/2015 ενώ προστέθηκαν τα προσωπικά στατιστικά που περιγράφηκαν στο κεφάλαιο 4. Με βάση αυτές τις προσθήκες τα αποτελέσματα ήταν:

Ζευγάρια	Αποτελέσματα
Paris Saint Germain - Chelsea FC	47% - 53%
FC Sakhtar Donetsk - FC Bayern Munchen	27% - 73%
FC Schalke 04 - Real Madrid CF	25% - 75%
FC Basel 1893 - FC Porto	25% - 75%
Manchester City FC - FC Barcelona	29% - 71%
Juventus - Borussia Dortmund	70% - 30%
Arsenal FC - AS Monaco FC	50% - 50%
Bayer 04 Leverkusen - Club Atletico de Madrid	47% - 53%

Πίνακας 5.6: Ο πίνακας με την αποτελέσματα του σεναρίου 4 της β' φάσης

Τα αποτελέσματα του σεναρίου 4 προβλέπουν τις έξι (6) από τις οχτώ (8) ομάδες που προκρίνονται στην προημιτελική φάση της διοργάνωσης. Για τα τέσσερα ζευγάρια (FC Sakhtar Donetsk - FC Bayern Munchen, FC Basel 1893 - FC Porto, Manchester City FC - FC Barcelona, Juventus - Borussia Dortmund) οι διαφορές των ποσοστών πρόκρισης αντικατοπτρίζουν και τις πραγματικές διαφορές των αποτελεσμάτων των αγώνων. Για το ζευγάρι (FC Schalke 04 - Real Madrid CF), η διαφορά των ποσοστών πρόκρισης μειώθηκε κατά ένα ποσοστό σε σχέση με το τρίτο σενάριο και αρχίζει να πλησιάζει το πραγματικό αποτέλεσμα. Για το επόμενο ζευγάρι (Bayer 04 Leverkusen - Club Atletico de Madrid), η μικρή διαφορά δείχνει το οριακό του αποτελεσματος (κρίθηκε στα πέναλτι) και περιγράφει πλήρως την πραγματικότητα. Για το επόμενο ζευγάρι (Arsenal FC - AS Monaco FC) το αποτέλεσμα δείχνει ίσες πιθανότητες και για τις δύο ομάδες (AS Monaco FC προκρίθηκε με το εκτός έδρας γκολ), οπότε μπορεί να θεωρηθεί επιτυχής η πρόβλεψη, αφού το αποτέλεσμα του ζευγαριού δείχνει την ανταγωνιστικότητα μεταξύ των δύο ομάδων. Για το τελευταίο ζευγάρι, (Paris Saint Germain - Chelsea FC,) η εφαρμογή παρόλο που δεν προβλέπει σωστά την ομάδα που προκρίνεται, παρουσιάζεται αισθητή μείωση στη διαφορά μεταξύ των ποσοστών των δύο ομάδων και προσεγγίζεται το πραγματικό αποτέλεσμα. Έτσι, με αυτόν τον τρόπο περιγράφεται η ισοδυναμία του ζευγαριού (η Paris Saint Germain προκρίθηκε με τον κανονισμό του εκτός έδρας γκολ).

Σενάριο 5

Για το τελευταίο σενάριο 5 που εκτελέστηκε, έγιναν κάποιες αλλαγές σχετικές με τις στατιστικές συγκρίσεις και με τους πόντους που θα κερδίζει μία ομάδα. Συγκεκριμένα για τα αμυντικά δεδομένα `In_6_Yards_Box_Attempts_Against`, `Interceptions`, `In_18_Yards_Box_Attempts_Against`, `Middle_Side_Attempts_Against`, `Interceptions` και τα επιθετικά `In_6_Yards_Box_Attempts_For`, `In_18_Yards_Box_Attempts_For`, `Left_Side_Attempts_For`, `Ball_Possession`, `Right_Side_Attack`, `Middle_Side_Attempts_For`, `ThroughBall_Per_Game`, `Crosses_Per_Game`, ομάδα που κέρδιζε παίρνει ακόμα έναν πόντο μπόνους. Επιπλέον, άλλαξε η σειρά κάποιων ανεξάρτητων στατιστικών (από το πρώτο μέχρι το τέταρτο επίπεδο). Με βάση αυτή την αλλαγή τα αποτελέσματα έδειξαν:

Ζευγάρια	Αποτελέσματα
Paris Saint Germain - Chelsea FC	52% - 48%
FC Sakhtar Donetsk - FC Bayern Munchen	28% - 72%
FC Schalke 04 - Real Madrid CF	25% - 75%
FC Basel 1893 - FC Porto	21% - 79%
Manchester City FC - FC Barcelona	24% - 76%
Juventus - Borussia Dortmund	64% - 36%
Arsenal FC - AS Monaco FC	47% - 53%
Bayer 04 Leverkusen - Club Atletico de Madrid	48% - 52%

Πίνακας 5.7: Ο πίνακας με την αποτελέσματα του σεναρίου 5 της β' φάσης

Τα αποτελέσματα του πέμπτου και τελευταίου σεναρίου δείχνουν και τις οχτώ (8) ομάδες που προκρίνονται στην επόμενη φάση. Για τα επτά (7) ζευγάρια (Paris Saint Germain - Chelsea FC, FC Sakhtar Donetsk - FC Bayern Munchen, FC Basel 1893 - FC Porto, Manchester City FC - FC Barcelona, Juventus - Borussia Dortmund, Arsenal FC - AS Monaco FC, Bayer 04 Leverkusen - Club Atletico de Madrid) οι διαφορές των ποσοστών πρόκρισης είναι πολύ κοντά στην πραγματική εξέλιξη και στα τελικά αποτελέσματα των αγώνων. Το ζευγάρι (FC Schalke 04 - Real Madrid CF) δείχνει τον νικητή αλλά με λίγο μεγαλύτερη διαφορά από την πραγματική έκβαση του ζευγαριού. Παρακάτω, παρουσιάζεται ο συγκεντρωτικός πίνακας με τις προβλέψεις όλων των σεναρίων για την β' φάση της διοργάνωσης.

Ζευγάρια	Σενάριο 1	Σενάριο 2	Σενάριο 3	Σενάριο 4	Σενάριο 5	Σχόλια
Paris Saint Germain - Chelsea FC	36% - 64%	30% - 70%	42% - 58%	47% - 53%	52% - 48%	Καλύτερο το σενάριο 5 - δείχνει νίκη της Paris
FC Sakhtar Donetsk - FC Bayern Munchen	26% - 74%	25% - 75%	28% - 72%	27% - 73%	28% - 72%	Σχεδόν ίδιες διαφορές για όλα τα σενάρια - Δεκτό το 5 (μεγάλη διαφορά δυναμικότητας)
FC Schalke 04 - Real Madrid CF	22% - 78%	20% - 80%	23% - 77%	25% - 75%	25% - 75%	Σχεδόν ίδιες διαφορές για όλα τα σενάρια - Δεκτό το 5 (μεγάλη διαφορά δυναμικότητας)

FC Basel 1893 - FC Porto	24% - 76%	20% - 80%	23% - 77%	25% - 75%	21% - 79%	Καλύτερο σενάριο 4 - Εύκολη πρόκριση για Porto
Manchester City FC - FC Barcelona	33% - 67%	26% - 74%	28% - 72%	29% - 71%	24% - 76%	Καλύτερο σενάριο 4 - Εύκολη πρόκριση για Barcelona
Juventus - Borussia Dortmund	68% - 32%	73% - 27%	71% - 29%	70% - 30%	64% - 36%	Καλύτερα σενάρια 1, 5 - Εύκολη πρόκριση για Juventus
Arsenal FC - AS Monaco FC	63% - 37%	62% - 38%	54% - 46%	50% - 50%	47% - 53%	Καλύτερο σενάριο 5 - δείχνει νίκη της Monaco
Bayer 04 Leverkusen - Club Atletico de Madrid	43% - 57%	35% - 65%	46% - 54%	47% - 53%	48% - 52%	Καλύτερα σενάρια 4, 5 - Δείχνουν νίκη Atletico και τον ανταγωνισμό του ζευγαριού

