ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Οδηγικού Προφίλ Χρηστών μέσω Υβριδικής Πλατφόρμας Δεδομένων Μεγάλης Κλίμακας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΣΕΡΑΦΕΙΜ ΠΑΝΑΓΙΩΤΙΔΗ

Επιβλέπων: Ευστάθιος Δ. Συκάς
Καθηγητής Ε.Μ.Π.

Αθήνα, 30 Ιουνίου 2021

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΕΠΙΚΟΙΝΩΝΙΩΝ, ΗΛΕΚΤΡΟΝΙΚΗΣ ΚΑΙ ΣΥΣΤΗΜΑΤΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

Ανάλυση Οδηγικού Προφίλ Χρηστών μέσω Υβριδικής Πλατφόρμας Δεδομένων Μεγάλης Κλίμακας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΣΕΡΑΦΕΙΜ ΠΑΝΑΓΙΩΤΙΔΗ

Επιβλέπων: Ευστάθιος Δ. Συκάς
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30η Ιουνίου 2021

.....	.....	.....
Ε. Συκάς	Ι. Ρουσσάκη	Ν. Μήτρου
Καθηγητής Ε.Μ.Π.	Επ. Καθηγήτρια Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, 30 Ιουνίου 2021

.....

Σεραφείμ Παναγιωτίδης

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη ενός συστήματος ανάλυσης και εξαγωγής οδηγικών προφίλ, τα οποία προκύπτουν από οδηγικά δεδομένα που συλλέγονται και αποθηκεύονται σε πλατφόρμα δεδομένων μεγάλης κλίμακας (Big Data Platform), με εφαρμογή σύγχρονων μεθόδων μηχανικής μάθησης (Machine Learning), στα πλαίσια ανάπτυξης των ευφυών συστημάτων μεταφορών (Intelligent Transportation Systems - ITS). Η ταχεία εξέλιξη της τεχνολογίας στον τομέα των έξυπνα διασυνδεδεμένων συσκευών σε συνδυασμό με τις συνεχώς αυξανόμενες απαιτήσεις της κοινωνίας για ασφαλή,γρήγορη και εύκολη μετακίνηση έχουν αναπόφευκτα οδηγήσει στην ανάγκη ανάπτυξης αποδοτικότερων και αποτελεσματικότερων μηχανισμών μεταφοράς. Σε αυτό το πλαίσιο, το σύστημα που προτείνεται προσφέρει τη δυνατότητα λήψης και επεξεργασίας κυκλοφοριακών δεδομένων σε πραγματικό χρόνο καθώς και την ιστορική αναδρομή αυτών, με σκοπό την αξιολόγηση της οδηγικής συμπεριφοράς των χρηστών που έχουν επιλέξει να συμμετέχουν σε αυτό. Αρχικά, μελετάται η αποδοτική αποθήκευση του μεγάλου όγκου των κυκλοφοριακών δεδομένων που συλλέγονται με σκοπό να εξασφαλίζεται η εύκολη και γρήγορη αναζήτησή τους. Τις προϋποθέσεις αυτές πληρούν τα NoSQL συστήματα, όπως το MongoDB, που είναι μια από τις πιο δημοφιλείς κατανεμημένες βάσεις και εξασφαλίζει την αποτελεσματική διαχείριση δεδομένων μεγάλου όγκου. Στη συνέχεια, τα δεδομένα υποβάλλονται σε μηχανισμούς προεπεξεργασίας και ανάλυσης, μέσω κατάλληλων frameworks, όπως είναι τα Apache Kafka και Apache Spark, προκειμένου να μετασχηματιστούν σε μορφή που εξυπηρετεί την αποτελεσματική εξαγωγή συμπερασμάτων από αυτά. Η ανάπτυξη μεθόδων για την εξαγωγή συμπερασμάτων από τα οδηγικά δεδομένα είναι ιδιαίτερα σημαντική για την αξιολόγηση και την κατηγοριοποίηση της οδηγικής συμπεριφοράς. Για τον λόγο αυτό, αναζητούνται τα αποτελεσματικότερα μοντέλα ανάλυσης και εξαγωγής δεδομένων με σκοπό την δημιουργία αξιόπιστων οδηγικών προφίλ (Driver Profiling). Στη συνέχεια, αξιολογούνται και συγκρίνονται τα αποτελέσματα των μοντέλων που εξετάστηκαν με σκοπό την επιλογή της αποδοτικότερης μεθόδου ανάλυσης της εξαγωγής οδηγικών προφίλ, με βάση τα κυκλοφοριακά δεδομένα που χρησιμοποιήθηκαν. Τελος, εξασφαλίζεται η αποτελεσματική ανάκτηση και αναπαράσταση των οδηγικών προφίλ που έχουν σχηματιστεί, με χρήση κατάλληλων εργαλείων, όπως είναι τα PostgreSQL και Vue.js framework, και τεχνολογιών, όπως είναι το REST API, με σκοπό την εύκολη και αποτελεσματική αξιολόγηση της οδηγικής συμπεριφοράς των χρηστών του συστήματος.

Λέξεις κλειδιά: Κατηγοριοποίηση Οδηγικής Συμπεριφοράς (Driver Profiling), Ευφυή Συστήματα Μεταφορών (Intelligent Transportation Systems - ITS), Μεγάλα Δεδομένα (Big Data), Μηχανική Μάθηση (Machine Learning), Κατηγοριοποίηση Πολυδιάστατων Χρονοσειρών (Multivariate Timeseries Classification), MongoDB, Apache Kafka, Apache Spark, PostgreSQL, Vue.js, REST API, k-Nearest Neighbors, Μηχανές Διανυσμάτων Υποστύριξης (Support Vector Machines - SVM), Τυχαία Δάση (Random Forest), XGBoost, Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Echo State Networks (ESN), Timeseries Shapelets, Dynamic Time Warping (DTW), Symbolic Aggregate Approximation (SAX)

Abstract

The purpose of this dissertation is to develop a system in order to analyse and export driving profiles, which result from driving data collected and stored on big data platform, using modern machine learning methods, in the context of the development of intelligent transport systems (ITS). The rapid evolution of technology in the field of smart interconnected devices combined with the ever-increasing demands of society for safe, fast and easy transportation have inevitably led to the need to develop more efficient and effective transport mechanisms. In this context, the proposed system offers the ability to receive and process traffic data in real time as well as their historical background, in order to evaluate the driving behavior of users who have chosen to participate in it. Initially, this dissertation studies the efficient storage of the large volume of traffic data collected in order to ensure their easy and fast retrieval. These conditions are met by NoSQL systems, such as MongoDB, which is one of the most popular distributed databases and ensures efficient management of large volumes of data. The data is then subjected to preprocessing and analysis mechanisms, through appropriate frameworks, such as Apache Kafka and Apache Spark, in order to be transformed into a format that serves to effectively draw conclusions from them. The development of methods for drawing conclusions from driving data is particularly important for the evaluation and categorization of driving behavior. For this reason, the most efficient models of data analysis and data exportation are sought in order to create reliable driving profiles (Driver Profiling). Then, the results of the models examined are evaluated and compared in order to select the most efficient method of analysis of the formation of driving profiles, based on the traffic data used. Finally, the effective retrieval and representation of the driving profiles that have been formed is ensured by using appropriate tools, such as PostgreSQL and Vue.js, and technologies, such as REST API, in order to easily and effectively evaluate the driving behavior of users of the system.

Key words: Driver Profiling, Intelligent Transportation Systems (ITS), Big Data, Machine Learning, Multivariate Timeseries Classification, MongoDB, Apache Kafka, Apache Spark, PostgreSQL, Vue.js, REST API, k-Nearest Neighbors, Support Vector Machines (SVM), Random Forest, XGBoost, Multilayer Perceptron (MLP), Long Short-Term Memory (LSTM), Echo State Networks (ESN), Timeseries Shapelets, Dynamic Time Warping (DTW), Symbolic Aggregate Approximation (SAX)

Ευχαριστίες

Με την ολοκήρωση της παρούσας διπλωματικής εργασίας θα ήθελα να ευχαριστήσω όλους όσους με βοήθησαν κατά την εκπόνησή της. Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή κ. Ευστάθιο Συκά για την εμπιστοσύνη που μου έδειξε με την ανάθεση της παρούσας εργασίας καθώς και για τις συμβουλές του και τη στήριξή του. Επιπλέον, θα ήθελα να ευχαριστήσω τον Θεόδωρο Αλεξάκη και τον Νικόλαο Πεππέ για την εξαιρετική συνεργασία που είχαμε, για την καθοδήγηση που μου παρείχαν και την συνολικότερη συμβολή τους στην ολοκλήρωση αυτής της διπλωματικής εργασίας. Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους μου, που με στήριξαν καθ' όλη τη διάρκεια των σπουδών μου.

Περιεχόμενα

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1: Εισαγωγή

1.1 Περιγραφή του Προβλήματος

Η ταχεία εξέλιξη της τεχνολογίας στον τομέα των έξυπνα διασυνδεδεμένων συσκευών σε συνδυασμό με τις συνεχώς αυξανόμενες απαιτήσεις της κοινωνίας για ασφαλή, γρήγορη και εύκολη μετακίνηση έχουν αναπόφευκτα οδηγήσει στην ανάγκη ανάπτυξης αποδοτικότερων και αποτελεσματικότερων μηχανισμών μεταφοράς. Το πρόβλημα των μεταφορών δεν εντοπίζεται μόνο στην εξασφάλιση της ασφαλούς μετακίνησης προσώπων και αντικειμένων, αλλά και στις επιπτώσεις που έχει η συνολικότερη λειτουργία του κυκλοφοριακού συστήματος τόσο στην οικονομία όσο και στο φυσικό περιβάλλον. Πιο συγκεκριμένα, η οδηγική συμπεριφορά είναι ένας παράγοντας ο οποίος έχει μεγάλο αντίκτυπο στην οδική ασφάλεια, τις εκπομπές βλαβερών αερίων και την ποσότητα καυσίμου που απαιτείται για τις μετακινήσεις. Σύμφωνα, με έρευνες που έχουν πραγματοποιηθεί από τον παγκόσμιο οργανισμό υγείας (World Health Organization) υπολογίζεται ότι περίπου 1350000 άνθρωποι χάνουν την ζωή τους ετησίως σε οδικά τροχαία ατυχήματα [1]. Επιπλέον, τα συστήματα μεταφοράς είναι μια από τις μεγαλύτερες πηγές εκπομπής διοξειδίου του άνθρακα (σε παγκόσμιο επίπεδο περίπου το 18% του διοξειδίου του άνθρακα προέρχεται από τα οδικά μέσα μεταφοράς) [2].

Για την αντιμετώπιση των παραπάνω προκλήσεων η ερευνητική κοινότητα έχει στρέψει το ενδιαφέρον της στα ευφυή συστήματα μεταφορών, τα οποία με χρήση νέων τεχνολογιών και καινοτομιών στοχεύουν στη βελτίωση του συστήματος μεταφορών. Στο πλαίσιο αυτό, οι μεγάλες βιομηχανίες αυτοκινήτων παράγουν όλο και περισσότερα οχήματα τα οποία είναι εξοπλισμένα με ειδικούς αισθητήρες που συλλέγουν δεδομένα σχετικά με τη κίνηση του οχήματος. Η ανάλυση των δεδομένων αυτών συμβάλλει άμεσα στην κατανόηση τόσο της συμπεριφοράς του οδηγού όσο και της κατάστασης του οχήματος και επομένως αποτελεί βασικό παράγοντα για τη συνολικότερη βελτίωση του οδικού κυκλοφοριακού συστήματος. Είναι προφανές ότι η σωστή οδηγική συμπεριφορά μπορεί να οδηγήσει όχι μόνο στην αποτροπή και τον περιορισμό των τροχαίων ατυχημάτων, αλλά και στη χαμηλότερη κατανάλωση καυσίμου.

Ωστόσο, η ανάλυση της οδηγικής συμπεριφοράς με στόχο τη μετατόπισή της προς μια σωστότερη κατεύθυνση δεν είναι μια απλή διαδικασία. Ο τεράστιος όγκος των δεδομένων και η ταχύτητα με την οποία αυτά συλλέγονται από τους αισθητήρες του οχήματος απαιτούν τη χρήση έξυπνων και αποδοτικών μεθόδων για τη διαχείρισή τους. Ζητούμενο είναι η ανάλυση των δεδομένων σε πραγματικό χρόνο και η εξαγωγή συμπερασμάτων για τον εκάστοτε οδηγό, προκειμένου να εξασφαλίζεται πάντα η εγγύηση ότι τα οχήματα οδηγούνται σύμφωνα με τις προτεινόμενες συστάσεις οδήγησης. Πρόκειται για μια συνεχή διαδικασία κατά την οποία κυκλοφοριακά δεδομένα μεγάλης κλίμακας θα λαμβάνονται με αδιάκοπο ρυθμό και στη συνέχεια θα επεξεργάζονται σε σχεδόν μηδενικό χρόνο, έτσι ώστε να ληφθεί άμεσα το αποτέλεσμα της ανάλυσης τους.

Συνοψίζοντας, μπορεί κανείς εύκολα να καταλάβει την αναγκαιότητα ύπαρξης σύγχρονων τεχνολογιών και μεθόδων οι οποίες θα έχουν πρόσβαση σε δεδομένα που συλλέγονται από κάποιο όχημα, θα είναι ικανές να διαχειριστούν και να αναλύσουν τον τεράστιο όγκο πληροφορίας που λαμβάνουν σε πραγματικό χρόνο και τελικά θα εξάγουν συμπεράσματα τα οποία μπορούν να βοηθήσουν τον οδηγό να αναπτύξει μια πιο ασφαλή, οικονομική και οικολογική συμπεριφορά.

1.2 Συνεισφορά Εργασίας

Στην παρούσα διπλωματική εργασία προτείνεται ένα σύστημα ανάλυσης και εξαγωγής οδηγικών προφίλ, τα οποία προκύπτουν από οδηγικά δεδομένα που συλλέγονται και αποθηκεύονται σε πλατφόρμα δεδομένων μεγάλης κλίμακας, με εφαρμογή σύγχρονων μεθόδων μηχανικής μάθησης, στα πλαίσια ανάπτυξης των ευφυών συστημάτων μεταφορών. Το σύστημα που υλοποιήθηκε προσφέρει τη δυνατότητα λήψης και επεξεργασίας των δεδομένων σε πραγματικό χρόνο καθώς και την ιστορική αναδρομή αυτών, με σκοπό την αξιολόγηση της οδηγικής συμπεριφοράς των χρηστών που έχουν επιλέξει να συμμετέχουν σε αυτό. Η υλοποίηση του συστήματος βασίζεται σε πληθώρα προγραμματιστικών εργαλείων τα οποία εξειδικεύονται στην αποθήκευση, στην διαχείριση και στην επεξεργασία δεδομένων μεγάλου όγκου, όπως είναι τα Apache Spark, Apache Kafka και MongoDB, καθώς και στις πλέον σύγχρονες τεχνολογίες ανάπτυξης εφαρμογών, όπως είναι η αρχιτεκτονική REST, το Vue framework και η σχεσιακή βάση δεδομένων PostgreSQL.

Πιο συγκεκριμένα, στην παρούσα διπλωματική εργασία μελετάται η πορεία που ακολουθούν τα οδηγικά δεδομένα από την συλλογή τους από τα οχήματα με χρήση των κατάλληλων αισθητήρων, την αποδοτική τους αποθήκευσή σε κατάλληλη πλατφόρμα δεδομένων μεγάλου όγκου, την ανάκτησή τους σε πραγματικό χρόνο, την προεπεξεργασία και τον μετασχηματισμό τους σε μορφή που εξυπηρετεί τις ανάγκες ανάλυσης τους με παράλληλο και κατανεμημένο τρόπο με σκοπό την εξαγωγή και την αποθήκευση των συμπερασμάτων που προκύπτουν από αυτά και τελικά μέχρι την αποδοτική ανάκτηση και αναπαράσταση της παραγόμενης γνώσης. Συνεχίζοντας, για την αξιοποίηση των οδηγικών δεδομένων και την εξαγωγή συμπερασμάτων από αυτά παρουσιάζονται, εξετάζονται και συγκρίνονται διαφορετικοί συνδυασμοί μεθόδων και μοντέλων ανάλυσης και εξαγωγής δεδομένων με χρήση αλγορίθμων μηχανικής μάθησης, με σκοπό την εύρεση και την επιλογή της αποδοτικότερης μεθόδου ανάλυσης και εξαγωγής οδηγικών προφίλ.

Κλείνοντας, παρουσιάζεται σε ένα θεωρητικό επίπεδο η έννοια των ευφυών συστημάτων μεταφορών, στο πλαίσιο των οποίων εντάσσεται η λειτουργία του συστήματος που υλοποιήθηκε, καθώς και τα οφέλη που προκύπτουν από την ένταξη τους στην σύγχρονη κοινωνία και καθημερινότητα. Επιπλέον, ιδιαίτερη έμφαση δίνεται στην ανάλυση της σημασίας τους όρου των μεγάλων δεδομένων καθώς και στις τεχνικές διαχείρισης και αξιοποίησης τους. Τέλος, παρουσιάζεται αναλυτικά το μαθηματικό θεωρητικό υπόβαθρο που είναι απαραίτητο για την κατανόηση των μεθόδων και των μοντέλων μηχανικής μάθησης που εξετάστηκαν.

1.3 Οργάνωση της Εργασίας

Στην παρούσα διπλωματική εργασία περιλαμβάνονται τα παρακάτω κεφάλαια:

- Στο κεφάλαιο 2 παρουσιάζεται και αναλύεται η λειτουργία των ευφυών συστημάτων μεταφορών καθώς και τα οφέλη που μπορούν να προκύψουν από την ενσωμάτωσή τους στη σύγχρονη πραγματικότητα.

- Στο κεφάλαιο 3 παρουσιάζεται και αναλύεται η έννοια των μεγάλων δεδομένων, ενώ ιδιαίτερη έμφαση δίνεται στις τεχνικές της αποδοτικής αποθήκευσης τους, στην προεπεξεργασία και το μετασχηματισμό τους καθώς και στην εξόρυξη αξιοποιήσιμης πληροφορίας από αυτά.

- Στο κεφάλαιο 4 παρουσιάζονται θεωρητικές γνώσεις που είναι απαραίτητες για την κατανόηση του συστήματος που υλοποιήθηκε και οι οποίες σχετίζονται κυρίως με την κατανόηση του κλάδου της μηχανικής μάθησης και τη διαχείριση χρονοσειρών.

- Στο κεφάλαιο 5 παρουσιάζεται το σύνολο δεδομένων που χρησιμοποιήθηκε για την εκπόνηση της παρούσης διπλωματικής εργασίας.

- Στο κεφάλαιο 6 αξιολογούνται και συγκρίνονται τα αποτελέσματα των μοντέλων μηχανικής μάθησης που εξετάστηκαν με σκοπό την εξαγωγή συμπερασμάτων από τα κυκλοφοριακά δεδομένα που χρησιμοποιήθηκαν.

- Στο κεφάλαιο 7 παρουσιάζεται το σύστημα που υλοποιήθηκε καθώς και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξή του.

- Στο κεφάλαιο 8 παρουσιάζονται τα συμπεράσματα που προέκυψαν από την εκπλήρωση της παρούσης διπλωματικής εργασίας καθώς και μελλοντικές προεκτάσεις αυτής.

Κεφάλαιο 2: Ευφυή Συστήματα Μεταφορών

2.1 Ευφυή Συστήματα Μεταφορών (Intelligent Transportation Systems - ITS)

Τα ευφυή συστήματα μεταφορών (Intelligent Transportation Systems - ITS) συνιστούν ένα σύνολο τεχνολογιών πληροφόρησης και επικοινωνίας που μπορούν να βελτιώσουν τον τρόπο με τον οποίο τα οχήματα αλληλεπιδρούν τόσο μεταξύ τους όσο και με την οδική υποδομή. Η ενσωμάτωση των ITS στη σύγχρονη πραγματικότητα αναμένεται να αυξήσει την αποτελεσματικότητα χρήσης του οδικού δικτύου καθώς και να ενισχύσει την οδική ασφάλεια και τις περιβαλλοντικές επιδόσεις των οχημάτων. Τα ITS βασίζονται στις ασύρματες αυτοματοποιημένες επικοινωνίες προκειμένου να εξάγουν, να αναλύσουν και να επεξεργαστούν δεδομένα σχετικά με τα οχήματα, τους επιβάτες, το οδικό δίκτυο και τους πεζούς σε πραγματικό χρόνο. Αυτή η παρακολούθηση σε πραγματικό χρόνο επιτρέπει τη διάγνωση και αξιολόγηση περιστατικών με σκοπό την ενημέρωση του χρήστη του οχήματος ή ακόμη και την εκτέλεση δράσεων σε πολλές περιπτώσεις από το ίδιο το σύστημα. Για παράδειγμα, οι σύγχρονες βιομηχανίες κατασκευής αυτοκινήτων στρέφουν την προσοχή τους όλο και περισσότερο σε ευφυή συστήματα που προσφέρουν δυνατότητες σχετικές με την εύρεση της βέλτιστης διαδρομής συναρτήσει του χρόνου και της απόστασης, την αποφυγή κυκλοφοριακής συμφόρησης, την αποφυγή/πρόληψη συγκρούσεων, την τήρηση του ορίου ταχύτητας κ.α. . Στα παρακάτω υποκεφάλαια θα παρουσιαστούν αναλυτικότερα τα βασικά συστατικά που απαρτίζουν ένα έξυπνο σύστημα μεταφορών, τα οποία είναι τα έξυπνα οχήματα, τα έξυπνα συστήματα αυτοκινητόδρομων καθώς και ο τρόπος με τον οποίο αυτά αλληλεπιδρούν μεταξύ τους.

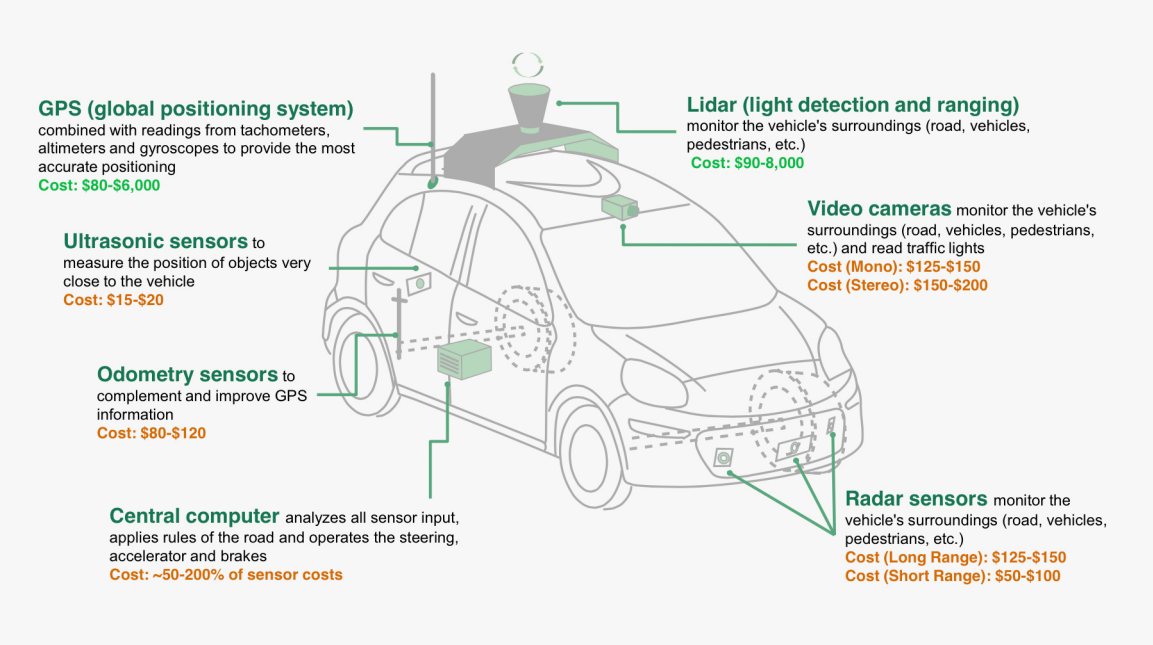
2.2 Έξυπνα Οχήματα (Intelligent Vehicles)

Τα έξυπνα οχήματα (Intelligent Vehicles) αποτελούν οχήματα που διαθέτουν πληθώρα συστημάτων για τη συλλογή δεδομένων και προσφέρουν τη δυνατότητα εκτέλεσης ενεργειών, τις οποίες θα πραγματοποιούσε και ο οδηγός, με αυτοματοποιημένο τρόπο. Σε αυτές τις ενέργειες περιλαμβάνονται η ασφαλής κίνηση του οχήματος με σεβασμό στις λωρίδες, η αποφυγή εμποδίων, η αποφυγή της κυκλοφοριακής συμφόρησης, η επιλογή της βέλτιστης διαδρομής, η αξιολόγηση και αποφυγή επικίνδυνων καταστάσεων, η μετακίνηση με σεβασμό στο περιβάλλον κ.α. . Η εκτέλεση των παραπάνω ενεργειών με αυτοματοποιημένο τρόπο γίνεται δυνατή με τη χρήση συσκευών και αισθητήρων (κάμερες, σόναρ, ραντάρ κ.α.) που είναι εγκατεστημένα πάνω στο εν λόγω όχημα. Μέσω των αισθητήρων αυτών συλλέγονται και αξιολογούνται σε πραγματικό χρόνο δεδομένα σχετικά με:

- τη θέση και τη δυναμική-κινηματική κατάσταση του οχήματος
- την κατάσταση του περιβάλλοντος

- την κατάσταση του οδηγού και των υπόλοιπων επιβατών

Τα δεδομένα αυτά επίσης μπορεί να ανταλλάσσονται και επομένως να αποκτώνται κιόλας, από τα υπόλοιπα οχήματα στον γύρω χώρο καθώς και από το οδικό δίκτυο (όπως παρουσιάζεται και παρακάτω στο υποκεφάλαιο 2.3), αν έχουν εγκατασταθεί οι κατάλληλοι μηχανισμοί και πρωτόκολλα τα οποία εξασφαλίζουν την επικοινωνία μεταξύ τους.



Πηγή: `https://www.wired.com/wp-content/uploads/2015/04/sensor_info.jpg`

Σχήμα 2.1: Έξυπνο όχημα εξοπλισμένο με αισθητήρες για αναγνώριση του περιβάλλοντος

2.2.1 Θέση και Δυναμική-Κινηματική Κατάσταση του Οχήματος

Η συλλογή και η επεξεργασία δεδομένων που αφορούν τη θέση και την κατάσταση του οχήματος είναι σημεία κλειδιά για την ανάπτυξη έξυπνων συστημάτων μεταφοράς. Η μελέτη αυτών των δεδομένων επιτρέπει την πρόβλεψη της κίνησης του οχήματος στον χώρο και συνεπώς καθιστά δυνατό τον αυτοματοποιημένο έλεγχο αυτού [3]. Πιο συγκεκριμένα:

- Θέση του οχήματος: Γνωρίζοντας την ακριβή θέση (γεωγραφικό μήκος και πλάτος, ύψος κ.α.) του οχήματος καθίσταται δυνατή η κίνηση του οχήματος στο οδικό δίκτυο με αυτοματοποιημένο τρόπο καθώς και η επιλογή των διαδρομών που αυτό δύναται να ακολουθήσει. Προκειμένου να λειτουργήσει αυτόνομα, το ίδιο το έξυπνο όχημα θα πρέπει να είναι ικανό να αντιληφθεί και να ορίσει τη θέση που αυτό έχει μέσα στο περιβάλλον στο οποίο βρίσκεται. Για το σκοπό αυτό είναι απαραίτητη η χρήση αισθητήρων, GPS (Global Positioning System) και ειδικών αλγορίθμων SLAM (Simultaneous Localization and Mapping). Οι αλγόριθμοι SLAM προσπαθούν να αναπαραστήσουν και να χαρτογραφήσουν ένα άγνωστο περιβάλλον (mapping), έτσι ώστε ανά πάσα στιγμή να μπορούν να ανιχνεύσουν τη θέση ή την κίνηση του οχήματος (localization). Η διαδικασία

αυτή οδηγεί στη δημιουργία ενός ψηφιακού χάρτη, ο οποίος αποθηκεύεται τοπικά στο όχημα, και περιέχει πληροφορίες για το χώρο που το περιβάλλει. Η ανάγνωση του χάρτη αυτού επιτρέπει στο έξυπνο όχημα να κατανοήσει το περιβάλλον του και να εκτελέσει τις ανάλογες ενέργειες και κινήσεις. Για τη δημιουργία του ψηφιακού χάρτη χρησιμοποιούνται δεδομένα που συλλέγονται μέσω GPS και αισθητήρων. Τα δεδομένα που συλλέγονται μέσω GPS αφορούν κυρίως θέματα εύρεσης διαδρομών που μπορούν να ακολουθηθούν, ενώ αυτά που συλλέγονται μέσω αισθητήρων στοχεύουν στην ανίχνευση πιθανών εμποδίων. Στη συνέχεια, τα δεδομένα αυτά τροφοδοτούνται ως είσοδος σε ένα φίλτρο EKF (Extended Kalman Filter), μέσω του οποίου γίνεται μια αρκετά ακριβής εκτίμηση για την κατάσταση-θέση του οχήματος ως προς τον χώρο στον οποίο βρίσκεται [4].

- Δυναμική-κινηματική κατάσταση του οχήματος: Η γνώση της δυναμικής και της κινηματικής κατάστασης του οχήματος είναι απαραίτητη για ενέργειες όπως αποφυγή ή πρόβλεψη συγκρούσεων, έλεγχος πεταλιών (γκάζι, φρένο) και τιμονιού, οικονομική κατανάλωση καυσίμου, αυτοματοποιημένη εναλλαγή λωρίδων κ.α. . Η κατάσταση του οχήματος μπορεί να ερμηνευθεί χρησιμοποιώντας είτε το δυναμικό είτε το κινηματικό μοντέλο. Σε κάθε μία από τις δύο αυτές περιπτώσεις, η κατάσταση του οχήματος δίνεται από ένα σύνολο προκαθορισμένων μαθηματικών εξισώσεων [5]. Οι εξισώσεις αυτές μπορούν να χρησιμοποιηθούν ως είσοδος σε ένα μοντέλο MPC (Model Predictive Control), προκειμένου να υπάρξει πλήρως αυτοματοποιημένος έλεγχος του οχήματος [6]. Το MPC είναι ένας αλγόριθμος ελέγχου που λειτουργεί με ανατροφοδότηση και χρησιμοποιεί ένα μοντέλο προκειμένου να κάνει προβλέψεις για μελλοντικές τιμές παραμέτρων που ορίζουν μια διαδικασία-κατάσταση. Ουσιαστικά, το μοντέλο MPC προσπαθεί να επιλύσει ένα πρόβλημα βελτιστοποίησης, όπως είναι αυτό που ορίζεται από τις εξισώσεις που δίνουν την κινηματική ή τη δυναμική κατάσταση του αμαξιού, βρίσκοντας τις βέλτιστες τιμές των παραμέτρων. Η διαδικασία αυτή πραγματοποιείται επαναληπτικά με κάποια ορισμένη συχνότητα. Συνεπώς, ανά πάσα στιγμή καθίσταται δυνατός ο αυτοματοποιημένος έλεγχος των βασικών ενεργειών ενός αυτοκινήτου, όπως καθορισμός ταχύτητας, διατήρηση λωρίδας, έλεγχος τιμονιού κ.α. .

2.2.2 Κατάσταση του Περιβάλλοντος

Ένα από τα πιο σημαντικά χαρακτηριστικά ενός έξυπνου οχήματος είναι η ικανότητα αυτού να μπορεί να αντιληφθεί με αποδοτικό και ρεαλιστικό τρόπο τον χώρο που το περιβάλλει. Με τη χρήση αισθητήρων το όχημα είναι σε θέση να αντιληφθεί τον δρόμο, τις λωρίδες, άλλα οχήματα, πεζούς, φανάρια, πινακίδες, εμπόδια κ.α., και να προσαρμόσει αναλόγως την πορεία που πρέπει να ακολουθήσει ή να υπολογίσει τις ενέργειες που πρέπει να εκτελέσει. Στη συνέχεια, παρουσιάζονται συνοπτικά οι επιμέρους διεργασίες στις οποίες πρέπει να προβεί το έξυπνο όχημα, προκειμένου να σχηματίσει μια ρεαλιστική απεικόνιση του περιβάλλοντός του [3].

- Εντοπισμός δρόμου/λωρίδων: Για το σκοπό αυτό, συνήθως, χρησιμοποιούνται κάμερες οι οποίες παρακολουθούν τις γραμμές που ορίζουν τη λωρίδα που βρίσκεται το όχημα.

- Εντοπισμός και αναγνώριση σήμανσης: Για τον εντοπισμό των πινακίδων σήμανσης, συνήθως, χρησιμοποιούνται κάμερες. Στη συνέχεια, η αναγνώριση

αυτών στηρίζεται σε νευρωνικά δίκτυα και αλγορίθμους τεχνητής νοημοσύνης σχετικούς με αναγνώριση εικόνων.

- Αναγνώριση φωτεινού σηματοδότη: Στηρίζεται σε αλγορίθμους color pattern matching προκειμένου να προσδιοριστεί η ένδειξη του φωτεινού σηματοδότη. Σε αυτήν την περίπτωση πρέπει να δοθεί ιδιαίτερη έμφαση και να εξακριβωθεί αν ο φωτεινός σηματοδότης που εντοπίστηκε και αναγνωρίστηκε, όντως απευθύνεται στο εν λόγω όχημα.

- Αναγνώριση άλλων οχημάτων: Για το σκοπό αυτό χρησιμοποιούνται διάφορα είδη αισθητήρων όπως κάμερες, σόναρ, ραντάρ κ.α. . Λόγω δυσκολιών που μπορεί να προκύψουν από φυσικούς ή περιβαλλοντικούς παράγοντες η χρήση μόνο κάμερας καθίσταται αναξιόπιστη. Η πιο συνηθισμένη και αποτελεσματική προσέγγιση είναι η συνδυαστική χρήση κάμερας και ραντάρ.

- Αναγνώριση πεζών: Πρόκειται για μια από τις πιο απαιτητικές διεργασίες που καλείται να εκτελέσει ένα έξυπνο όχημα. Αυτό συμβαίνει διότι οι πεζοί χαρακτηρίζονται από έντονη κινητικότητα και τα χαρακτηριστικά τους είναι αρκετά δύσκολο να εντοπιστούν και να αναγνωριστούν από αλγορίθμους μηχανικής μάθησης μέσω κάμερας. Για τον λόγο αυτό, η χρήση καμερών θερμικής απεικόνισης φαίνεται να αποτελεί την πιο υποσχόμενη μέθοδο. Ωστόσο, θα πρέπει να επιλυθούν προβλήματα όπως η χρήση αυτών σε περιβάλλοντα υψηλής θερμοκρασίας.

2.2.3 Κατάσταση του Οδηγού και των Υπόλοιπων Επιβατών

Ένα έξυπνο όχημα θα πρέπει να μπορεί να παρακολουθεί και να αξιολογεί την κατάσταση του οδηγού προκειμένου να αναλάβει δράση, αν χρειαστεί, ή να προειδοποιήσει τον οδηγό για πιθανούς κινδύνους. Για παράδειγμα, μέσω κάμερας μπορεί να παρακολουθείται η κατάσταση του οδηγού και στη συνέχεια να υπολογίζονται ο βαθμός κόπωσης και προσήλωσης του, τα οποία θα αξιολογούνται παρατηρώντας και εξάγοντας δεδομένα από τα μάτια του και γενικότερα από το πρόσωπό του. Σε όσες περιπτώσεις κρίνεται απαραίτητο ο οδηγός μπορεί να τροφοδοτείται με αντίστοιχο μήνυμα προειδοποίησης, είτε οπτικό είτε ακουστικό.

2.2.4 Προηγμένα Συστήματα Υποβοήθησης του Οδηγού

Η χρήση έξυπνων οχημάτων έχει στόχο, σε ένα πρώτο στάδιο, να παίξει βοηθητικό ρόλο δίπλα στον οδηγό. Αυτό σημαίνει ότι τα δεδομένα που συλλέγονται από τους αισθητήρες χρησιμοποιούνται για την αξιολόγηση καταστάσεων και εξαγωγή αποτελεσμάτων χωρίς όμως την πραγματοποίηση κάποιας ενέργειας. Σε αυτήν την περίπτωση το έξυπνο όχημα προειδοποιεί για πιθανές επικίνδυνες καταστάσεις και γενικότερα δίνει συμβουλές στον οδηγό του οχήματος, λειτουργώντας έτσι σαν ένα προηγμένο σύστημα υποβοήθησης (Advanced Driver Assistance System - ADAS). Σε ένα δεύτερο στάδιο, είναι βέβαιο ότι κάποια στιγμή στο μέλλον τα έξυπνα οχήματα θα λειτουργούν εντελώς αυτόνομα χωρίς την παρουσία οδηγού. Σε αυτήν την περίπτωση τα έξυπνα οχήματα θα μπορούν να εκτελούν ενέργειες και να μετακινούνται σαν να οδηγούνταν κανονικά από κάποιον άνθρωπο. Ακόμη, θα πρέπει να αναφερθεί ότι μπορεί να υπάρξει ένας συνδυασμός τον δύο παραπάνω περιπτώσεων, δηλαδή κατά κύριο λόγο το έξυπνο όχημα να λειτουργεί συμβουλευτικά ως προς τον οδηγό, αλλά σε ορισμένες

επικίνδυνες καταστάσεις, που κρίνεται απαραίτητο, να αναλαμβάνει πλήρως τον έλεγχο του οχήματος. Στη συνέχεια, παρουσιάζονται συνοπτικά οι προκλήσεις στις οποίες καλείται να ανταπεξέλθει ένα προηγμένο σύστημα υποβοήθησης του οδηγού [3].

- Ανίχνευση και αποφυγή συγκρούσεων: Το σύστημα, με χρήση ειδικών αισθητήρων στα κατάλληλα σημεία του οχήματος, μπορεί να προβλέψει οποιαδήποτε πιθανή σύγκρουση και να ειδοποιήσει εγκαίρως τον οδηγό. Για παράδειγμα, σήμερα ένας τέτοιος μηχανισμός χρησιμοποιείται πολύ συχνά κατά το παρκάρισμα ενός οχήματος. Ένας ειδικός αισθητήρας στο πίσω μέρος του οχήματος ειδοποιεί τον οδηγό με ηχητικό μήνυμα εάν υπάρχει αντικείμενο πίσω από το όχημα.

- Προσαρμοζόμενος έλεγχος πορείας (Adaptive Cruise Control - ACC): Ο προσαρμοζόμενος έλεγχος πορείας εφαρμόζεται σε ολοένα και περισσότερα αυτοκίνητα. Μέσω αισθητήρων, το όχημα ανιχνεύει εάν υπάρχει άλλο όχημα μπροστά από αυτό και σε τι απόσταση βρίσκεται, και αναλόγως αποφασίζει μόνο του εάν μπορεί να επιταχύνει έως την ταχύτητα που έχει τεθεί ως άνω όριο ή εάν χρειάζεται να ελαττώσει ταχύτητα ή ακόμα και να φρενάρει και να έρθει σε πλήρη στάση για αποφυγή σύγκρουσης.

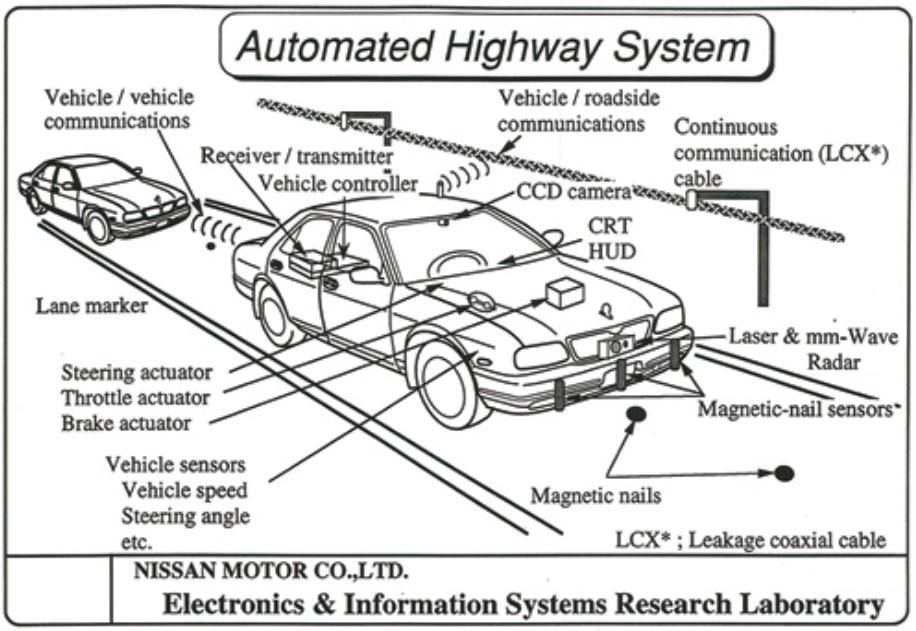
- Stop-and-Go: Ο μηχανισμός Stop-and-Go φροντίζει έτσι ώστε σε καταστάσεις κυκλοφοριακής συμφόρησης το όχημα να ξεκινάει και να σταματάει από μόνο του. Αυτό γίνεται δυνατό με χρήση αισθητήρων που παρατηρούν και ερμηνεύουν την κίνηση του μπροστινού οχήματος. Όταν το μπροστινό όχημα ξεκινάει τότε και το έξυπνο όχημα μπορεί να ξεκινήσει, ενώ όταν το μπροστινό όχημα σταματάει τότε και το έξυπνο όχημα σταματάει επίσης.

- Διατήρηση λωρίδας: Το σύστημα αυτό, μέσω της χρήσης αισθητήρων που ανιχνεύουν τις δύο γραμμές που ορίζουν την λωρίδα που κινείται το έξυπνο όχημα, μπορεί να ελέγχει το τιμόνι και να διατηρεί σταθερή πορεία κεντραρισμένη στην λωρίδα που βρίσκεται το όχημα.

- Μηχανισμός έκτακτης ανάγκης: Στην περίπτωση ατυχήματος, το έξυπνο όχημα μπορεί να αποκτήσει τις συντεταγμένες της τοποθεσίας μέσω συστήματος GPS, και να αποστείλει από μόνο του σχετικά μηνύματα στην οδική βοήθεια, αστυνομία κ.α. .

2.3 Έξυπνα Συστήματα Αυτοκινητόδρομων (Automated Highway Systems)

Τα έξυπνα συστήματα αυτοκινητόδρομων (Automated Highway Systems – AHS) [7] έχουν σκοπό να ενισχύσουν ακόμη παραπάνω την εύρυθμη λειτουργία της αυτοματοποιημένης οδήγησης. Ουσιαστικά, η έννοια των AHS συνδυάζει την παρουσία έξυπνων οχημάτων εξοπλισμένων με αισθητήρες στους δρόμους με την εγκατάσταση έξυπνων τεχνολογιών πληροφόρησης και επικοινωνιών στους ίδιους τους αυτοκινητόδρομους. Με αυτόν τον τρόπο εξασφαλίζεται όχι μόνο η συνεχής επικοινωνία μεταξύ του οδικού δικτύου και των οχημάτων γεγονός που βελτιώνει αισθητά το πως αντιλαμβάνεται κάθε όχημα το περιβάλλον του, αλλά και ο ασφαλέστερος και αποδοτικότερος συντονισμός της κυκλοφοριακής κίνησης ως σύνολο.



Πηγή: https://hackaday.com/wp-content/uploads/2015/08/thing.jpg

Σχήμα 2.2: Έξυπνα συστήματα αυτοκινητόδρομων και έξυπνα οχήματα

2.3.1 Διάκριση των Έξυπνων Συστημάτων Αυτοκινητόδρομων

Ανάλογα με τον βαθμό επικοινωνίας που υπάρχει μεταξύ των υποδομών και των οχημάτων, τα έξυπνα συστήματα αυτοκινητόδρομων μπορούν να διακριθούν στις εξής κατηγορίες [7]:

- Αυτόνομα: Σε αυτήν την περίπτωση οι αυτοκινητόδρομοι δεν είναι εξοπλισμένοι με αισθητήρες και έξυπνες τεχνολογίες, και δεν συμμετέχουν καθόλου στη διαδικασία της αυτοματοποιημένης οδήγησης. Τα έξυπνα οχήματα θα πρέπει μόνα τους, το κάθε ένα ξεχωριστά, να αντιμετωπίσουν καταστάσεις όπως αλλαγή λωρίδας, προσαρμογή ταχύτητας, αποφυγή εμποδίων και συγκρούσεων κ.α. .

- Συνεργατικά: Σε αυτήν την περίπτωση η υλοποίηση είναι ακριβώς ίδια με αυτή των αυτόνομων συστημάτων, με τη μόνη διαφορά ότι τα έξυπνα οχήματα επικοινωνούν μεταξύ τους. Αυτή η ανταλλαγή πληροφοριών μεταξύ των οχημάτων διευκολύνει ενέργειες όπως αλλαγή λωρίδας ή αποφυγή εμποδίων, καθώς υπάρχει συνεχής ενημέρωση και οι αποφάσεις παίρνονται συνεργατικά με κατανεμημένο τρόπο.

- Υποστηριζόμενα από την υποδομή: Σε αυτήν την περίπτωση πέρα από την επικοινωνία μεταξύ των έξυπνων οχημάτων, υπάρχει και επικοινωνία αυτών με τον αυτοκινητόδρομο. Τα δεδομένα που ανταλλάσσονται μεταξύ του αυτοκινητόδρομου και του οχήματος παίζουν βοηθητικό ρόλο, καθώς η τελική απόφαση για την εκτέλεση κάποιας ενέργειας παίρνεται αποκλειστικά από το έξυπνο όχημα.

- Ελεγχόμενα από την υποδομή: Σε αυτήν την περίπτωση ο αυτοκινητόδρομος είναι υπεύθυνος για οποιαδήποτε απόφαση παρθεί. Η κάθε ενέργεια που πραγματοποιείται από κάποιο όχημα καθορίζεται πλήρως από την υποδομή. Πρόκειται, ουσιαστικά, για ένα κατανεμημένο σύστημα στο οποίο υπάρχει κεντρικός έλεγχος. Μέσω αισθητήρων ο αυτοκινητόδρομος αντιλαμβάνεται τις

θέσεις των οχημάτων και ορίζει το πώς θα πραγματοποιηθούν οι διάφορες ενέργειες οδήγησης, όπως η απόσταση των οχημάτων, η ταχύτητα του κάθε οχήματος, το πότε θα πατηθεί φρένο, πιθανές κινήσεις που θα χρειαστούν για την αποφυγή εμποδίου ή σύγκρουσης, αλλαγή λωρίδας κάποιου οχήματος κ.α. . Όλα αυτά γίνονται δυνατά μέσω μηνυμάτων ελέγχου και εντολών που στέλνει ο έξυπνος αυτοκινητόδρομος στα οχήματα.

Θα πρέπει να αναφερθεί ότι μπορεί να υπάρξει και συνδυασμός των δύο τελευταίων περιπτώσεων, δηλαδή τα έξυπνα οχήματα επικοινωνούν μεταξύ τους και με τον αυτοκινητόδρομο για να πάρουν αποφάσεις, αλλά σε ορισμένες περιπτώσεις, όπως είσοδος/έξοδος από τον αυτοκινητόδρομο, αλλαγή λωρίδας κ.α., οι αποφάσεις παίρνονται από την υποδομή.

Από όλα τα είδη έξυπνων συστημάτων αυτοκινητόδρομων που παρουσιάστηκαν παραπάνω, τα συστήματα αυτοκινητόδρομων ελεγχόμενα από την υποδομή είναι αυτά στα οποία εστιάζεται η προσοχή της επιστημονικής κοινότητας. Αυτό συμβαίνει διότι πρόκειται για την πιο σύγχρονη και καινοτόμα μέθοδο υλοποίησης συστημάτων ITS, ενώ παράλληλα φαίνεται να είναι και το σύστημα το οποίο θα είναι το πιο αποτελεσματικό.

2.3.2 Συστήματα Αυτοκινητόδρομων Ελεγχόμενα από την Υποδομή

Σε αυτό το υποκεφάλαιο θα γίνει μια συνοπτική παρουσίαση της αρχιτεκτονικής και των εργασιών που καλούνται να πραγματοποιήσουν τα ελεγχόμενα από την υποδομή έξυπνα συστήματα μεταφορών. Όπως ήδη προαναφέρθηκε στο υποκεφάλαιο 2.3.1, ο αυτοκινητόδρομος είναι πλήρως υπεύθυνος για τη ρύθμιση της κίνησης και την ομαλή λειτουργία του κυκλοφοριακού συστήματος.

Αρχικά, η υποδομή ορίζει σύνολα, τα οποία καλούνται platoons, και αποτελούνται από έναν συγκεκριμένο αριθμό έξυπνων οχημάτων. Κάθε platoon έχει έναν αρχηγό, ο οποίος είναι το όχημα που βρίσκεται στην πιο μπροστινή θέση. Όλα τα υπόλοιπα οχήματα πρέπει να ακολουθούν τον αρχηγό του platoon στο οποίο ανήκουν. Η απόσταση μεταξύ οχημάτων που ανήκουν στο ίδιο platoon είναι σχετικά μικρή. Η απόσταση μεταξύ δύο διαφορετικών platoons είναι μεγαλύτερη. Αυτή η διαδικασία διαχωρισμού των οχημάτων σε μικρότερες ομάδες πραγματοποιείται έτσι ώστε να είναι ευκολότερη η διαχείριση και ο έλεγχος της κυκλοφοριακής κίνησης, με σεβασμό πάντα στην ασφάλεια των επιβατών.

Υπάρχουν τρεις βασικές λειτουργίες ελέγχου που ένας έξυπνος αυτοκινητόδρομος καλείται να υλοποιήσει. Αυτές είναι οι εξής [8]:

- Ανάθεση διαδρομής: Κατά την είσοδό του στον αυτοκινητόδρομο, το έξυπνο όχημα πρέπει να ανακοινώσει τον προορισμό του. Αυτό επιτυγχάνεται με ειδικά πρωτόκολλα που εξασφαλίζουν την επικοινωνία μεταξύ του αυτοκινητόδρομου και του οχήματος. Στη συνέχεια, η υποδομή αναλαμβάνει να αναθέσει στο αυτοκίνητο μια λωρίδα, η οποία διατηρείται για το μεγαλύτερο μέρος της διαδρομής, και φροντίζει να στείλει κατάλληλη εντολή αλλαγή λωρίδας όταν το όχημα πλησιάζει στην έξοδο που πρέπει να πάρει για να φτάσει στον προορισμό του.

- Οργάνωση κυκλοφοριακής κίνησης: Η υποδομή είναι υπεύθυνη για τον σχηματισμό των διάφορων platoons. Η διεργασία αυτή περιλαμβάνει και δυναμικές αλλαγές που μπορεί να συμβούν στο εσωτερικό των platoons, οι οποίες εξαρτώνται από τις εκάστοτε συνθήκες. Για παράδειγμα, μπορεί ο αυτοκινητόδρομος να χρειαστεί να στείλει εντολή σε ένα όχημα που να του λέει να ελαττώσει ταχύτητα, προκειμένου να εισέλθει στο platoon που βρίσκεται πίσω από αυτό.

- Εντολές πορείας: Η υποδομή επικοινωνεί με κάθε έξυπνο όχημα ξεχωριστά και στέλνει εντολές με σκοπό να ρυθμίσει την πορεία αυτού. Οι εντολές αυτές έχουν σχέση με την ταχύτητα που πρέπει να υιοθετεί το κάθε όχημα, με πιθανή αλλαγή λωρίδας που πρέπει να πραγματοποιηθεί, με τον σχηματισμό και την οργάνωση των επιμέρους οχημάτων σε platoons καθώς και με τη διαδικασία εισόδου/εξόδου κάποιου οχήματος στον αυτοκινητόδρομο.

Η αρχιτεκτονική και ο σχεδιασμός των έξυπνων συστημάτων αυτοκινητόδρομων μπορεί να χωριστεί σε επιμέρους επίπεδα τα οποία ακολουθούν ιεραρχική δομή. Κάθε ένα από αυτά τα επίπεδα είναι υπεύθυνα για την εκτέλεση διαφορετικών διεργασιών, ενώ η λειτουργία τους ως σύνολο εξασφαλίζει την προσδοκώμενη απόδοση της ευφυούς υποδομής.

Αναλυτικότερα, τα επιμέρους επίπεδα ,από κάτω προς τα πάνω, είναι τα εξής [8, 9]:

- Physical layer: Το επίπεδο αυτό περιλαμβάνει τους αισθητήρες που είναι εξοπλισμένοι τόσο στα έξυπνα οχήματα όσο και στον αυτοκινητόδρομο. Οι αισθητήρες αυτοί σχετίζονται με το πρακτικό κομμάτι της επικοινωνίας του ευφυούς συστήματος, με τη συλλογή δεδομένων, με τη λήψη εντολών ελέγχου που στέλνει ο αυτοκινητόδρομος και με τη διαχείριση των επιμέρους λειτουργιών που πρέπει να πραγματοποιηθούν μεμονωμένα για κάθε έξυπνο όχημα, βάσει των εντολών που στέλνονται από την υποδομή.

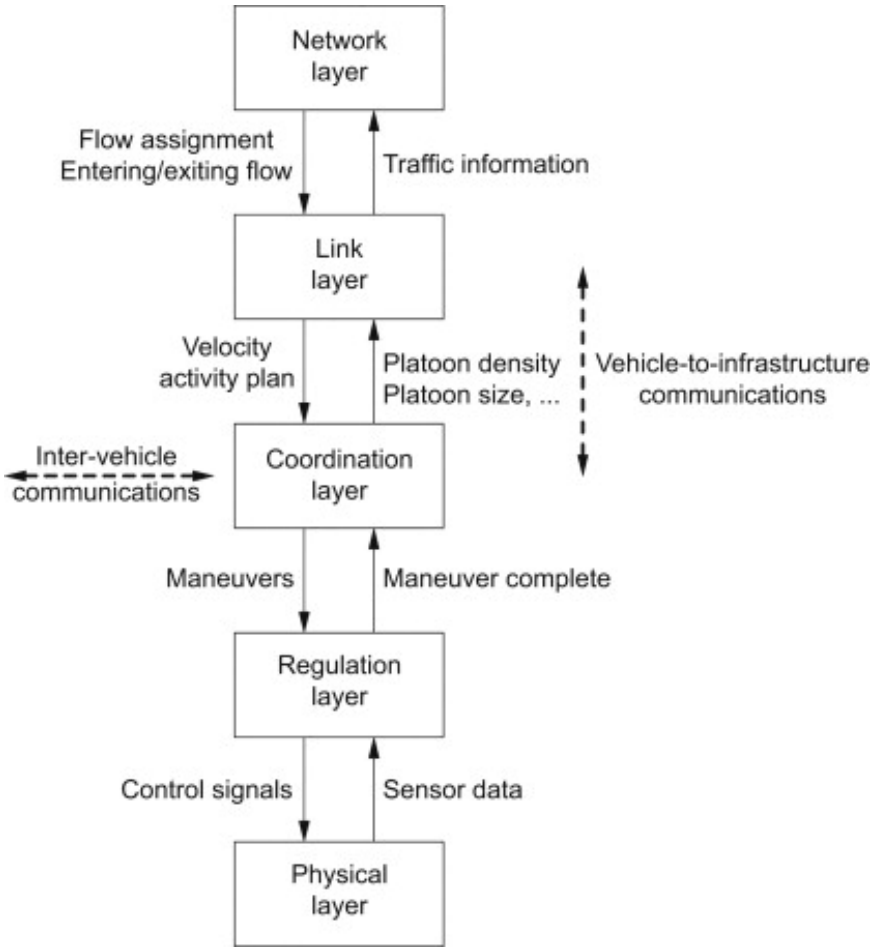
- Regulation layer: Το επίπεδο αυτό αναλαμβάνει να μετατρέψει τις εντολές ελέγχου που δέχεται από το πάνω ακριβώς επίπεδο (coordination layer), σε μορφή που είναι κατανοητή από τους αισθητήρες των έξυπνων οχημάτων έτσι ώστε να μπορέσουν να πραγματοποιηθούν οι αντίστοιχες ενέργειες. Ακόμη, φροντίζει να ενημερώσει το coordination layer για την επιτυχή ή μη έκβαση αυτών.

