ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ
ΥΠΟΛΟΓΙΣΤΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ
ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη Αποτελεσμάτων Αθλητικών Γεγονότων με χρήση Δικτύων Μακράς Βραχύχρονης Μνήμης (LSTM)

ΣΑΡΡΗΣ ΑΝΤΩΝΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ:
ΔΟΥΛΑΜΗΣ ΝΙΚΟΛΑΟΣ,
Καθηγητής ΕΜΠ

ΦΕΒΡΟΥΑΡΙΟΣ 2023

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΝΑΥΤΙΛΙΑΣ ΚΑΙ ΒΙΟΜΗΧΑΝΙΑΣ
ΤΜΗΜΑΤΟΣ ΒΙΟΜΗΧΑΝΙΚΗΣ ΔΙΟΙΚΗΣΗΣ & ΤΕΧΝΟΛΟΓΙΑΣ
ΔΙΑΠΑΝΕΠΙΣΤΗΜΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΤΕΧΝΟ-ΟΙΚΟΝΟΜΙΚΑ ΣΥΣΤΗΜΑΤΑ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόβλεψη Αποτελεσμάτων Αθλητικών Γεγονότων με χρήση Δικτύων Μακράς Βραχύχρονης Μνήμης (LSTM)

ΣΑΡΡΗΣ ΑΝΤΩΝΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: ΔΟΥΛΑΜΗΣ ΝΙΚΟΛΑΟΣ, Καθηγητής ΕΜΠ

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 27η Ιανουαρίου 2023

ΔΟΥΛΑΜΗΣ	ΔΟΥΛΑΜΗΣ	ΒΑΡΒΑΡΙΓΟΥ
ΝΙΚΟΛΑΟΣ	ΑΝΑΣΤΑΣΙΟΣ	ΘΕΟΔΩΡΑ
ΚΑΘΗΓΗΤΗΣ	ΑΝΑΠΛΗΡΩΤΗΣ	ΚΑΘΗΓΗΤΡΙΑ
Ε.Μ.Π	ΚΑΘΗΓΗΤΗΣ	Ε.Μ.Π
	Ε.Μ.Π	

ΠΕΡΙΛΗΨΗ

Τα τελευταία χρόνια ο τομέας της αθλητικής ανάλυσης παρουσιάζει μια συναρπαστική ευκαιρία για την εφαρμογή συστημάτων τεχνητής νοημοσύνης προκειμένου να υποβοηθηθεί η λήψη σύνθετων, σε πραγματικό χρόνο αποφάσεων σε ένα δυναμικό περιβάλλον με δεκάδες άτομα σε αλληλεπίδραση. Το National Basketball Association (NBA) είναι ένα επαγγελματικό πρωτάθλημα καλαθοσφαίρισης στη Βόρεια Αμερική και θεωρείται το καλύτερο και πιο απαιτητικό πρωτάθλημα στον κόσμο. Συνεπώς, η συλλογή στατιστικών των play-by-play δεδομένων του εν λόγω πρωταθλήματος αποτυπώνει την λεπτομερής εξέλιξη του αγώνα, αποτελώντας την πιο αναλυτική στατιστική πληροφορία μιας αναμέτρησης. Στο πλαίσιο της παρούσας εργασίας θα πραγματοποιηθεί η συγκέντρωση και κανονικοποίηση αθλητικών δεδομένων ώστε να έρθουν σε μια μορφή που να μπορούν να αναλυθούν με χρήση βελτιωμένων μορφών τεχνητών ανατροφοδοτούμενων νευρωνικών δικτύων που χρησιμοποιούνται στον τομέα της βαθιάς μάθησης, όπως τα Δίκτυα Μακράς Βραχύχρονης Μνήμης (LSTM). Στόχος είναι η μέθοδος που θα ακολουθηθεί να πετύχει ένα υψηλό ποσοστό πρόβλεψης του τελικού αποτελέσματος πριν την λήξη ενός αθλητικού γεγονότος.

Λέξεις κλειδιά: Μηχανική Μάθηση, Βαθιά Μάθηση, Ανάλυση Αθλητικών Γεγονότων, Δίκτυα Μακράς Βραχύχρονης Μνήμης

ABSTRACT

In recent years, the field of sports analytics has presented an exciting opportunity to apply artificial intelligence systems to support complex, real-time decision making in a dynamic environment with dozens of interacting individuals. The National Basketball Association (NBA) is a professional basketball league in North America and is considered the best and most demanding league in the world. Therefore, the statistical collection of the play-by-play data of the said league captures the detailed evolution of the match, constituting the most detailed statistical information of a match. This work will involve gathering and normalizing sports data into a form that can be analyzed using improved forms of artificial feedback neural networks used in the field of deep learning, such as Long Short-Term Memory (LSTM) Networks. The aim is for the method to be followed to achieve a high percentage of prediction of the final result before the end of a sporting event.

Key Words: Machine Learning, Deep Learning, Sport Analytics, Long Short-Term Memory, LSTM

ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία εκπονήθηκε στη σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Εθνικού Μετσόβιου Πολυτεχνείου κατά το ακαδημαϊκό έτος 2022 - 2023, στο πλαίσιο της ενασχόλησής μου με το πρόγραμμα μεταπτυχιακών σπουδών «Τεχνο-οικονομικά Συστήματα».

Θα ήθελα να ευχαριστήσω τον επιβλέποντα Καθηγητή Δουλάμη Νικόλαο για την αμέριστη υποστήριξή του καθ' όλη την διάρκεια της συγγραφής της παρούσας μεταπτυχιακής εργασίας.

Επιπρόσθετα θα ήθελα να ευχαριστήσω ιδιαίτερα την Μαρία για την ψυχολογική υποστήριξη και την υπομονή της, στο πρόσωπό μου, σε όλη την διάρκεια των μεταπτυχιακών σπουδών μου.

«… Πάντα προσπάθεια. Πάντα αποτυχία. Δεν πειράζει.
Προσπάθησε ξανά !! Απότυχε ξανά !! Απότυχε καλύτερα !! …»

Σάμουελ Μπέκετ

«… Ever tried. Ever failed. No matter.
Try again !! Fail again !! Fail better !! …»

Samuel Beckett

Περιεχόμενα

Λίστα Εικόνων

Λίστα Γραφημάτων

Λίστα Πινάκων

Λίστα Διαγραμμάτων

Εισαγωγή

Τα Analytics είναι η συστηματική υπολογιστική ανάλυση δεδομένων για την ανακάλυψη, ερμηνεία και επικοινωνία σημαντικών προτύπων σε δεδομένα. Ειδικότερα η ανάλυση αθλητικών γεγονότων αποτελεί την διαχείριση δομημένων ιστορικών δεδομένων, την εφαρμογή προγνωστικών αναλυτικών μοντέλων και την χρήση πληροφοριακών συστημάτων για την ενημέρωση των υπευθύνων λήψης αποφάσεων. Το National Basketball Association (NBA) είναι ένα επαγγελματικό πρωτάθλημα καλαθοσφαίρισης στη Βόρεια Αμερική και θεωρείται το καλύτερο και πιο απαιτητικό πρωτάθλημα στον κόσμο. Η συλλογή στατιστικών των play-by-play δεδομένων αποτελεί σαφώς την πιο αναλυτική στατιστική πληροφορία μιας αναμέτρησης αφού καταγράφει κάθε γεγονός – στιγμιότυπο με ένα πλήθος αρκετών χαρακτηριστικών, αποτυπώνοντας την λεπτομερής εξέλιξη του. Στο πλαίσιο της παρούσας εργασίας δημιουργείται ένα μοντέλο νευρωνικού δικτύου και πιο συγκεκριμένα ενός δικτύου μακράς βραχύχρονης μνήμης (LSTM), που με την είσοδο μέρους στατιστικών δεδομένων μιας αναμέτρησης καλαθοσφαίρισης έχει την ικανότητα της πρόβλεψης του τελικού αποτελέσματος.

Ειδικότερα στο πλαίσιο της παρούσας εργασίας κατά την διάρκεια του πρώτου κεφαλαίου αναφέρονται έννοιες σχετικές με την ανάλυση δεδομένων και ειδικότερα της ανάλυσης αθλητικών δεδομένων. Επίσης αναλύονται εφαρμογές χρήσης αυτών σε διάφορες κατηγορίες όπως η ανάπτυξη των αθλητικών συλλόγων, η πρόληψη των τραυματισμών και τα συστήματα αισθητήρων αθλητικών εγκαταστάσεων.

Το δεύτερο κεφάλαιο είναι αφιερωμένο στα δίκτυα μακράς βραχύχρονης μνήμης (LSTM), μια αρχιτεκτονική που επιλύει τα προβλήματα που παρουσιάζουν τα παραδοσιακά ανατροφοδοτούμενα νευρωνικά δίκτυα.

Στο τρίτο κεφάλαιο περιγράφεται η δομή των πρωτογενών δεδομένων – dataset, καθώς και τα αναλυτικά βήματα διαμόρφωσης που προηγήθηκαν, πριν την εισαγωγή τους στο σύστημα υλοποίησης.

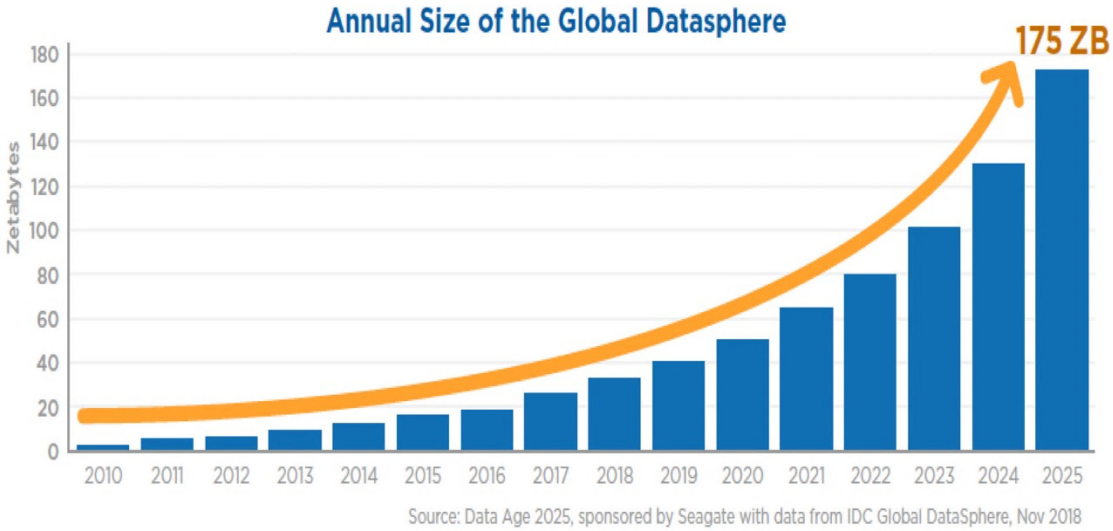
Στο τέταρτο κεφάλαιο αναφέρεται αναλυτικά η μοντελοποίηση που πραγματοποιήθηκε καθώς και στην αρχιτεκτονική του μοντέλου.

Το πέμπτο κεφάλαιο εμπεριέχει αναλυτικά τα πειράματα που πραγματοποιήθηκαν καθώς και τα αποτελέσματα αυτών ανά περίπτωση.

Ακολουθεί ξεχωριστή παράγραφός για τις μελλοντικές εργασίες όπου αναπτύσσονται ιδέες που μπορούν να προεκτείνουν σημαντικά την παρούσα εργασία.

1 Ανάλυση Δεδομένων

Ο ρυθμός με τον οποίο δημιουργούμε πληροφορίες αυξάνεται εδώ και χρόνια με λίγο πολύ προβλέψιμο ρυθμό. Ως εκ τούτου, μπορούμε να προβλέψουμε με σχετική ασφάλεια ότι μέχρι το 2025 θα υπάρχουν 175 zettabyte δεδομένων. [1]



Εικόνα 1: Ετήσιο Μέγεθος Δεδομένων

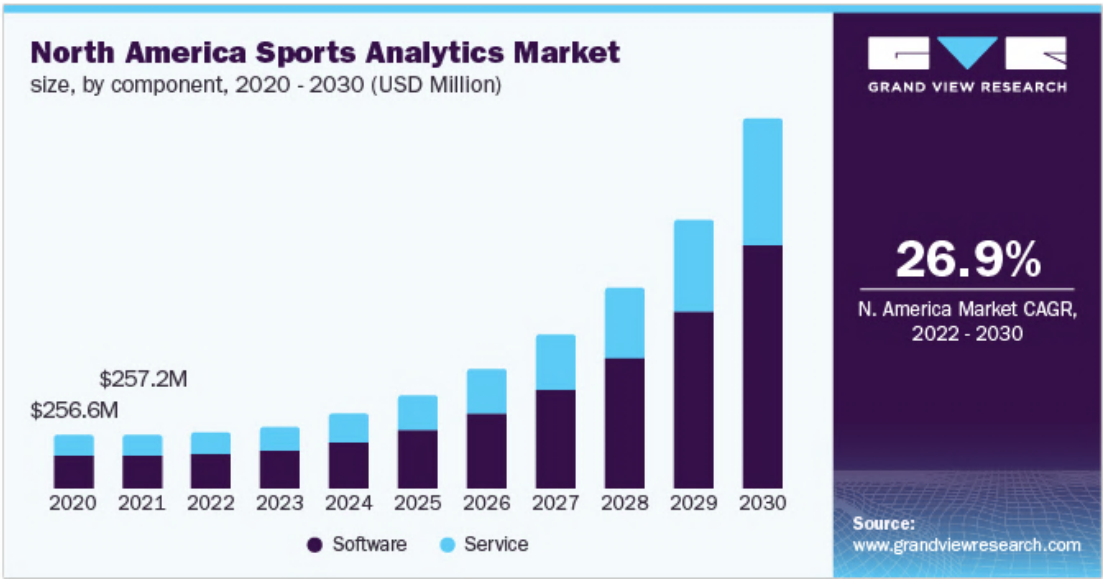
Τα Analytics είναι η συστηματική υπολογιστική ανάλυση δεδομένων ή στατιστικών για την ανακάλυψη, ερμηνεία και επικοινωνία σημαντικών προτύπων σε δεδομένα. [2] Η αποτελεσματική λήψη αποφάσεων προϋποθέτει την εφαρμογή προτύπων δεδομένων για την ποσοτικοποίηση της απόδοσης τους, με την ταυτόχρονη εφαρμογή επιχειρησιακής έρευνας, στατιστικών και προγραμματισμού υπολογιστών.

Η ανάλυση δεδομένων μπορεί να χωριστεί σε τέσσερις βασικούς τύπους, την περιγραφική ανάλυση (Descriptive), την ανάλυση πρόβλεψης (Predictive), την διαγνωστική ανάλυση (Diagnostic) και την ανάλυση της προστακτικής (Prescriptive). [3] Η προστακτική ανάλυση είναι μια μορφή επιχειρηματικής ανάλυσης που προτείνει επιλογές απόφασης για το πώς να εκμεταλλευτεί κανείς μια μελλοντική ευκαιρία ή να μετριάσει έναν μελλοντικό κίνδυνο και δείχνει τις επιπτώσεις κάθε επιλογής απόφασης. Επιτρέπει σε μια επιχείρηση να εξετάσει την βέλτιστη πορεία δράσης, υπό το φως των πληροφοριών που προέρχονται από περιγραφικές και προγνωστικές αναλύσεις. [4] Η προγνωστική ανάλυση περιλαμβάνει μια ποικιλία στατιστικών τεχνικών από εξόρυξη δεδομένων, προγνωστική μοντελοποίηση και μηχανική μάθηση που αναλύουν τρέχοντα και ιστορικά γεγονότα για να κάνουν προβλέψεις για μελλοντικά ή άγνωστα γεγονότα. [5] Η περιγραφική ανάλυση περιγράφει ποσοτικά ή συνοψίζει χαρακτηριστικά από μια συλλογή πληροφοριών. [6] Η διαγνωστική ανάλυση εστιάζει περισσότερο στη απάντηση του ερωτήματος «γιατί» συνέβη κάτι.

Συνήθως κάνει μια βαθιά «βουτιά» στα δεδομένα για την αναζήτηση πολύτιμών πληροφοριών, με στόχο να αποκαλύψει το σκεπτικό πίσω από τα αποτελέσματα.
Η αύξηση της υπολογιστικής ισχύς των τελευταίων ετών έχει επιτρέψει την ανάλυση δεδομένων με τις πιο σύγχρονες μεθόδους τόσο στα μαθηματικά και την στατιστική όσο και στην επιστήμη των υπολογιστών. Η συνήθεις εφαρμογή γίνεται στα οικονομικά, το μάρκετινγκ, τα αθλητικά γεγονότα, την ασφάλεια πληροφοριών και γενικότερα σε όλες τις εν δυνάμει υπηρεσίες λογισμικού.
Η παρούσα εργασία εστιάζει στην ανάλυση πρόβλεψης – Predictive Analysis.

1.1 Ανάλυση Αθλητικών Γεγονότων

Η ανάλυση αθλητικών γεγονότων αποτελεί την διαχείριση δομημένων ιστορικών δεδομένων, την εφαρμογή προγνωστικών αναλυτικών μοντέλων και την χρήση πληροφοριακών συστημάτων για την ενημέρωση των υπευθύνων λήψης αποφάσεων με τη δυνατότητα να βοηθήσουν τους εμπλεκόμενους οργανισμούς να αποκτήσουν ανταγωνιστικό πλεονέκτημα στον αγωνιστικό χώρο. [7] [8] Συνεπώς ο τρόπος με τον οποίο οι προπονητές παίρνουν αποφάσεις εντός του παιχνιδιού, την πρόσληψη παικτών και όλα τα ενδιάμεσα είναι πλέον θεμελιωδώς διαφορετικό από αυτό που ήταν μόλις πριν από δέκα χρόνια.



Εικόνα 2: Ετήσιο Μέγεθος Αγοράς | Sport Analytics (πηγή: **[10]**)

Το μέγεθος της παγκόσμιας αγοράς αθλητικών αναλυτικών στοιχείων αναμένεται να φτάσει τα 4,6 δισεκατομμύρια δολάρια ΗΠΑ έως το 2025, αυξάνοντας σε CAGR 31,2%,

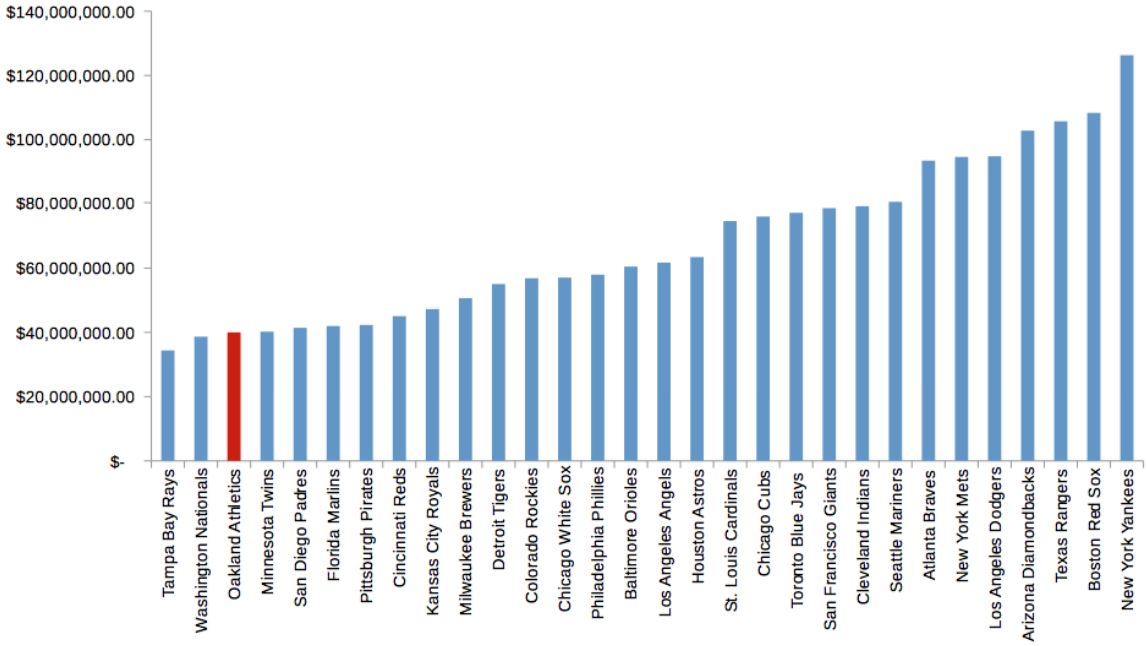
σύμφωνα με μια νέα μελέτη της Grand View Research, Inc. [9] Η χρήση της ανάλυσης αθλητικών γεγονότων βοηθά διαφορετικούς ενδιαφερόμενους φορείς, συμπεριλαμβανομένων των αθλητών, των ενώσεων και των φιλάθλων να λάβουν πληροφορίες τόσο για τη ζωντανή δραστηριότητα εντός του παιχνιδιού, όσο και για την ιστορικότητα του συγκεκριμένου ή παρόμοιων γεγονότων.

Στη ανάλυση αθλητικών γεγονότων υπάρχουν δύο βασικές κατηγορίες, η ανάλυση εντός (on-field) και εκτός (off-field) αγωνιστικού χώρου. Η ανάλυση εντός αγωνιστικού χώρου περιλαμβάνει την παρακολούθηση βασικών μετρήσεων δεδομένων στο γήπεδο με μεθοδολογίες που μπορούν να χρησιμοποιηθούν για τη βελτίωση των στρατηγικών εντός του παιχνιδιού καθώς και άλλων ζωτικών τομέων που θα μπορούσαν να ενισχύσουν αγωνιστικά ή και ηθικά τα επίπεδα απόδοσης των αθλητών. Η ανάλυση εκτός αγωνιστικού χώρου εστιάζει στην επιχειρηματική πλευρά του αθλητισμού, όπως οι πωλήσεις εισιτηρίων, οι πωλήσεις εμπορευμάτων και προσεγγίσεις για την αύξηση θαυμαστών του αθλητικού συλλόγου, βοηθώντας στην λήψη καλύτερων αποφάσεων με στόχο την αύξηση της ανάπτυξης και της κερδοφορίας. Η παρούσα εργασία εστιάζει στην πρώτη κατηγορία και στις μεθοδολογίας εντός του αγωνιστικού χώρου.

1.1.1 **Ανάπτυξη Ανάλυσης Αθλητικών Συλλόγων**

Το Moneyball: The Art of Winning an Unfair Game είναι ένα βιβλίο του Michael Lewis, που εκδόθηκε το 2003, σχετικά με την αληθινή ιστορία μιας ομάδα μπέιζμπολ του Oakland Athletics και τον γενικό διευθυντή της Billy Beane. Το Moneyball αναφέρεται ιδιαίτερα στα εργαλεία ανάλυσης αθλητικών δεδομένα αποδεικνύοντας στην πράξη την χρησιμότητά τους, αποτελώντας αργότερα τη βάση της ανάλυσης στο Μπέιζμπολ. Η μεθοδολογία του Paul DePodesta περιείχε βασικές στατιστικές μεθόδους για την εύρεση λύσεων σε έναν τομέα υποτιμημένων παικτών. Στόχος του ήταν να δημιουργήσει ένα πλαίσιο που θα αύξανε την πιθανότητα για να οδηγηθεί στα Playoffs, ξεκινώντας από την σκέψη ότι η εν λόγω πιθανότητα εξαρτιόταν από τον αριθμό των αγώνων που κέρδισε στην κανονική περίοδο. Συνεπώς αναζητούσε μια μέθοδο εύρεσης της βέλτιστης δομής της ομάδας του, ώστε στην συνέχεια να πετύχει ένα ικανό πλήθος νικών, στην κανονική διάρκεια, που θα τον οδηγήσουν στα Playoffs. Μεταξύ των στατιστικών εργαλείων, χρησιμοποιήθηκαν μοντέλα γραμμικής παλινδρόμησης σε αθλητικά δεδομένα πριν το 2001, για να προβλεφθεί το πλήθος των απαραίτητων νικών καθώς και των runs. Για τον σωστό συνδυασμό παικτών, έγινε χρήση των στατιστικών παικτών της προηγούμενης χρονιάς του 2001, με έμφαση στα ποσοστά του παίκτη στη «βάση» και όχι στις μεταβλητές μέσου όρου

χτυπήματος που είχαν μέχρι τότε υπερεκτιμηθεί. Το αποτέλεσμα ήταν μεταξύ του διαστήματος 2000 - 2006 η χαμηλού προϋπολογισμού ομάδα Oakland Athletics να σημειώσει κατά μέσο όρο 95 νίκες, κατακτώντας τέσσερις τίτλους American League West και έχοντας πέντε εμφανίσεις στα Playoffs. Η σημαντικότητα του στόχου γίνεται ευκολότερα αντιληπτή στην εικόνα 3 με τους προϋπολογισμούς ανά αθλητικό σύλλογο.



Εικόνα 3: Προϋπολογισμός Ομάδων Baseball 2002 | Oakland Athletics (Πηγή: **[11]**

Η ιστορία του Moneyball, εκτός από την σημαντικότητα των στατιστικών στον τομέα των αθλητικών γεγονότων, διδάσκει και την ιδιαίτερη χρησιμότητα να εντοπίσουμε τις σωστές μεταβλητές όταν κάνουμε προβλέψεις με σκοπό την παραγωγή σημαντικών συμπερασμάτων. Τα αθλητικά γεγονότα άλλωστε αποτελούν έναν κόσμο προκαθορισμένων κανόνων και προβλέψιμων συμπεριφορών.