Πίνακας 5.8: Συγκεντρωτικός πίνακας με τα αποτελέσματα όλων των σεναρίων της β' φάσης

Συμπερασματικά, οι συνεχείς αναπροσαρμογές της εφαρμογής με την εισαγωγή τεχνικών σύγκρισης (χρήση μοτίβων, εισαγωγή προσωπικών στατιστικών, εισαγωγή μπόνους πόντους για κάποια χαρακτηριστικά) και οι σταδιακές ανανεώσεις της βάση δεδομένων, συνέβαλαν στην σωστή πρόβλεψη και των οχτώ (8) νικητών (100% επιτυχίας) της β' φάσης. Το μοναδικό ζευγάρι σωστής πρόβλεψης νικητή αλλά με μεγαλύτερη διαφορά ποσοστών από το πραγματικό αποτέλεσμα, είναι το FC Schalke 04 - Real Madrid CF. Η διαφορά μεταξύ των δύο ομάδων σταδιακά μειωνόταν σε κάθε σενάριο, παρόλα αυτά η διαφορά δυναμικότητας μεταξύ τους είναι τεράστια γεγονός που δικαιολογεί το μεγάλο εύρος νίκης υπέρ της Real Madrid CF.

5.2 Προημιτελική Φάση

Η προημιτελική της διοργάνωσης του Champions League διεξήχθη το διάστημα 14 - 15 Απριλίου με 21 - 22 Μαρτίου. Στη συγκεκριμένη φάση συμμετείχαν μία (1) ομάδα από τη Γερμανία (FC Bayern Munchen), τρεις (3) από την Ισπανία (Real Madrid CF, FC Barcelona, Club Atletico de

Madrid), δύο (2) από την Γαλλία (Paris Saint Germain, AS Monaco FC), μία (1) από την Ιταλία (Juventus) και τέλος μία (1) από την Πορτογαλία (FC Porto). Τα ζευγάρια των ομάδων ήταν τα ακόλουθα:

Προημιτελικά
Club Atletico de Madrid - Real Madrid CF
Juventus - AS Monaco FC
Paris Saint Germain - FC Barcelona
FC Porto - FC Bayern Munchen

Πίνακας 5.9: Τα ζευγάρια της προημιτελικής φάσης

Μετά τη διεξαγωγή των αγώνων, τα αποτελέσματα και οι ομάδες που προκρίθηκαν ήταν:

Ζευγάρια	Αποτελέσματα	Πρόκριση
Club Atletico de Madrid - Real Madrid CF	0-0, 0-1, (0-1)	Real Madrid CF
Juventus - AS Monaco FC	1-0, 0-0, (1-0)	Juventus
Paris Saint Germain - FC Barcelona	1-3, 0-2, (1-5)	FC Barcelona
FC Porto - FC Bayern Munchen	3-1, 1-6, (4-7)	FC Bayern Munchen

Πίνακας 5.10: Ο πίνακας με την αποτελέσματα και τις ομάδες που προκρίθηκαν από την προημιτελική φάση

5.2.1 Σενάρια Προημιτελικής Φάσης

Στην παρών στάδιο παρουσιάζονται τα διάφορα σενάρια αποτελεσμάτων της εφαρμογής για την προημιτελική της διοργάνωσης.