- Coordination layer: Το επίπεδο αυτό αναλαμβάνει να συντονίσει την λειτουργία των platoons με τους γείτονες αυτών. Υπάρχουν τρεις βασικές εντολές ελέγχου που χρησιμοποιούνται για την οργάνωση της κυκλοφοριακής κίνησης σε platoons. Αυτές είναι οι join, split και change lane. Με την join ο αρχηγός ενός platoon επιταχύνει και εισέρχεται στο platoon που βρίσκεται μπροστά από αυτόν, χάνοντας προφανώς το ρόλο του αρχηγού πλέον. Με την split ένα platoon χωρίζεται σε δύο επιμέρους. Με την change lane ένα έξυπνο όχημα πραγματοποιεί αλλαγή λωρίδας. Το coordination layer λαμβάνει πληροφορίες από το παραπάνω επίπεδο (link layer) και από τους αισθητήρες των έξυπνων οχημάτων μεμονωμένα, και φροντίζει να τις μετατρέψει σε εντολές ελέγχου και οργάνωσης των platoons με τον πιο αποδοτικό τρόπο. Ακόμη, φροντίζει να ενημερώνει το link layer για τα χαρακτηριστικά των διαφόρων platoons που έχουν διαμορφωθεί.

- Link layer: Ο αυτοκινητόδρομος χωρίζεται σε μικρότερα κομμάτια μήκους 1-5 χιλιόμετρα, τα οποία καλούνται links. Το link layer αναφέρεται ξεχωριστά σε

κάθε ένα από τα επιμέρους links. Ο κυριότερος σκοπός αυτού του επιπέδου είναι η μέγιστη αξιοποίηση της χωρητικότητας του αυτοκινητόδρομου διατηρώντας πάντα το επιθυμητό επίπεδο ασφάλειας που πρέπει να παρέχεται. Είναι υπεύθυνο για ενέργειες όπως ο υπολογισμός του βέλτιστου μεγέθους των platoons, των ταχυτήτων που πρέπει να υιοθετηθούν από τα διάφορα οχήματα καθώς και την ανάθεση λωρίδων που αυτά πρέπει να ακολουθήσουν. Συνεπώς, σε αυτό το επίπεδο καταστρώνεται το σχέδιο διαχείρισης της κυκλοφοριακής κίνησης, το οποίο στη συνέχεια μεταβιβάζεται στο coordination layer προκειμένου να λάβουν χώρα οι απαραίτητες ενέργειες.

- Network Layer: Το επίπεδο αυτό αναλαμβάνει το έργο διαχείρισης της εισερχόμενης/εξερχόμενης κυκλοφορίας καθώς και την επιλογή και ανάθεση των επιμέρους διαδρομών που πρέπει να ακολουθήσει το κάθε όχημα προκειμένου να φτάσει στον προορισμό του. Όλα αυτά πραγματοποιούνται με τρόπο που εξασφαλίζει την μέγιστη αξιοποίηση της χωρητικότητας του αυτοκινητόδρομου αποφεύγοντας την κυκλοφοριακή συμφόρηση , και τον βέλτιστο μέσο χρόνο ταξιδιού για κάθε όχημα. Οι πληροφορίες αυτές μεταβιβάζονται στο link layer προκειμένου να καταστρωθεί το πλάνο πορείας που θα ακολουθηθεί.



Πηγή: https://ars.els-cdn.com/content/image/3-s2.0-B9781782422112000131-f13-03-9781782422112.jpg

Σχήμα 2.3: Η αρχιτεκτονική των έξυπνων συστημάτων αυτοκινητόδρομων

2.3.3 Πλεονεκτήματα των Έξυπνων Συστημάτων Αυτοκινητόδρομων

Τα πλεονεκτήματα που προκύπτουν από την χρήση έξυπνων συστημάτων αυτοκινητόδρομων, αναμφίβολα βελτιώνουν την απόδοση του υπάρχοντος συστήματος μεταφορών. Πιο συγκεκριμένα:

- αξιοποιείται σε μεγαλύτερο βαθμό η χωρητικότητα του αυτοκινητόδρομου καθώς τα οχήματα θα κινούνται με μεγαλύτερες και σταθερές ταχύτητες, ενώ οι αποστάσεις μεταξύ τους θα είναι μικρότερες.

- εξασφαλίζεται η ασφαλής μεταφορά, αφού θα έχει αφαιρεθεί από την εξίσωση ο παράγοντας του ανθρώπινου λάθους.

- επιτυγχάνεται υψηλότερη απόδοση καθώς κακές καιρικές συνθήκες και περιβαλλοντικοί παράγοντες που πιθανότατα επηρεάζουν το πεδίο ορατότητας του οδηγού και οδηγούν σε επικίνδυνες καταστάσεις , πλέον δεν θα αποτελούν εμπόδιο στην οδήγηση.

- μειώνονται σε μεγάλο βαθμό η κατανάλωση καυσίμου και οι εκπομπές βλαβερών αερίων και ουσιών, καθώς η συχνότητα της απότομης εναλλαγής ταχυτήτων θα είναι αρκετά μικρότερη, ενώ παράλληλα ειδικοί αισθητήρες θα φροντίζουν για τη λειτουργία του οχήματος με τη βέλτιστη απόδοση.

Κεφάλαιο 3: Ανάλυση Δεδομένων Μεγάλου Όγκου

3.1 Μεγάλα Δεδομένα (Big Data)

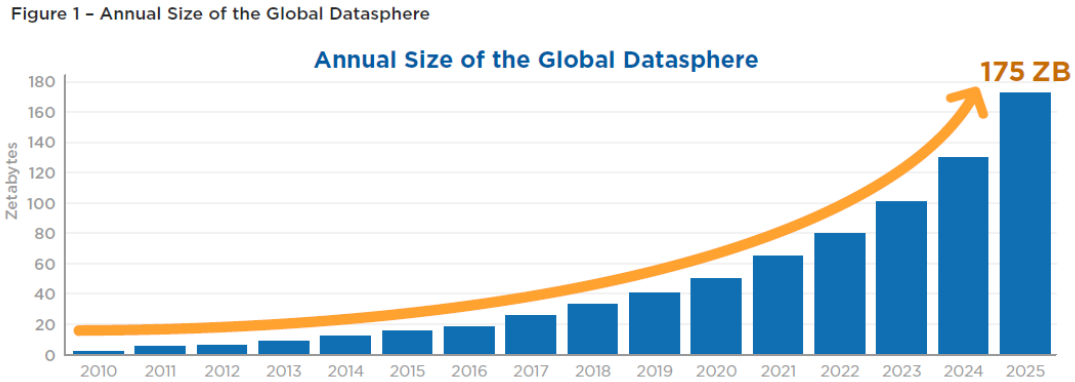
3.1.1 Η Ιστορία των Μεγάλων Δεδομένων

Εάν και η έννοια των δεδομένων μεγάλης κλίμακας (Big Data) εμφανίστηκε τα τελευταία χρόνια, τα θεμέλια πάνω στα οποία στηρίχθηκε η εμφάνιση και η εξέλιξη τους τέθηκαν πολλά χρόνια πριν. Ανέκαθεν ο άνθρωπος είχε την ανάγκη να αποθηκεύει και να αναλύει τις πληροφορίες που συνέλεγε. Η εξέλιξη της τεχνολογίας σε συνδυασμό με τη συνεχή αύξηση των πηγών παραγωγής δεδομένων, ιδιαίτερα τις τελευταίες δεκαετίες με την εμφάνιση του ίντερνετ και της ψηφιακής πληροφορίας, έχει οδηγήσει στην ανάγκη εύρεσης νέων αποδοτικότερων μεθόδων διαχείρισης της πληροφορίας. Στη συνέχεια, θα γίνει μια σύντομη ιστορική αναδρομή στα συστήματα διαχείρισης και αποθήκευσης πληροφορίας τον τελευταίο αιώνα με έμφαση στην περίοδο που εμφανίστηκαν οι υπολογιστές [10].

Τα πρώτα υπολογιστικά συστήματα εμφανίστηκαν γύρω στη δεκαετία του 1950. Ωστόσο, οι υπολογιστές εκείνη την εποχή δεν ήταν διαδεδομένοι στο ευρύ κοινό και συνεπώς η χρήση τους περιοριζόταν κυρίως στην εκτέλεση υπολογιστικών προγραμμάτων από εκπαιδευτικούς και ερευνητικούς οργανισμούς. Καθώς περνούσε ο χρόνος οι υπολογιστές ξεκίνησαν να χρησιμοποιούνται τόσο από μεγάλες εταιρίες όσο και από απλούς ανθρώπους και αναπόφευκτα προέκυψε η ανάγκη εύρεσης αποδοτικών μεθόδων αποθήκευσης και διαχείρισης της παραγόμενης πληροφορίας. Το 1960, ο Charles Bachman πρότεινε το πρώτο γνώστο σύστημα διαχείρισης δεδομένων το οποίο ονομαζόταν Integrated Database System (IDS) [11]. Λίγα χρόνια αργότερα, το 1966, η IBM προτείνει ένα νέο σύστημα το οποίο ονομάζεται Information Management System (IMS) σύμφωνα με το οποίο τα δεδομένα μοντελοποιούνται με ιεραρχικό τρόπο [12]. Τα συστήματα αυτά αποτέλεσαν τη βάση για την ανάπτυξη ενός πολύ σπουδαίου συστήματος αποθήκευσης και διαχείρισης πληροφοριών το οποίο χρησιμοποιείται ακόμη και σήμερα. Πρόκειται για το σχεσιακό μοντέλο βάσεων δεδομένων (Relational Database Management System - RDBMS), το οποίο προτάθηκε από την IBM το 1970 [13]. Το σχεσιακό μοντέλο αποτελεί μια σπουδαία καινοτομία στο χώρο της επιστήμης των δεδομένων. Όπως υποδεικνύει και η ονομασία του, ορίζονται σχέσεις μεταξύ των δεδομένων οι οποίες επιτρέπουν την οργανωμένη αποθήκευσή τους και επομένως την ταχύτερη και ευκολότερη ανάκτησή τους. Θα πρέπει να αναφερθεί ότι έχουν δημιουργηθεί γλώσσες προγραμματισμού οι οποίες χρησιμοποιούνται αποκλειστικά για το χειρισμό των RDBMSs. Η πιο γνώστη και ευρέως χρησιμοποιούμενη είναι η SQL (Structured Query Language).

Το σχεσιακό μοντέλο φαινόταν να καλύπτει τις ανάγκες διαχείρισης και αποθήκευσης των δεδομένων για αρκετό καιρό. Ωστόσο, το πέρασμα του χρόνου ανέδειξε τις αδυναμίες του καθώς αυξάνεται ο όγκος της πληροφορίας που καλείται να διαχειριστεί και ιδιαίτερα

σε περιπτώσεις που η σχέση μεταξύ των δεδομένων δεν μπορεί να εκφραστεί με σαφή τρόπο. Με την εμφάνιση του Internet of Things (IoT) όλο και περισσότερες συσκευές, οι οποίες συλλέγουν και παράγουν δεδομένα σε πραγματικό χρόνο, συνδέονται στο διαδίκτυο με συνέπεια να αυξάνεται δραστικά ο όγκος της πληροφορίας [14]. Επιπλέον, ο αριθμός των χρηστών που είναι συνδεδεμένος στο διαδίκτυο αυξάνεται μέρα με τη μέρα, με συνέπεια να αυξάνεται και ο αριθμός των παραγόμενων δεδομένων. Για παράδειγμα, για το Facebook, μια από τις πιο γνωστές διαδικτυακές πλατφόρμες, υπολογίζεται ότι το πλήθος των χρηστών που το χρησιμοποιούν το 2020 είναι περίπου 1.69 δισεκατομμύρια [15], ενώ το μέγεθος των δεδομένων που παράγεται κάθε μέρα είναι περίπου ίσο με 4 petabytes ($4 \cdot 10^{15}$ bytes) [16].



Πηγή: https://www.csa.gov.sg/singcert/-/media/singcert/cybersense/idc_datasphere.png

Σχήμα 3.1: Η ετήσια αύξηση των δεδομένων συναρτήσει του χρόνου

Στο σχήμα 3.1 μπορεί κανείς να δει ότι το σύνολο των παραγόμενων δεδομένων αυξάνεται εκθετικά με τον χρόνο, ενώ το 2025 υπολογίζεται ότι το μέγεθος των δεδομένων θα είναι περί τα 175 zettabytes ($1.75 \cdot 10^{23}$ bytes).

Για τους λόγους που περιγράφηκαν παραπάνω, το ενδιαφέρον έχει επικεντρωθεί στην αναζήτηση πιο αποδοτικών τρόπων διαχείρισης και αποθήκευσης του τεράστιου όγκου της παραγόμενης πληροφορίας. Η αναζήτηση αυτή είχες ως αποτέλεσμα τη δημιουργία ενός νέου μοντέλου βάσεων δεδομένων το οποίο ονομάζεται NoSQL. Ο όρος "NoSQL" προτάθηκε για πρώτη φορά το 1998 από τον Carlo Strozzi, αλλά επανήλθε δυναμικά στο προσκήνιο το 2010 περίπου [17]. Όπως προκύπτει και από την ονομασία τους, οι βάσεις NoSQL ήρθαν για να αναιρέσουν την κλασική λειτουργία των σχεσιακών βάσεων. Πρόκειται για ένα μοντέλο το οποίο προσφέρει μεγαλύτερη ευελιξία στον τρόπο αποθήκευσης των δεδομένων, επιτρέπει την ταχύτατη ανάκτηση πληροφορίων και μπορεί να λειτουργήσει με κατανεμημένο τρόπο.

Κλείνοντας, θα πρέπει να αναφερθεί ότι τα τελευταία χρόνια έχουν αναπτυχθεί αρκετά λογισμικά που εξειδικεύονται στην επεξεργασία και την ανάλυση δεδομένων μεγάλης κλίμακας. Δύο από τα σημαντικότερα είναι το Hadoop και το Spark τα οποία επιτρέπουν τη διαχείριση δεδομένων με παράλληλο και κατανεμημένο τρόπο [18].

3.1.2 Ορισμός των Μεγάλων Δεδομένων

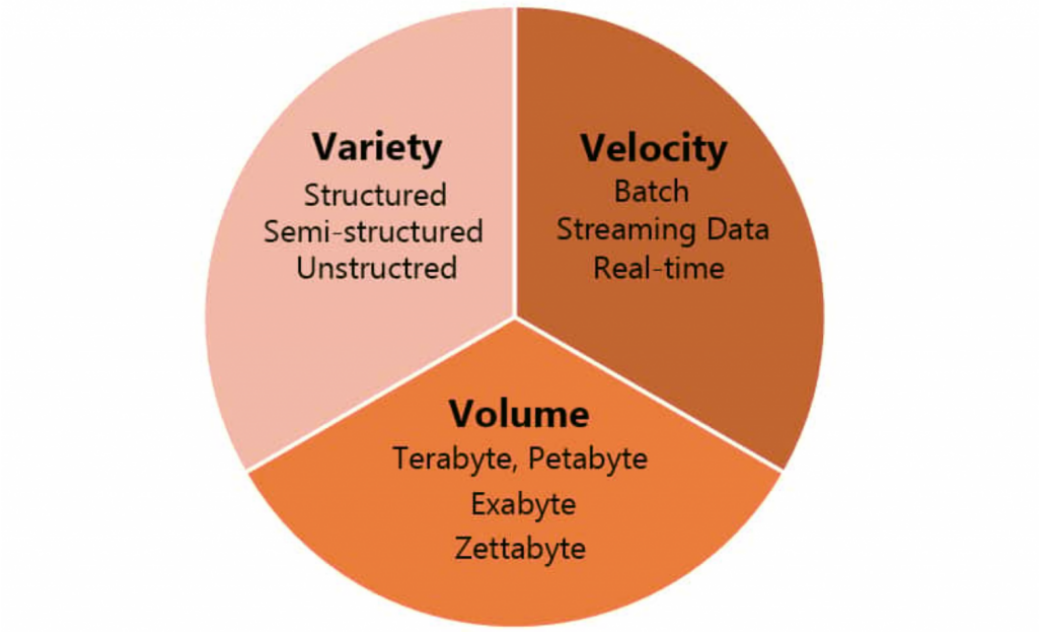
Ο όρος μεγάλα δεδομένα (Big Data) χρησιμοποιείται για να περιγράψει δεδομένα τα οποία χαρακτηρίζονται από εξαιρετικά μεγάλο όγκο, ο οποίος καθιστά ιδιαίτερα

δύσκολη την αποθήκευση και την ανάλυση τους από τις παραδοσιακές μεθόδους διαχείρισης βάσεων δεδομένων. Δεν υπάρχει κάποιος επίσημος ορισμός για το τι είναι τα δεδομένα μεγάλης κλίμακας, ωστόσο ένας κοινώς αποδεκτός και ευρέως διαδεδομένος ορισμός που χαρακτηρίζει τα μεγάλα δεδομένα είναι αυτός που έχει αποδοθεί από την εταιρία Gartner [19]. Πρόκειται για μια εταιρία παγκόσμιου βεληνεκούς σύμφωνα με την οποία:

"Τα Big Data είναι υψηλού όγκου, υψηλής ταχύτητας και υψηλής ποικιλίας στοιχεία που απαιτούν αποδοτικές και καινοτόμες μορφές επεξεργασίας πληροφοριών"

Η διατύπωση αυτή οδήγησε σε ένα από τα πιο εύστοχα μοντέλα περιγραφής της έννοιας των μεγάλων δεδομένων. Πρόκειται για το μοντέλο 3V, σύμφωνα με το οποίο τα κύρια χαρακτηριστικά των μεγάλων δεδομένων είναι τρία και είναι τα εξής:

- Όγκος (Volume): Μεγάλη ποσότητα δεδομένων
- Ταχύτητα (Velocity): Μεγάλη ταχύτητα δεδομένων
- Ποικιλία (Variety): Ευρεία ποικιλία δεδομένων



Πηγή: https://cdn.theviable.co/assets/content/uploads/2018/01/3V-Model-of-Big-Data-1024x633.png

Σχήμα 3.2: Τα τρία V των μεγάλων δεδομένων

Ο όρος volume αναφέρεται στον τεράστιο όγκο δεδομένων που τα συστήματα διαχείρισης καλούνται να αντιμετωπίσουν. Αυτό οφείλεται, κυρίως, στη ψηφιοποίηση της πληροφορίας. Όπως έχει ήδη αναφερθεί στο υποκεφάλαιο 3.1.1, το συνολικό μέγεθος των δεδομένων είναι της τάξης των zettabytes και συνεπώς οι απαιτήσεις που αφορούν την ταχύτατη και αποτελεσματική ανάλυση των δεδομένων συνεχώς αυξάνονται.

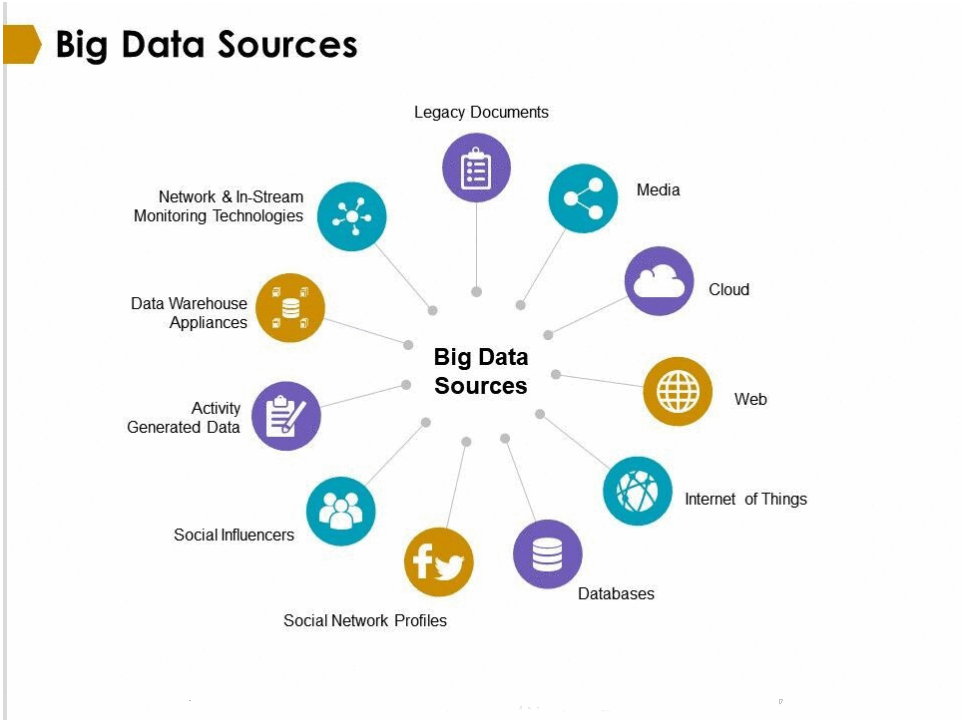
Ο όρος velocity αναφέρεται στον ταχύτατο ρυθμό με τον οποίο εισέρχονται νέα δεδομένα αλλά και ανανεώνονται τα ήδη υπάρχοντα. Η κυρία πρόκληση είναι η δημιουργία συστημάτων διαχείρισης και αποθήκευσης πληροφοριών τα οποία θα μπορούν να ανταπεξέλθουν σε μεγάλο όγκο πληροφορίας ο οποίος έρχεται συνεχώς και ασταμάτητα. Επιπλέον, ο όρος velocity έχει να κάνει και με τον χρόνο που απαιτείται για την επεξεργασία και την ανάλυση των δεδομένων καθώς αυτά εισέρχονται στο σύστημα. Η ανάλυση των δεδομένων και η εξαγωγή συμπερασμάτων σε πραγματικό χρόνο είναι μια βασική απαίτηση.

Ο όρος variety αναφέρεται στο μεγάλο εύρος των πιθανών διαφορετικών τύπων δεδομένων που εισέρχονται στο σύστημα. Αυτό οφείλεται, κυρίως, στην αύξηση των πηγών που μπορούν να παράξουν πληροφορία. Όπως αναφέρθηκε και στο υποκεφάλαιο 3.1.1, η είσοδος του Internet of Things στην ανθρώπινη ζωή επιτρέπει τη συλλογή και δημιουργία δεδομένων από πολλές μικροσυσκευές και αισθητήρες. Οι συνεχόμενα αυξανόμενες πηγές πληροφορίας οδηγούν αναπόφευκτα σε μεγαλύτερη ποικιλομορφία δεδομένων. Γενικότερα, οι τύποι των μεγάλων δεδομένων μπορούν να διακριθούν σε τρεις μεγάλες κατηγορίες:

- Δομημένα δεδομένα (Structured Data): Πρόκειται για δεδομένα τα οποία έχουν μια προκαθορισμένη δομή και οργάνωση. Για παράδειγμα, τέτοιου είδους δεδομένα μπορεί να είναι δεδομένα που παράγονται από τους αισθητήρες μέτρησης της επιτάχυνσης ενός έξυπνου οχήματος, δεδομένα GPS, γενικότερα δεδομένα στα οποία ορίζεται μια σχέση όπως είναι το ονοματεπώνυμο και το ΑΜΚΑ, κ.α. .

- Αδόμητα δεδομένα (Unstructured Data): Πρόκειται για δεδομένα τα οποία ούτε έχουν κάποια συγκεκριμένη μορφή ούτε ακολουθούν κάποια προκαθορισμένη δομή. Για παράδειγμα, τέτοιου είδους δεδομένα μπορεί να είναι μια εικόνα, ένα βίντεο, ένα έγγραφο κειμένου κ.α. .

- Ημι-δομημένα δεδομένα (Semi-structured Data): Πρόκειται για δεδομένα τα οποία δεν μπορούν να χαρακτηριστούν ως δομημένα, ωστόσο ακολουθούν ένα ελαστικό σύνολο κανόνων. Για παράδειγμα, τέτοιου είδους δεδομένα μπορεί να είναι ένα αρχείο JSON (JavaScript Object Notation), ένα αρχείο XML (eXtensible Markup Language) κ.α. .



Πηγή: https://www.slideteam.net/slide01.jpg

Σχήμα 3.3: Οι πηγές των μεγάλων δεδομένων

Κλείνοντας, θα πρέπει να αναφερθεί ότι στο κλασικό μοντέλο περιγραφής των μεγάλων δεδομένων 3V έχουν προστεθεί τρία ακόμη V, τα οποία ορίζουν τα εξής:

- Ειλικρίνεια (Veracity): Ο όρος αυτός αναφέρεται στην απαίτηση που υπάρχει για έλεγχο της αξιοπιστίας των μεγάλων δεδομένων. Λόγω των πολλών και διαφορετικών μορφών και πηγών που έχουν τα μεγάλα δεδομένα η ποιότητα και η ακρίβεια των δεδομένων είναι λιγότερο ελεγχόμενες παράμετροι.

- Αξία (Value): Ο όρος αυτός αναφέρεται στην απαίτηση που υπάρχει για τη μετατροπή των μεγάλων δεδομένων σε αξιοποιήσιμη και χρήσιμη πληροφορία.

- Μεταβλητότητα (Variability): Ο όρος αυτός αναφέρεται στην απαίτηση που υπάρχει για προσαρμογή στη συνεχόμενη αλλαγή της μορφής των μεγάλων δεδομένων. Ο αποτελεσματικός χειρισμός αυτής της μεταβλητότητας εξασφαλίζει την ορθή ερμηνεία της πληροφορίας.

3.1.3 Μη Σχεσιακές Βάσεις Δεδομένων (NoSQL Databases)

Οι βάσεις NoSQL [20] προέκυψαν από την ανάγκη διαχείρισης δεδομένων τα οποία δεν ακολουθούν κάποια συγκεκριμένη δομή. Οι σχεσιακές βάσεις δεδομένων μπορεί να είναι πολύ αποτελεσματικές για την αποθήκευση δομημένων δεδομένων, ωστόσο η λειτουργία τους δεν μπορεί να επεκταθεί για αδόμητα δεδομένα. Ο όρος NoSQL (Not Only SQL και όχι No SQL) αναφέρεται σε ένα σύνολο βάσεων δεδομένων οι οποίες ούτε έχουν τα τυπικά χαρακτηριστικά μιας σχεσιακής βάσης ούτε χρησιμοποιούν την SQL για τη διαχείριση της βάσης. Σε αντίθεση με μια τυπική βάση SQL, όπου χρησιμοποιούνται πίνακες για να αποθηκευτούν τα δεδομένα και να οριστούν οι μεταξύ

τους σχέσεις, μια NoSQL βάση χρησιμοποιεί εντελώς διαφορετικό τρόπο για τη διαχείριση και την αποθήκευση των δεδομένων. Τα NoSQL συστήματα έχουν την ικανότητα να αποθηκεύουν και να ανακτούν μεγάλες ποσότητες δεδομένων, αδιαφορώντας για τις όποιες σχέσεις μπορεί να υπάρχουν μεταξύ των δεδομένων. Τα κυριότερα χαρακτηριστικά μιας NoSQL βάσης δεδομένων είναι τα εξής:

- Το εσωτερικό μιας NoSQL βάσης δεν διατηρεί κάποια συγκεκριμένη και προκαθορισμένη δομή (schema free). Αντίθετα, μια SQL βάση ορίζεται από αυστηρούς κανόνες απαραίτητους για την αποτύπωση των σχέσεων μεταξύ των δεδομένων.

- Δεν υπάρχει κάποια εγγύηση για την αξιοπιστία των συναλλαγών που πραγματοποιούνται με τη βάση. Αντίθετα, μια SQL βάση εξασφαλίζει ότι μια συναλλαγή θα γίνει υπό τους περιορισμούς του θεωρήματος ACID (Atomicity, Consistency, Isolation, Durability). Η ατομικότητα (atomicity) εξασφαλίζει ή ότι θα γίνει ολόκληρη η συναλλαγή ή ότι δεν θα γίνει καθόλου. Δεν μπορεί να γίνει κάποιο μέρος της συναλλαγής μόνο δηλαδή. Η συνέπεια (consistency) διασφαλίζει ότι η βάση θα διατηρείται σε μια συνεπή κατάσταση. Για παράδειγμα, δεν μπορεί να τοποθετηθούν σε μια βάση δεδομένα τύπου string σε μια θέση που η αναμενόμενη τιμή είναι ακέραιος. Η απομόνωση (isolation) αναφέρεται στην απαίτηση ότι τη στιγμή που τροποποιούνται κάποια δεδομένα από μια συναλλαγή δεν μπορούν να τροποποιηθούν τα ίδια δεδομένα από κάποια άλλη, καθώς κάτι τέτοιο θα οδηγούσε σε ασυνέπεια. Η μονιμότητα (durability) εξασφαλίζει ότι οι αλλαγές που έχει πραγματοποιήσει με επιτυχία μια συναλλαγή δεν πρόκειται να χαθούν ακόμη και αν σταματήσει η λειτουργία της βάσης για κάποιο λόγο. Τα NoSQL συστήματα δεν συμμορφώνονται απόλυτα στους κανόνες ACID μιας σχεσιακής βάσης δεδομένων, γεγονός που επιτρέπει την ταχύτερη διαχείριση των δεδομένων και την εγγύηση υψηλής διαθεσιμότητας.

- Η μέθοδος υλοποίησης των NoSQL βάσεων ευνοεί τον κατανεμημένο τρόπο λειτουργίας του συστήματος. Αυτό σημαίνει ότι σε περίπτωση που χρειαστεί, για παράδειγμα λόγω του όγκου των δεδομένων, το σύστημα μπορεί εύκολα να κλιμακωθεί οριζόντια (horizontal scalability) με την προσθήκη επιπλέον κόμβων. Αντίθετα, σε μια αντίστοιχη περίπτωση μια σχεσιακή βάση θα έπρεπε να κλιμακωθεί κάθετα (vertical scalabilty) με την προσθήκη νέων πόρων αποθήκευσης στον ήδη υπάρχων server, κάτι το οποίο κοστίζει και δεν μπορεί να επιτευχθεί εύκολα.

Με βάση τα όσα ειπώθηκαν παραπάνω, είναι φανερό πως οι NoSQL βάσεις είναι πιο κατάλληλες για τη διαχείριση των μεγάλων δεδομένων από ότι είναι οι σχεσιακές βάσεις. Σε αυτό το σημείο, πρέπει να αναφερθεί ότι μια βάση NoSQL μπορεί να ανήκει σε μια από τις εξής τέσσερις κατηγορίες:

- Key-Values Stores: Η βάση μπορεί να παρομοιαστεί με μια δομή κατακερματισμού (hash table) της οποίας κάθε στοιχείο (value) προσδιορίζεται μοναδικά από κάποιο κλειδί (key).

- Column Family Stores: Η βάση αποτελείται από δομές οι οποίες ονομάζονται συλλογές στηλών (column families). Κάθε συλλογή στηλών περιέχει δεδομένα σε μορφή γραμμών, όπως συμβαίνει και στις σχεσιακές βάσεις. Μια βασική διαφορά είναι ότι στο ίδιο column family δύο γραμμές δεν απαιτείται ούτε να έχουν το

ίδιο μήκος ούτε τις ίδιες στήλες. Επίσης, δεν υπάρχει κάποια σχέση που μπορεί να οριστεί μεταξύ των γραμμών. Κάθε γραμμή περιέχει όλη τη πληροφορία του αντικειμένου στο οποίο αναφέρεται. Τέλος, κάθε γραμμή χαρακτηρίζεται μοναδικά από κάποιο κλειδί.

- Document Databases: Η βάση απαρτίζεται από έγγραφα (documents) τα οποία αποτελούν συλλογές από άλλες key-valued συλλογές. Για την αποθήκευση των δεδομένων χρησιμοποιείται συνήθως το μορφότυπο JSON η κάποια παραλλαγή του.

- Graph Databases: Η βάση μπορεί να παρομοιαστεί με ένα γράφο (graph). Κάθε κόμβος χρησιμοποιείται για να αναπαραστήσει ένα αντικείμενο της βάσης, ενώ μια ακμή εκφράζει μια σχέση που έχουν οι κόμβοι, και επομένως τα αντικείμενα, που ενώνει. Η μορφή γράφου, σε αντίθεση με τους πίνακες των σχεσιακών βάσεων δεδομένων, επιτρέπει την κατανεμημένη λειτουργία του συστήματος.

Κλείνοντας, θα παρουσιαστεί το θεώρημα CAP (Consistency, Availability, Partition Tolerance) [21] καθώς σχετίζεται άμεσα με τη κατανεμημένη λειτουργία των NoSQL βάσεων. Σύμφωνα με αυτό είναι αδύνατο για ένα κατανεμημένο σύστημα υπολογιστών να παρέχει ταυτόχρονα τις τρεις εξής εγγυήσεις:

- Συνέπεια (Consistency): Όλοι οι κόμβοι πρέπει να βλέπουν τα ίδια δεδομένα κάθε χρονική στιγμή.

- Διαθεσιμότητα (Availability): Εκφράζει την εγγύηση ότι κάθε αίτημα παίρνει μια απάντηση σχετικά με το αν πέτυχε ή όχι.

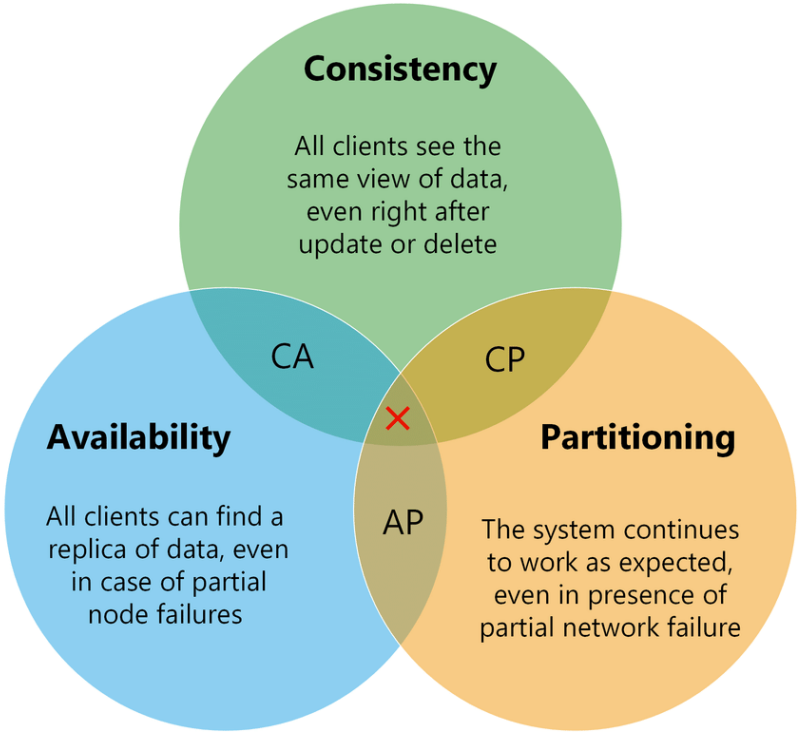
- Ανοχή διαμερισμού (Partition Tolerance): Εκφράζει την εγγύηση ότι το κατανεμημένο σύστημα συνεχίζει να λειτουργεί ορθά παρά πιθανό διαχωρισμό που μπορεί να συμβεί σε αυτό (π.χ. λόγω βλάβης στο δίκτυο).

Κάθε κατανεμημένο σύστημα πρέπει να έχει την ιδιότητα της ανοχής διαμερισμού, καθώς οι βλάβες είναι αναπόφευκτες και πάντα πρέπει να εξασφαλίζεται η σωστή λειτουργία του συστήματος. Έτσι, αναλόγως με ποια ιδιότητα επιλέγουν να θυσιάσουν οι NoSQL βάσεις μπορούν να διακριθούν σε:

- CP βάσεις: Σε αυτήν την περίπτωση θυσιάζεται η ιδιότητα της διαθεσιμότητας προκειμένου να εξασφαλιστούν η συνέπεια και η ανοχή διαμερισμού. Κάθε φορά που διακόπτεται η επικοινωνία μεταξύ δύο κόμβων, το σύστημα αναστέλλει τη λειτουργία του μη συνεπή κόμβου μέχρι να ανακτηθεί η σύνδεση.

- AP βάσεις: Σε αυτήν την περίπτωση θυσιάζεται η ιδιότητα της συνέπειας προκειμένου να εξασφαλιστούν η διαθεσιμότητα και η ανοχή διαμερισμού. Κάθε φορά που διακόπτεται η επικοινωνία μεταξύ δύο κόμβων, όλοι οι κόμβοι παραμένουν ενεργοί ακόμη και αυτοί που είναι μη συνεπείς. Όταν ανακτηθεί η σύνδεση ακολουθεί μια διαδικασία συγχρονισμού μέσω της οποίας γίνεται αποκατάσταση της συνέπειας στους μη συνεπείς κόμβους.

- CA βάσεις: Σε αυτήν την περίπτωση θυσιάζεται η ιδιότητα της ανοχής διαμερισμού προκειμένου να εξασφαλιστούν η συνέπεια και η διαθεσιμότητα. Αυτές οι βάσεις υφίστανται μόνο σε θεωρητικό επίπεδο καθώς κάθε κατανεμημένο σύστημα πρέπει να έχει την ιδιότητα της ανοχής διαμερισμού αφού οι βλάβες είναι αναπόφευκτες.



Πηγή: https://learncodeshare.files.wordpress.com/2018/12/visualization-of-cap-theorem.png

Σχήμα 3.4: Το θεώρημα CAP

3.1.4 Μεγάλα Δεδομένα και Ευφυή Συστήματα Μεταφορών

Τα έξυπνα οχήματα, όπως αναφέρθηκε και στο υποκεφάλαιο 2.2, διαθέτουν πληθώρα αισθητήρων οι οποίοι συλλέγουν δεδομένα σε πραγματικό χρόνο. Πρόκειται για δεδομένα μεγάλης κλίμακας τα οποία πρέπει να αξιοποιηθούν με σωστό και αποδοτικό τρόπο έτσι ώστε να επιτευχθεί η αυτοματοποιημένη λειτουργία ενός οχήματος. Τα μεγάλα δεδομένα που συλλέγονται μπορεί να σχετίζονται με:

- τη δυναμική και την κινηματική κατάσταση του έξυπνου οχήματος (accelometer, gyrometer, GPS, κ.α.).
- το περιβάλλον μέσα στο οποίο βρίσκεται το έξυπνο όχημα (camera, lidar, GPS, κ.α.).
- τη κατάσταση του οδηγού και των επιβατών (camera, κ.α.).
- την επικοινωνία του έξυπνου οχήματος με κάποιο άλλο όχημα ή κάποιον έξυπνο αυτοκινητόδρομο.

Στα πλαίσια της εγκαθίδρυσης των ευφυών συστημάτων μεταφορών, όλο και περισσότερες αυτοκινητιστικές βιομηχανίες στρέφουν το ενδιαφέρον τους στη δημιουργία συστημάτων και τεχνολογιών που διαχειρίζονται και αξιοποιούν τα μεγάλα δεδομένα που συλλέγονται από τους αισθητήρες ενός έξυπνου οχήματος. Πιο συγκεκριμένα, μερικές από τις πιο σύγχρονες εφαρμογές είναι οι εξής:

- τα προηγμένα συστήματα υποβοήθησης του οδηγού (ADAS) τα οποία λειτουργούν με βάση τα μεγάλα δεδομένα που συλλέγονται από τους αισθητήρες του οχήματος [22].

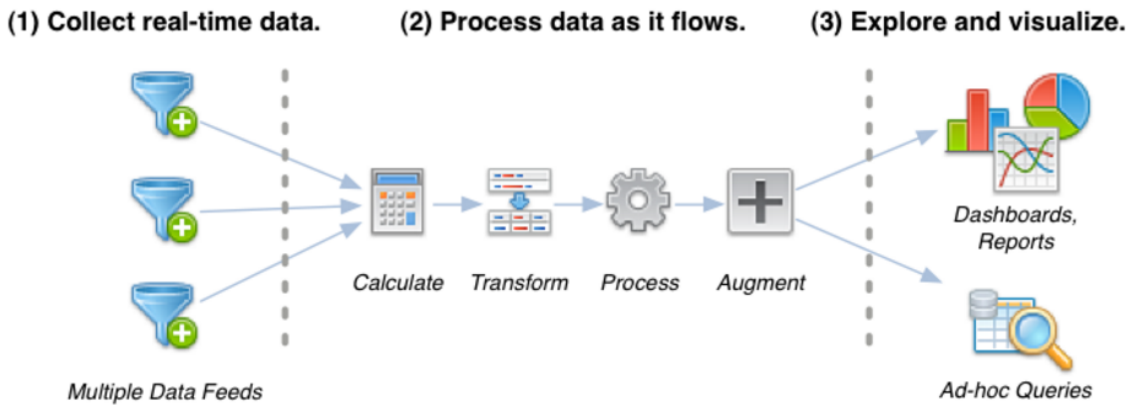
- τα συστήματα ανάλυσης της συμπεριφοράς του οδηγού. Τα μεγάλα δεδομένα που συλλέγονται από τους αισθητήρες του οχήματος χρησιμοποιούνται για την ερμηνεία και την αξιολόγηση της οδηγικής συμπεριφοράς. Τέτοιου είδους συστήματα χρησιμοποιούνται κυρίως από ασφαλιστικές εταιρίες προκειμένου να προσδιοριστεί η οδηγική συμπεριφορά των εμπλεκομένων μελών σε περίπτωση ατυχήματος (Pay-How-You-Drive - PHYD)[23].

- τα έξυπνα οχήματα της εταιρίας Tesla [24] τα οποία με βάση τα μεγάλα δεδομένα που συλλέγουν από τους αισθητήρες που έχουν δύνανται να μετακινούνται με αυτοματοποιημένο τρόπο.

Συνεπώς, είναι ξεκάθαρο ότι ο τομέας των ευφυών συστημάτων μεταφορών σχετίζεται άμεσα με την έννοια των μεγάλων δεδομένων. Η μεγαλύτερη πρόκληση σχετικά με την υλοποίηση συστημάτων παρόμοιων με αυτά που παρουσιάστηκαν παραπάνω, έγκειται στον τρόπο με τον οποίο μπορεί να αξιοποιηθεί ο τεράστιος όγκος δεδομένων που παράγεται από τους αισθητήρες. Η διαδικασία που ακολουθείται για την εξαγωγή χρήσιμης πληροφορίας από δεδομένα μεγάλης κλίμακας παρουσιάζεται αναλυτικά στο υποκεφάλαιο 3.2.

3.2 Ανακάλυψη Γνώσης (Knowledge Discovery in Databases - KDD)

Τα μεγάλα δεδομένα, όπως αναφέρθηκε και στο υποκεφάλαιο 3.1.2, χαρακτηρίζονται από εξαιρετικά μεγάλο όγκο, ταχύτητα και ποικιλομορφία με αποτέλεσμα να απαιτούνται ειδικές τεχνολογίες και αναλυτικές μέθοδοι για τη μετατροπή τους σε αξιοποιήσιμη πληροφορία. Επιπλέον, σε πολλές περιπτώσεις η εξαγωγή της χρήσιμης πληροφορίας απαιτείται να γίνει με γρήγορο και αποδοτικό τρόπο. Δεν είναι λίγες οι εφαρμογές που στοχεύουν στην ανακάλυψη γνώσης από τεράστιους όγκους δεδομένων σε πραγματικό χρόνο. Ο όρος "σε πραγματικό χρόνο" υποδηλώνει ότι τα δεδομένα υφίστανται επεξεργασία, αναλύονται και μετασχηματίζονται ακριβώς τη στιγμή που εισέρχονται σύστημα (Real-time Big Data Analytics). Σε αυτό το υποκεφάλαιο, λοιπόν, θα γίνει μια περιγραφή της μεθοδολογίας που ακολουθείται προκειμένου να εξαχθεί γνώση από τα δεδομένα μεγάλης κλίμακας.



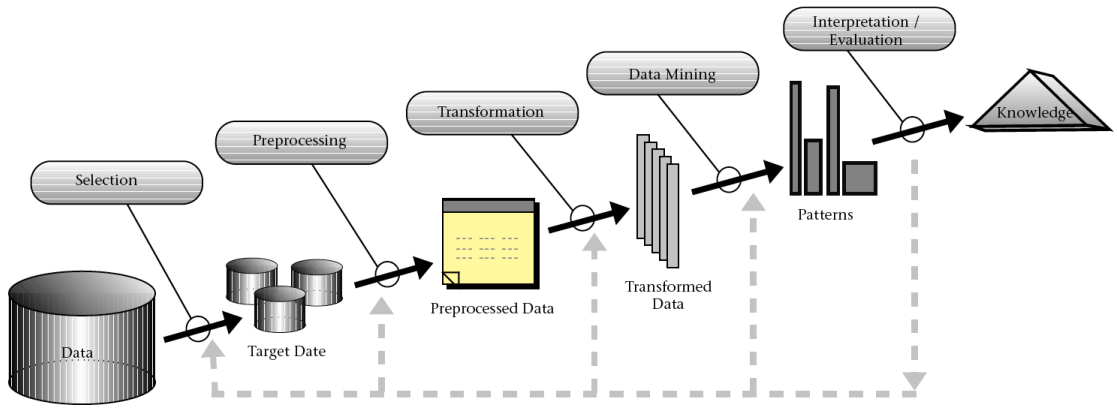
Πηγή: https://static.packt-cdn.com/products/9781787281202/graphics/4615c687-a0f1-4ce6-8c15-b3637f264486.png

Σχήμα 3.5: Ανάλυση δεδομένων σε πραγματικό χρόνο

3.2.1 Ορισμός της Ανακάλυψης Γνώσης

Η ανακάλυψη γνώσης (Knowledge Discovery – KDD) [25] είναι μια διαδικασία που περιλαμβάνει την ανακάλυψη νέας, χρήσιμης και αξιοποιήσιμης γνώσης μέσω αναγνώρισης μοτίβων (patterns) σε δεδομένα μεγάλης κλίμακας. Η διαδικασία αυτή αποτελείται από επιμέρους στάδια τα οποία είναι τα εξής:

- Data Collection: Το στάδιο αυτό αφορά τη συλλογή και την αποθήκευση των δεδομένων που πρόκειται να αναλυθούν.
- Data Preprocessing: Το στάδιο αυτό αφορά την προεπεξεργασία των δεδομένων με σκοπό την ανίχνευση και την αντιμετώπιση πιθανών σφαλμάτων που έχουν συμβεί κατά τη διαδικασία συλλογής.
- Data Transformation: Το στάδιο αυτό αφορά το μετασχηματισμό των δεδομένων. Σε αυτό το στάδιο, τα δεδομένα μετατρέπονται σε μια μορφή από την οποία μπορούν ευκολότερα να εξαχθούν συμπεράσματα.
- Data Mining: Το στάδιο αυτό αφορά την εξόρυξη γνώσης από τα μετασχηματισμένα δεδομένα. Συνήθως χρησιμοποιείται κάποιο μοντέλο μηχανικής μάθησης το οποίο προσπαθεί να αναγνωρίσει πιθανά μοτίβα στα δεδομένα.
- Interpretation/Evaluation: Το στάδιο αυτό περιλαμβάνει την ερμηνεία και την αξιολόγηση των αποτελεσμάτων που επιτεύχθηκαν από τη διαδικασία ανακάλυψης γνώσης.



Πηγή: https://infovis-wiki.net/w/images/4/4d/Fayyad96kdd-process.png

Σχήμα 3.6: Τα στάδια ανακάλυψης γνώσης

3.2.2 Προεπεξεργασία Δεδομένων (Data Preprocessing)

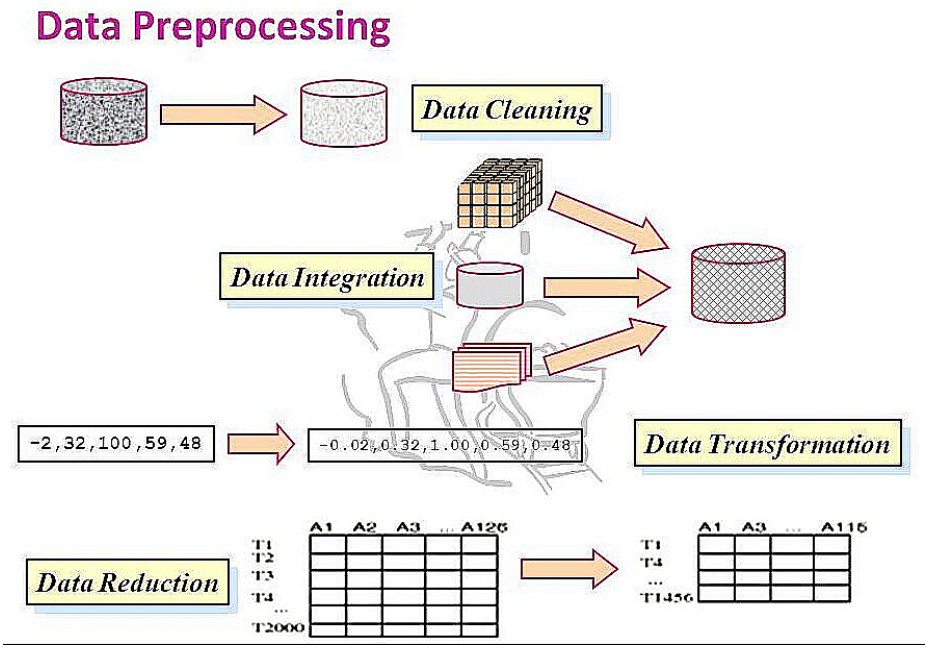
Η προεπεξεργασία δεδομένων [26] στοχεύει στην αντιμετώπιση προβλημάτων που έχουν προκύψει κατά τη συλλογή των δεδομένων και γενικότερα στη μετατροπή του συνόλου δεδομένων σε μια ποιοτικότερη μορφή η οποία θα μπορεί να αξιοποιηθεί πιο εύκολα. Πρόκειται για ένα από τα σημαντικότερα στάδια της ανακάλυψης γνώσης, το οποίο καθορίζει σε μεγάλο βαθμό την επιτυχία ή μη της όλης διαδικασίας. Η προεπεξεργασία δεδομένων ορίζεται από επιμέρους στάδια τα οποία είναι τα εξής:

- Data Cleaning: Το στάδιο αυτό αφορά τον καθαρισμό των δεδομένων. Πολλές φορές τα δεδομένα είναι "ακάθαρτα" υπό την έννοια ότι μπορεί να λείπουν τιμές, να περιέχουν θόρυβο κ.α. .

- Data Integration: Το στάδιο αυτό αφορά την ενοποίηση των δεδομένων. Τα δεδομένα μπορεί να προέρχονται από διαφορετικές βάσεις και τα ίδια χαρακτηριστικά να περιγράφονται με διαφορετικά πεδία. Μέσω της ενοποίησης εξασφαλίζεται ότι θα υπάρχει μια κοινή απεικόνιση που θα αναπαριστά τα δεδομένα.

- Data Transformation: Είναι το στάδιο που παρουσιάστηκε και παραπάνω. Κατά κοινή σύμβαση εντάσσεται στη διαδικασία προεπεξεργασίας των δεδομένων. Σε αυτό το στάδιο αναζητείται η μορφή των δεδομένων που επιτρέπει την αποδοτικότερη ανάλυσή τους.

- Data Reduction: Το στάδιο αυτό αφορά την μείωση των διαστάσεων των δεδομένων φροντίζοντας παράλληλα να μη χαθεί κανένα μέρος της πληροφορίας. Αυτό γίνεται προκειμένου να εξασφαλιστεί η ταχύτατη ανάλυση των δεδομένων.



Πηγή: https://slideplayer.com/slide/9285765/28/images/Transformation.jpg

Σχήμα 3.7: Τα στάδια της προεπεξεργασίας δεδομένων

3.2.2.1 Καθαρισμός Δεδομένων (Data Cleaning)

Ο καθαρισμός των δεδομένων [26] συνίσταται στη διαχείριση τιμών που λείπουν (missing values) από τα δεδομένα, στην εξομάλυνση του θορύβου στα δεδομένα (noise) και στον εντοπισμό ασυνεπών δεδομένων (outliers). Αναλυτικότερα:

- Απουσιάζουσες τιμές (Missing Values): Είναι πολύ πιθανό κατά τη συλλογή των δεδομένων να χαθούν κάποιες τιμές. Αυτό μπορεί να οφείλεται είτε σε ανθρώπινο λάθος είτε σε κάποιο σφάλμα λειτουργίας του αισθητήρα ή της μονάδας που συλλέγει τα δεδομένα. Για την αντιμετώπιση αυτού του προβλήματος έχουν οριστεί δύο προσεγγίσεις. Σύμφωνα με την πρώτη, όποια

εγγραφή περιέχει "κενή" τιμή δεν χρησιμοποείται στη διαδικασία ανακάλυψης γνώσης (ignore the tuple). Πρόκειται για μια απλή μέθοδο η οποία μπορεί να εφαρμοστεί πολύ εύκολα. Ωστόσο, δεν προτιμάται καθώς μπορεί να απορριφθεί μια ολόκληρη εγγραφή δεδομένων η οποία απλά δεν έχει τιμή για ένα πεδίο της, γεγονός που έχει ως αποτέλεσμα να χάνεται σημαντική πληροφορία. Για το λόγο αυτό προτιμάται η δεύτερη μέθοδος, σύμφωνα με την οποία το "κενό" πεδίο κάποιας εγγραφής συμπληρώνεται με μια τιμή. Αυτό μπορεί να γίνει με τους εξής τρόπους:

1. Η τιμή συμπληρώνεται χειροκίνητα από τον προγραμματιστή. Κάτι τέτοιο είναι πρακτικά αδύνατο σε περιπτώσεις που το πλήθος των τιμών που λείπουν είναι μεγάλο.
2. Η τιμή συμπληρώνεται με μια σταθερά τύπου "unknown value". Κάτι τέτοιο μπορεί να οδηγήσει σε εσφαλμένα αποτελέσματα και ερμηνείες, καθώς πεδία με εντελώς διαφορετικό νόημα θα περιέχουν την ίδια τιμή.
3. Για κάθε πεδίο όπου υπάρχει "κενή" τιμή υπολογίζεται η μέση τιμή του πεδίου με βάση τις υπόλοιπες εγγραφές του συνόλου δεδομένων. Στη συνέχεια, κάθε "κενή" τιμή συμπληρώνεται με τη μέση τιμή του αντίστοιχου πεδίου. Θα πρέπει να αναφερθεί ότι σε προβλήματα που αφορούν την κατηγοριοποίηση των δεδομένων η διαδικασία αυτή συμβαίνει ξεχωριστά για κάθε πιθανή κλάση (δηλαδή οι μέσες τιμές των πεδίων βρίσκονται ξεχωριστά για κάθε κλάση).
4. Για κάθε "κενή" τιμή μπορεί, μέσω συγκεκριμένων αλγορίθμων πρόβλεψης, να γίνει μια εκτίμηση της τιμής που θα έπρεπε να είχε και να συμπληρωθεί με αυτήν.

- Θορυβώδη δεδομένα (Noisy Data): Τα δεδομένα που έχουν συλλεχθεί μπορεί να περιέχουν θόρυβο. Ο θόρυβος παραμορφώνει τα δεδομένα και παρεμποδίζει την ορθή ανάλυση τους. Η εξομάλυνση του θορύβου μπορεί να επιτευχθεί με τους εξής τρόπους:

  1. Binning: Με αυτή τη τεχνική τα δεδομένα ταξινομούνται με βάση κάποιο χαρακτηριστικό και στη συνέχεια διαχωρίζονται σε μικρότερες ομάδες οι οποίες ονομάζονται bins. Τα bins που έχουν δημιουργηθεί πρέπει να περιέχουν το ίδιο πλήθος στοιχείων. Στη συνέχεια, για κάθε bin υπολογίζεται η μέση τιμή του εν λόγω χαρακτηριστικού η οποία και αντικαθιστά την τιμή των στοιχείων που ανήκουν στο εκάστοτε bin.
  2. Παλινδρόμηση (Regression): Σε αυτήν την περίπτωση η εξομάλυνση των δεδομένων επιτυγχάνεται μέσω παλινδρόμησης. Χαρακτηριστικό παράδειγμα αυτής της μεθόδου αποτελεί το φίλτρο Κάλμαν, το οποίο χρησιμοποιείται ευρέως ιδιαίτερα από εφαρμογές που αφορούν ευφυή συστήματα μεταφορών και παρουσιάζεται αναλυτικότερα στο υποκεφάλαιο 4.2.3.