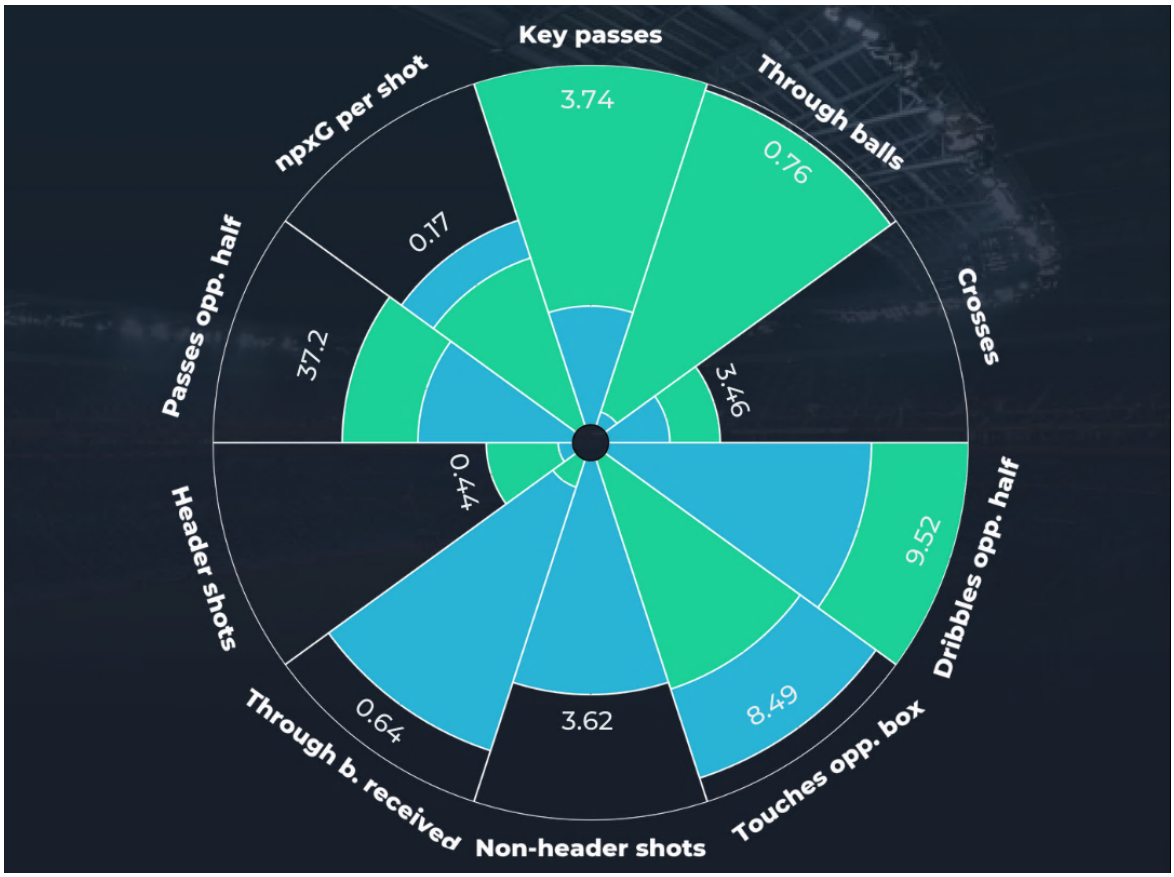
1.1.2 **Αισθητήρες Απόδοσης Παιχτών**

Η αυξανόμενη απαίτηση κάθε είδους στατιστικών στοιχείων σχετικά με του παίχτες που συμμετέχουν σε αθλητικά γεγονότα δημιούργησε την ανάγκη δημιουργίας μιας αγοράς συστημάτων φορητών αισθητήρων. Εκτός από την ιδανική σωματική διάπλαση του αθλητή που είναι μετρήσιμη και διαφέρει από άθλημα σε άθλημα είναι πλέον, μέρος της απαίτησης εργασίας των νέων προσλήψεων, τα δομικά χαρακτηριστικά που προσφέρουν σαφή πλεονεκτήματα στην απόδοση. Πιο συγκεκριμένα, δεδομένα όπως συνολική απόσταση που

διανύθηκε ή ακριβή θέση στον αγωνιστικό χώρο είναι πολύ εύκολο να καταγραφεί με ακρίβεια από μία φορητή συσκευή είτε κατά την διάρκεια της προπόνησης, είτε του αγώνα. Στο παρελθόν ανάλογα συστήματα ήταν απαγορευτικά κυρίως λόγω του κόστους, όμως πλέον τόσο διαδεδομένα που εφαρμόζονται ακόμη και σε ερασιτεχνικές περιπτώσεις.

Μια τυπική φορητή συσκευή συνήθως τοποθετείται στην άνω περιοχή της πλάτης του κορμού, σε μορφή «γιλέκου». Οι βασικοί αισθητήρες που παρέχουν άμεσες πληροφορίες, είναι συνήθως μια μονάδα GPS, ένα μαγνητόμετρο, ένα επιταχυνσιόμετρο και ένα γυροσκόπιο. Οι μονάδες GPS είναι πολύ φθηνές και χρήσιμες και κυρίως πολύ ακριβείς για τη συλλογή μετρικών απόστασης. Επειδή υστερούν λίγο στον ρυθμό δειγματοληψίας, οι μονάδες GPS δεν είναι ιδανικές για τη λήψη ακριβών ταχυτήτων και καμπυλών ταχύτητας των παικτών. Στο τομέα αυτό υποβοηθούνται από συστήματα που βρίσκονται εντός του γηπέδου σε συνδυασμό με άλλους αισθητήρες. Το επιταχυνσιόμετρο μετράει τις αλλαγές στους ρυθμούς της αντιληπτής δύναμης, ανιχνεύοντας την αλλαγή της επιτάχυνσης μέσα στη συσκευή. Ένα μαγνητόμετρο επεκτείνει την μέτρηση του επιταχυνσιόμετρο για να ενισχύσει τον συντονισμό των δεδομένων βοηθώντας τον προσανατολισμό της κατεύθυνσης των δεδομένων. Τέλος, το γυροσκόπιο βοηθάει στην παροχή προσανατολισμού χρησιμοποιώντας τη βαρύτητα της Γης και χρησιμοποιείται ενισχυτικά στο επιταχυνσιόμετρο για να δώσει κατεύθυνση στα δεδομένα. Εκτός από τους βασικούς αισθητήρες, υπάρχουν αναρίθμητοι αισθητήρες σε φορητές συσκευές με διαφοροποιήσεις από άθλημα σε άθλημα που μπορούν να λαμβάνουν συγκεκριμένα βιομετρικά δεδομένα, όπως καρδιακοί παλμοί, αρτηριακή πίεση, ηλεκτροκαρδιογράφημα, οξυγόνο αίματος, παλμούς αναπνοής και πίεσης ροπής άκρων.

Σήμερα το μεγαλύτερο βάρος του κόστους δεν είναι τόσο στους αισθητήρες απόδοσης παιχτών, όσο στους αλγορίθμους υπολογισμών για την ορθή κατανόηση των πρωτογενών δεδομένων και την μετατροπή αυτών σε χρήσιμη πληροφορία, ικανή εκτός από την απλή παρακολούθηση, για περαιτέρω αναλύσεις. Με βάση τα δεδομένα ο αλγόριθμος πρέπει να είναι σε θέση να αναγνωρίζει προφίλ αθλητών και να μπορεί να τα συγκρίνει με αναγνωρίσιμα χαρακτηριστικά κάθε αθλήματος. Συνεπώς μια ενδεχόμενη βαθμολογία ενός παίχτη προκύπτει από την ποσοτική εκτίμηση της ποιότητας απόδοσης του. Σε αυτήν την πρώτη εκτίμηση αξιολόγησης, δεν χρησιμοποιείται η μεροληπτική γνώμη ενός ειδικού του αθλήματος, αλλά οι αλγόριθμοί που μεταφράζουν τα ποιοτικά χαρακτηριστικά σε ποσοτικά. Στην εικόνα 4 εμφανίζεται ένα παράδειγμα σύγκρισης χαρακτηριστικών ποδοσφαίρου μεταξύ δυο πολύ γνωστών ποδοσφαιριστών, Kylian Mbappe και Neymar da Silva Santos Junior.



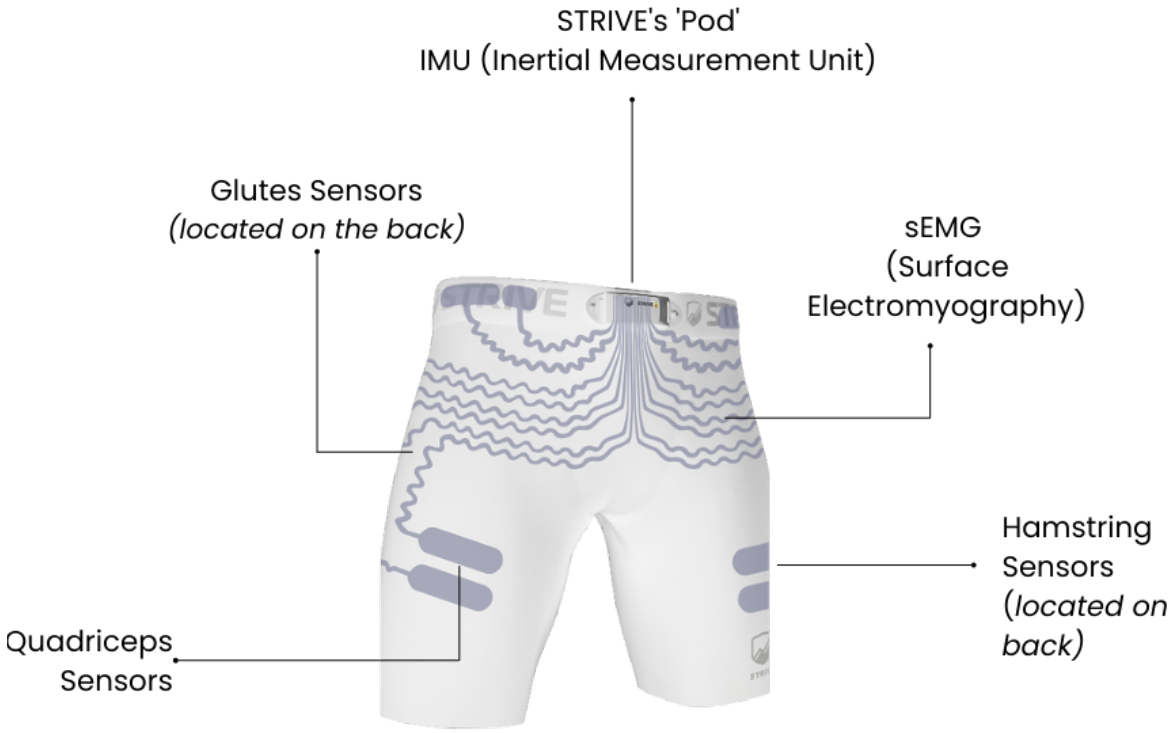
Εικόνα 4: Mbappe - Neymar Stats | 2020 – 2021 (πηγή: **[12]**)

1.1.3 **Πρόληψη Τραυματισμών Αθλητών**

Είναι σημαντικό για όλες τις πλευρές όπως αθλητές, αθλητικοί σύλλογοι, διοργανωτές και φιλάθλους να μην τραυματισμοί παιχτών. Τα σύγχρονα κορυφαία πρωταθλήματα μέσω των αθλητικών συλλόγων επικεντρώνονται στο θέμα της ισορροπίας πάνω στην ανάγκη διαρκούς διαθεσιμότητας των αθλητών βάση προγράμματος και στην προγραμματισμένη προσέγγιση ανάπαυσής τους, ώστε να μην υπάρξουν τραυματισμοί.

Οι αισθητήρες και οι αλγόριθμοι που περιεγράφηκαν στη προηγούμενη ενότητα μπορούν με κατάλληλο εμπλουτισμό να βοηθήσουν ώστε να εντοπίζονται προληπτικά επικίνδυνα χρονικά σημεία ενδιαφέροντος κατά την διάρκεια του αγώνα, αλλά είναι ακόμη σημαντικότερη η εν λόγω παρακολούθηση των δεδομένων κατά την προπόνηση. Η πρόληψη είναι σημαντική για τους τραυματισμούς και η όποια καταπόνηση ξεκινάει από την προπόνηση και το αποτέλεσμα είναι αυτό που θα προκύψει σε έναν αγώνα, εξαιρώντας τις κακές στιγμές ενός ατυχούς απρόβλεπτου συμβάντος. Σε αυτή την κατεύθυνση χρησιμοποιούνται αισθητήρες που μετρούν τις επιδράσεις ειδικής δύναμης στα άκρα με μεγάλη ακρίβεια όπως στο παράδειγμα του ποδιού ξεχωριστές μετρήσεις σε δάκτυλα, στο

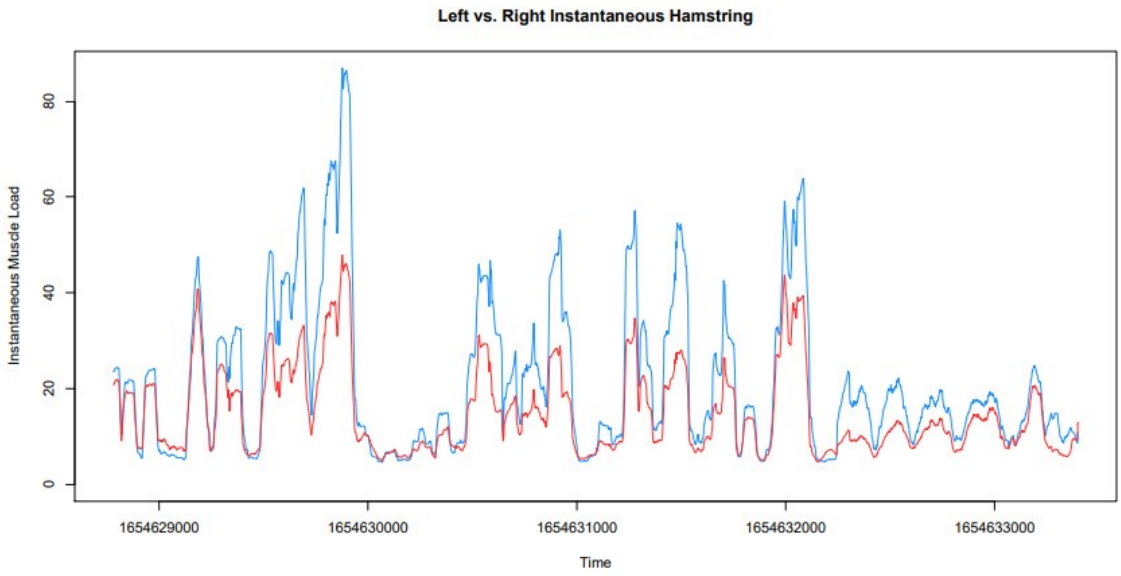
πέλμα, στην φτέρνα, το αστράγαλο, γωνίες βάδισης, προσανατολισμός κα. Επιπρόσθετα, η ύπαρξη ρουχισμού που επικεντρώνεται στην παρακολούθηση της απόδοσης των μυών, με αισθητήρες ηλεκτρομυογραφίας (EMG) που μετρούν μικρά ηλεκτρικά σήματα των μυών.



Εικόνα 5: Σορτς Συμπίεσης με αισθητήρες ηλεκτρομυογραφίας

Το σορτς που εμφανίζεται στην εικόνα 5 μπορεί να συλλέξει δεδομένα όπως το πλάτος των μυών από τις διάφορες περιοχές του σώματος όπως του γλουτούς, τους μηριαίους και τον τετρακέφαλο. Φυσικά τα δεδομένα των αισθητήρων αποτελούν εισόδους για τους κατάλληλους αλγορίθμους όπου θα μπορούν να εκτιμήσουν τα αποτελέσματα, δίνοντας σε δεύτερο χρόνο ακόμη και προτάσεις αποκατάστασης. Για παράδειγμα στην εικόνα 6 φαίνεται ο εντοπισμός μιας διαφοράς μεταξύ αριστερού και δεξιού μηριαίου, που μεταφράζεται σε αποκλίνοντα μυϊκά φορτία σε κάθε πλευρά.

Η συστηματική καταγραφή του συνολικού μυϊκού φορτίου μπορεί να οδηγήσει σε αποφάσεις κομβικές για μια διοργάνωση όπως η βελτίωση του προγραμματισμό των ημερομηνιών διεξαγωγής των αγώνων, αποφεύγοντας ενδεχόμενες πιέσεις των αθλητών σε συγκεκριμένους περιόδους.



Εικόνα 6: Σύγκριση αριστερού και δεξιού μηριαίου

1.1.4 Συστήματα Αισθητήρων Αθλητικών Εγκαταστάσεων

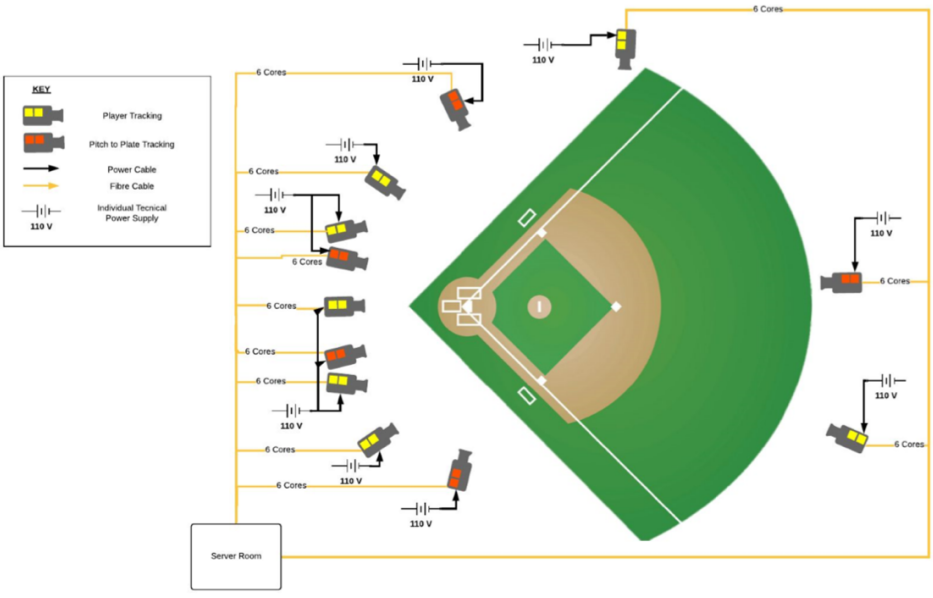
Τα συστήματα αισθητήρων στους αγωνιστικούς χώρους δεν αποτελεί μια καινούργια τακτική. Η αρχή έγινε στο Μπέιζμπολ - Major League Baseball (MLB) στην εποχή του moneyball, το οποίο και έχει πλέον μετατραπεί σε μια διοργάνωση με τα περισσότερα δεδομένα. Μέχρι τη σεζόν του 2015, όλα τα στάδια του MLB είχαν ενσωματώσει το σύστημα Statcast για την παρακολούθηση της μπάλας και του κάθε παίκτη στον αγωνιστικό χώρο. [13] Η τεχνολογία που χρησιμοποιείται για την παρακολούθηση της μπάλας από το σύστημα The TrackMan ονομάζεται Doppler Radar. Ανάμεσα στις δυνατότητες του συστήματος είναι η προσέγγιση της διαδρομής, ο ρυθμός περιστροφής, η ταχύτητα κάθε βήματος, η αρχική ταχύτητα καθώς και οι οριζόντιες και κατακόρυφες γωνίες εκτόξευσης της μπάλας. [14] Επίσης, το Εθνικό πρωτάθλημα Ποδοσφαίρου της Βόρειας Αμερικής, National Football League (NFL), υιοθέτησε περισσότερη τεχνολογία παρακολούθησης δεδομένων όπως το RFID, για την παρακολούθηση της κίνησης των παικτών και της απόστασης που αυτοί τρέχουν σε πραγματικό χρόνο. Η Εθνική Ομοσπονδία Καλαθοσφαίρισης (NBA) ακολούθησε από τα τέλη της δεκαετίας του 2000 την εγκατάσταση ειδικών καμερών σε όλους τους αγωνιστικούς χώρους για καλύτερη παρακολούθηση της απόδοσης των παικτών. Οι κάμερες παρακολούθησης του συστήματος ονόματι SportVU και Second Spectrum στην εξέλιξή του, χρησιμοποιεί έξι κάμερες που εστιάζουν στο περίγραμμα των παικτών. Οι κάμερες παρακολουθούν τα πάντα κατά τη διάρκεια μιας αναμέτρησης, με στόχο την καταγραφή κινήσεων, θέσεων και στατιστικών όπως βολές, blocks και περισσότερες από 500 άλλες γνωστά προφίλ κινήσεων

ανά παίκτη. Οι εν λόγω πληροφορίες αποστέλλονται στους συλλόγους για περαιτέρω ανάλυση από τμήματα αναλυτών ώστε να προκύψουν συμπεράσματα που μπορούν πραγματικά να βελτιώσουν την απόδοση της ομάδας.

Το NBA όμως συνέχισε να εξελίσσεται πραγματοποιώντας το καλοκαίρι του 2021 εκτεταμένες δοκιμές του συστήματος παρακολούθησης της επόμενης γενιάς Hawk-Eye της Sony τόσο για εφαρμογές μετάδοσης όσο και για εφαρμογές παιχνιδιών. Η Sony Hawk-Eye δοκιμαστικά εγκατέστησε 14 κάμερες στο γήπεδο του Thomas & Mack Center στο Las Vegas με δυνατότητα παρακολούθησης 18 θέσεων του ανθρώπινου σώματος των αθλητών καθώς και τη θέση της μπάλας σε τρεις διαστάσεις και εξαιρετική ακρίβεια.

Είναι σημαντικό να αναφερθεί πως το εν λόγω σύστημα δεν περιορίζεται μόνο στην παρακολούθηση των παιχτών αλλά και στην παροχή δεδομένων στους διαιτητές, που θα λειτουργούν επικουρικά στις αποφάσεις τους.

Καταλήγοντας φαίνεται πως δεν υπάρχει σημείο σύγκρισης με το παρελθόν αφού για παράδειγμα σε ένα σουτ στη καλαθοσφαίριση γνωρίζαμε το στατιστικό της ευστοχίας ή της αστοχίας του και πλέον έχουμε φτάσει στο σημείο να γνωρίζουμε που ακριβώς χτύπησε η μπάλα, ταμπλό ή στη στεφάνη και ακριβώς το σημείο αφού μπορούμε να ξέρουμε αν ήταν στο εσωτερικό της στεφάνης ή στο εξωτερικό καθώς και την συγκεκριμένη γωνία και τροχιά της μπάλας.



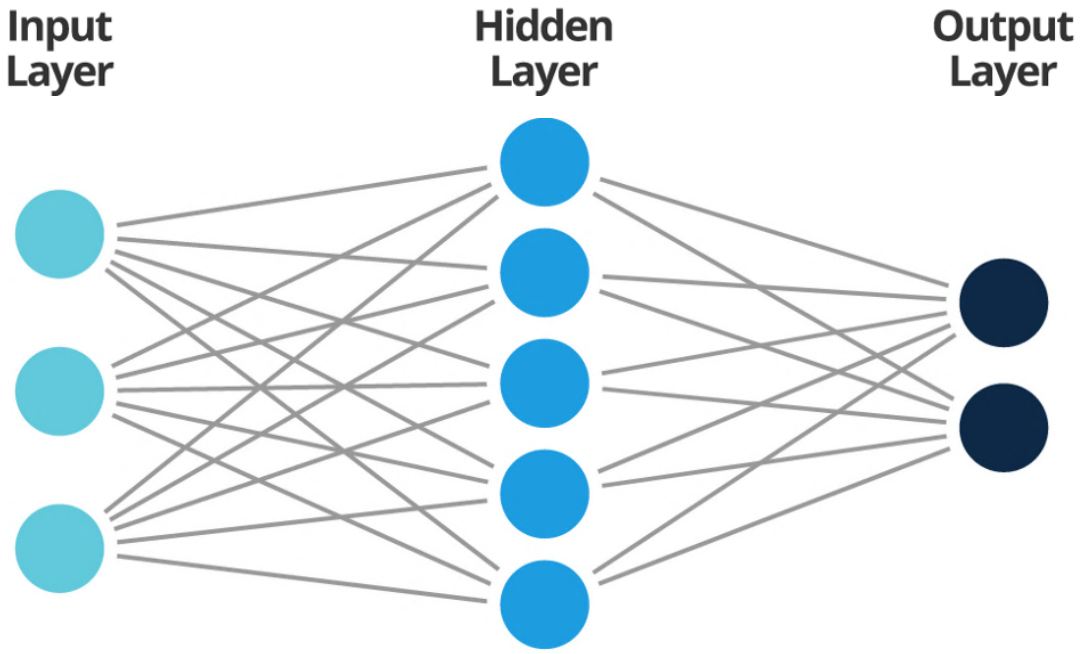
Εικόνα 7: Sony Hawk-Eye Technology | MLB (πηγή: **[15]**)

2 Τεχνητή Νοημοσύνη

Η τεχνητή νοημοσύνη μπορεί να περιγραφεί συνοπτικά ως η προσπάθεια αυτοματοποίησης των πνευματικών εργασιών που συνήθως εκτελούνται από ανθρώπους. Στο παρόν κεφάλαιο θα αναφερθούν οι βασικές έννοιες των τεχνητών νευρωνικών δικτύων, της μηχανικής μάθησης και βαθιάς μάθησης με ιδιαίτερη έμφαση και επικέντρωση στα δίκτυα μακράς βραχύχρονης μνήμης (LSTM).

2.1 Τεχνητά Νευρωνικά Δίκτυα

Ένα τεχνητό νευρωνικό δίκτυο αποτελείται από τεχνητούς νευρώνες και χρησιμοποιείται για την επίλυση προβλημάτων τεχνητής νοημοσύνης (AI). [16] Τα τεχνικά νευρωνικά δίκτυα έχουν συγκεκριμένη μορφή αρχιτεκτονικής, εμπνευσμένη από ένα βιολογικό νευρικό σύστημα, όπως η δομή του ανθρώπινου εγκεφάλου που αποτελείται από νευρώνες σε πολύπλοκη και μη γραμμική μορφή. Οι νευρώνες συνδέονται μεταξύ τους με σταθμισμένους συνδέσμους και μέσω μεθόδων εκμάθησης και εκπαίδευσης, πραγματοποιείται η συλλογή και ανάλυση δεδομένων, ο σχεδιασμός της δομής του δικτύου, ο αριθμός των κρυφών επιπέδων και η προσομοίωση του δικτύου. Τα τεχνικά νευρωνικά δίκτυα βρίσκουν εφαρμογή σε τρία βασικά επιστημονικά πεδία όπως η ταξινόμηση, η πρόβλεψη, ο έλεγχος και η βελτιστοποίηση.