Σενάριο 1

Για το σενάριο 1, προστέθηκαν τα προσωπικά στατιστικά που παρουσιάστηκαν στο κεφάλαιο 4 στη βάση δεδομένων και η ενημερώθηκαν τα δεδομένα μέχρι τις 8-3-2015 (αλλαγές σεναρίου 4 β' φάσης). Με βάση τις αλλαγές τα αποτελέσματα ήταν:

Ζευγάρια	Αποτελέσματα
Club Atletico de Madrid - Real Madrid CF	43% - 57%
Juventus - AS Monaco FC	62% - 38%
Paris Saint Germain - FC Barcelona	30% - 70%
FC Porto - FC Bayern Munchen	39% - 61%

Πίνακας 5.11: Ο πίνακας με την αποτελέσματα του σεναρίου 1 της προημιτελικής φάσης

Τα αποτελέσματα του σεναρίου 1 δείχνουν και τις τέσσερις (4) ομάδες που προκρίνονται στην ημιτελική φάση της διοργάνωσης. Για τα τρία ζευγάρια (Club Atletico de Madrid - Real Madrid CF, Paris Saint Germain - FC Barcelona, FC Porto - FC Bayern Munchen) οι διαφορές των ποσοστών πρόκρισης αντικατοπτρίζουν και την πραγματική εξέλιξη των αγώνων. Για το τελευταίο ζευγάρι (Juventus - AS Monaco FC), η εφαρμογή προβλέπει σωστά την ομάδα που προκρίνεται στην επόμενη φάση, αλλά η διαφορά των ποσοστών είναι μεγαλύτερη από την πραγματική διαφορά των ανταγωνιζόμενων ομάδων.

Σενάριο 2

Για το σενάριο 2, έγιναν κάποιες αλλαγές σχετικές με τις στατιστικές συγκρίσεις όπως ακριβώς παρουσιάστηκαν στο σενάριο 5 της β' φάσης της. Με βάση τις αλλαγές τα αποτελέσματα ήταν:

Ζευγάρια	Αποτελέσματα
Club Atletico de Madrid - Real Madrid CF	48% - 52%
Juventus - AS Monaco FC	55% - 45%
Paris Saint Germain - FC Barcelona	33% - 67%
FC Porto - FC Bayern Munchen	42% - 58%

Πίνακας 5.12: Ο πίνακας με την αποτελέσματα του σεναρίου 2 της προημιτελικής φάσης

Τα αποτελέσματα του σεναρίου 2 δείχνουν και τις τέσσερις (4) ομάδες που προκρίνονται στην ημιτελική φάση της διοργάνωσης. Οι διαφορές των ποσοστών για όλα τα ζευγάρια αντικατοπτρίζουν πλήρως τα πραγματικά αποτελέσματα των αγώνων και αποδεικνύεται ότι το συγκεκριμένο σενάριο βελτίωσε τα αποτελέσματα του σεναρίου 1.

Συμπερασματικά, για την προημιτελική φάση εκτελεστήκανε δύο (2) σενάρια από τα οποία έγινε πρόβλεψη και των τεσσάρων νικητών (100% επιτυχία), ενώ οι αναπροσαρμογές στον αριθμό των κερδισμένων πόντων για κάποιες συγκρίσεις στατιστικών, συνέβαλε στην αισθητή βελτίωση των ποσοστιαίων διαφορών στα διάφορα ζευγάρια και στην πληρέστερη απεικόνιση των πραγματικών αποτελεσμάτων.

5.3 Ημιτελική Φάση

Η ημιτελική της διοργάνωσης του Champions League διεξήχθη το διάστημα 5 - 6 Μαΐου με 12 - 13 Μαΐου. Στη συγκεκριμένη φάση συμμετέχουν μία (1) ομάδα από τη Γερμανία (FC Bayern Munchen), δύο (2) από την Ισπανία (Real Madrid CF, FC Barcelona) και μία (1) από την Ιταλία (Juventus). Τα ζευγάρια των ομάδων είναι τα ακόλουθα:

Ημιτελικά
Juventus - Real Madrid CF
FC Barcelona - FC Bayern Munchen

Πίνακας 5.13: Τα ζευγάρια της ημιτελικής φάσης

Μετά τη διεξαγωγή των αγώνων, τα αποτελέσματα και οι ομάδες που προκρίθηκαν ήταν:

Ζευγάρια	Αποτελέσματα	Πρόκριση
Juventus - Real Madrid CF	2-1, 1-1, (3-2)	Juventus
FC Barcelona - FC Bayern Munchen	3-0, 2-3, (5-3)	FC Barcelona