- Ασυνεπή δεδομένα (Outliers): Ως outliers χαρακτηρίζονται δεδομένα τα οποία παρουσιάζουν εξαιρετικά περίεργες τιμές σε σχέση με δεδομένα με τα οποία θεωρούνται συγγενικά. Για παράδειγμα, σε ένα πρόβλημα κατηγοριοποίησης μπορεί δύο δείγματα να ανήκουν στην ίδια κλάση, ωστόσο να έχουν τιμές που παρουσιάζουν τεράστια απόκλιση. Κάτι τέτοιο αποτελεί εμπόδιο στην ανακάλυψη γνώσης καθώς δυσκολεύει την αναγνώριση μοτίβων. Για το λόγο αυτό προτείνεται η αφαίρεση των outliers από το σύνολο δεδομένων. Ο εντοπισμός των outliers επιτυγχάνεται μέσω της συσταδοποίησης (clustering) των

δεδομένων. Αυτό που συμβαίνει είναι ότι τα δεδομένα οργανώνονται σε μικρότερες ομάδες (clusters) με βάση τα χαρακτηριστικά τους. Έπειτα, εντοπίζονται οι outliers, τα δεδομένα που ανήκουν στη λάθος ομάδα δηλαδή, και τελικά αφαιρούνται.

3.2.2.2 Ενοποίηση Δεδομένων (Data Integration)

Η ενοποίηση των δεδομένων [26] είναι μια διαδικασία η οποία συνίσταται στην ένωση δεδομένων που προέρχονται από διαφορετικές πηγές, προκειμένου να σχηματιστεί ένα ολικό σύνολο δεδομένων. Κατά την ενοποίηση των δεδομένων μπορεί να προκύψουν διάφορα προβλήματα τα οποία θα πρέπει να αντιμετωπιστούν. Το βασικότερο πρόβλημα είναι η πλεονάζουσα πληροφορία (reduntant information) που εισάγεται στο σύνολο των δεδομένων κατά την ενοποίηση. Για παράδειγμα, μπορεί να υπάρχει η ίδια εγγραφή σε δύο διαφορετικές βάσεις απλά στη μια τα πεδία της εγγραφής να έχουν διαφορετικά ονόματα από ότι στην άλλην (double values). Προφανώς, η εγγραφή αυτή θα πρέπει να περιέχεται μια φορά στο τελικό σύνολο που προκύπτει από την ένωση των επιμέρους βάσεων.

3.2.2.3 Μετασχηματισμός Δεδομένων (Data Transformation)

Ο μετασχηματισμός των δεδομένων [26] συνίσταται στη μετατροπή των δεδομένων σε μια μορφή η οποία μπορεί να εκμεταλλευτεί ευκολότερα και στην οποία τα μοτίβα είναι πιο ξεκάθαρα. Συνεπώς, πρόκειται για μια πολύ σημαντική διαδικασία η οποία εξασφαλίζει την αποδοτική ανάλυση των δεδομένων και την αποτελεσματική εξαγωγή συμπερασμάτων από αυτά. Οι κυριότερες τεχνικές που χρησιμοποιούνται για το μετασχηματισμό των δεδομένων είναι οι εξής:

- Κανονικοποίηση (Normalization): Με τη κανονικοποίηση οι τιμές των δεδομένων κλιμακώνονται έτσι ώστε να περιοριστούν σε ένα εύρος τιμών. Δύο από τις πιο γνωστές μεθόδους κανονικοποίησης είναι οι εξής:

  1. Κανονικοποίηση Ελάχιστου-Μέγιστου (Min-Max Normalization): Έστω ότι min_A και max_A η ελάχιστη και η μέγιστη τιμή ενός χαρακτηριστικού A, αντίστοιχα. Η Min-Max κανονικοποίηση αντιστοιχίζει την τιμή v που έχει ένα δείγμα για το χαρακτηριστικό A στο διάστημα $[newMin_A, newMax_A]$ με βάση τον τύπο:

  $$v' = \frac{v - min_A}{max_A - min_A} \cdot (newMax_A - newMin_A) + newMin_A \tag{3.1}$$

  2. Κανονικοποίηση Z (Z-Normalization): Έστω ότι μ_A και σ_A η μέση τιμή και η τυπική απόκλιση ενός χαρακτηριστικού A, αντίστοιχα. Η νέα τιμή v' που έχει ένα δείγμα για το χαρακτηριστικό A δίνεται από τον τύπο:

  $$v' = \frac{v - \mu_A}{\sigma_A} \tag{3.2}$$

- Συνάθροιση χαρακτηριστικών (Feature Aggregation): Με τη συνάθροιση χαρακτηριστικών δημιουργείται μια σύνοψη για ένα σύνολο χαρακτηριστικών. Ουσιαστικά δημιουργείται ένα νέο χαρακτηριστικό, το οποίο περιεχέι συνοπτικά τη πληροφορία όλων των χαρακτηριστικών που αντικατέστησε. Θα πρέπει να αναφερθεί ότι αυτή η μέθοδος μπορεί να θεωρηθεί κομμάτι και του σταδίου data reduction.

- Κατασκευή νέων χαρακτηριστικών (Feature Construction): Η μέθοδος αυτή χρησιμοποιείται προκειμένου να δημιουργηθεί ένα νέο χαρακτηριστικό, με βάση τα ήδη υπάρχοντα, το οποίο θα εκφράζει καλύτερα τις ιδιότητες των δεδομένων. Πρόκειται για μια πολύ σημαντική τεχνική, καθώς όσο πιο εύστοχη είναι η περιγραφή των δεδομένων τόσο πιο αποτελεσματική θα είναι η ανακάλυψη γνώσης και η εξαγωγή συμπερασμάτων.

- Κωδικοποίηση χαρακτηριστικών (Feature Encoding): Προκειμένου να εξασφαλιστεί η σωστή λειτουργία του σταδίου της εξόρυξης γνώσης συνίσταται η μετατροπή των κατηγορικών δεδομένων (δεδομένα τύπου string) σε αριθμητική αναπαράσταση. Τα κατηγορικά δεδομένα (categorical data) μπορεί να είναι είτε ordinal (παίρνουν τιμές από ένα διατεταγμένο σύνολο) είτε nominal (παίρνουν τιμές από ένα σύνολο στο οποίο δεν υπάρχει διάταξη). Κάποιες από τις πιο γνωστές τεχνικές κωδικοποίησης είναι οι εξής:

  1. Label Encoding: Η τεχνική αυτή χρησιμοποιείται για ordinal categorical data. Κάθε τιμή του διατεταγμένου συνόλου τιμών αναπαρίσταται με έναν μοναδικά προσδιορίσιμο ακέραιο αριθμό.
  2. One-Hot Encoding: Η τεχνική αυτή χρησιμοποιείται για nominal categorical data. Κάθε τιμή του συνόλου τιμών μετατρέπεται σε χαρακτηριστικό που προστίθεται στα δεδομένα. Αν κάποιο δείγμα έχει το νέο αυτό χαρακτηριστικό ορίζει την τιμή του ίση με 1, ενώ σε αντίθετη περίπτωση ορίζει την τιμή του ίση με 0.

3.2.2.4 Μείωση Δεδομένων (Data Reduction)

Η μείωση των δεδομένων [26] εξασφαλίζει τον μειωμένο όγκο αναπαράστασης των δεδομένων φροντίζοντας να μην χαθεί κανένα μέρος της πληροφορίας. Είναι μια διαδικασία η οποία αποσκοπεί στη μείωση του πλήθους των χαρακτηριστικών που απαιτούνται για τη περιγραφή των δεδομένων, συμβάλλοντας έτσι στη γρηγορότερη ανάλυση των δεδομένων. Οι κυριότερες τεχνικές που χρησιμοποιούνται για τη μείωση των δεδομένων είναι οι εξής:

- Μείωση διαστασιμότητας (Dimensionality Reduction): Τα δεδομένα απλοποιούνται και αντιστοιχίζονται σε ένα χώρο λιγότερων διαστάσεων από αυτόν που ανήκαν πριν. Κάποιες από τις πιο γνωστές τεχνικές μείωσης διαστάσεων των δεδομένων είναι οι εξής:

  1. Principal Components Analysis (PCA): Ο PCA [27] είναι ένας αλγόριθμος ο οποίος προέρχεται από το πεδίο των Στατιστικών Επιστημών. Η εφαρμογή του αλγορίθμου πάνω σε ένα σύνολο δεδομένων οδηγεί στην δημιουργία νέων χαρακτηριστικών τα οποία εκφράζονται σε χώρο χαμηλότερων διαστάσεων.
  2. Επιλογή χαρακτηριστικών (Feature Selection): Περιλαμβάνει ενέργειες που αφορούν την επιλογή των χαρακτηριστικών τα οποία είναι πιο σημαντικά για την εξαγωγή γνώσης από τα εκάστοτε δεδομένα. Τα χαρακτηριστικά τα οποία δεν θεωρείται ότι είναι τόσο κρίσιμα για την περιγραφή των δεδομένων αποβάλλονται από το σύνολο δεδομένων.

- Data Cube Aggregation: Πρόκειται για μια μέθοδο κατά την οποία εγγραφές του συνόλου δεδομένων ομαδοποιούνται προκειμένου να μειωθεί ο όγκος των δεδομένων. Για παράδειγμα, δεδομένα που έχουν τη μορφή ακολουθίας θα

μπορούσαν να συναθροιστούν ανά δύο (χρησιμοποιώντας τη μέση τιμή των παλιών τιμών για τα νέα χαρακτηριστικά), γεγονός που θα μείωνε τον όγκο των δεδομένων στο μισό.

3.2.3 Εξόρυξη Δεδομένων (Data Mining)

Η εξόρυξη δεδομένων [28] είναι το στάδιο που ακολουθεί αφού έχει ολοκληρωθεί η προεπεξεργασία των δεδομένων. Αυτό σημαίνει ότι τα δεδομένα έχουν έρθει σε κατάλληλη μορφή αναπαράστασης και επομένως μπορεί να ξεκινήσει η διαδικασία αναγνώρισης μοτίβων και εξαγωγής συμπερασμάτων από αυτά. Γενικότερα, όταν πρόκειται για δεδομένα μεγάλης κλίμακας η έννοια του Data Mining είναι στενά συνδεδεμένη με την έννοια της μηχανικής μάθησης (Machine Learning), καθώς ο όγκος και η ποικιλομορφία της πληροφορίας δεν επιτρέπουν την ορθή ανάλυση των δεδομένων από τις παραδοσιακές μεθόδους. Σε αυτήν την περίπτωση λοιπόν, η διαδικασία περιλαμβάνει την εφαρμογή κάποιου αλγορίθμου μέσω του οποίου σχηματίζεται ένα μοντέλο μηχανικής μάθησης, το οποίο στη συνέχεια εξάγει συμπεράσματα από τα δεδομένα με αυτοματοποιημένο τρόπο. Συνεπώς, θα μπορούσε να πει κανείς ότι η εξόρυξη γνώσης από μεγάλα δεδομένα είναι μια διαδικασία που εφαρμόζει την επιστήμη της μηχανικής μάθησης, η οποία αναλύεται με μεγάλη λεπτόμερεια στο υποκεφάλαιο 4.1, προκειμένου να αναλύσει μεγάλα δεδομένα με αποτελεσματικό τρόπο.

3.2.4 Ερμηνεία και Αξιολόγηση (Interpretation/Evaluation)

Αφού ολοκληρωθεί το στάδιο εξόρυξης γνώσης από τα δεδομένα, ακολουθεί η ερμηνεία και η αξιολόγηση των αποτελεσμάτων που επιτεύχθηκαν. Το στάδιο αυτό περιλαμβάνει την επαλήθευση της ορθότητας των μοτίβων που αναγνωρίστηκαν και της γνώσης που ανακαλύφθηκε μέσω της εφαρμογής τους σε ένα νέο σύνολο δεδομένων με παρόμοια μοτίβα, το οποίο δεν είχε χρησιμοποιηθεί στο στάδιο εξόρυξης γνώσης. Η αξιολόγηση προκύπτει από τη σύγκριση των αναμενόμενων και των εκτιμώμενων αποτελεσμάτων, ενώ η ερμηνεία των αποτελεσμάτων δίνεται μέσω γραφικών παραστάσεων.

Κεφάλαιο 4: Απαραίτητο Θεωρητικό Υπόβαθρο

4.1 Μηχανική Μάθηση (Machine Learning)

Η μηχανική μάθηση (Machine Learning) είναι ένας κλάδος της επιστήμης των υπολογιστών που επικεντρώνεται στην δημιουργία έξυπνων και αυτοματοποιημένων συστημάτων. Τα συστήματα αυτά καλούνται μοντέλα μηχανικής μάθησης. Πρόκειται για εφαρμογές οι οποίες έχουν τη δυνατότητα να μάθουν από την εμπειρία που αποκτούν μέσω της αλληλεπίδρασής τους με το περιβάλλον και τα δεδομένα με τα οποία τροφοδοτούνται. Καθώς εκπαιδεύεται, το μοντέλο μαθαίνει από τα δεδομένα που του παρέχει ο προγραμματιστής και όχι από κάποιο σύνολο κανόνων/ενεργειών που αυτός θα μπορούσε να έχει ορίσει. Ο απώτερος σκοπός είναι η δημιουργία έξυπνων συστημάτων που μαθαίνουν από μόνα τους πως να εκτελούν ενέργειες, χωρίς να υπάρχει κάποια ανθρώπινη παρέμβαση.

4.1.1 Βασικές Έννοιες

Οι τεχνικές που χρησιμοποιούνται στους αλγορίθμους μηχανικής μάθησης χωρίζονται σε τρεις μεγάλες κατηγορίες:

- Επιβλεπόμενη μάθηση (Supervised Machine Learning): Με αυτήν την τεχνική, το μοντέλο τροφοδοτείται με δεδομένα εισόδου για τα οποία η επιθυμητή έξοδος είναι γνωστή. Από την ανάλυση των δεδομένων αυτών, το μοντέλο σχηματίζει από μόνο του ένα σύνολο κανόνων με βάση το οποίο λειτουργεί. Με αυτόν τον τρόπο, είναι σε θέση να προβλέψει την έξοδο για νέα άγνωστα δεδομένα που τροφοδοτούνται σε αυτό.

- Μη επιβλεπόμενη μάθηση (Unsupervised Machine Learning): Με αυτήν την τεχνική, το μοντέλο τροφοδοτείται με δεδομένα εισόδου για τα οποία η επιθυμητή έξοδος δεν είναι γνωστή. Το μοντέλο καλείται να ενεργήσει μόνο του πάνω στο σύνολο δεδομένων που του δίνεται και να βρει μοτίβα που μπορεί να υπάρχουν σε αυτό. Με αυτόν τον τρόπο, είναι σε θέση να προβλέψει την έξοδο για νέα άγνωστα δεδομένα που τροφοδοτούνται σε αυτό.

- Ενισχυτική μάθηση (Reinforcement Machine Learning): Με αυτήν την τεχνική, το μοντέλο αφήνεται να λειτουργήσει εντελώς αυτόνομα με σκοπό να μάθει δυναμικά από το περιβάλλον του. Αυτό επιτυγχάνεται μέσω δοκιμών και ενεργειών που εκτελεί το μοντέλο και για τις οποίες παίρνει ανάδραση. Η ανάδραση μπορεί να έχει τη μορφή επιβράβευσης (reward), εάν η ενέργεια που εκτελέστηκε ήταν η επιθυμητή, ή τη μορφή ποινής (penalty) σε αντίθετη περίπτωση.

Τα βασικότερα προβλήματα τα οποία καλούνται να επιλύσουν οι αλγόριθμοι μηχανικής μάθησης μπορούν να χωριστούν στις εξής κατηγορίες:

- Ταξινόμηση (Classification): Πρόκειται για προβλήματα κατηγοριοποίησης. Σε αυτήν την περίπτωση τα δεδομένα ανήκουν σε μια κατηγορία/ομάδα (class), με βάση τα χαρακτηριστικά (features) που αυτά έχουν. Κάθε μια από αυτές τις κατηγορίες χαρακτηρίζεται από μια μοναδική ετικέτα (label). Το μοντέλο εκπαιδεύεται πάνω σε ένα σύνολο δεδομένων εκπαίδευσης (training set) προκειμένου να μπορέσει να βρει την κατάλληλη ετικέτα για νέα δεδομένα (σύνολο δεδομένων ελέγχου - test set) που δίνονται ως είσοδος σε αυτό. Κατά τη φάση εκπαίδευσης του μοντέλου (training phase) τα βασικά προβλήματα που μπορεί να προκύψουν είναι η υποπροσαρμογή (underfitting) και η υπερπροσαρμογή (overfitting). Στην περίπτωση της υποπροσαρμογής σχηματίζεται ένα πολύ γενικό μοντέλο το οποίο δεν έχει καταφέρει να εντοπίσει τα μοτίβα που κρύβουν τα δεδομένα εκπαίδευσης, ενώ στην περίπτωση της υπερπροσαρμογής σχηματίζεται ένα πολύ συγκεκριμένο μοντέλο το οποίο έχει μοντελοποιήσει υπερβολικά τα δεδομένα εκπαίδευσης και δεν έχει την ικανότητα γενίκευσης. Κλείνοντας, θα πρέπει να αναφερθεί ότι τα προβλήματα ταξινόμησης διακρίνονται σε επιμέρους κατηγορίες ανάλογα με το πλήθος των πιθανών κλάσεων στις οποίες μπορεί να ανήκει κάποια είσοδος. Ο όρος δυαδική ταξινόμηση (binary classification) αναφέρεται σε προβλήματα που το πλήθος των πιθανών ετικετών κατηγοριοποίησης είναι ίσο με δύο, ενώ ο όρος πολυ-κατάταξη ταξινόμηση (multiclass classification) αναφέρεται σε προβλήματα που το πλήθος των πιθανών ετικετών κατηγοριοποίησης είναι μεγαλύτερο του δύο.

- Συσταδοποίηση (Clustering): Πρόκειται για προβλήματα ομαδοποίησης. Τα προβλήματα αυτά είναι αρκετά παρόμοια με αυτά της ταξινόμησης, με τη μόνη διαφορά να είναι ότι τα δεδομένα δεν έχουν κάποια προκαθορισμένη ετικέτα εξ αρχής. Το μοντέλο καλείται να βρει ομοιότητες στο σύνολο δεδομένων που του παρέχεται και να το χωρίσει σε μικρότερες ομάδες (clusters) των οποίων τα μέλη έχουν όμοια χαρακτηριστικά.

- Παλινδρόμηση (Regression): Πρόκειται για προβλήματα που αποσκοπούν στην πρόβλεψη της τιμής μια συνεχούς μεταβλητής. Το μοντέλο σε αυτήν την περίπτωση δεν ψάχνει να βρει την κατάλληλη ετικέτα ή ομάδα, αλλά κάποια αριθμητική τιμή. Αυτό επιτυγχάνεται αναλύοντας την κατανομή που ακολουθεί η τυχαία μεταβλητή καθώς και τιμές που έχει λάβει στο παρελθόν.

4.1.2 Νευρωνικά Δίκτυα (Neural Networks)

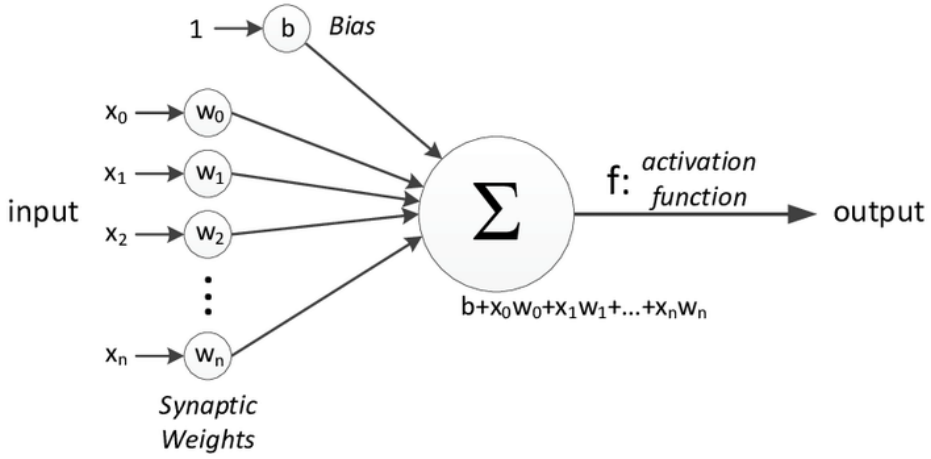
Τα νευρωνικά δίκτυα (Neural Networks) αποτελούν ένα από τα σημαντικότερα υποσύνολα της μηχανικής μάθησης. Ως νευρωνικό δίκτυο χαρακτηρίζεται ένα κύκλωμα το οποίο αποτελείται από έναν αριθμό διασυνδεδεμένων υπολογιστικών μονάδων οι οποίοι ονομάζονται νευρώνες. Η ονομασία του είναι εμπνευσμένη από το κεντρικό νευρικό σύστημα του ανθρώπινου οργανισμού, τη λειτουργία του οποίου προσπαθεί και να προσομοιώσει. Αυτό έγκειται στο γεγονός ότι το δίκτυο λαμβάνει γνώση από το περιβάλλον του και επιχειρεί να την ερμηνεύσει και να παράγει νέα γνώση. Στόχος των νευρωνικών δικτύων είναι η αναγνώριση προτύπων που κρύβονται σε σύνολα δεδομένων και η απόκτησης εμπειρικής γνώσης μέσα από αυτά.

4.1.2.1 Νευρώνας (Perceptron)

Οι νευρώνες (Perceptrons) αποτελούν δομικά στοιχεία των νευρωνικών δικτύων. Η ονομασία τους προκύπτει από τους νευρώνες του ανθρώπινου εγκέφαλου που δέχονται

ερεθίσματα, τα επεξεργάζονται και στη συνέχεια παράγουν το αντίστοιχο σήμα εξόδου. Στον τομέα της επιστήμης των υπολογιστών ο νευρώνας ορίζεται ως μια υπολογιστική μονάδα, η οποία δέχεται ως είσοδο ένα διάνυσμα $x = [x_1, x_2, ..., x_n]$ το οποίο πολλαπλασιάζεται με ένα διάνυσμα βαρών (weights) $w = [w_1, w_2, ..., w_n]$. Στην είσοδο συμπεριλαμβάνεται και μια σταθερά η οποία καλείται bias και συμβολίζεται με b. Τελικά, ο νευρώνας λαμβάνει ως είσοδο το διάνυσμα $[w_1 \cdot x_1, w_2 \cdot x_2, ..., w_n \cdot x_n, b]$. Η έξοδος $\hat{y}$ λαμβάνεται από την εφαρμογή μιας συνάρτησης ενεργοποίησης (activation function) πάνω στο άθροισμα των επιμέρους στοιχείων του διανύσματος εισόδου. Συνεπώς, η έξοδος του νευρώνα δίνεται από την εξίσωση:

$$\hat{y} = f((\sum_{i=1}^{n} w_i \cdot x_i) + b) \tag{4.1}$$



Πηγή: https://www.researchgate.net/profile/Perceptron-30.png

Σχήμα 4.1: Ο νευρώνας (perceptron)

Η συνάρτηση ενεργοποίησης είναι αυτή που καθορίζει την έξοδο του νευρώνα. Οι συναρτήσεις ενεργοποίησης που χρησιμοποιούνται πιο συχνά στα νευρωνικά δίκτυα είναι οι εξής [29]:

- Σιγμοειδής (Sigmoid) Η έξοδος της σιγμοειδούς συνάρτησης κυμαίνεται στο διάστημα [0, 1], γεγονός που τη καθιστά ιδιαίτερα αποτελεσματική σε προβλήματα ταξινόμησης (όπου και γίνεται χρήση πιθανοτήτων).

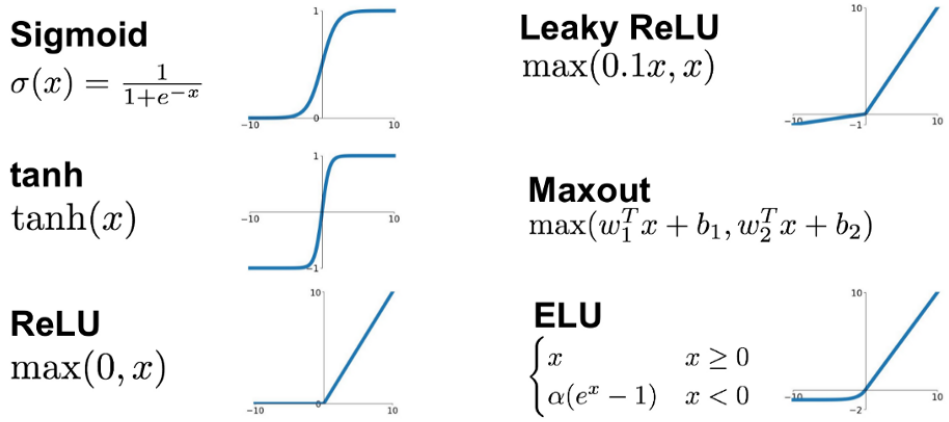
- Υπερβολική εφαπτομένη (tanh) Η συνάρτηση υπερβολικής εφαπτομένης μοιάζει αρκετά με τη σιγμοειδή συνάρτηση, με τη διαφορά ότι είναι κεντραρισμένη στο μηδέν και επομένως η έξοδος της κυμαίνεται στο διάστημα [-1, 1]. Το γεγονός αυτό διευκολύνει τη φάση εκπαίδευσης.

- Γραμμική ανόρθωση (Rectified Linear Unit – ReLU) Πρόκειται για μια μη γραμμική συνάρτηση η οποία ουσιαστικά κόβει τις εισόδους που έχουν αρνητική τιμή. Η χρήση της συνάρτησης αυτής μπορεί να οδηγήσει σε ενημέρωση των βαρών με τέτοιο τρόπο έτσι ώστε κάποιος νευρώνας να μην ενεργοποιείται ποτέ, ανεξαρτήτως της εισόδου (πρόβλημα του νεκρού νευρώνα [29]).

- Διαρρέουσα γραμμική ανόρθωση (Leaky ReLU) Η συνάρτηση αυτή μοιάζει με τη ReLU και χρησιμοποιείται για να αντιμετωπιστεί το πρόβλημα του νεκρού νευρώνα. Αυτό επιτυγχάνεται με τη γραμμική απόδοση μικρών τιμών σε αρνητικές εισόδους.

- Εκθετική γραμμική ανόρθωση (Exponential Linear Unit - ELU) Η συνάρτηση αυτή μοιάζει με τη ReLU και χρησιμοποιείται για να αντιμετωπιστεί το πρόβλημα του νεκρού νευρώνα. Αυτό επιτυγχάνεται με τη εκθετική απόδοση μικρών τιμών σε αρνητικές εισόδους.

- Maxout Η συνάρτηση αυτή αποτελεί γενίκευση των συναρτήσεων ReLU και Leaky ReLU.



Πηγή: https://cdn-images-1.medium.com/max/1000/1*4ZEDRpFuCIpUjNgjDdT2Lg.png

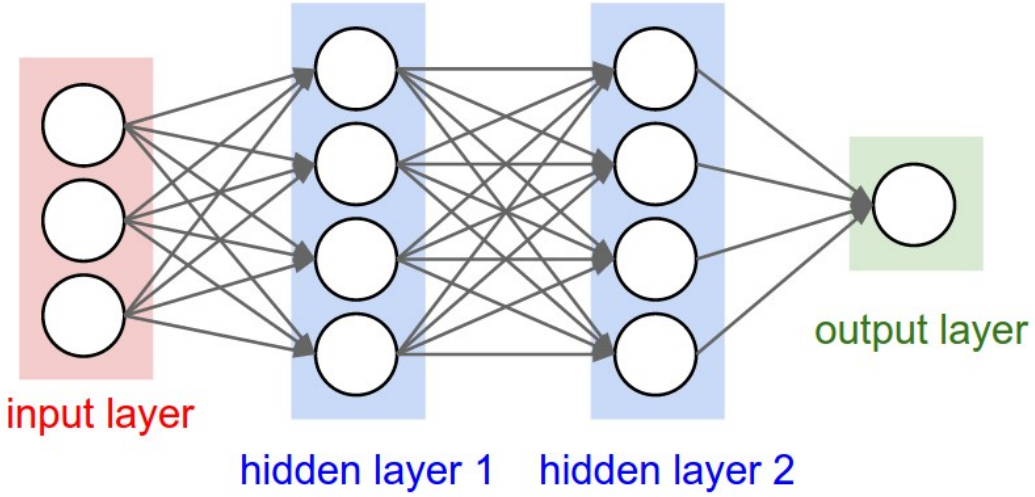
Σχήμα 4.2: Συναρτήσεις ενεργοποίησης

4.1.2.2 Κατηγοριοποίηση Νευρωνικών Δικτύων

Πολλοί νευρώνες συνδέονται μεταξύ τους μέσω συνάψεων (synapses) προκειμένου να σχηματίσουν νευρωνικά δίκτυα. Αυτή η σύνδεση των νευρώνων οδηγεί στη δημιουργία μιας πιο σύνθετης τοπολογίας η οποία μπορεί να διακριθεί σε περαιτέρω επίπεδα. Το πρώτο επίπεδο καλείται επίπεδο εισόδου (input ή visible layer) και αναλαμβάνει να προωθήσει την είσοδο στο υπόλοιπο δίκτυο, χωρίς να πραγματοποιήσει κανέναν υπολογισμό πάνω σε αυτήν. Συνήθως, το επίπεδο εισόδου έχει ένα νευρώνα για κάθε μεταβλητή του διανύσματος εισόδου. Το τελευταίο επίπεδο καλείται επίπεδο εξόδου (output layer) και είναι υπεύθυνο για την ανάδειξη της τιμής εξόδου. Σε προβλήματα ταξινόμησης (όπως είναι αυτό που μελετάται στην παρούσα διπλωματική) το επίπεδο εξόδου περιέχει ένα νευρώνα για κάθε κλάση κατηγοριοποίησης. Ανάμεσα στα επίπεδα εισόδου και εξόδου υπάρχουν κι άλλα επίπεδα τα οποία καλούνται κρυφά επίπεδα (hidden layers) καθώς δεν είναι άμεσα εκτεθειμένα στο διάνυσμα εισόδου.

Με βάση την αρχιτεκτονική τους, τα νευρωνικά δίκτυα μπορούν να διακριθούν σε δύο μεγάλες κατηγορίες:

- Νευρωνικά δίκτυα πρόσθιας τροφοδότησης (Feedforward Neural Networks - FNN): Στα νευρωνικά δίκτυα πρόσθιας τροφοδότησης [30], τα οποία ονομάζονται και Multi Layer Perceptrons ή MLP, τα γειτονικά επίπεδα συνδέονται μεταξύ τους με μονόδρομους συνδέσμους που έχουν κατεύθυνση από το χαμηλότερο προς το υψηλότερο επίπεδο. Πιο συγκεκριμένα, η έξοδος κάθε νευρώνα του επιπέδου i τροφοδοτείται ως είσοδος σε κάθε νευρώνα του αμέσως επόμενου επίπεδου $i + 1$. Η διαδικασία αυτή επαναλαμβάνεται μέχρι το επίπεδο εξόδου να λάβει και αυτό την είσοδο του και να παράγει την κατάλληλη έξοδο.



Πηγή: https://cs231n.github.io/assets/nn1/neural_net2.jpeg

Σχήμα 4.3: Νευρωνικό δίκτυο πρόσθιας τροφοδότησης 2 κρυφών επιπέδων

- Αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks - RNN): Τα αναδρομικά νευρωνικά δίκτυα [31] αποτελούνται επιπλέον από βρόχους ανάδρασης που τους δίνουν τη δυνατότητα να συνδέουν προηγούμενες εισόδους με την τρέχουσα. Η ιδιότητα αυτή δίνει την εντύπωση ύπαρξης μνήμης και τα καθιστά πολύ αποτελεσματικά σε προβλήματα όπου η είσοδος έχει τη μορφή ακολουθίας. Ένα αναδρομικό νευρωνικό δίκτυο ισοδυναμεί με πολλαπλά δίκτυα πρόσθιας τροφοδότησης τα οποία ανταλλάσσουν πληροφορίες με ακολουθιακό τρόπο. Για να γίνει καλύτερα κατανοητό αυτό, παρουσιάζεται ένα παράδειγμα. Στο αριστερό μέρος του σχήματος 4.4 απεικονίζεται ένα αναδρομικό δίκτυο ενός κρυφού επιπέδου, ενώ στο δεξί ένα ισοδύναμο δίκτυο το οποίο αποτελείται από τέσσερα νευρωνικά δίκτυα πρόσθιας τροφοδότησης. Το αναδρομικό δίκτυο στο αριστερό μέρος του σχήματος 4.4 τροφοδοτείται με ακολουθιακή είσοδο 4 στοιχείων ($i[1-4]$) τα οποία επεξεργάζεται επαναληπτικά, μέσω του βρόχου ανάδρασης, και παράγει έξοδο o_4. Σε κάθε επανάληψη το κρυφό επίπεδο λαμβάνει πληροφορίες για την κατάσταση της αμέσως προηγούμενης επανάληψης. Στο δεξί μέρος του σχήματος 4.4 απεικονίζεται το εσωτερικό του αναδρομικού δικτύου όταν αυτό "ανοιχτεί". Αποτελείται από 4 πανομοιότυπα (ίδια βάρη και ίδια συνάρτηση ενεργοποίησης) δίκτυα πρόσθιας τροφοδότησης τα οποία συνδέονται ακολουθιακά. Κάθε ένα από αυτά καλείται κρυφή μονάδα (hidden unit) ή κελί (cell). Κάθε κελί μεταβιβάζει την κρυφή του κατάσταση (hidden state) στο αμέσως επόμενο (το πρώτο κελί δεν τροφοδοτείται με τη κρυφή κατάσταση κάποιου άλλου, ενώ το τελευταίο δεν μεταβιβάζει τη κρυφή του κατάσταση σε κάποιο άλλο). Θα πρέπει να αναφερθεί πως στο συγκεκριμένο παράδειγμα η κρυφή κατάσταση ταυτίζεται με την έξοδο κάθε κελιού καθώς το αναδρομικό νευρωνικό δίκτυο που μελετάται αποτελείται από ένα κρυφό επίπεδο. Συνεπώς, προκύπτουν οι εξής σχέσεις:

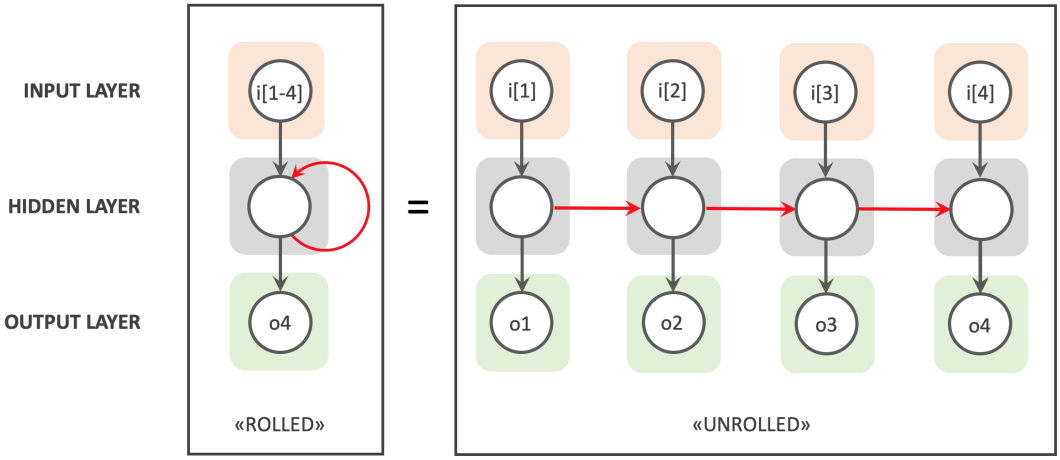
$$o_1 = f(i[1] \cdot W_i + b), o_1 = h_1 \tag{4.2}$$

$$o_2 = f(i[2] \cdot W_i + h_1 \cdot W_h + b), o_2 = h_2 \tag{4.3}$$

$$o_3 = f(i[3] \cdot W_i + h_2 \cdot W_h + b), o_3 = h_3 \tag{4.4}$$

$$o_4 = f(i[4] \cdot W_i + h_3 \cdot W_h + b) \tag{4.5}$$

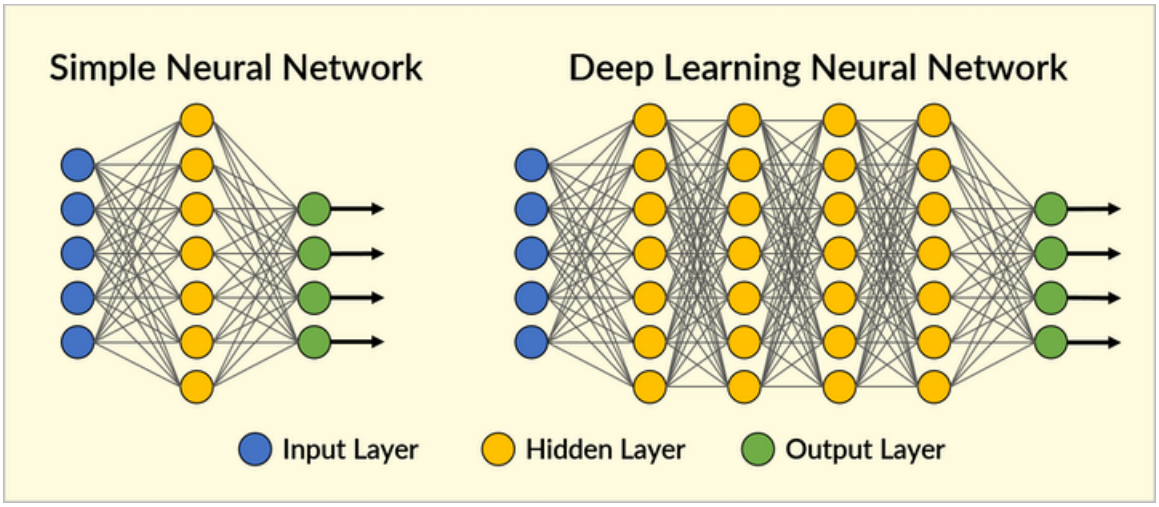
όπου με o συμβολίζεται η έξοδος, με i η είσοδος, με h η κρυφή κατάσταση, με $f(\cdot)$ η συνάρτηση ενεργοποίησης και με b η σταθερά bias. Με W_i συμβολίζεται ο πίνακας των βαρών που πολλαπλασιάζεται με το διάνυσμα εισόδου. Οι διαστάσεις αυτού του πίνακα καθορίζονται από το πλήθος των δεδομένων εισόδου και από το πλήθος των χαρακτηριστικών που έχει το κάθε δείγμα. Με W_h συμβολίζεται ο πίνακας των βαρών που πολλαπλασιάζονται με τη κρυφή κατάσταση του προηγούμενου κελιού. Οι διαστάσεις αυτού του πίνακα καθορίζονται από το πλήθος των δεδομένων εισόδου και από μια παράμετρο που ορίζεται από τον προγραμματιστή και προσδιορίζει το μέγεθος της κρυφής κατάστασης (hidden state size).



Πηγή: https://www.bouvet.no/bouvet-deler/explaining-recurrent-neural-networks/2016.31.24.png

Σχήμα 4.4: Αναδρομικό νευρωνικό δίκτυο με 1 κρυφό επίπεδο

Σε αυτό το σημείο θα πρέπει να αναφερθεί ότι νευρωνικά δίκτυα τα οποία έχουν μεγάλο αριθμό κρυφών επιπέδων ονομάζονται βαθιά νευρωνικά δίκτυα (Deep Neural Networks - DNN) [32]. Τα DNN ανήκουν σε ένα υποσύνολο της μηχανικής μάθησης το οποίο ονομάζεται βαθιά μάθηση (Deep Learning), λόγω του βάθους και της πολυπλοκότητας που κρύβεται πίσω από την αρχιτεκτονική τους. Η χρήση περισσότερων κρυφών επιπέδων καθιστά τη διαχείριση μεγάλου όγκου δεδομένων πιο αποτελεσματική, ενώ παράλληλα αυξάνει και την απόδοση του δικτύου.



Πηγή: https://img.securityinfowatch.com/files/base/cygnus/siw/image/2019/02/Figure_01.5c7712513151e.png

Σχήμα 4.5: Βαθιά νευρωνικά δίκτυα (Deep Neural Networks)

4.1.2.3 Διαδικασία Εκμάθησης

Τα νευρωνικά δίκτυα εκπαιδεύονται πάνω σε κάποιο σύνολο δεδομένων έτσι ώστε όταν τροφοδοτηθούν με άγνωστα δεδομένα να μπορούν να κάνουν προβλέψεις και να παράγουν σωστές εξόδους. Ο τρόπος με τον οποίο ένα νευρωνικό δίκτυο μαθαίνει συνδέεται άμεσα με την εύρεση των τιμών που πρέπει να έχουν τα βάρη των εισόδων στα διάφορα επίπεδα. Τα βάρη εκφράζουν την αλληλεπίδραση που υπάρχει μεταξύ των νευρώνων του δικτύου και καθορίζουν άμεσα την προβλεπόμενη τιμή. Συνεπώς, η φάση εκπαίδευσης αφορά ουσιαστικά την εφαρμογή τεχνικών για την εύρεση των τιμών των βαρών που δίνουν το αποδοτικότερο μοντέλο [33].

Για το σκοπό αυτό είναι απαραίτητη μια συνάρτηση κόστους (cost function). Η έξοδος της συνάρτησης κόστους εκφράζει το μέτρο της λάθους πρόβλεψης του μοντέλου. Η χρήση της στοχεύει στην ελαχιστοποίηση του σφάλματος, της απόκλισης ,δηλαδή, που υπάρχει μεταξύ της αναμενόμενης τιμής και της τιμής που προέβλεψε το μοντέλο. Συμβολίζοντας με $\hat{y}$ την έξοδο του νευρωνικού δικτύου και με y την αναμενόμενη έξοδο, μερικές από τις πιο συχνά χρησιμοποιούμενες συναρτήσεις κόστους είναι οι εξής:

- MSE:
$$MSE = \frac{1}{2} \cdot \sum (y - \hat{y})^2 \tag{4.6}$$
- Cross-Entropy:
$$CE = -\sum (\hat{y} \cdot \ln(y) + (1 - \hat{y}) \cdot \ln(1 - y)) \tag{4.7}$$
- Exponential Cost:
$$EC = \tau \cdot \hat{y}^{\frac{1}{\tau} \cdot \sum (y - \hat{y})^2} \tag{4.8}$$
- Hellinger Distance:
$$HD = \frac{1}{\sqrt{2}} \cdot \sum (\sqrt{y} - \sqrt{\hat{y}})^2 \tag{4.9}$$

- Kullback-Leibner Divergence:

$$KLD = \sum(\hat{y} \cdot \log(\hat{y}) - \hat{y} \cdot \log(y)) \tag{4.10}$$

Σε πολλές περιπτώσεις εισάγεται ένας επιπλέον όρος κανονικοποίησης στη συνάρτηση κόστους. Η κανονικοποίηση είναι μια τεχνική που χρησιμοποιείται για να περιοριστούν οι μεγάλες αποκλίσεις που μπορεί να παρουσιάζουν τα δεδομένα εισόδου και βοηθάει στο σχηματισμό ενός μοντέλου που μπορεί να γενικεύσει πιο εύκολα. Με αυτόν τον τρόπο ενισχύεται η αντιμετώπιση του φαινομένου της υπερπροσαρμογής. Δύο από τις πιο γνωστές μεθόδους κανονικοποίησης είναι οι εξής [34]:

- Lasso Regularization (L1 Regularization):

$$C(y, \hat{y}) + \alpha \cdot \sum_{i=1}^{n} |w_i| \tag{4.11}$$

- Ridge Regularization (L2 Regularization):

$$C(y, \hat{y}) + \alpha \cdot \sum_{i=1}^{n} w_i^2 \tag{4.12}$$

όπου $\hat{y} = f((\sum_{i=1}^{n} w_i \cdot x_i) + b)$ η προβλεπόμενη έξοδος και $C(\cdot)$ η συνάρτηση κόστους. Η κανονικοποίηση επιτυγχάνεται εισάγοντας έναν όρο που έχει μορφή ποινής και περιορίζει τις τιμές που μπορούν να πάρουν τα βάρη. Στην ουσία παρεμποδίζει τα βάρη να λάβουν υψηλές τιμές κατά τη φάση εκπαίδευσης. Η παράμετρος α καθορίζει το μέγεθος της ποινής που εισάγεται.

Είναι προφανές ότι όσο μικρότερη είναι η τιμή της συνάρτησης κόστους (συνυπολογίζοντας και την ποινή), τόσο μικρότερο είναι και το λάθος εκτίμησης του μοντέλου. Προκειμένου να ελαχιστοποιηθεί η συνάρτηση κόστους χρησιμοποιείται μια τεχνική που ονομάζεται βαθμωτή κατάβαση (Gradient Descent) [35]. Πρόκειται για έναν αλγόριθμο βελτιστοποίησης ο οποίος μέσω μιας επαναληπτικής διαδικασίας υπολογίζει το τοπικό ελάχιστο της συνάρτησης κόστους. Στηρίζεται στο γεγονός ότι το ελάχιστο κάποιας συνάρτησης μπορεί να βρεθεί κάνοντας βήματα με κατεύθυνση προς την αρνητική κλίσης της συνάρτησης. Αναλυτικότερα, η διαδικασία που εκτελεί ο αλγόριθμος βαθμωτής κατάβασης είναι η εξής:

1. Αρχικοποιεί τα βάρη με τυχαίες τιμές (συνήθως τιμές κοντά στο 1).

2. Δέχεται κάποια είσοδο (ή κάποιο πλήθος εισόδων) και παράγει την αντίστοιχη έξοδο έστω y. Η έξοδος αυτή εξαρτάται από τις τιμές που έχουν τα βάρη w.

3. Δεδομένης μιας συνάρτησης κόστους υπολογίζει το μέτρο της λάθους εκτίμησης του μοντέλου, έστω E. Στη συνέχεια, υπολογίζει τον όρο *gradient* ως εξής:

$$gradient = \frac{dE}{dW} \tag{4.13}$$

Ο όρος *gradient* είναι θετικός όταν το E αυξάνεται καθώς αυξάνονται και οι τιμές των βαρών w, ενώ είναι αρνητικός όταν το E μειώνεται καθώς αυξάνονται οι τιμές των βαρών w. Έτσι, σε κάθε περίπτωση η αρνητική κλίση του E δείχνει

την κατεύθυνση κατά την οποία πρέπει να μετακινηθούν τα βάρη έτσι ώστε να ελαχιστοποιηθεί το σφάλμα.

4. Τα βάρη ενημερώνονται βάσει της εξίσωσης:

$$w = w - \eta \cdot |\frac{dE}{dW}| \tag{4.14}$$

Η παράμετρος η ονομάζεται ρυθμός μάθησης (learning rate) και καθορίζει το μέγεθος της μετατόπισης των βαρών. Ουσιαστικά, ο ρυθμός μάθησης καθορίζει το πόσο γρήγορα προσαρμόζεται το μοντέλο. Μεγάλες τιμές του ρυθμού μάθησης οδηγούν σε γρήγορη σύγκλιση του αλγορίθμου. Ωστόσο, συμβαίνουν μεγάλες αλλαγές στις τιμές των βαρών και είναι πιθανό να μην βρεθεί η βέλτιστη λύση. Μικρές τιμές του ρυθμού μάθησης εγγυώνται καλύτερα αποτελέσματα με επιβάρυνση, ωστόσο, στην ταχύτητα σύγκλισης του αλγορίθμου.

5. Τα βήματα 2, 3, 4 επαναλαμβάνονται μέχρι να τελειώσουν τα δεδομένα εισόδου ή να επιτευχθεί κάποιο επιθυμητό επίπεδο ανοχής σφαλμάτων.

Κατά τη φάση εκπαίδευσης, λοιπόν, η μετατόπιση των βαρών υπολογίζεται από τον αλγόριθμο Gradient Descent. Για την ενημέρωση τους, ωστόσο, υπεύθυνος είναι ένας αλγόριθμος που ονομάζεται οπισθοδρομική διάδοση (Back Propagation) [33]. Ο αλγόριθμος αυτός βασίζεται στον αλγόριθμο Gradient Descent και χρησιμοποιείται για τον υπολογισμό των μερικών παραγώγων που σχετίζονται με το σφάλμα και το εκάστοτε βάρος. Ο αλγόριθμος αυτός φροντίζει να παράγει ένα σήμα λάθους, το οποίο προκύπτει από την σύγκριση της προβλεπόμενης και της αναμενόμενης εξόδου, και στη συνέχεια το διαδίδει σε όλο το δίκτυο ακολουθώντας αντίστροφη πορεία. Το σήμα, δηλαδή, μεταβιβάζεται από το επίπεδο εξόδου προς το επίπεδο εισόδου και με βάση αυτό προσαρμόζονται οι τιμές των βαρών στις συνάψεις ολόκληρου του νευρωνικού δικτύου.

Τέλος, αφού έγινε μια σύντομη περιγραφή του διαδικασίας εκπαίδευσης των νευρωνικών δικτύων, θα γίνει και μια συνοπτική παρουσίαση της παραμετροποίησης αυτών. Οι σημαντικότερες παράμετροι ενός νευρωνικού δικτύου, οι οποίες ορίζονται από τον προγραμματιστή, είναι οι εξής:

- Batch: Ο αριθμός των δειγμάτων που πρέπει να τροφοδοτηθεί στο δίκτυο πριν πραγματοποιηθεί ενημέρωση των βαρών [36].
- Epoch: Ο αριθμός των επαναλήψεων εκπαίδευσης πάνω σε ολόκληρο το training set [36].
- Activation Function: Επιλογή της συνάρτησης ενεργοποίησης που θα χρησιμοποιηθεί.
- Cost Function: Επιλογή της συνάρτησης κόστους που θα χρησιμοποιηθεί.
- Learning Rate: Επιλογή του ρυθμού μάθησης.
- Optimizers: Πρόκειται για αλγόριθμους βελτιστοποίησης που βασίζονται στον Gradient Descent [37]. Αυτοί που χρησιμοποιούνται κυρίως είναι οι Adam, Adagrad, RMSProp και SGD.

- Dropout: Είναι μια παράμετρος που αναφέρεται σε κάποιο επίπεδο του νευρωνικού δικτύου. Καθορίζει το ποσοστό των δειγμάτων που θα απορριφθούν και δεν θα περάσουν στο επόμενο επίπεδο. Χρησιμοποιείται για λόγους αντιμετώπισης του φαινομένου της υπερπροσαρμογής.

4.1.3 Μοντέλα Μηχανικής Μάθησης

Στο πλαίσιο αυτής της διπλωματικής εργασίας το ενδιαφέρον εστιάζεται σε αλγορίθμους επιβλεπομένης μηχανικής μάθησης που χρησιμοποιούνται για binary classification προβλήματα των οποίων τα δεδομένα εισόδου έχουν μορφή χρονικής ακολουθίας. Στη συνέχεια, ακολουθεί μια συνοπτική παρουσίαση των μοντέλων που θα χρησιμοποιηθούν.

4.1.3.1 k-Nearest Neighbors (kNN)

Ο k-Nearest Neighbors [38] είναι ένας αλγόριθμος επιβλεπομένης μηχανικής μάθησης ο οποίος χρησιμοποιείται για προβλήματα ταξινόμησης. Πρόκειται για έναν αρκετά απλό μη παραμετρικό αλγόριθμο ο οποίος βασίζεται στην απόσταση των σημείων προκειμένου να τα ταξινομήσει. Συνήθως, ως κριτήριο για την μέτρηση επιλέγεται η ευκλείδεια απόσταση.

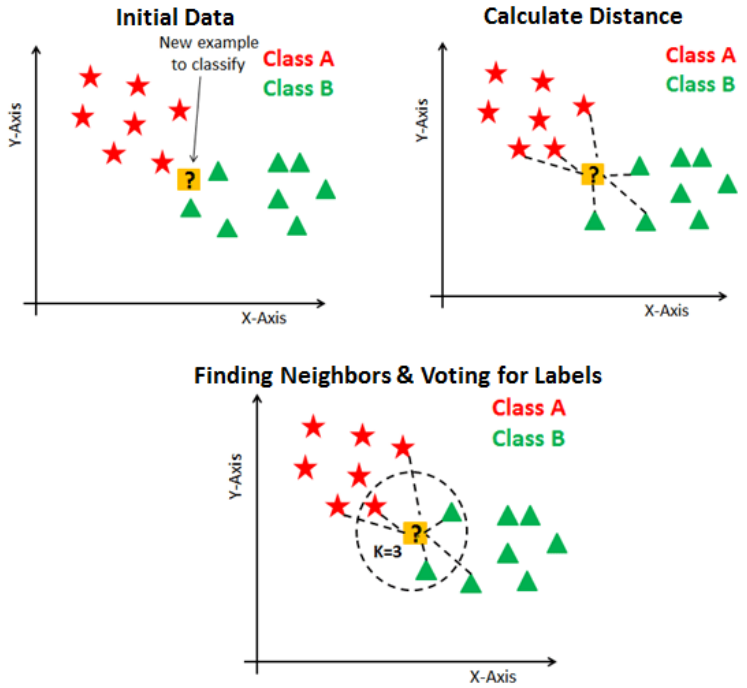
Ο kNN ταξινομητής ανήκει σε μια κατηγορία αλγορίθμων η οποία ονομάζεται lazy learning. Αυτό προκύπτει από το γεγονός ότι το μοντέλο δεν εκπαιδεύεται, αλλά κάθε φορά που κάνει κάποια πρόβλεψη χρησιμοποιεί το σύνολο δεδομένων εκπαίδευσης προκειμένου να ταξινομήσει το εν λόγω δείγμα. Στη συνέχεια, ακολουθεί η μαθηματική περιγραφή του αλγορίθμου.

Υποθέτοντας σύνολο δεδομένων εκπαίδευσης S και δείγμα x που δίνεται ως είσοδος στον αλγόριθμο, υπολογίζεται η απόσταση του x από κάθε δείγμα του συνόλου εκπαίδευσης. Πιο συγκεκριμένα υπολογίζεται η τιμή:

$$d(x, x') = \sqrt{(x_1 - x'_1)^2 + ... + (x_n - x'_n)^2} \tag{4.15}$$

όπου n το πλήθος των χαρακτηριστικών που έχουν τα δείγματα.

Αμέσως μετά, επιλέγονται τα πρώτα k σημεία των οποίων η απόσταση από τα δείγμα εισόδου είναι και η μικρότερη. Το k είναι μια παράμετρος η οποία ορίζεται από τον προγραμματιστή και αναφέρεται στον αριθμό των γειτονικών δειγμάτων (neighbors) που θα ληφθούν υπόψιν. Τελικά, η ταξινόμηση του δείγματος x πραγματοποιείται βάσει των κλάσεων που ανήκουν οι k γείτονές του. Πιο συγκεκριμένα, το δείγμα ταξινομείται στην κλάση της οποία μέλη είναι η πλειοψηφία των γειτόνων του.



Πηγή: https://res.cloudinary.com/dyd911kmh/image/upload/f_auto,q_auto:best/v1531424125/KNN_final1_ibdm8a.png

Σχήμα 4.6: Εφαρμογή του kNN

Στο παράδειγμα που παρουσιάζεται στο σχήμα 4.6 το δείγμα προς εξέταση ταξινομείται τελικά με την ετικέτα της κατηγορίας B, καθώς η παράμετρος k έχει οριστεί με την τιμή 3 και δύο από τους τρεις γείτονες του δείγματος ανήκουν στην κλάση B.

Η λειτουργία του αλγορίθμου στηρίζεται στην ιδέα ότι δεδομένα με παρόμοια χαρακτηριστικά θα ανήκουν στην ίδια κατηγορία, καθώς οι αναπαραστάσεις στους στον χώρο θα έχουν μικρές αποστάσεις. Ωστόσο, μια τέτοια απλούστευση δεν είναι πάντα αληθής για αυτό και ο kNN δεν προτιμάται. Επιπλέον, ένα από τα βασικότερα ελαττώματα του k-Nearest Neighbors αλγορίθμου είναι το γεγονός ότι σε κάθε πρόβλεψη που κάνει πρέπει να διατρέξει ολόκληρο το σύνολο των δεδομένων εκπαίδευσης. Αυτό μπορεί να παρουσιάσει προβλήματα τόσο σε σχέση με την ταχύτητα της πρόβλεψης, καθώς πρέπει να υπολογιστούν οι αποστάσεις με όλα τα δείγματα του συνόλου εκπαίδευσης, όσο και με την μνήμη του συστήματος, καθώς τα δεδομένα του συνόλου εκπαίδευσης πρέπει μονίμως να είναι αποθηκευμένα στο σύστημα.