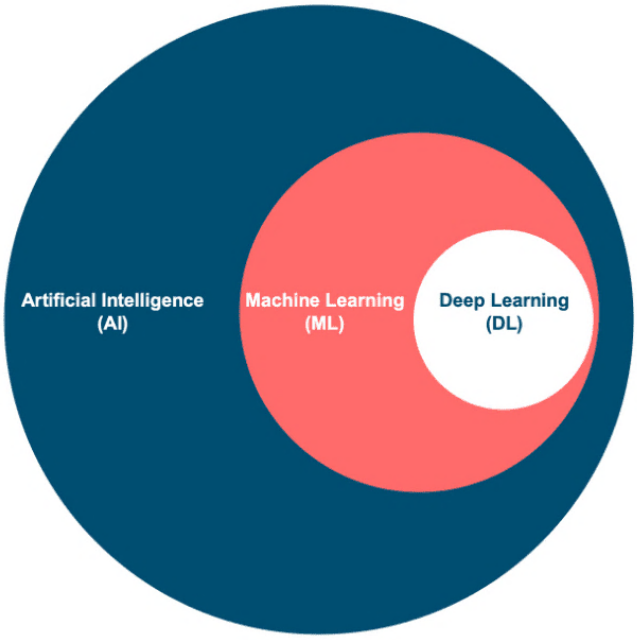


Εικόνα 8: Τεχνικά Νευρωνικά Δίκτυα

Ένα τεχνητό νευρωνικό δίκτυο είναι οργανωμένο σε τρία βασικά επίπεδα, το επίπεδο εισόδου, το κρυφό επίπεδο και το επίπεδο εξόδου. Σκοπός του επιπέδου εισόδου είναι να λάβει ως είσοδο τις τιμές των χαρακτηριστικών για κάθε παρατήρηση. Το πλήθος των κόμβων εισόδου είναι ίσο με τον πλήθος των χαρακτηριστικών. Το επίπεδο εισόδου επικοινωνεί με ένα ή περισσότερα κρυφά επίπεδα όπου εφαρμόζονται οι μετασχηματισμοί μέσα στο δίκτυο. Η πραγματική επεξεργασία γίνεται μέσω ενός συστήματος σταθμισμένων συνδέσεων αφού οι τιμές πολλαπλασιάζονται με βάρη για να προστεθούν και στη συνέχεια να παραχθεί ένας μόνο αριθμός. Τέλος, τα κρυφά επίπεδα συνδέονται με το επίπεδο εξόδου που λαμβάνει συνδέσεις και επιστρέφει μια τιμή εξόδου που αντιστοιχεί στην πρόβλεψη της μεταβλητής απόκρισης. Η ικανότητα του νευρωνικού δικτύου να παρέχει χρήσιμο χειρισμό δεδομένων έγκειται στη σωστή επιλογή των βαρών γεγονός που το κάνει να διαφέρει σημαντικά από τη συμβατική επεξεργασία πληροφοριών.

2.2 Μηχανική Μάθηση

Εστιάζοντας στην αξιοποίηση εφαρμοσμένων μαθηματικών τεχνικών και συγκεκριμένων αλγορίθμων για τη δημιουργία μιας πρόβλεψης συναντάμε ένα υποσύνολο της τεχνητής νοημοσύνης που ονομάζεται Μηχανική Μάθηση. [17] Τα μοντέλα μηχανικής μάθησης έχουν την ικανότητα να προσαρμόζονται και να «μαθαίνουν» με την πάροδο του χρόνου καθώς εκτίθενται συνεχώς σε νέα δεδομένα. Με βάση την διαρκή εκπαίδευση νέων δεδομένων, ο αλγόριθμος προσαρμόζει αυτόματα τις παραμέτρους για να ελέγξει, εφόσον υφίσταται, την όποια αλλαγή μοτίβου, χωρίς την αλλαγή του γενικότερου μοντέλου.



Εικόνα 9: Τεχνητή Νοημοσύνη | Μηχανική Μάθηση | Βαθιά Μάθηση

Οι περισσότεροι αλγόριθμοι μηχανικής μάθησης μπορούν να χωριστούν στις κατηγορίες της εποπτευόμενης μάθησης και της μάθησης χωρίς επίβλεψη.

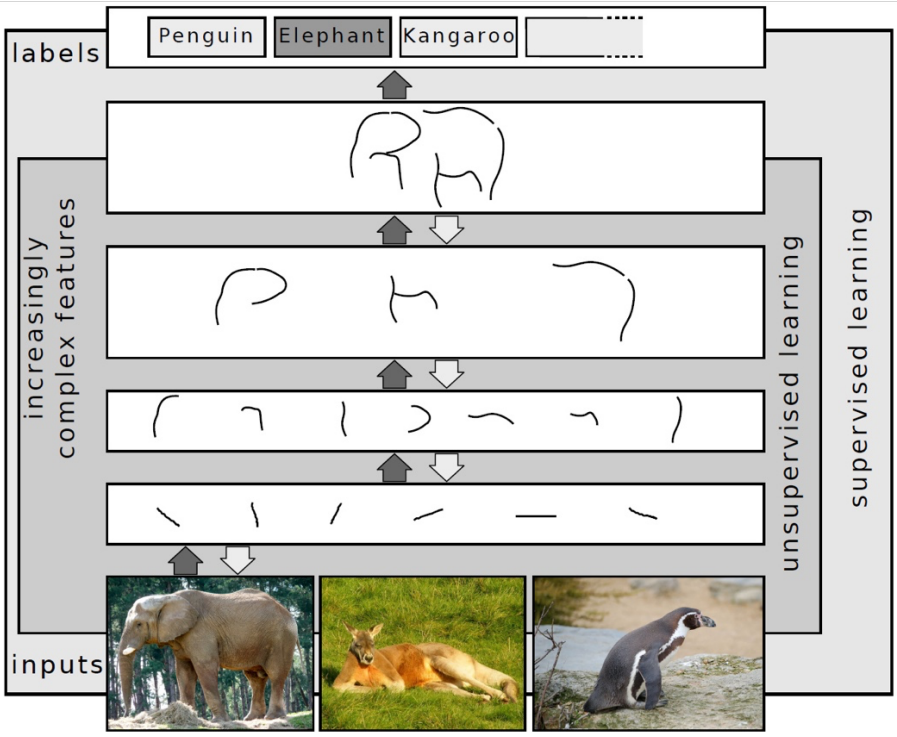
Η εποπτευόμενη μάθηση είναι ένα παράδειγμα μηχανικής μάθησης για προβλήματα όπου τα διαθέσιμα δεδομένα αποτελούνται από παραδείγματα με ετικέτα, που σημαίνει ότι κάθε σημείο δεδομένων περιέχει χαρακτηριστικά και μια σχετική ετικέτα. [18] Ο στόχος τους είναι η εκμάθηση μιας συνάρτησης που αντιστοιχίζει διανύσματα χαρακτηριστικών ως είσοδο, σε ετικέτες ως έξοδο. Συνεπώς, κάθε παράδειγμα είναι ένα ζεύγος που αποτελείται από ένα διάνυσμα εισόδου και την επιθυμητή τιμή εξόδου. Με αυτόν τον τρόπο καταφέρνει να αναλύει τα δεδομένα εκπαίδευσης και να παράγει μια συνάρτηση που προκύπτει και μπορεί να χρησιμοποιηθεί για τη χαρτογράφηση νέων παραδειγμάτων. Αλγόριθμοι εποπτευόμενης μάθησης θεωρούνται τα δένδρα αποφάσεων, η παλινδρόμηση και οι μηχανές διανυσμάτων υποστήριξης (SVM).

Η μέθοδος της μάθησης χωρίς επίβλεψη είναι ένας τύπος αλγορίθμου που μαθαίνει μοτίβα από δεδομένα χωρίς ετικέτα. Η ιδέα είναι ότι μέσω της μίμησης, που αποτελεί ένα σημαντικό τρόπο μάθησης των ανθρώπων, ο αλγόριθμος αναγκάζεται να οικοδομήσει μια συνοπτική αναπαράσταση της χρήσιμης πληροφορίας. [19] Γνωστοί αλγόριθμοι που χρησιμοποιούνται στην μάθηση χωρίς επίβλεψη είναι η ομαδοποίηση k-means και οι πιθανοτικές μέθοδοι ομαδοποίησης.

2.1 Βαθιά Μάθηση

Η βαθιά μάθηση είναι μέρος μιας ευρύτερης οικογένειας μεθόδων μηχανικής μάθησης που βασίζονται σε τεχνητά νευρωνικά δίκτυα με μάθηση αναπαράστασης. Η μάθηση μπορεί να είναι με ή χωρίς επίβλεψη. [9] Πιο συνοπτικά αποτελεί μια μέθοδο που χρησιμοποιεί πολλαπλά επίπεδα για να εξάγει σταδιακά χαρακτηριστικά υψηλότερου επιπέδου από μια ακατέργαστη είσοδο. Οι περισσότεροι αλγόριθμοι βαθιάς μάθησης βασίζονται σε έναν αλγόριθμο βελτιστοποίησης που ονομάζεται στοχαστική διαβάθμιση. Κάθε επίπεδο μαθαίνει να μετατρέπει τα δεδομένα εισόδου του σε μια ελαφρώς πιο αφηρημένη και σύνθετη αναπαράσταση. Είναι σημαντικό ότι μια διαδικασία βαθιάς μάθησης μπορεί να μάθει από μόνη της ποια χαρακτηριστικά θα τοποθετήσει βέλτιστα και σε ποιο επίπεδο, εξαλείφοντας την ανάγκη για χειροκίνητο συντονισμό του συστήματος. Για παράδειγμα στην εικόνα 5, φαίνεται παραστατικά η διαδικασία μεταξύ των επιπέδων που επιτρέπει τη σταδιακή αναγνώριση ενός ελέφαντα. Η αρχική ακατέργαστη είσοδος μπορεί να είναι μια μήτρα εικονοστοιχείων, όπου στο πρώτο επίπεδο προκύπτουν οι άκρες, στο δεύτερο η

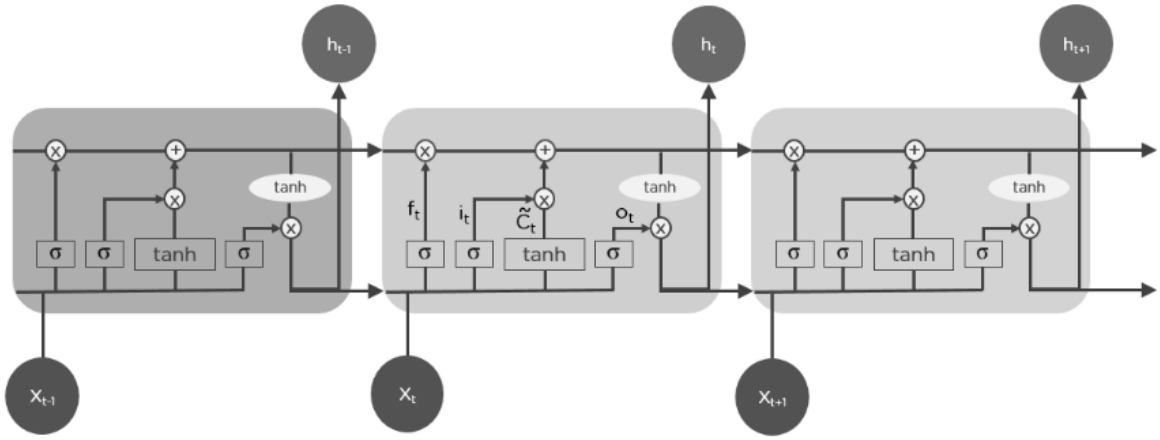
αποκωδικοποίηση αυτών για να ακολουθήσει στο τρίτο και τέταρτο επίπεδο η σύνθεση και η αναγνώριση της εικόνας. Ανάμεσα στους γνωστότερους αλγορίθμους βαθιάς μάθησης, βρίσκουμε τα Recurrent Neural Networks (RNN) και τα Long Short Term Memory (LSTM).



Εικόνα 10: Αναπαράσταση εικόνων σε πολλαπλά επίπεδα (πηγή: **[20]**)

2.2 Δίκτυα Μακράς Βραχύχρονης Μνήμης

Τα Δίκτυα Μακράς Βραχύχρονης Μνήμης (LSTM) έχουν την ικανότητα να μάθουν και να απομνημονεύσουν μακροπρόθεσμες εξαρτήσεις. Σε αντίθεση με τα τυπικά νευρωνικά δίκτυα ανάδρασης, έχουν συνδέσεις ανάδρασης πετυχαίνοντας την επεξεργασία όχι μόνο μεμονωμένων σημείων δεδομένων, αλλά και ολόκληρων ροών αυτών. Η αρχιτεκτονική LSTM αναλαμβάνει να παρέχει μια βραχυπρόθεσμη μνήμη για το τεχνικό νευρωνικό δίκτυο, ενώ ταυτόχρονα μπορεί να διαρκέσει χιλιάδες χρονικά βήματα μετατρέποντας την σε μακροπρόθεσμη μνήμη. [21] Τα δίκτυα LSTM είναι κατάλληλα για την ταξινόμηση, την επεξεργασία και την πραγματοποίηση προβλέψεων με δεδομένα ειδικά χρονοσειρών, αφού μπορεί να υπάρξουν ενδεχόμενες καθυστερήσεις άγνωστης διάρκειας μεταξύ σημαντικών γεγονότων.



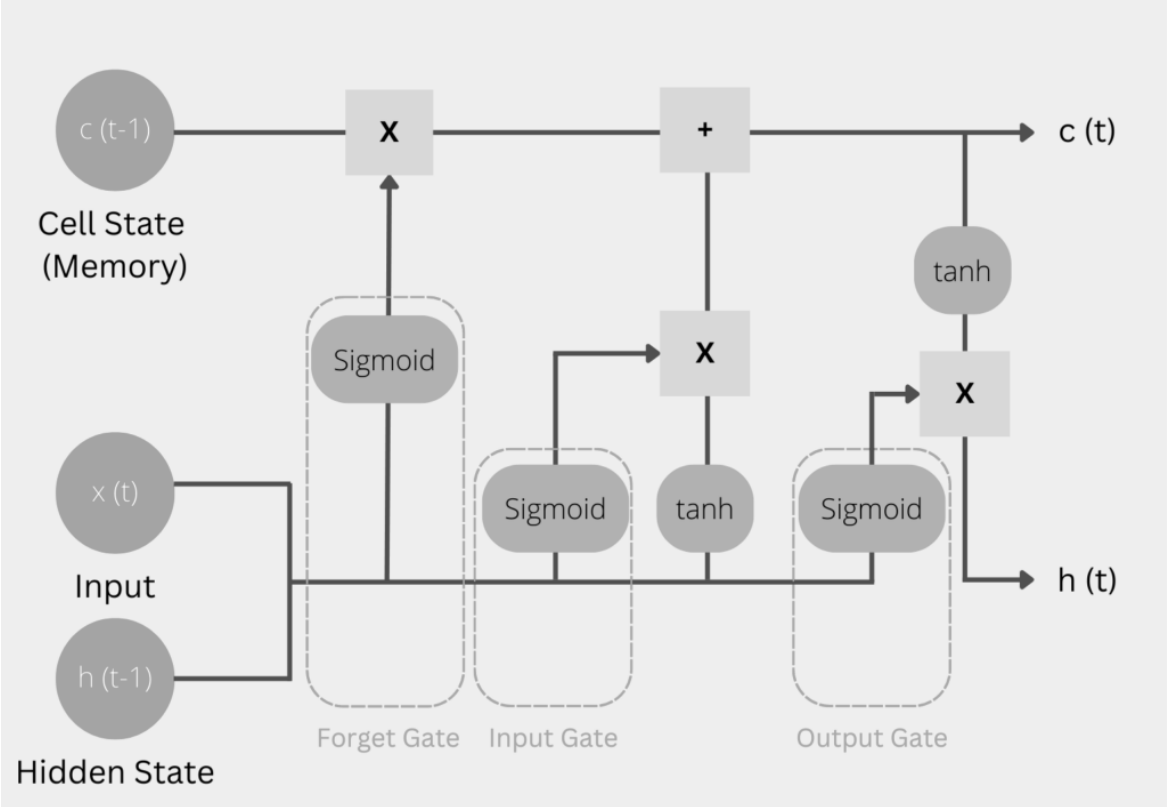
Εικόνα 11: Διάγραμμα Δικτύου Μακράς Βραχύχρονης Μνήμης (πηγή: **[22]**)

Για αναλύσουμε σε βάθος την λειτουργία των Δικτύων Μακράς Βραχύχρονης Μνήμης - LSTM είναι σημαντικό να κατανοήσουμε τα Recurrent Neural Networks - RNN. Στα κλασσικά τεχνητά νευρωνικά δίκτυα, ένας τουλάχιστον νευρώνας του κρυφού επιπέδου συνδέεται με τους νευρώνες από το προηγούμενο και από το επόμενο επίπεδο, με την έξοδο ενός νευρώνα να μπορεί να μεταβιβαστεί μόνο προς τα εμπρός και ποτέ στο ίδιο ή στο προηγούμενο επίπεδο. Όταν εκπαιδεύονται τα νευρωνικά δίκτυα με τη μέθοδο gradient descent, μπορεί η κλίση να λάβει είτε πολύ μικρές τιμές κοντά στο 0, είτε πολύ μεγάλες τιμές κοντά στο άπειρο, με αποτέλεσμα να μην είναι εφικτή η αλλαγή των βαρών στους νευρώνες κατά την οπίσθια διάδοση, καθώς το βάρος είτε δεν αλλάζει καθόλου είτε δεν μπορούμε να πολλαπλασιάσουμε τον αριθμό με τόσο μεγάλη τιμή.

Τα RNN έφεραν μια σημαντική καινοτομία στον τομέα της γλωσσικής επεξεργασίας και γενικότερα της βαθιάς μάθησης καθώς για πρώτη φορά συμπεριλήφθηκαν και οι υπολογισμοί από το πρόσφατο παρελθόν με αποτέλεσμα να «θυμούνται» περιβάλλοντα και να τα ενσωματώνουν στις προβλέψεις. Ωστόσο κατά την διάρκεια της εκπαίδευσης έχουνε ορισμένα σημαντικά προβλήματα, όπως το γεγονός ότι διαθέτουν βραχύχρονης μνήμη για να διατηρήσουν προηγούμενες πληροφορίες στον τρέχοντα νευρώνα. Βραχύχρονη, σε βαθμό που εν λόγω ικανότητα μειώνεται πολύ γρήγορα για μεγαλύτερες ακολουθίες.

Τα LSTM αποφεύγουν το εν λόγω πρόβλημα διατηρώντας μόνο επιλεγμένες πληροφορίες στη βραχύχρονη μνήμη τους. Ποιο συγκεκριμένα στην εικόνα 7 που φαίνεται η αρχιτεκτονική του LSTM, η λεγόμενη forget gate αποφασίζεται ποιες τρέχουσες και προηγούμενες πληροφορίες φυλάσσονται καθώς και ποιες μένουν εκτός. Οι τιμές μεταβιβάζονται σε μια σιγμοειδή συνάρτηση, η οποία μπορεί να εξάγει μόνο τιμές μεταξύ 0 και 1. Η τιμή 0 σημαίνει ότι όλες οι προηγούμενες πληροφορίες έχουν ξεχαστεί και 1

αντίστοιχα ότι όλες οι προηγούμενες πληροφορίες διατηρούνται. Στην συνέχεια τα αποτελέσματα πολλαπλασιάζονται με την τρέχουσα μνήμη cell state, έτσι ώστε η πληροφορία που δεν χρειάζεται πλέον να «ξεχνιέται», καθώς πολλαπλασιάζεται επί 0 και έτσι εγκαταλείπεται. Στην input gate, αποφασίζεται πόσο πολύτιμη είναι η τρέχουσα είσοδος για την επίλυση της εργασίας. Συγκεκριμένα η τρέχουσα είσοδος πολλαπλασιάζεται με την hidden state και τον πίνακα βάρους της τελευταίας εκτέλεσης. Στη συνέχεια στην cell state, όλες οι πληροφορίες που φαίνονται σημαντικές στην input gate προστίθενται και σχηματίζουν τη νέα cell state c(t). Η c(t) ως νέα τρέχουσα κατάσταση της βραχυχρόνιας μνήμης θα χρησιμοποιηθεί στην επόμενη εκτέλεση. Η output gate υπολογίζεται στη συνέχεια στην hidden state του LSTM με την βοήθεια της σιγμοειδής συνάρτησης που αποφασίζει ποιες πληροφορίες μπορούν να έρθουν μέσω της output gate και, στη συνέχεια, η cell state πολλαπλασιάζεται αφού ενεργοποιηθεί με τη συνάρτηση tanh. Η tanh είναι μια μη γραμμική συνάρτηση ενεργοποίησης που ρυθμίζει τις τιμές που ρέουν μέσω του δικτύου, διατηρώντας τις τιμές μεταξύ -1 και 1. [23]

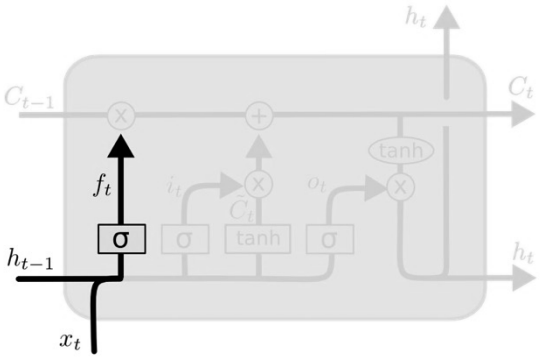


Εικόνα 12: Αρχιτεκτονική LSTM

2.3 Αναλυτικά Βήματα Επίλυσης LSTM

Το πρώτο βήμα του LSTM είναι η απόφαση ποιες πληροφορίες πρέπει να μείνουν εκτός της μνήμης cell state. Όπως αναφέρθηκε νωρίτερα αυτή η απόφαση λαμβάνεται με την βοήθεια μιας σιγμοειδής συνάρτησης:

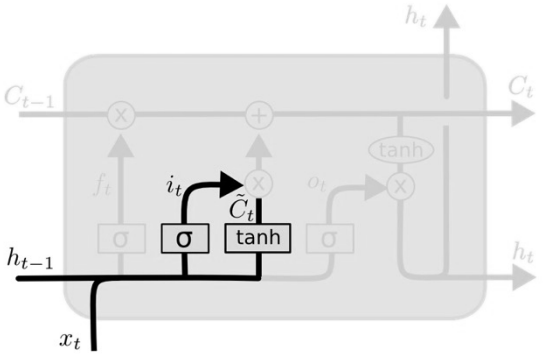
$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$



Εικόνα 13: 1ο Βήμα LSTM

Στο δεύτερο βήμα με την βοήθεια δύο επιπέδων αποφασίσουμε ποιες νέες πληροφορίες πρόκειται να αποθηκεύσουμε στην cell state. Στο πρώτο επίπεδο μια σιγμοειδής συνάρτηση αποφασίζει ποιες τιμές θα ενημερώσουμε και στο δεύτερο επίπεδο η γραμμική συνάρτηση tanh δημιουργεί ένα διάνυσμα νέων υποψήφιων τιμών, $\tilde{C}_t$ που εν δυνάμει θα μπορούσε να προστεθεί στην υπάρχουσα κατάσταση:

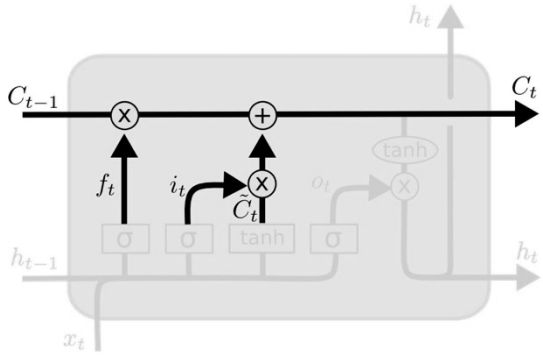
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$



Εικόνα 14: 2ο βήμα LSTM

Στο τρίτο βήμα ενημερώνεται η παλιά cell stateC_{t-1} στη νέα κατάσταση κελιού C_t. Προχωράμε κανονικά στην υλοποίηση όσων αποφασίστηκαν στα προηγούμενα βήματα. Η παλαιότερη κατάσταση πολλαπλασιάζεται με το f_t ώστε να μείνει η πληροφορία που αποφασίστηκε πριν εκτός. Στη συνέχεια το προσθέτουμε το $i_t * \tilde{C}_t$ που αποτελεί τις νέες υποψήφιες τιμές.

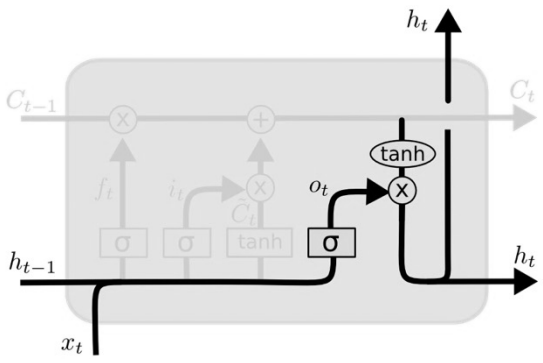
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



Εικόνα 15: 3ο Βήμα LSTM

Τέλος, αποφασίζεται η τελική έξοδος που βασίζεται σε μια φιλτραρισμένη έκδοση της cell state. Αρχικώς εφαρμόζουμε μια σιγμοειδές συνάρτηση οποία αποφασίζει ποια μέρη της cell state θα εξάγουμε και στη συνέχεια την πολλαπλασιάζουμε με την γραμμική συνάρτηση tanh.