Πίνακας 5.14: Ο πίνακας με την αποτελέσματα και τις ομάδες που προκρίθηκαν από την ημιτελική φάση

5.3.1 Σενάριο Ημιτελικής Φάσης

Σύμφωνα με τις αλλαγές που έγιναν στον αλγόριθμο της εφαρμογής (όπως παρουσιάστηκαν στο σενάριο 5 της β'φάσης και στο σενάριο 2 της προημιτελικής φάσης) τα αποτελέσματα πρόβλεψης ήταν:

Ζευγάρια	Αποτελέσματα
Juventus - Real Madrid CF	51% - 49%
FC Barcelona - FC Bayern Munchen	54% - 46%

Πίνακας 5.15: Ο πίνακας με την αποτελέσματα πρόβλεψης της ημιτελικής φάσης

Τα αποτελέσματα της εφαρμογής πρόβλεψης δείχνει και τις δύο ομάδες που προκρίνονται στον τελικό του Champions League. Για το πρώτο ζευγάρι (Juventus - Real Madrid CF), η διαφορά των ποσοστών είναι οριακή, γεγονός που αποδεικνύει την ανταγωνιστικότητα μεταξύ των δύο ομάδων και του τελικού αποτελέσματος, ενώ για το δεύτερο ζευγάρι (FC Barcelona - FC Bayern Munchen) η διαφορά είναι λίγο μεγαλύτερη, γεγονός που αποδεικνύεται η υπεροχή της FC Barcelona και απεικονίζεται η τελική διαφορά των συνολικών γκολ μεταξύ των δύο ομάδων.

Συμπερασματικά, για την ημιτελική φάση για το μοναδικό σενάριο που εκτελέστηκε πραγματοποιήθηκε πρόβλεψη και των δύο νικητών (100% επιτυχία), ενώ ο αλγόριθμος με την παρούσα μορφή του απεικονίζει με ρεαλιστικό τρόπο τις ποσοστιαίες διαφορές στα δύο (2) ζευγάρια και δείχνει πλήρως τα πραγματικά αποτελέσματα.

5.4 Τελική Φάση

Ο τελικός της διοργάνωσης του Champions League είναι προγραμματισμένος να διεξαχθεί στις 6 Ιουνίου και θα συμμετάσχουν η Juventus από την Ιταλία και η FC Barcelona από την Ισπανία. Η καταληκτική ημερομηνία παράδοσης της μεταπτυχιακής διατριβής ήταν πριν την διεξαγωγή του τελικού της διοργάνωσης (17 Μαΐου). Για το γεγονός αυτό θα γίνει μόνο προσπάθεια πρόβλεψης της ομάδας που θα κατακτήσει την διοργάνωση του Champions League. Σύμφωνα με τις αλλαγές που έγιναν στον αλγόριθμο της εφαρμογής (όπως παρουσιάστηκαν στο σενάριο 5 της β'φάσης, στο σενάριο 2 της προημιτελικής φάσης και στο σενάριο της ημιτελικής φάσης) το αποτέλεσμα πρόβλεψης είναι:

Τελικός Champions League	
Ζευγάρι	Αποτέλεσμα
Juventus - FC Barcelona	43% - 57%

Πίνακας 5.16: Ο πίνακας με το αποτέλεσμα πρόβλεψης του τελικού

Τα αποτελέσματα δείχνει αρκετά μεγάλη ποσοστιαία διαφορά ανάμεσα στις δύο ομάδες και περιγράφεται το γεγονός ότι η Ισπανική ομάδα FC Barcelona έχει τον πρώτο λόγο για την κατάκτηση του τίτλου έναντι της Ιταλικής Juventus.