4.1.3.2 Μηχανές Διανυσμάτων Υποστύριξης (Support Vector Machines - SVM)

Ο Support Vector Machines [39] είναι ένας αλγόριθμος επιβλεπομένης μηχανικής μάθησης ο οποίος χρησιμοποιείται για προβλήματα ταξινόμησης. Ο SVM χρησιμοποιείται κυρίως για προβλήματα δυαδικής ταξινόμησης, χωρίς όμως αυτό να είναι δεσμευτικό καθώς μπορεί να επεκταθεί και για multiclass classification προβλήματα [40]. Στη συνέχεια, γίνεται μια σύντομη περιγραφή της λειτουργίας του αλγορίθμου για προβλήματα δυαδικής ταξινόμησης.

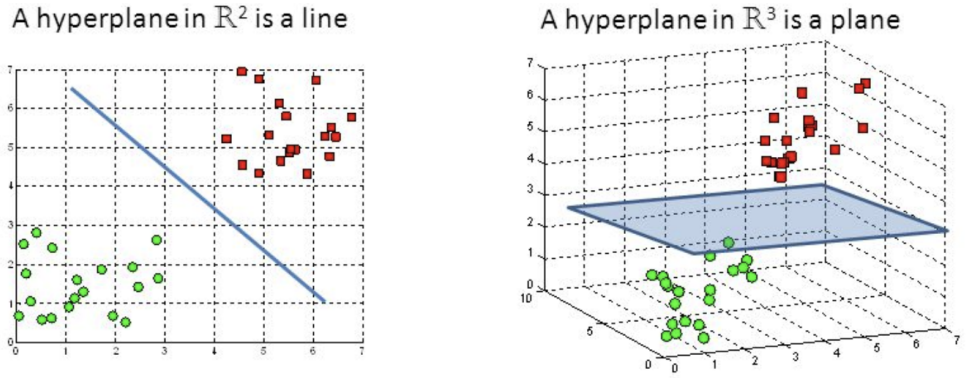
Ο SVM απεικονίζει τα δεδομένα εισόδου ως σημεία σε χώρο n διαστάσεων (όπου n είναι το πλήθος των χαρακτηριστικών που έχουν τα δείγματα) και επιχειρεί να βρει

το υπερεπίπεδο (hyperplane) το οποίο διαχωρίζει τις κλάσεις αυτών με τον καλύτερο τρόπο.

Ένα υπερεπίπεδο είναι το n-διάστατο όριο σύμφωνα με το οποίο ο αλγόριθμος αποφασίζει σε ποια κλάση ανήκει το δείγμα εισόδου και παριστάνεται από την παρακάτω εξίσωση:

$$w^T \cdot x + b = 0 \tag{4.16}$$

όπου w είναι το διάνυσμα των βαρών του υπερεπιπέδου και b μια σταθερά η οποία καλείται bias. Για παράδειγμα, σε δισδιάστατο χώρο το υπερεπίπεδο είναι μια ευθεία, ενώ σε τρισδιάστατο είναι ένα επίπεδο.

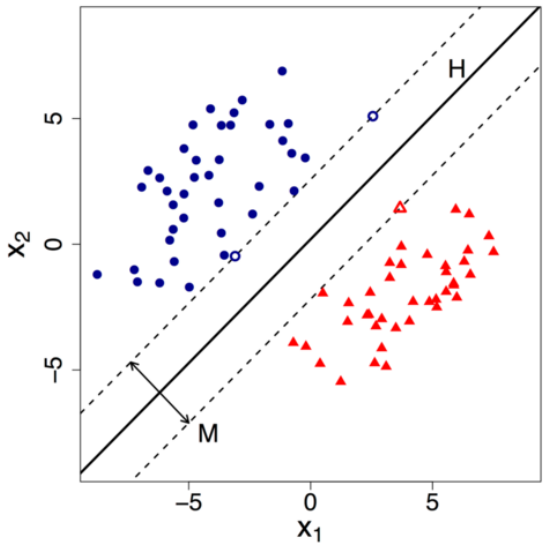


Πηγή: https://miro.medium.com/max/2800/0*6VMrGr6j0-Dlns6-.png

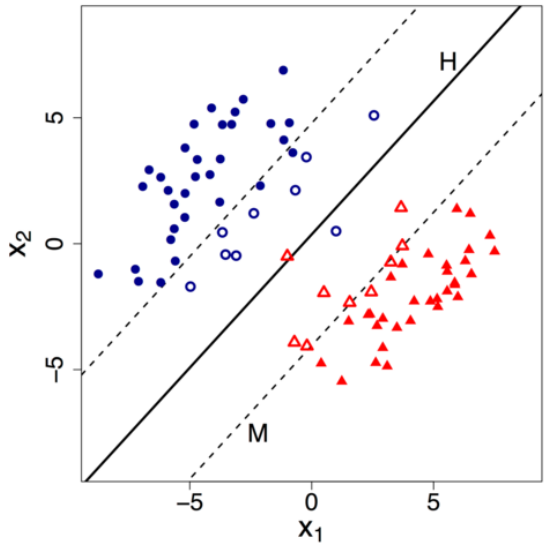
Σχήμα 4.7: Υπερεπίπεδο σε χώρο 2 και 3 διαστάσεων

Η θέση ενός υπερεπιπέδου καθορίζεται από τα σημεία εισόδου που βρίσκονται πλησιέστερα σε αυτό. Τα σημεία αυτά ονομάζονται διανύσματα υποστήριξης (support vector points) και αποτελούν τα σημεία των οποίων η ταξινόμηση είναι και η πιο δύσκολη. Η απόσταση μεταξύ των διανυσμάτων υποστήριξης των δύο διαφορετικών κλάσεων ονομάζεται περιθώριο (margin) και υπολογίζεται βάση της ευκλείδειας νόρμας.

Το πλήθος των υπερεπιπέδων που διαχωρίζει με επιτυχημένο τρόπο τις κλάσεις των δεδομένων εισόδου είναι άπειρο. Ο SVM προσπαθεί να προσδιορίσει το βέλτιστο υπερεπίπεδο (optimal hyperplane), το υπερεπίπεδο δηλαδή που μεγιστοποιεί το περιθώριο μεταξύ των δύο κλάσεων. Στη συνέχεια, ακολουθεί η μαθηματική περιγραφή του αλγορίθμου μέσω ενός παραδείγματος. Στο σχήμα 4.8.α παρουσιάζεται ένα πρόβλημα δυαδικής ταξινόμησης. Έστω, ότι οι δύο κλάσεις (μπλε κύκλοι, κόκκινα τρίγωνα) έχουν τις ετικέτες $y = 1$ και $y = -1$, αντίστοιχα. Με τη μαύρη συνεχόμενη γραμμή απεικονίζεται το υπερεπίπεδο, ενώ με τις διακεκομμένες τα διανύσματα υποστήριξης.



(α′) Hard Margin



Πηγή: https://editor.analyticsvidhya.com/uploads/24817choose_the_hyperplane.png

(β′) Soft Margin

Σχήμα 4.8: Hard Margin και Soft Margin SVM

Το υπερεπίπεδο, το διάνυσμα υποστήριξης που ορίζεται από τα ακραία σημεία της κλάσης των μπλε κύκλων και το διάνυσμα υποστήριξης που ορίζεται από τα ακραία σημεία της κλάσης των κόκκινων τρίγωνων περιγράφονται από τις εξής εξισώσεις, αντίστοιχα:

$$w^T \cdot x + b = 0, w^T \cdot x + b = 1, w^T \cdot x + b = -1 \tag{4.17}$$

Για τα σημεία εισόδου με ετικέτα 1 ισχύει η εξίσωση:

$$w^T \cdot x + b \geq 1 \tag{4.18}$$

ενώ, για τα σημεία εισόδου με ετικέτα -1 ισχύει η εξίσωση:

$$w^T \cdot x + b \leq 1 \tag{4.19}$$

Η απόσταση ενός σημείου x_0 από το υπερεπίπεδο δίνεται από την εξίσωση:

$$dist(x_0) = \frac{|w^T \cdot x_0 + b|}{\|w\|} \tag{4.20}$$

Έπειτα από κανονικοποίηση του διανύσματος w, έτσι ώστε το μέτρο της απόστασης των σημείων που ορίζουν τα διανύσματα υποστήριξης από το υπερεπίπεδο να ισούται είτε με 1 είτε με -1 (αναλόγως την κλάση στην οποία ανήκει το εκάστοτε σημείο), προκύπτει ότι το margin δίνεται από την εξίσωση:

$$\frac{2}{\|w\|} = \frac{2}{\sqrt{w^T \cdot w}} \tag{4.21}$$

Συνεπώς, ο αλγόριθμος SVM προσπαθεί να βρει τις τιμές των βαρών που μεγιστοποιούν την παραπάνω συνάρτηση, φροντίζοντας παράλληλα να γίνεται σωστή ταξινόμησή των δεδομένων εκπαίδευσης. Πρόκειται για ένα πρόβλημα βελτιστοποίησης το οποίο εκφράζεται από τις εξής σχέσεις:

$$max(\frac{2}{\sqrt{w^T \cdot w}}), y_i(w^T \cdot x_i + b) \geq 1, i = 1, 2, 3, ... \tag{4.22}$$

Ωστόσο, ο SVM προσπαθεί να υπολογίσει ένα ισοδύναμο πρόβλημα του οποίου η επίλυση είναι ευκολότερη. Αυτό είναι το εξής:

$$min(\frac{\sqrt{w^T \cdot w}}{2}), y_i(w^T \cdot x_i + b) \geq 1, i = 1, 2, 3, ... \tag{4.23}$$

Πρόκειται για ένα τετραγωνικό πρόβλημα γραμμικού προγραμματισμού του οποίου η λύση προκύπτει από εφαρμογή των πολλαπλασιαστών Lagrange.

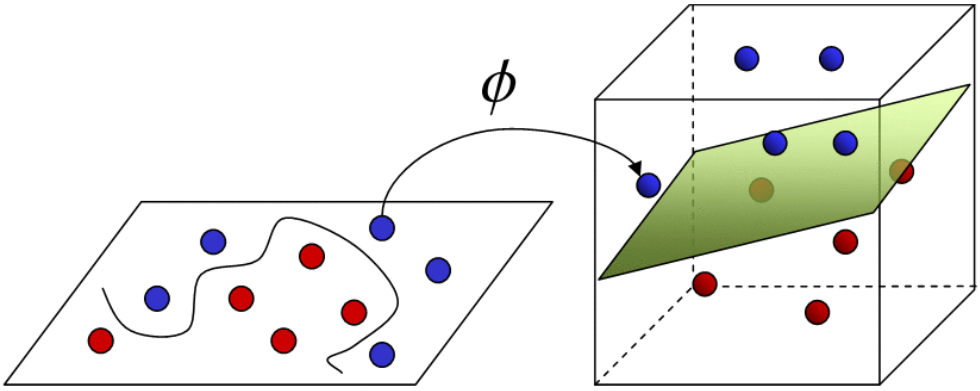
Θα πρέπει να σημειωθεί πως ότι έχει περιγραφεί μέχρι τώρα ανήκει σε μια κατηγορία αλγορίθμων SVM, η οποία ονομάζεται Hard Margin. Στην περίπτωση του Soft Margin [41], εισάγεται ένας βαθμός χαλάρωσης (παράμετρος C) στη διαδικασία εύρεσης του βέλτιστου υπερεπιπέδου, γεγονός που επιτρέπεται να υπάρχουν παραβιάσεις (outliers) στη συνθήκη ταξινόμησης. Αυτό επιτυγχάνεται με την εισαγωγή των μεταβλητών slack (ξ_i), οι οποίες χρησιμοποιούνται για να εκφράσουν το μέτρο της λάθους ταξινόμησης. Εάν το δείγμα i έχει ταξινομηθεί σωστά τότε θα είναι $\xi_i = 0$, ενώ αν έχει ταξινομηθεί εσφαλμένα θα είναι $\xi_i > 0$. Συνεπώς, το πρόβλημα το οποίο καλείται να επιλύσει ο SVM έχει την εξής μορφή:

$$min(\frac{\sqrt{w^T \cdot w}}{2} + C \cdot \sum_{i=1}^{n} \xi_i), y_i(w^T \cdot x_i + b) \geq 1 - \xi_i, i = 1, 2, 3, ...n \tag{4.24}$$

Η παράμετρος C είναι μια παράμετρος κανονικοποίησης μέσω της οποίας ρυθμίζεται το trade-off που υπάρχει μεταξύ της μεγιστοποίησης του margin και της ελαχιστοποίησης του σφάλματος κατά τη διαδικασία εκπαίδευσης. Μικρές τιμές της παραμέτρου υπονοούν "softer margin", ενώ μεγάλες "harder margin".

Μέχρι στιγμής έχει γίνει η υπόθεση ότι τα δεδομένα εισόδου είναι γραμμικά διαχωρίσιμα. Ωστόσο, θα πρέπει να σημειωθεί ότι, στις περισσότερες περιπτώσεις, τα

δεδομένα που αφορούν εφαρμογές του πραγματικού κόσμου δεν είναι γραμμικά διαχωρίσιμα. Η σημαντικότητα του SVM αλγορίθμου έγκειται στο γεγονός ότι μπορεί να επιλύσει και προβλήματα τα οποία δεν είναι γραμμικά διαχωρίσιμα. Αυτό επιτυγχάνεται μέσω μιας τεχνικής η οποία ονομάζεται μέθοδος πυρήνα (kernel method). Σύμφωνα με αυτή, χρησιμοποιούνται συναρτήσεις, οι οποίες καλούνται συναρτήσεις πυρήνα (kernel functions), που παίρνουν τα δεδομένα εισόδου και τα απεικονίζουν σε χώρο μεγαλύτερων διαστάσεων. Πρόκειται για μια επαναληπτική διαδικασία στης οποίας κάθε βήμα οι διαστάσεις του χώρου που απεικονίζεται το πρόβλημα αυξάνονται κατά ένα. Η διαδικασία τερματίζει την πρώτη φορά που μπορεί να βρεθεί υπερεπίπεδο το οποίο διαχωρίζει τα δεδομένα εισόδου.



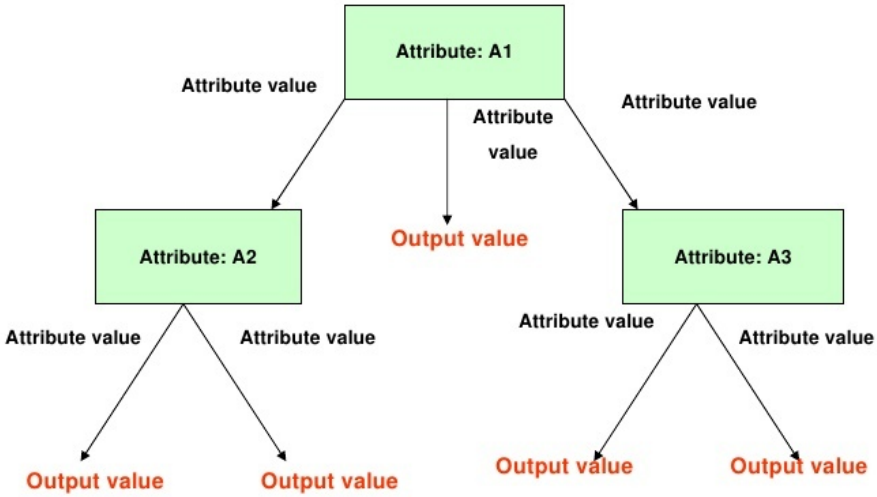
Πηγή: https://miro.medium.com/max/872/1*zWzeMGyCc7KvGD9X8lwlnQ.png

Σχήμα 4.9: Kernelling για μη διαχωρίσιμα δεδομένα εισόδου

4.1.3.3 Τυχαία Δάση (Random Forest)

Ο Random Forest [42] είναι ένας αλγόριθμος επιβλεπομένης μηχανικής μάθησης ο οποίος χρησιμοποιείται για προβλήματα ταξινόμησης και παλινδρόμησης. Δομικό στοιχείο αυτού του αλγορίθμου αποτελεί το δένδρο απόφασης (Decision Tree). Πρόκειται για έναν ακόμη αλγόριθμο επιβλεπομένης μηχανικής μάθησης πάνω στον οποίο στηρίζεται ο Random Forest.

Ένα δένδρο απόφασης είναι ένας γράφος δενδρικής δομής του οποίου τα φύλλα (τερματικοί κόμβοι) φέρουν τις ετικέτες των κλάσεων του προβλήματος ταξινόμησης. Κάθε εσωτερικός κόμβος (κόμβος απόφασης) αναφέρεται σε κάποιο χαρακτηριστικό του συνόλου δεδομένων εκπαίδευσης, ενώ κάθε ακμή αναφέρεται σε κάποια τιμή που μπορεί να πάρει το χαρακτηριστικό του κόμβου από τον οποίο ξεκινάει. Η δομή ενός δένδρου απόφασης φαίνεται καλύτερα στο σχήμα 4.10.



Πηγή: https://image.slidesharecdn.com/machine-learning-lecture-3-1205567881737248-5/3-8-728.jpg

Σχήμα 4.10: Δομή δένδρου απόφασης

Ένα δένδρο απόφασης σχηματίζεται με βάση τα δεδομένα που τροφοδοτούνται σε αυτό κατά τη φάση εκπαίδευσης. Πρόκειται για μια διαδικασία η οποία περιλαμβάνει το διαχωρισμό (splitting) ενός κόμβου σε επιμέρους κόμβους με βάση κάποιο κριτήριο. Αρχικά όλα τα δείγματα του συνόλου δεδομένων εκπαίδευσης τοποθετούνται στη ρίζα του δένδρου. Στη συνέχεια, χρησιμοποιώντας κάποια κριτήρια επιλογής, διαλέγεται το χαρακτηριστικό με βάση το οποίο θα γίνει ο διαχωρισμός. Οι νέοι κόμβοι που δημιουργούνται περιέχουν τα δείγματα των οποίων η τιμή για το χαρακτηριστικό με βάση το οποίο έγινε ο διαχωρισμός είναι ίση με αυτή που υπαγορεύει η ακμή που τους ενώνει με τη ρίζα. Η διαδικασία αυτή επαναλαμβάνεται για όλους τους κόμβους μέχρις ότου να σχηματιστεί το δένδρο.

Τα κριτήρια που χρησιμοποιούνται για την επιλογή του χαρακτηριστικού με βάση το οποίο θα γίνει ο διαχωρισμός πηγάζουν από την επιστήμη της θεωρίας πληροφορίας. Οι μετρικές που χρησιμοποιούνται κυρίως είναι οι εξής [43]:

- Εντροπία (Entropy): Η εντροπία εκφράζει το μέτρο της τυχαιότητας της πληροφορίας προς επεξεργασία. Όσο μεγαλύτερη είναι η τιμή της τόσο δυσκολότερη είναι η εξαγωγή συμπερασμάτων. Συνεπώς, για το διαχωρισμό των κόμβων συνίσταται η επιλογή χαρακτηριστικών που οδηγούν σε χαμηλότερη εντροπία. Υποθέτοντας σύνολο δεδομένων εκπαίδευσης S για c διαφορετικές κατηγορίες, η εντροπία ορίζεται από τη σχέση

$$E(S) = -\sum_{i=1}^{c} p_i \cdot \log_2 p_i \tag{4.25}$$

  όπου p_i το ποσοστό των παραδειγμάτων του S που ανήκουν στην κατηγορία i.

- Κέρδος πληροφορίας (Information Gain): Το κέρδος πληροφορίας αναπαριστά τη μείωση της εντροπίας του συνόλου εκπαίδευσης S αν επιλεγεί ως παράμετρος διαχωρισμού κάποιο συγκεκριμένο χαρακτηριστικό A. Όσο μειώνεται η εντροπία

τόσο αυξάνεται το κέρδος της πληροφορίας. Συνεπώς, για τον διαχωρισμό των κόμβων συνίσταται η επιλογή χαρακτηριστικών που οδηγούν σε μεγαλύτερο κέρδος πληροφορίας. Το κέρδος πληροφορίας ορίζεται από τη σχέση

$$IG(S, A) = E(S|_{before\ splitting}) - \sum_{i=1}^{K} E(i) \tag{4.26}$$

όπου K τα υποσύνολα που προέκυψαν από το splitting με βάση το χαρακτηριστικό A. Δύο από τους πιο γνωστούς αλγορίθμους που χρησιμοποιούν το κέρδος πληροφορίας για να διαχωρίσουν τους κόμβους είναι ο ID3 και ο C4.5.

- Συντελεστής Gini (Gini Index): Ο συντελεστής Gini εκφράζει την ομοιογένεια ενός συνόλου εκπαίδευσης S και μπορεί να λάβει τιμές ανάμεσα στο διάστημα [0, 1]. Όσο η τιμή του πλησιάζει στο 0 το σύνολο περιέχει κυρίως παραδείγματα που ανήκουν στην ίδια κατηγορία, ενώ όσο η τιμή του πλησιάζει στο 1 το σύνολο περιέχει τυχαία κατανεμημένα παραδείγματα. Συνεπώς, για το διαχωρισμό των κόμβων συνίσταται η επιλογή χαρακτηριστικών που οδηγούν σε μικρότερη τιμή του συντελεστή Gini. Θα πρέπει να σημειωθεί πως η χρήση αυτού το κριτηρίου επιτρέπει το σχηματισμό μόνο δύο υποσυνόλων, σε αντίθεση με το κέρδος πληροφορίας που επιτρέπει διαχωρισμό σε περισσότερα υποσύνολα. Ο συντελεστής Gini ενός συνόλου S για c διαφορετικές κατηγορίες ορίζεται από τη σχέση

$$Gini(S) = 1 - \sum_{i=1}^{c} p_i^2 \tag{4.27}$$

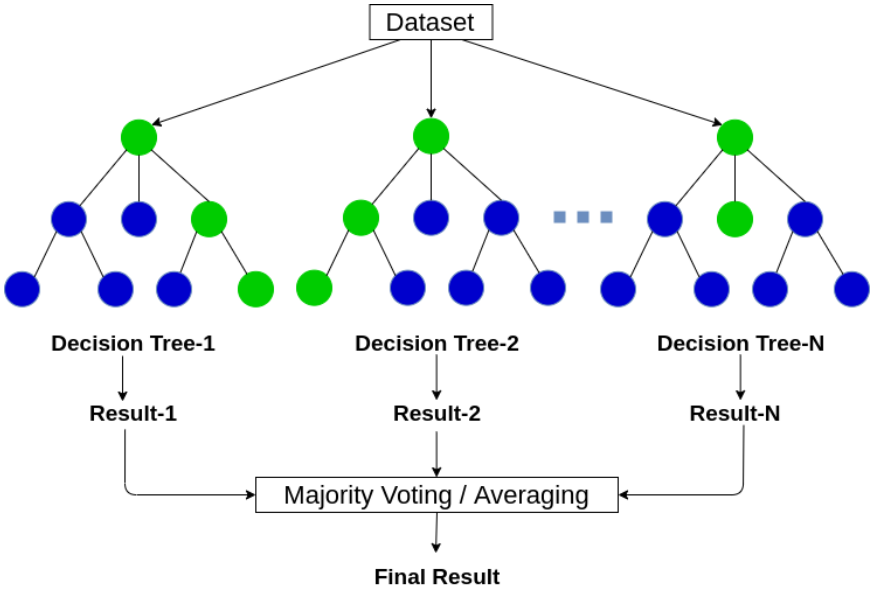
όπου p_i το ποσοστό των παραδειγμάτων του S που ανήκουν στην κατηγορία i. Ο συντελεστής Gini ενός split με βάση κάποιο χαρακτηριστικό A υπολογίζεται λαμβάνοντας το σταθμισμένο άθροισμα των συντελεστών Gini των δύο υποσυνόλων που προκύπτουν από το διαχωρισμό. Ένας από τους πιο γνωστούς αλγορίθμους που χρησιμοποιεί το συντελεστή Gini για να διαχωρίσει τους κόμβους είναι ο CART.

Ένα από τα βασικότερα προβλήματα που μπορεί να προκύψουν κατά τη δημιουργία ενός δένδρου απόφασης είναι το φαινόμενο της υπερπροσαρμογής. Για τον λόγο αυτό, είναι δυνατή η ρύθμιση παραμέτρων που περιορίζουν την ανάπτυξη του δένδρου κατά το σχηματισμό του (pruning). Οι πιο σημαντικές από αυτές σχετίζονται με:

- το μέγιστο βάθος του δένδρου απόφασης
- το ελάχιστο πλήθος δειγμάτων που απαιτείται για το διαχωρισμό ενός κόμβου
- το ελάχιστο πλήθος δειγμάτων που απαιτείται προκειμένου ένας κόμβος να μπορεί να θεωρηθεί φύλλο
- το μέγιστο πλήθος φύλλων του δένδρου απόφασης

Ο αλγόριθμος Decision Tree, αφού έχει ολοκληρώσει τη δημιουργία του δένδρου απόφασης μέσω της διαδικασίας που αναφέρθηκε παραπάνω, ταξινομεί κάποιο νέο δείγμα εισόδου ακολουθώντας το μονοπάτι του δένδρου που ταιριάζει περισσότερο με τα χαρακτηριστικά του δείγματος και αναθέτει σε αυτό την ετικέτα του φύλλου του μονοπατιού που ακολουθήθηκε.

Τώρα που έχει αναλυθεί σε βάθος η έννοια του δένδρου απόφασης θα γίνει η περιγραφή του αλγορίθμου Random Forest. Ο αλγόριθμος αυτός δημιουργεί ένα "δάσος" από δένδρα απόφασης. Το πλήθος n των δένδρων τα οποία συνιστούν το "δάσος" είναι μια παράμετρος η οποία ορίζεται από τον προγραμματιστή. Ένα από τα βασικότερα πλεονεκτήματα του Random Forest είναι ότι εισάγει την έννοια της τυχαιότητας. Αυτό συμβαίνει διότι κάθε δένδρο του "δάσους" εκπαιδεύεται με διαφορετικό τυχαίο υποσύνολο του συνόλου δεδομένων εκπαίδευσης. Επίσης, κάθε δένδρο του "δάσους" σχηματίζεται μέσω splitting από χαρακτηριστικό που προκύπτει από τυχαίο υποσύνολο των χαρακτηριστικών του συνόλου δεδομένων εκπαίδευσης. Το γεγονός αυτό καθιστά τον Random Forest πιο αποδοτικό σε σχέση με ένα απλό Decision Tree, καθώς εισάγεται μεγαλύτερη ποικιλομορφία στη διαδικασία εύρεσης της κατάλληλης ετικέτας για κάποιο δείγμα εισόδου, ενώ παράλληλα αντιμετωπίζεται σε μεγαλύτερο βαθμό και το φαινόμενο της υπερπροσαρμογής.



Πηγή: https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/rfc_vs_dt1.png

Σχήμα 4.11: Ο αλγόριθμος Random Forest

Όταν ο Random Forest λάβει ένα νέο δείγμα υπολογίζει την ετικέτα που δίνεται σε αυτό για κάθε δένδρο του "δάσους" και τελικά ταξινομεί το δείγμα στη κατηγορία της οποίας την ετικέτα βρήκε η πλειοψηφία των δένδρων.

4.1.3.4 Extreme Gradient Boosting (XGBoost)

Ο XGBoost [44, 45] είναι ένας αλγόριθμος επιβλεπομένης μηχανικής μάθησης ο οποίος χρησιμοποιείται για προβλήματα ταξινόμησης και παλινδρόμησης. Όπως και στην περίπτωση του Random Forest, δομικό στοιχείο αυτού του αλγορίθμου αποτελεί το δένδρο απόφασης (Decision Tree).

Η βασικότερη διαφορά του σε σχέση με τον Random Forest είναι ο τρόπος με τον οποίο εκπαιδεύεται το μοντέλο. Στον Random Forest δημιουργούνται n δέντρα με παράλληλο τρόπο, ενώ στον XGBoost δημιουργούνται n δέντρα με ακολουθιακό τρόπο. Πιο συγκεκριμένα, κάθε νέο δένδρο που σχηματίζεται λαμβάνει υπόψιν το λάθος της εκτίμησης του προηγούμενου και προσπαθεί με βάση αυτό να βελτιώσει την προβλεπόμενη έξοδο. Προκειμένου να γίνει πλήρως κατανοητός ο αλγόριθμος θα

πρέπει να προηγηθεί μια σύντομη επεξήγηση της έννοιας Gradient Boosting Learning [46].

Ο όρος Boosting αναφέρεται σε μια μεθοδολογία σύμφωνα με την οποία πολλαπλά απλά μοντέλα (weak learners) σχηματίζουν ένα πιο σύνθετο και αποδοτικό μοντέλο (strong learner). Χάριν απλότητας, το αρχικό μοντέλο θα συμβολίζεται με $f(\cdot)$ και τα υπόλοιπα απλά μοντέλα με $\Delta(\cdot)$, ενώ με $F(\cdot)$ θα συμβολίζεται η έξοδος του πρώτου και του σύνθετου μοντέλου. Υποθέτοντας διάνυσμα εισόδου x και αναμενόμενη έξοδο y, το πρώτο μοντέλο που δημιουργείται παράγει έξοδο:

$$F_0(x) = f_0(x) \tag{4.28}$$

Στη συνέχεια, το μοντέλο $\Delta 1(\cdot)$ δέχεται ως είσοδο το λάθος του προηγούμενου μοντέλου και παράγει μια πρόβλεψη για αυτό. Η έξοδος του μοντέλου αυτού είναι ίση με:

$$F_1(x) = F_0(x) + \eta \cdot \Delta_1 \tag{4.29}$$

όπου $\Delta_1 = \Delta_1(y - F_0(x))$ και η ο ρυθμός μάθησης. Συνεπώς, το σύνθετο μοντέλο που προκύπτει μετά από τη λειτουργία των n μοντέλων παράγει έξοδο που περιγράφεται από τις σχέσεις:

$$F_0(x) = f_0(x), \hat{y} = F_n(x) = F_{n-1}(x) + \eta \cdot \Delta_n, \Delta_n = \Delta_n(y - F_{n-1}(x)) \tag{4.30}$$

Στην πραγματικότητα δεν χρησιμοποιείται το μέτρο της λάθους εκτίμησης αλλά το πρόσημο αυτής. Συνεπώς, στη περίπτωση του Gradient Boosting χρησιμοποιείται ο όρος gradient προκειμένου να προσδιορισθεί προς ποια κατεύθυνση πρέπει να κινηθεί η προβλεπόμενη έξοδος του μοντέλου ώστε να είναι πιο κοντά στην αναμενόμενη έξοδο. Επομένως:

$$F_0(x) = f_0(x), \hat{y} = F_n(x) = F_{n-1}(x) + \eta \cdot |\frac{\partial C(y, F_{n-1}(x))}{\partial F_{n-1}(x)}| \tag{4.31}$$

όπου C μια συνάρτηση κόστους. Όλα τα απλά μοντέλα $\Delta(\cdot)$ εκπαιδεύονται με είσοδο το σύνολο που προκύπτει από την εφαρμογή κάποιας συνάρτησης κόστους σε κάθε δείγμα του συνόλου εκπαίδευσης. Συνεπώς, το μόνο που χρειάζεται για την εκπαίδευση τους είναι η αναμενόμενη τιμή κάποιου δείγματος και όχι το ίδιο το δείγμα.

Θα πρέπει να σημειωθεί πως οι αλγόριθμοι Gradient Descent και Gradient Boosting είναι δύο εντελώς διαφορετικές προσεγγίσεις που το μόνο κοινό που έχουν είναι ο υπολογισμός όρων gradients. Ο αλγόριθμος Gradient Descent χρησιμοποιείται στα νευρωνικά δίκτυα για την ενημέρωση των βαρών τους, ενώ ο αλγόριθμος Gradient Boosting χρησιμοποιείται για να συνθέσει την έξοδο πολλών απλών μοντέλων σε μια πιο ακριβή και αποτελεσματική έξοδο.

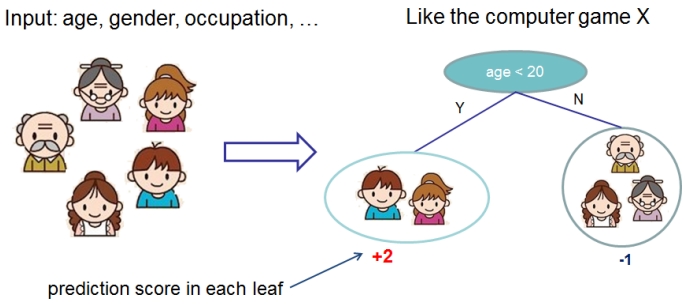
Τώρα που έχει γίνει μια σύντομη περιγραφή της έννοιας του Gradient Boosting Learning θα παρουσιαστεί ο αλγόριθμος XGBoost.

Ο αλγόριθμος Extreme Gradient Boosting στηρίζει την υλοποίηση του σε μεγάλο βαθμό στον απλό Gradient Boosting αλγόριθμο. Ωστόσο, ο XGBoost πραγματοποιεί υπολογισμούς όρων gradient δεύτερης τάξης. Οι μερικές παράγωγοι δεύτερης τάξης προσφέρουν περισσότερη πληροφορία σχετικά με την κατεύθυνση που πρέπει να

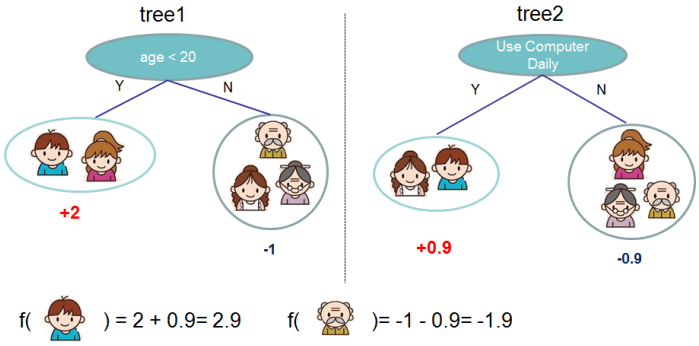
ακολουθηθεί έτσι ώστε να προσεγγιστεί το τοπικό ελάχιστο της συνάρτησης κόστους. Επιπλέον, ο XGBoost προσφέρει τη δυνατότητα κανονικοποίησης μέσω των μεθόδων Lasso και Ridge Regularization.

Ο αλγόριθμος XGBoost χρησιμοποιεί δένδρα ταξινόμησης και παλινδρόμησης (classification and regression tree - CART) για τα απλά μοντέλα. Όπως έχει ήδη αναφερθεί και στο υποκεφάλαιο 4.1.3.3, στα δέντρα αυτά μπορεί να πραγματοποιηθεί μόνο δυαδικός διαχωρισμός. Ένα πολύ σημαντικό χαρακτηριστικό του αλγορίθμου είναι το γεγονός ότι αναθέτει στα φύλλα των δένδρων κάποιο σκορ (πραγματικός αριθμός). Αυτό διαφέρει από τα κλασσικά δένδρα απόφασης τα οποία στα φύλλα τους περιέχουν μόνο την ετικέτα της αντίστοιχης κατηγορίας. Μέσω αυτών των τιμών υλοποιείται μια αντιστοίχιση των πιθανών κατηγοριών του προβλήματος ταξινόμησης σε πραγματικές τιμές, γεγονός που επιτρέπει την εκτέλεση του gradient boosting αλγορίθμου.

Στα σχήματα 4.12.α και 4.12.β παρουσιάζεται ένα παράδειγμα το οποίο έχει παρθεί από το επίσημο documentation του XGBoost [44]. Το παράδειγμα αφορά την ταξινόμηση των μελών μιας οικογένειας με βάση το αν τους αρέσουν τα παιχνίδια στον υπολογιστή ή όχι.



(α′) Δένδρο ταξινόμησης και παλινδρόμησης



Πηγή: https://raw.githubusercontent.com/dmlc/web-data/master/xgboost/model/cart.png

(β′) Ακολουθιακό ενισχυτικό μοντέλο δένδρων ταξινόμησης και παλινδρόμησης

Σχήμα 4.12: Παράδειγμα εφαρμογής του XGBoost

Με βάση τα όσα έχουν ειπωθεί μέχρι τώρα, ακολουθεί η μαθηματική περιγραφή του αλγορίθμου XGBoost. Έστω σύνθετο μοντέλο που αποτελείται από m weak learners. Χάριν απλότητας, με $f(\cdot)$ θα συμβολίζονται τα απλά CART μοντέλα που χρησιμοποιούνται. Η προβλεπόμενη έξοδος του σύνθετου μοντέλου εκφράζεται από τις εξής σχέσεις:

$$\hat{y}_i^{(0)} = f_0(x_i), \hat{y}_i^{(m)} = \hat{y}_i^{(m-1)} + f_m(x_i) \tag{4.32}$$

Σε έναν απλό Gradient Boosting αλγόριθμο ο όρος $f_m(x_i)$ σχετίζεται με την ελαχιστοποίηση του σφάλματος προκειμένου να υπάρξει μια καλύτερη προσέγγιση της τιμής εξόδου. Στην περίπτωση του XGBoost, ο όρος αυτός σχετίζεται με την ελαχιστοποίηση του σφάλματος προκειμένου να προσδιοριστεί το επόμενο βελτιωμένο απλό μοντέλο. Υποθέτοντας ότι εισάγονται στο μοντέλο n δεδομένα εισόδου το συνολικό σφάλμα για το βήμα t, όπου $0 < t < m$, θα είναι ίσο με:

$$Loss^{(t)} = \sum_{i=1}^{n} C(y_i, \hat{y}_i^{(t)}) + \sum_{i=1}^{t} \Omega(f_i) = \sum_{i=1}^{n} C(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{i=1}^{t} \Omega(f_i) \quad (4.33)$$

όπου $C(\cdot)$ κάποια συνάρτηση κόστους και $\Omega(\cdot)$ ο όρος κανονικοποίησης. Παίρνοντας το ανάπτυγμα Taylor δεύτερης τάξης της παραπάνω σειράς προκύπτει:

$$Loss^{(t)} = \sum_{i=1}^{n} g_i \cdot f_t(x_i) + \frac{1}{2} \cdot h_i \cdot f_t^2(x_i) + \sum_{i=1}^{t} \Omega(f_i) \quad (4.34)$$

όπου $g_i = \frac{\partial C(y_i, \partial \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ η μερική παράγωγος πρώτης τάξης του σφάλματος του προηγούμενου βήματος και $h_i = \frac{\partial^2 C(y_i, \partial \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ η μερική παράγωγος δεύτερης τάξης του σφάλματος του προηγούμενου βήματος. Η ελαχιστοποίηση του παραπάνω σφάλματος είναι ο στόχος του t-οστού απλού μοντέλου. Όπως έχει ήδη αναφερθεί, οι έξοδοι των απλών μοντέλων ταυτίζονται με κάποιους πραγματικούς αριθμούς. Υποθέτοντας T το πλήθος των φύλλων του t-οστού δένδρου, I_j το σύνολο των παραδειγμάτων του φύλλου j και w_j το σκορ που αντιστοιχεί στο φύλλο j, ο όρος κανονικοποίησης γράφεται ως εξής:

$$\Omega(f) = \gamma \cdot T + \frac{1}{2} \cdot \lambda \cdot \sum_{j=1}^{T} w_j^2 \quad (4.35)$$

όπου γ και λ παράμετροι οριζόμενοι από τον προγραμματιστή. Από τη παραπάνω εξίσωση φαίνεται ότι χρησιμοποιείται η μέθοδος Ridge για κανονικοποίηση. Επομένως, το συνολικό σφάλμα γράφεται ως εξής:

$$Loss^{(t)} = \sum_{i=1}^{n} g_i \cdot w_{q(x_i)} + \frac{1}{2} \cdot h_i \cdot w_{q(x_i)}^2 + \gamma \cdot T + \frac{1}{2} \cdot \lambda \cdot \sum_{j=1}^{T} w_j^2 \quad (4.36)$$

όπου $q(\cdot)$ μια μέθοδος που αντιστοιχίζει κάποια είσοδο σε κάποιο σκορ. Συνεχίζοντας:

$$Loss^{(t)} = \sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) \cdot w_j + \frac{1}{2} \cdot (\sum_{i \in I_j} h_i + \lambda) \cdot w_j^2] + \gamma \cdot T \quad (4.37)$$

$$Loss^{(t)} = \sum_{j=1}^{T} [G_j \cdot w_j + \frac{1}{2} \cdot H_j \cdot w_j^2] + \gamma \cdot T \quad (4.38)$$

όπου $G_j = \sum_{i \in I_j} g_i$ και $H_j = \sum_{i \in I_j} (h_i + \lambda)$. Λύνοντας το πρόβλημα $\frac{\partial Loss^{(t)}}{\partial w_j} = 0$ προκύπτει ότι η βέλτιστη τιμή των σκορ είναι ίση με:

$$Loss^{(t)} = -\frac{1}{2} \sum_{j=1}^{T} (\frac{G_j^2}{H_j + \lambda}) + \gamma \cdot T \quad (4.39)$$

Η παραπάνω συνάρτηση αποτελεί, ουσιαστικά, ένα τρόπο μέτρησης της "ποιότητας" της δομής του δένδρου. Όσο μικρότερο είναι το σφάλμα τόσο καλύτερη είναι η δομή. Το επόμενο δένδρο της ακολουθίας προκύπτει από το προηγούμενο πραγματοποιώντας διαχωρισμό κάποιου φύλλου του σε δύο, έχοντας προσδιορίσει κιόλας μάλιστα ποιες είναι οι βέλτιστες τιμές των σκορ των φύλλων του.

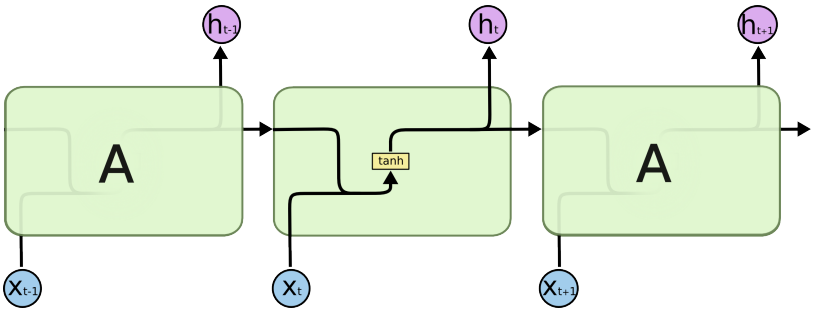
Συνεπώς, το επόμενο δένδρο της ακολουθίας προκύπτει με ένα split στο προηγούμενο. Το split που επιλέγεται είναι αυτό που ελαχιστοποιεί το σφάλμα με βάση την εξίσωση:

$$Gain = \frac{1}{2} \cdot [\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}] - \gamma \tag{4.40}$$

όπου οι όροι στη παραπάνω σχέση εκφράζουν την "ποιότητα" του αριστερού νέου φύλλου, την "ποιότητα" του δεξιού νέου φύλλου και την "ποιότητα" του φύλλου πριν πραγματοποιηθεί split σε αυτό. Το Gain εκφράζει ουσιαστικά τη βελτίωση του σφάλματος μετά από το εκάστοτε split και επιλέγεται εκείνο που έχει την υψηλότερη τιμή. Αξίζει να σημειωθεί ότι η παράμετρος γ, η οποία ορίζεται από τον προγραμματιστή, αποτελεί ουσιαστικά ένα threshold που καθορίζει το ελάχιστο επίπεδο βελτίωσης που πρέπει να επιτευχθεί προκειμένου να συμβεί το split.

4.1.3.5 Long Short Term Memory (LSTM)

Τα LSTM [47] ανήκουν στην κατηγορία των αναδρομικών νευρωνικών δικτύων και αποτελούν μια προέκταση αυτών. Ο βασικότερος τους στόχος είναι η αντιμετώπιση ενός προβλήματος που εντοπίζεται στα RNN και ονομάζεται Vanishing Gradient. Το πρόβλημα αυτό εμφανίζεται όταν το απλό RNN τροφοδοτείται με μεγάλες ακολουθίες εισόδου, οπότε και η έξοδος επηρεάζεται κυρίως από τα τελευταία στοιχεία της ακολουθίας και ελάχιστα από τα αρχικά. Προκειμένου να αντιμετωπιστεί αυτό, τα LSTM εισάγουν ένα μεγαλύτερο βαθμό πολυπλοκότητας στην αρχιτεκτονική του εσωτερικού των κελιών τους.



Πηγή: https://colah.github.io/posts/2015-08-Understanding-LSTMs/img/LSTM3-SimpleRNN.png

Σχήμα 4.13: Κελί απλού RNN

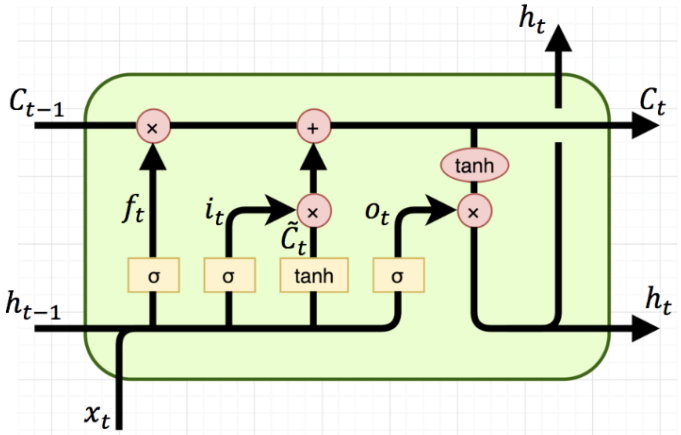
Η δομή ενός κελιού σε ένα απλό RNN αποτελείται από ένα επίπεδο. Η έξοδος του κελιού τη χρονική στιγμή t εκφράζεται από τη σχέση:

$$h(t) = tanh(W_{hh} \cdot h_{t-1} + W_{hx} \cdot x_t + b_h) \tag{4.41}$$

όπου W_{hh} είναι ο πίνακας βαρών των κρυφών μονάδων, h_{t-1} είναι η κρυφή κατάσταση της προηγούμενης χρονικής στιγμής $t-1$, W_{hx} είναι ο πίνακας βαρών των χαρακτηριστικών

της εισόδου, x_t είναι η είσοδος τη χρονική στιγμή t, b_h είναι η σταθερά βιας και $tanh$ η συνάρτηση υπερβολικής εφαπτομένης που χρησιμοποιείται ως συνάρτηση ενεργοποίησης.

Η δομή ενός κελιού σε ένα LSTM είναι πιο περίπλοκη και αποτελείται από 4 επίπεδα. Η πολυπλοκότητα του αυτή είναι που το καθιστά πιο ικανό στη διαχείριση πληροφοριών που περιέχονται σε μεγάλες ακολουθίες. Ένα πολύ σημαντικό χαρακτηριστικό που εισάγεται είναι η κατάσταση κελιού (cell state). Πρόκειται για την οριζόντια γραμμή που απεικονίζεται στο πάνω μέρος του παρακάτω σχήματος και η οποία διατρέχει ολόκληρη την αλυσίδα κελιών. Κάθε κελί περιέχει τρεις δομές που ονομάζονται πύλες (gates) οι οποίες του επιτρέπουν να ρυθμίζει τον όγκο της πληροφορίας πληροφορίας που εισάγει στην κατάσταση κελιού. Αυτό επιτυγχάνεται με τη χρήση της σιγμοειδούς συνάρτησης (βλέπε σχήμα 3.2) της οποίας η έξοδος κυμαίνεται στο διάστημα [0, 1]. Μια τιμή που είναι κοντά στο 0 σημαίνει αποκοπή ολόκληρης της πληροφορίας, ενώ μια τιμή που είναι κοντά στο 1 σημαίνει ελεύθερη διέλευση της πληροφορίας.



Πηγή: https://cdn-images-1.medium.com/max/1200/1*-kBdBYzR7lpimgb3AIRkOw.png

Σχήμα 4.14: Κελί LSTM

Αρχικά, το πρώτο επίπεδο του κελιού δέχεται ως είσοδο την κατάσταση του προηγούμενου κελιού h_{t-1} και την τρέχουσα είσοδο x_t. Η είσοδος διέρχεται μέσω ενός σιγμοειδούς επιπέδου, το οποίο καλείται forget gate layer, και μέσω του οποίου αποφασίζεται το ποσό της πληροφορίας που θα απορριφθεί.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \text{ (forget gate layer)} \tag{4.42}$$

όπου W_f ένας πίνακας βαρών και b_f μια σταθεράς bias. Στη συνέχεια, αποφασίζεται η πληροφορία που θα αποθηκευτεί στο κελί. Αυτό πραγματοποιείται σε δύο μέρη. Πρώτα, ένα σιγμοειδές επίπεδο, το οποίο καλείται input gate layer, αποφασίζει ποιες τιμές θα ενημερωθούν, ενώ στη συνέχεια ένα επίπεδο υπερβολικής εφαπτομένης δημιουργεί ένα διάνυσμα υποψήφιων τιμών, $\widetilde{C}_t$, το οποίο ίσως προστεθεί στην κατάσταση κελιού.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \text{ (input gate layer)} \tag{4.43}$$

$$\widetilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_c) \text{ (cell input)} \tag{4.44}$$

όπου W_i, W_C είναι πίνακες βαρών και b_i, b_C είναι σταθερές bias. Ακολουθεί η ενημέρωση της παλιάς κατάστασης του κελιού, C_{t-1}, στη νέα κατάσταση C_t. Αυτό

επιτυγχάνεται πολλαπλασιάζοντας την παλιά κατάσταση με την έξοδο του forget gate επιπέδου (ξεχνιέται η πληροφορία που αποφασίστηκε να ξεχαστεί) και στη συνέχεια προσθέτοντας το γινόμενο της εξόδου του input gate επιπέδου με το cell input (προστίθεται η πληροφορία που αποφασίστηκε να προστεθεί).

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{4.45}$$

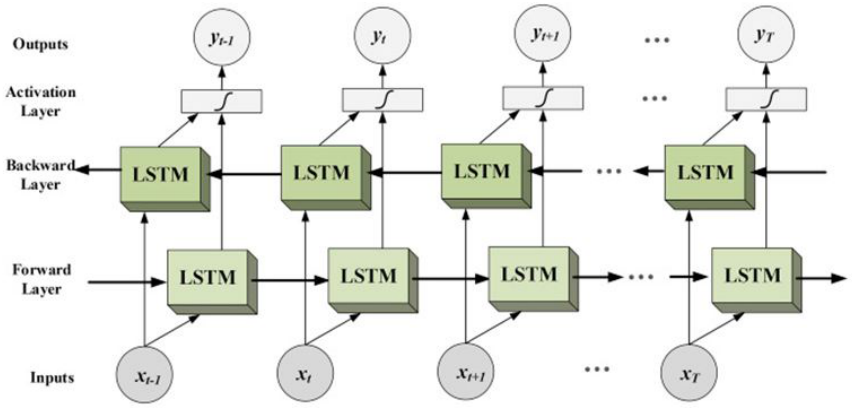
Τέλος, υπολογίζεται η έξοδος εφαρμόζοντας τη συνάρτηση υπερβολικής εφαπτομένης στη κατάσταση κελιού που έχει υπολογιστεί. Προηγουμένως, η κατάσταση κελιού έχει περάσει από ένα σιγμοειδές επίπεδο, το οποίο καλείται output gate layer, προκειμένου να αποφασιστεί ποια μέρη του cell state θα συμβάλλουν στην έξοδο.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \text{ (output gate layer)} \tag{4.46}$$

$$h_t = o_t \cdot C_t \text{ (hidden state)} \tag{4.47}$$

όπου W_o ένας πίνακας βαρών και b_o μια σταθερά bias.

Μια προσθήκη που μπορεί να ενισχύσει τη λειτουργία των LSTM είναι η αμφίδρομη ροή πληροφορίας. Ιδιαίτερα όταν τα δεδομένα εισόδου είναι ακολουθίες μια τέτοια τεχνική μπορεί να αποδειχτεί εξαιρετικά χρήσιμη. Στα μονόδρομα LSTM η κατάσταση ενός κελιού μπορεί να εξαρτάται από προηγούμενες καταστάσεις, όπως παρουσιάστηκε παραπάνω, ή και από επόμενες (αν η είσοδος τροφοδοτούταν με την αντίστροφη σειρά). Η αρχιτεκτονική των αμφίδρομων LSTM (Bi-LSTM ή Bidirectional LSTM) [48] επιτρέπει την εξάρτηση μιας κατάστασης τόσο από παρελθοντικές όσο και από μελλοντικές καταστάσεις. Μπορεί κανείς να φανταστεί τα αμφίδρομα LSTM σαν δύο μονόδρομα LSTM (ένα πρόσθιας και ένας οπισθοδρομικής ροής της πληροφορίας) τα οποία συνεργάζονται και λειτουργούν παράλληλα.



Πηγή: https://www.i2tutorials.com/wp-content/media/2019/05/Deep-Dive-into-Bidirectional-LSTM-i2tutorials.jpg

Σχήμα 4.15: Αμφίδρομο LSTM

Η έξοδος προκύπτει από τη συνένωση των εξόδων των επιμέρους μονόδρομων LSTMs. Χάριν απλότητας, με δείκτη f θα συμβολίζεται το LSTM στο οποίο η πληροφορία έχει ροή από το παρελθόν προς το μέλλον, ενώ με δείκτη b θα συμβολίζεται το LSTM στο οποίο η πληροφορία έχει ροή από το μέλλον προς το παρελθόν.

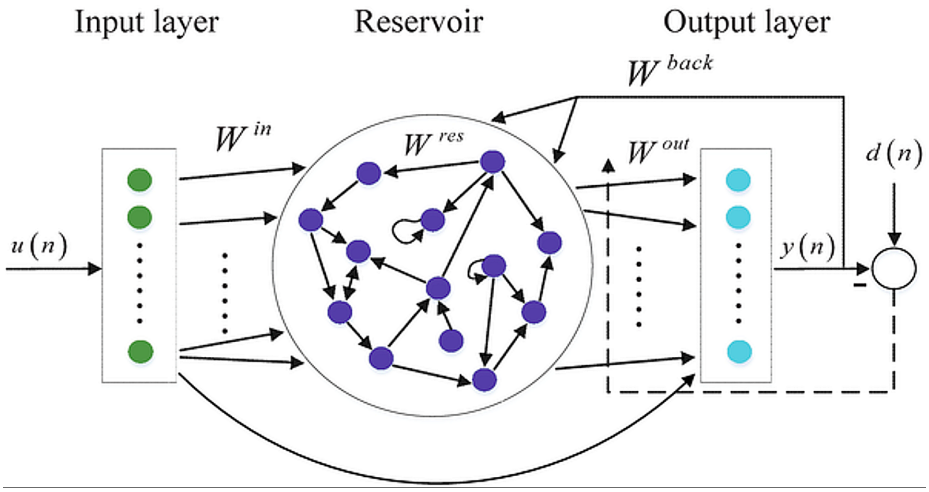
$$h_t^{(f)} = LSTM^{(f)}(x_t, h_{t-1}) \text{ (forward layer)} \tag{4.48}$$

$$h_t^{(b)} = LSTM^{(b)}(x_t, h_{t-1}) \text{ (backward layer)} \tag{4.49}$$

$$h_t = [h_t^{(f)}, h_t^{(b)}] \text{ (final hidden state)} \tag{4.50}$$

4.1.3.6 Echo State Networks (ESN)

Τα ESN [49] αποτελούν ένα παράδειγμα εφαρμογής μιας πιο γενικής έννοιας η οποία ονομάζεται Reservoir Computing [50]. Ως Reservoir Computing μπορεί να οριστεί μια μεθοδολογία μοντελοποίησης σύνθετων δυναμικών δεδομένων μέσω αναδρομικών νευρωνικών δικτύων. Πρόκειται για μια τεχνική που αντιμετωπίζει με μεγάλη επιτυχία το φαινόμενο Vanishing Gradient καθώς επιτρέπει στο δίκτυο να "θυμάται" παλαιότερες καταστάσεις. Σύμφωνα με αυτή, το νευρωνικό δίκτυο υπό μελέτη χωρίζεται σε τρία επιμέρους μικρότερα μέρη: το Input Layer, το Reservoir Layer και το Readout Layer.