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$



Εικόνα 16: 4ο βήμα LSTM

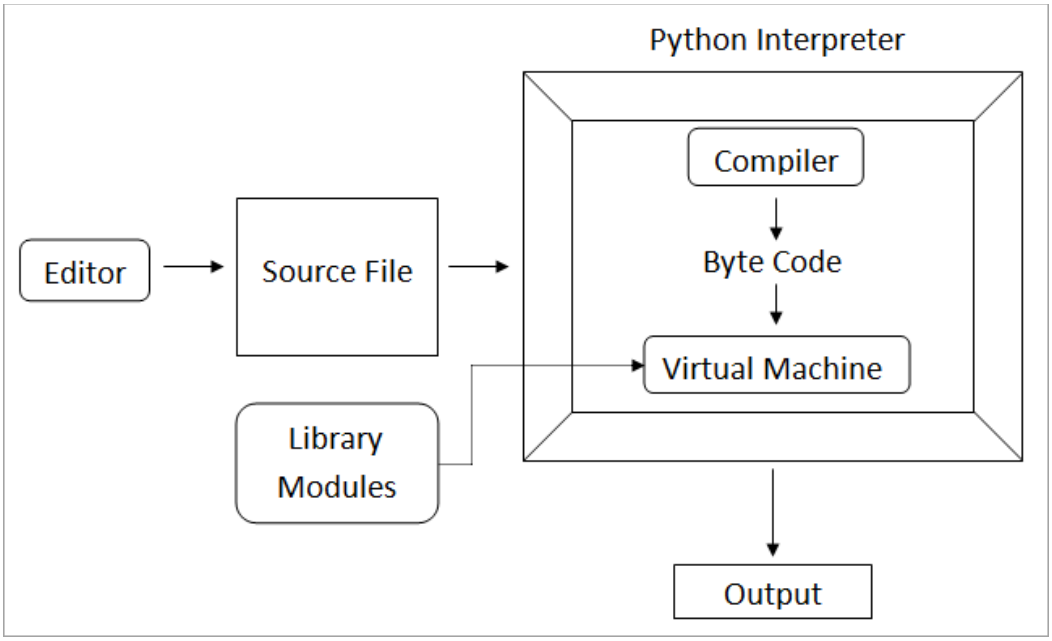
3 Περιγραφή Συνόλου Δεδομένων

Η ουσιαστική γνώση από μεγάλα ακατέργαστα δεδομένα εγκυμονεί κινδύνους από «θόρυβο», ακανόνιστες και ασυνεπείς τιμές. Τα ακατέργαστα δεδομένα θα αποδίδουν πάντα ανακριβή και άχρηστα αποτελέσματα ακόμα και μετά την ανάλυση μέχρι να «καθαριστούν» καλά. Συνεπώς, αν τα δεδομένα είναι λανθασμένα, τα αποτελέσματα και οι αλγόριθμοι θα είναι πάντα αναξιόπιστοι, παρόλο που μπορεί να φαίνονται σωστοί. Σε αυτό το κεφάλαιο αναφέρεται η πηγή των δεδομένων, καθώς και ο τρόπος συλλογής τους και περιγράφεται ο καθαρισμός των δεδομένων και η κανονικοποίησή τους -ενέργειες- ζωτικής σημασίας για τη διασφάλιση της ακεραιότητας τους.

3.1 Γλώσσα Προγραμματισμού | Python

Η Python είναι μια ανοιχτού κώδικα αντικειμενοστραφής γλώσσα προγραμματισμού υψηλού επιπέδου. Οι ενσωματωμένες δομές δεδομένων υψηλού επιπέδου, την καθιστούν πολύ ελκυστική για την γρήγορη ανάπτυξη εφαρμογών, καθώς και για χρήση ως script γλώσσα προγραμματισμού. Η Python υποστηρίζει αναρίθμητα module και packages, επιτρέποντας ταυτόχρονα τη σπονδυλωτή και ιεραρχική μορφή της δομής ενός προγράμματος ενισχύοντας την επαναχρησιμοποίηση κώδικα. Ο διερμηνέας Python και η εκτεταμένη τυπική βιβλιοθήκη της διατίθενται ελεύθερα για όλα τα λειτουργικά συστήματα, ενώ ο γρήγορος κύκλος επεξεργασίας – δοκιμής - εντοπισμού σφαλμάτων την καθιστά πολύ αποτελεσματική. Στην παρούσα εργασία κρίθηκε σκόπιμη η εκτενής χρήση της Python τόσο για την συλλογή των ακατέργαστων δεδομένων, τον καθαρισμό και την κανονικοποίηση αυτών. Επιπρόσθετα για τις ανάγκες της Μηχανικής Μάθησης χρησιμοποιήθηκαν packages που επέτρεψαν την εκπαίδευση, ανάλυση, έλεγχο και εφαρμογή των Δικτύων Μακράς Βραχύχρονης Μνήμης (LSTM) με στόχο την πρόβλεψη των αποτελεσμάτων αθλητικών γεγονότων.

Στην Python ένα τεχνητό νευρωνικό δίκτυα ορίζεται στο Keras ως μια ακολουθία επιπέδων. Ο container για αυτά τα επίπεδα είναι η κλάση Sequential όπου είναι απαραίτητο να δημιουργηθεί ένα στιγμιότυπο της κλάσης Sequential και στην συνέχεια να προστεθούν με τη σειρά που πρέπει να συνδεθούν. Το πακέτο NumPy είναι μια βιβλιοθήκη της Python απαραίτητη εδώ αφού παρέχει μια ποικιλία από ρουτίνες για γρήγορες λειτουργίες σε πίνακες, συμπεριλαμβανομένων μαθηματικών, λογικών, χειρισμού σχήματος, ταξινόμησης, επιλογής εισόδου/εξόδου, διακριτούς μετασχηματισμούς Fourier, βασική γραμμική άλγεβρα και ορισμένες βασικές στατιστικές πράξεις.



Εικόνα 17: Διάγραμμα Λειτουργίας Python

3.2 Δεδομένα Καλαθοσφαίρισης

Η καλαθοσφαίριση είναι ένα ομαδικό άθλημα στο οποίο δύο ομάδες αποτελούμενες από πέντε παίκτες η καθεμία, ανταγωνίζονται σε ένα γήπεδο με κύριο στόχο να βάλουν μια μπάλα μέσα από το στεφάνι που βρίσκεται τοποθετημένο σε ύψος 3 μέτρων και με ένα ταμπλό σε κάθε άκρο του γηπέδου. Η ομάδα με τους περισσότερους πόντους στο τέλος του παιχνιδιού κερδίζει και εφόσον στην κανονική διάρκεια έχουμε ισοπαλία, απαιτείται επιπλέον περίοδος παιχνιδιού. Στην καλαθοσφαίριση, η συλλογή μεγάλων δεδομένων και οι συνεπαγόμενες αναλυτικές γνώσεις που παράγονται έχει γίνει ιδιαίτερα σημαντική.

3.2.1 National Basketball Association (NBA)

Το National Basketball Association (NBA) είναι ένα επαγγελματικό πρωτάθλημα μπάσκετ στη Βόρεια Αμερική. Το πρωτάθλημα αποτελείται από 30 ομάδες και αποτελεί το πιο κορυφαίο πρωτάθλημα επαγγελματικού μπάσκετ ανδρών στον κόσμο. [24] Η σεζόν διαρκεί από τον Οκτώβριο έως τον Απρίλιο, με μια κανονική περίοδο 82 αγώνων που την καθιστά ένα από τα πιο απαιτητικά επαγγελματικά αθλήματα. Οι 30 ομάδες είναι χωρισμένες σε δύο conference 15 ομάδων (Ανατολή και Δύση) και κάθε conference αποτελείται από τρία division 5 ομάδων, όπως φαίνεται στην εικόνα 13. Η τελική φάση του πρωταθλήματος ονομάζεται playoffs όπου κάθε ομάδα αγωνίζεται με τον ίδιο αντίπαλο μέχρι 7 φορές, με νικητή της σειράς την ομάδα που έχει κάνει 4 νίκες. Η πρώτη ομάδα σε κάθε division λαμβάνει αυτόματα μια θέση από τις 16 των playoffs. Οι υπόλοιπες 8 θέσεις

αποφασίζονται από τα 4 επόμενα υψηλότερα ρεκόρ ομάδων. Έτσι οι 16 από τις 30 ομάδες μπαίνουν στα playoffs, 8 ομάδες από τη Δύση και 8 ομάδες από την Ανατολή. Οι τελικοί δεν είναι τίποτα άλλο από μια κανονική σειρά 7 αγώνων μεταξύ μιας ομάδας Δύσης και μιας ομάδας Ανατολής.

Οι ομάδες του conference Ανατολής είναι οι εξής:

Atlantic Division: Boston Celtics, Philadelphia 76ers, Toronto Raptors, Brooklyn Nets, New York Knicks

Central Division: Milwaukee Bucks, Chicago Bulls, Cleveland Cavaliers, Indiana Pacers, Detroit Pistons

Southeast Division:

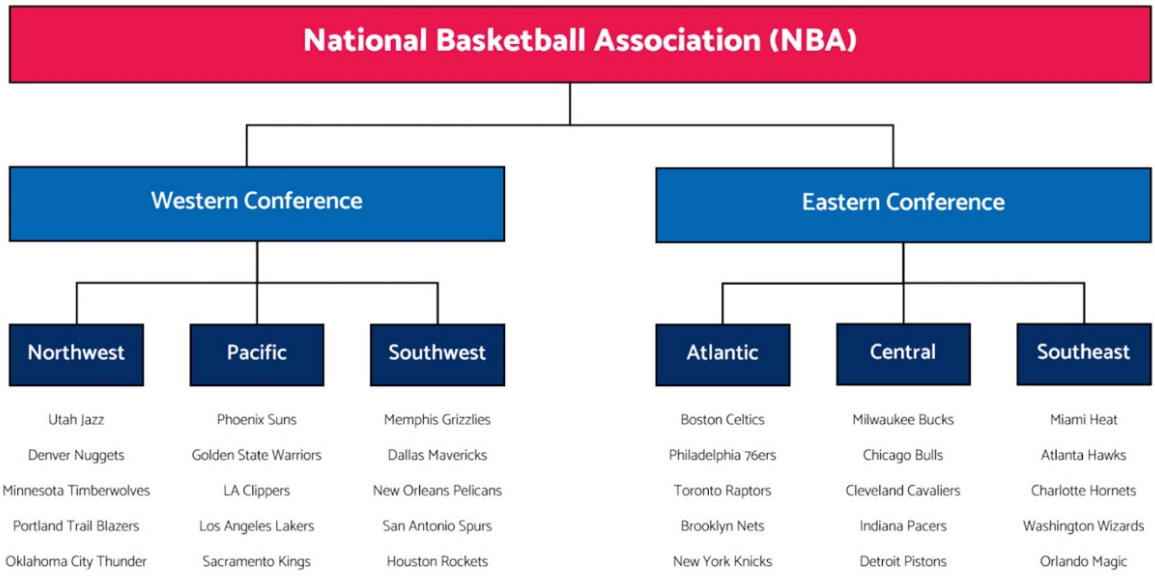
Atlanta Hawks, Charlotte Hornets, Miami Heat, Orlando Magic, Washington Wizards.

Οι ομάδες του conference της Δύσης είναι οι εξής:

Northwest Division: Utah Jazz, Denver Nuggets, Minnesota Timberwolves, Portland Trail Blazers.

Pacific Division: Phoenix Suns, Golden State Warriors, Los Angeles Clippers, Los Angeles Lakers, Sacramento Kings.

Southwest Division: Dallas Mavericks, Houston Rockets, Memphis Grizzlies, New Orleans Pelicans, San Antonio Spurs.



Εικόνα 18: Επισκόπηση Πρωταθλήματος NBA

Οι ομάδες του NBA έχουν αρχίσει να τοποθετούν κάμερες παρακολούθησης δεδομένων σε κάθε γωνία του γηπέδου, προσφέροντας τη δυνατότητα να παρακολουθούν κάθε κίνηση που κάνει ένας παίκτης στο γήπεδο και να τη συγχρονίζουν με τα ατομικά τους στατιστικά. Αν και η τοποθέτηση καμερών είναι ιδιαίτερα ακριβή τεχνολογία, αυτά τα δεδομένα είναι

εξαιρετικά ωφέλιμα για να αναλύσουν οι ομάδες την απόδοση του παιχνιδιού τους, προκειμένου σε δεύτερο χρόνο να βελτιώσουν την απόδοση τους.

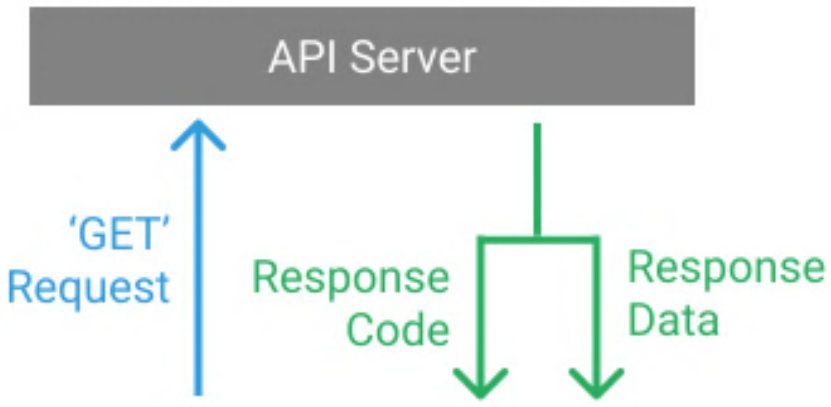
3.3 Πηγή Δεδομένων &Υλοποίηση Προγράμματος

Η ποιότητα των δεδομένων είναι πολύ σημαντική, συνεπώς είναι σημαντικό να έχουμε αξιόπιστη πηγή των δεδομένων. Για αυτό το λόγο επιλέχθηκε να συλλεχθούν τα δεδομένα από επίσημο ιστότοπο του NBA. Σύμφωνα με τους όρους της ιστοσελίδας η χρήση, η εμφάνιση και δημοσίευση των στατιστικών επιτρέπεται για μη εμπορικούς σκοπούς, αρκεί να συνοδεύεται από εμφανή αναφορά στο NBA.com.

Τα παραδοσιακά στατιστικά στοιχεία

Για τις ανάγκες της συλλογής των δεδομένων, στο πλαίσιο της παρούσας εργασίας γράφτηκε το πρόγραμμα ονόματι GetPlaybyPlayData σε γλώσσα προγραμματισμού Python. Η συλλογή δεδομένων από ιστότοπους χρησιμοποιώντας μια αυτοματοποιημένη διαδικασία είναι γνωστή ως scraping.

Τα δεδομένα στις ιστοσελίδες δεν είναι δομημένα, συνεπώς είναι επιβεβλημένη η συγγραφή κώδικα. Όταν εκτελείται κώδικας για scraping ιστού, αποστέλλεται ένα αίτημα στη διεύθυνση url, αναμένοντας απάντηση από τον διακομιστή που στέλνει τα δεδομένα και επιτρέπει την ανάγνωση της html ιστοσελίδας.



Εικόνα 19: Διάγραμμα Ακολουθίας Requests

Συγκεκριμένα έγινε χρήση της βιβλιοθήκης requests για την πραγματοποίηση http αιτημάτων από την Python. Για την εφαρμογή της βιβλιοθήκης είναι απαραίτητος ο ορισμός του «headers» που παρέχει πληροφορίες σχετικά με το πλαίσιο αιτήματος, έτσι ώστε ο διακομιστής να μπορεί να προσαρμόσει την απάντηση. Επιγραμματικά επιλέχθηκε

η κλήση της μορφής JSON (JavaScript Object Notation) και η αναφορά στο url stats.nba.com.

Μια από τις σημαντικότερες βιβλιοθήκες για τη συλλογή των δεδομένων NBA με την Python ονομάζεται nba_api. Απλοποιεί σημαντικά τη διαδικασία λήψης δεδομένων από τον επίσημο ιστότοπο του NBA αφού διαχειρίζεται ιδανικά το μεγάλο αριθμό συνόλων δεδομένων που συλλέγονται. Με την μέθοδο leaguegamefinder.LeagueGameFinder δίνεται η δυνατότητα της συλλογής των κωδικών παιχνιδιών ανά χρονιά τόσο στην κανονική περίοδο όσο και στα Playoffs του NBA. Οι συγκεκριμένοι κωδικοί είναι προ απαιτούμενοι για την συγκεκριμένη JSON κλήση με την requests στο ιστότοπο του NBA και ποιο συγκεκριμένα στο https://cdn.nba.com/static/json/liveData/playbyplay/ playbyplay [.....] ακολουθούμενο από τον κωδικό παιχνιδιού της JSON κλήσης.

Για την προσωρινή αποθήκευση χρησιμοποιήθηκε το πακέτο pandas που παρέχει γρήγορες και ευέλικτες δομές δεδομένων που έχουν σχεδιαστεί για την αλληλεπίδραση σχεσιακών δεδομένων. Η μονοδιάστατη δομή δεδομένων του πακέτου panda χρησιμοποιείται στην συντριπτική πλειοψηφία στα οικονομικά, τη στατιστική και σε πολλούς τομείς της μηχανικής. Επιπρόσθετα, το εν λόγω πακέτο είναι χτισμένο πάνω στο NumPy που ενσωματώνεται πολύ καλά σε οποιοδήποτε επιστημονικό υπολογιστικό περιβάλλον με πολλές άλλες βιβλιοθήκες τρίτων.

Τέλος, για την εξαγωγή των δεδομένων σε αρχεία έγινε η χρήση της βιβλιοθήκης xlwt που βοήθησε στην δημιουργία αρχείων υπολογιστικών φύλλων συμβατά με τις εκδόσεις του Microsoft Excel, σε συνδυασμό με τις μεθόδους του πακέτου pandas, to_csv και to_excel. Ενδεικτικό στιγμιότυπο του κώδικα Python παρατίθεται στο παράρτημα Α.

3.4 Περιγραφή, Ανάλυση, Σχεδιασμός & Μορφοποίηση Δεδομένων

Τα παραδοσιακά στατιστικά στοιχεία box-score δείχνουν συγκεντρωμένα στοιχεία ανά παιχνίδι τόσο για την ομάδα όσο και για τους παίχτες. Η εν λόγω η πληροφορία κρύβει την ιστορικότητα που διαμορφώθηκε η τελική μορφή των στατιστικών, με αποτέλεσμα να υπάρχουν ορισμένες περιπτώσεις που είναι παραπλανητική. Τα δεδομένα που συλλέχθηκαν είναι τα play-by-play από δυο ολόκληρες χρονιές του πρωταθλήματος NBA 2018 – 2020 που αφορούν 2.455 διαφορετικούς αγώνες μεταξύ 30 διαφορετικών ομάδων με στόχο μια δυναμική έκδοση των box-score στατικών στοιχείων.

3.4.1 **Αναλυτική Περιγραφή Δεδομένων | Play-by-Play**

Τα play-by-play δεδομένα αποτελούν σαφώς την πιο αναλυτική στατιστική πληροφορία μιας αναμέτρησης. Το χαρακτηριστικό τους είναι ότι καταγράφουν κάθε γεγονός – στιγμιότυπο ανά αγώνα με ένα πλήθος άνω των 60 χαρακτηριστικών. Είναι αξιοσημείωτο ότι για κάθε αγώνα καταγράφονται κατά μέσο όρο περίπου 450 στιγμιότυπα – γεγονότα που αποτελούν και την λεπτομερής εξέλιξη του αγώνα. Το μεγάλο πλήθος της πληροφορίας επιτρέπει εύκολα την διαμόρφωση του κατάλληλου κουρδίσματος ανάλογα με τις ανάγκες, μετατρέποντας τα δεδομένα από μικρό σε μεγάλο βάθος και αντιστρόφως.

Αναλυτικότερα περιγράφεται η διαθέσιμη και αποθηκευμένη πληροφορία σε μορφή πινάκων ανά χαρακτηριστικό για τις εν λόγω αναμετρήσεις:

Action Number: Αποτελεί τον αύξων αριθμό και μοναδικό κλειδί για κάθε γεγονός της αναμέτρησης

Action Type: Αναφέρεται το είδος με βάση το γεγονός που συνέβη στην αναμέτρηση.

Assist: Αναφέρεται το όνομα του παίχτη που πραγματοποίησε μια assist μια δεδομένη χρονική στιγμή.

Away Score: Αναφέρεται στο -ανά χρονικό σημείο της αναμέτρησης- ακέραιο πλήθος των πόντων της φιλοξενούμενης ομάδας.

Block: Αναφέρεται το όνομα του παίχτη που πραγματοποίησε ένα block μια δεδομένη χρονική στιγμή.

Data Set: Διακρίνεται η ακριβή χρονιά που διεξάγεται η αναμέτρηση, καθώς και φάση του πρωταθλήματος όπως regular season ή playoffs.

Elapsed Time: Αποτελεί τον χρόνο που παρήλθε από τον συνολικό χρόνο της περιόδου (12 λεπτά), ανά γεγονός.

Game Date: Αποτελεί την ακριβή ημερομηνία διεξαγωγής της αναμέτρησης.

Game ID: Αναπαριστά έναν αριθμό που αποτελεί το μοναδικό κλειδί αναγνώρισης της εκάστοτε αναμέτρησης.

Home Score: Αναφέρεται στο -ανά χρονικό σημείο της αναμέτρησης- ακέραιο πλήθος των πόντων της γηπεδούχου ομάδας.

Line UP: Αναγράφονται τα ονόματα των παιχτών που συμμετέχουν στην αναμέτρηση την δεδομένη χρονική στιγμή. Υπάρχουν συνολικά 10 αθλητές με την κατανομή να είναι 5 αθλητές από την κάθε ομάδα.

Period Type: Η συγκεκριμένη τιμή αποτελεί ένα flag που υποδεικνύει αν η συγκεκριμένη αναμέτρηση βρίσκεται στην φάση της κανονικής διάρκειας του παιχνιδιού (regular) ή σε φάση παράτασης (overtime)

Period: Αναφέρεται ο αύξων αριθμός της περιόδου κάθε αγώνα. Οι συχνότερες ακέραιες τιμές που μπορεί να πάρει το συγκεκριμένο χαρακτηριστικό είναι μεταξύ των 1-4, αλλά υπάρχει και το ενδεχόμενο της παράτασης που οδηγεί σε ακέραιες τιμές ≥ 5 ανάλογα με το πλήθος των επιπλέον περιόδων.

Player: Αναφέρεται το όνομα του αθλητή που ενεπλάκη στο συγκεκριμένο γεγονός.

Points: Το πλήθος των ωφέλιμων πόντων από το συγκεκριμένο γεγονός

Shoot Distance: Στην περίπτωση του σουτ, αποτελεί την απόσταση που πραγματοποιήθηκε αυτό.

Short Formatted Clock: Αποτελεί τον απομένοντα χρόνο από τον συνολικό χρόνο της περιόδου (12 λεπτά), ανά γεγονός.

Shot Result: Στην περίπτωση του σουτ, αποτελεί την πληροφορία του αποτελέσματος, πετυχημένο ή χαμένο.

Steal: Αναφέρεται το όνομα του παίχτη που πραγματοποίησε ένα κλέψιμο (steal) μια δεδομένη χρονική στιγμή.

Team ID: Το μοναδικό κλειδί της ομάδας που αφορά άμεσα το γεγονός που αναφέρεται.

X (Legacy): Στην περίπτωση του σουτ, η συντεταγμένη X από το σημείο του σουτ.

Y (Legacy): Στην περίπτωση του σουτ, η συντεταγμένη Y από το σημείο του σουτ.

Description: Παρατίθεται μια τυποποιημένη αν και συμπτυγμένη περιγραφή του συγκεκριμένου στιγμιότυπου.

Οι ενδεχόμενες κατηγορίες ανά γεγονός της αναμέτρησης είναι οι εξής:

2pt: Ένα καλάθι δύο πόντων (δίποντο) προκύπτει από ένα σουτ που γίνεται πατώντας την γραμμή ή/και πραγματοποιείται μέσα από αυτήν των τριών πόντων.

3pt: Ένα καλάθι τριών πόντων (τρίποντο) προκύπτει από ένα σουτ που γίνεται πέρα από τη γραμμή των τριών πόντων, δηλαδή ένα καθορισμένο τόξο που περιβάλλει το καλάθι που στο NBA έχει απόσταση από το καλάθι, 7,24 μέτρα.

Block: Στην καλαθοσφαίριση ένα μπλοκ (block) συμβαίνει όταν ένας αμυντικός παίκτης εκτρέπει νόμιμα μια προσπάθεια σουτ από έναν επιθετικό παίκτη, αποτρέποντας το καλάθι.

Ejection: Η αποβολή (ejection) είναι η απομάκρυνση ενός συμμετέχοντα από την αναμέτρηση του αγώνα λόγω παραβίασης των κανόνων του αθλήματος.

Foul: Στο μπάσκετ, ένα φάουλ συμβαίνει ως αποτέλεσμα παράνομης προσωπικής επαφής με αντίπαλο ή/και αντιαθλητικής συμπεριφοράς. Τα φάουλ μπορεί να οδηγήσουν σε μία ή περισσότερες από τις ακόλουθες ποινές:

- Ο παίκτης με το φάουλ λαμβάνει μία ή περισσότερες ελεύθερες βολές.
- Η ομάδα της οποίας ο παίκτης έκανε το φάουλ χάνει την κατοχή της μπάλας από την άλλη ομάδα.
- Ο παίκτης που διαπράττει το φάουλ αποβάλλεται μετά από το όριο των έξι.

Free Throw: Στο μπάσκετ, οι ελεύθερες βολές (Free Throw) είναι προσπάθειες χωρίς αντίπαλο, με σουτ πίσω από τη γραμμή των ελεύθερων βολών (αξία ενός πόντου), στην κορυφή της περιοχής της ρακέτας. Οι ελεύθερες βολές χορηγούνται γενικά μετά από ένα φάουλ στον σουτέρ από την αντίπαλη ομάδα.

Game: Χρησιμοποιείται για να υποδηλώσει την λήξη του παιχνιδιού και πάντα σε συνδυασμό με την υποκατηγορία ονόματι end.

Jump Ball: Το jump ball είναι μια μέθοδος που χρησιμοποιείται για την έναρξη ή την επανέναρξη του παιχνιδιού στο μπάσκετ. Δύο αντίπαλοι παίκτες προσπαθούν να αποκτήσουν τον έλεγχο της μπάλας αφού ένας διαιτητής την πετάει στον αέρα μεταξύ τους. Επίσης μπορεί να επαναληφθεί σε περίπτωση αμφισβητούμενης κατοχής μεταξύ των αντίπαλων παιχτών.

Rebound: Στη καλαθοσφαίριση, ένα ριμπάουντ (rebound) είναι ένα στατιστικό στοιχείο που απονέμεται σε έναν παίκτη που ανακτά την μπάλα μετά από ένα χαμένο καλάθι εντός του πεδίου ή στην διαδικασία των ελεύθερων βολών.

Steal: Ένα κλέψιμο (steal) συμβαίνει όταν ένας αμυντικός παίκτης αποχτά την κατοχή της μπάλας χωρίς να αγγίζει τα χέρια του επιθετικού παίκτη, διαφορετικά γίνεται φάουλ.

Substitution: Αλλαγή στην καλαθοσφαίριση μπορεί να γίνει απεριόριστες φορές και πάντα στο πλαίσιο μιας διακοπής του χρονομέτρου.

Time Out: Ένα τάιμ άουτ (time out) είναι μια διακοπή της αναμέτρησης, που επιτρέπει στους προπονητές κάθε ομάδας να επικοινωνούν με τους αθλητές ώστε να καθορίσουν τη στρατηγική ή να εμπνεύσουν ηθικό. Οι ομάδες επιτρέπονται να καλούν μέχρι και επτά τάιμ άουτ, διάρκειας το καθένα από 1 λεπτό και 15 δευτερόλεπτα. Σε περιόδους παράτασης, σε κάθε ομάδα επιτρέπονται δύο επιπλέον τάιμ άουτ. Μπορεί να ζητηθεί είτε από έναν παίκτη, είτε απο τον προπονητή.