Κεφάλαιο 6

Συμπεράσματα και Μελλοντικές Προεκτάσεις

6.1 Ανακεφαλαίωση

Πρωταρχικός σκοπός της παρούσας μεταπτυχιακής διατριβής, ήταν η δημιουργία μίας εφαρμογής στατιστικής ανάλυσης αριθμητικών δεδομένων αγωνιστικής απόδοσης για τις δέκα έξι (16) ποδοσφαιρικές ομάδες, που προκρίθηκαν στην β' φάση της ευρωπαϊκής διασυλλογικής διοργάνωσης του Champions League. Με βάση αυτή την εφαρμογή, επιχειρήθηκε η πρόβλεψη των νικητών των διαφόρων ζευγαριών που προέκυψαν, σε διάφορες φάσεις της διοργάνωσης, με απώτερο σκοπό την τελική πρόβλεψη του πρωταθλητή.

Έχουν παρατηρηθεί, στο παρελθόν κάποιες πρόσφατες απόπειρες πρόβλεψης νικητών ποδοσφαιρικών διοργανώσεων όπως πχ η επιτυχής πρόβλεψη των αποτελεσμάτων της knockout φάσης και του νικητή του Παγκοσμίου Πρωταθλήματος Εθνικών Ομάδων 2014 (Mundial) από την μηχανή αναζήτησης Bing της εταιρίας Microsoft, η οποία χρησιμοποίησε μαθηματικό μοντέλο συνδυάζοντας κάποιες τεχνικές μηχανικής μάθησης ή απόπειρα πρόβλεψης πάλι του νικητή του Mundial μέσω της γλώσσας Wolfram με μεθόδους πρόβλεψης και ταξινόμησης (classification). Παρόλα αυτά, σημειώνεται σημαντική έλλειψη εφαρμογών που συνδυάζουν τεχνικές τεχνητής νοημοσύνης για την πρόβλεψη έγκυρων αποτελεσμάτων ποδοσφαιρικών αγώνων, γεγονός που αποτέλεσε βασικό κίνητρο για τη υλοποίηση της παρούσας έρευνας.

Η επιτυχία της εφαρμογής μεταφράζεται, με την σωστή πρόβλεψη όλων των νικητριών ομάδων τόσο στην β' φάση και στην προημιτελική φάση όσο και στα ημιτελικά της διοργάνωσης του Champions League και οφείλεται στον συνδυασμό κάποιων καινοτόμων τεχνικών τεχνητής νοημοσύνης και στατιστικής ανάλυσης. Χρησιμοποιήθηκαν περίπου 130 κατηγορίες ομαδικών στατιστικών, τα οποία χωρίστηκαν σε τέσσερις υποκατηγορίες των αμυντικών, επιθετικών, γενικών και ιστορικών χαρακτηριστικών. Για όλα τα αριθμητικά δεδομένα, καταγράφηκαν οι συχνότητες εμφάνισης τους, και ορίστηκαν βαρύτητες για τον καθορισμό επιπέδων προτεραιότητας στη σειρά σύγκρισής τους. Με βάση, την επιλογή αλγορίθμου αναζήτησης τεχνητής νοημοσύνης (Κατά Πλάτος) και την υλοποίηση της εφαρμογής πρόβλεψης, καθορίστηκε η σειρά στατιστικής ανάλυσης των συλλεγόμενων αριθμητικών δεδομένων για την εξαγωγή πρόβλεψης του νικητή στα διάφορα ζευγάρια που προέκυψαν στη διοργάνωση.

Η ανάλυση των πραγματικών αποτελεσμάτων των αγώνων, συνέβαλλε στις συνεχείς αναπροσαρμογές του αρχικού κώδικα της εφαρμογής με την υλοποίηση κάποιων αλλαγών σχετικών με την προτεραιότητα που θα δοθεί σε κάποια στατιστικά, με αποτέλεσμα την αλλαγή στη σειρά σύγκρισης των στατιστικών δεδομένων και την προσπάθεια συνεχούς αύξησης του ποσοστού επιτυχίας πρόβλεψης και την εξαγωγή ρεαλιστικότερου και εγκυρότερου αποτελέσματος.