Πηγή: https://journals.plos.org/plosone/article/figure/image?size=medium&id=10.1371/journal.pone.0181816.g001

Σχήμα 4.16: Η δομή του ESN

Αυτό που κάνει τα δίκτυα αυτού του τύπου να ξεχωρίζουν είναι το γεγονός ότι κατά τη φάση εκπαίδευσης ενημερώνονται μόνο τα βάρη των συνάψεων που ανήκουν στο Readout Layer. Τα βάρη των συνάψεων του Reservoir Layer, το οποίο συνιστά ένα αναδρομικό δίκτυο, αρχικοποιούνται κατά την εκπαίδευση του δικτύου και οι τιμές τους παραμένουν σταθερές σε όλη τη διάρκεια αυτής. Το ίδιο συμβαίνει και με τα βάρη του Input Layer. Ένα επιπλέον χαρακτηριστικό που διακρίνει τα ESN είναι το γεγονός ότι οι νευρώνες του Reservoir Layer είναι αραιά συνδεδεμένοι, ενώ οι συνάψεις μεταξύ τους σχηματίζονται με εντελώς τυχαίο τρόπο και δεν ακολουθούν κάποια προκαθορισμένη δομή. Στη συνέχεια, παρουσιάζεται η μαθηματική περιγραφή της λειτουργίας των ESN.

Έστω K το πλήθος των νευρώνων του Input Layer, N το πλήθος των νευρώνων του Reservoir Layer και L το πλήθος των νευρώνων του Readout Layer. Με W_{in} συμβολίζεται ο πίνακας διαστάσεων $N \times K$ που περιέχει τις τιμές για τα βάρη των συνάψεων που συνδέουν το Input με το Reservoir Layer, με W_{res} συμβολίζεται ο πίνακας διαστάσεων $N \times N$ που περιέχει τις τιμές για τα βάρη των συνάψεων των νευρώνων του Reservoir Layer, με W_{out} συμβολίζεται ο πίνακας διαστάσεων $L \times (K + N + L)$ που περιέχει τις τιμές για τα βάρη των συνάψεων που απολήγουν στους νευρώνες του Readout Layer και με W_{back} συμβολίζεται ο πίνακας διαστάσεων $N \times L$ που περιέχει τις τιμές για τα βάρη των συνάψεων που σχετίζονται με τις εξόδους που επιστρέφουν πίσω στο Reservoir Layer. Θα πρέπει να σημειωθεί, όπως φαίνεται και από τις διαστάσεις

των πινάκων που προαναφέρθηκαν, ότι είναι δυνατή η απευθείας σύνδεση νευρώνων του Input Layer με νευρώνες που ανήκουν στο Readout Layer καθώς και η σύνδεση μεταξύ νευρώνων του Readout Layer.

Συνεχίζοντας, έστω ακολουθία εισόδου της μορφής $U = [U_1, U_2, ..., U_n, ...]$. Τη χρονική στιγμή n η ενεργοποίηση των νευρώνων του Input, του Reservoir και του Readout Layer θα συμβολίζονται αντίστοιχα:

$$u(n) = (u_1(n), u_2(n), ..., u_K(n)) \tag{4.51}$$

$$x(n) = (x_1(n), x_2(n), ..., x_N(n)) \tag{4.52}$$

$$\hat{y}(n) = (\hat{y}_1(n), \hat{y}_2(n), ..., \hat{y}_L(n)) \tag{4.53}$$

Με τον όρο "echo state" εκφράζεται η ιδιότητα της κατάστασης $x(n)$ να θυμάται πληροφορίες και, επομένως, να εξαρτάται από όλες τις προηγούμενες εισόδους $u(n), u(n-1), ..., u(1)$.

Συμβολίζοντας με $f = [f_1, f_2, ..., f_N]$ και με $g = [g_1, g_2, ..., g_L]$ τις συναρτήσεις ενεργοποίησης των νευρώνων του Reservoir και του Readout Layer αντίστοιχα, η ενημέρωση ενός ESN τη χρονική στιγμή $n+1$ περιγράφεται από τις εξής σχέσεις:

$$x(n+1) = f(W_{in} \cdot u(n+1) + W_{res} \cdot x(n) + W_{back} \cdot \hat{y}(n)) \tag{4.54}$$

$$\hat{y}(n+1) = g(W_{out} \cdot [u(n+1), x(n+1), \hat{y}(n)]) \tag{4.55}$$

όπου με $[u(n+1), x(n+1), \hat{y}(n)]$ συμβολίζεται η συνένωση των επιμέρους στοιχείων.

Το ζητούμενο στην περίπτωση των ESN είναι ο προσδιορισμός των παραμέτρων του συστήματος, έτσι ώστε αυτό να έχει την ιδιότητα "echo state". Τελικά, αποδεικνύεται ότι για να έχει το σύστημα αυτήν την ιδιότητα ο πίνακας W_{res} εξαρτάται από τις εξής δύο συνθήκες:

1. Επαρκής συνθήκη: Η μέγιστη ιδιάζουσα τιμή του πίνακα W_{res} να είναι μικρότερη του 1. Οι ιδιάζουσες τιμές ενός πίνακας συμβολίζονται με $\sigma(\cdot)$, ενώ η μέγιστη ιδιάζουσα τιμή ενός πίνακα ισούται με τη δεύτερη νόρμα του πίνακα.

$$max(\sigma(W_{res})) = \|W_{res}\|_2 < 1 \tag{4.56}$$

2. Αναγκαία συνθήκη: Η φασματική ακτίνα του πίνακα W_{res} να είναι μικρότερη του 1. Η φασματική ακτίνα ενός πίνακα συμβολίζεται με $p(\cdot)$ και ισούται με τη μεγαλύτερη απόλυτη τιμή των ιδιοτιμών του.

$$p(W_{res}) < 1 \tag{4.57}$$

Στη συνέχεια, θα παρουσιαστεί η διαδικασία εκπαίδευσης των Echo State Networks. Δεδομένου συνόλου εκπαίδευσης μεγέθους T με δείγματα x_i και αναμενόμενες τιμές y_i, η διαδικασία έχει ως εξής:

- Αρχικοποίηση βαρών: Οι πίνακες W_{in}, W_{res} και W_{back} αρχικοποιούνται με τυχαίες τιμές. Προκειμένου να ικανοποιείται η συνθήκη για την "echo state" ιδιότητα του

συστήματος, ο πίνακας W_{res} ενημερώνεται με βάση την εξίσωση:

$$W_{res} = \frac{W_{res} \cdot \alpha}{p(W_{res})} \tag{4.58}$$

όπου α μια παράμετρος που ορίζεται από τον προγραμματιστή. Θα πρέπει να σημειωθεί ότι ο W_{res} πρέπει να σχηματιστεί με τέτοιο τρόπο ώστε να αποτελεί αραιό πίνακα.

- Εκπαίδευση:

  1. Αρχικοποίηση του "echo state" και της εξόδου (συνήθως $x(0) = 0$ και $y(0) = 0$).
  2. Το μοντέλο εκπαιδεύεται πάνω στα δεδομένα του συνόλου εκπαίδευσης και παράγονται οι αντίστοιχες έξοδοι σε κάθε επίπεδο. Σε κάθε βήμα i, λαμβάνεται η συνένωση $[u(i), x(i), \hat{y}(i-1)]$ και αποθηκεύεται σε έναν πίνακα M. Κατά τη φάση εκπαίδευσης χρησιμοποιείται μια τεχνική η οποία ονομάζεται Teacher Forcing. Σύμφωνα με αυτή, κατά το βήμα $i+1$ οι έξοδοι $x(i+1)$ και $y(i+1)$ υπολογίζονται με βάση την αναμενόμενη τιμή y_i και όχι με βάση αυτή που προέβλεψε το μοντέλο. Επιπλέον, οι αναμενόμενες τιμές y_i χρησιμοποιούνται ως είσοδοι στην αντίστροφη συνάρτηση υπερβολικής εφαπτομένης και οι έξοδοι $tanh^{-1}(y_i)$ τοποθετούνται σε έναν πίνακα T. Ακόμη, θα πρέπει να αναφερθεί ότι υπάρχει μια παράμετρος T_0 η οποία ορίζεται από τον προγραμματιστή και καθορίζει τη χρονική στιγμή μετά από την οποία θα αρχίσουν να συλλέγονται τα δεδομένα των πινάκων M και T. Αυτό συμβαίνει έτσι ώστε να προλάβει να σταθεροποιηθεί το σύστημα. Επομένως, όταν ολοκληρωθεί η εκπαίδευση του μοντέλου οι τελικές διαστάσεις των πινάκων M και T είναι $(T - T_0 + 1) \times (K + N + L)$ και $(T - T_0 + 1) \times L$ αντίστοιχα.
  3. Ο πίνακας βαρών W_{out} λαμβάνεται παίρνοντας τον ανάστροφο πίνακα του αποτελέσματος που ορίζει η παρακάτω σχέση.

     $$W_{out}^T = M^{-1} \cdot T \tag{4.59}$$

     Είναι προφανές ότι οι διαστάσεις του πίνακα που προκύπτει από τη παραπάνω εξίσωση είναι ίσες με $(K + N + L) \times L$. Συνεπώς, ο ανάστροφος πίνακας θα έχει τις αναμενόμενες διαστάσεις.

Κλείνοντας, θα πρέπει να αναφερθεί ότι ένα από τα πιο σημαντικά πλεονεκτήματα των Echo State Networks είναι το γεγονός ότι στο Readout επίπεδο μπορούν να τοποθετηθούν και να εκπαιδευτούν απλοί ταξινομητές, όπως για παράδειγμα ο SVM που παρουσιάστηκε παραπάνω, οι οποίοι από μόνοι τους δεν θα λάμβαναν υπόψιν παρελθοντικές καταστάσεις του προβλήματος. Αυτό είναι εξαιρετικά χρήσιμο, ιδιαίτερα σε προβλήματα των οποίων τα δεδομένα εισόδου είναι ακολουθίες, καθώς απλά μοντέλα δύναται να λειτουργούν σαν να είχαν μνήμη.

4.1.4 Μηχανική Μάθηση και Ευφυή Συστήματα Μεταφορών

Όπως έχει ήδη αναφερθεί στο υποκεφάλαιο 3.2.3, η μηχανική μάθηση αποτελεί το πιο ευρέως διαδεδομένο μέσο εξόρυξης γνώσης από δεδομένα μεγάλου όγκου. Για τον

λόγο αυτό, η μηχανική μάθηση χρησιμοποιείται σε μεγάλο βαθμό σε τεχνολογίες και συστήματα που σχετίζονται με τις ευφυείς μεταφορές. Στη συνέχεια, παρουσιάζονται κάποια παραδείγματα εφαρμογής της μηχανικής μάθησης σε συστήματα που αφορούν την αξιολόγηση και την ανάλυση της οδηγικής συμπεριφοράς.

Στο [51] υλοποιείται ένα σύστημα βαθμολόγησης της συμπεριφοράς του εκάστοτε οδηγού με βάση το βαθμό επικινδυνότητας της κάθε πράξης του. Τα δεδομένα που χρησιμοποιούνται συλλέγονται είτε άμεσα από το όχημα (ψύρα On-Board Diagnostics) είτε έμμεσα μέσω έξυπνων κινητών τηλεφώνων (smartphones). Ο βαθμός επικινδυνότητας κάποιας πράξης υπολογίζεται με εφαρμογή των αλγορίθμων Support Vector Machines και Decision Tree στα δεδομένα που έχουν συλλεχθεί για την αντίστοιχη χρονική στιγμή.

Στο [52] υλοποιείται ένα σύστημα εντοπισμού πιθανών επικίνδυνων ενεργειών που πραγματοποιούνται από τον οδηγό. Τα δεδομένα που χρησιμοποιήθηκαν προέρχονται από το driverBehaviorDataset [53]. Το σύστημα αποτελείται από δύο ταξινομητές, έναν για την εύρεση κάποιας επικίνδυνης πράξης και έναν για την λεπτομερέστερη κατηγοριοποίηση αυτής. Οι ταξινομητές υλοποιούνται είτε ως ένα αναδρομικό νευρωνικό δίκτυο LSTM είτε ως ένα μοντέλο που έχει εκπαιδευτεί με τον αλγόριθμο Random Forest.

Στο [54] υλοποιείται ένα σύστημα παρόμοιο με αυτό που αναφέρθηκε παραπάνω, με τη μόνη διαφορά ότι χρησιμοποιείται ένας ταξινομητής. Τα δεδομένα που χρησιμοποιήθηκαν προέρχονται από το driverBehaviorDataset. Ο ταξινομητής υλοποιείται είτε ως ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης είτε ως ένα μοντέλο που έχει εκπαιδευτεί με έναν αλγόριθμο από τους Random Forest ή SVM.

Στο [55] υλοποιείται ένα σύστημα σύστημα συνολικότερης κατηγοριοποίησης της συμπεριφοράς ενός οδηγού. Τα δεδομένα που χρησιμοποιήθηκαν ανήκουν σε δύο κατηγορίες (normal και hard) οι οποίες σχετίζονται με το "στυλ" του οδηγού. Ένας ταξινομητής χρησιμοποιείται για την εξαγωγή γνώσης από τα δεδομένα που έχουν συλλεχθεί και την περαιτέρω κατάταξη του "στυλ" κάποιου οδηγού με βάση αυτά. Ο ταξινομητής υλοποιείται είτε ως ένα νευρωνικό δίκτυο πρόσθιας τροφοδότησης είτε ως ένα μοντέλο που έχει εκπαιδευτεί με έναν αλγόριθμο από τους Random Forest, Decision Tree ή k-Nearest Neighbors.

4.2 Χρονοσειρές (Time Series)

4.2.1 Ορισμός και Βασικές Έννοιες

Ο όρος χρονοσειρά [56] αναφέρεται σε ένα διατεταγμένο σύνολο παρατηρήσεων οι οποίες λαμβάνονται ανά χρονικά διαστήματα τα οποία ισαπέχουν μεταξύ τους. Πρόκειται για ένα τρόπο περιγραφής της συμπεριφοράς μιας ή περισσότερων μεταβλητών σε σχέση με το χρόνο. Ανάλογα με το πλήθος των μεταβλητών που μελετώνται οι χρονοσειρές μπορούν να διακριθούν σε δύο μεγάλες κατηγορίες:

- Μονοδιάστατες χρονοσειρές: Η χρονοσειρά σχετίζεται με μια μεταβλητή.

$$\langle (t_1, x(t_1)), (t_2, x(t_2)), ..., (t_n, x(t_n)) \rangle \tag{4.60}$$

- Πολυδιάστατες χρονοσειρές: Η χρονοσειρά σχετίζεται με περισσότερες από μία μεταβλητές.

$$\langle (t_1, \langle x_1(t_1), x_2(t_1), ..., x_m(t_1) \rangle), (t_2, \langle x_1(t_2), x_2(t_2), ..., x_m(t_2), \\ ..., (t_n, \langle x_1(t_n), x_2(t_n), ..., x_m(t_n) \rangle) \rangle \tag{4.61}$$

Μια πολυδιάστατη χρονοσειρά μπορεί να ερμηνευθεί ως ένα σύνολο που απαρτίζεται από πολλές παράλληλες μονοδιάστατες χρονοσειρές. Οι μεταβλητές μιας πολυδιάστατης σειράς δεν είναι ανεξάρτητες μεταξύ τους αλλά, αντιθέτως, αλληλεπιδρούν μεταξύ τους και επηρεάζουν την κατάσταση του συστήματος υπό μελέτη.

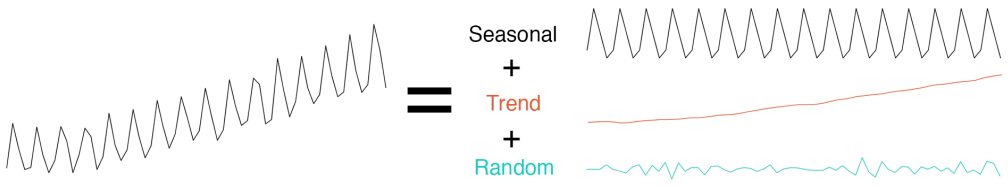
Συνεχίζοντας, παρουσιάζονται τα κυριότερα χαρακτηριστικά που ορίζουν μια χρονοσειρά. Αυτά είναι τα εξής:

- Τάση (Trend): Η τάση εκφράζει την "κατεύθυνση" (αύξηση ή μείωση) που έχουν, κατά μέσο όρο, οι τιμές της χρονοσειράς σε βάθος χρόνου.
- Περιοδικότητα (Periodicity): Η περιοδικότητα εκφράζει την επανάληψη μοτίβων που σχετίζονται με τις τιμές που παίρνει η χρονοσειρά στον άξονα του χρόνου.
- Στασιμότητα (Stationarity): Μια χρονοσειρά λέγεται στάσιμη όταν οι διακυμάνσεις των τιμών της δεν διαφοροποιούνται με το χρόνο. Αντίθετα, μια μη-στάσιμη χρονοσειρά μπορεί να έχεις τάσεις και να εμφανίζει περιοδικότητα.
- Στοχαστικότητα (Stochasticity): Οι τιμές των μεταβλητών μια χρονοσειράς μπορεί να περιέχουν θόρυβο ή/και να αλλάζουν με κάποιον βαθμό τυχαιότητας. Επομένως, μια χρονοσειρά μπορεί να θεωρηθεί ως μια στοχαστική διαδικασία. Η μεγαλύτερη πρόκληση στην ανάλυση των χρονοσειρών είναι ο εντοπισμός του ντετερμινιστικού μέρους του συστήματος που παράγει τη χρονοσειρά, ώστε να προσδιοριστεί ένα κατάλληλο μοντέλο που να περιγράφει ικανοποιητικά τα δεδομένα.

Με βάση όσα έχουν ειπωθεί παραπάνω, μια χρονοσειρά μπορεί να οριστεί ως εξής:

$$x_t = \mu_t + s_t + y_t \tag{4.62}$$

όπου μ_t είναι η συνιστώσα της τάσης, s_t είναι η συνιστώσα της περιοδικότητας και y_t είναι η συνιστώσα που μένει αν αφαιρεθούν η τάση και η περιοδικότητα από την αρχική χρονοσειρά. Η διαδικασία κατά την οποία μια χρονοσειρά διακρίνεται στα επιμέρους συστατικά της ονομάζεται αποσύνθεση.



Πηγή: https://anomaly.io/wp-content/uploads/2015/12/time-series-decomposition-seasonal-trend.png

Σχήμα 4.17: Αποσύνθεση χρονοσειράς

Η τάση και η περιοδικότητα δεν περιέχουν πληροφορία για τη δυναμική του συστήματος, την εξάρτηση της παρατήρησης x_t , δηλαδή, από τις προηγούμενες παρατηρήσεις. Σε προβλήματα στα οποία στόχος είναι η διερεύνηση της δυναμικής του συστήματος είναι πολύ σημαντικό να αφαιρεθούν οι συνιστώσες τάσης και περιοδικότητας από τη χρονοσειρά.

Σε αυτό το σημείο, θα πρέπει να αναφερθεί ότι, αφού αντικείμενο της παρούσης διπλωματικής εργασίας αποτελεί η ανάλυση ενός συνόλου οδηγικών δεδομένων τα οποία έχουν συλλεχθεί από αισθητήρες που λειτουργούν με υψηλή συχνότητα δειγματοληψίας, το ενδιαφέρον αυτού του κεφαλαίου εστιάζεται κυρίως στην ανάλυση πολυδιάστατων χρονοσειρών με στόχο την κατηγοριοποίηση τους. Η αποσύνθεση χρονοσειρών αποτελεί, αναμφίβολα, μια εξαιρετικά χρήσιμη διαδικασία η οποία όμως κρίνεται ότι συνεισφέρει περισσότερο σε προβλήματα παλινδρόμησης [57] και σε περιπτώσεις που το μήκος των χρονοσειρών είναι αρκετά μεγάλο (μήνες ή χρόνια), και επομένως δεν θα αναλυθεί περαιτέρω.

4.2.2 Μέθοδοι Κατηγοριοποίησης Χρονοσειρών

Η κατηγοριοποίηση δεδομένων που έχουν μορφή χρονοσειράς είναι ένα πρόβλημα που έχει απασχολήσει έντονα την επιστημονική κοινότητα. Τα βασικότερο ερώτημα που πρέπει να απαντηθεί είναι: ποιος είναι ο τρόπος με βάση τον οποίον μπορούν να συγκριθούν τα δεδομένα δύο χρονοσειρών;

Για την προσέγγιση ενός τέτοιου προβλήματος έχουν προσδιοριστεί κάποιες μέθοδοι [58, 59, 60] οι οποίες αναλύονται σε αυτό το υποκεφάλαιο.

4.2.2.1 Κατηγοριοποίηση Βασισμένη στα Χαρακτηριστικά (Feature Based Classification)

Πρόκειται για μία μέθοδο σύμφωνα με την οποία η περιγραφή των χρονοσειρών ορίζεται από στατιστικές μετρήσεις που εξάγονται από τα χαρακτηριστικά τους, και όχι από τα ίδια τα χαρακτηριστικά [61]. Για το σκοπό αυτό χρησιμοποιείται ένα κυλιόμενο παράθυρο σταθερού μήκους k. Για κάθε μεταβλητή του συνόλου δεδομένων που ορίζει το κυλιόμενο παράθυρο λαμβάνονται στατιστικές μετρήσεις όπως η μέση τιμή, η διακύμανση, η μέγιστη τιμή, η ελάχιστη τιμή, η εντροπία κ.α. . Για παράδειγμα, θεωρώντας μια πολυδιάστατη χρονοσειρά, η οποία περιγράφεται από την εξίσωση που παρουσιάστηκε παραπάνω, για κάθε χαρακτηριστικό των πρώτων k στοιχείων μπορούν να ληφθούν οι εξής μετρήσεις:

$$E^{(1-k)}[x_1], \sigma^{(1-k)}[x_1], E^{(1-k)}[x_2], \sigma^{(1-k)}[x_2], ..., E^{(1-k)}[x_m], \sigma^{(1-k)}[x_m] \tag{4.63}$$

όπου ο δείκτης $(1-k)$ προσδιορίζει το διάστημα για το οποίο υπολογίστηκαν οι τιμές, E είναι η μέση τιμή του εκάστοτε χαρακτηριστικού για το διάστημα που υποδεικνύει ο δείκτης και σ είναι διακύμανση του εκάστοτε χαρακτηριστικού για το διάστημα που υποδεικνύει ο δείκτης. Στη συνέχεια, επαναλαμβάνεται η διαδικασία υπολογισμού των μετρήσεων για τα επόμενα k στοιχεία:

$$E^{(k+1)-2\cdot k}[x_1], \sigma^{(k+1)-2\cdot k}[x_1], E^{(k+1)-2\cdot k}[x_2], \sigma^{(k+1)-2\cdot k}[x_2], ..., E^{(k+1)-2\cdot k}[x_m], \sigma^{(k+1)-2\cdot k}[x_m] \tag{4.64}$$

Η διαδικασία επαναλαμβάνεται μέχρις ότου να συμπεριλάβει όλα τα στοιχεία που ορίζουν την χρονοσειρά. Τελικά, όλες αυτές οι στατιστικές μετρήσεις μπορούν να συνδυαστούν για να σχηματίσουν ένα νέο μονοδιάστατο διάνυσμα χαρακτηριστικών το οποίο περιγράφει τη χρονοσειρά και μπορεί εύκολα να τροφοδοτηθεί σε έναν απλό ταξινομητή χωρίς μνήμη.

Θα πρέπει να σημειωθεί ότι αυτή η μέθοδος παρουσιάζει αρκετά αδύναμα σημεία. Αρχικά, ο προσδιορισμός του κατάλληλου μεγέθους k για το κυλιόμενο παράθυρο είναι μια χρονοβόρα διαδικασία η οποία γίνεται στα τυφλά μέσω δοκιμής διαφόρων τιμών. Επίσης, το μέγεθος του τελικού διανύσματος που σχηματίζεται μπορεί να προκύψει αρκετά μεγάλο, γεγονός που αποτελεί εμπόδιο στην αποδοτική χρησιμοποίησή του. Τέλος, είναι πολύ πιθανό να υπάρξει σημαντική απώλεια πληροφορίας καθώς χρησιμοποιούνται στατιστικές μετρήσεις που εξάγονται από τα χαρακτηριστικά, και όχι τα ίδια τα χαρακτηριστικά.

4.2.2.2 Dynamic Time Warping (DTW)

Δεδομένων δύο χρονοσειρών $X = \{X_1, X_2, ..., X_N\}$ και $Y = \{Y_1, Y_2, ..., Y_M\}$ μήκους N και M αντίστοιχα, ο αλγόριθμος DTW [62] προσπαθεί, βάσει κάποιων περιορισμών, να υπολογίσει την ομοιότητα μεταξύ των χρονοσειρών X και Y. Θα πρέπει να σημειωθεί ότι τα $X_1, X_2, ..., X_N$ και $Y_1, Y_2, ..., Y_M$ μπορεί να είναι απλά μονοδιάστατα χαρακτηριστικά ή διανύσματα χαρακτηριστικών.

Συνεχίζοντας, η ομοιότητα μεταξύ των δύο χρονοσειρών μπορεί να εκφραστεί ως ένα μονοπάτι το οποίο καλείται warping path. Ως warping path ορίζεται μια ακολουθία μήκους L της μορφής:

$$w = (w_1, ..., w_L) \text{ με } w_i = (n_i, m_i) \in [1, ...N] \times [1, ..., M], \forall i \in \{1, ...L\} \tag{4.65}$$

και η οποία πληροί τις εξής προϋποθέσεις:

- Οριακή συνθήκη (Boundary Condition):

$$w_1 = (1, 1) \text{ και } w_L = (N, M) \tag{4.66}$$

- Συνθήκη μονοτονίας (Monotonicity Condition):

$$n_1 \leq n_2 \leq ... \leq n_L \text{ και } m_1 \leq m_2 \leq ... \leq m_L \tag{4.67}$$

- Βηματική συνθήκη (Step-size Condition):

$$w_{i+1} - w_i \in \{(1, 0), (0, 1), (1, 1)\}, \forall i \in \{1, ..., L\} \tag{4.68}$$

Με απλά λόγια, ένα warping path ορίζει μια αντιστοίχιση μεταξύ δύο χρονοσειρών X και Y αναθέτοντας το στοιχείο X_{n_i} της χρονοσειράς X σε κάποιο στοιχείο Y_{m_i} της χρονοσειράς Y. Σε αυτήν την περίπτωση η χρονοσειρά Y καλείται χρονοσειρά αναφοράς. Η οριακή συνθήκη εξασφαλίζει ότι τα πρώτα και τα τελευταία στοιχεία των χρονοσειρών X και Y ευθυγραμμίζονται μεταξύ τους. Η συνθήκη μονοτονίας καθορίζει την κατεύθυνση που ακολουθεί το warping path. Πιο συγκεκριμένα, εξασφαλίζει ότι εάν ένα στοιχείο της χρονοσειράς X προηγείται ενός δεύτερου

στοιχείου της χρονοσειράς X, το ίδιο θα πρέπει να ισχύει και για τα αντίστοιχα στοιχεία της χρονοσειράς Y με τα οποία έχουν ευθυγραμμιστεί τα προαναφερθέντα στοιχεία της χρονοσειράς Y. Τέλος, η βηματική συνθήκη εξασφαλίζει ότι όλα τα στοιχεία και των δύο χρονοσειρών θα συμπεριληφθούν στο μονοπάτι και αποτελεί, ουσιαστικά, μια συνθήκη συνέχειας του μονοπατιού. Προκειμένου να γίνει πιο σαφής η έννοια του warping path παρουσιάζεται ένα παράδειγμα για δύο χρονοσειρές X και Y με μήκη $N = 9$ και $M = 7$ αντίστοιχα.

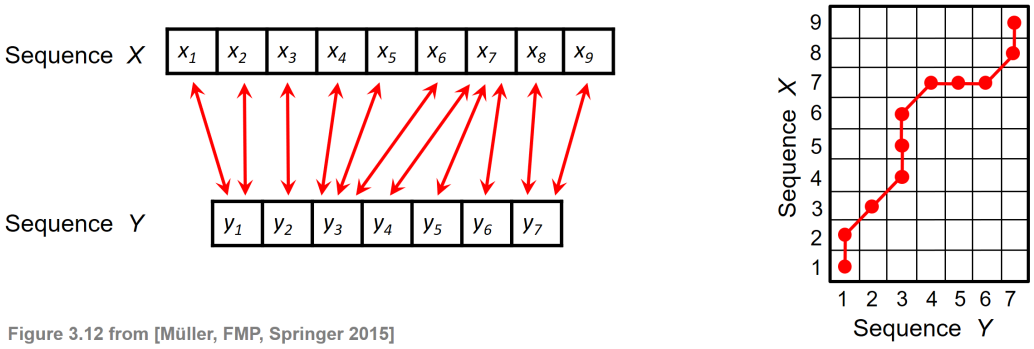


Figure 3.12 from [Müller, FMP, Springer 2015]

Πηγή: https://www.audiolabs-erlangen.de/resources/MIR/FMP/data/C3/FMP_C3_F12.png

Σχήμα 4.18: Warping Path

Είναι προφανές ότι μπορεί να υπάρχουν πολλά warping paths για δύο χρονοσειρές X και Y. Συνεπώς, πρέπει να προσδιοριστεί ένας τρόπος μέτρησης της ποιότητας ενός warping path. Για το σκοπό αυτό χρησιμοποιείται μια συνάρτηση κόστους $c(\cdot)$ η οποία μετράει την απόσταση μεταξύ δύο στοιχείων των χρονοσειρών X και Y ως εξής:

$$c(X_n, Y_m),\ n \in \{1, ..., N\},\ m \in \{1, ..., M\} \tag{4.69}$$

Όσο χαμηλότερη είναι η τιμή της συνάρτησης κόστους τόσο περισσότερο μοιάζουν τα στοιχεία X_n και Y_m. Με βάση τη συνάρτηση κόστους μπορεί να οριστεί ένας πίνακας κόστους C με διαστάσεις $N \times M$ και του οποίου το κάθε κελί περιέχει την έξοδο της συνάρτησης κόστους για το αντίστοιχο ζευγάρι των χρονοσειρών X και Y.

$$C(n, m) = c(X_n, Y_m),\ n \in \{1, ..., N\},\ m \in \{1, ..., M\} \tag{4.70}$$

Το συνολικό κόστος ενός warping path υπολογίζεται ως εξής:

$$c_w = \sum_{i=1}^{L} c(X_{n_i}, Y_{m_i}) = \sum_{i=1}^{L} C(n_i, m_i) \tag{4.71}$$

Αυτό σημαίνει ότι το κόστος ενός warping path ισούται με το άθροισμα των επιμέρους κοστών των κελιών που διατρέχει. Ένα warping path θεωρείται "καλό" όταν το συνολικό κόστος είναι χαμηλό, ενώ θεωρείται "κακό" όταν το συνολικό κόστος είναι υψηλό. Ο αλγόριθμος DTW προσπαθεί να υπολογίσει το βέλτιστο warping path (optimal warping path) w_{opt} μεταξύ δύο χρονοσειρών X και Y, όπου ως βέλτιστο ορίζεται αυτό με το χαμηλότερο κόστος. Επομένως, η απόσταση DTW μεταξύ δύο χρονοσειρών X και Y ορίζεται ως εξής:

$$DTW(X, Y) = c_{w_{opt}}(X, Y) = min\{c_w(X, Y) : w \text{ ένα warping path}\} \tag{4.72}$$

Το πρόβλημα εύρεσης του βέλτιστου warping path λύνεται αποδοτικά με δυναμικό προγραμματισμό. Επιπλέον, η ταχύτητα σύγκλισης και η απόδοση του αλγορίθμου βελτιώνονται αισθητά με την εισαγωγή μιας παραμέτρου η οποία ονομάζεται warping window. Με τη χρήση αυτής της παραμέτρου εξετάζονται μόνο μονοπάτια τα οποία κινούνται κοντά στη διαγώνιο του πίνακα που ορίζεται από τις χρονοσειρές X, Y. Πρόκειται για μια επιθυμητή ιδιότητα καθώς ένα warping path του οποίου μεγάλο μέρος κινείται αμιγώς οριζόντια ή αμιγώς κάθετα υποδηλώνει ότι ένα στοιχείο της μιας χρονοσειράς έχει αντιστοιχηθεί με πολλά στοιχεία της άλλης, και έτσι το κριτήριο ομοιότητας υπόκειται σε μεγάλη χρονική παραμόρφωση. Ακόμη, ορίζεται και μια παραλλαγή του DTW έτσι ώστε να μην αντιστοιχίζονται όλα τα στοιχεία των δύο χρονοσειρών μεταξύ τους, αλλά όλα τα στοιχεία της χρονοσειράς με το μικρότερο μήκος να αντιστοιχίζονται με κάποια στοιχεία της άλλης χρονοσειράς. Αυτό επιτυγχάνεται καταπατώντας την οριακή συνθήκη και επιτρέποντας στο warping path να ξεκινάει και να τελειώνει σε διαφορετικά σημεία από την κάτω αριστερή γωνία και την πάνω δεξιά γωνία αντίστοιχα [63].

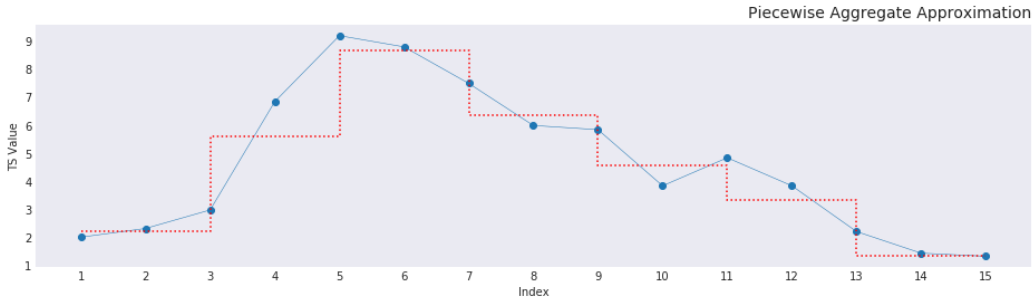
Είναι ξεκάθαρο το πως θα μπορούσε να χρησιμοποιηθεί ο αλγόριθμος Dynamic Time Warping σε συνδυασμό με ένα μοντέλο όπως είναι ο ταξινομητής kNN. Ωστόσο, καθώς ο kNN αποτελεί ένα μοντέλο το οποίο θεωρείται αδύναμο (τουλάχιστον με όσα άλλα μοντέλα παρουσιάστηκαν στο υποκεφάλαιο 4.1.3) έχουν γίνει αρκετές προσπάθειες έτσι ώστε να συνδυαστεί η χρήση του DTW με άλλα μοντέλα τα οποία δεν βασίζουν τη διαδικασία κατηγοριοποίησης στις αποστάσεις μεταξύ των δεδομένων [63, 64]. Η βασική ιδέα είναι ότι τα δεδομένα του συνόλου εκπαίδευσης ομαδοποιούνται (clustering) με βάση την ετικέτα που αυτά φέρουν. Στη συνέχεια, για κάθε πιθανή ετικέτα εξάγεται μια γενική χρονοσειρά αναφοράς που την αντιπροσωπεύει (template). Προκειμένου να ταξινομηθεί ένα νέο δείγμα υπολογίζεται η DTW απόσταση του από όλα τα templates που έχουν σχηματισθεί. Τελικά, ως είσοδος στο μοντέλο τροφοδοτείται ένα διάνυσμα που έχει μήκος ίσο με το πλήθος των πιθανών κατηγοριών του προβλήματος και του οποίου κάθε στοιχείο είναι η απόσταση του δείγματος υπό εξέταση από το εκάστοτε template.

4.2.2.3 Symbolic Aggregate Approximation (SAX)

Όπως υποδηλώνει και το όνομά του, ο αλγόριθμος SAX [65] χρησιμοποιείται έτσι ώστε να σχηματιστεί ένα string συμβόλων το οποίο αναπαριστά μια χρονοσειρά. Η λειτουργία του αλγορίθμου προϋποθέτει ότι η χρονοσειρά θα δίνεται σε μορφή PAA (Piecewise Aggregate Approximation). Δεδομένης μια χρονοσειράς $X = \{X_1, X_2, ..., X_N\}$ μήκους N, ο μετασχηματισμός PAA μιας χρονοσειράς δίνεται από τον εξής τύπο:

$$\bar{X} = \{\bar{X_1}, \bar{X_2}, ..., \bar{X_M}\},\ \bar{X_i} = \frac{M}{N} \cdot \sum_{j=N/(M-1)+1}^{N/M \cdot i} X_j \tag{4.73}$$

όπου M μια παράμετρος που ορίζει το μήκος της μετασχηματισμού PAA της χρονοσειράς. Ουσιαστικά ο μετασχηματισμός PAA μιας χρονοσειράς χρησιμοποιείται προκειμένου να μειωθεί το μήκος της από N σε M. Αυτό επιτυγχάνεται χωρίζοντας την αρχική χρονοσειρά σε M πλαίσια ίσου μήκους και παίρνοντας την μέση τιμή κάθε πλαισίου.



Πηγή: https://vigne.sh/images/PAAStepFinal.png

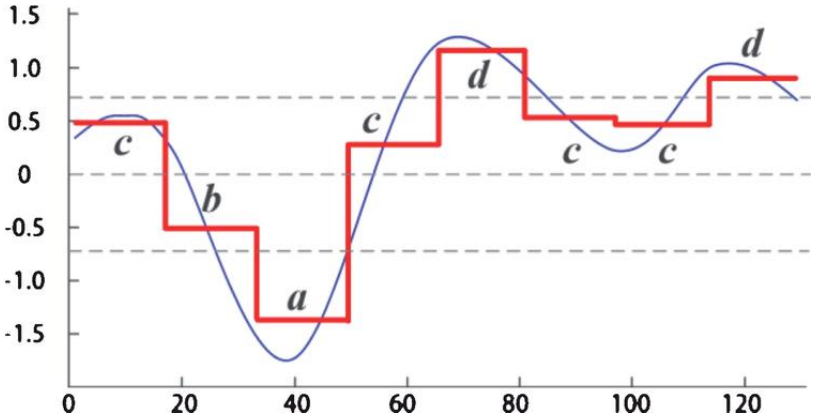
Σχήμα 4.19: Μετασχηματισμός PAA

Για τη σωστή λειτουργία του αλγορίθμου SAX, πριν πραγματοποιηθεί ο μετασχηματισμός PAA μιας χρονοσειράς θα πρέπει να έχει προηγηθεί η Z-κανονικοποίηση αυτής. Δεδομένης μια χρονοσειράς $X = \{X_1, X_2, ..., X_N\}$ η νέα Z-κανονικοποιημένη μορφή της δίνεται από τον εξής τύπο:

$$X_z = \{X_{z1}, ..., X_{zN}\},\ X_{zi} = \frac{x_i - \mu}{\sigma} \tag{4.74}$$

όπου μ και σ η μέση τιμή και η απόκλιση της χρονοσειράς αντίστοιχα.

Αφού έχουν προηγηθεί, λοιπόν, οι δύο διαδικασίες που αναφέρθηκαν μπορεί να υποτεθεί ότι η νέα κανονικοποιημένη χρονοσειρά μήκους M που έχει προκύψει ακολουθεί την κανονική κατανομή. Συνεπώς, με τη χρήση κάποιων στατιστικών πινάκων (των οποίων η ανάλυση ξεφεύγει αρκετά από τα όρια αυτής της διπλωματικής) είναι δυνατή η αντιστοίχιση των τιμών των M πλαισίων της χρονοσειράς σε σύμβολα. Έτσι, προκύπτει και η αναπαράσταση SAX της αρχικής χρονοσειράς.



Πηγή: https://content.iospress.com/media/ifs/2019/36-6/ifs-36-6-ifs181246/ifs-36-ifs181246-g001.jpg?width=755

Σχήμα 4.20: Μετασχηματισμός SAX

Για παράδειγμα η χρονοσειρά του σχήματος 4.20 μπορεί να αναπαρασταθεί από τη συμβολοσειρά "cbacdccd".

Δεδομένων δύο χρονοσειρών $X = \{X_1, X_2, ..., X_N\}$ και $Y = \{Y_1, Y_2, ..., Y_N\}$ μήκους N με αναπαραστάσεις SAX $X' = \{X'_1, X'_2, ..., X'_N\}$ και $Y' = \{Y'_1, Y'_2, ..., Y'_N\}$ μήκους M

αντίστοιχά, η απόσταση SAX μεταξύ των X και Y δίνεται από τον εξής τύπο:

$$SAXD(X', Y') = \sqrt{\frac{N}{M}} \cdot \sqrt{\sum_{i=1}^{M} dist^2(X'_i, Y'_i)} \tag{4.75}$$

όπου $dist(\cdot)$ μια συνάρτηση η οποία με τη χρήση ενός πίνακα αναζήτησης, που προκύπτει από στατιστικές ιδιότητες (των οποίων η ανάλυση ξεφεύγει αρκετά από τα όρια αυτής της διπλωματικής), αντιστοιχίζει την απόσταση δύο συμβόλων X'_i και Y'_i σε κάποιον πραγματικό αριθμό.

Σε αντίθεση με τον αλγόριθμο Dynamic Time Warping, στον αλγόριθμο SAX η απόσταση ορίζεται μόνο όταν οι δύο χρονοσειρές έχουν ίδιο μήκος.

4.2.2.4 Time Series Shapelets

Πρόκειται για μια πιο σύγχρονη προσέγγιση της οποίας η λειτουργία συνδυάζεται πολύ καλά με μοντέλα που χρησιμοποιούν δένδρα απόφασης. Την πρώτη φορά που προτάθηκε η μέθοδος Time Series Shapelets [66] χρησιμοποιήθηκε σε συνδυασμό με έναν Decision Tree ταξινομητή. Για τον λόγο αυτό, η περιγραφή που θα ακολουθήσει βασίζεται στην υπόθεση ότι το μοντέλο που χρησιμοποιείται είναι ένα δένδρο απόφασης.

Δεδομένης μιας χρονοσειράς $X = \{X_1, X_2, ..., X_N\}$ μήκους N, ως υποακολουθία S μήκους l της χρονοσειράς X ορίζεται ένα υποσύνολο συνεχόμενων στοιχείων της χρονοσειράς X, του οποίου το πλήθος είναι ίσο με l.

$$S = \{X_p, X_{p+1}, ..., X_{p+l-1}\},\ 1 \leq p \leq N - l + 1 \tag{4.76}$$

Υποθέτοντας ότι l είναι κάποιο σταθερό μήκος που έχει οριστεί από έναν χρήστη, είναι δυνατόν να βρεθούν όλες οι υποακολουθίες της χρονοσειράς X που έχουν μήκος l. Το σύνολο αυτών των υποακολουθιών θα συμβολίζεται ως εξής:

$$S_X^l = \{S_p^l,\ \forall p{:}1 \leq p \leq N - l + 1\} \tag{4.77}$$

όπου ο δείκτης p συμβολίζει τη θέση του στοιχείου από το οποίο ξεκινάει η υποακολουθία και ο δείκτης l ορίζει το μήκος της υποακολουθίας.

Συνεχίζοντας, δεδομένου κάποιου συνόλου εκπαίδευσης D, οπού τα δεδομένα είναι χρονοσειρές και C είναι το πλήθος των πιθανών κατηγοριών στις οποίες μπορεί να ανήκει μια χρονοσειρά, ως shapelet ορίζεται μια υποακολουθία μιας χρονοσειράς του συνόλου S η οποία θεωρείται ότι είναι αντιπροσωπευτική μιας συγκεκριμένης κατηγορίας.

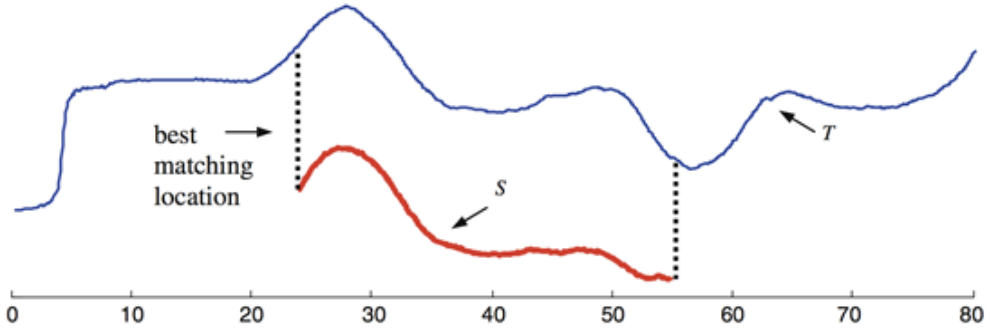


Fig. Illustration of the best matching location in time series T for subsequence S

Πηγή: https://static.wixstatic.com/media/6a6f1b_4dde3c226afc42389c5aaf0bd3ae6b4b~mv2.png

Σχήμα 4.21: Shapelet μιας χρονοσειράς

Η βασική ιδέα που κρύβεται από πίσω είναι ότι για διαφορετικές χρονοσειρές που ανήκουν στην ίδια κατηγορία θα υπάρχει μια υποακολουθία της οποίας η μορφή θα είναι παρόμοια για όλες τους, ενώ παράλληλα θα τις διαφοροποιεί από χρονοσειρές που ανήκουν σε άλλες κατηγορίες. Για την μέτρηση της ομοιότητας ενός shapelet S με μια χρονοσειρά X χρησιμοποιείται μια συνάρτηση η οποία υπολογίζει την απόσταση μεταξύ τους:

$$subsequenceDist(S, X) = min\{dist(S, S'), \forall S' \in S_X^{|S|}\} \tag{4.78}$$

όπου $dist(\cdot)$ μια συνάρτηση η οποία υπολογίζει την απόσταση δύο χρονοσειρών που έχουν το ίδιο μήκος (για παράδειγμα μπορεί να χρησιμοποιηθεί η ευκλείδεια απόσταση).

Διαισθητικά, η απόσταση ενός shapelet από μια χρονοσειρά είναι η απόσταση του shapelet από την υποακολουθία της χρονοσειράς στην οποία "εφαρμόζει" καλύτερα (best matching location).

Συνεχίζοντας την ανάλυση του αλγορίθμου, όπως έχει ήδη ειπωθεί και παραπάνω, γίνεται η υπόθεση ότι για το πρόβλημα κατηγοριοποίησης χρησιμοποιείται ένα δένδρο απόφασης. Για λόγους απλότητας της παρουσίασης γίνεται ακόμη η παραδοχή ότι το σύνολο των πιθανών κατηγοριών C είναι ίσο με 2 και ότι τα δεδομένα του συνόλου εκπαίδευσης είναι μονοδιάστατες χρονοσειρές (αυτό δεν είναι δεσμευτικό καθώς η μέθοδος αυτή μπορεί να επεκταθεί και για περισσότερες κατηγορίες και για πολυδιάστατες χρονοσειρές [67]). Αυτό σημαίνει ότι το δένδρο απόφασης που θα προκύψει θα αποτελείται από τη ρίζα και δύο κόμβους παιδιά. Είναι γνωστό από το υποκεφάλαιο 4.1.3.3 ότι το δένδρο απόφασης ξεκινάει από το σύνολο δεδομένων εκπαίδευσης D και προσπαθεί σε κάθε βήμα να βρει το βέλτιστο split ενός συνόλου βάσει κάποιων κριτηρίων. Στην περίπτωση της Time Series Shapelets μεθόδου η λογική είναι η ίδια. Ως κριτήριο αξιολόγησης κάποιου split χρησιμοποιείται το κέρδος πληροφορίας. Ένα split έγκειται στην εύρεση του κατάλληλου shapelet από το εκάστοτε σύνολο και τον προσδιορισμό ενός ορίου απόφασης (decision threshold) με βάση το οποίο τα δείγματα θα ταξινομούνται σε ένα από τα δύο υποσύνολα που προκύπτουν (αφού έγινε η υπόθεση ότι το πλήθος των πιθανών κατηγοριών είναι ίσο με δύο). Συμβολίζοντας με $D1$ και $D2$ τα υποσύνολα που προκύπτουν από το σύνολο δεδομένων εκπαίδευσης D το ζητούμενο είναι να προσδιοριστούν ένα shapelet S και ένα όριο απόφασης d_{th} έτσι ώστε μετά το split να επιτυγχάνεται το βέλτιστο κέρδος

πληροφορίας και να ισχύουν οι σχέσεις:

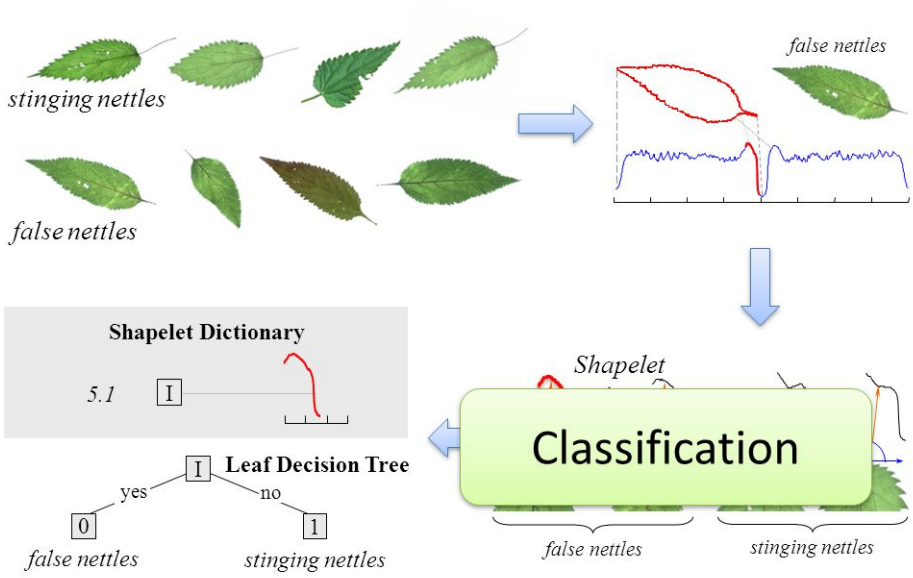
$$subsequenceDist(S, X_{1,i}) \leq d_{th} \; \forall X_{1,i} \in D1 \tag{4.79}$$

$$subsequenceDist(S, X_{2,i}) \geq d_{th} \; \forall X_{2,i} \in D2 \tag{4.80}$$

$$Gain(S, d_{th}) \geq Gain(S, d'_{th}) \tag{4.81}$$

$$Gain(S, d_{th}) \geq Gain(S', d'_{th}) \tag{4.82}$$

όπου $Gain(\cdot)$ μια συνάρτηση που εκφράζει το κέρδος πληροφορίας που επιτυγχάνεται για το εκάστοτε shapelet και decision threshold. Οι δύο πρώτες σχέσεις εκφράζουν ότι όλες οι χρονοσειρές του συνόλου εκπαίδευσης D των οποίων η απόσταση από το shapelet S είναι μικρότερη του ορίου απόφασης τοποθετούνται στο υποσύνολο $D1$, ενώ ολές χρονοσειρές του συνόλου εκπαίδευσης D των οποίων η απόσταση από το shapelet S είναι μεγαλύτερη του ορίου απόφασης τοποθετούνται στο υποσύνολο $D2$. Οι τελευταίες δύο σχέσεις εκφράζουν το γεγονός ότι έχουν επιλεγεί τα βέλτιστα shapelet και decision threshold, καθώς οποιοσδήποτε άλλος συνδυασμός shapelet και decision threshold οδηγεί σε χαμηλότερο κέρδος πληροφορίας. Για να γίνει καλύτερα κατανοητό αυτό παρουσιάζεται ένα παράδειγμα [66].



Πηγή: https://slideplayer.com/slide/1506862/5/images/20/Classification+stinging+nettles+false+nettles+Shapelet.jpg

Σχήμα 4.22: Παράδειγμα εφαρμογής της Time Series Shapelets μεθόδου

Ζητούμενο του παραδείγματος που παρουσιάζεται στο σχήμα 4.22 είναι η ταξινόμηση κάποιων φύλλων σε μια από δύο πιθανές κατηγορίες. Αρχικά γίνεται μια επεξεργασία στα φύλλα έτσι ώστε να μπορούν να εκφραστούν σε μορφή ακολουθίας η οποία προκύπτει από το σχήμα τους. Στη συνέχεια, τα δείγματα που έχουν συλλεχθεί διαχωρίζονται σε δύο υποσύνολα βάσει του shapelet που φαίνεται στο σχήμα 4.22 και με όριο απόφασης ίσο με 5.1.

Συμβολίζοντας με N_i το μήκος μιας χρονοσειράς X_i που ανήκει στο σύνολο δεδομένων εκπαίδευσης D ο αριθμός των υποψήφιων shapelets μήκους l είναι ίσος με:

$$\sum_{X_i \in D} (M_i - l + 1) \tag{4.83}$$

Συνεπώς, ο αριθμός των υποψήφιων shapelets των οποίων το μήκος κυμαίνεται στο διάστημα $[minLen, maxLen]$ είναι ίσος με:

$$\sum_{minLen}^{maxLen} \sum_{X_i \in D} (M_i - l + 1) \tag{4.84}$$

όπου $minLen$ και $maxLen$ παράμετροι που ορίζονται από το χρήστη. Θα πρέπει να σημειωθεί ότι η μέγιστη τιμή που μπορεί να λάβει η παράμετρος $maxLen$ ισούται με το μήκος της μικρότερης χρονοσειράς του συνόλου εκπαίδευσης.

Είναι προφανές ότι η επιλογή του βέλτιστου shapelet ύστερα από εξαντλητική αναζήτηση είναι μια υπολογιστικά ακριβής διαδικασία. Για τον λόγο αυτό έχουν προσδιοριστεί αλγόριθμοι οι οποίοι συνδιάζουν τόσο την ταχύτητα εκτέλεσης όσο και την αποδοτική επίλυση του προβλήματος εύρεσης του βέλτιστου shapelet [66].

4.2.3 Φίλτρο Καλμαν (Kalman Filter)

Πρόκειται για έναν αλγόριθμο ο οποίος χρησιμοποιείται για την εξομάλυνση δεδομένων και την πρόβλεψη μελλοντικών τιμών μιας χρονοσειράς. Αποτελεί μια ευρέως διαδεδομένη μέθοδο στο τομέα των ευφυών συστημάτων μεταφορών καθώς χρησιμοποιείται σε εφαρμογές που αφορούν χωρικά δεδομένα όπως είναι ο προσδιορισμός της θέσης του οχήματος [68], ενώ παράλληλα εξασφαλίζει και την αφαίρεση του θορύβου από τις μετρήσεις που καταγράφουν οι αισθητήρες (accelometer, gyrometer) [69]. Στη συνέχεια, παρουσιάζεται η βασική λειτουργία του φίλτρου Kalman [70] μέσω ενός παραδείγματος που αφορά ένα έξυπνο όχημα.

Έστω ότι τη χρονική στιγμή k η κατάσταση του οχήματος X ορίζεται από την ταχύτητα και τη θέση του (κάτι τέτοιο δεν θα ίσχυε στην πραγματικότητα καθώς είναι πολύ περισσότερες οι μεταβλητές που περιγράφουν την κατάσταση του οχήματος αλλά γίνεται αυτή η απλούστευση για την εξυπηρέτηση της παρουσίασης) ως εξής:

$$\vec{x_k} = (\vec{p}, \vec{v}) \tag{4.85}$$

όπου τα διανύσματα $\vec{p}$ και $\vec{v}$ καθορίζουν τη θέση και τη ταχύτητα του οχήματος αντίστοιχα.