Turnover: Μια ανατροπή (turnover) συμβαίνει όταν μια ομάδα χάνει την κατοχή της μπάλας από την αντίπαλη ομάδα πριν ένας παίκτης σουτάρει στο καλάθι της ομάδας της. Αυτό μπορεί να προκύψει από έναν παίκτη που μπορεί να κλέψει την μπάλα, να βγει εκτός ορίων, να διαπράξει παράβαση ή επιθετικό φάουλ.

Violation: Οι περισσότερες παραβιάσεις (violation) γίνονται από την ομάδα με την κατοχή της μπάλας, όταν ένας παίκτης δεν χειρίζεται σωστά την μπάλα. Η τυπική ποινή για μια παράβαση είναι η απώλεια της μπάλας στην άλλη ομάδα.

Για κάθε κατηγορία υπάρχει μια υποκατηγορία ανά γεγονός της αναμέτρησης, με το σύνολο αυτών να φτάνει περίπου στις 50. Αναλυτικά οι όροι και η επεξήγηση των υποκατηγοριών αναφέρονται στο παράρτημα Β.

3.4.2 **Σχεδιασμός Διαμόρφωσης Δεδομένων**

Το Box-Score αποτελεί το γνωστότερο των στατιστικών, αφού είναι μια δομημένη περίληψη των αποτελεσμάτων από έναν αθλητικό αγώνα. Όπως φαίνεται και στο

παράδειγμα της εικόνας 20, παραθέτει τη βαθμολογία καθώς και τα ατομικά / ομαδικά επιτεύγματα της αναμέτρησης. Συνήθως, χρησιμοποιείται για να βοηθήσει στον προσδιορισμό της επιτυχίας μιας ομάδας, σε συσχέτιση με έναν παίκτη, αποκτώντας μια γενική ιδέα για το πώς διεξήχθη η αναμέτρηση ή/και πώς απέδωσε ο παίκτης κατά τη διάρκεια του παιχνιδιού. [25] Σε κάθε αναμέτρηση υπάρχει η δυνατότητα ανάκτησης ενός box-score, όμως η πληροφορία κρύβει την ιστορικότητα των στατιστικών που διαμόρφωσαν την τελική εικόνα των εν λόγω στατιστικών.

Dallas Mavericks

STARTERS	MIN	FG	3PT	FT	OREB	DREB	REB	AST	STL	BLK	TO	PF	+/-	PTS
D. Finney-S... PF	38	7-11	4-7	1-2	0	4	4	3	2	1	2	1	+3	19
D. Powell C	23	1-3	0-1	2-2	1	2	3	2	0	1	2	3	-9	4
S. Dinwiddie PG	37	5-14	3-6	5-9	0	5	5	4	1	0	2	1	+13	18
J. Brunson PG	37	6-12	2-4	0-0	1	3	4	2	0	0	3	3	+8	14
L. Doncic PG	39	8-16	4-10	6-6	0	8	8	8	3	0	5	0	+7	26
BENCH	**MIN**	**FG**	**3PT**	**FT**	**OREB**	**DREB**	**REB**	**AST**	**STL**	**BLK**	**TO**	**PF**	**+/-**	**PTS**
M. Kleber PF	29	0-2	0-1	0-0	1	12	13	0	0	3	1	3	+7	0
D. Bertans SF	6	1-3	1-3	0-0	0	3	3	0	1	0	0	0	+5	3
S. Brown SF	4	0-2	0-1	0-0	0	1	1	0	0	0	0	0	-4	0
T. Burke PG	11	1-4	0-1	0-0	0	0	0	1	0	0	0	0	-5	2
J. Green SG	17	4-7	1-3	0-0	1	0	1	0	0	0	2	5	-10	9
B. Marjanovic C	DNP-COACH'S DECISION													
F. Ntilikina PG	DNP-COACH'S DECISION													
TEAM		**33-74**	**15-37**	**14-19**	**4**	**38**	**42**	**20**	**7**	**5**	**17**	**16**		**95**
		44.6%	**40.5%**	**73.7%**										

Εικόνα 20: Παράδειγμα Box-Score | NBA (πηγή: **[26]**)

Στο πλαίσιο της παρούσας εργασίας, αναζητήθηκε ένας τρόπος συγκράτησης της ιστορικότητας, ώστε να εκπαιδεύσουμε το μοντέλο μας με τον τρόπο που πραγματοποιείται μια νίκη, ανεξάρτητα της ταυτότητας των ομάδων. Η συγκράτηση ενός δυναμικού box-score ανά τακτά χρονικά διαστήματα θεωρείται η ιδανική πληροφορία εισόδου, αφού τροφοδοτώντας σταδιακά ως συσσωμάτωση τα Δίκτυα Μακράς Βραχύχρονης Μνήμης – LSTM, μπορούμε να καταλήξουμε σε υψηλό ποσοστά επιτυχών προβλέψεων. Τα δεδομένα play-by-play παρέχουν ολοκληρωμένα την αναλυτική ροή της πληροφορία των γεγονότων στον χρόνο, που συντέλεσαν στα τελικά στατιστικά τόσο των ομάδων, όσο και των παιχτών. Συνεπώς, παρατηρούμε ότι αποκτώντας τα δεδομένα play-by-play και με την κατάλληλη επεξεργασία μπορούμε να καταλήξουμε στην βαθύτερη δυνατή λεπτομέρεια σε επίπεδο πτυχής γεγονότων από οποιοδήποτε άλλο αρχείο συγκεντρωτικών στατιστικών. Μια λεπτομέρεια που εξάγεται κατά την διάρκεια του παιχνιδιού δημιουργώντας μια ροή πληροφορίας ιδανική για τα LSTM που έχουν την ικανότητα να κρατούν μόνο επιλεγμένες και ωφέλιμες πληροφορίες στη βραχύχρονη μνήμη τους.

Αναλυτικά, με βάση τα εν λόγω δεδομένα σε κάθε ξεχωριστό γεγονός είναι δυνατό εξαχθεί η εξής πληροφορία:

Game ID: Αποτελεί το μοναδικό κλειδί της αναμέτρησης

Away / Home Team: Η φιλοξενούμενη / γηπεδούχος Ομάδα

FGM: Είναι το πλήθος των επιτυχημένων καλαθιών που πραγματοποιήθηκαν στο πεδίο από έναν αθλητή ή αθροιστικά από την ομάδα, ανεξαρτήτως της αξίας των πόντων.

FGA: Είναι το πλήθος των προσπαθειών για καλάθι που πραγματοποιήθηκαν στο πεδίο από έναν αθλητή ή αθροιστικά από την ομάδα, ανεξαρτήτως της αξίας των πόντων.

FG%: Το ποσοστό των επιτυχημένων καλαθιών προς τις προσπάθειες που πραγματοποιήθηκαν στο πεδίο από έναν αθλητή ή αθροιστικά από την ομάδα, ανεξαρτήτως της αξίας των πόντων.

3PM: Το συνολικό πλήθος των επιτυχημένων καλαθιών τριών πόντων που επιχείρησε ένας αθλητής ή αθροιστικά η ομάδα.

3PA: Το συνολικό πλήθος των προσπαθειών για καλάθι τριών πόντων που επιχείρησε ένας αθλητής ή αθροιστικά η ομάδα.

3P%: Το ποσοστό του συνολικού πλήθους των επιτυχημένων καλαθιών προς τις προσπάθειες τριών πόντων που επιχείρησε ένας αθλητής ή αθροιστικά η ομάδα.

FTM: Αναφέρεται στο πλήθος των επιτυχημένων ελεύθερων βολών που έγιναν από έναν αθλητή ή αθροιστικά από μια ομάδα. Κάθε ελεύθερη βολή αξίζει έναν πόντο.

FTA: Αναφέρεται στο πλήθος των προσπαθειών για ελεύθερες βολές που έγιναν από έναν αθλητή ή αθροιστικά από μια ομάδα.

FT%: Το ποσοστό του πλήθους των επιτυχημένων ελεύθερων βολών προς τις προσπάθειες που έγιναν από έναν αθλητή ή αθροιστικά από μια ομάδα.

REB: Το πλήθος επιθετικών και αμυντικών ριμπάουντ που συλλέγονται από έναν παίκτη ή αθροιστικά από την ομάδα.

OREB Το πλήθος των ριμπάουντ που συλλέγονται από έναν παίκτη ή αθροιστικά μια ομάδα ενώ παίζει επιθετικά.

DREB Το πλήθος των ριμπάουντ που συλλέγει ένας παίκτης ή αθροιστικά μια ομάδα ενώ παίζει άμυνα.

AST: Το πλήθος των ασίστ που γίνονται από έναν παίκτη ή αθροιστικά από την ομάδα. Μια ασίστ καταγράφεται εφόσον μια πάσα οδηγήσει απευθείας στο καλάθι από τον συμπαίκτη που την δέχθηκε.

STL: Το πλήθος των κλεψιμάτων που έγιναν από έναν παίκτη ή αθροιστικά από την ομάδα. Κλέψιμο θεωρείται όταν ένας αμυντικός παίκτης αφαιρέσει την μπάλα από έναν επιθετικό παίκτη είτε παρεμποδίζοντας μια πάσα είτε κλέβοντας στην ντρίμπλα του επιθετικού παίκτη.

BLK: Το πλήθος των μπλοκ που έγιναν από έναν αμυντικό παίκτη ή αθροιστικά από την ομάδα.

TOV: Το πλήθος των ανατροπών που έγιναν από έναν παίκτη ή αθροιστικά από την ομάδα. Turnover έχουμε όταν ένας επιθετικός παίκτης πριν επιχειρήσει σουτ, χάνει την κατοχή της μπάλας στην άμυνα.

PF: Το πλήθος των προσωπικών φάουλ που έγιναν από έναν παίκτη ή αθροιστικά από την ομάδα. Ένα προσωπικό φάουλ συμβαίνει όταν ένας παίκτης έχει παράνομη προσωπική επαφή με έναν αντίπαλο.

Points: Το πλήθος των επιτυχημένων πόντων του αθλητή ή αθροιστικά για όλη την ομάδα.

3.5 Υλοποίηση Προγράμματος Διαμόρφωσης Δεδομένων

Στην φάση του σχεδιασμού αποφασίστηκε η κατάλληλη διαμόρφωση των play-by-play δεδομένων σε μια δυναμική μορφή box-score δεδομένων. Για τις ανάγκες της αναφερόμενης διαμόρφωσης έγινε εκ νέου χρήση της Python, μέσω της οποίας γράφτηκε κατάλληλος κώδικας για την δημιουργία προγράμματος.

Το νέο πρόγραμμα ονόματι PlayByPlay2dBoxScore θα παίρνει ως είσοδο το αρχείο που εξάγεται από το πρόγραμμα GetPlaybyPlayData που υλοποιήσαμε στο πλαίσιο της φάσης εύρεσης και αποθήκευσης των play-by-play δεδομένων του NBA. Στόχος του προγράμματος είναι μετατροπή του play-by-play αρχείου σε μια δυναμική μορφή box-score που να μπορεί να εξάγεται για κάθε αγώνα όσο συχνά ορίζει ο χρήστης. Το αρχείο θα εξάγεται ξεχωριστά για κάθε αγώνα και θα αποτελείται από εγγραφές τόσων box-scores

ανάλογα του time resolution. Όπως αναφέρθηκε πρωτύτερα στο ίδιο κεφάλαιο, το πλεονέκτημα των play-by-play δεδομένων είναι ότι έχοντας την μέγιστη δυνατή πληροφορία μπορούμε να «κουρδίζουμε» από το ειδικότερο στο γενικότερο. Συνεπώς, το πρόγραμμα σχεδιάστηκε ώστε να διαμορφώνει τα δεδομένα ανάλογα με την ανάλυση ή time resolution που εισάγει ο χρήστης. Η μέθοδος που γράφτηκε για το time resolution δημιουργεί 200 εγγραφές box-score εφόσον δοθεί time resolution 0,005 (1/0,005), 100 εγγραφές box-score εφόσον δοθεί time resolution 0,01 (1/0,01) και 10 εγγραφές box-score εφόσον δοθεί time resolution 0,1 (1/0,1). Είναι σημαντικό να αναφερθεί ότι επιλέχθηκε, τα δυναμικά box-scores που δημιουργούνται ανά εγγραφή να μη περιέχουν συσσωρευτικά την πληροφορία των στατιστικών, αλλά να είναι ανεξάρτητα.

Οι εξαγόμενες εγγραφές δυναμικών box-score περιέχουν τις πληροφορίες που αναφέρθηκαν κατά τον σχεδιασμό. Στο πίνακα 1, παρατίθενται η λίστα των στηλών που εξάγονται για κάθε παιχνίδι. Οι στήλες με το χαρακτηριστικό «_H» αφορούν την γηπεδούχο ομάδα και με το χαρακτηριστικό «_R» αφορούν την φιλοξενούμενη.

Επιπλέον των αρχείων εξάγεται και πληροφορία με την τελική έκβαση του αγώνα ανά game_id για την χρήση του στον supervised μοντέλο του αλγορίθμου.

game_id	team_H	team_R	fg_H	fg_R	fga_H	fga_R	3p_H
3p_R	ft_H	ft_R	fta_H	fta_R	orb_H	orb_R	drb_H
drb_R	trb_H	trb_R	ast_H	ast_R	stl_H	stl_R	blk_H
blk_R	tov_H	tov_R	pf_H	pf_R	pts_H	pts_R	

Πίνακας 1: Λίστα Χαρακτηριστικών Dynamic Box-Score

Αναλυτικά, τα χαρακτηριστικά περιέχουν την πληροφορία ως εξής:

Game ID: Αποτελεί το μοναδικό κλειδί της αναμέτρησης

team_H: Η γηπεδούχος Ομάδα

Team_R Η φιλοξενούμενη ομάδα

FG_H: Είναι το πλήθος των επιτυχημένων καλαθιών που πραγματοποιήθηκαν στο πεδίο αθροιστικά από την γηπεδούχο ομάδα, ανεξαρτήτως της αξίας των πόντων.

FG_R: Είναι το πλήθος των επιτυχημένων καλαθιών που πραγματοποιήθηκαν στο πεδίο αθροιστικά από την φιλοξενούμενη ομάδα, ανεξαρτήτως της αξίας των πόντων.

FGA_H: Είναι το πλήθος των προσπαθειών για καλάθι που πραγματοποιήθηκαν στο πεδίο αθροιστικά από την γηπεδούχο ομάδα, ανεξαρτήτως της αξίας των πόντων.

FGA_R: Είναι το πλήθος των προσπαθειών για καλάθι που πραγματοποιήθηκαν στο πεδίο αθροιστικά από την φιλοξενούμενη ομάδα, ανεξαρτήτως της αξίας των πόντων.

3p_H: Το συνολικό πλήθος των επιτυχημένων καλαθιών τριών πόντων που επιχείρησε αθροιστικά η γηπεδούχος ομάδα.

3p_R: Το συνολικό πλήθος των επιτυχημένων καλαθιών τριών πόντων που επιχείρησε αθροιστικά η φιλοξενούμενη ομάδα.

ft_H: Αναφέρεται στο πλήθος των επιτυχημένων ελεύθερων βολών που έγιναν αθροιστικά από την γηπεδούχο ομάδα. Κάθε ελεύθερη βολή αξίζει έναν πόντο.

ft_R: Αναφέρεται στο πλήθος των επιτυχημένων ελεύθερων βολών που έγιναν αθροιστικά από την φιλοξενούμενη ομάδα. Κάθε ελεύθερη βολή αξίζει έναν πόντο.

fta_H: Αναφέρεται στο πλήθος των προσπαθειών για ελεύθερες βολές που έγιναν αθροιστικά από την γηπεδούχο ομάδα.

fta_R: Αναφέρεται στο πλήθος των προσπαθειών για ελεύθερες βολές που έγιναν αθροιστικά από την φιλοξενούμενη ομάδα.

orb_H: Το πλήθος επιθετικών ριμπάουντ που συλλέγονται αθροιστικά από την γηπεδούχο ομάδα.

orb_R: Το πλήθος των επιθετικών ριμπάουντ που συλλέγονται αθροιστικά από την φιλοξενούμενη ομάδα.

drb_H: Το πλήθος των αμυντικών ριμπάουντ που συλλέγονται αθροιστικά η γηπεδούχος ομάδα ενώ παίζει άμυνα.

drb_R: Το πλήθος των αμυντικών ριμπάουντ που συλλέγονται αθροιστικά η φιλοξενούμενη ομάδα ενώ παίζει άμυνα.

ast_H: Το πλήθος των ασίστ που γίνονται αθροιστικά από την γηπεδούχο ομάδα. Μια ασίστ καταγράφεται εφόσον μια πάσα οδηγήσει απευθείας στο καλάθι από τον συμπαίκτη που την δέχθηκε.

ast_R: Το πλήθος των ασίστ που γίνονται αθροιστικά από την φιλοξενούμενη ομάδα. Μια ασίστ καταγράφεται εφόσον μια πάσα

οδηγήσει απευθείας στο καλάθι από τον συμπαίκτη που την δέχθηκε.

stl_H: Το πλήθος των κλεψιμάτων που έγιναν αθροιστικά από την γηπεδούχο ομάδα. Κλέψιμο θεωρείται όταν ένας αμυντικός παίκτης αφαιρέσει την μπάλα από έναν επιθετικό παίκτη είτε παρεμποδίζοντας μια πάσα είτε κλέβοντας στην ντρίμπλα του επιθετικού παίκτη.

stl_R: Το πλήθος των κλεψιμάτων που έγιναν αθροιστικά από την φιλοξενούμενη ομάδα. Κλέψιμο θεωρείται όταν ένας αμυντικός παίκτης αφαιρέσει την μπάλα από έναν επιθετικό παίκτη είτε παρεμποδίζοντας μια πάσα είτε κλέβοντας στην ντρίμπλα του επιθετικού παίκτη.

blk_H: Το πλήθος των μπλοκ που έγιναν αθροιστικά από την γηπεδούχο ομάδα.

blk_R: Το πλήθος των μπλοκ που έγιναν αθροιστικά από την φιλοξενούμενη ομάδα.

tov_H: Το πλήθος των ανατροπών που έγιναν αθροιστικά από την γηπεδούχο ομάδα. Turnover έχουμε όταν ένας επιθετικός παίκτης πριν επιχειρήσει σουτ, χάνει την κατοχή της μπάλας στην άμυνα.

tov_R: Το πλήθος των ανατροπών που έγιναν αθροιστικά από την φιλοξενούμενη ομάδα. Turnover έχουμε όταν ένας επιθετικός παίκτης πριν επιχειρήσει σουτ, χάνει την κατοχή της μπάλας στην άμυνα.

pf_H: Το πλήθος των προσωπικών φάουλ που έγιναν αθροιστικά από την γηπεδούχο ομάδα. Ένα προσωπικό φάουλ συμβαίνει όταν ένας παίκτης έχει παράνομη προσωπική επαφή με έναν αντίπαλο.

pf_R: Το πλήθος των προσωπικών φάουλ που έγιναν αθροιστικά από την φιλοξενούμενη ομάδα. Ένα προσωπικό φάουλ συμβαίνει όταν ένας παίκτης έχει παράνομη προσωπική επαφή με έναν αντίπαλο.

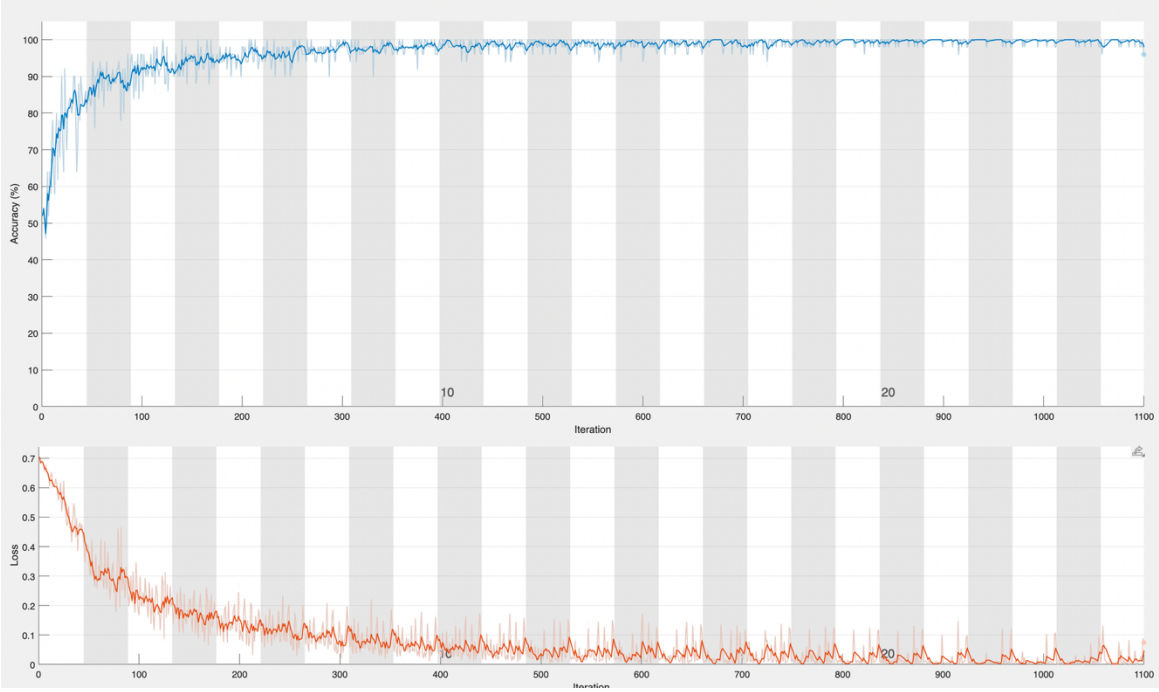
pts_H: Το πλήθος των επιτυχημένων πόντων αθροιστικά για όλη την γηπεδούχο ομάδα.

pts_R: Το πλήθος των επιτυχημένων πόντων αθροιστικά για όλη την φιλοξενούμενη ομάδα.

4 Μοντελοποίηση & Αρχιτεκτονική Αγώνων

4.1 MATrix LABoratory

Στην παρούσα φάση της μοντελοποίησης και μετά την κατάλληλη διαμόρφωση των play-by-play δεδομένων σε μια δυναμική μορφή box-score δεδομένων, γράφτηκε κατάλληλος κώδικας σε MATLAB (MATrix LABoratory) [27]. Το MATLAB είναι μια γλώσσα προγραμματισμού πολλαπλών-παραδειγμάτων σε ένα περιβάλλον αριθμητικού υπολογισμού που αναπτύχθηκε από τη MathWorks. Με το εν λόγω λογισμικό Επιτρέπει σχεδίαση συναρτήσεων και δεδομένων, χειρισμούς μήτρας, δημιουργία διεπαφών χρήστη και βέβαια την εύκολη υλοποίηση γνωστών αλγορίθμων [28]. Πιο συγκεκριμένα έγινε χρήση του Bioinformatics Toolbox που παρέχει αλγόριθμους και εφαρμογές για ανάλυση μικροσυστοιχιών, φασματομετρία μάζας και γονιδιακή οντολογία. Η εργαλειοθήκη παρέχει επίσης στατιστικές τεχνικές για την ανίχνευση κορυφών, τον καταλογισμό τιμών για δεδομένα που λείπουν και την επιλογή χαρακτηριστικών. Στην παρούσα φάση η χρήση της εν λόγω εργαλειοθήκης περιορίστηκε στις λειτουργίες της οπτικοποίησης δεδομένων, των γραφημάτων συμπλέγματος και της k-fold cross-validation. Επιπρόσθετα χρησιμοποιήθηκε το Deep Learning Toolbox που παρέχει λειτουργίες και εφαρμογές για την περιγραφή, την ανάλυση και τη μοντελοποίηση δεδομένων. Η εν λόγω βιβλιοθήκη εργαλειοθήκη παρέχει εποπτευόμενους και μη εποπτευόμενους αλγόριθμους μηχανικής μάθησης, συμπεριλαμβανομένων Support Vector Machines (SVM), ενισχυμένων δέντρων αποφάσεων, ρηχών νευρωνικών δικτύων, k-means και άλλων μεθόδων ομαδοποίησης. Επιπρόσθετα παρέχει ένα πλαίσιο για το σχεδιασμό και την υλοποίηση βαθιών νευρωνικών δικτύων με αλγόριθμους, προ-εκπαιδευμένα μοντέλα και εφαρμογές. Εξίσου σημαντικό είναι το block simulink που επιτρέπει τη χρηση των μοντέλων πρόβλεψης με προσομοιώσεις και σχεδιασμό βάσει μοντέλου. Για τις ανάγκες της εργασίας προτιμήθηκε η χρήση της εν λόγω εργαλειοθήκης για τα δίκτυα μακράς βραχυπρόθεσμης μνήμης (LSTM) που αποτελεί ένα τύπο επαναλαμβανόμενου νευρωνικού δικτύου. Αξίζει να σημειωθεί ότι το MATLAB δίνει την δυνατότητα οπτικοποίησης των επιπέδων και της παρακολούθησης της προόδου της εκπαίδευσης γραφικά ενώ παράλληλα παρακολουθούμε τις παραμέτρους εκπαίδευσης αναλύοντας αποτελέσματα συγκρίνοντας κώδικα από διαφορετικά πειράματα.