6.2 Μελλοντική Χρήση

Η εφαρμογή πρόβλεψης του νικητή της διοργάνωσης του Champions League που παρουσιάστηκε παραπάνω, φιλοδοξεί να αποτελέσει καινοτόμο εργαλείο στο πεδίο των προβλέψεων διαφόρων αθλητικών γεγονότων. Η έλλειψη ανάλογων εφαρμογών, μπορεί να συμβάλλει στην καθιέρωσή του και στην μελλοντική αναβάθμισή του για την αποτελεσματικότερη εξαγωγή έγκυρων αποτελεσμάτων σε διάφορες αθλητικές διοργανώσεις.

Η εφαρμογή κάνει χρήση ενός μεγάλου αριθμού στατιστικών, παρόλα αυτά θα μπορούσε να πραγματοποιηθεί μεγαλύτερη έρευνα σε ιστοσελίδες δεδομένων αγωνιστικής απόδοσης, έτσι ώστε να χρησιμοποιηθούν ακόμα περισσότερες κατηγορίες (αμυντικά, επιθετικά, ιστορικά, γενικά), τόσο ομαδικών όσο και ατομικών χαρακτηριστικών, για την αποτελεσματικότερη επεξεργασία των δεδομένων και την εξαγωγή εγκυρότερων αποτελεσμάτων. Επιπλέον, θα ήταν εφικτό να γίνει ανάλυση της αγωνιστικής απόδοσης των διαφόρων ομάδων τόσο για τα παιχνίδια εντός έδρας, όσο και τα εκτός, για την ανεξάρτητη πρόβλεψη των αποτελεσμάτων σε μεμονωμένους αγώνες και την παράλληλη πρόβλεψη των νικητών των διαφόρων ζευγαριών.

Επίσης, η μακροχρόνια χρήση της εφαρμογής σε παρόμοιες ποδοσφαιρικές διοργανώσεις (Europa League, Διοργανώσεις Εθνικών Ομάδων, πρωταθλητές εγχώριων πρωταθλημάτων κτ) αλλά και σε επόμενες αγωνιστικές χρονιές του Champions League, μπορεί να βοηθήσει στην αναπροσαρμογή της και στη βελτίωσή της, με απώτερο σκοπό την εξαγωγή εγκυρότερων και ρεαλιστικότερων αποτελεσμάτων. Επιπροσθέτως, η χρήση συσχετισμών μεταξύ των στατιστικών και η καταγραφή της συχνότητας εμφάνισης των δεδομένων αποτελεί μία καινοτόμος μέθοδος για την εξαγωγή συμπερασμάτων στο τομέα της στατιστικής ανάλυσης. Παρόλα αυτά, η εφαρμογή θα μπορούσε να συνδυάσει και κάποιες τεχνικές στατιστικής ανάλυσης, όπως είναι οι παλινδρομήσεις (linear - logistic regression) ή ανάλυση συστάδων (cluster analysis), έτσι ώστε να μπορεί να στηρίζει τα συμπεράσματά της, όχι μόνο στην παρατήρηση αλλά και στην επιστημονική στατιστική επεξεργασία αριθμητικών δεδομένων.

Τέλος, η συγκεκριμένη εφαρμογή με κάποιες συμπληρωματικές αλλαγές στο σχεδιασμό της, μπορεί να καλύψει τις ανάγκες και σε άλλα αθλητικά γεγονότα. Μπορεί να αποτελέσει παράδειγμα και οδηγό πάνω στον οποίο θα στηριχθεί η δημιουργία νέων εφαρμογών, που σκοπό θα έχουν να προβλέψουν τους νικητές σε άλλα και διαφορετικότερα αθλητικά γεγονότα.