Έστω τώρα ότι το έξυπνο όχημα είναι εξοπλισμένο με έναν αισθητήρα GPS. Οι μετρήσεις που παρέχονται από τον αισθητήρα σίγουρα θα περιέχουν σφάλματα και ανακρίβειες. Με τη εφαρμογή ενός φίλτρου Kalman είναι δυνατή μια καλύτερη εκτίμηση για την κατάσταση του οχήματος τη χρονική στιγμή k χρησιμοποιώντας την κατάσταση της προηγούμενης χρονικής στιγμής X_{k-1}. Αρχικά, το φίλτρο Kalman κάνει την υπόθεση ότι τα $\vec{p}$ και $\vec{v}$ είναι τυχαίες μεταβλητές οι οποίες ακολουθούν τη Γκαουσιανή κατανομή [71]. Αυτό σημαίνει ότι κάθε μεταβλητή έχει μια μέση τιμή μ, η

οποία εκφράζει την πιθανότερη κατάστασή της, και μια διακύμανση σ^2, η οποία εκφράζει ένα βαθμό αβεβαιότητας της πρόβλεψης. Συνεχίζοντας, το φίλτρο Kalman λαμβάνει υπόψιν και την αλληλεξάρτηση που μπορεί να υπάρχει μεταξύ των μεταβλητών. Στην επιστήμη της Στατιστικής, το μέτρο του βαθμού συσχετίσεως δύο τυχαίων μεταβλητών X και Y εκφράζεται από ένα μέγεθος το οποίο ονομάζεται συνδιακύμανση και ορίζεται ως:

$$cov(X,Y) = E[(X - E[X]) \cdot (Y - E[Y])] = \sigma_{xy} \tag{4.86}$$

όπου με $E[\cdot]$ συμβολίζεται η μέση τιμή μιας τυχαίας μεταβλητής. Για ένα σύνολο περισσότερων τυχαίων μεταβλητών μπορεί να οριστεί ένας συμμετρικός πίνακας συνδιακύμανσης, όπου σε κάθε κελί περιέχει τη συνδιακύμανση των μεταβλητών που ορίζονται από την αντίστοιχη γραμμή και στήλη του πίνακα. Χάριν απλότητας της παρουσίασης δεν γίνεται ανάλυση των συνιστωσών της ταχύτητας και της θέση του οχήματος στους τρεις άξονες x, y και z. Συνεπώς, οι εξισώσεις που ορίζουν τη μεταβολή της θέσης και της ταχύτητας του οχήματος είναι αντίστοιχα (στην πραγματικότητα είναι):

$$p_k = p_{k-1} + \Delta t \cdot v_{k-1},\ v_k = v_{k-1} \tag{4.87}$$

Για τη μετάβαση από την κατάσταση $k-1$ στην κατάσταση k μπορεί να γίνει μια πρόβλεψη βάσει των εξισώσεων:

$$\hat{x}_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \cdot \hat{x}_{k-1} = F_k \cdot \hat{x}_{k-1},\ P_k = F_k \cdot P_{k-1} \cdot F_k^T \tag{4.88}$$

με $\hat{x}_k \begin{pmatrix} p_k \\ v_k \end{pmatrix}$, $\hat{x}_{k-1} \begin{pmatrix} p_{k-1} \\ v_{k-1} \end{pmatrix}$, $F_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$, $P_k = \begin{pmatrix} \sigma_{pp}^{(k)} & \sigma_{pv}^{(k)} \\ \sigma_{vp}^{(k)} & \sigma_{vv}^{(k)} \end{pmatrix}$, $P_{k-1} = \begin{pmatrix} \sigma_{pp}^{(k-1)} & \sigma_{pv}^{(k-1)} \\ \sigma_{vp}^{(k-1)} & \sigma_{vv}^{(k-1)} \end{pmatrix}$ όπου ο πίνακας F_k καλείται πίνακας πρόβλεψης (prediction matrix), x_k η εκτίμηση για τη χρονική στιγμή k, x_{k-1} η εκτίμηση για τη χρονική στιγμή $k-1$, P_k ο πίνακας συνδιακυμάνσεων τη χρονική στιγμή k και P_{k-1} ο πίνακας συνδιακυμάνσεων τη χρονική στιγμή $k-1$.

Ωστόσο, μια τέτοια προσέγγιση δεν αρκεί για να περιγράψει την δυναμική συμπεριφορά του οχήματος. Εξωτερικοί παράγοντες, όπως είναι η επιτάχυνση, μπορεί να επηρεάζουν τη κατάσταση του. Υποθέτοντας ότι οι εξισώσεις της κινηματικής κατάστασης του οχήματος είναι οι εξής:

$$p_k = p_{k-1} + \Delta t \cdot v_{k-1} + \frac{1}{2} \cdot \alpha \cdot \Delta t^2,\ v_k = v_{k-1} + \alpha \cdot \Delta t \tag{4.89}$$

όπου με α συμβολίζεται η επιτάχυνση, για τη μετάβαση από την κατάσταση $k-1$ στην κατάσταση k μπορεί να γίνει μια πρόβλεψη βάσει της εξίσωσης:

$$\hat{x}_k = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \cdot \hat{x}_{k-1} + \begin{pmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{pmatrix} \cdot \alpha = F_k \cdot \hat{x}_{k-1} + B_k \cdot \vec{u} \tag{4.90}$$

όπου ο πίνακας B_k καλείται πίνακας ελέγχου (control matrix) και το διάνυσμα u καλείται διάνυσμα ελέγχου (control vector).

Συνεχίζοντας, στη εξίσωση με τους πίνακες συνδιακύμανσης προστίθεται ένας επιπλέον πίνακας συνδιακύμανσης Q_k, ο οποίος ορίζεται από το χρήστη, προκειμένου να ενισχυθεί κατά ένα βαθμό η αβεβαιότητα της πρόβλεψης.

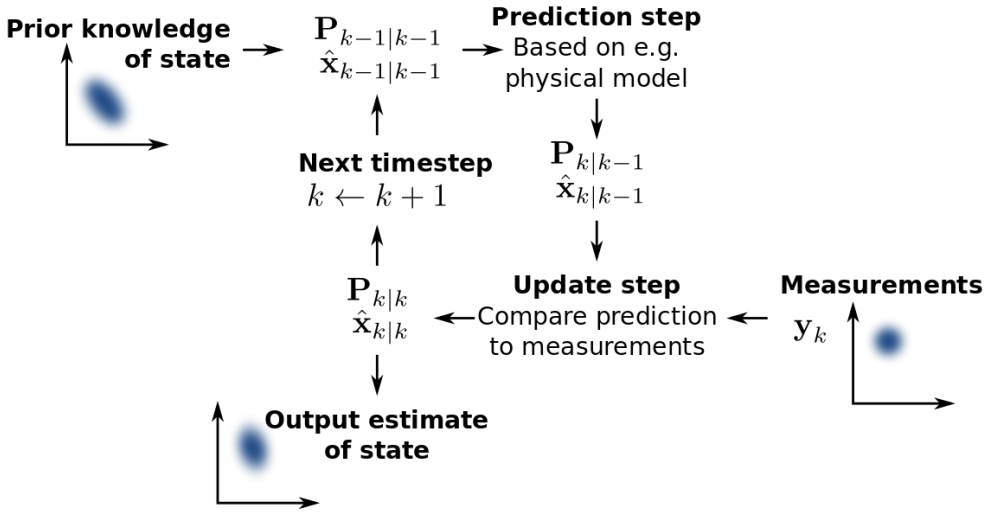
$$\hat{x}_k = F_k \cdot \hat{x}_{k-1} + B_k \cdot \vec{u},\ P_k = F_k \cdot P_{k-1} \cdot F_k^T + Q_k \tag{4.91}$$

Με απλά λόγια, μέχρι στιγμής, η εκτίμηση για την κατάσταση τη χρονική στιγμή k υπολογίζεται από την εκτίμηση της προηγούμενης χρονικής κατάστασης λαμβάνοντας υπόψιν και εξωτερικούς παράγοντες που επηρεάζουν το σύστημα. Ακόμη, ο βαθμός αβεβαιότητας της εκτίμησης προσδιορίζεται από το βαθμό αβεβαιότητας της προηγούμενης εκτίμησης συν έναν επιπλέον όρο αβεβαιότητας ο οποίος εκφράζει την αβεβαιότητα που εισάγεται από το περιβάλλον (πχ. λόγω της επιρροής αέρα κατά τη κίνηση του οχήματος). Ότι έχει περιγραφεί μέχρι τώρα αφορά τη φάση πρόβλεψης (prediction phase) του φίλτρου Kalman. Τώρα θα περιγραφεί η φάση ενημέρωσης (update phase).

Έστω πίνακας H_k ο οποίος περιέχει τις μετρήσεις των αισθητήρων του οχήματος τη χρονική στιγμή k. Έστω z_k το διάνυσμα που εκφράζει τη παρατηρούμενη (τη πραγματική δηλαδή) κατάσταση του οχήματος και R_k ένας πίνακας συνδιακυμάνσεων που χρησιμοποιείται για να εκφράσει το θόρυβο που έχει εισαχθεί στις μετρήσεις των αισθητήρων. Η τελική έξοδος του φίλτρου Kalman, που αποτελεί και τη βέλτιστη εκτίμηση της κατάστασης του οχήματος, δίνεται από τη σύγκριση της παρατηρούμενης κατάστασης του οχήματος με αυτή που προέβλεψε το φίλτρο Kalman στη φάση πρόβλεψης. Έπειτα από εκτέλεση κάποιων μαθηματικών πράξεων, η έξοδος του φίλτρου Kalman δίνεται από τις εξής εξισώσεις:

$$\hat{x'}_k = \hat{x}_k + K \cdot (\vec{z_k} - H_k \cdot \hat{x}_k),\ \hat{P'}_k = \hat{P}_k + K \cdot H_k \cdot P_k) \tag{4.92}$$

με $K = P_k \cdot H_k^T \cdot (H_k \cdot P_k \cdot H_k^T + R_k)^{-1}$, όπου το K ονομάζεται κέρδος Kalman (Kalman Gain). Η έξοδος x'_k χρησιμοποείται ως είσοδος για την επόμενη εκτίμηση του φίλτρου.



Πηγή: https://upload.wikimedia.org/wikipedia/commons/thumb/a/a5/Basic_concept_of_Kalman_filtering/Kalman_filtering.svg.png

Σχήμα 4.23: Το φίλτρο Kalman

Σε αυτό το σημείο θα πρέπει να αναφερθεί ότι το φίλτρο Kalman εφαρμόζεται μόνο σε γραμμικά δυναμικά συστήματα. Ωστόσο, μπορεί να επεκταθεί η λειτουργία του και

να εφαρμοστεί και σε μη γραμμικά συστήματα (Extended Kalman Filter - EKF) [70].

Κεφάλαιο 5: Παρουσίαση Συνόλου Δεδομένων

5.1 Περιγραφή του Συνόλου Δεδομένων

Το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία είναι το "Traffic, Driving Style and Road Surface Condition" [72] και μπορεί να βρεθεί στη διαδικτυακή πλατφόρμα Kaggle [73]. Πρόκειται για δεδομένα που έχουν συλλεχθεί είτε μέσω της διαγνωστικής θύρας On-Board Diagnostics του οχήματος είτε μέσω του κινητού τηλεφώνου του οδηγού.

Το σύνολο δεδομένων αποτελείται από τέσσερα αρχεία csv, κάθε ένα από τα οποία αναπαριστά μια ξεχωριστή διαδρομή που πραγματοποιήθηκε. Οι τέσσερις αυτές διαδρομές εκτελέστηκαν από δύο διαφορετικά οχήματα, ένα Peugeot 207 1.4 HDi και ένα Opel Corsa 1.3 HDi, κάθε ένα από τα οποία πραγματοποίησε δύο διαδρομές. Συνεχίζοντας, τα δεδομένα μιας διαδρομής συλλέγονται με συχνότητα 1 Hertz, σχηματίζοντας έτσι ένα συνεχόμενο σύνολο εγγραφών που προσδιορίζουν με μεγάλη ακρίβεια την κατάσταση και την κίνηση του οχήματος σε όλο το μήκος της διαδρομής που καταγράφηκε. Κάθε μια από αυτές τις εγγραφές, εκτός από τα οδηγικά δεδομένα που έχουν συλλεχθεί, περιέχει, επιπλέον, και τρία πεδία που χαρακτηρίζουν την κατάσταση του δρόμου που βρίσκεται το όχημα, τον βαθμό κυκλοφοριακής συμφόρησης καθώς και τη συμπεριφορά του οδηγού, την ανάλογη χρονική στιγμή. Τα πεδία αυτά χρησιμοποιούνται για την κατηγοριοποίηση μιας εγγραφής, ως προς το εκάστοτε πεδίο, και δεν αποτελούν δεδομένα που συλλέγονται μέσω κάποιου αισθητήρα.

Συνεπώς, το σύνολο δεδομένων "Traffic, Driving Style and Road Surface Condition" ορίζει ένα πρόβλημα ταξινόμησης στο οποίο τα τρία πεδία που αναφέρθηκαν παραπάνω θα πρέπει να κατηγοριοποιηθούν σε μια συγκεκριμένη ομάδα. Ωστόσο, το αντικείμενο της παρούσης διπλωματικής εργασίας εστιάζει στη συλλογή οδηγικών δεδομένων με σκοπό την ανάλυση της οδηγικής συμπεριφοράς και τον σχηματισμό οδηγικού προφίλ. Επομένως, στο πλαίσιο υλοποίησης αυτής της διπλωματικής εργασίας, οι τιμές των πεδίων που αφορούν την κατάσταση του δρόμου και τον βαθμό κυκλοφοριακής συμφόρησης δεν χρησιμοποιούνται ως ετικέτες ταξινόμησης αλλά ως επιπλέον δεδομένα τα οποία συμβάλλουν στην κατηγοριοποίηση της οδηγικής συμπεριφοράς.

Στη συνέχεια, παρουσιάζονται συνοπτικά όλα τα πεδία που εμπεριέχονται στο σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία. Όπως έχει ήδη αναφερθεί, το σύνολο δεδομένων περιέχει πληροφορίες που σχετίζονται τόσο με την κατάσταση όσο και με την κίνηση του οχήματος. Πιο συγκεκριμένα, τα πεδία που συνιστούν μια εγγραφή του συνόλου δεδομένων είναι τα εξής:

- AltitudeVariation: Το πεδίο αυτό εκφράζει τη μεταβλητότητα του ύψους (ανά δευτερόλεπτο) που βρίσκεται το όχημα. Οι τιμές αυτού του πεδίου είναι null για

τα πρώτα δέκα δευτερόλεπτα.

- VehicleSpeedInstantaneous: Το πεδίο αυτό εκφράζει την ταχύτητα με την οποία κινείται το όχημα την εκάστοτε χρονική στιγμή.

- VehicleSpeedAverage: Το πεδίο αυτό εκφράζει τη μέση ταχύτητα του οχήματος τα τελευταία εξήντα δευτερόλεπτα. Οι τιμές αυτού του πεδίου είναι null για τα πρώτα εξήντα δευτερόλεπτα.

- VehicleSpeedVariance: Το πεδίο αυτό εκφράζει τη διακύμανση της ταχύτητας του οχήματος τα τελευταία εξήντα δευτερόλεπτα. Οι τιμές αυτού του πεδίου είναι null για τα πρώτα εξήντα δευτερόλεπτα.

- VehicleSpeedVariation: Το πεδίο αυτό εκφράζει τη μεταβλητότητα της ταχύτητας (ανά δευτερόλεπτο) του οχήματος.

- LongitudinalAcceleration: Το πεδίο αυτό εκφράζει το διάνυσμα της επιτάχυνσης του οχήματος ως προς τον άξονα της κατεύθυνσής του. Θετικές τιμές αυτού του πεδίου υποδεικνύουν επιτάχυνση του οχήματος, ενώ αρνητικές τιμές υποδεικνύουν επιβράδυνση του οχήματος. Οι τιμές που συλλέχθηκαν έχουν δεχθεί προεπεξεργασία καθώς έχουν περάσει από ένα βαθυπερατό φίλτρο.

- EngineLoad: Το πεδίο αυτό εκφράζει το "φορτίο" του κινητήρα. Με απλά λόγια, θα μπορούσε κανείς να πει ότι το engine load είναι ένα μέγεθος το οποίο εκφράζει την ποσότητα αέρα και καυσίμου που χρησιμοποιεί ο κινητήρας την εκάστοτε χρονική στιγμή προς τη μέγιστη θεωρητική ποσότητα που δύναται να χρησιμοποιήσει. Το κλάσμα αυτό εκφράζεται σαν ποσοστό επί τοις εκατό.

- EngineCoolantTemperature: Το πεδίο αυτό εκφράζει τη θερμοκρασία του συστήματος που είναι υπεύθυνο για την αποφυγή της ψύξης ή της υπερθέρμανσης του κινητήρα. Ουσιαστικά, το μέγεθος αυτό εκφράζει το μέτρο της θερμότητας που παράγεται από τη λειτουργία του κινητήρα. Οι τιμές που συλλέχθηκαν είναι μετρημένες σε βαθμούς Κελσίου.

- ManifoldAbsolutePressure: Το πεδίο αυτό εκφράζει το σήμα που παράγει ο αισθητήρας απόλυτης πίεσης (Manifold Absolute Pressure Sensor) του οχήματος. Η μέτρηση του αισθητήρα αυτού αποτελεί ένα μέτρο έκφρασης του φορτίου του κινητήρα, ενώ παράλληλα χρησιμοποιείται και από το καρμπυρατέρ του οχήματος έτσι ώστε να επιτυγχάνεται, κάθε χρονική στιγμή, η βέλτιστη αναλογία ατμοσφαιρικής πίεσης/καυσίμου στον κινητήρα. Οι τιμές που συλλέχθηκαν είναι μετρημένες σε kilo Pascal.

- EngineRPM: Το πεδίο αυτό εκφράζει τις στροφές του κινητήρα ανά λεπτό. Ουσιαστικά, αποτελεί ένα μέτρο έκφρασης της ταχύτητας λειτουργίας του κινητήρα και συνδέεται άμεσα με το πόσο πιέζεται το πετάλι του γκαζιού.

- MassAirFlow: Το πεδίο αυτό εκφράζει το σήμα που παράγει ο αισθητήρας μάζας αέρα (Mass Air Flow Sensor) του οχήματος. Η μέτρηση του αισθητήρα αυτού εκφράζει τη μάζα του αέρα που εισέρχεται, κάθε χρονική στιγμή, στον κινητήρα του οχήματος. Οι τιμές που συλλέχθηκαν είναι μετρημένες σε γραμμάρια.

- IntakeAirTemperature: Το πεδίο αυτό εκφράζει τη θερμοκρασία του αέρα που εισάγεται στον κινητήρα. Η μέτρηση αυτή χρησιμοποιείται από το καρμπυρατέρ

προκειμένου να επιτυγχάνεται, κάθε χρονική στιγμή, η βέλτιστη αναλογία ατμοσφαιρικής πίεσης/καυσίμου στον κινητήρα. Όσο χαμηλότερη είναι η θερμοκρασία του αέρα, τόσο μεγαλύτερη είναι η πυκνότητά του και συνεπώς απαιτείται μεγαλύτερη ποσότητα καυσίμου προκειμένου να επιτυγχάνεται η βέλτιστη αναλογία. Οι τιμές που συλλέχθηκαν είναι μετρημένες σε βαθμούς Κελσίου.

- VerticalAcceleration: Το πεδίο αυτό εκφράζει το διάνυσμα της επιτάχυνσης του οχήματος ως προς τον άξονα που είναι κάθετος στην κατεύθυνση του οχήματος και το επίπεδο της γης. Οι τιμές που συλλέχθηκαν έχουν δεχθεί προεπεξεργασία καθώς έχουν περάσει από ένα βαθυπερατό φίλτρο.

- FuelConsumptionAverage: Το πεδίο αυτό εκφράζει την ποσότητα καυσίμου που καταναλώνεται. Οι τιμές που συλλέχθηκαν είναι μετρημένες σε λίτρα ανά εκατό χιλιόμετρα.

- roadSurface: Το πεδίο αυτό εκφράζει την κατάσταση του δρόμου στον οποίο βρίσκεται το όχημα. Οι πιθανές τιμές αυτού του πεδίου είναι οι SmoothCondition, FullOfHolesCondition και UnevenCondition, οι οποίες υποδηλώνουν ομαλό δρόμο, δρόμο γεμάτο με τρύπες και ανώμαλο δρόμο, αντίστοιχα.

- traffic: Το πεδίο αυτό εκφράζει το βαθμό κυκλοφοριακής συμφόρησης την εκάστοτε χρονική στιγμή. Οι πιθανές τιμές αυτού του πεδίου είναι οι LowCongestionCondition, NormalCongestionCondition και HighCongestionCondition, οι οποίες υποδηλώνουν χαμηλή, κανονική και υψηλή κυκλοφοριακή συμφόρηση, αντίστοιχα.

- drivingStyle: Το πεδίο αυτό εκφράζει τον τρόπο τρόπο με τον οποίο κινείται το όχημα την εκάστοτε χρονική στιγμή. Πιο συγκεκριμένα, οι πιθανές τιμές αυτού του πεδίου είναι οι EvenPaceStyle και AggressiveStyle, οι οποίες υποδηλώνουν κανονικό και επιθετικό στυλ οδήγησης, αντίστοιχα. Ουσιαστικά, βάσει της τιμής αυτού του πεδίου πραγματοποιείται η κατηγοριοποίηση της συμπεριφοράς του οδηγού.

Κλείνοντας, στον πίνακα 5.1 παρουσιάζονται συγκεντρωτικά τα πεδία του συνόλου δεδομένων καθώς και ο τύπος που αυτά έχουν.

Όνομα Πεδίου	Τύπος
AltitudeVariation	δεκαδικός αριθμός (στατιστικό μέγεθος)
VehicleSpeedInstantaneous	km/h (kilo meters per hour)
VehicleSpeedAverage	km/h (kilo meters per hour)
VehicleSpeedVariance	δεκαδικός αριθμός (στατιστικό μέγεθος)
VehicleSpeedVariation	δεκαδικός αριθμός (στατιστικό μέγεθος)
LongitudinalAcceleration	m/s^2 (metre per second squared)
EngineLoad	ποσοστό επί τοις εκατό %
EngineCoolantTemperature	^oC (Celsius degrees)
ManifoldAbsolutePressure	kPa (kilo Pascal)
EngineRPM	rev/min (revolutions per minute)
MassAirFlow	g (grams)
IntakeAirTemperature	^oC (Celsius degrees)

VerticalAcceleration	m/s^2 (metre per second squared)
FuelConsumptionAverage	l/100 km (litres per 100 kilo meters)
roadSurface	{SmoothCondition, FullOfHolesCondition, UnevenCondition}
traffic	{LowCongestionCondition, NormalCongestionCondition, HighCongestionCondition}
drivingStyle	{EvenPaceStyle, AggressiveStyle}

Πίνακας 5.1: Τα πεδία του συνόλου δεδομένων

5.2 Εφαρμογή Κυλιόμενου Παραθύρου (Sliding Window) στο Σύνολο Δεδομένων

Τα δεδομένα που αφορούν μια συγκεκριμένη διαδρομή μπορούν να εκφραστούν ως συνεχόμενες εγγραφές των πεδίων που αναφέρθηκαν στο υποκεφάλαιο 5.1 και οι οποίες συλλέγονται κάθε δευτερόλεπτο. Είναι προφανές ότι η ετικέτα (η τιμή του πεδίου drivingStyle) που δίνεται στον οδηγό του οχήματος κάποια τυχαία χρονική στιγμή, εξαρτάται άμεσα από τις πράξεις που αυτός πραγματοποίησε σε προηγούμενες χρονικές στιγμές. Αυτό σημαίνει ότι οι τιμές των πεδίων κάποιας τυχαίας εγγραφής της διαδρομής που καταγράφεται αποτελούν αποτέλεσμα και λογική συνέχεια των τιμών των πεδίων των προηγούμενων εγγραφών. Συνεπώς, μια διαδρομή μπορεί να θεωρηθεί ως μια πολυδιάστατη χρονοσειρά (όπως αυτή έχει οριστεί στο υποκεφάλαιο 4.2.1), τα δεδομένα της οποίας συλλέγονται με συχνότητα 1 Hertz. Για κάθε εγγραφή της χρονοσειράς αυτής, μπορούν να συμπεριληφθούν πληροφορίες από εγγραφές προηγούμενων χρονικών στιγμών με την εφαρμογή ενός κυλιόμενου παραθύρου.

Η τεχνική του κυλιόμενου παραθύρου εφαρμόζεται σε περιπτώσεις όπου τα δεδομένα ορίζουν μια χρονοσειρά και χρησιμοποιείται για την ομαδοποίηση των δεδομένων και τον προσδιορισμό χρονικών διαστημάτων που αποτελούν μικρότερα τμήματα της χρονοσειράς που μελετάται. Ουσιαστικά, αυτό που επιτυγχάνεται είναι ο διαχωρισμός ενός μεγάλου συνόλου δεδομένων, δηλαδή ολόκληρη η χρονοσειρά, σε επιμέρους μικρότερα κομμάτια, γεγονός που ευνοεί την αποτελεσματική ανάλυση και αξιοποίηση των δεδομένων.

Συνεχίζοντας, δύο είναι οι βασικές παράμετροι που ορίζονται για ένα κυλιόμενο παράθυρο. Η πρώτη από αυτές είναι το μέγεθος του κυλιόμενου παραθύρου (sliding window size), μέσω του οποίου ορίζονται τα χρονικά όρια του μικρότερου τμήματος δεδομένων (παράθυρο) που επιλέγεται και επομένως το σύνολο των εγγραφών που αυτό θα περιλαμβάνει. Η δεύτερη από αυτές είναι το βήμα του κυλιόμενου παραθύρου (sliding window step), μέσω του οποίου ορίζεται ο τρόπος με τον οποίο το κυλιόμενο παράθυρο θα κινείται "πάνω" στα δεδομένα.

Στη συνέχεια, παρουσιάζεται ένα παράδειγμα προκειμένου να γίνει καλύτερα κατανοητή η τεχνική του κυλιόμενου παραθύρου. Έστω μια διαδρομή η οποία καταγράφηκε και περιέχει δέκα έγγραφες. Η διαδρομή αυτή μπορεί να αναπαρασταθεί

με μαθηματικό τρόπο ως εξής:

$$\{(t_1, \vec{x}_{t_1}), (t_2, \vec{x}_{t_2}), ..., (t_{10}, \vec{x}_{t_{10}})\} = \{(1, \vec{x}_1), (2, \vec{x}_2), ..., (10, \vec{x}_{10})\} \tag{5.1}$$

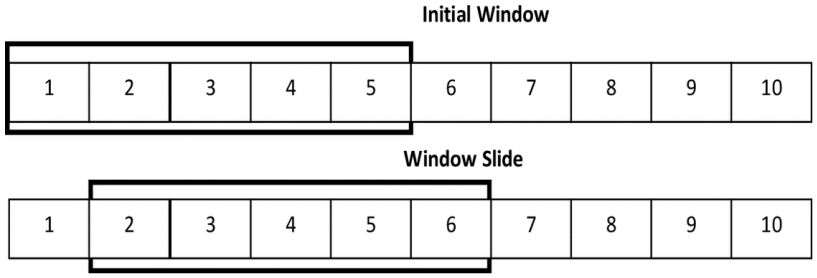
όπου $\vec{x_i} = \{x^{AltitudeVariation}(i), ..., x^{drivingStyle}(i)\}$. Το διάνυσμα x περιέχει όλα τα πεδία που παρουσιάστηκαν στο υποκεφάλαιο 5.1 για κάποια συγκεκριμένη χρονική στιγμή i.

Έστω κυλιόμενο παράθυρο W σταθερού μεγέθους 5 και βήματος 1. Τότε, τη χρονική στιγμή 5 το κυλιόμενο παράθυρο ορίζεται ως εξής:

$$W = \{(1, \vec{x}_1), (2, \vec{x}_2), ..., (5, \vec{x}_5)\} \tag{5.2}$$

Εφόσον το βήμα έχει τιμή 1, τη χρονική στιγμή 6 το κυλιόμενο παράθυρο θα περιέχει τις εξής εγγραφές:

$$W = \{(2, \vec{x}_2), (3, \vec{x}_3), ..., (6, \vec{x}_6)\} \tag{5.3}$$



Πηγή: Hota, H. S. et al. "Time Series Data Prediction Using Sliding Window Based RBF Neural Network." (2017), Figure 2

Σχήμα 5.1: Κυλιόμενο παράθυρο (Sliding Window)

5.3 Προεπεξεργασία του Συνόλου Δεδομένων

5.3.1 Καθαρισμός Δεδομένων

Ένα από τα σημαντικότερα βήματα κατά την προεπεξεργασία ενός συνόλου δεδομένων είναι ο έλεγχος για ύπαρξη απουσιαζουσών τιμών (null values). Όπως φαίνεται και από το σχήμα 5.2, στο σύνολο δεδομένων που χρησιμοποιήθηκε για την παρούσα διπλωματική εργασία εντοπίστηκαν μερικές null τιμές. Πιο συγκεκριμένα, παρατηρήθηκε ότι το συνολικό πλήθος των εγγραφών του συνόλου δεδομένων είναι ίσο με 24957, ενώ τα μόνα πεδία που δεν περιέχουν null τιμές είναι τα LongitudinalAcceleration, VerticalAcceleration, roadSurface, traffic και drivingStyle. Τα πεδία με τις περισσότερες null τιμές είναι τα VehicleSpeedAverage και VehicleSpeedVariance. Αυτό, πιθανότατα, οφείλεται στο γεγονός ότι για τα πεδία αυτά απαιτείται να περάσουν 60 δευτερόλεπτα πριν ξεκινήσει η καταγραφή των τιμών τους. Κάτι τέτοιο ίσως χρειάστηκε να συμβεί και στις περιπτώσεις όπου το όχημα ακινητοποιήθηκε για μεγάλο χρονικό διάστημα ή έσβησε η μηχανή του. Καθώς, λοιπόν, το σύνολο των null τιμών αυτών των δύο πεδίων είναι σχετικά μικρό, σε σχέση με το συνολικό πλήθος εγγραφών, σε συνδυασμό με τη γνώση ότι περίπου το 25% των null τιμών αυτών σχετίζεται με τα πρώτα 60 δευτερόλεπτα ύστερα από την εκκίνηση του οχήματος, αποφασίστηκε να απορριφθούν (drop) όλες οι εγγραφές των οποίων

έστω και ένα πεδίο περιέχει τιμή null. Συνεπώς, προέκυψε ένα σύνολο δεδομένων συνολικού πλήθους εγγραφών ίσο με 23762.

```
RangeIndex: 24957 entries, 0 to 24956
Data columns (total 17 columns):
 #   Column                      Non-Null Count
---  ------                      --------------
 0   AltitudeVariation           24777 non-null
 1   VehicleSpeedInstantaneous   24913 non-null
 2   VehicleSpeedAverage         23775 non-null
 3   VehicleSpeedVariance        23775 non-null
 4   VehicleSpeedVariation       24769 non-null
 5   LongitudinalAcceleration    24957 non-null
 6   EngineLoad                  24952 non-null
 7   EngineCoolantTemperature    24952 non-null
 8   ManifoldAbsolutePressure    24952 non-null
 9   EngineRPM                   24952 non-null
 10  MassAirFlow                 24952 non-null
 11  IntakeAirTemperature        24952 non-null
 12  VerticalAcceleration        24957 non-null
 13  FuelConsumptionAverage      24671 non-null
 14  roadSurface                 24957 non-null
 15  traffic                     24957 non-null
 16  drivingStyle                24957 non-null
```

Σχήμα 5.2: Έλεγχος για null τιμές

5.3.2 Μετασχηματισμός Δεδομένων

Όπως έχει ήδη αναφερθεί και στο υποκεφάλαιο 5.1, τα πεδία roadSurface, traffic και drivingStyle παίρνουν τιμές από ένα σύνολο τιμών string. Καθώς τα δεδομένα αυτά θα χρησιμοποιηθούν για την εκπαίδευση κάποιου μοντέλου μηχανικής μάθησης είναι ιδιαίτερα σημαντικό να κωδικοποιηθούν και να έρθουν σε μια μορφή την οποία το εκάστοτε μοντέλο που θα χρησιμοποιηθεί θα καταλαβαίνει. Συνεπώς, για τα πεδία roadSurface και traffic χρησιμοποιήθηκε η τεχνική Label Encoding, η οποία έχει περιγραφεί στο υποκεφάλαιο 3.2.2.3. Πιο συγκεκριμένα, χρησιμοποιήθηκαν οι εξής κωδικοποιήσεις:

- traffic:

$$LowCongestionCondition \Rightarrow 0,\ NormalCongestionCondition \Rightarrow 1 \text{ και } HighCongestionCondition \Rightarrow 2 \tag{5.4}$$

- roadSurface:

$$SmoothCondition \Rightarrow 0,\ UnevenCondition \Rightarrow 1 \text{ και } FullOfHolesCondition \Rightarrow 2 \tag{5.5}$$

Για το πεδίο drivingStyle χρησιμοποιήθηκε τόσο η τεχνική Label Encoding όσο και η τεχνική One-Hot Encoding, οι οποίες έχουν περιγραφεί στο υποκεφάλαιο 3.2.2.3. Αυτό συνέβη διότι κάποια μοντέλα μηχανικής μάθησης που χρησιμοποιήθηκαν απαιτούν η κωδικοποίηση των ετικετών των δειγμάτων να είναι One-Hot Encoding. Στο κεφάλαιο 6, όπου θα παρουσιαστούν τα αποτελέσματα των μοντέλων μηχανικής μάθησης που εξετάστηκαν, θα επισημαίνεται κάθε φορά ποια τεχνική κωδικοποίησης χρησιμοποιήθηκε. Συνεπώς, για το πεδίο drivingStyle χρησιμοποιήθηκαν οι εξής κωδικοποιήσεις:

- drivingStyle - Label Encoding

$$AggressiveStyle \Rightarrow 0 \text{ και } EvenPaceStyle \Rightarrow 1 \tag{5.6}$$

- drivingStyle – One-Hot Encoding: Σε αυτή την περίπτωση δημιουργήθηκαν δύο νέα πεδία με ονόματα AggressiveStyle και EvenPaceStyle τα οποία συνδυαστικά αντικατέστησαν το πεδίο drivingStyle. Αν κάποιο δείγμα έχει την ετικέτα AggressiveStyle τότε στο πεδίο AggressiveStyle που δημιουργήθηκε θα έχει την τιμή 1, ενώ στο πεδίο EvenPaceStyle θα έχει την τιμή 0. Ομοίως, αν κάποιο δείγμα έχει την ετικέτα EvenPaceStyle τότε στο πεδίο AggressiveStyle θα έχει την τιμή 0, ενώ στο πεδίο EvenPaceStyle θα έχει την τιμή 1.

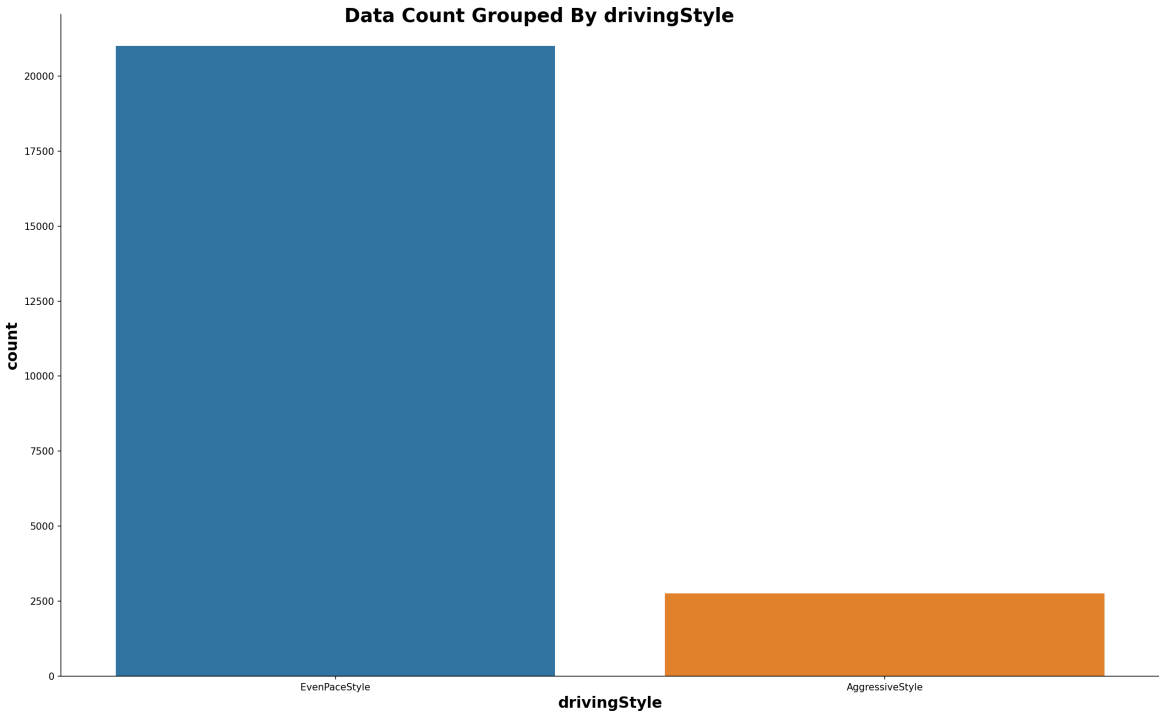
Συνεχίζοντας, πραγματοποιήθηκε η κανονικοποίηση των δεδομένων που απαρτίζουν το σύνολο δεδομένων που χρησιμοποιήθηκε. Όπως και στην περίπτωση της κωδικοποίησης του πεδίου drivingStyle, ανάλογα με το μοντέλο μηχανικής μάθησης που εξετάστηκε, χρησιμοποιήθηκε τόσο η στρατηγική της κανονικής κανονικοποίησης όσο και η στρατηγική της κανονικοποίησης Ελάχιστου-Μέγιστου, οι οποίες έχουν περιγραφεί στο υποκεφάλαιο 3.2.2.3. Στο κεφάλαιο 6, όπου θα παρουσιαστούν τα αποτελέσματα των μοντέλων μηχανικής μάθησης που εξετάστηκαν, θα επισημαίνεται κάθε φορά ποια τεχνική κανονικοποίησης χρησιμοποιήθηκε.

Τέλος, κρίθηκε ότι το πεδίο AltitudeVariation δεν επηρεάζει άμεσα την κατηγοριοποίηση της συμπεριφοράς του οδηγού του οχήματος και για αυτό το λόγο αποφασίστηκε το πεδίο αυτό να αφαιρεθεί από όλες τις εγγραφές του συνόλου δεδομένων. Ακόμη, αποφασίστηκε να αφαιρεθεί και το πεδίο VehicleSpeedInstantaneous καθώς υπερκαλύπτεται από το πεδίο VehicleSpeedAverage, γεγονός το οποίο παρουσιάζεται αναλυτικότερα στο υποκεφάλαιο 5.4.

5.4 Οπτικοποίηση του Συνόλου Δεδομένων (Data Visualization)

Για την οπτικοποίηση του συνόλου δεδομένων χρησιμοποιήθηκε η γλώσσα προγραμματισμού python [74] και πιο συγκεκριμένα οι βιβλιοθήκες Pandas [75] και Seaborn [76]. Στα διαγράμματα που παρουσιάζονται παρακάτω έχει χρησιμοποιηθεί μπλε χρώμα για την αναπαράσταση δεδομένων που αφορούν την ετικέτα EvenPaceStyle και πορτοκαλί χρώμα για την αναπαράσταση δεδομένων που αφορούν την ετικέτα AggressiveStyle.

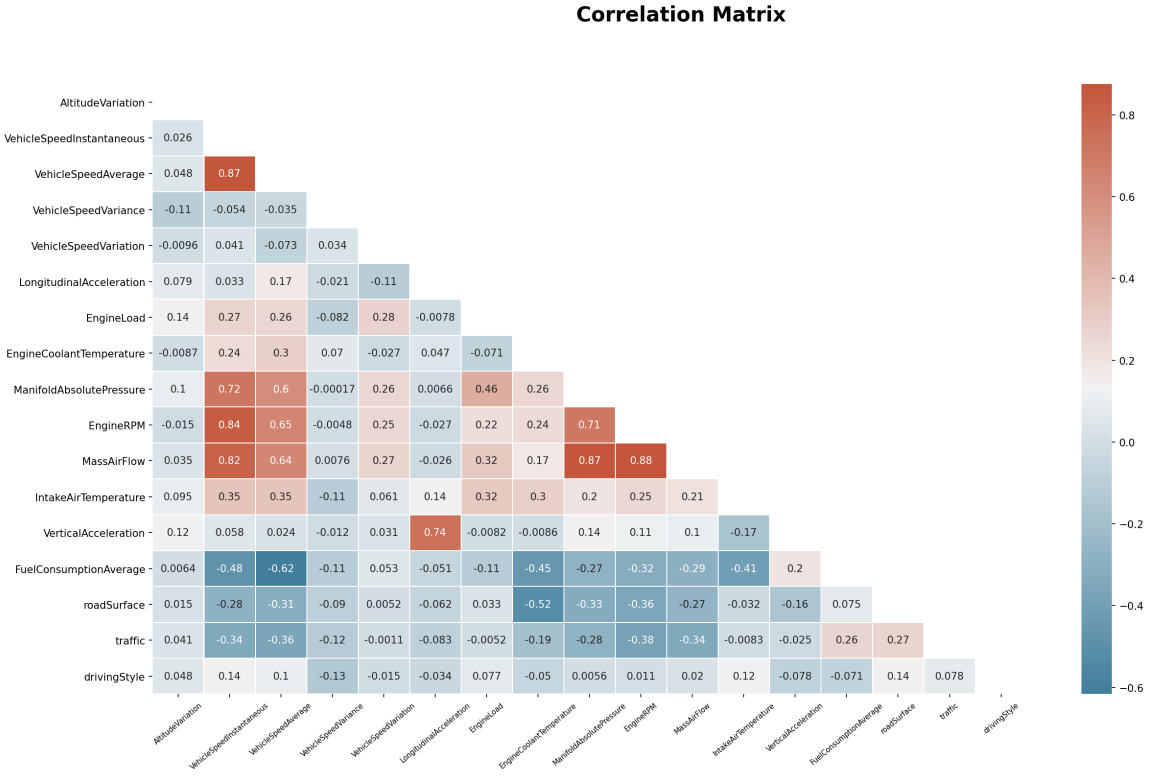
Στο σχήμα 5.3 παρουσιάζεται το πλήθος των εγγραφών που περιέχει το σύνολο δεδομένων ανάλογα με την κατηγορία στην οποία αυτά ανήκουν. Είναι εμφανές ότι τα δεδομένα που έχουν χαρακτηριστεί από την ετικέτα EvenPaceStyle είναι αρκετά περισσότερα.



Σχήμα 5.3: Πλήθος εγγραφών ανά κατηγορία

Συνεχίζοντας, στο σχήμα 5.4 παρουσιάζεται ο πίνακας συσχετίσεων (Correlation Matrix) των πεδίων του συνόλου δεδομένων. Πρόκειται για έναν τριγωνικό πίνακα στα κελιά του οποίου απεικονίζονται οι βαθμοί συσχέτισης των πεδίων που ορίζονται από το αντίστοιχο ζευγάρι γραμμής-στήλης. Ο βαθμός συσχέτισης δύο πεδίων μπορεί να πάρει τιμές από το -1 έως το 1. Όσο πιο κοντά στο 1 είναι ο βαθμός συσχέτισης δύο πεδίων, τόσο περισσότερο θετικά συσχετισμένα είναι αυτά τα πεδία. Όταν δύο πεδία είναι θετικά συσχετισμένα συμπεριφέρονται με παρόμοιο τρόπο. Αυτό σημαίνει ότι η αύξηση της τιμής ενός πεδίου συνεπάγεται αύξηση της τιμής ενός θετικά συσχετισμένου με αυτό πεδίου. Όσο πιο κοντά στο -1 είναι ο βαθμός συσχέτισης δύο πεδίων, τόσο περισσότερο αρνητικά συσχετισμένα είναι αυτά τα πεδία. Όταν δύο πεδία είναι αρνητικά συσχετισμένα συμπεριφέρονται με αντίθετο τρόπο. Αυτό σημαίνει ότι η αύξηση της τιμής ενός πεδίου συνεπάγεται μείωση της τιμής ενός αρνητικά συσχετισμένου με αυτό πεδίου. Αν η τιμή του βαθμού συσχέτισης δύο πεδίων είναι κοντά στο 0, αυτό σημαίνει ότι δεν μπορεί να εξαχθεί άμεσα κάποιο συμπέρασμα για τη σχέση μεταξύ των πεδίων αυτών.

Παρατηρώντας τον πίνακα συσχετίσεων των πεδίων του συνόλου δεδομένων, μπορεί κανείς να δει ότι ο βαθμός συσχέτισης του πεδίου drivingStyle με οποιοδήποτε άλλο πεδίο έχει τιμή που είναι κοντά στο 0. Αυτό σημαίνει ότι δεν υπάρχει, μεμονωμένα, ένα πεδίο βάσει του οποίου θα μπορούσε να γίνει η κατηγοριοποίηση. Παρόμοια συμπεριφορά φαίνεται να έχει και το πεδίο AltitudeVariation, γεγονός που επισημαίνει ότι η αφαίρεση του από το σύνολο δεδομένων ήταν δικαιολογημένη. Ακόμη, παρατηρώντας τους βαθμούς συσχέτισης των πεδίων VehicleSpeedInstantaneous και VehicleSpeedAverage φαίνεται ότι οι τιμές τους είναι αρκετά όμοιες. Αυτό σημαίνει ότι κάποιο από τα δύο πεδία είναι περιττό και ότι πιθανότατα προσδίδουν την ίδια σχεδόν πληροφορία. Καθώς το πεδίο VehicleSpeedInstantaneous έχει περισσότερους βαθμούς συσχέτισης που τείνουν στο 1, επιλέχτηκε να αφαιρεθεί αυτό.



Σχήμα 5.4: Πίνακας συσχετίσεων των πεδίων του συνόλου δεδομένων

Κεφάλαιο 6: Αξιολόγηση Μοντέλων Μηχανικής Μάθησης

6.1 Προσέγγιση του Προβλήματος

Στο παρόν υποκεφάλαιο γίνεται μια συνοπτική παρουσίαση της μεθοδολογίας που ακολουθήθηκε κατά τη φάση εκπαίδευσης των μοντέλων μηχανικής μάθησης που εξετάστηκαν.

6.1.1 Ορισμός του Προβλήματος Κατηγοριοποίησης

Στο κεφάλαιο 5 έγινε μια αναλυτική παρουσίαση του σύνολου δεδομένων που χρησιμοποιήθηκε. Σε αυτό το κεφάλαιο παρουσιάζεται το πρόβλημα κατηγοριοποίησης που κρύβεται πίσω από τα δεδομένα καθώς και ο τρόπος με τον οποίο αυτά μπορούν να αξιοποιηθούν από ένα μοντέλο μηχανικής μάθησης. Το πρόβλημα κατηγοριοποίησης που καλούνται να επιλύσουν τα μοντέλα μηχανικής μάθησης που εξετάστηκαν συνίσταται στην πρόβλεψη της τιμής του πεδίου drivingStyle βάσει των τιμών που έχουν τα υπόλοιπα πεδία. Τελικά, για ολόκληρη τη διαδρομή θα προκύψει ένα σύνολο ετικετών, που έχουν προβλεφθεί από το εκάστοτε μοντέλο, και οι οποίες θα έχουν είτε την τιμή AggressiveStyle είτε την τιμή EvenPaceStyle. Από την ανάλυση των ετικετών αυτών μπορεί να πραγματοποιηθεί κατηγοριοποίηση του οδηγού του οχήματος σε μια συγκεκριμένη κατηγορία, η οποία θα είναι αντιπροσωπευτική των ενεργειών του και της συνολικότερης οδηγικής συμπεριφοράς που αυτός είχε. Συνεπώς, το πρόβλημα κατηγοριοποίησης, όσον αφορά τα μοντέλα μηχανικής μάθησης, συνίσταται στην πρόβλεψη των τιμών του πεδίου drivingStyle και στην περαιτέρω κατηγοριοποίηση τους στην ομάδα AggressiveStyle ή στην ομάδα EvenPaceStyle. Ο τρόπος με τον οποίο πραγματοποιείται το γενικότερο profiling ενός οδηγού παρουσιάζεται στο κεφάλαιο 7, καθώς δεν αφορά άμεσα το πρόβλημα κατηγοριοποίησης που καλούνται να επιλύσουν τα μοντέλα μηχανικής μάθησης.

6.1.2 Εκπαίδευση Μοντέλων

Τα μοντέλα που εξετάστηκαν μπορούν να διακριθούν σε δύο βασικές κατηγορίες, οι οποίες είναι οι εξής:

- Απλά Μοντέλα (Simple Models): Σε αυτήν την κατηγορία ανήκουν τα μοντέλα που εκπαιδεύτηκαν με είσοδο μεμονωμένες εγγραφές του συνόλου δεδομένων.
- Μοντέλα Χρονοσειρών (Time Series Models): Σε αυτήν την κατηγορία ανήκουν τα μοντέλα που εκπαιδεύτηκαν με είσοδο περισσότερες από μία εγγραφές δεδομένων, οι οποίες ορίζουν μια πολυδιάστατη χρονοσειρά. Για το σκοπό αυτό χρησιμοποιήθηκε η τεχνική του κυλιόμενου παραθύρου.

Για την εκπαίδευση των μοντέλων χρησιμοποιήθηκε το 70% του συνόλου δεδομένων ως σύνολο δεδομένων εκπαίδευσης και το 30% ως σύνολο δεδομένων ελέγχου. Τα

δεδομένα έχουν δεχθεί την προεπεξεργασία που περιγράφηκε στο υποκεφάλαιο 5.3, ενώ σε κάποιες περιπτώσεις έχει πραγματοποιηθεί και υποδειγματοληψία του συνόλου δεδομένων καθώς παρατηρήθηκε ότι αυξανόταν η απόδοση των μοντέλων. Στη συνέχεια, που θα παρουσιαστούν αναλυτικότερα τα μοντέλα μηχανικής μάθησης που εξετάστηκαν, θα αναφέρεται για κάθε μοντέλο ποια τεχνική κωδικοποίησης και ποια μέθοδος κανονικοποίησης έχει χρησιμοποιηθεί.

Συνεχίζοντας, θα πρέπει να αναφερθεί ότι για την εκπαίδευση των μοντέλων χρονοσειρών χρησιμοποιήθηκε κυλιόμενο παράθυρο μεγέθους 20 και βήματος 1. Το βήμα του παραθύρου επιλέχτηκε ίσο με 1 έτσι ώστε να συμπεριληφθούν όλες οι εγγραφές στη διαδικασία εκπαίδευσης. Το μέγεθος του παραθύρου επιλέχτηκε ίσο με 20 ύστερα από δοκιμή διάφορων τιμών από τα αποτελέσματα των οποίων προέκυψε ότι το 20 αποτελεί τη βέλτιστη τιμή για το μέγεθος του παραθύρου. Ουσιαστικά, το γεγονός ότι το μέγεθος του παραθύρου είναι ίσο με 20 σημαίνει ότι κάθε εγγραφή, η οποία αποτελεί σημείο ενδιαφέροντος, θα πρέπει να συνοδεύεται από 19 εγγραφές των αμέσως προηγούμενων χρονικών στιγμών. Με αυτόν τον τρόπο σχηματίζεται μια εικοσάδα δεδομένων, την οποία μπορεί κανείς να φανταστεί σαν ένα δισδιάστατο πίνακα που ορίζει μια πολυδιάστατη χρονοσειρά, και η οποία συνοδεύεται από την ετικέτα drivingStyle του τελευταίου δείγματος που εμπεριέχει. Ένα μειονέκτημα που προκύπτει από τον τρόπο με τον οποίον εφαρμόστηκε η τεχνική του κυλιόμενου παραθύρου είναι το γεγονός ότι χάνονται οι πρώτες 19 εγγραφές κάθε διαδρομής, καθώς δεν είναι δυνατόν να σχηματιστεί παράθυρο μεγέθους ίσου με 20 από αυτές. Ένας τρόπος να αντιμετωπιστεί αυτό το φαινόμενο είναι η επιλογή μιας μικρότερης τιμής για το μέγεθος του παραθύρου, έτσι ώστε να χάνονται λιγότερες εγγραφές. Ωστόσο, αποφασίστηκε να διατηρηθεί το μέγεθος του παραθύρου ίσο με 20 καθώς τα αποτελέσματα ήταν αρκετά καλύτερα για αυτό το μέγεθος.

Τέλος, θα πρέπει να αναφερθεί ότι ύστερα από την ολοκλήρωση της διαδικασίας εκπαίδευσης των μοντέλων, πραγματοποιήθηκε επιπλέον δοκιμή του εκπαιδευμένου μοντέλου σε κάθε μια από τις 4 διαδρομές ξεχωριστά, με σκοπό την περαιτέρω αξιολόγηση του μοντέλου. Συνεπώς, στα υποκεφάλαια που ακολουθούν θα παρουσιαστούν τα αποτελέσματα των μοντέλων τόσο στο σύνολο δεδομένων ελέγχου όσο και στην πρώτη από τις διαδρομές που πραγματοποίησε το Opel Corsa 1.3 HDi. Η συγκεκριμένη διαδρομή επιλέχτηκε καθώς περιέχει αρκετά δείγματα και από τις δύο πιθανές ετικέτες που ορίζονται από το πεδίο drivingStyle.

6.1.3 Μετρικές Αξιολόγησης

Σε αυτό το υποκεφάλαιο παρουσιάζονται οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση των μοντέλων μηχανικής μάθησης που εξετάστηκαν. Ουσιαστικά, οι μετρικές αυτές αποτελούν έναν τρόπο έκφρασης του πόσο εύστοχες ή άστοχες ήταν οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου. Η επιλογή των μετρικών αξιολόγησης που παρουσιάζονται παρακάτω έγινε με γνώμονα τη φύση του προβλήματος κατηγοριοποίησης που πρέπει να επιλυθεί. Αφού πρόκειται για ένα πρόβλημα δυαδικής ταξινόμησης ως μετρικές αξιολόγησης επιλέχτηκαν η ακρίβεια (accuracy) και η εκτίμηση περιοχής κάτω από την καμπύλη λειτουργικών χαρακτηριστικών (AUCROC curve).

6.1.3.1 Ακρίβεια (Accuracy)

Προκειμένου να γίνει πλήρως κατανοητή η έννοια της ακρίβειας [77] θα πρέπει πρώτα να αναλυθούν κάποιες βασικές έννοιες που αφορούν την ερμηνεία των αποτελεσμάτων ενός δυαδικού προβλήματος ταξινόμησης [78]. Σε ένα δυαδικό πρόβλημα ταξινόμησης μια από τις δύο κλάσεις μπορεί να χαρακτηριστεί ως η θετική κλάση (positive class) και η άλλη ως η αρνητική κλάση (negative class). Για παράδειγμα, για το σύνολο δεδομένων που παρουσιάστηκε στο κεφάλαιο 5 η κλάση AggressiveStyle μπορεί να οριστεί ως η θετική κλάση, καθώς υποδηλώνει την ύπαρξη επιθετικής συμπεριφοράς, ενώ η κλάση EvenPaceStyle μπορεί να οριστεί ως η αρνητική κλάση, καθώς υποδηλώνει την απουσία επιθετικής συμπεριφοράς. Με βάση την έννοια της αρνητικής και της θετικής κλάσης, σε ένα πρόβλημα δυαδικής ταξινόμησης ορίζονται οι εξής έννοιες:

- True Positive (TP): Ο όρος αυτός αναφέρεται σε ένα δείγμα που φέρει θετική ετικέτα και το μοντέλο το ταξινόμησε σωστά στη θετική κλάση.
- True Negative (TN): Ο όρος αυτός αναφέρεται σε ένα δείγμα που φέρει αρνητική ετικέτα και το μοντέλο το ταξινόμησε σωστά στην αρνητική κλάση.
- False Positive (FP): Ο όρος αυτός αναφέρεται σε ένα δείγμα που φέρει αρνητική ετικέτα και το μοντέλο το ταξινόμησε εσφαλμένα στη θετική κλάση.
- False Negative (FN): Ο όρος αυτός αναφέρεται σε ένα δείγμα που φέρει θετική ετικέτα και το μοντέλο το ταξινόμησε εσφαλμένα στην αρνητική κλάση.

Συνεχίζοντας, ορίζεται ένας πίνακας, ο οποίος ονομάζεται confusion matrix, και χρησιμοποιείται για την απεικόνιση των τιμών TP, TN, FP και FN. Αυτός ο πίνακας έχει την εξής μορφή:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	TP	FP
Predicted Class Negative	FN	TN

Πίνακας 6.1: Confusion matrix

Η κύρια διαγώνιος του παραπάνω πίνακα απεικονίζει το πλήθος των σωστών προβλέψεων του μοντέλου (True Positive και True Negative δείγματα), ενώ η δευτερεύουσα διαγώνιος απεικονίζει το πλήθος των εσφαλμένων προβλέψεων του μοντέλου (False Positive και False Negative δείγματα).

Με βάση τα όσα ειπώθηκαν παραπάνω, η ακρίβεια ορίζεται ως εξής:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6.1}$$

όπου με TP, TN, FP και FN συμβολίζονται τα πλήθη των True Positive, True Negative, False Positive και False Negative δειγμάτων, αντίστοιχα. Με απλά λόγια, η ακρίβεια ορίζεται ως το κλάσμα των σωστών προβλέψεων του μοντέλου προς το συνολικό πλήθος των δειγμάτων. Όσο περισσότερα είναι τα δείγματα που το μοντέλο έχει προβλέψει σωστά, τόσο μεγαλύτερη είναι η ακρίβεια. Συνεπώς, η ακρίβεια αποτελεί ένα μέτρο προσδιορισμού των εύστοχων προβλέψεων που πραγματοποίησε το μοντέλο.