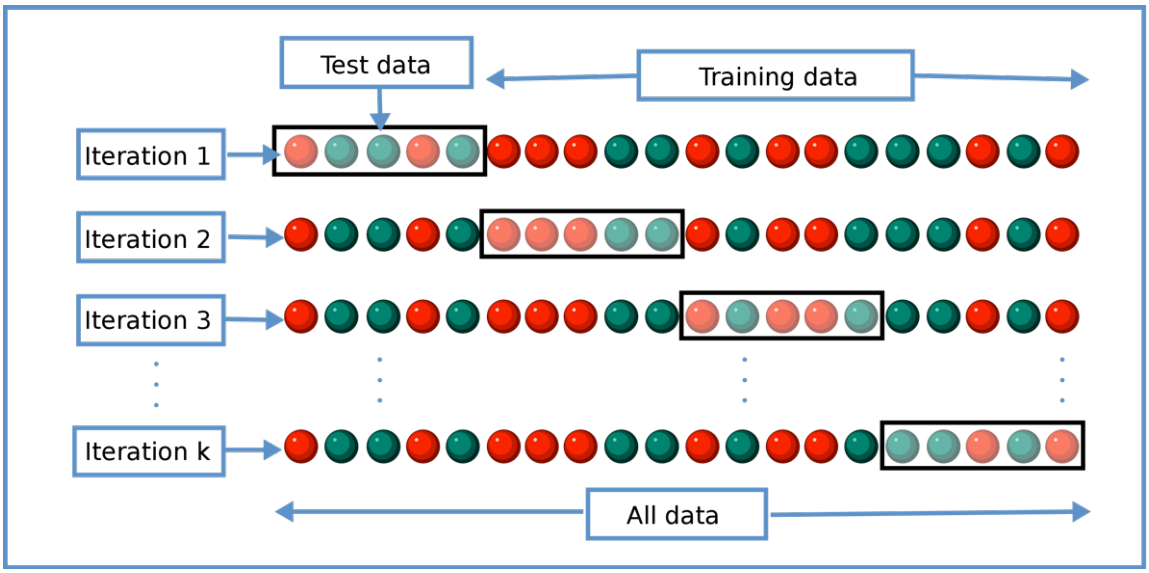
Εικόνα 21: Γραφική Απεικόνιση Εξέλιξης Εκπαίδευσης Δικτύου

4.2 Περιγραφή Μοντελοποίησης

Στο πλαίσιο της παρούσας εργασίας η μοντελοποίηση της έκβασης ενός αθλητικού αγώνα προκύπτει από τον τρόπο που διαμορφώνεται το κάθε στιγμιότυπο. Η διαμόρφωση πραγματοποιείται σταδιακά και με ρυθμό ανάλογο με το time resolution που έχει επιλεγεί. Το προτεινόμενο μοντέλο προσπαθεί να προβλέψει το αποτέλεσμα του αγώνα, βάση των διαφορών επιλεγμένων χαρακτηριστικών της γηπεδούχου και φιλοξενούμενης ομάδας και τον τρόπο εξέλιξης αυτών. Κάθε σύνολο χαρακτηριστικών που αναφέρονται στον πίνακα 1, αποτελεί μια πολυδιάστατη χρονοσειρά δεδομένων η οποία τροφοδοτείται βάση της αρχιτεκτονικής στα νευρωνικά δίκτυα LSTM. Το πλήθος των διαθέσιμων χρονοσειρών επιμερίζεται σε δεδομένα εκπαίδευσης και ελέγχου σύμφωνα με την μέθοδο της δεκαπλής διασταυρούμενης επικύρωσης, δηλαδή της ειδικής κατηγορίας της κ-διασταυρούμενης επικύρωσης k-fold cross-validation με k = 10.

Η διασταυρούμενη επικύρωση [29] (cross-validation) που μερικές φορές ονομάζεται εκτίμηση περιστροφής ή δοκιμή εκτός δείγματος, είναι μια εκ των τεχνικών επικύρωσης μοντέλων για την αξιολόγηση του τρόπου με τον οποίο τα αποτελέσματα μιας στατιστικής γενικεύονται σε ένα ανεξάρτητο σύνολο δεδομένων. Η μέθοδος αυτή χρησιμοποιεί διαφορετικά τμήματα των δεδομένων για να δοκιμάσει και να εκπαιδεύσει ένα μοντέλο σε διαφορετικές επαναλήψεις. Ειδικότερα στην k-fold cross-validation, το αρχικό δείγμα κατανέμεται τυχαία σε k ίσου μεγέθους υποδείγματα. Από τα k υποδείγματα, ένα

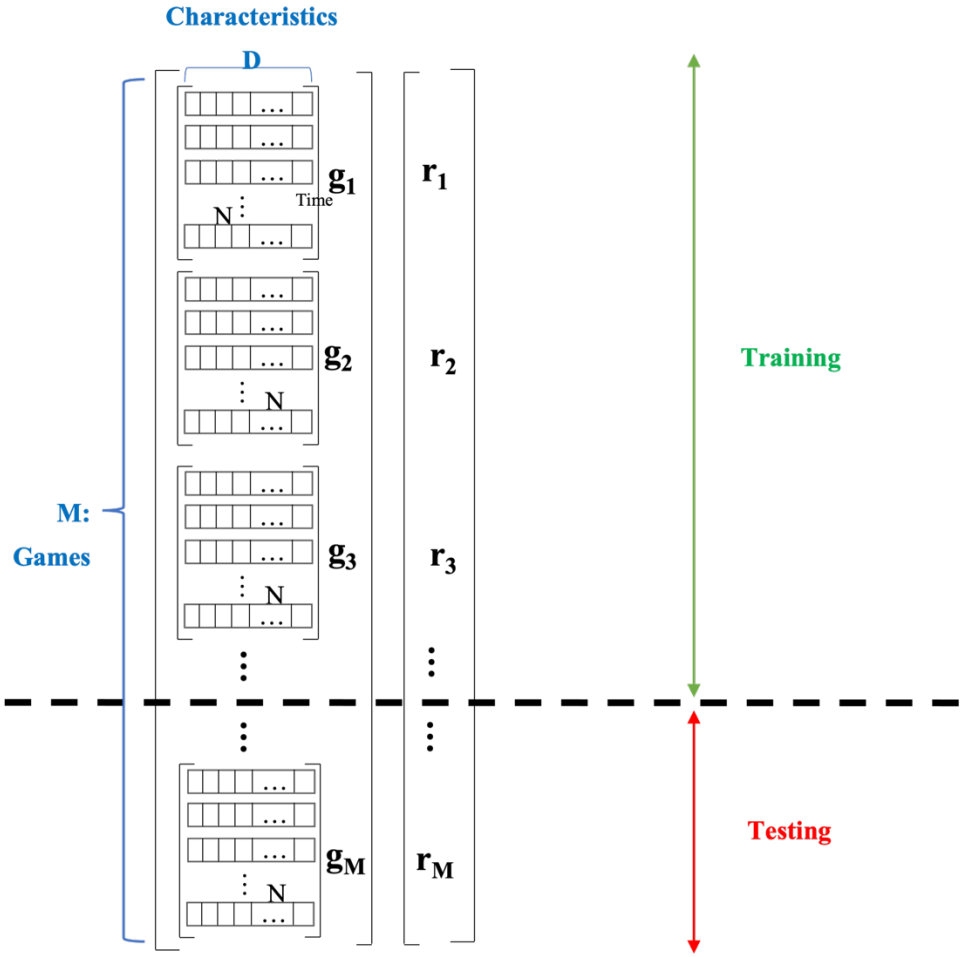
μεμονωμένο υπόδειγμα διατηρείται ως δεδομένο επικύρωσης για τη δοκιμή του μοντέλου και τα υπόλοιπα k − 1 υποδείγματα χρησιμοποιούνται ως δεδομένα εκπαίδευσης. Στη συνέχεια, η διαδικασία διασταυρούμενης επικύρωσης επαναλαμβάνεται k φορές, με καθένα από τα k υποδείγματα να χρησιμοποιείται ακριβώς μία φορά ως δεδομένα επικύρωσης (Εικ. 22). Τα αποτελέσματα k μπορούν στη συνέχεια να υπολογιστούν κατά μέσο όρο για να παραχθεί μια ενιαία εκτίμηση. Το πλεονέκτημα αυτής της μεθόδου έναντι της επαναλαμβανόμενης τυχαίας υποδειγματοληψίας είναι ότι όλες οι παρατηρήσεις χρησιμοποιούνται τόσο για εκπαίδευση όσο και για επικύρωση, και κάθε παρατήρηση χρησιμοποιείται για επικύρωση ακριβώς μία φορά. Στην παρούσα εργασία χρησιμοποιήθηκε η 10-πλάσια (κ=10) διασταυρούμενη επικύρωση [30].



Εικόνα 22: k-fold cross-validation (πηγή: [31])

Η χρονοσειρές εκπαίδευσης τροφοδοτούνται εξ ολοκλήρου στο προτεινόμενο μοντέλο ενώ οι χρονοσειρές ελέγχου τροφοδοτούνται μέχρι ένα επιλεγμένο ποσοστό του παιχνιδιού. Επομένως στόχος του μοντέλου είναι η πρόβλεψη της τελικής έκβασης ενός αγώνα βασιζόμενο στο προεπιλεγμένο ποσοστό στιγμιότυπων.

Σε αντίθεση με τις υπάρχουσες προσεγγίσεις, στην πρόβλεψη των τελικών αποτελεσμάτων ενός αγώνα, η παρούσα τεχνική δεν βασίζεται στην υλοποίηση ενός εξειδικευμένου μοντέλου για κάθε διαφορετική ομάδα, όπως επίσης δεν χρειάζεται ιστορικότητα προηγούμενων αποτελεσμάτων.



Διάγραμμα 1: Μοντελοποίηση Μοντέλου

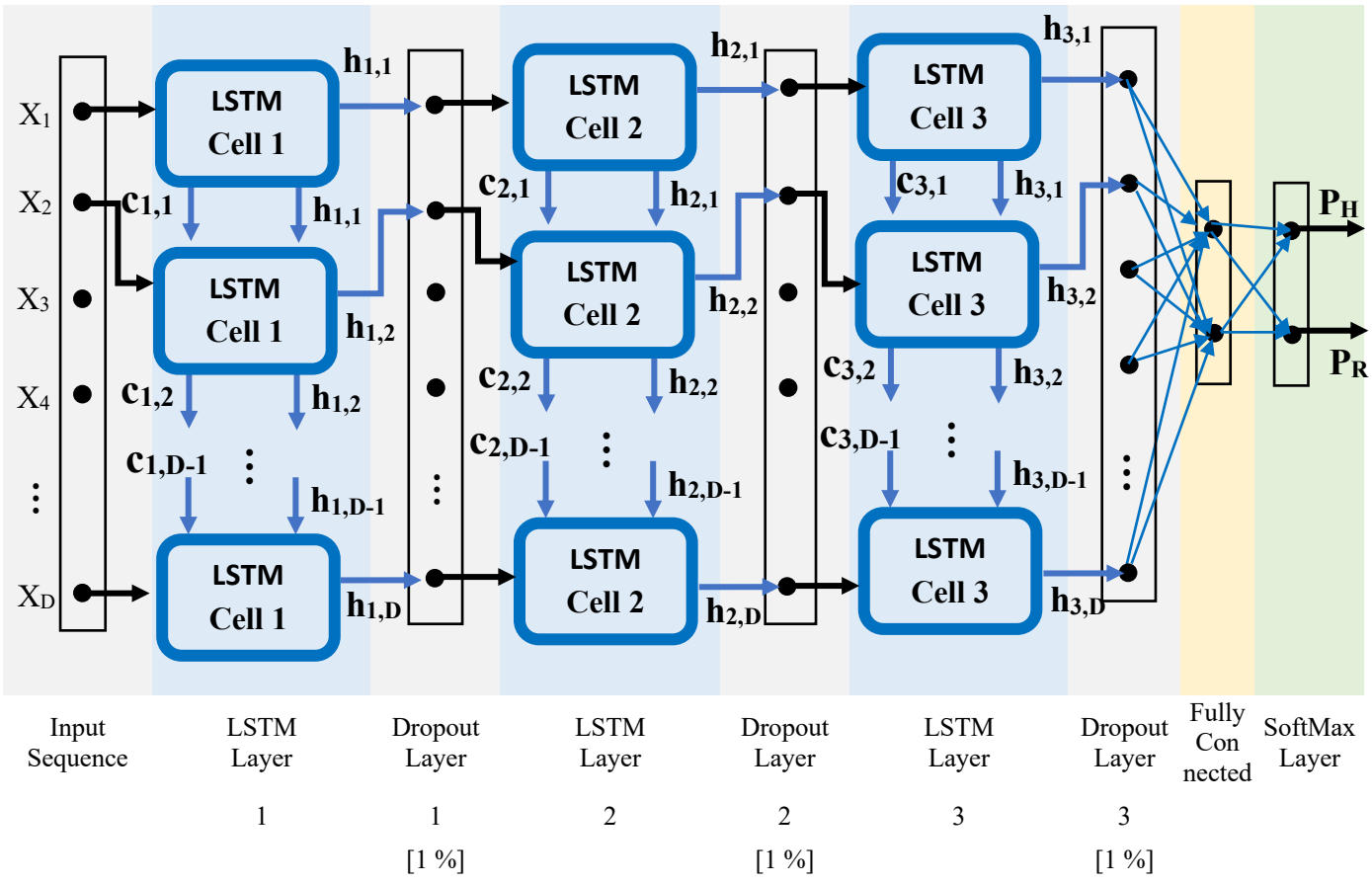
Πρόκειται δηλαδή για ένα μοντέλο που κατά την διάρκεια της χρονικής εξέλιξης ενός παιχνιδιού προσπαθεί να αντιληφθεί μικρής εμβέλειας μεταβολές στην επίδοση των ομάδων που αγωνίζονται, έχοντας την ικανότητα να προβλέψει το τελικό αποτέλεσμα βασιζόμενο σε ένα περιορισμένο σύνολο στιγμιότυπων.

Συνεπώς, εκπαιδεύεται στο τρόπο με το οποίο μια οποιαδήποτε ομάδα προσπαθεί να φέρει το τελικό θετικό αποτέλεσμα, βασιζόμενο στα δυναμικά στατιστικά που λαμβάνει κατά την εξέλιξη του παιχνιδιού.

4.3 Αρχιτεκτονική Μοντέλου

Η βασική αρχιτεκτονική του προτεινόμενου νευρωνικού δικτύου περιλαμβάνει το επίπεδο εισόδου στο οποίο τροφοδοτούνται σειριακά τα στιγμιότυπα του εκάστοτε αγώνα καθώς και τρία κρυφά επίπεδα LSTM κυττάρων, διάστασης όση και η διάσταση του των διανυσμάτων εισόδου ή στιγμιοτύπων του αγώνα. Μεταξύ των τριών κρυφών επιπέδων που αποτελούνται από κύτταρα LSTM παρεμβάλλονται επίπεδα αποκοπής (dropout), εντός

των οποίων ένα ποσοστό 1% των συνδέσεων αποκόπτεται με τυχαίο τρόπο. Ειδικότερα, κάθε ενδιάμεσο κρυφό επίπεδο κυττάρων LSTM μεταδίδει τις κρυφές καταστάσεις των κυττάρων του, ως εισόδους στο αμέσως επόμενο επίπεδο dropout οι οποίες με την σειρά τους θα τροφοδοτηθούν ως είσοδοι στο επόμενο επίπεδο κυττάρων LSTM. Το τελευταίο επίπεδο dropout ακολουθείται από ένα πλήρως συνδεδεμένο επίπεδο δύο κόμβων το οποίο παρέχει την ενδιάμεση έξοδο του νευρωνικού δικτύου πριν από την τελική έξοδο, την οποία παρέχει ένα softmax επίπεδο. Στο τελευταίο επίπεδο της προτεινόμενη αρχιτεκτονικής υπάρχουν δυο κόμβοι εξόδου οι οποίοι υπολογίζουν την πιθανότητα για την κάθε διαφορετική έκβαση του αγώνα.



Διάγραμμα 2: Διάγραμμα Αρχιτεκτονικής Μοντέλου

Το διάνυσμα $\boldsymbol{x}(t) = [x_1(t) \quad x_2(t) \quad \dots \quad x_D(t)]$ αντιστοιχεί στο εκάστοτε στιγμιότυπο του αγώνα όπως έχει αυτό περιγραφεί. Το διάνυσμα $\boldsymbol{c_k}(t) = [c_{k,1}(t) \quad c_{k,2}(t) \quad \dots \quad c_{k,D}(t)]$ αντιστοιχεί στις καταστάσεις των LSTM κυττάρων του κ-οστού επιπέδου ($1 \le k \le 3$) κατά την στιγμή όπου το νευρωνικό δίκτυο έχει τροφοδοτηθεί με το t-οστό στιγμιότυπο, όπου $1 \le t \le N$ με το N να συμβολίζει το πλήθος των στιγμιοτύπων του αγώνα. Επιπλέον, το διάνυσμα $\boldsymbol{h_k}(t) =$

$[h_{k,1}(t) \quad h_{k,2}(t) \quad \dots \quad h_{k,D}(t)]$ αποδίδει τις αντίστοιχες κρυφές καταστάσεις των LSTM κυττάρων του κ-οστού επιπέδου. Τέλος, η έξοδος του νευρωνικού δικτύου δίνεται από το διάνυσμα των πιθανοτήτων $\boldsymbol{P} = [P_H \quad P_R]$, όπου P_H είναι η πιθανότητα να κερδίσει τον αγώνα η ομάδα που αγωνίζεται εντός έδρας και P_R η πιθανότητα να κερδίσει τον αγώνα η φιλοξενούμενη ομάδα.

5 Αποτελέσματα Πειραμάτων & Συμπεράσματα

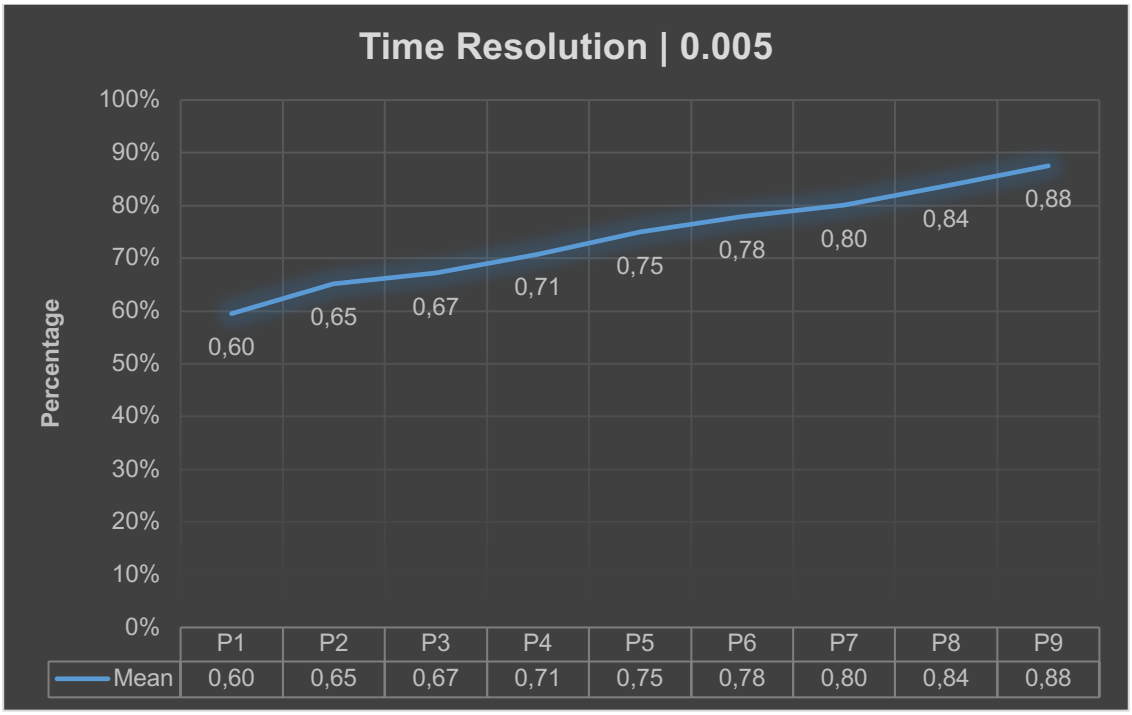
Η διαδικασία που αναπτύχθηκε προκειμένου να επαληθευθεί η ικανότητα πρόγνωσης του προτεινόμενου νευρωνικού δικτύου περιλαμβάνει ένα σύνολο διαφορετικών πειραμάτων στα οποία μεταβλήθηκαν η παράμετρος time resolution που αντιστοιχεί στο πλήθος των στιγμιοτύπων του αγώνα και το ποσοστό των στιγμιοτύπων του αγώνα που χρησιμοποιήθηκαν κατά την φάση ελέγχου πριν την απόδοση της τελικής πρόγνωσης. Με αυτό τον τρόπο διερευνήθηκε η επίδραση του time resolution στην ακρίβεια του μοντέλου αλλά και το πόσο επηρεάζεται αυτή όταν το νευρωνικό δίκτυο καλείται να προβλέψει την τελική έκβαση του αγώνα βασιζόμενο σε ολοένα και μικρότερο ποσοστό του. Συγκεκριμένα στα πειράματα που διερευνήθηκαν η παράμετρος του time resolution πήρε τιμές ίσες με 0.005 , 0.01 και 0.1 για την χρήση 200, 100 και 10 στιγμιοτύπων αντίστοιχα. Για κάθε διαφορετική τιμή του time resolution, εκτελέστηκε ένα πλήθος πειραμάτων για εννιά διαφορετικά χρονικά ορόσημα της αναμέτρησης. Πιο συγκεκριμένα, στο P_1 γίνεται πρόβλεψη στο 10% της αναμέτρησης, στο P_2 για το 20%, στο P_3 για το 30%, στο P_4 για το 40%, στο P_5 για το 50%, στο P_6 για το 60%, στο P_7 για το 70%, στο P_8 για το 80% και στο P_9 για το 90%.

5.1 Πρόβλεψη Νικήτριας Ομάδας | Time Resolution 0,005

Στον πίνακα 2 παρατίθενται τα αποτελέσματα για το time resolution ίσο με 0,005 που αντιστοιχεί σε 200 στιγμιότυπα της αναμέτρησης. Στο ίδιο πίνακα αναφέρονται οι επιμέρους τιμές της ακρίβειας για κάθε διαφορετικό χρονικό ορόσημο P_i καθώς και ο αντίστοιχος μέσος όρος.

%	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	Mean
P_1	0,62	0,60	0,59	0,57	0,61	0,56	0,57	0,60	0,64	0,59	**0,60**
P_2	0,69	0,62	0,69	0,62	0,64	0,65	0,63	0,65	0,63	0,69	**0,65**
P_3	0,73	0,71	0,61	0,74	0,67	0,66	0,68	0,70	0,58	0,64	**0,67**
P_4	0,72	0,70	0,67	0,72	0,69	0,69	0,71	0,70	0,72	0,74	**0,71**
P_5	0,79	0,75	0,79	0,72	0,77	0,72	0,68	0,78	0,71	0,78	**0,75**
P_6	0,81	0,78	0,79	0,80	0,77	0,74	0,80	0,76	0,75	0,78	**0,78**
P_7	0,78	0,81	0,81	0,80	0,79	0,82	0,81	0,83	0,76	0,80	**0,80**
P_8	0,84	0,87	0,83	0,84	0,79	0,82	0,87	0,83	0,82	0,86	**0,84**
P_9	0,85	0,88	0,87	0,89	0,87	0,87	0,87	0,88	0,89	0,89	**0,88**

Πίνακας 2: Αναλυτικά Αποτελέσματα 10-fold cross-validation | Time Resolution 0,005



Γράφημα 1: Percentage VS Mean | Time Resolution 0,005

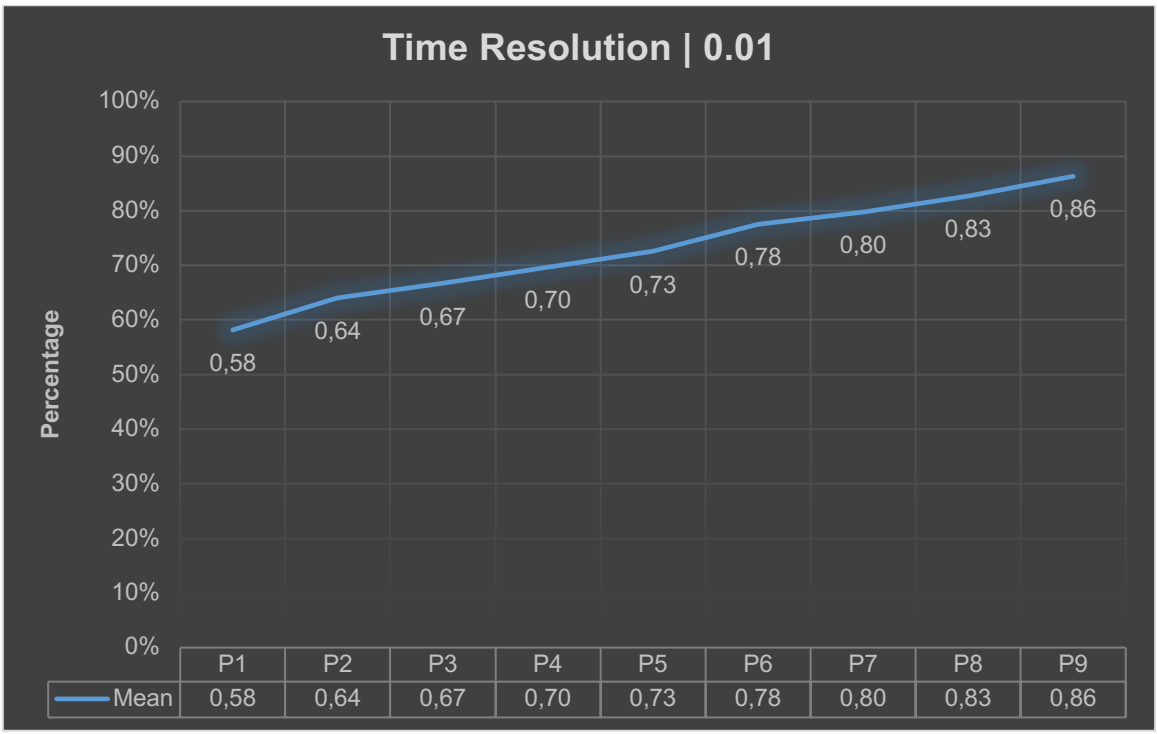
Παρατηρούμε ότι η ακρίβεια του μοντέλου μειώνεται όσο αυξάνεται ο χρονικός ορίζοντας της πρόβλεψης χωρίς ωστόσο να πέφτει κάτω από 60% ακόμα και στην περίπτωση όπου το ποσοστό των στιγμιοτύπων του αγώνα που χρησιμοποιήθηκαν για την πρόβλεψη ήταν ίσο με το 10% του αγώνα. Επίσης είναι αξιοσημείωτο ότι στο 50% της αναμέτρησης η ακρίβεια της πρόγνωσης είναι της τάξης του 75%.

5.2 Πρόβλεψη Νικήτριας Ομάδας | Time Resolution 0,01

Στον πίνακα 3 παρατίθενται τα αποτελέσματα για το time resolution ίσο με 0,01 που αντιστοιχεί σε 100 στιγμιότυπα της αναμέτρησης. Στο ίδιο πίνακα αναφέρονται οι επιμέρους τιμές της ακρίβειας για κάθε διαφορετικό χρονικό ορόσημο Pi καθώς και ο αντίστοιχος μέσος όρος.