6.3 Επίλογος

Στην παρούσα μεταπτυχιακή διατριβή, παρουσιάστηκε η ανάγκη συλλογής στατιστικών διαφορών ποδοσφαιρικών συλλόγων για την αρτιότερη επεξεργασία τους και την εγκυρότερη πρόβλεψη του νικητή του Champions League. Συνδύασε, πρωτότυπες τεχνικές συσχετισμών αριθμητικών δεδομένων, μέσα από την παρατήρηση και την καταγραφή συχνοτήτων εμφάνισης, για τον καθορισμό προτεραιοτήτων στη σειρά επεξεργασίας των συλλεγόμενων στατιστικών. Η επιτυχής πρόβλεψη των περισσότερων αγώνων της διοργάνωσης, συμβάλλει στην προβολή της αποτελεσματικότητας της εφαρμογής και μπορεί να βοηθήσει στο να μετατραπεί σε οδηγό για την δημιουργία ανάλογων εγχειρημάτων σε μελλοντικές διοργανώσεις.

Βιβλιογραφία

- [01] Alan Mackworth, David Poole (2010). *Artificial Intelligence: Foundations Of Computational Agents*. Cambridge University Press.
- [02] FourFourTwo (2014). *STATS ZONE 2014 - 2015*. Retrieved From: <http://www.fourfourtwo.com/statszone>
- [03] George F. Luger (2005). *Artificial Intelligence Structures and Strategies for Complex Problems*. 5th ed. Pearson Education Limited.
- [04] Isqt Exam Certification (2015). What is Iterative model - advantages, disadvantages and when to use it?. Retrieved From: <http://istqbexamcertification.com/what-is-iterative-model-advantages-disadvantages-and-when-to-use-it/>
- [05] Max Cherney (2014). *The Technology Behind the World Cup's Advanced Analytics*. Retrieved From: http://motherboard.vice.com/en_ca/read/this-system-turns-the-beautiful-game-into-big-data
- [06] Peter Norvig, Stuart Russel (2010). *Artificial Intelligence: A Modern Approach - Third Edition*. Pearson Education.
- [07] Squawka (2014). *Football Statistics for all Squawka competitions*. Retrieved from: <http://www.squawka.com/football-stats>
- [08] Soccerwidow (2012). *How to Calculate Odds - Over/Under 'X' Goals*. 1st Edition. Soccerwidow Ltd.
- [09] Tom J Doyle (2013). *Discovering The Stats Powering Football Manager 2014*. Retrieved From: <http://www.squawka.com/news/fm14/20564>
- [10] TvbEurope (2012). *Deltatre scores Champions League contract*. Retrieved From: <http://legacy.tvbeurope.com/main-content/full/deltatre-scores-champions-league-contract/dl15#.VUkt7Pntmkp>

- [11] Uefa.com (2014). *Uefa Champions League 2014 - 2015*. Retrieved from: <http://www.uefa.com/uefachampionsleague/season=2015/statistics/index.html>
- [12] WhoScored.com (2014). *Football Statistics 2014 - 2015*. Retrieved from: <http://www.whoscored.com/Statistics>
- [13] Βλαχάβας Ιωάννης, Κεφάλας Πέτρος, Βασιλειάδης Νικόλαος, Κόκκορας Φώτης, Σακελλαρίου Ηλίας (2011). *Τεχνητή Νοημοσύνη - 3η Έκδοση*. Εκδόσεις Πανεπιστημίου Μακεδονίας.
- [14] Ματσατσίνης Νικόλαος Φ., Σπαναδάκης Νικόλαος, Σαμαράς Ανδρέας (2005). *Εισαγωγή στην Τεχνητή Νοημοσύνη και στα Συστήματα Πολλαπλών Πρακτόρων*. Εκδόσεις Νέων Τεχνολογιών.

Παράρτημα Α

Κώδικας και Βάση Δεδομένων

A.1 Πηγαίος Κώδικας Εφαρμογής

Ο πηγαίος κώδικας της εφαρμογής πρόβλεψης ChampionsLeaguePredictor, περιέχεται στο σχετικό cd που συνοδεύει το κείμενο της μεταπτυχιακής διατριβής.

A.2 Βάση Δεδομένων

Το backup της βάσης δεδομένων FootballStats, με τους πίνακες και τα καταχωρημένα στατιστικά περιέχεται στο σχετικό cd που συνοδεύει το κείμενο της μεταπτυχιακής διατριβής.