Κλείνοντας, θα πρέπει να αναφερθεί ότι στη συγκεκριμένη περίπτωση η ακρίβεια δεν μπορεί να χρησιμοποιηθεί από μόνη της για την αξιολόγηση της απόδοσης των μοντέλων. Αυτό συμβαίνει διότι, όπως παρουσιάστηκε και στο υποκεφάλαιο 5.4, τα πλήθη των δεδομένων ανά κατηγορία είναι αρκετά ανισόρροπα. Για παράδειγμα, αν χρησιμοποιούταν μόνο η ακρίβεια ως μετρική για την αξιολόγηση των μοντέλων, θα μπορούσε να υπάρχει ένα μοντέλο το οποίο έχει επιτύχει μεγάλη ακρίβεια, προβλέποντας σωστά όλα τα δείγματα της κλάσης EvenPaceStyle, η οποία έχει αρκετά περισσότερα δείγματα, και λάθος τα περισσότερα δείγματα της κλάσης AggressiveStyle, η οποία έχει αρκετά λιγότερα δείγματα. Κάτι τέτοιο θα οδηγούσε στο λάθος συμπέρασμα ότι έχει σχηματιστεί ένα πολύ αποδοτικό μοντέλο το οποίο είναι κατάλληλο για την επίλυση του προβλήματος, ενώ στην πραγματικότητα το μοντέλο χαρακτηρίζεται από υψηλή ανικανότητα διάκρισης της επιθετικής από την αποδεκτή οδηγική συμπεριφορά.

6.1.3.2 Εκτίμηση Περιοχής Κάτω από την Καμπύλη Λειτουργικών Χαρακτηριστικών (AUCROC Curve)

Η καμπύλη λειτουργικών χαρακτηριστικών (ROC - Receiver Operating Characteristic Curve) [79] αποτελεί ένα τρόπο έκφρασης της απόδοσης του μοντέλου. Η καμπύλη ROC εξαρτάται από δύο παραμέτρους, οι οποίες είναι οι εξής:

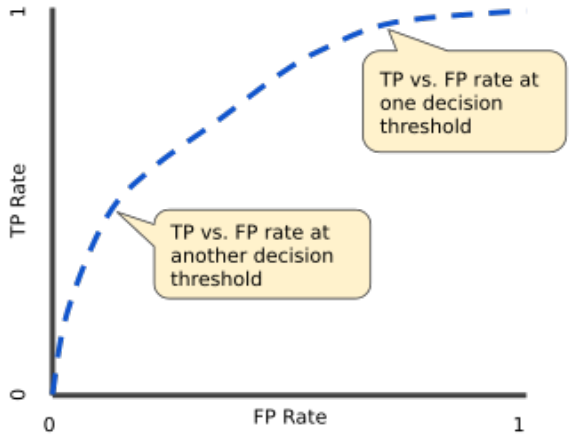
- True Positive Rate (TPR): Ο όρος αυτός εκφράζει το ποσοστό των δειγμάτων της θετικής κλάσης που ταξινομήθηκαν σωστά και τοποθετείται στον άξονα y.

$$TPR = \frac{TP}{TP + FN} \tag{6.2}$$

- False Positive Rate (FPR): Ο όρος αυτός εκφράζει το ποσοστό των δειγμάτων της αρνητικής κλάσης που ταξινομήθηκαν εσφαλμένα και τοποθετείται στον άξονα x.

$$FPR = \frac{FP}{FP + TN} \tag{6.3}$$

Η καμπύλη ROC σχεδιάζεται βάσει των τιμών που έχουν τα TPR και FPR για διάφορα κατώφλια ταξινόμησης (classification threshold). Ως classification threshold ορίζεται μια τιμή βάσει της οποίας γίνεται η ταξινόμηση των δειγμάτων. Αν η πιθανότητα ενός δείγματος να ανήκει στη θετική κλάση είναι μεγαλύτερη από το classification threshold, τότε το δείγμα ταξινομείται στη θετική κλάση. Σε αντίθετη περίπτωση, το δείγμα ταξινομείται στην αρνητική κλάση. Μια τυπική καμπύλη ROC απεικονίζεται στο σχήμα 6.1.



Πηγή: https://developers.google.com/machine-learning/crash-course/images/ROCCurve.svg

Σχήμα 6.1: Καμπύλη ROC

Συνεχίζοντας, το εμβαδόν της περιοχής κάτω από την καμπύλη ROC ορίζει έναν όρο ο οποίος ονομάζεται AUC (Area Under Curve). Ο όρος AUC αναπαριστά τον βαθμό διαχωρισιμότητας μεταξύ της θετικής και της αρνητικής κλάσης. Ουσιαστικά, εκφράζει το κατά πόσο μπορεί το εκάστοτε μοντέλο να διακρίνει την κλάση στην οποία ανήκει ένα τυχαίο δείγμα. Όσο μεγαλύτερος είναι ο όρος AUC τόσο αποτελεσματικότερα μπορεί το μοντέλο να διαχωρίσει τις δύο κλάσεις.

Με βάση τα όσα ειπώθηκαν παραπάνω, γίνεται πλήρως κατανοητό ότι η μετρική AUCROC μπορεί να χρησιμοποιηθεί σε συνδυασμό με την ακρίβεια, έτσι ώστε να επιλεχτεί το αποτελεσματικότερο μοντέλο. Με αυτόν τον τρόπο, είναι πολύ εύκολο να εντοπιστούν τα μοντέλα τα οποία δεν είναι τόσο καλά αλλά έχουν μεγάλη ακρίβεια, καθώς η μετρική AUCROC θα έχει αρκετά χαμηλή τιμή.

6.2 Πειραματική Αξιολόγηση Μοντέλων

Στο παρόν υποκεφάλαιο θα παρουσιαστούν τα αποτελέσματα των μοντέλων μηχανικής μάθησης που χρησιμοποιήθηκαν για την επίλυση του προβλήματος κατηγοριοποίησης της οδηγικής συμπεριφοράς. Για τον σχηματισμό και την εκπαίδευση των μοντέλων χρησιμοποιήθηκε η γλώσσα προγραμματισμού python [74] και πιο συγκεκριμένα οι βιβλιοθήκες xgboost [80], easyesn [81], tensorflow [82], scikit-learn [83] και tslearn [84]. Ακόμη, θα πρέπει να αναφερθεί ότι για τη βέλτιστη παραμετροποίηση των μοντέλων χρησιμοποιήθηκε ο αλγόριθμος Grid Search [85] της βιβλιοθήκης scikit-learn, προκειμένου να προσδιοριστούν οι τιμές των παραμέτρων του εκάστοτε μοντέλου. Θα πρέπει να αναφερθεί ότι στην ανάλυση που ακολουθεί παρουσιάζονται μόνο οι βέλτιστες παραμετροποιήσεις για κάθε μοντέλο και όχι όλες οι πιθανές παραμετροποιήσεις που εξετάστηκαν. Τέλος, υπενθυμίζεται ότι στο πρόβλημα κατηγοριοποίησης που μελετάται ως θετική κλάση ορίζεται η κατηγορία AggressiveStyle, ενώ ως αρνητική κλάση ορίζεται η κατηγορία EvenPaceStyle.

6.2.1 Απλά Μοντέλα

Κατά τη διαδικασία εκπαίδευσης των απλών μοντέλων, τα δεδομένα του συνόλου εκπαίδευσης έχουν προεπεξεργαστεί με τον τρόπο που παρουσιάστηκε στο

υποκεφάλαιο 5.3. Για τη κωδικοποίηση του πεδίου drivingStyle έχει χρησιμοποιηθεί η τεχνική Label Encoding σε όλα τα μοντέλα εκτός από το MLP, όπου έχει χρησιμοποιηθεί η τεχνική One-Hot Encoding. Τέλος, για την κανονικοποίηση των δεδομένων χρησιμοποιήθηκε η μέθοδος της κανονικής κανονικοποίησης. Συνεχίζοντας, θα πρέπει να αναφερθεί ότι για όλα τα απλά μοντέλα, εκτός του MLP δικτύου, παρατηρήθηκε ότι τα αποτελέσματα ήταν αρκετά καλύτερα όταν το σύνολο δεδομένων ήταν πιο ισορροπημένο. Για τον λόγο αυτό, αποφασίστηκε να απορριφθούν, με τυχαίο τρόπο, εγγραφές του συνόλου δεδομένων εκπαίδευσης που ανήκουν στην κατηγορία EvenPaceStyle. Πιο συγκεκριμένα, απορρίφθηκε το 40% των εγγραφών (περίπου 6000 εγγραφές) που έχουν χαρακτηριστεί από την ετικέτα EvenPaceStyle.

6.2.1.1 kNN

Για το μοντέλο kNN βρέθηκε ότι ο βέλτιστος αριθμός γειτόνων που πρέπει να εξετάζονται κατά τη κατηγοριοποίηση ενός δείγματος είναι ίσος με 5.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	675	654
Predicted Class Negative	112	5688

Πίνακας 6.2: kNN confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 89.2% και 87.7%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	940	958
Predicted Class Negative	347	4793

Πίνακας 6.3: kNN confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 81.5% και 78.2%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο kNN μπορεί να πραγματοποιήσει 6421 προβλέψεις/δευτερόλεπτο.

6.2.1.2 SVM

Για το μοντέλο SVM βρέθηκε ότι η βέλτιστη παραμετροποίηση είναι η εξής:

- $C = 100$
- $kernel = `rbf'$ [86]

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	622	678
Predicted Class Negative	165	5664

Πίνακας 6.4: SVM confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 88.2% και 84.2%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	732	1348
Predicted Class Negative	555	4403

Πίνακας 6.5: SVM confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 73% και 66.7%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο SVM μπορεί να πραγματοποιήσει 9445 προβλέψεις/δευτερόλεπτο.

6.2.1.3 Random Forest

Για το μοντέλο Random Forest βρέθηκε ότι η βέλτιστη παραμετροποίηση είναι η εξής:

- $criterion =$ '$entropy$'
- $max\_depth = 100$ (μέγιστο βάθος δένδρου)
- $min\_samples\_leaf = 1$ (ελάχιστο πλήθος δειγμάτων προκείμενου ένας κόμβος να μπορεί να θεωρηθεί φύλλο)
- $min\_samples\_split = 2$ (ελάχιστο πλήθος δειγμάτων που απαιτείται για το διαχωρισμό ενός κόμβου)
- $n\_estimators = 150$ (πλήθος δένδρων του δάσους)

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	690	412
Predicted Class Negative	97	5930

Πίνακας 6.6: Random Forest confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 92.9% και 90.6%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	768	826
Predicted Class Negative	519	4925

Πίνακας 6.7: Random Forest confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 80.9% και 72.6%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο Random Forest μπορεί να πραγματοποιήσει 62925 προβλέψεις/δευτερόλεπτο.

6.2.1.4 XGBoost

Για το μοντέλο XGBoost βρέθηκε ότι η βέλτιστη παραμετροποίηση είναι η εξής:

- $learning\_rate = 0.2$
- $max\_depth = 20$
- $alpha = 0$ (όρος κανονικοποίησης)
- $gamma = 0$ (όρος κανονικοποίησης)
- $lambda = 1$ (όρος κανονικοποίησης)
- $n\_estimators = 200$ (πλήθος απλών μοντέλων)

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	709	353
Predicted Class Negative	78	5989

Πίνακας 6.8: XGBoost confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 93.4% και 92.3%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	804	823
Predicted Class Negative	483	4928

Πίνακας 6.9: XGBoost confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 81.4% και 74.1%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο XGBoost μπορεί να πραγματοποιήσει 358901 προβλέψεις/δευτερόλεπτο.

6.2.1.5 MLP

Η δομή του μοντέλου MLP που χρησιμοποιήθηκε είναι η εξής:

- Επίπεδο εισόδου: Στο επίπεδο εισόδου το πλήθος των νευρώνων είναι ίσο με 14, καθώς τόσα είναι και τα πεδία μιας εγγραφής που δέχεται το νευρωνικό δίκτυο ως είσοδο.
- Κρυφά επίπεδα: Χρησιμοποιήθηκαν τρία κρυφά επίπεδα. Στο πρώτο το πλήθος των νευρώνων είναι ίσο με 256, στο δεύτερο ίσο με 128 και στο τρίτο ίσο με 64. Σε όλα τα κρυφά επίπεδα ως συνάρτηση ενεργοποίησης των νευρώνων χρησιμοποιήθηκε η συνάρτηση γραμμικής ανόρθωσης (Rectified Linear Unit), η οποία έχει περιγραφεί στο υποκεφάλαιο 4.1.2.1.
- Επίπεδο εξόδου: Στο επίπεδο εξόδου το πλήθος των νευρώνων είναι ίσο με 2, καθώς τόσες είναι και οι πιθανές ομάδες του προβλήματος κατηγοριοποίησης.

Για την ενημέρωση των βαρών των συνάψεων του νευρωνικού δικτύου ως συνάρτηση κόστους επιλέχτηκε η συνάρτηση Cross-Entropy σε συνδυασμό με τον αλγόριθμο βελτιστοποίησης Adam. Ο ρυθμός μάθησης ορίστηκε ίσος με 0.001.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	624	320
Predicted Class Negative	163	6022

Πίνακας 6.10: MLP confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 93.2% και 87.1%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	549	627
Predicted Class Negative	738	5124

Πίνακας 6.11: MLP confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 80.6% και 65.9%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο MLP μπορεί να πραγματοποιήσει 54969 προβλέψεις/δευτερόλεπτο.

6.2.2 Μοντέλα Χρονοσειρών

Κατά τη διαδικασία εκπαίδευσης των μοντέλων χρονοσειρών, τα δεδομένα του συνόλου εκπαίδευσης έχουν προεπεξεργαστεί με τον τρόπο που παρουσιάστηκε στο υποκεφάλαιο 5.3. Στη συνέχεια, έχει εφαρμοστεί ένα κυλιόμενο παράθυρο μεγέθους ίσου με 20 πάνω στα δεδομένα, έτσι ώστε τα δεδομένα που τροφοδοτούνται ως είσοδος στο εκάστοτε μοντέλο να έχουν τη μορφή πολυδιάστατης χρονοσειράς. Για την κωδικοποίηση του πεδίου drivingStyle έχει χρησιμοποιηθεί η τεχνική Label Encoding σε όλα τα μοντέλα εκτός από τα ESN και LSTM, όπου έχει χρησιμοποιηθεί η τεχνική One-Hot Encoding. Ακόμη, για την κανονικοποίηση των δεδομένων χρησιμοποιήθηκε η μέθοδος της κανονικής κανονικοποίησης σε όλα τα μοντέλα εκτός από το LSTM, που έχει χρησιμοποιηθεί η μέθοδος Ελάχιστου-Μέγιστου για το διάστημα [-1, 1]. Η επιλογή αυτή έγινε λόγω της συνάρτησης υπερβολικής εφαπτομένης που χαρακτηρίζει τη λειτουργία των LSTM δικτύων, και η οποία έχει σύνολο τιμών το διάστημα [-1, 1]. Συνεχίζοντας, θα πρέπει να αναφερθεί ότι για το μοντέλο ESN παρατηρήθηκε ότι τα αποτελέσματα ήταν αρκετά καλύτερα όταν το σύνολο δεδομένων ήταν πιο ισορροπημένο. Για τον λόγο αυτό, αποφασίστηκε να απορριφθούν, με τυχαίο τρόπο, εγγραφές του συνόλου δεδομένων που ανήκουν στην κατηγορία EvenPaceStyle και στη συνέχεια να γίνει ο διαχωρισμός του συνόλου δεδομένων σε σύνολο εκπαίδευσης (70%) και σύνολο ελέγχου (30%). Πιο συγκεκριμένα, απορρίφθηκε το 30% των εγγραφών (περίπου 6500 εγγραφές) που έχουν χαρακτηριστεί από την ετικέτα EvenPaceStyle. Τέλος, θα πρέπει να διευκρινιστεί ότι για τη σύγκριση των χρονοσειρών στα μοντέλα kNN και SVM δοκιμάστηκαν οι αλγόριθμοι SAX και DTW. Ωστόσο, στα αποτελέσματα των μοντέλων αναφέρεται μόνο ο αλγόριθμος DTW καθώς οδήγησε σε αρκετά καλύτερα αποτελέσματα σε σχέση με τον αλγόριθμο SAX.

6.2.2.1 kNN με DTW

Για το μοντέλο kNN με χρήση του αλγορίθμου DTW για τη σύγκριση χρονοσειρών, βρέθηκε ότι ο βέλτιστος αριθμός γειτόνων που πρέπει να εξετάζονται κατά τη κατηγοριοποίηση ενός δείγματος είναι ίσος με 5.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	715	125
Predicted Class Negative	124	5954

Πίνακας 6.12: kNN με DTW confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 96.4% και 91.6%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1168	127
Predicted Class Negative	110	5443

Πίνακας 6.13: kNN με DTW confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 96.5% και 94.5%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο kNN με DTW μπορεί να πραγματοποιήσει 0.8 προβλέψεις/δευτερόλεπτο.

6.2.2.2 SVM με DTW

Για το μοντέλο SVM με χρήση του αλγορίθμου DTW για τη σύγκριση χρονοσειρών, βρέθηκε ότι η βέλτιστη παραμετροποίηση είναι η εξής:

- $C = 100$
- $kernel =$ 'gak' [87]

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	720	100
Predicted Class Negative	119	5979

Πίνακας 6.14: SVM με DTW confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 96.8% και 92%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1235	61
Predicted Class Negative	43	5509

Πίνακας 6.15: SVM με DTW confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 98.4% και 97.7%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο SVM με DTW μπορεί να πραγματοποιήσει 0.6 προβλέψεις/δευτερόλεπτο.

6.2.2.3 Learning Shapelets

Για το μοντέλο Learning Shapelets [88], το οποίο ουσιαστικά αποτελεί ένα απλό νευρωνικό δίκτυο που βασίζεται στη λογική των Time Series Shapelets για τη σύγκριση των χρονοσειρών, βρέθηκε ότι η βέλτιστη παραμετροποίηση είναι η εξής:

- $shapelet\_length = 0.3$ (ποσοστό υποακολουθίας που εξετάζεται)
- $total\_lengths = 3$ (μέσω αυτής της παραμέτρου ορίζεται ότι θα εξετάζονται υποακολουθίες μήκους $[1 \cdot shapelet\_length,\ 2 \cdot shapelet\_length,\ 3 \cdot shapelet\_length]$)
- $batch\_size = 128$
- $max\_iter = 12000$ (epochs)
- $optimizer =$ '$adam$'
- $weight\_regularizer = 0.1$ (παράμετρος α για L2 Regularization)

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	631	308
Predicted Class Negative	208	5771

Πίνακας 6.16: Learning Shapelets confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 92.5% και 85%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1157	347
Predicted Class Negative	121	5223

Πίνακας 6.17: Learning Shapelets confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 93.1% και 92.1%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο Learning Shapelets μπορεί να πραγματοποιήσει 2922 προβλέψεις/δευτερόλεπτο.

6.2.2.4 LSTM

Η δομή του μοντέλου LSTM που χρησιμοποιήθηκε είναι η εξής:

- Επίπεδο εισόδου: Στο επίπεδο εισόδου το πλήθος των κελιών LSTM είναι ίσο με 128 και ως συνάρτηση ενεργοποίησης κάθε κελιού έχει οριστεί η συνάρτηση

υπερβολικής εφαπτομένης.

- Κρυφά επίπεδα: Χρησιμοποιήθηκαν δύο κρυφά επίπεδα. Στο πρώτο το πλήθος των κελιών LSTM είναι ίσο με 64 και στο δεύτερο ίσο με 32. Και στα δύο κρυφά επίπεδα ως συνάρτηση ενεργοποίησης των κελιών ορίστηκε η συνάρτηση υπερβολικής εφαπτομένης.

- Επίπεδο εξόδου: Στο επίπεδο εξόδου το πλήθος των νευρώνων είναι ίσο με 2, καθώς τόσες είναι και οι πιθανές ομάδες του προβλήματος κατηγοριοποίησης.

Για την ενημέρωση των βαρών των συνάψεων του νευρωνικού δικτύου ως συνάρτηση κόστους επιλέχτηκε η συνάρτηση Cross-Entropy σε συνδυασμό με τον αλγόριθμο βελτιστοποίησης Adam. Ο ρυθμός μάθησης ορίστηκε ίσος με 0.001.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	728	96
Predicted Class Negative	104	5990

Πίνακας 6.18: LSTM confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 97.1% και 93%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1243	67
Predicted Class Negative	35	5503

Πίνακας 6.19: LSTM confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 98.5% και 98%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο LSTM μπορεί να πραγματοποιήσει 3156 προβλέψεις/δευτερόλεπτο.

6.2.2.5 ESN

Η δομή του μοντέλου ESN που χρησιμοποιήθηκε είναι η εξής:

- Επίπεδο εισόδου: Στο επίπεδο εισόδου το πλήθος των νευρώνων είναι ίσο με 14, καθώς τόσα είναι και τα πεδία μιας εγγραφής που δέχεται το νευρωνικό δίκτυο ως είσοδο.

- Επίπεδο Reservoir: Στο επίπεδο Reservoir το πλήθος των νευρώνων είναι ίσο με 1700.

- Επίπεδο εξόδου: Στο επίπεδο εξόδου το πλήθος των νευρώνων είναι ίσο με 2, καθώς τόσες είναι και οι πιθανές ομάδες του προβλήματος κατηγοριοποίησης.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στο σύνολο δεδομένων ελέγχου ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	626	114
Predicted Class Negative	180	1903

Πίνακας 6.20: ESN confusion matrix test set

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 89.6% και 86%, αντίστοιχα.

Οι προβλέψεις που πραγματοποίησε το μοντέλο στην πρώτη από τις διαδρομές του Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1128	529
Predicted Class Negative	150	5041

Πίνακας 6.21: ESN confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 90% και 89.4%, αντίστοιχα.

Τέλος, βρέθηκε ότι το μοντέλο ESN μπορεί να πραγματοποιήσει 27 προβλέψεις/δευτερόλεπτο.

6.3 Σχολιασμός Αποτελεσμάτων

Σε αυτό το υποκεφάλαιο θα παρουσιαστεί μια σύνοψη των αποτελεσμάτων που παρουσιάστηκαν στο υποκεφάλαιο 6.2 καθώς και μια σύγκριση μεταξύ των επιδόσεων που επιτεύχθηκαν από τα διάφορα μοντέλα μηχανικής μάθησης που εξετάστηκαν. Για την αναπαράσταση των γραφικών παραστάσεων χρησιμοποιήθηκε η γλώσσα προγραμματισμού python [74] και πιο συγκεκριμένα η βιβλιοθήκη Matplotlib [89].

Στους πίνακες 6.22 και 6.23 παρουσιάζονται συγκεντρωτικά τα αποτελέσματα των επιμέρους μοντέλων που εξετάστηκαν τόσο για τη μετρική accuracy όσο και για τη μετρική AUCROC.

Μοντέλο	Test Set Accuracy	Opel Corsa Route Accuracy	Προβλέψεις/ Δευτερόλεπτο
kNN	89.2%	81.5%	6421
SVM	88.2%	73%	9445
Random Forest	92.9%	80.9%	62925
XGB	93.4%	81.4%	358901
MLP	93.2%	80.6%	54969

kNN με DTW	96.4%	96.5%	0.8
SVM με DTW	96.8%	98.4%	0.6
Learning Shapelets	92.5%	93.1%	2922
LSTM	97.1%	98.5%	3156
ESN	89.6%	90%	27

Πίνακας 6.22: Η μετρική accuracy για όλα τα μοντέλα

Μοντέλο	Test Set AUCROC	Opel Corsa Route AUCROC	Προβλέψεις/ Δευτερόλεπτο
kNN	87.7%	78.2%	6421
SVM	84.2%	66.7%	9445
Random Forest	90.6%	72.6%	62925
XGB	92.3%	74.1%	358901
MLP	87.1%	65.9%	54969
kNN με DTW	91.6%	94.5%	0.8
SVM με DTW	92%	97.7%	0.6
Learning Shapelets	85%	92.1%	2922
LSTM	93%	98%	3156
ESN	86%	89.4%	27

Πίνακας 6.23: Η μετρική AUCROC για όλα τα μοντέλα

Στον πίνακα 6.24 παρουσιάζονται τα πλήθη των false positive και των false negative προβλέψεων των μοντέλων για τη διαδρομή που πραγματοποίησε το όχημα Opel Corsa 1.3 HDi. Θα πρέπει να σημειωθεί ότι για τα μοντέλα που εκπαιδεύτηκαν με υποδειγματοληψία του συνόλου δεδομένων παρουσιάζονται επίσης και τα αποτελέσματα που προέκυψαν από την εκπαίδευσή τους σε ολόκληρο το σύνολο δεδομένων. Για τη συγκεκριμένη διαδρομή το πλήθος των positive εγγραφών είναι ίσο με 1287, ενώ το πλήθος των negative εγγραφών είναι ίσο με 6307.

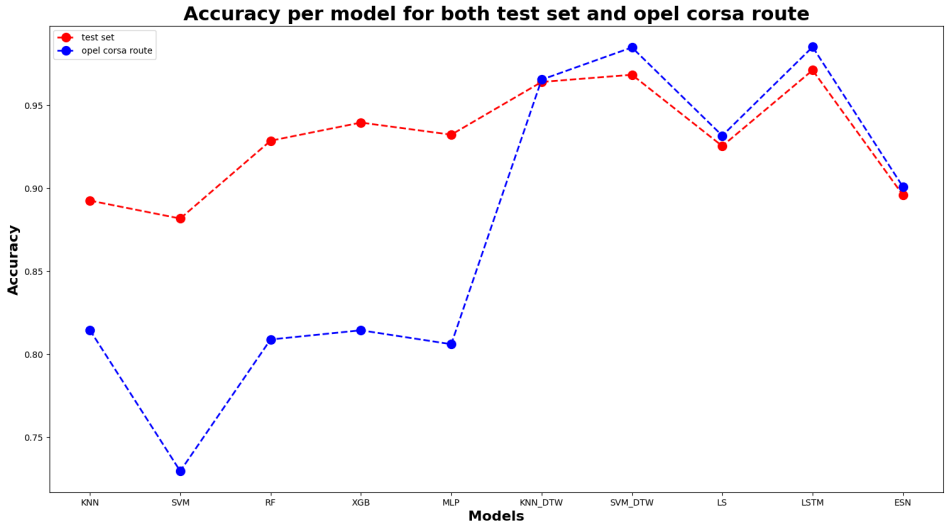
Μοντέλο	FP με δειγματοληψία	FP χωρίς δειγματοληψία	FN με δειγματοληψία	FN χωρίς δειγματοληψία
kNN	958	296	347	710
SVM	1348	1522	555	500
Random Forest	826	211	519	959
XGB	823	331	483	825
MLP	-	627	-	738
kNN με DTW	-	127	-	110
SVM με DTW	-	61	-	43
Learning Shapelets	-	347	-	121
LSTM	-	67	-	35
ESN	529	50	150	704

Πίνακας 6.24: FP και FN με και χωρίς δειγματοληψία

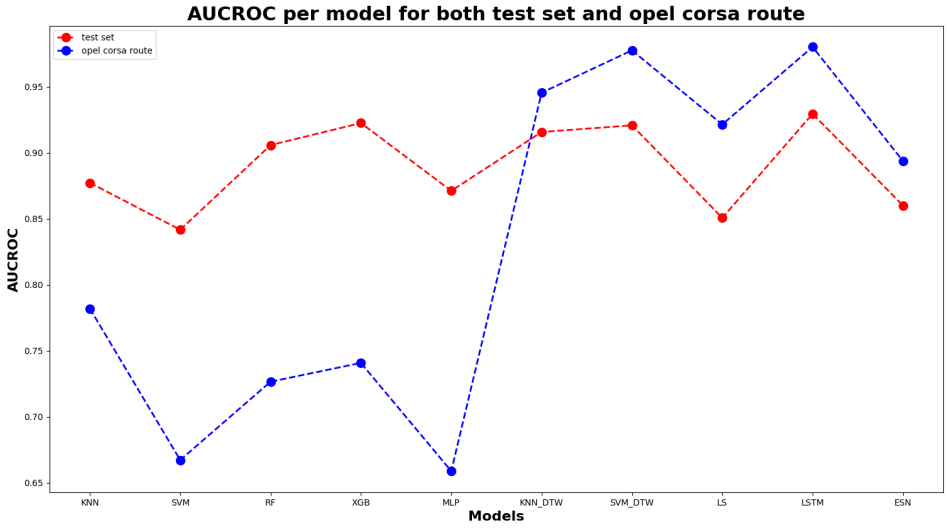
Ξεκινώντας, θα πρέπει να σχολιασθεί το γεγονός ότι στα περισσότερα μοντέλα που πραγματοποιήθηκε υποδειγματοληψία του συνόλου δεδομένων κατά τη φάση εκπαίδευσης (kNN, SVM, Random Forest, XGBoost και ESN) το πλήθος των false positive εγγραφών (εγγραφές που ανήκουν στην κλάση EvenPaceStyle και ταξινομήθηκαν στην κλάση AggressiveStyle) είναι αρκετά μεγαλύτερο από το πλήθος των false negative εγγραφών (εγγραφές που ανήκουν στην κλάση AggressiveStyle και ταξινομήθηκαν στην κλάση EvenPaceStyle). Αυτό συμβαίνει διότι με την υποδειγματοληψία μειώθηκε το πλήθος των εγγραφών της κλάσης EvenPaceStyle, γεγονός που δεν επέτρεψε στα μοντέλα που εκπαιδεύτηκαν με αυτόν τον τρόπο να συλλάβουν πλήρως τα μοτίβα και τις σχέσεις που ορίζουν οι εγγραφές τύπου EvenPaceStyle. Αυτό είχε ως αποτελέσματα την αύξηση των false positive προβλέψεων καθώς αυξάνονται οι πιθανότητες μια εγγραφή τύπου EvenPaceStyle, η οποία είναι εντελώς διαφορετική από αυτές που χρησιμοποιήθηκαν στη φάση εκπαίδευσης, να ταξινομηθεί στην κλάση AggressiveStyle. Αντίστοιχα, η εκπαίδευση των ίδιων μοντέλων χωρίς να έχει προηγηθεί υποδειγματοληψία του συνόλου δεδομένων έχει ως αποτέλεσμα το πλήθος των false negative προβλέψεων να είναι μεγαλύτερο από το πλήθος των false positive προβλέψεων. Κάτι τέτοιο δεν είναι αποδεκτό καθώς θα δημιουργούνταν μοντέλα τα οποία δεν θα μπορούσαν να εντοπίσουν τις επιθετικές συμπεριφορές των οδηγών των οχημάτων, αφού, όπως φαίνεται και από τον πίνακα 6.23, το μεγαλύτερο πλήθος των AggressiveStyle δειγμάτων θα ταξινομούνταν εσφαλμένα ως EvenPaceStyle. Συνεπώς, μπορεί η υποδειγματοληψία που πραγματοποιήθηκε να οδήγησε στην αύξηση των false positive προβλέψεων, ωστόσο είχε ως αποτέλεσμα τη δημιουργία αποτελεσματικότερων μοντέλων τα οποία μπορούν να γενικεύσουν περισσότερο πάνω στο σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία. Ακόμη, αξίζει να σημειωθεί ότι στην περίπτωση του SVM η υποδειγματοληψία δεν φαίνεται να κάνει τόσο μεγάλη διαφορά, γεγονός που υποδεικνύει ότι το συγκεκριμένο μοντέλο δεν είναι κατάλληλο για την επίλυση του προβλήματος. Τέλος, επισημαίνεται ότι το ESN ήταν το μοναδικό μοντέλο από τα μοντέλα χρονοσειρών για το οποίο παρατηρήθηκε βελτίωση των επιδόσεων του με την υποδειγματοληψία του συνόλου δεδομένων κατά τη φάση εκπαίδευσης. Αυτό υποδηλώνει ότι ο ιδιαίτερος τρόπος εκπαίδευσης του ESN μοντέλου (τυχαιότητα του επιπέδου Reservoir) δεν είναι τόσο αποδοτικός για την επίλυση του προβλήματος κατηγοριοποίησης της οδηγικής συμπεριφοράς που ορίζεται από το σύνολο δεδομένων που χρησιμοποιήθηκε.

Στη συνέχεια πραγματοποιείται η αξιολόγηση και η σύγκριση των μοντέλων που παρουσιάστηκαν παραπάνω με σκοπό τον εντοπισμό του αποτελεσματικότερου μοντέλου, το οποίο στη συνέχεια θα χρησιμοποιηθεί στο σύστημα που υλοποιήθηκε σε αυτήν την διπλωματική εργασία. Θα πρέπει να σημειωθεί ότι αφού τα διάφορα μοντέλα που εξετάστηκαν έχουν εκπαιδευτεί με διαφορετικό τρόπο, η αξιολόγηση και η σύγκριση τους βασίζεται στη διαδρομή που πραγματοποίησε το Opel Corsa 1.3 HDi. Η διαδρομή αυτή ορίζει ένα κοινό σύνολο εγγραφών βάσει του οποίου η απόδοση των επιμέρους μοντέλων μπορεί να ελεγχθεί με αντικειμενικό και δίκαιο τρόπο.

Στα σχήματα 6.2 και 6.3 παρουσιάζονται οι γραφικές παραστάσεις που ορίζονται από τις τιμές των μετρικών accuracy και AUCROC των πινάκων 6.22 και 6.23, αντίστοιχα. Επισημαίνεται ότι η διαδρομή που πραγματοποίησε το Opel Corsa 1.3 HDi χαρακτηρίζεται από τη γραμμή που έχει μπλε χρώμα.



Σχήμα 6.2: Η μετρική accuracy των μοντέλων

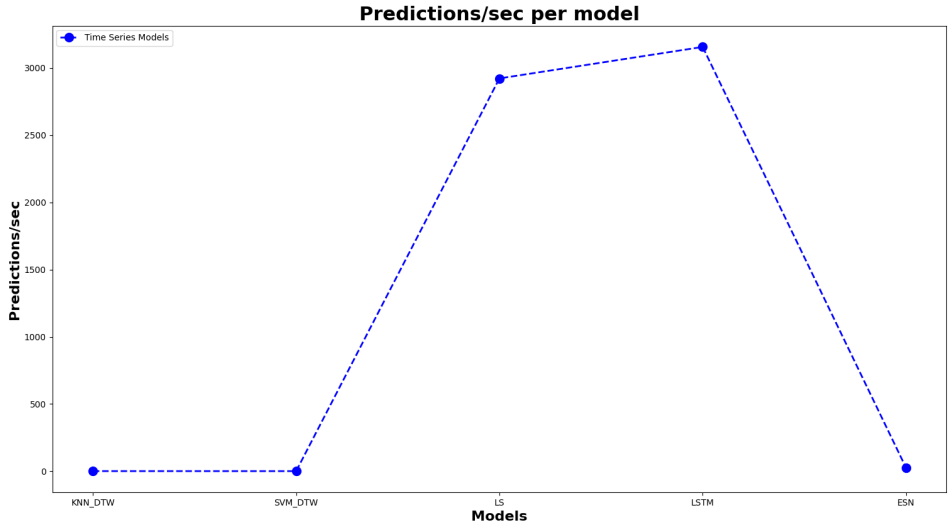


Σχήμα 6.3: Η μετρική AUCROC των μοντέλων

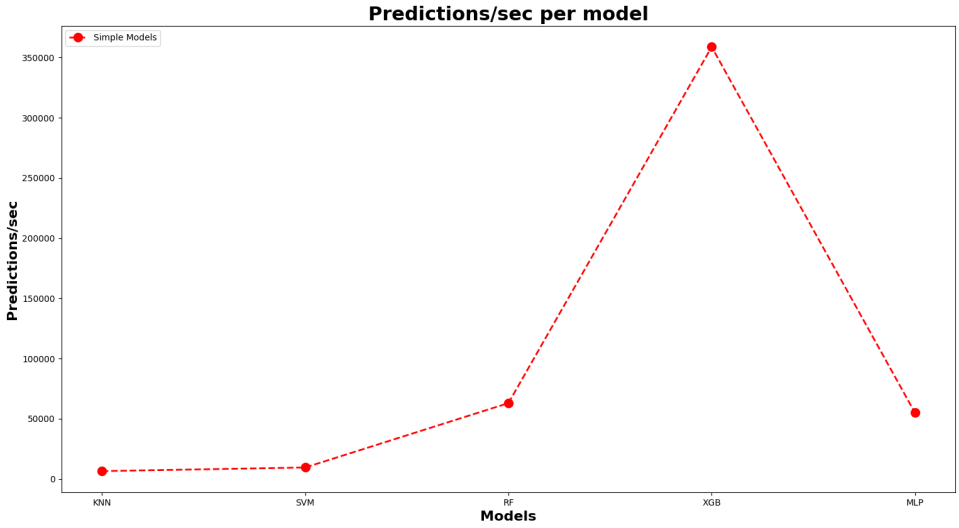
Παρατηρώντας τα σχήματα 6.2 και 6.3 και τους πίνακες 6.22, 6.23 και 6.24, γίνεται αμέσως αντιληπτό ότι η απόδοση των μοντέλων χρονοσειρών είναι αρκετά υψηλότερη από αυτή που επιτεύχθηκε από τα απλά μοντέλα. Πιο συγκεκριμένα, τα μοντέλα χρονοσειρών πέτυχαν υψηλότερη τιμή τόσο για τη μετρική accuracy όσο και για τη μετρική AUCROC, ενώ παράλληλα τα πλήθη των false positive και των false negative προβλέψεών τους ήταν αρκετά χαμηλότερα από αυτά των απλών μοντέλων. Αυτό ήταν αναμενόμενο διότι το πρόβλημα κατηγοριοποίησης που πρέπει να επιλυθεί απαρτίζεται από δεδομένα τα οποία δεν είναι ανεξάρτητα μεταξύ τους. Κάθε διαδρομή που έχει πραγματοποιηθεί ορίζεται από συνεχόμενες εγγραφές δεδομένων μεταξύ των οποίων υπάρχουν σχέσεις εξάρτησης και αίτιου-αποτελέσματος. Οι σχέσεις αυτές ορίζουν μοτίβα τα οποία δεν μπορούν να εντοπιστούν από τα απλά μοντέλα κατά τη φάση

εκπαίδευσης. Συνεπώς, για την επίλυση του προβλήματος κατηγοριοποίησης της οδηγικής συμπεριφοράς θα πρέπει να αναζητηθεί ένα πιο σύνθετο μοντέλο, δηλαδή ένα μοντέλο το οποίο μπορεί να εντοπίσει τα μοτίβα που υπάρχουν στις χρονοσειρές που ορίζουν τα δεδομένα των διαδρομών. Από τα μοντέλα χρονοσειρών που εξετάστηκαν φαίνεται να ξεχωρίζουν τα μοντέλα LSTM, kNN με DTW και SVM με DTW, τα οποία έχουν επιτύχει εξαιρετικά αποτελέσματα. Πιο συγκεκριμένα, είναι τα μοναδικά μοντέλα τα οποία έχουν πετύχει τιμές υψηλότερες του 94.5% και για τις δύο μετρικές αξιολόγησης που χρησιμοποιήθηκαν. Θα πρέπει να σημειωθεί ότι και τα αποτελέσματα των μοντέλων Learning Shapelets και ESN είναι αρκέτα καλύτερα από αυτά των απλών μοντέλων, ωστόσο οι επιδόσεις τους δεν συγκρίνονται με αυτές των υπόλοιπων μοντέλων χρονοσειρών. Το πλήθος των false postive προβλέψεων του μοντέλου Learning Shapelets είναι αρκέτα μεγαλύτερο από αυτό των υπόλοιπων μοντέλων χρονοσειρών, ενώ το πλήθος των false positive προβλέψεων του μοντέλου ESN είναι ιδιαίτερα μεγάλο λόγω της υποδειγματοληψίας που πραγματοποιήθηκε κατά τη φάση εκπαίδευσης. Σε αυτό το σημείο, θα πρέπει να τονισθεί πως το γεγονός ότι η απόδοση του μοντέλου ESN είναι χειρότερη από αυτήν όλων των υπόλοιπων μοντέλων χρονοσειρών πιθανότατα οφείλεται στον ιδιαίτερο τρόπο λειτουργίας αυτού του μοντέλου και στην τυχαιότητα του Reservoir επιπέδου.

Συνεχίζοντας, στα σχήματα 6.4 και 6.5 παρουσιάζονται οι ρυθμοί απόδοσης των μοντέλων που εξετάστηκαν. Ως ρυθμός απόδοσης του μοντέλου ορίζεται το πλήθος των προβλέψεων που μπορεί αυτό να πραγματοποιήσει στη μονάδα του χρόνου. Είναι εμφανές ότι τα απλά μοντέλα χαρακτηρίζονται από υψηλότερο ρυθμό απόδοσης και επομένως από μεγαλύτερη ταχύτητα στις προβλέψεις τους. Κάτι τέτοιο ήταν αναμενόμενο καθώς τα δεδομένα που καλούνται να αντιμετωπίσουν τα απλά μοντέλα είναι μεμονωμένες εγγραφές του συνόλου δεδομένων. Αντίθετα, τα μοντέλα χρονοσειρών δέχονται περισσότερη πληροφορία που πρέπει να επεξεργαστούν, καθώς τροφοδοτούνται με είσοδο δεδομένα που έχουν τη μορφή πολυδιάστατης χρονοσειράς μήκους 20 εγγραφών. Για παράδειγμα, για τα μοντέλα SVM και kNN με DTW παρατηρείται ότι ενώ μπορούν να πραγματοποιήσουν προβλέψεις με μεγάλη ακρίβεια ο ρυθμός απόδοσης του είναι υπερβολικά χαμηλός. Πιο συγκεκριμένα, τα μοντέλα αυτά είναι ικανά να πραγματοποιήσουν κάτι λιγότερο από μια πρόβλεψη το δευτερόλεπτο. Είναι πλήρως κατανοητό ότι ένα τέτοιο χαρακτηριστικό δεν είναι επιθυμητό αφού το μοντέλο θα πρέπει να είναι ικανό να διαχειριστεί και να πραγματοποιήσει προβλέψεις πάνω σε μεγάλο όγκο δεδομένων. Ακόμη, θα πρέπει να σημειωθεί ότι ο χαμηλός ρυθμός απόδοσης των μοντέλων kNN με DTW και SVM με DTW οφείλεται στην χρονική πολυπλοκότητα εφαρμογής του αλγορίθμου DTW, ιδιαίτερα στην περίπτωση όπου τα δεδομένα αποτελούν πολυδιάστατες χρονοσειρές.



Σχήμα 6.4: Ρυθμοί απόδοσης μοντέλων χρονοσειρών



Σχήμα 6.5: Ρυθμοί απόδοσης απλών μοντέλων

Με βάση τα όσα ειπώθηκαν παραπάνω, κρίθηκε ότι το μοντέλο με τη καλύτερη απόδοση πάνω στο σύνολο δεδομένων είναι το νευρωνικό δίκτυο LSTM. Είναι ξεκάθαρο ότι για την αποτελεσματική επίλυση του προβλήματος της κατηγοριοποίησης της οδηγικής συμπεριφοράς απαιτείται ένα μοντέλο χρονοσειρών το οποίο θα είναι σε θέση να αναλύσει τον τρόπο με τον οποίο παρελθοντικές καταστάσεις επηρέασαν την τρέχουσα συμπεριφορά του οδηγού του εκάστοτε οχήματος. Από τα μοντέλα χρονοσειρών που εξετάστηκαν στην παρούσα διπλωματική εργασία το μοντέλο LSTM ήταν αυτό που ξεχώρισε. Πιο συγκεκριμένα, το μοντέλο LSTM έχει επιτύχει την υψηλότερη τιμή τόσο για τη μετρική accuracy (98.5%) όσο και για τη μετρική AUCROC (98%). Αυτό οφείλεται στην φύση του προβλήματος κατηγοριοποίησης που πρέπει να επιλυθεί, το οποίο απαρτίζεται από εγγραφές δεδομένων μεταξύ των οποίων

υπάρχει εξάρτηση, και στην καταλληλότητα του LSTM δικτύου να εντοπίζει μοτίβα σε δεδομένα χρονοσειρών. Συνεχίζοντας, θα πρέπει να τονιστεί ότι οι τιμές των μετρικών accuracy και AUCROC που πέτυχαν τα μοντέλα kNN και SVM με DTW ήταν αρκετά κοντά σε αυτές που πέτυχε το LSTM. Ωστόσο, το γεγονός ότι οι τιμές που πέτυχε το LSTM ήταν ελάχιστα καλύτερες σε συνδυασμό με το γεγονός ότι ο ρυθμός απόδοσης του μοντέλου LSTM είναι αρκετά μεγαλύτερος από αυτούς των μοντέλων kNN και SVM με DTW οδήγησε στο συμπέρασμα ότι το μοντέλο LSTM αποτελεί το καταλληλότερο μοντέλο για την επίλυση του προβλήματος κατηγοριοποίησης της οδηγικής συμπεριφοράς που ορίζεται από το δοσμένο σύνολο δεδομένων. Κλείνοντας, θα πρέπει να αναφερθεί ότι μπορεί ο ρυθμός απόδοσης του μοντέλου LSTM να είναι χαμηλός σε σχέση με τους ρυθμούς απόδοσης των απλών μοντέλων, ωστόσο το μοντέλο LSTM μπορεί να πραγματοποιήσει 3000 προβλέψεις ανά δευτερόλεπτο, γεγονός που του επιτρέπει να διαχειριστεί και να εκτελέσει προβλέψεις σε μεγάλο όγκο δεδομένων με ταχύτατο ρυθμό.

6.4 Παραλλαγές του LSTM

Στο υποκεφάλαιο 6.3 βρέθηκε ότι το νευρωνικό δίκτυο LSTM είναι το πιο κατάλληλο μοντέλο για την επίλυση του προβλήματος κατηγοριοποίησης της οδηγικής συμπεριφοράς που ορίζεται από το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία. Σε αυτό το υποκεφάλαιο εξετάζονται οι πιο ευρέως διαδεδομένες και πλέον σύγχρονες παραλλαγές του νευρωνικού δικτύου LSTM, ενώ στη συνέχεια συγκρίνονται τα αποτελέσματά τους με αυτά τα οποία επιτεύχθηκαν από το απλό μοντέλο LSTM και παρουσιάστηκαν στο υποκεφάλαιο 6.3. Θα πρέπει να σημειωθεί ότι για όλες τις παραλλαγές που δοκιμάστηκαν διατηρήθηκε η αρχιτεκτονική του νευρωνικού δικτύου που παρουσιάστηκε στο υποκεφάλαιο 6.2.2.4. Ουσιαστικά οι παραλλαγές του LSTM εντοπίζονται στον τρόπο με το οποίο λειτουργεί το LSTM cell, το οποίο παρουσιάστηκε στο υποκεφάλαιο 4.1.3.5.

Η πρώτη παραλλαγή που δοκιμάστηκε ήταν το κελί GRU (Gated Recurrent Unit) [90]. Οι προβλέψεις που πραγματοποίησε το μοντέλο GRU στην πρώτη από τις διαδρομές που πραγματοποίησε το Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1235	65
Predicted Class Negative	43	5505

Πίνακας 6.25: GRU confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 98.4% και 97.7%, ενώ ο ρυθμός απόδοσης του μοντέλου βρέθηκε ίσος με 4000 προβλέψεις/δευτερόλεπτο.

Η δεύτερη παραλλαγή που δοκιμάστηκε ήταν το κελί Peephole LSTM cell [91]. Οι προβλέψεις που πραγματοποίησε το μοντέλο Peephole LSTM στην πρώτη από τις διαδρομές που πραγματοποίησε το Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1215	64

Predicted Class Negative	63	5506

Πίνακας 6.26: Peephole LSTM confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 98.1% και 96.9%, ενώ ο ρυθμός απόδοσης του μοντέλου βρέθηκε ίσος με 3434 προβλέψεις/δευτερόλεπτο.

Η τρίτη και τελευταία παραλλαγή που δοκιμάστηκε ήταν το κελί NAS (Neural Architecture Search) LSTM [92]. Οι προβλέψεις που πραγματοποίησε το μοντέλο NAS LSTM στην πρώτη από τις διαδρομές που πραγματοποίησε το Opel Corsa 1.3 HDi ήταν οι εξής:

	Actual Class Positive	Actual Class Negative
Predicted Class Positive	1235	73
Predicted Class Negative	43	5497

Πίνακας 6.27: NAS LSTM confusion matrix opel corsa

Οι τιμές των μετρικών accuracy και AUCROC για τις παραπάνω προβλέψεις βρέθηκαν ίσες με 98.3% και 97.7%, ενώ ο ρυθμός απόδοσης του μοντέλου βρέθηκε ίσος με 2623 προβλέψεις/δευτερόλεπτο.

Στον πίνακα 6.28 παρουσιάζονται συγκεντρωτικά τα αποτελέσματα τόσο του κλασικού LSTM όσο και των παραλλαγών αυτού, οι οποίες παρουσιάστηκαν παραπάνω, για τη διαδρομή που πραγματοποίησε το Opel Corsa 1.3 HDi.

Μοντέλο	Accuracy	AUCROC	Προβλέψεις/ Δευτερόλεπτο
LSTM	98.5%	98%	3156
GRU	98.4%	97.7%	4000
Peephole LSTM	98.1%	96.9%	3434
NAS LSTM	98.3%	97.7%	2623

Πίνακας 6.28: Σύγκριση του απλού LSTM και των παραλλαγών του

Παρατηρώντας τον πίνακα 6.28 φαίνεται ότι δεν υπάρχουν ουσιαστικές διαφορές μεταξύ των επιδόσεων του κλασικού LSTM και των παραλλαγών του. Το κλασικό LSTM έχει επιτύχει υψηλότερες τιμές για τις μετρικές accuracy και AUCROC, ωστόσο ο ρυθμός απόδοσης του είναι χαμηλότερος από αυτούς των μοντέλων GRU και Peephole LSTM. Αν και οι διαφορές μεταξύ των μοντέλων του πίνακα 6.28 είναι πολύ μικρές, το κλασικό μοντέλο LSTM παραμένει το αποδοτικότερο καθώς έχει τις υψηλότερες τιμές για τις μετρικές accuracy και AUCROC. Συνεπώς, με βάση τα όσα ειπώθηκαν στο κεφάλαιο 6 αποφασίστηκε ότι το κλασικό μοντέλο LSTM είναι το μοντέλο μηχανικής μάθησης που θα χρησιμοποιηθεί για την ανάπτυξη του μηχανισμού που υλοποιήθηκε στα πλαίσια αυτής της διπλωματικής εργασίας.

Κεφάλαιο 7: Παρουσίαση του Μηχανισμού και Ανάλυση της Υλοποίησης

7.1 Προγραμματιστικά Εργαλεία

Πριν την αναλυτική παρουσίαση του συστήματος που υλοποιήθηκε στην παρούσα διπλωματική εργασία, κρίνεται απαραίτητο να προηγηθεί μια σύντομη επεξήγηση των προγραμματιστικών εργαλείων που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος. Σε αυτό το υποκεφάλαιο λοιπόν, παρουσιάζονται οι βασικές πλατφόρμες και τα λογισμικά που χρησιμοποιήθηκαν για την ανάπτυξη των συστατικών που απαρτίζουν το σύστημα.

7.1.1 MongoDB

Η MongoDB [93] είναι μια από τις πιο διαδεδομένες Document-oriented NoSQL βάσεις δεδομένων ανοιχτού κώδικα και υπάγεται στην ευρύτερη κατηγορία των CP βάσεων, οι οποίες περιγράφηκαν στο υποκεφάλαιο 3.1.3. Αποδεικνύεται ιδιαίτερα αποτελεσματική στη διαχείριση και την αποθήκευση μεγάλων δεδομένων, ενώ παράλληλα παρέχει υψηλή απόδοση, υψηλή διαθεσιμότητα, εύκολη κλιμάκωση και ταχύτατη ανάκτηση δεδομένων, γεγονός που ευνοεί την ανάκτηση δεδομένων σε πραγματικό χρόνο.

Πηγή: https://webassets.mongodb.com/_com_assets/cms/mongodb_logo1-76twgcu2dm.png

Σχήμα 7.1: MongoDB

7.1.1.1 Μοντέλο Δεδομένων

Η MongoDB χρησιμοποιεί το έγγραφο (document) ως τη βασική μονάδα αποθήκευσης δεδομένων. Το έγγραφο είναι ένα σύνολο από ζεύγη κλειδιού-τιμής το οποίο έχει δυναμικό σχήμα (dynamic schema). Αυτό σημαίνει ότι τα έγγραφα δεν ακολουθούν κάποια συγκεκριμένη δομή και επομένως μπορούν να περιέχουν τόσο διαφορετικό πλήθος όσο και διαφορετικό σύνολο πεδίων. Τα έγγραφα αποθηκεύονται στον δίσκο σε φόρμα σειριοποίησης BSON (Binary JSON), δηλαδή σαν μια δυαδική αναπαράσταση JSON αρχείων. Θα πρέπει να σημειωθεί ότι κάθε έγγραφο χαρακτηρίζεται από ένα κλειδί το οποίο συμβολίζεται ως id και χρησιμοποιείται ως το

μοναδικό αναγνωριστικό για το εκάστοτε έγγραφο. Οι τιμές που μπορεί να έχει ένα πεδίο ενός document δεν υπόκεινται σε κανέναν περιορισμό. Μπορεί να είναι μια τιμή, μια λίστα τιμών ή ακόμη και ένα εμφωλευμένο document.

Πηγή: https://docs.mongodb.com/manual/_images/crud-annotated-document.bakedsvg.svg

Σχήμα 7.2: Ένα έγγραφο

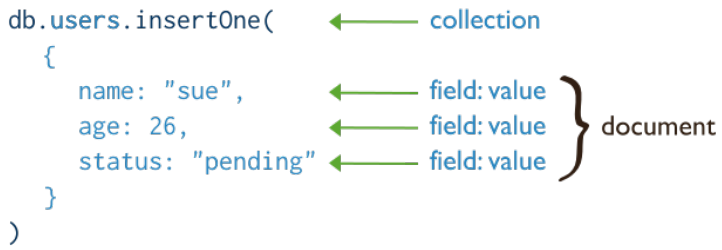
Ένα ακόμη βασικό συστατικό της MongoDB είναι η συλλογή (collection). Οι συλλογές ουσιαστικά είναι ένα σύνολο πολλών εγγράφων. Τα έγγραφα που ανήκουν στην ίδια συλλογή δεν απαιτείται να είναι όμοια, ωστόσο κάτι τέτοιο συνίσταται καθώς οι συλλογές προσφέρουν ακριβώς αυτή τη δυνατότητα ομαδοποίησης.

Πηγή: https://docs.mongodb.com/manual/_images/crud-annotated-collection.bakedsvg.svg

Σχήμα 7.3: Μια συλλογή από έγγραφα

7.1.1.2 Μοντέλο Ερωτημάτων

Η MongoDB υποστηρίζει τις απαραίτητες CRUD (create, read, update, delete) λειτουργίες για να αλληλεπιδρά με τα αποθηκευμένα δεδομένα. Για το σκοπό αυτό χρησιμοποιείται μια γλώσσα η οποία ονομάζεται MongoDB Query Language (MQL) και επιτρέπει στον χρήστη να ορίζει εντολές που θα εκτελεστούν. Η παραμετροποίηση των εντολών αυτών γίνεται με ένα φορματ που θυμίζει JSON αρχείο. Στη συνέχεια παρουσιάζονται ενδεικτικά κάποια παραδείγματα βασικών εντολών. Η εισαγωγή εγγράφων σε μια συλλογή πραγματοποιείται με τις εντολές *db.collection.insertOne*() για ένα έγγραφο και *db.collection.insertMany*() για περισσότερα έγγραφα.



Πηγή: https://docs.mongodb.com/manual/_images/crud-annotated-mongodb-insertOne.bakedsvg.svg

Σχήμα 7.4: Εισαγωγή εγγράφου

Η ανάγνωση και ανάκτηση εγγράφων από μια συλλογή πραγματοποιείται με την εντολή *db.collection.find*(). Η εντολή επιστρέφει όλα τα έγγραφα που ικανοποιούν την πρώτη παράμετρο (query). Η δεύτερη παράμετρος προσδιορίζει τα πεδία που θα επιστραφούν (projection).