%	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	Mean
P_1	0,59	0,62	0,58	0,56	0,59	0,56	0,56	0,61	0,56	0,59	**0,58**
P_2	0,62	0,67	0,64	0,58	0,67	0,58	0,61	0,67	0,71	0,65	**0,64**
P_3	0,63	0,69	0,67	0,65	0,68	0,67	0,71	0,65	0,69	0,65	**0,67**
P_4	0,72	0,69	0,69	0,69	0,70	0,75	0,69	0,64	0,67	0,74	**0,70**
P_5	0,77	0,75	0,74	0,75	0,72	0,73	0,71	0,69	0,71	0,69	**0,73**
P_6	0,72	0,79	0,76	0,78	0,78	0,71	0,81	0,77	0,80	0,83	**0,78**
P_7	0,81	0,80	0,79	0,80	0,83	0,80	0,80	0,76	0,80	0,78	**0,80**
P_8	0,84	0,84	0,81	0,85	0,86	0,81	0,80	0,81	0,77	0,87	**0,83**
P_9	0,87	0,86	0,86	0,89	0,87	0,86	0,86	0,87	0,86	0,84	**0,86**

Πίνακας 3: Αναλυτικά Αποτελέσματα 10-fold cross-validation | Time Resolution 0,01



Γράφημα 2: Percentage VS Mean | Time Resolution 0,01

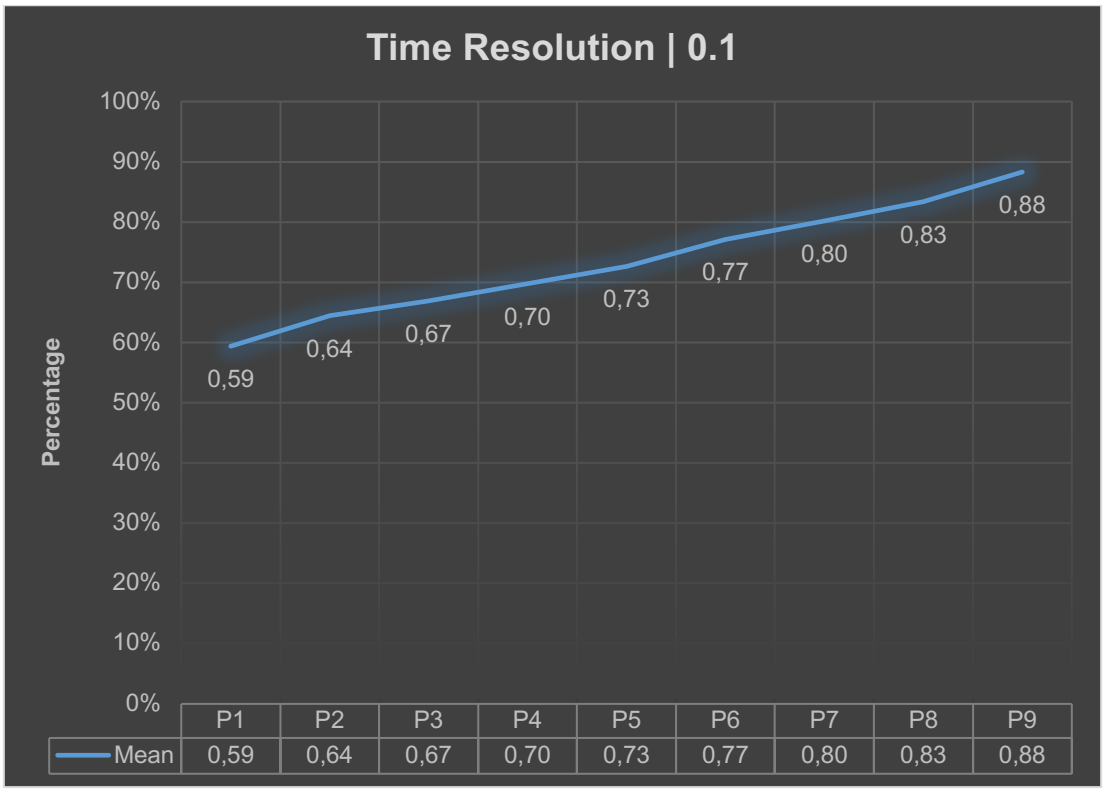
Παρατηρούμε ότι η ακρίβεια του μοντέλου μειώνεται όσο αυξάνεται ο χρονικός ορίζοντας της πρόβλεψης χωρίς ωστόσο να πέφτει κάτω από **58**% ακόμα και στην περίπτωση όπου το ποσοστό των στιγμιοτύπων του αγώνα που χρησιμοποιήθηκαν για την πρόβλεψη ήταν ίσο με το 10% του αγώνα. Γενικότερα οι διαφορές με το time resolution 0,005 δεν παρουσιάζουν μεγάλες αποκλίσεις αφού κυμαίνονται από 0,02 – 0,04. Τέλος, είναι αξιοσημείωτο ότι στο 50% της αναμέτρησης η ακρίβεια της πρόγνωσης είναι της τάξης του 73% ελαφρώς μειωμένη από το αντίστοιχο πείραμα με time resolution 0,005.

5.3 Πρόβλεψη Νικήτριας Ομάδας | Time Resolution 0,1

Στον πίνακα 3 παρατίθενται τα αποτελέσματα για το time resolution ίσο με 0,1 που αντιστοιχεί σε 10 στιγμιότυπα της αναμέτρησης. Στο ίδιο πίνακα αναφέρονται οι επιμέρους τιμές της ακρίβειας για κάθε διαφορετικό χρονικό ορόσημο Pi καθώς και ο αντίστοιχος μέσος όρος.

%	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	Mean
P_1	0,57	0,63	0,56	0,66	0,52	0,62	0,56	0,61	0,60	0,60	**0,59**
P_2	0,64	0,69	0,62	0,58	0,63	0,69	0,67	0,67	0,60	0,67	**0,64**
P_3	0,65	0,69	0,67	0,67	0,64	0,69	0,71	0,62	0,65	0,70	**0,67**
P_4	0,69	0,71	0,70	0,73	0,69	0,70	0,70	0,67	0,72	0,66	**0,70**
P_5	0,73	0,70	0,75	0,78	0,72	0,74	0,72	0,68	0,73	0,72	**0,73**
P_6	0,80	0,69	0,78	0,79	0,78	0,76	0,80	0,80	0,78	0,74	**0,77**
P_7	0,79	0,81	0,80	0,80	0,82	0,82	0,78	0,79	0,80	0,80	**0,80**
P_8	0,80	0,84	0,86	0,83	0,82	0,86	0,84	0,83	0,83	0,84	**0,83**
P_9	0,86	0,89	0,88	0,87	0,90	0,89	0,90	0,90	0,89	0,84	**0,88**

Πίνακας 4: Αναλυτικά Αποτελέσματα 10-fold cross-validation | Time Resolution 0,1



Γράφημα 3: Percentage VS Mean | Time Resolution 0,1

Παρατηρούμε ότι η ακρίβεια του μοντέλου μειώνεται όσο αυξάνεται ο χρονικός ορίζοντας της πρόβλεψης χωρίς ωστόσο να πέφτει κάτω από 59% ακόμα και στην περίπτωση όπου το ποσοστό των στιγμιοτύπων του αγώνα που χρησιμοποιήθηκαν για την πρόβλεψη ήταν ίσο με το 10% του αγώνα. Γενικότερα οι διαφορές με το time resolution 0,005 και 0,01 δεν παρουσιάζουν μεγάλες αποκλίσεις αφού κυμαίνονται από 0,02 – 0,06. Αξιοσημείωτο είναι για άλλη μια φορά το γεγονός ότι στο 50% της αναμέτρησης η ακρίβεια της πρόγνωσης είναι της τάξης του 73% πολύ κοντά στα αντίστοιχα πειράματα με time resolution 0,005 & 0,01.

5.4 Συμπεράσματα

Με βάση το σύνολο των πειραμάτων που αναπτύχθηκαν παραπάνω μπορεί να υποστηριχθεί προς το προτεινόμενο μοντέλο διαθέτει επαρκή ακρίβεια πρόγνωσης όταν έχει τροφοδοτηθεί με το 40% της αναμέτρησης. Που αντικειμενικά είναι ένα σχετικά μικρό μέρος της αναμέτρησης αν αναλογιστεί κανείς την ακρίβεια της πρόβλεψης που ξεπερνά το 70% σε κάθε σενάριο δοκιμών.

Παρατηρείται ότι η συνολική ακρίβεια του μοντέλου δεν επηρεάζεται σημαντικά από την επιλεγμένη τιμή του time resolution. Το γεγονός αυτό μπορεί να παρέχει την δυνατότητα μια περισσότερο μακροπρόθεσμης πρόβλεψης, όταν το πλήθος των στιγμιοτύπων του κάθε αγώνα λαμβάνει μικρότερες τιμές.

Μελλοντικές Εργασίες

Για τις ανάγκες των δοκιμών χρησιμοποιήθηκε time resolution ίσο με 0.005 , 0.01 και 0.1 για την χρήση 200, 100 και 10 στιγμιοτύπων αντίστοιχα. Το γεγονός ότι το time resolution δεν επηρεάζει σημαντικά τα αποτελέσματα δίνει μια προοπτική εναλλακτικού τρόπου εκπαίδευσης. Ειδικότερα στην περίπτωση του ενός μεγάλου time resolution ο κάθε αγώνας θα μπορούσε να θεωρηθεί ως μια συλλογή περισσότερων μικρών αγώνων. Η συγκεκριμένη πρακτική μπορεί να δώσει την δυνατότητα για μια μελλοντική πρόγνωση σε πολύ μικρή χρονική κλίμακα και ταυτόχρονα με μεγαλύτερη ακρίβεια.

Το προτεινόμενο μοντέλο έχει την ικανότητα να παράγει βραχυπρόθεσμες προβλέψεις για καθένα από αυτά τα χαρακτηριστικά - στατιστικά που συνθέτουν το στιγμιότυπο του αγώνα. Σε παρόμοια λογική θα μπορούσε να χρησιμοποιηθεί ανάλογο μοντέλο που να καλύπτει επαρκώς την πρόβλεψη της τελικής διαφοράς των πόντων των δύο ομάδων. Πιο συγκεκριμένα η μέθοδος αυτή, ανάλογα με το time resolution που θα εξεταστεί μπορεί να παράγει αντίστοιχα στιγμιότυπα. Τα στιγμιότυπα αυτά μπορούν αθροιστικά να μας δίνουν την εκάστοτε διαφορά των δυο ομάδων στην εξέλιξη των αγώνων.

Τέλος, θα μπορούσε να χτιστεί ένα πιο εξειδικευμένο μοντέλο εξέλιξης ενός παιχνιδιού . Ένα μοντέλο εξατομικευμένο σε κάθε ομάδα με απώτερο στόχο την ακρίβεια της πρόβλεψης. Σε αυτή την περίπτωση είναι βέβαιο ότι θα χρειαστεί επιπλέον συλλογή χαρακτηριστικών ικανών να υποδεικνύουν την δυναμικότητα κάθε ομάδας, αναδεικνύοντας χαρακτηριστικά που σηματοδοτούν την πρόβλεψη της νίκη με μεγαλύτερη πιθανότητα.

Βιβλιογραφικές Αναφορές

[1] «175 Zettabytes By 2025,» Forbes, 2018. [Ηλεκτρονικό]. Available: https://www.forbes.com/sites/tomcoughlin/2018/11/27/175-zettabytes-by-2025/?sh=5f8c04e45459.

[2] "Oxford definition of analytics". Archived from the original on August 10, 2020..

[3] "Cognitive Analytics - combining Artificial Intelligence (AI) and Data Analytics". www.ulster.ac.uk. March 8, 2017. Retrieved January 7, 2022..

[4] Basu, Atanu, Five pillars of prescriptive analytics success, Analytics, March / April 2013, accessed 3 December 2022.

[5] "To predict or not to Predict". mccoy-partners.com. Retrieved 2022-05-05..

[6] "Drawing Conclusions From Data: Descriptive Statistics, Inferential Statistics, and Hypothesis Testing", Interpreting and Using Statistics in Psychological Research, Thousand Oaks, CA: SAGE Publications, Inc, pp. 145–183, 2017, doi:10.4135/9781506304144..

[7] Alamar, B. (2013). Sports analytics: A guide for coaches, managers, and other decision makers. New York: Columbia University Press..

[8] Caya, O., & Bourdon, A. (2016). A Framework of Value Creation from Business Intelligence and Analytics in Competitive Sports. In System Sciences (HICSS), 2016 49th Hawaii International Conference on (pp. 1061-1071). IEEE..

[9] Sports Analytics Market Size, Share & Trends Analysis Report By Component (Software, Service), By Analysis Type (On-field, Off-field), By Sports (Football, Cricket, Basketball, Baseball), And Segment Forecasts, 2022 - 2030.

[10] «Grand View Research,» [Ηλεκτρονικό]. Available: www.grandviewresearch.com.

[11] «Wikipedia | Moneyball,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Moneyball.

[12] «Soccerment,» [Ηλεκτρονικό]. Available: analytics.soccerment.com.

[13] Healey G. Combining radar and optical sensor data to measure player value in baseball. Sensors (Switzerland) MDPI AG. 2021;21:1–14..

[14] Healey G. The new moneyball: how ballpark sensors are changing baseball. Proceedings of the IEEE. Institute of Electrical and Electronics Engineers Inc.; 2017. p. 1999–2002..

[15] «Technology.mlblogs.com,» [Ηλεκτρονικό]. Available: https://technology.mlblogs.com/introducing-statcast-2020-hawk-eye-and-google-cloud-a5f5c20321b8.

[16] Yang, Z.R.; Yang, Z. (2014). Comprehensive Biomedical Physics. Karolinska Institute, Stockholm, Sweden: Elsevier. p. 1. ISBN 978-0-444-53633-4..

[17] Mitchell, Tom (1997). Machine Learning. New York: McGraw Hill. ISBN 0-07-042807-7. OCLC 36417892..

[18] Stuart J. Russell, Peter Norvig (2010) Artificial Intelligence: A Modern Approach, Third Edition, Prentice Hall ISBN 9780136042594..

[19] Hinton & Sejnowski 1999.

[20] Schulz, Hannes; Behnke, Sven (1 November 2012). "Deep Learning". KI - Künstliche Intelligenz. 26 (4): 357–363. doi:10.1007/s13218-012-0198-z. ISSN 1610-1987. S2CID 220523562..

[21] Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276. S2CID 1915014..

[22] «intelligenzaartificialeitalia.net,» [Ηλεκτρονικό]. Available: https://www.intelligenzaartificialeitalia.net/post/gli-algoritmi-di-deep-learning-o-apprendimento-profondo-più-diffusi-e-usati-nel-2021.

[23] In addition to the original authors, a lot of people contributed to the modern LSTM. A non-comprehensive list is: Felix Gers, Fred Cummins, Santiago Fernandez, Justin Bayer, Daan Wierstra, Julian Togelius, Faustino Gomez, Matteo Gagliolo, and Alex Graves..

[24] Rathborn, Jack (November 18, 2020). "NBA Draft 2020: What time does it start in the UK, who has the No 1 pick and how can I watch it?". The Independent. Archived from the original on June 18, 2022. Retrieved December 10, 2020. The 2020 NBA Draft is here.

[25] «1Courel, Javier; Suárez, Ernesto; Ortega, Enrique; Piñar, Maribel; Cárdenas, David. “Is the inside pass a performance indicator? Observational analysis of elite basketball

teams.” Revista de Psicología del Deporte, Universitat de les Illes Balears, Jan. 2,» [Ηλεκτρονικό].

[26] «Basketball Noise,» [Ηλεκτρονικό]. Available: https://basketballnoise.com/how-do-you-read-an-nba-box-score/.

[27] www.mathworks.com, "Matrices and Arrays - MATLAB & Simulink"., www.mathworks.com , 2022.

[28] N. Chonacky και D. Winch, "Reviews of Maple, Mathematica, and Matlab: Coming Soon to a Publication Near You"., Computing in Science & Engineering.: Institute of Electrical and Electronics Engineers (IEEE). 7 (2): 9–10, 2005.

[29] D. M. Allen, "The Relationship between Variable Selection and Data Agumentation and a Method for Prediction"., Technometrics. 16 (1): 125–127. doi:10.2307/1267500. JSTOR 1267500., 1974.

[30] G. J. McLachlan, K.-A. Do και C. Ambroise, Analyzing microarray gene expression data., Wiley, 2004.

[31] «Wikipedia | Cross-Validation,» [Ηλεκτρονικό]. Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics)#cite_note-2.

[32] LeCun, Yann; Bengio, Yoshua; Hinton, Geoffrey (2015). "Deep Learning". Nature. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. S2CID 3074096..

Παράρτημα Α: Πρόγραμμα Συλλογής Play-by-Play | NBA

Στιγμιότυπα σχολιασμένου κώδικα Python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Dec 14 20:37:07 2022

@author: Sarris Antonios
"""

# import our packages
import pandas as pd
import sys

headers  = {
    'Connection': 'keep-alive',
    'Accept': 'application/json, text/plain, */*',
    'x-nba-stats-token': 'true',
    'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X
    'x-nba-stats-origin': 'stats',
    'Sec-Fetch-Site': 'same-origin',
    'Sec-Fetch-Mode': 'cors',
    'Referer': 'https://stats.nba.com/',
    'Accept-Encoding': 'gzip, deflate, br',
    'Accept-Language': 'en-US,en;q=0.9',
}

from nba_api.stats.endpoints import leaguegamefinder
from nba_api.stats.endpoints import playbyplayv2

# get game logs from the reg season
gamefinder                                              =
leaguegamefinder.LeagueGameFinder(season_nullable='2022-
23',  league_id_nullable='00',season_type_nullable='Regular
Season')

games = gamefinder.get_data_frames()[0]
#print(games.head(10))

gameIDs = games['GAME_ID'].unique().tolist()
#print(gameIDs)

gameIDs_sample = ['0022001072', '0022001070', '0022001067']

def get_data(game_id):
    play_by_play_url                                    =
"https://cdn.nba.com/static/json/liveData/playbyplay/playb
yplay_"+game_id+".json"
```

```
    response = requests.get(url=play_by_play_url,
headers=headers).json()
    play_by_play = response['game']['actions']
    df = pd.DataFrame(play_by_play)
    df['gameid'] = game_id
    return df

#def print_progress_bar(index, total, label):
#    sys.stdout.write('\r')
#            sys.stdout.write(f"[{'='   *   int(n_bar   *
progress):{n_bar}s}] {int(100 * progress)}%  {label}")
#    sys.stdout.flush()
def  percent_complete(step,  total_steps,  bar_width=60,
title="", print_perc=True):

    # UTF-8 left blocks: 1, 1/8, 1/4, 3/8, 1/2, 5/8, 3/4,
7/8
    utf_8s = ["█", "▏", "▎", "▍", "▌", "▋", "▊", "▉"]
    perc = 100 * float(step) / float(total_steps)
    max_ticks = bar_width * 8
    num_ticks = int(round(perc / 100 * max_ticks))
    full_ticks = num_ticks / 8      # Number of full blocks
    part_ticks = num_ticks % 8      # Size of partial block
(array index)

    disp = bar = ""                 # Blank out variables
    bar += utf_8s[0] * int(full_ticks)  # Add full blocks
into Progress Bar

    # If part_ticks is zero, then no partial block, else
append part char
    if part_ticks > 0:
        bar += utf_8s[part_ticks]

    # Pad Progress Bar with fill character
    bar += "▒" * int((max_ticks/8 - float(num_ticks)/8.0))

    if len(title) > 0:
        disp = title + ": "           # Optional title to
progress display

    #      Print      progress      bar      in      green:
https://stackoverflow.com/a/21786287/6929343
    disp += "\x1b[0;32m"            # Color Green
    disp += bar                          # Progress bar to
progress display
    disp += "\x1b[0m"               # Color Reset
    if print_perc:
        #  If  requested,  append  percentage  complete  to
progress display
        if perc > 100.0:
```

```
            perc = 100.0           # Fix "100.04 %" rounding
error
        disp += " {:6.2f}".format(perc) + " %"

    # Output to terminal repetitively over the same line
using '\r'.
    sys.stdout.write("\r" + disp)
    sys.stdout.flush()

pbpdata = []
total = len(gameIDs)
print(total,"Games")
for index, game_id in enumerate(gameIDs):
    game_data = get_data(game_id)
    pbpdata.append(game_data)
    #print_progress_bar(index, total, "NBA bar")
    percent_complete(index,   total,   title="NBA   Scraping
Progress")

final_df = pd.concat(pbpdata, ignore_index=True)

# Save  DataFrame to disk
final_df.to_csv(r'NBA_Play-by-Play_2022_23.csv',
index=False)
final_df.to_excel(r'NBA_Play-by-Play_2022_23.xlsx',
index=False)
```

Παράρτημα Β: Πρόγραμμα Διαμόρφωσης Play-by-Play Δεδομένων

Στιγμιότυπα σχολιασμένου κώδικα Python

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on 30 Dec 2023  10:20:45 2022

@author: Sarris Antonios
"""

#imports
### imports
import numpy as np
import pandas as pd
import re
import datetime as dt

from functools import reduce
pd.set_option('precision', 2)

import warnings
from pandas.core.common import SettingWithCopyWarning

warnings.simplefilter(action="ignore",
category=SettingWithCopyWarning)

import glob
import os
import sys
import time

pd.set_option('max_columns', None)
#pd.set_option('max_rows', None)

def getTeams(game_id,session):
    #print("game_id: ",game_id)

    teams = pd.read_excel('Data/teams.xlsx')
    teams['INITIALS'] = teams['INITIALS'].str.upper()
    teams = teams.rename(columns={'SHORT NAME':'team'})
    temp = pd.merge(session, teams, on='team')

    session = temp[['game_id','player','AKR']]

    session['game_id'] = session['game_id'].apply(str)
    temp86 = session[session['game_id'] == game_id]
    #team = temp86['AKR'].iloc[0]
```

```
    team = temp86

    #print('team:',team)
    return team

def getTeams1(game_id,session):
    #print("game_id: ",game_id)

    teams = pd.read_excel('Data/teams.xlsx')
    teams['INITIALS'] = teams['INITIALS'].str.upper()
    teams = teams.rename(columns={'SHORT NAME':'team'})
    temp = pd.merge(session, teams, on='team')

    session = temp[['game_id','player','AKR']]

    session['game_id'] = session['game_id'].apply(str)
    #temp86 = session[session['game_id'] == game_id]
    #team = temp86['AKR'].iloc[0]
    #team = temp86
    team = session
    #print('team:',team)
    return team

    pbpDF['play_length'] = np.where((pbpDF['play_length']
== '00:-12:00') | (pbpDF['play_length'] == '00:-5:00')
,'00:00:00',pbpDF['play_length'])

    #pbpDF['play_length'].replace('00:-12:00' ,'00:00:00')
    #print(pbpDF['play_length'])
    pbpDF['play_length'] =
pd.to_timedelta(pbpDF['play_length'])
    pbpDF['play_length'] =
pbpDF['play_length'].dt.total_seconds()

    # Change the other time columns to time deltas
    pbpDF['elapsed'] = pd.to_timedelta(pbpDF['elapsed'])
    pbpDF['elapsed'].replace('00:-12:00' ,'00:00:00')
    pbpDF['remaining_time'] =
pd.to_timedelta(pbpDF['remaining_time'])
    pbpDF['remaining_time'].replace('00:-12:00'
,'00:00:00')
    #print(pbpDF.away.loc[~pbpDF.away.isnull()].iloc[0])
    game_id = str(pbpDF['game_id'].iloc[0]).lstrip('0')
    #print(game_id)

    # Create dummy variables for the player columns
    player_df = pd.get_dummies(pbpDF.filter(regex='a[1-
5]|h[1-5]'), prefix='player')

    # Remove the whitespace in the dummy player columns
```

```
    player_df.columns = [x.strip().replace(' ', '_') for x
in player_df.columns]

    # Collapse the duplicate dummy player columns and sum
the column values
    player_df = player_df.groupby(lambda x:x,
axis=1).sum()


    # Bring the dummy columns into the main dataframe
    pbpDF = pd.concat([pbpDF, player_df], axis=1)

    #dff = playersDF2[playersDF2['player'] ==

    three_df =
pbpDF[pbpDF.description.str.contains('3PT').fillna(False)]

    three_df = pd.pivot_table(three_df, index=['player'],
                          columns=['result'],
                          values=['play_id'],

aggfunc='count').reset_index(col_fill='player')

    three_df['3pa'] =
np.where((three_df['3pa'].isnull()),three_df['3p'],three_d
f['3pa'])
    three_df['3p%'] = three_df['3p'] / three_df['3pa']


    reb_df = pbpDF[pbpDF.type.isin(['rebound offensive',
'rebound defensive'])]
    #print(reb_df)
    reb_df = pd.pivot_table(reb_df, index=['player'],
                                columns=['type'],
                                values=['play_id'],

aggfunc='count').reset_index(col_fill='player')


    reb_df.columns = reb_df.columns.droplevel(0)
    reb_df.columns.name = None
    reb_df = reb_df.rename(columns={'rebound
offensive':'orb','rebound defensive':'drb'})

    #print(reb_df)

    ### if column not exist- create with zeroes
    if 'orb' not in reb_df.columns: reb_df['orb'] = 0
    reb_df['orb'] = reb_df['orb'].fillna(0)

    ### if column not exist- create with zeroes
```

```
    if 'drb' not in reb_df.columns: reb_df['drb'] = 0
    reb_df['drb'] = reb_df['drb'].fillna(0)


    reb_df['trb'] = reb_df['drb'] + reb_df['orb']


    ast_df =
pd.DataFrame(pbpDF.assist.value_counts()).reset_index().re
name(columns={'index':'player',

'assist':'ast'})
    #print(ast_df)
    #print()

    stl_df =
pd.DataFrame(pbpDF.steal.value_counts()).reset_index().ren
ame(columns={'index':'player',

'steal':'stl'})
    blk_df =
pd.DataFrame(pbpDF.block.value_counts()).reset_index().ren
ame(columns={'index':'player',

'block':'blk'})

    # types = sorted(df.type.unique())
    # foul_types = [s for s in types if "foul" in s]
    # print(foul_types)

    ## foul keyword kinds
    fouls = ['foul', 'off.foul', 'offensive charge foul',
'p.foul', 's.foul','l.b.foul',
         'away.from.play.foul','flagrant.foul.type1']
         #,'c.p.foul','flagrant.foul.type1','t.foul def. 3
sec',
         #'personal take
foul','unknown','hanging.tech.foul','o','flagrant.foul.typ
e2']

    pf_df = pbpDF[pbpDF.type.isin(fouls)]

    pf_df = pd.pivot_table(pf_df, index=['player'],
values=['play_id'],
aggfunc='count').reset_index().rename(columns={'play_id':'
pf'})

    ### if column not exist- create with zeroes
    if 'pf' not in pf_df.columns: pf_df['pf'] = 0

played'] = mp_list
```

```
    #print(min_df)

    pm_df = pd.merge(pts_df, min_df, on='player')
    #print(pts_df)
    #pm_df.head()
    #print(pm_df)
    p_list = list(pm_df.player_unformatted)
    t_list = list(pm_df.team)

    plus = []
    minus = []

    for player, team in zip(p_list, t_list):
        p_nested = []
        plus.append(p_nested)

        m_nested = []
        minus.append(m_nested)

        for i, row in pbpDF.iterrows():
            if (row[player] == 1) & (row['team'] == team):
                p_nested.append(row['points'])
            elif (row[player] == 1) & (row['team'] !=
team):
                m_nested.append(row['points'])

    p_list = []
    m_list = []


    for value in plus:
        value = np.sum(value)
        p_list.append(value)

    for value in minus:
        value = np.sum(value)
        m_list.append(value)

    pm_df['plus'] = p_list
    pm_df['minus'] = m_list
    pm_df['plus_minus'] = pm_df['plus'] - pm_df['minus']
    #print( playersDF)
    # Merge all the dataframes
    #if not playersDF.empty:

    dfs = [pm_df, playersDF,fg_df, ft_df, three_df,
reb_df, ast_df, stl_df, blk_df, to_df, pf_df]
    #from tabulate import tabulate
    #print(tabulate(dfs, headers='keys', tablefmt='psql'))
    #print("############################")
    #print(dfs[0]['pts'])
```

```
    #print("############################")
    #print()
    #else:
        #dfs = [pm_df,fg_df, ft_df, three_df, reb_df,
ast_df, stl_df, blk_df, to_df, pf_df]
    #print(dfs[7])
    #print()
    boxscore = reduce(lambda left, right: pd.merge(left,
right, on=['player'], how='outer'), dfs).fillna(0)
    boxscore['game_id'] =
boxscore['game_id'].astype(np.int64)
    #print(boxscore.info())


    # Drop unnecessary columns
    drop_cols = ['min_bench', 'player_unformatted',
'plus', 'minus']
    boxscore.drop(drop_cols, axis=1, inplace=True)

    # Reorder columns for visuals
    '''
    reorder_cols = ['game_id','team', 'player',
'min_played', 'fg', 'fga', 'fg%',
                '3p', '3pa', '3p%', 'ft', 'fta', 'ft%',
'orb', 'drb',
                'trb', 'ast', 'stl', 'blk', 'tov', 'pf',
'pts', 'plus_minus']
    '''
    reorder_cols1 = ['game_id','team', 'player',

    boxscore = boxscore[reorder_cols1]

    # Reformat column names for visuals
    boxscore = boxscore.rename(columns={"min_played":"mp",
"plus_minus":"+/-"})
    ##boxscore = boxscore[boxscore.team != 0]
    #print(boxscore['pts'])
    #print(str(boxscore['player'].iloc[0]))

##########################################################
########################################

#print(getTeam(game_id,session,str(boxscore['player'].iloc
[0])))
    #print("$$$")
    #print(boxscore)
    #print()
    real = getTeams(game_id,session)
    #real1 = getTeams1(game_id,session)
    #print(real)
    #print()
```

```
    #temp = pd.merge(boxscore,real['AKR'], on='player')
    #print(boxscore.dtypes)
    #print(boxscore)


    if (boxscore ['player']==0).any() :
        boxscore['player'] =
boxscore['player'].astype(int).astype(str)

    #print(real.dtypes)

    temp =
pd.merge(boxscore,real[['AKR','player']],on='player',
how='left')
    temp['AKR']= temp['AKR'].fillna(method='ffill')
    #temp['mp']= temp['mp'].replace(to_replace=0,
method='ffill')
    #print(temp[['team','player','pts']])
    #if (temp ['AKR'].isnull().values.any() ):
    #    print('nan')
    #    temp =
pd.merge(real1[['AKR','player'],boxscore],on='player',
how='left')
    #    print(temp)

    temp.drop(['team'], axis=1, inplace=True)
    temp.rename(columns={'AKR':'team'}, inplace=True)

    boxscore = temp
    #print(boxscore)
    #print(boxscore)
    #print(temp)
    #print("$$$")
    #print("#1#")
    #print(boxscore[['team', 'stl','player']])
    #print()

##########################################################
#####

##########################################################
#######

    #if (boxscore ['team']==0).any():
        #real = getTeams(game_id,session)
        #temp = pd.merge(boxscore,real['AKR'],
on='player')
        #print(temp)
        #input()
        #temp.drop(['game_id','team'], axis=1,
inplace=True)
```

```
        #print(temp)

##########################################################
############
    ##############errorr#############################

##########################################################
#############
    #print("#2#")
    #print(boxscore[['team', 'stl','player']])
    #print()

#print(getTeam(game_id,session,str(boxscore['player'].iloc
[0])))
    boxscore['game_id'] = np.where((boxscore['game_id'] ==
0),game_id,boxscore['game_id'])
    boxscore['team'] = boxscore['team'].astype(str)
    boxscore = boxscore.sort_values('team',
ascending=True).reset_index(drop=True)
    #if any(boxscore.team == ""):
    #print('0: ',boxscore)

    #print(boxscore)
    total =
boxscore.groupby(by=['team','game_id']).sum().reset_index(
)
    #total.loc[(total[['team']] != 0).all(axis=1)]

    ### if rows has zeroes drop rows
    total = total[~(total[['team','game_id']] ==
0).any(axis=1)].reset_index()
    #print()
    #print('###################')
    #print('1: ',total['stl'])
    #print('###################')
    if len(total) ==2:
        ## multiple rows to one
        #print('if')
        ndf = total.unstack().to_frame().T
        ndf.columns =
ndf.columns.map('{0[0]}_{0[1]}'.format)
        ndf = ndf.drop(columns=['index_0', 'index_1',
'game_id_1'])
        ndf = ndf.rename(columns={"game_id_0":"game_id"})
        #ndf.columns = ndf.columns.str.replace("_0", "_H")
        #ndf.columns = ndf.columns.str.replace("_1", "_R")

#########################################################
        #print(ndf.iloc[0]['team_0'])
        #print(ht)
        if ndf.iloc[0]['team_0'] == ht:
```

```
            #print("_0 = home")
            ndf.columns = ndf.columns.str.replace("_0",
"_H")
            ndf.columns = ndf.columns.str.replace("_1",
"_R")
            #print(ndf)
        elif ndf.iloc[0]['team_0'] == at:
            #print("_0 = away")
            ndf.columns = ndf.columns.str.replace("_0",
"_R")
            ndf.columns = ndf.columns.str.replace("_1",
"_H")
        '''
        else:
            print('####################')
            #print(ndf)
            #print('####################')
            ndf.columns = ndf.columns.str.replace("_0",
"_H")
            ndf.columns = ndf.columns.str.replace("_1",
"_R")
        '''


############################################################
        #print
        return ndf
    else:
        #print('Antonis...error...groupbyteams > 2 ')
        #print(len(total))
        #print(total['team'])
        #print(ht)
        if total.iloc[0]['team'] == ht:

            total = total.drop(columns=['mp'], axis=1,
errors='ignore')
            #print("#else#if#: ",total)
            if len(total.columns)  == 19:
                total.loc[1] = [1,at, game_id, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
            elif len(total.columns)  == 18:
                total.loc[1] = [1,at, game_id, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
            #print("total2: ",total)
            total['game_id'] = np.where((total['game_id']
== 0),game_id,total['game_id'])
            #print("total3: ",total)
            #print()
            #print(total)
            #total =
```

```
            if total.empty:
                #print('#else#else#if',total)
                total = total.drop(columns=['mp'], axis=1,
errors='ignore')
                #print('$$$$$$$$$$$$$$$$$')
                #print(total.columns)
                total.loc[0] = [0,ht, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0]
                total.loc[1] = [1,at, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0]
            else:
                #print('#else#else#if#else',total)
                #print('+++++++++++++')
                #print("total0: ", total)
                #print('+++++++++++++')
                if total.loc[0]['team'] == 'None':
                    #print('#else#else#if#else#if',total)
                    #total = total['team'].replace('None',
np.nan, inplace=True)
                    #print("YEEEEEESSSSSSSSSSS!!!!")
                    #total = total.iloc[0:0]
                    total['team'] = 0
                    total = total.drop(columns=['mp'],
axis=1, errors='ignore')
                    #print("total22: ", total)
                    total_temp_1 = total
                    total_temp_1.team = at
                    #total.loc[0] = [0,ht, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
                    #print("total23: ", total_temp_1)
                    #total['team'] = ht
                    total =
total.append(total_temp_1,ignore_index=True)

                    total = total.reset_index()
                    total.loc[1, 'team'] = ht
                    #total_temp_1.loc[1] = [1,at, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
                    #print("total24: ", total)
                    #print("ht: ", ht)
                    #print("at: ", at)
                    #print(total)

                else:

#print('!!!!!!!!!!!!!!!!!!!!!!!!!!!!!')

#print('#else#else#if#else#if#else',total)
                    #print('-->  antonis33')

                    #print('-------------')
```

```
                    total_temp_11 = total
                    total_temp_11.team = at
                    #print("total_temp: ", total_temp_11)
                    #print('-------------')
                    total = total.iloc[0:0]
                    #print("total1: ", total)
                    #print('-------------')
                    total = total.drop(columns=['mp'],
axis=1, errors='ignore')
                    #print("total3: ", total)
                    if len(total.columns)  == 19:
                        total.loc[0] = [1,ht, game_id, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
                    elif len(total.columns)  == 18:
                        total.loc[0] = [1,ht, game_id, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
total.drop(columns=['level_0'])
            #print('total: ',total)
            #print(total)
            #time.sleep(1000)
            #input()

        ndf = total.unstack().to_frame().T
        #print("@@@@@")
        #print(ndf)
        #print("@@@")
        #print()
        #print("ggggggggggggggggg")
        #print(ndf)
        ndf.columns =
ndf.columns.map('{0[0]}_{0[1]}'.format)
        ndf = ndf.drop(columns=['index_0', 'index_1',
'game_id_1'])

        ndf = ndf.rename(columns={"game_id_0":"game_id"})

##########################################################
        #print(ndf.iloc[0]['team_0'])
        #print(ht)
        if ndf.iloc[0]['team_0'] == ht:
            #print("_0 = home")
            ndf.columns = ndf.columns.str.replace("_0",
"_H")
            ndf.columns = ndf.columns.str.replace("_1",
"_R")
            #print(ndf)
        elif ndf.iloc[0]['team_0'] == at:
            #print("_0 = away")
            ndf.columns = ndf.columns.str.replace("_0",
"_R")
```

```
            ndf.columns = ndf.columns.str.replace("_1",
"_H")
        '''
        else:
            ndf.columns = ndf.columns.str.replace("_0",
"_H")
            ndf.columns = ndf.columns.str.replace("_1",
"_R")
        '''

##########################################################
        #print(ndf)
        #print("############################")
        #print(ndf)
        #print("############################")
        #print()
        return ndf




    drop_cols_1 = ['game_id', 'team_H', 'team_R']
    total_result.drop(drop_cols_1, axis=1, inplace=True)

    #print(real_result.dtypes)

    object_cols = [('F', 'T1'), ('F', 'T2'), ('FG', 'T1'),
('FG', 'T2'), ('FGA', 'T1'), ('FGA', 'T2'),
                   ('3P', 'T1'), ('3P', 'T2'), ('3PA',
'T1'), ('3PA', 'T2'), ('FT', 'T1'), ('FT', 'T2'),
                   ('FTA', 'T1'), ('FTA', 'T2'), ('OR',
'T1'), ('OR', 'T2'), ('DR', 'T1'), ('DR', 'T2'),
                   ('TOT', 'T1'), ('TOT', 'T2'), ('A',
'T1'), ('A', 'T2'), ('PF', 'T1'), ('PF', 'T2'),
                   ('ST', 'T1'), ('ST', 'T2'), ('TO',
'T1'), ('TO', 'T2'), ('BL', 'T1'), ('BL', 'T2')]

    real_result[object_cols] =
real_result[object_cols].astype(np.int64)


    drop_cols_2 = [('GAME-ID', ''), ('TEAM', 'T1'),
('TEAM', 'T2'), ('VENUE', 'T1'), ('VENUE', 'T2')]
    real_result.drop(drop_cols_2, axis=1, inplace=True)




    reorder_cols_3 = ['pts_R','pts_H', 'fg_R','fg_H',
'fga_R', 'fga_H','3p_R', '3p_H',
```

```
                        '3pa_R', '3pa_H', 'ft_R', 'ft_H',
'fta_R', 'fta_H', 'orb_R','orb_H',
                         'drb_R','drb_H',  'trb_R', 'trb_H',
'ast_R','ast_H', 'pf_R','pf_H',
                        'stl_R','stl_H',   'tov_R', 'tov_H',
'blk_R','blk_H',]



    total_result = total_result[reorder_cols_3]
    total_result.columns = [''] *
len(total_result.columns)
    total_result.reset_index(level=0, inplace=True)
    total_result.drop('index', axis=1, inplace=True)
    #print(total_result)
    #print(list(total_result))


    #input()
    #real_result = total_result.copy()
    #real_result['fg_H'] = 1

def scoreValidation(result):

    result.loc['Total']= result.sum()
    total_result = result.tail(1)
    result.drop(result.tail(1).index,inplace=True)

    #fix 3 first columns
    total_result['game_id'] = result['game_id'].iloc[0]
    total_result['team_H'] = result['team_H'].iloc[0]
    total_result['team_R'] = result['team_R'].iloc[0]

    #print(total_result.dtypes)
    #real_result = teamStatsDF['GAME-ID'] ==
result['game_id'].iloc[0]
    #array = ['GAME-ID', 'green']
    game_id_int = int(result['game_id'].iloc[0])
    real_result = teamStatsDF.loc[(teamStatsDF[('GAME-ID',
'')]== game_id_int)]

    reorder_cols_3 = ['pts_R','pts_H', 'fg_R','fg_H',
'fga_R', 'fga_H','3p_R', '3p_H',
                        '3pa_R', '3pa_H', 'ft_R', 'ft_H',
'fta_R', 'fta_H', 'orb_R','orb_H',
                         'drb_R','drb_H',  'trb_R', 'trb_H',
'ast_R','ast_H', 'pf_R','pf_H',
                        'stl_R','stl_H',   'tov_R', 'tov_H',
'blk_R','blk_H',]

    total_result = total_result[reorder_cols_3]
```

```
    #print(list(total_result))
    total_result =  total_result[['pts_R','pts_H']]
    total_result.columns = [''] *
len(total_result.columns)
    total_result.reset_index(level=0, inplace=True)
    total_result.drop('index', axis=1, inplace=True)
    #print(total_result)
    #print(list(total_result))


    #input()
    #real_result = total_result.copy()
    #real_result['fg_H'] = 1
    real_result =  real_result[[('F', 'T1'), ('F', 'T2')]]
    real_result.columns

def getGcsv(session,df,res,ht,at):
    print(len(df))
    #size = int(1/res)
    size=int(len(df)-1)
    df_list = np.array_split(df, size)
    index = 0
    result = pd.DataFrame()
    # check elements of df_list
    for df1 in df_list:
        #print(loadPbP(df1))
        temp=loadPbP(session,df1,ht,at)
        #index += 1
        #print(index,df1)
        #print('22222222222222222222')
        #print('2: ',temp)
        #print('22222222222222222222')
        #print()
        result = pd.concat([result, temp])
        #print('result:  ' ,result)

    result = result.fillna(0)
    result['team_H'] = np.where((result['team_H'] ==
0),ht,ht)
    result['team_H'] = np.where((result['team_H'] !=
ht),ht,ht)
    result['team_R'] = np.where((result['team_R'] ==
0),at,at)
    result['team_R'] = np.where((result['team_R'] !=
at),at,at)

    gameid = result['game_id'].iloc[0]

    #s = result[0]
    #s.append(result[1])
    #print(result)
```

```
    #s = pd.DataFrame(np.concatenate(result))
    #result.to_csv('game.csv', index=False)

    ### Conver float to int
    result[float64_cols] =
result[float64_cols].astype(np.int64)

    ###########disable / enable validation)
##########################
    #val = statsValidation(result)
    val = scoreValidation(result)
    #val="OK"

##########################################################
#######
    if val:
        check = "OK"
    else:
        check ="DIF"

    result.to_csv(f'{check}_{gameid}_{res}.csv', sep=',',
index=False)

def main():

    res = float(input("Please enter the resolution
(example 0.2):\n"))
    print(f'You entered resolution {res} %')
    print( 'Converting...' )

    csvlist = glob.glob("Data/test/*.csv")

    csvlist.sort()

def readTeamStats(path):
    #read Box_Score_Team-Stats
    #path = "Data/*Box_Score_Team-Stats.xlsx"
    teamStatsDF = pd.DataFrame()
    for file in glob.glob(path):
        print(file)
        # create a df of that file path
        df = pd.read_excel(file, sheet_name = 0)
        # appened it
        teamStatsDF = teamStatsDF.append(df)

    ### Conversation
    #drop cols
    drop_cols = ['DATASET',
'DATE','1Q','2Q','3Q','4Q','OT1','OT1','OT2','OT3',
```

```
    scoresDF_new = teamStatsDF.pivot(index='GAME-ID',
columns='by', values=['TEAM','VENUE','F', 'FG', 'FGA',
'3P', '3PA',
'OR', 'DR', 'TOT', 'A', 'PF', 'ST',
'TO', 'BL'])
    scoresDF_new.reset_index(level=0, inplace=True)

    #print(scoresDF_new )
    return scoresDF_new
if __name__ == '__main__':

    session = pd.read_excel('Data/players.xlsx')

    teamStatsDF = readTeamStats("Data2/*Box_Score_Team-
Stats.xlsx")

############################################################
####
    #print(teamStatsDF)
    #sys.exit()
    main()
```

Παράρτημα Γ: Πρόγραμμα Πρόβλεψης Αποτελεσμάτων

Στιγμιότυπα σχολιασμένου κώδικα Matlab

```
% This script file provides fundamental preprocessing
functionality for the
% NBA play-by-play datasets.

% This a slightly modified version of the
LSTM_NBA_Experiment_Ia.m script
% file which provides the possibility to use a different
sequence size for
% training and testing.

% An additional feature of this script file is that it
provides an
% implementation of the K-Fold cross validation.

clc
clear

% Set the location of the play-by-play .csv files for each
game.
data_directory = 'Features/res_0.01/';
% Set the location of the outcome .csv files for each
game.
labels_directory = 'Labels';

% Set the game specific variables.
game_variables = {'game_id','team_H','team_R'};
```

```
% Get the current folder.
current_directory = pwd;

% Compose the full path for the play-by-play .csv files.
full_data_directory =
fullfile(current_directory,data_directory);
% Compose the full path for the final outcome .csv files.
full_labels_directory  =
fullfile(current_directory,labels_directory);

% Create the NBA text datastore for the play-by-play
features.
nba_features_ds =
tabularTextDatastore(full_data_directory,...
                              'FileExtensions','.csv',...

'VariableNamingRule','preserve',...
                              'IncludeSubfolders', true);

% Create the NBA text datastore for the final outcome of
each game.
nba_labels_ds =
tabularTextDatastore(full_labels_directory,...

'VariableNamingRule','preserve',...
                              'FileExtensions','.csv',...
                              'IncludeSubfolders', true);

% Set the selected variable names to retrieve game
specific information.
nba_features_ds.SelectedVariableNames = game_variables;

% Set the percentage of sequence data per game that will
be used for
% training.
SequencePercentage = 1.00;

% Set the percentage of sequence data per game that will
be used in order
% to form the leader-based prediction.
LeaderSequencePercentage = 0.70;

% Set the selected variable names to retrieve play-by-play
specific
% information.
reset(nba_features_ds);
nba_features_ds.SelectedVariableNames = data_variables;

% Retrieve all label-based ds games.
labels_games = readall(nba_labels_ds);
```

```
% Set the label for each game that indicates its final
outcome.
FeatureLabels = zeros(GamesNumber,1);
% Set the label for each game according to the outcome of
the game at the
% percentage of the sequence that is used during training.
LeaderLabels = zeros(GamesNumber,1);

% Initialize a cell array for storing the sequence of
feature vectors for
% each game. Sequences of features will be stored in a
row-wise manner
% within each cell array matrix.
FeatureSequences = cell(GamesNumber,1);

% Initialize internal game index.
game_index = 1;

% Incrementally load the sequence of feature vectors for
each game.
    % Keep only the training percent of the sequence data
for each game.
    % Mind at this point that the number of points scored
by each team may
    % or may not be taken into consideration.
    FeatureSequences{game_index} =
FeatureSequences{game_index}(1:end,1:TrainingSequenceLengt
h);
    % Derive the outcome of the current game taking into
consideratation
    % the fact that the points of each team constitutes
the last feature.
    if(sum(home_sequence(:,end)) >
sum(road_sequence(:,end)))
        FeatureLabels(game_index) = 1;
    else
        FeatureLabels(game_index) = 2;
    end
    % Derive the outcome of the current game at the
training cuttoff
    % point.
    if(sum(home_sequence(1:LeadingSequenceLength,end)) >
sum(road_sequence(1:LeadingSequenceLength,end)))
        LeaderLabels(game_index) = 1;
    else
        LeaderLabels(game_index) = 2;
    end
    game_index = game_index + 1;
end
```

```
% Keep the original feature sequences.
OriginalFeatureSequences = FeatureSequences;

% Aggregate the elements of each sequence by computing the
% corresponding cumulative sum.
FeatureSequences = cellfun(@(C)
cumsum(C,2),FeatureSequences,'UniformOutput',false);

% THE FOLLOWING CODE LINE WILL HAVE TO BE EXCLUDED IN THE
FUTURE.
% Exclude the game points feature.
% FeatureSequences = cellfun(@(C) C(1:end-
1,:),FeatureSequences,'UniformOutput',false);

% Convert feature labels to categorical variables.
FeatureLabels = categorical(FeatureLabels);
% Convert leader labels to categorical variables.
LeaderLabels = categorical(LeaderLabels);

% Update the dimensionality of the input space.
Dimensionality = size(FeatureSequences{1},1);

% Set the number of cross validation folds.
FoldsNumber = 10;

% Set a vector of the cross validation indices.
cross_validation_indices =
crossvalind('Kfold',GamesNumber,FoldsNumber);

% Set the percentage of games utilized during training.
GamesPercentage = 0.90;
% Get the number of games to be used during training.
TrainingGamesNumber = round(GamesNumber *
GamesPercentage);

% Get the global sequence length.
SEQUENCE_LENGTH = unique(SequenceLength);
if (length(SEQUENCE_LENGTH) > 1)
    error('Game sequences are of the same length')

    % according to the prespecified parameter values.
    TrainingFeatureSequences{fold_index} = ...
    cellfun(@(C)
C(:,1:TrainingSequenceLength),TrainingFeatureSequences{fol
d_index},'UniformOutput',false);
    TestingFeatureSequences{fold_index} = ...
    cellfun(@(C)
C(:,1:TestingSequenceLength),TestingFeatureSequences{fold_
index},'UniformOutput',false);
end
```

```
% Set fundamental parameters for the LSTM Network.
HiddenUnitsNumber1 = 15;
HiddenUnitsNumber2 = 15;
HiddenUnitsNumber3 = 15;
ClassesNumber = 2;

% Set fundamental parameters for the training process.
MaxEpochs = 25;
MiniBatchSize = 50;
% Set the training options structure.
options = trainingOptions('adam', ...
    'ExecutionEnvironment','cpu', ...
    'GradientThreshold',1.00, ...
'MiniBatchSize',MiniBatchSize, ...

            'SequenceLength','longest');
    % Compute the training accuracy of the LSTM network
per fold.
    TrainingAccuracy(fold_index) =
sum(EstimatedTrainingFeatureLabels{fold_index} ==
TrainingFeatureLabels{fold_index}) ...


% Define the LSTM Network architecture.
layers = [ ...
    sequenceInputLayer(Dimensionality)
    lstmLayer(HiddenUnitsNumber1,'OutputMode','sequence');
    dropoutLayer(0.05);
    lstmLayer(HiddenUnitsNumber2,'OutputMode','sequence');
    dropoutLayer(0.05);
    lstmLayer(HiddenUnitsNumber3,'OutputMode','last');
    dropoutLayer(0.05);
    fullyConnectedLayer(ClassesNumber)
    softmaxLayer
    classificationLayer];

% Set fundamental parameters for the training process.
MaxEpochs = 25;
MiniBatchSize = 50;
% Set the training options structure.
options = trainingOptions('adam', ...
    'ExecutionEnvironment','cpu', ...
    'GradientThreshold',1.00, ...
'MiniBatchSize',MiniBatchSize, ...

            'SequenceLength','longest');
    % Compute the training accuracy of the LSTM network
per fold.
    TrainingAccuracy(fold_index) =
sum(EstimatedTrainingFeatureLabels{fold_index} ==
TrainingFeatureLabels{fold_index}) ...
```

```
                    /
numel(TrainingFeatureLabels{fold_index});
    % Compute the training accuracy according to the
leading score inference
    % strategy.
    LeadingTrainingAccuracy(fold_index) =
sum(TrainingLeaderLabels{fold_index}==TrainingFeatureLabel
s{fold_index})...


         / numel(TrainingFeatureLabels{fold_index});
    % Estimate the predicted labels during testing.
    EstimatedTestingFeatureLabels{fold_index} =
classify(lstm{fold_index},TestingFeatureSequences{fold_ind
ex},...



% Define the LSTM Network architecture.
layers = [ ...
    sequenceInputLayer(Dimensionality)
    lstmLayer(HiddenUnitsNumber1,'OutputMode','sequence');
    dropoutLayer(0.05);
    lstmLayer(HiddenUnitsNumber2,'OutputMode','sequence');
    dropoutLayer(0.05);
    lstmLayer(HiddenUnitsNumber3,'OutputMode','last');
    dropoutLayer(0.05);
    fullyConnectedLayer(ClassesNumber)
    softmaxLayer
    classificationLayer];


'SequenceLength','longest');
    % Compute the testing accuracy of the LSTM network.
    TestingAccuracy(fold_index) =
sum(EstimatedTestingFeatureLabels{fold_index} ==
TestingFeatureLabels{fold_index}) ...
                    /
numel(TestingFeatureLabels{fold_index});
    % Compute the testing accuracy according to the
leading score inference
    % strategy.
    LeadingTestingAccuracy(fold_index) =
sum(TestingLeaderLabels{fold_index}==TestingFeatureLabels{
fold_index})...
                   /
numel(TestingFeatureLabels{fold_index});
end
```