Πηγή: https://docs.mongodb.com/manual/_images/crud-annotated-mongodb-find.bakedsvg.svg

Σχήμα 7.5: Ανάγνωση εγγράφων

Η ενημέρωση εγγράφων σε μια συλλογή πραγματοποιείται με τις εντολές *db.collection.updateOne*() ή *db.collection.replaceOne*() για ένα έγγραφο και *db.collection.updateMany*() για περισσότερα έγγραφα. Η πρώτη παράμετρος καθορίζει το σύνολο των εγγράφων που θα ενημερωθούν, ενώ η δεύτερη καθορίζει τα πεδία που θα ενημερωθούν.

Πηγή: https://docs.mongodb.com/manual/_images/crud-annotated-mongodb-updateMany.bakedsvg.svg

Σχήμα 7.6: Ενημέρωση εγγράφων

Τέλος, η διαγραφή εγγράφων σε μια συλλογή πραγματοποιείται με τις εντολές *db.collection.deleteOne*() για ένα έγγραφο και *db.collection.deleteMany*() για περισσότερα έγγραφα. Η παράμετρος καθορίζει το σύνολο των εγγράφων που θα διαγραφούν.

Πηγή: https://docs.mongodb.com/manual/_images/crud-annotated-mongodb-deleteMany.bakedsvg.svg

Σχήμα 7.7: Διαγραφή εγγράφων

7.1.1.3 Indexing

Ο όρος ευρετήριο (index) αναφέρεται σε μια δομή την οποία χρησιμοποιεί η βάση προκειμένου να επιταχύνει το χρόνο εκτέλεσης ενός ερωτήματος (query). Αυτό επιτυγχάνεται με τη διάταξη των δεδομένων σε συγκεκριμένη σειρά, έτσι ώστε κατά τη διαδικασία της αναζήτησης να μην είναι απαραίτητος ο έλεγχος όλων των δεδομένων.

Όπως αναφέρθηκε και παραπάνω κάθε έγγραφο χαρακτηρίζεται από ένα μοναδικό id. Το πεδίο αυτό είναι ένας default index που δημιουργεί η MongoDB κατά την εισαγωγή ενός εγγράφου. Η MongoDB υποστηρίζει κι άλλους indexes, όπως:

- Single Key Index: Index στο πεδίο ενός εγγράφου.

- Compound Index: Index σε ένα σύνολο από πεδία ενός εγγράφου.
- Multikey Index: Index που χρησιμοποιείται για πεδία εγγράφων των οποίων η τιμή είναι πίνακας. Για κάθε στοιχείο του πίνακα δημιουργείται ένα ξεχωριστό index.
- Geospatial Index: Index για πεδία που σχετίζονται με γεωγραφικά δεδομένα.
- Text Index: Index για πεδία τύπου string.
- Hashed Index: Index που χρησιμοποιείται στη τιμή κατακερματισμού (hash value) ενός πεδίου.

Μια επιπλέον κατηγοριοποίηση των indexes μπορεί να γίνει με βάση τις ιδιότητες που αυτά έχουν. Οι επιμέρους κατηγορίες είναι οι εξής:

- Unique Index: Το πεδίο στο οποίο εφαρμόζεται το index δεν μπορεί να έχει duplicate values. Για παράδειγμα το index που εφαρμόζεται στο id είναι unique.
- Partial Index: Index που χρησιμοποιείται σε κάποιο υποσύνολο εγγράφων μιας συλλογής τα οποία πληρούν κάποια συνθήκη που έχει οριστεί.
- Sparse Index: Η ιδιότητα αυτή εξασφαλίζει ότι το index περιέχει entries μόνο για έγγραφα που περιλαμβάνουν το πεδίο που έχει εφαρμοστεί το index.
- TTL Index: Χρησιμοποιείται έτσι ώστε το έγγραφο στο οποίο εφαρμόζεται το index να διαγράφεται αυτόματα μετά από ένα συγκεκριμένο χρονικό διάστημα. Η χρήση του TTL Index σε συνδυασμό με τις δυνατότητες συμπίεσης των δεδομένων που προσφέρει η MongoDB αποτελεί μια πολύ καλή πρακτική για δεδομένα που συλλέγονται σε πραγματικό χρόνο και έχουν μορφή ακολουθίας.
- Hidden Index: Ένα ινδεξ το οποίο δεν "φαίνεται" στον query optimizer.

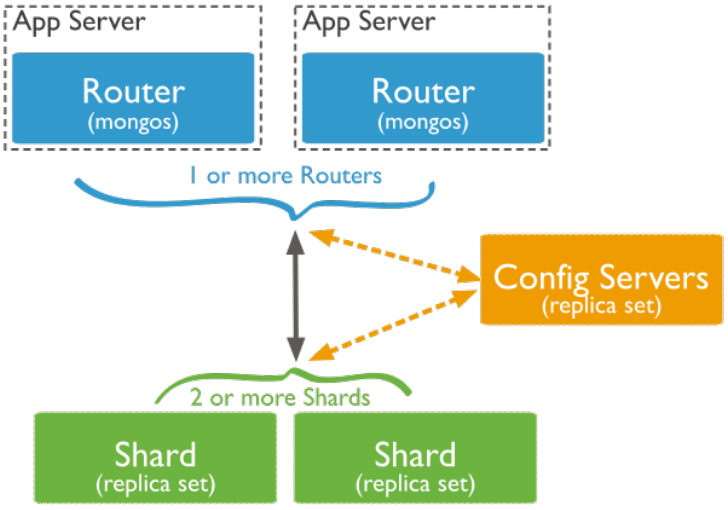
Αξίζει να αναφερθεί ότι η MongoDB διαθέτει έναν βελτιστοποιητή ερωτημάτων (query optimizer) προκειμένου να επιλέξει το πιο αποτελεσματικό πλάνο (query plan) ανάκτησης των ζητούμενων δεδομένων με βάση τα indexes που έχουν οριστεί στο σύστημα. Αυτό μπορεί να γίνει καλύτερα κατανοητό μέσω ενός παραδείγματος. Έστω μια βάση δεδομένων που περιέχει 100.000 έγγραφα και ζητούμενο είναι να ανακτηθούν τα έγγραφα στα οποία η τιμή του πεδίου "όνομα" είναι ίση με "Νίκος". Ένα πιθανό query plan είναι να ελεγχθούν και οι 100.000 εγγραφές προκειμένου να προσδιοριστούν αυτές που πληρούν τη συνθήκη. Ένα άλλο query plan είναι να ελεγχθεί εάν υπάρχει κάποιος index στο πεδίο "όνομα", γεγονός που θα επέτρεπε την άμεση ανάκτηση των ζητούμενων εγγραφών. Ο query optimizer αναλαμβάνει να βρει και ναι εκτελέσει το καλύτερο query plan. Στη συνέχεια, αποθηκεύει το βέλτιστο query plan για το συγκεκριμένο query σε μια μνήμη cache (query plans cache), έτσι ώστε την επόμενη φορά που θα γίνει το ίδιο ή κάποιο παρόμοιο query να ξέρει αμέσως το query plan που πρέπει να ακολουθήσει.

7.1.1.4 Sharding

Η MongoDB υποστηρίζει τη μέθοδο sharding. Πρόκειται για μια τεχνική μέσω της οποίας τα δεδομένα της βάσης κατανέμονται σε πολλαπλά μηχανήματα και συνεπώς εξασφαλίζεται η δυνατότητα οριζόντιας κλιμάκωσης (scalability) του συστήματος. Πολλά μηχανήματα μαζί συνιστούν ένα Sharded Cluster, ενώ διαφορετικά Sharded Clusters συνδέονται μεταξύ τους. Με βάση το ρόλο που έχει ο κάθε κόμβος στο cluster μπορεί να χαρακτηριστεί ως:

- Shard: Ένας shard κόμβος έχει αποθηκευμένο ένα υποσύνολο του ολικού συνόλου δεδομένων.
- Query Router (mongos): Ένας mongo κόμβος είναι υπεύθυνος για την αντιστοίχιση των ερωτημάτων που διατυπώνονται στη βάση προς τους κατάλληλους κόμβους shard (τους κόμβους που έχουν πληροφορία σχετική με το ερώτημα δηλαδή). Επίσης, φροντίζει να γυρίζει τα αποτελέσματα του ερωτήματος στο χρήστη ή την εφαρμογή που διατύπωσε το εκάστοτε ερώτημα.
- Configuration Server: Ένας configuration server αποθηκεύει πληροφορίες (metadata) δρομολόγησης που υποδεικνύουν ποια δεδομένα διατηρούνται σε κάθε shard.

Προκειμένου να γίνει ο διαμοιρασμός των δεδομένων στους shard κόμβους είναι απαραίτητη η χρήση ενός κλειδιού το οποίο ονομάζεται shard key. Το shard key μπορεί να αναφέρεται είτε σε ένα πεδίο για το οποίο υπάρχει single key index είτε σε ένα σύνολο πεδίων για το οποίο υπάρχει compound index. Η τιμή του shard key, το πεδίο στο οποίο αναφέρεται δηλαδή, είναι αυτή που καθορίζει τον τρόπο με τον οποίο θα "σπάσουν" τα δεδομένα σε μικρότερα κομμάτια (chunks) τα οποία στη συνέχεια θα διαμοιραστούν στους shard κόμβους.

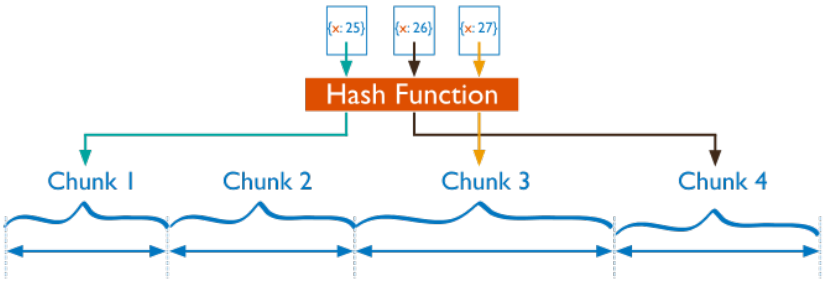


Πηγή: https://docs.mongodb.com/manual/_images/sharded-cluster-production-architecture.bakedsvg.svg

Σχήμα 7.8: Ένα Sharded Cluster

Η MongoDB προσφέρει δύο στρατηγικές sharding για το διαχωρισμό των δεδομένων σε επιμέρους chunks. Αυτές είναι οι εξής:

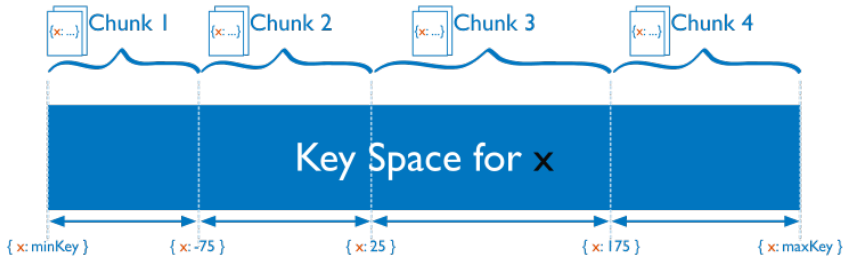
- Hashed Sharding: Το shard key τροφοδοτείται σε μια συνάρτηση κατακερματισμού (hash function) και η έξοδος καθορίζει το chunk στο οποίο θα τοποθετηθεί το εκάστοτε έγγραφο.



Πηγή: https://docs.mongodb.com/manual/_images/sharding-hash-based.bakedsvg.svg

Σχήμα 7.9: Hashed Sharding

- Ranged Sharding: Τα έγγραφα χωρίζονται σε διαστήματα με βάση την τιμή που έχει το εκάστοτε πεδίο το οποίο καθορίζει shard key.



Πηγή: https://docs.mongodb.com/manual/_images/sharding-range-based.bakedsvg.svg

Σχήμα 7.10: Ranged Sharding

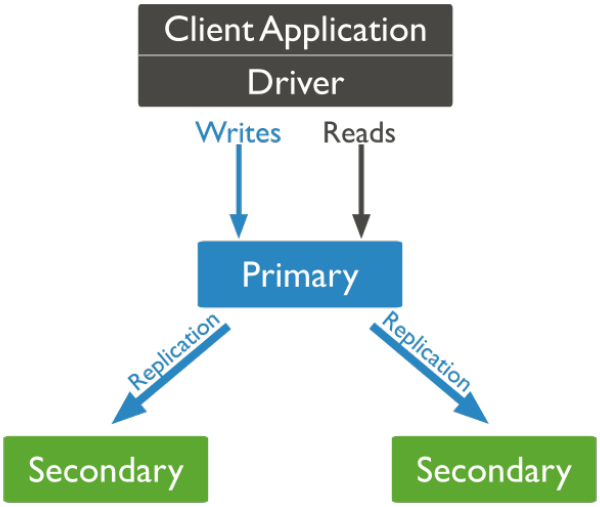
Θα πρέπει να αναφερθεί ότι και οι δύο στρατηγικές που περιγράφτηκαν παραπάνω μπορούν να οδηγήσουν σε άνιση κατανομή των δεδομένων στους shard κόμβους. Για αυτό τον λόγο υπάρχει μια διαδικασία που τρέχει στο παρασκήνιο (background process) και ονομάζεται balancer. Ο balancer παρακολουθεί τα chunks όπως αυτά εισάγονται στους κόμβους και σε όποια περίπτωση διακρίνει ότι υπάρχει άνιση κατανομή παίρνει chunks από τον πιο "φορτωμένο" κόμβο και τα μεταβιβάζει στον πιο "ελαφρύ" κόμβο.

7.1.1.5 Replication

Η MongoDB εξασφαλίζει υψηλή διαθεσιμότητα των δεδομένων και ανοχή στις βλάβες του δικτύου μέσω της διατήρησης αντιγράφων (replication). Αυτό επιτυγχάνεται με τη χρήση των replica sets. Ένα replica set αποτελείται από έναν πρωτεύων κόμβο shard (primary node) και από έναν ή περισσότερους δευτερεύοντες κόμβους shard (secondary node), οι οποίοι φροντίζουν ανά πάσα στιγμή να έχουν τα ίδια δεδομένα με τον πρωτεύων.

Προκειμένου να γίνει αυτό, ο primary node διατηρεί ένα log αρχείο, το οποίο ονομάζεται oplog, στο οποίο καταγράφει την δραστηριότητά του. Το αρχείο αυτό διαμοιράζεται σε όλους τους secondary nodes, οι οποίοι εκτελούν μία προς μία τις εντολές που αυτό περιέχει, έτσι ώστε να έρθουν στην ίδια κατάσταση με τον primary node. Για να λειτουργήσει αυτό θα πρέπει οι router nodes του cluster να κατευθύνουν

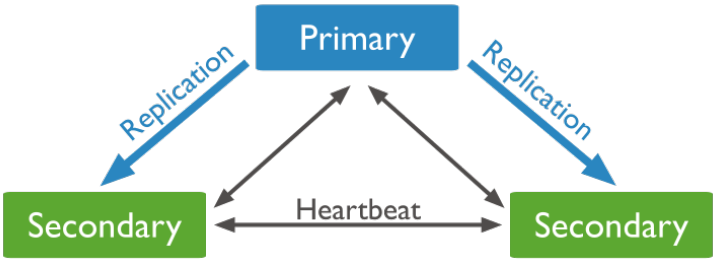
όλα τα αιτήματα εγγραφής στη βάση στον primary node. Τα αιτήματα που αφορούν ανάγνωση δεδομένων προτιμάται να πηγαίνουν προς τον primary node, έτσι ώστε να εξασφαλίζεται η συνέπεια, αλλά μπορούν να έχουν κατεύθυνση προς οποιοδήποτε κόμβο του replica set. Έτσι, εξασφαλίζεται η ταχύτερη ανάγνωση δεδομένων σε περίπτωση που ο primary node είναι υπερφορτωμένος. Ωστόσο, κάτι τέτοιο μπορεί να οδηγήσει σε ασυνέπεια καθώς μπορεί να ανακτηθούν δεδομένα από έναν secondary node, τα οποία δεν έχουν προλάβει να ενημερωθούν.



Πηγή: https://webassets.mongodb.com/_com_assets/cms/mongodb-replication-pnxoiu53rz.svg

Σχήμα 7.11: Replication

Θα πρέπει να αναφερθεί ότι οι κόμβοι ενός replica ανταλλάζουν μηνύματα μεταξύ τους ανά τακτικά χρονικά διαστήματα (heartbeat messages), προκειμένου να επιβεβαιώνουν ότι όλοι οι κόμβοι του replica set είναι "ζωντανοί". Εάν ο primary node αποτύχει (failover), ξεκινάει μια διαδικασία εκλογής νέου primary node η οποία πραγματοποιείται με κατανεμημένο τρόπο. Με αυτόν τον τρόπο εξασφαλίζεται κάθε στιγμή η ομαλή λειτουργία του replica set.



Πηγή: https://docs.mongodb.com/manual/_images/replica-set-primary-with-two-secondaries.bakedsvg.svg

Σχήμα 7.12: Ανταλλαγή μηνυμάτων Heartbeat

7.1.2 Apache Spark

Το Apache Spark [94] είναι μια ανοιχτού κώδικα πλατφόρμα γενικού σκοπού η οποία κατά κύριο λόγο χρησιμοποιείται για την ανάλυση και την επεξεργασία δεδομένων με κατανεμημένο τρόπο. Πρόκειται για ένα προγραμματιστικό μοντέλο το οποίο επιτρέπει την παράλληλη εκτέλεση αλγορίθμων σε συστάδες (clusters) υπολογιστικών συστημάτων. Το Apache Spark έχει σχεδιαστεί με τέτοιο τρόπο έτσι

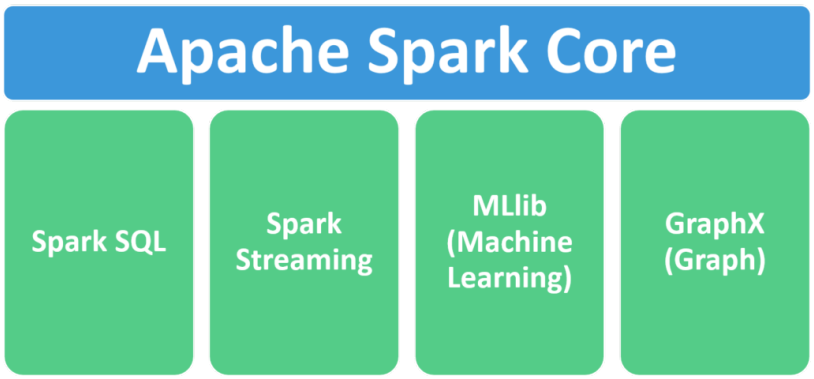
ώστε να ευνοεί την ταχύτατη ανάλυση δεδομένων μεγάλου όγκου. Αυτό επιτυγχάνεται λόγω της ικανότητας του να εκτελεί υπολογισμούς στη μνήμη (in-memory computing). Ακόμη, θα πρέπει να αναφερθεί ότι το Apache Spark παρέχει υψηλού επιπέδου APIs σε Java, Scala, Python και R καθώς και πλήθος ενσωματωμένων βιβλιοθηκών, γεγονός που το καθιστά ένα από τα δημοφιλέστερα εργαλεία σε εφαρμογές διαχείρισης και ανάλυσης μεγάλων δεδομένων.

Πηγή: https://upload.wikimedia.org/wikipedia/commons/thumb/f/f3/Apache_Spark_logo.svg/1280px-Apache_Spark_logo.svg.png

Σχήμα 7.13: Apache Spark

7.1.2.1 Δομή του Apache Spark

Όπως φαίνεται και από το σχήμα 7.14, η δομή του Apache Spark μπορεί να περιγραφεί ως μια στοίβα στη κορυφή της οποίας βρίσκονται διαφορές βιβλιοθήκες ενώ στη βάση της βρίσκεται ο πυρήνας Spark (Spark Core).



Πηγή: https://chartio.com/assets/a808a7/tutorials/what-is-spark/apache-spark-components.png

Σχήμα 7.14: Δομή του Apache Spark

Στο παρόν υποκεφάλαιο θα γίνει μια σύντομη περιγραφή των επιμέρους συστατικών του Apache Spark. Αναλυτικότερα:

- Spark Core: Το Spark Core είναι η θεμελιώδης μονάδα της πλατφόρμας Apache Spark και στεγάζει όλη τη λειτουργικότητα του framework. Πιο συγκεκριμένα, παρέχει λειτουργίες όπως κατανεμημένη αποστολή εργασιών, χρονοδρομολόγηση εργασιών, διαχείριση μνήμης και in-memory computing, αλληλεπίδραση με συστήματα αποθήκευσης κ.α. . Επιπλέον, στο Spark Core ορίζεται το Resilient Distributed Dataset (RDD) [95], το οποίο αποτελεί τη δομική οντότητα αναπαράστασης δεδομένων του Apache Spark.

- Spark SQL: Το Spark SQL είναι ένα από τα βασικότερα κομμάτια του Apache Spark και επιτρέπει τη διερεύνηση και την επεξεργασία δομημένων και ημι-δομημένων δεδομένων που είναι αποθηκευμένα τόσο σε RDD μορφή όσο και σε εξωτερικές πηγές. Ουσιαστικά, το Spark SQL επιτρέπει στο Spark να λειτουργεί σαν ένα κατανεμημένο σύστημα υποβολής ερωτημάτων SQL, χωρίς όμως να περιορίζεται σε σχεσιακές βάσεις δεδομένων καθώς υποστηρίζει πολλές πηγές δεδομένων. Το αποτέλεσμα αποθηκεύεται σε μια αφαιρετική μορφή δεδομένων η οποία ονομάζεται DataFrame και συνδέεται άμεσα με το RDD.

- Spark Streaming: Το Spark Streaming αποτελεί μια βιβλιοθήκη η οποία επιτρέπει την επεξεργασία ροών δεδομένων σε πραγματικό χρόνο. Τα δεδομένα αυτά μπορεί να προέρχονται από πολλές πηγές. Για την αναπαράσταση των ροών δεδομένων χρησιμοποιούνται τα RDD, γεγονός που επιτρέπει την ενσωμάτωση του Spark Streaming με οποιοδήποτε άλλο εργαλείο του Apache Spark (Spark SQL, MLlib). Το Spark Streaming λαμβάνει τις ροές δεδομένων σε πραγματικό χρόνο, διαχωρίζει τα δεδομένα σε μικρότερες ομάδες (batches), επεξεργάζεται τα batches δεδομένων και τελικά παράγει ένα αποτέλεσμα βάσει της επεξεργασίας αυτής. Ακόμη, θα πρέπει να αναφερθεί ότι το Spark Streaming έχει σχεδιαστεί έτσι ώστε να παρουσιάζει μεγάλη ανοχή σε σφάλματα και να προσφέρει την καλύτερη δυνατή εξισορρόπηση φορτίου στους διαθέσιμους υπολογιστικούς πόρους.

- MLlib: Το MLlib αποτελεί μια βιβλιοθήκη η οποία παρέχει αλγόριθμους μηχανικής μάθησης και έχει σχεδιαστεί έτσι ώστε να υποστηρίζει την κατανεμημένη εκτέλεση αυτών. Πέρα από τους κοινούς αλγορίθμους που αφορούν προβλήματα κατηγοριοποίησης, ομαδοποίησης και παλινδρόμησης, το MLlib προσφέρει βοηθητικές εφαρμογές σχετικές με την εξαγωγή, το μετασχηματισμό και την επιλογή των δεδομένων.

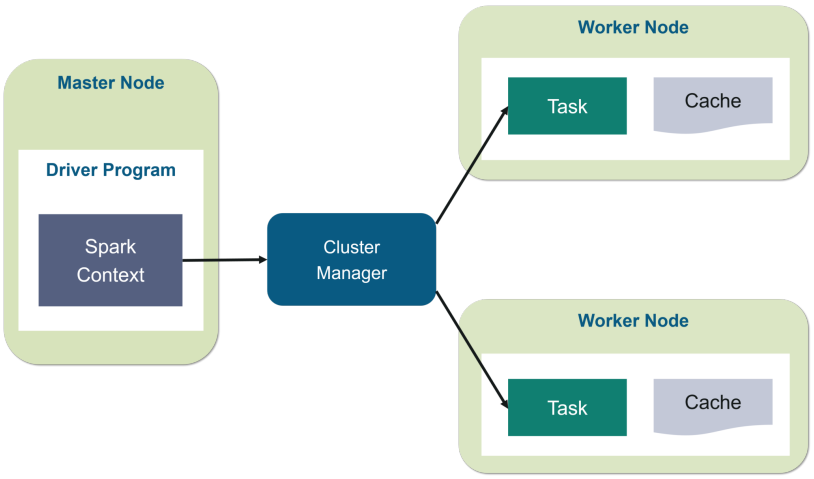
- GraphX: Το GraphX αποτελεί το κομμάτι του Spark που είναι υπεύθυνο για τη δημιουργία, την ανάλυση και την επεξεργασία γράφων με παράλληλο και κατανεμημένο τρόπο. Όπως στο Spark Streaming και στο Spark SQL, έτσι και στο GraphX δομικό στοιχείο αναπαράστασης των δεδομένων αποτελεί το RDD.

7.1.2.2 Αρχιτεκτονική του Apache Spark

Μια εφαρμογή του Spark μπορεί να εκτελεστεί είτε τοπικά (local mode) είτε κατανεμημένα (cluster mode). Η τοπική λειτουργία του Spark καθιστά εύκολη και γρήγορη τη δοκιμή εφαρμογών, καθώς δεν απαιτείται προηγούμενη διαμόρφωση του Spark έτσι ώστε να υποστηρίζεται η κατανεμημένη λειτουργία του προγράμματος της εφαρμογής. Ωστόσο, το Spark έχει σχεδιαστεί έτσι ώστε να επιτρέπει την παράλληλη εκτέλεση αλγορίθμων σε συστάδες (clusters) υπολογιστικών κόμβων. Αυτό επιτυγχάνεται με τη χρήση ενός διαχειριστή συμπλέγματος (cluster scheduler) ο οποίος είναι υπεύθυνος για τον προγραμματισμό και την κατανομή των πόρων σε όλους τους κόμβους μιας συστάδας. Το Spark προσφέρει τρεις διαχειριστές συμπλέγματος οι οποίοι είναι οι Standalone Scheduler, Hadoop YARN και Apache Mesos.

Στην περίπτωση του cluster mode, το Spark ακολουθεί την αρχιτεκτονική master/slave. Πιο συγκεκριμένα, ορίζεται ένα πρόγραμμα οδήγησης (driver) το οποίο "τρέχει" την κύρια διεργασία της εφαρμογής που υποβάλλεται στο Spark. Το πρόγραμμα οδήγησης είναι υπεύθυνο για τη μετατροπή της υποβληθείσας εφαρμογής σε

μικρότερες μονάδες εκτέλεσης (tasks) οι οποίες στη συνέχεια εκτελούνται από τους workers του cluster. Προκειμένου να επιτευχθεί αυτό, το πρόγραμμα οδήγησης ζητά executors από τον διαχειριστή συμπλέγματος. Ως executor ορίζεται μια διεργασία η οποία εκτελείται σε κάποιον κόμβο worker. Κάθε executor συνοδεύεται από έναν αριθμό επεξεργαστών και μια ποσότητα μνήμης, τα οποία δεσμεύονται από τον διαχειριστή συμπλέγματος. Στη συνέχεια, ο διαχειριστής συμπλέγματος αναλαμβάνει να διαμοιράσει και να εκκινήσει τους executors στους workers του cluster. Στο σχήμα 7.15 απεικονίζεται η αρχιτεκτονική που ακολουθεί το Spark όταν τρέχει σε κατανεμημένη λειτουργία.



Πηγή: https://www.edureka.co/blog/wp-content/uploads/2018/09/Picture6-2.png

Σχήμα 7.15: Αρχιτεκτονική του Apache Spark

7.1.3 Apache Kafka

Το Apache Kafka [96] είναι μια ανοιχτού κώδικα πλατφόρμα επεξεργασίας ροών δεδομένων, η οποία έχει σχεδιαστεί έτσι ώστε να λειτουργεί με κατανεμημένο τρόπο και να διαχειρίζεται δεδομένα που χαρακτηρίζονται από μεγάλο όγκο. Πιο συγκεκριμένα, πρόκειται για ένα σύστημα μεταφοράς δεδομένων που έχουν τη μορφή μηνυμάτων από μια πηγή σε πολλαπλές ομάδες παραληπτών. Αξίζει να αναφερθεί το γεγονός ότι το Apache Kafka έχει σχεδιαστεί έτσι ώστε να παρέχει υψηλή διαθεσιμότητα δεδομένων, ανοχή σε σφάλματα, αποθήκευση παλιών δεδομένων, συμπίεση δεδομένων καθώς και εύκολη κλιμάκωση.

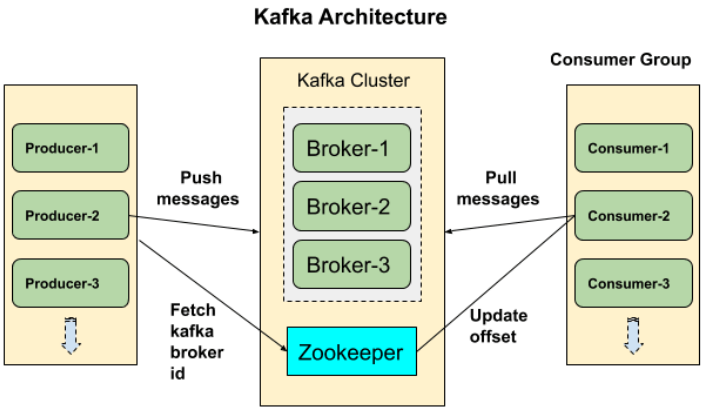
Πηγή: https://miro.medium.com/max/702/1*D0Xk8hmBB3zuyhGl1TDu3A.png

Σχήμα 7.16: Apache Kafka

7.1.3.1 Αρχιτεκτονική του Apache Kafka

Προκειμένου να γίνει πλήρως κατανοητή η λειτουργία του Apache Kafka, θα πρέπει πρώτα να γίνει μια συνοπτική παρουσίαση των βασικών συστατικών που το απαρτίζουν. Αυτά είναι τα εξής:

- Θέμα (Topic): Πρόκειται για ομαδοποιημένες ροές δεδομένων οι οποίες αναφέρονται σε μια συγκεκριμένη κατηγορία, βάσει της οποίας γίνεται ομαδοποίηση του συνόλου δεδομένων σε μικρότερες ομάδες. Τα διάφορα θέματα τα οποία εξυπηρετούνται από το Kafka cluster διαμερίζονται σε επιμέρους partitions.
- Παραγωγός (Producer): Πρόκειται για εφαρμογές οι οποίες δημοσιεύουν μηνύματα σε ένα η περισσότερα topics τα οποία εξυπηρετεί το Kafka cluster.
- Μεσίτης (Broker): Όπως έχει αναφερθεί το Kafka έχει σχεδιαστεί έτσι ώστε να λειτουργεί σε clusters κόμβων. Ο κάθε κόμβος του Kafka cluster ονομάζεται μεσίτης και χαρακτηρίζεται από ένα μοναδικό αναγνωριστικό (id).
- Καταναλωτής (Consumer): Πρόκειται για εφαρμογές οι οποίες διαβάζουν μηνύματα από τους μεσίτες για κάποιο συγκεκριμένο topic. Προκειμένου να πραγματοποιηθεί το διάβασμα των μηνυμάτων επιτυχώς, θα πρέπει πρώτα να έχει προηγηθεί η εγγραφή (subscribe) των καταναλωτών στο συγκεκριμένο θέμα.



Πηγή: https://programmertoday.com/wp-content/uploads/2019/12/Kafka-architecture.png

Σχήμα 7.17: Αρχιτεκτονική του Apache Kafka

Από την αρχιτεκτονική του Apache Kafka, όπως αυτή παρουσιάζεται στο σχήμα 7.17, γίνεται πλήρως αντιληπτό ότι το κατανεμημένο σύστημα διατηρεί δεδομένα ομαδοποιημένα σε topics και δίνει τη δυνατότητα σε πολλαπλούς παραγωγούς να εισάγουν μηνύματα σε ένα ή περισσότερα από αυτά και σε πολλαπλούς καταναλωτές να κάνουν εγγραφή σε ένα ή περισσότερα topics και να διαβάζουν τα αντίστοιχα μηνύματα. Κλείνοντας, θα πρέπει να αναφερθεί ότι η ορθή λειτουργία του Apache Kafka στηρίζεται στη λειτουργία του Apache Zookeeper [97]. Το Apache Zookeeper είναι μια κατανεμημένη υπηρεσία που παρέχει υπηρεσίες σχετικές με το συντονισμό και τον συγχρονισμό κατανεμημένων συστημάτων και χρησιμοποιείται από το Apache Kafka προκειμένου να εξασφαλίζεται η ομαλή λειτουργία των κόμβων που ανήκουν στο Kafka cluster.

7.1.4 PostgreSQL

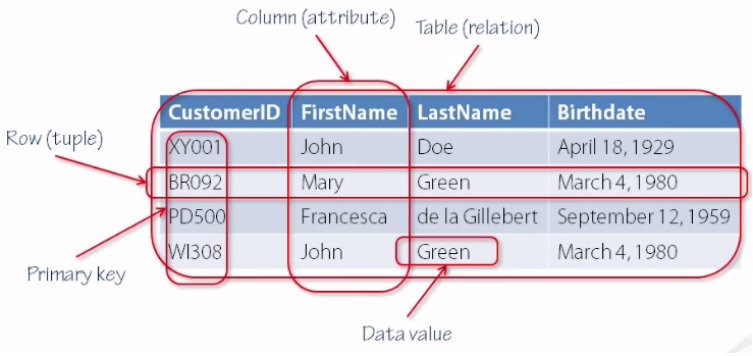
Το PostgreSQL [98] είναι ένα από τα δημοφιλέστερα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων το οποίο είναι ανοιχτού κώδικα. Πρόκειται για μια βάση δεδομένων η οποία είναι πλήρως συμβατή με το πρότυπο ACID (όπως αυτό περιγράφτηκε στο υποκεφάλαιο 3.1.3) και είναι ευρέως διαδεδομένη καθώς έχει σχεδιαστεί έτσι ώστε να παρέχει αξιοπιστία, ακεραιότητα δεδομένων, ταχύτατη αποθήκευσή και ανάκτηση δεδομένων καθώς και εύκολη ανάλυση των δεδομένων μέσω της γλώσσας SQL.

Πηγή: https://miro.medium.com/max/765/0*WoBbI7tHbFkaXbsj.png

Σχήμα 7.18: PostgreSQL

7.1.4.1 Μοντέλο Δεδομένων

Η PostgreSQL είναι μια σχεσιακή βάση δεδομένων και επομένως ακολουθεί το σχεσιακό μοντέλο δεδομένων (Relational Model) [99]. Σύμφωνα με αυτό, μια βάση δεδομένων μπορεί να αναπαρασταθεί ως ένα σύνολο σχέσεων (relations) που ορίζονται μεταξύ των δεδομένων. Οι σχέσεις αυτές εκφράζονται με τη μορφή πινάκων (tables) οι οποίοι αναπαριστούν οντότητες και τις συσχετίσεις μεταξύ αυτών. Κάθε πίνακας αποτελείται από γραμμές (rows) και από στήλες (columns). Η κάθε στήλη του πίνακα χαρακτηρίζει κάποια ιδιότητα της οντότητας και αποκαλείται χαρακτηριστικό (attribute) ή πεδίο (field), ενώ η κάθε γραμμή περιέχει όλες τις πληροφορίες που αφορούν ένα στοιχείο (instance) της οντότητας που αναπαριστά ο εν λόγω πίνακας και αποκαλείται πλειάδα (tuple) ή εγγραφή (record). Συνήθως, για έναν πίνακα ορίζεται ένα χαρακτηριστικό το οποίο είναι μοναδικό για κάθε εγγραφή που αυτός περιέχει και αποκαλείται πρωτεύον κλειδί (primary key).

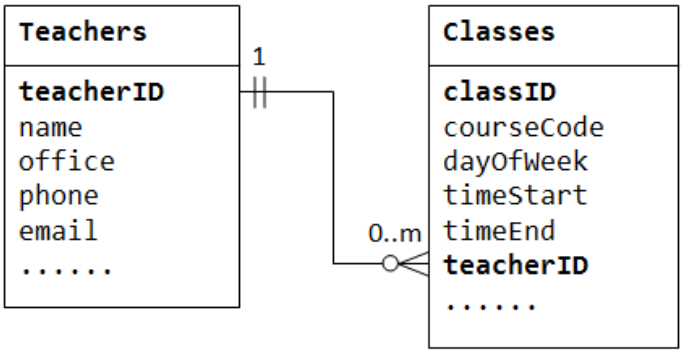


Πηγή: https://miro.medium.com/max/955/0*kBYg1f1lVSFE5cY6.PNG

Σχήμα 7.19: Η οντότητα πελάτης ως πίνακας

Ένας πίνακας που χρησιμοποιείται για να περιγράψει τη σχέση μεταξύ δύο ή περισσότερων οντοτήτων μιας βάσης δεδομένων, συνήθως, περιέχει τα πρωτεύοντα κλειδιά των πινάκων που ορίζουν τη σχέση, τα οποία σε αυτή τη περίπτωση ονομάζονται ξένα κλειδιά (foreign keys). Υπάρχουν τρία βασικά είδη σχέσεων μεταξύ των οντοτήτων μιας βάσης δεδομένων και αυτά είναι τα εξής:

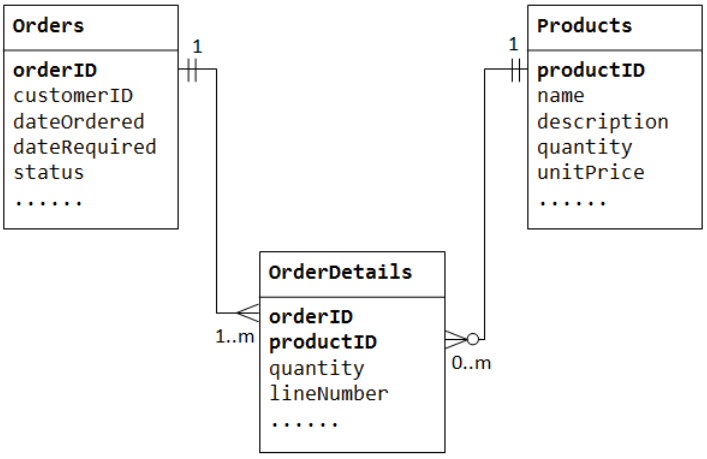
- Ένα-προς-πολλά (one-to-many): Σε αυτήν την περίπτωση, ένα στοιχείο μιας οντότητας αντιστοιχίζεται σε πολλά μιας άλλης οντότητας.



Πηγή: https://www3.ntu.edu.sg/home/ehchua/programming/sql/images/ManyToOne.png

Σχήμα 7.20: Η σχέση one-to-many

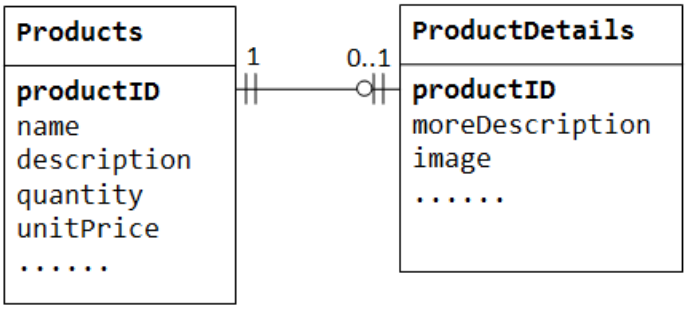
- Πολλά-προς-πολλά (many-to-many): Σε αυτήν την περίπτωση, πρέπει αναγκαστικά να οριστεί ένας επιπλέον πίνακας ο οποίος εκφράζει τη σχέση μεταξύ δύο ή περισσότερων οντοτήτων και μπορεί να αναφέρεται στο ίδιο στοιχείο κάποιας οντότητας περισσότερες από μία φορές.



Πηγή: https://www3.ntu.edu.sg/home/ehchua/programming/sql/images/ManyToMany.png

Σχήμα 7.21: Η σχέση many-to-many

- Ένα-προς-ένα (one-to-one): Σε αυτήν την περίπτωση, ένα στοιχείο μιας οντότητας αντιστοιχίζεται μοναδικά σε ένα στοιχείο μιας άλλης οντότητας.



Πηγή: https://www3.ntu.edu.sg/home/ehchua/programming/sql/images/OneToOne.png

Σχήμα 7.22: Η σχέση one-to-one

Το σύνολο όλων των πινάκων και των στηλών που απαρτίζουν μια βάση δεδομένων ονομάζεται σχήμα (schema) της βάσης και αρκεί για να περιγράψει το σύνολο των οντοτήτων και των σχέσεων που υπάρχουν μεταξύ αυτών.

Κλείνοντας, θα αναφερθούν οι βασικοί κανόνες που πρέπει να ακολουθεί μια σχεσιακή βάση δεδομένων. Αυτοί είναι οι εξής:

- Κάθε οντότητα πρέπει να παριστάνεται ως ένας ξεχωριστός πίνακας.
- Κάθε χαρακτηριστικό μιας οντότητας πρέπει να παριστάνεται από μια ξεχωριστή στήλη.
- Κάθε γραμμή ενός πίνακα πρέπει να είναι μοναδική (δεν γίνεται να υπάρχουν δύο ή περισσότερες γραμμές που περιέχουν ακριβώς τα ίδια χαρακτηριστικά).
- Το πρωτεύον κλειδί μιας οντότητας πρέπει πάντα να περιέχει τιμή, η οποία είναι ξεχωριστή από όλες τις άλλες.

7.1.4.2 Μοντέλο Δεδομένων

Η PostgreSQL υποστηρίζει τις απαραίτητες CRUD (create, read, update, delete) λειτουργίες για να αλληλεπιδρά με τα αποθηκευμένα δεδομένα. Για το σκοπό αυτό χρησιμοποιείται η γλώσσα SQL [99] η οποία επιτρέπει στον χρήστη να ορίζει εντολές που θα εκτελεστούν.

Υποθέτοντας ότι οι λέξεις που βρίσκονται μέσα σε εισαγωγικά είναι παράμετροι οριζόμενες από κάποιο χρήστη, οι βασικές λειτουργίες μιας βάσης δεδομένων, με τις αντίστοιχες SQL εντολές, είναι οι εξής:

- Δημιουργία Βάσης Δεδομένων:

  CREATE DATABASE <db_name>

- Διαγραφή Βάσης Δεδομένων:

  DROP DATABASE <db_name>

- Δημιουργία Πίνακα:

```
CREATE TABLE <table_name> (
    <column_name1> <datatype1>
    <column_name2> <datatype2>
    ...
)
```

- Διαγραφή Πίνακα:

```
DROP TABLE <table_name>
```

- Εισαγωγή Δεδομένων σε Πίνακα:

```
INSERT INTO <table_name> (<column_name1>, <column_name2>, ...
VALUES (<value1>, <value2>)
```

- Ενημέρωση Δεδομένων σε Πίνακα:

```
UPDATE <table_name>
SET <column_name1>=<value1>, <column_name2>=<value2>, ...
WHERE <condition>
```

- Διαγραφή Δεδομένων από Πίνακα:

```
DELETE FROM <table_name>
WHERE <condition>
```

- Ανάκτηση Δεδομένων:

```
SELECT <column_name1>, <column_name2>, ...
FROM <table_name>
WHERE <condition>
```

7.1.5 Flask (Web Framework)

Πηγή: https://upload.wikimedia.org/wikipedia/commons/thumb/3/3c/Flask_logo.svg/1200px-Flask_logo.svg.png

Σχήμα 7.23: Flask Web Framework

Το Flask [100] είναι ένα web framework το οποίο είναι γραμμένο σε python [74] και χρησιμοποιείται κυρίως για την υλοποίηση εφαρμογών στο διαδίκτυο. Πρόκειται για ένα από τα δημοφιλέστερα web frameworks καθώς είναι απλό στη χρήση, παρέχει μεγάλη ευελιξία στην υλοποίηση του κώδικα της εκάστοτε εφαρμογής και υποστηρίζει

πληθώρα επεκτάσεων που μπορούν να προσθέσουν επιπλέον λειτουργικότητα στην εφαρμογή προς ανάπτυξη. Ένα από τα βασικότερα χαρακτηριστικά του Flask είναι ότι εξυπηρετεί RESTful requests. Για τον λόγο αυτό χρησιμοποιείται κατά κόρον για την ανάπτυξη RESTful APIs, η λειτουργία των οποίων παρουσιάζεται παρακάτω.

7.1.5.1 RESTful API

Το REST (Representational State Transfer) [101] αποτελεί μια αρχιτεκτονική βάσει της οποίας ορίζεται ένα σύνολο κανόνων (API) που αφορούν την πρόσβαση και τη διαχείριση διαδικτυακών πόρων, οι οποίοι προσδιορίζονται μονοσήμαντα από τις διευθύνσεις URI (Uniform Resource Identifier) τους. Ουσιαστικά αφορά την επικοινωνία μεταξύ client και server. Ένα από τα βασικότερα πλεονεκτήματα της αρχιτεκτονικής REST είναι το γεγονός ότι έχει σχεδιαστεί έτσι ώστε να ταιριάζει με το πρωτόκολλο HTTP [102] και τις μεθόδους του, οι βασικότερες από τις οποίες είναι οι εξής:

- GET: Χρησιμοποιείται για τη λήψη δεδομένων.
- POST: Χρησιμοποιείται για την αποστολή δεδομένων.
- PUT: Χρησιμοποιείται για την αποστολή δεδομένων με σκοπό την ολική ή τη μερική αντικατάσταση όσων υπάρχουν ήδη.
- DELETE: Χρησιμοποιείται για τη διαγραφή δεδομένων.

Σε μια δικτυακή υπηρεσία REST, ο client στέλνει ένα αίτημα HTTP, με την αντίστοιχη μέθοδο, σε κάποιο συγκεκριμένο URI (endpoint). Είναι προφανές ότι αναλόγως με τη μέθοδο HTTP που έχει επιλεγεί μπορεί να πραγματοποιηθεί ανάκτηση πληροφοριών σχετικά με κάποιο διαδικτυακό πόρο (ο πόρος προσδιορίζεται από το αντίστοιχο URI), τροποποίηση των πληροφοριών που αφορούν τον πόρο ή ακόμη και διαγραφή του πόρου. Για κάθε αίτημα που λαμβάνεται, ο server επιστρέφει την αντίστοιχη απάντηση η οποία μπορεί να περιέχει τις πληροφορίες που ζητήθηκαν για κάποιον διαδικτυακό πόρο ή ότι πραγματοποιήθηκε με επιτυχία μια αλλαγή που αφορά κάποιον διαδικτυακό πόρο.

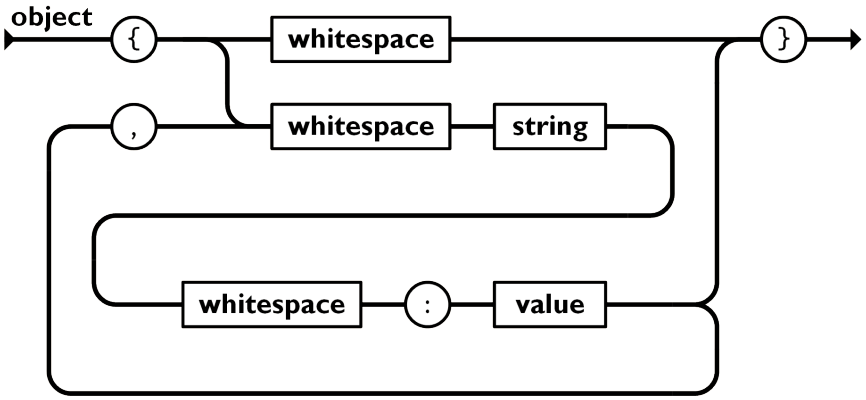
Συνεχίζοντας, θα πρέπει να αναφερθεί ότι οι υπηρεσίες που ακολουθούν την αρχιτεκτονική REST χαρακτηρίζονται ως stateless. Αυτό σημαίνει ότι κάθε αίτημα που πραγματοποιείται από κάποιον client προς τον server θα πρέπει να περιέχει όλη τη σχετική πληροφορία που χρειάζεται για την επιτυχή εκπλήρωση του αιτήματος. Ο server δεν μπορεί να αποθηκεύσει πληροφορίες που δίνονται από έναν client σε ένα παλαιότερο αίτημα του και να τις χρησιμοποιήσει σε κάποιο αίτημα που ο ίδιος client θα πραγματοποιήσει στο μέλλον.

Τέλος, ένα ακόμη χαρακτηριστικό των υπηρεσιών που υλοποιούν την αρχιτεκτονική REST είναι το γεγονός ότι η ανταλλαγή πληροφοριών μεταξύ του client και του server θα πρέπει να ακολουθεί κάποια συγκεκριμένη μορφή. Αν και υπάρχουν πολλές επιλογές, αυτή που χρησιμοποιείται σε μεγαλύτερο βαθμό είναι η μορφή JSON.

7.1.5.2 JSON

Το μορφότυπο JSON [103] (JavaScript Object Notation) αποτελεί έναν τρόπο ανταλλαγής δεδομένων σε μορφή ευανάγνωστη και εύκολα διαχειρίσιμη τόσο από τους

ανθρώπους όσο και από τους υπολογιστές. Πρόκειται ουσιαστικά, για ένα σύνολο συμβάσεων και απλών κανόνων, οι οποίοι είναι ανεξάρτητοι από τη γλώσσα προγραμματισμού που χρησιμοποιείται και ορίζουν τη μορφή που πρέπει να ακολουθούν τα δεδομένα προς μετάδοση. Η βασική κωδικοποίηση που χρησιμοποιείται στηρίζεται στην απεικόνιση της πληροφορίας που μεταφέρουν τα δεδομένα σε ζευγάρια κλειδιού-τιμής. Όλα τα ζευγάρια περιέχονται μέσα σε αγκύλες και είναι χωρισμένα με κόμμα, ενώ το κλειδί και η τιμή κάθε ζευγαριού διαχωρίζονται μεταξύ τους με άνω-κάτω τελεία. Το μορφότυπο JSON προσφέρει μεγάλη ευελιξία καθώς, ως τιμή μπορεί να τοποθετηθεί ένα string, ένας αριθμός, μια boolean τιμή, ένας πίνακας, ένα αντικείμενο JSON ή οποιοδήποτε άλλο αντικείμενο που μπορεί να εκφραστεί ως ένα ή περισσότερα ζευγάρια κλειδιού-τιμής. Τέλος, θα πρέπει να αναφερθεί ότι δίνεται η δυνατότητα αποστολής ενός πίνακα που περιέχει JSON αντικείμενα.



Πηγή: https://www.json.org/img/object.png

Σχήμα 7.24: Μορφότυπο JSON

7.1.6 Vue (Javascript)

Πηγή: https://upload.wikimedia.org/wikipedia/commons/thumb/9/95/Vue.js_Logo_2.svg/1200px-Vue.js_Logo_2.svg.png

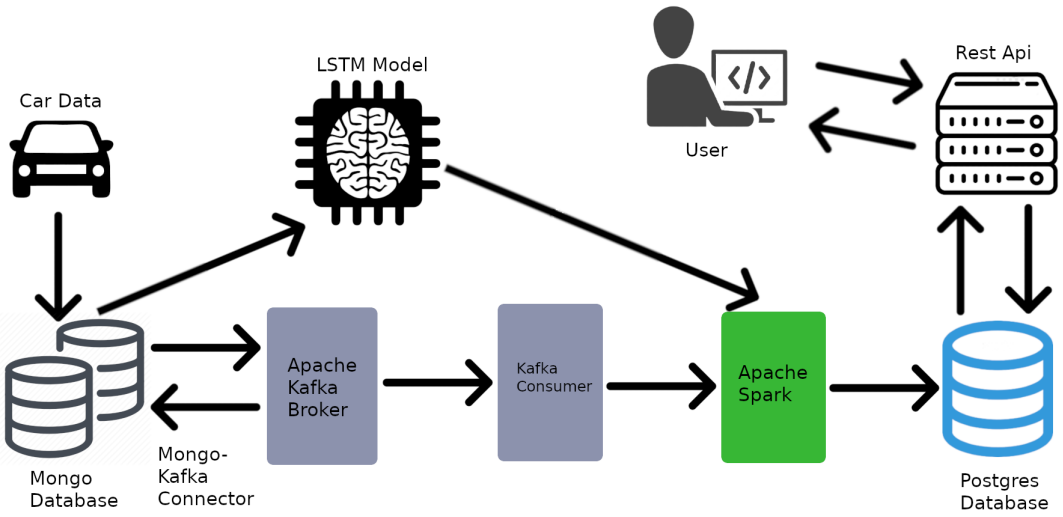
Σχήμα 7.25: Vue.js Framework

Το Vue.js [104] είναι ένα ανοιχτού κώδικα javascript framework το οποίο χρησιμοποιείται κυρίως για την ανάπτυξη προηγμένων διεπαφών χρήστη (user interfaces). Το Vue.js έχει υλοποιηθεί και στηρίζεται πλήρως από την

προγραμματιστική κοινότητα, σε αντίθεση με άλλα αντίστοιχα frameworks τα οποία έχουν κατασκευαστεί από μεγάλες εταιρίες, και πρόκειται για ένα από τα πιο δημοφιλή javascript frameworks καθώς συνδυάζει την ευκολία χρήσης με την ανάπτυξη ταχύτατων και "ελαφρών" (lightweight) εφαρμογών. Ένα από τα σημαντικότερα στοιχεία που εισήγαγε το Vue.js είναι τα συστατικά (components). Τα components βοηθούν τόσο στη συνολικότερη οργάνωση του κώδικα όσο και στη δημιουργία προσαρμοσμένων στοιχείων, τα οποία μπορούν να επαναχρησιμοποιηθούν σε HTML κώδικα. Συνεχίζοντας, το Vue.js υποστηρίζει αρκετές επεκτάσεις οι οποίες διευκολύνουν την ανάπτυξη μια εύχρηστης και πλήρους λειτουργικής εφαρμογής. Για παράδειγμα, η βιβλιοθήκη Vuetify [105] προσφέρει ένα σύνολο από components, που καλούνται material components, και η χρήση των οποίων μπορεί να εξοικονομήσει μεγάλο ποσοστό κώδικα HTML και CSS. Ακόμη, το Vue.js παρέχει ένα σύνολο ενσωματωμένων ενεργειών (directives), όπως v-if, v-for, v-bind, v-on, κ.α., οι οποίες μπορούν να χρησιμοποιηθούν άμεσα σε στοιχεία HTML και διευκολύνουν την εκτέλεση διάφορων ενεργειών στο frontend. Τέλος, θα πρέπει να αναφερθεί ότι το Vue.js χρησιμοποιείται πρωτίστως για την ανάπτυξη εφαρμογών μιας σελίδας (one-page applications). Για το σκοπό αυτό χρησιμοποιείται η βιβλιοθήκη vue-router, μέσω της οποίας υλοποιείται ο μηχανισμός που είναι υπεύθυνος για τη πλοήγηση μεταξύ των διάφορων σελίδων.

7.2 Παρουσίαση του Μηχανισμού

Ο μηχανισμός που υλοποιήθηκε στα πλαίσια της παρούσης διπλωματικής εργασίας έχει στόχο τη συλλογή δεδομένων από τις διαδρομές που πραγματοποιούν διάφοροι χρήστες, την ανάλυση και την επεξεργασία των δεδομένων αυτών με σκοπό την εξαγωγή συμπερασμάτων και την κατηγοριοποίηση της συμπεριφοράς των οδηγών των οχημάτων καθώς και την αναπαράσταση των αποτελεσμάτων με ευανάγνωστο τρόπο έτσι ώστε να ευνοείται η κατανόηση των αποτελεσμάτων από κάποιον άνθρωπο. Η αρχιτεκτονική διάταξη του μηχανισμού που υλοποιήθηκε παρουσιάζεται στο σχήμα 7.26.



Σχήμα 7.26: Η αρχιτεκτονική του μηχανισμού

Αναλυτικότερα, τα βασικά συστατικά που απαρτίζουν το σύστημα που υλοποιήθηκε είναι μια NoSQL βάση δεδομένων MongoDB, ένα νευρωνικό δίκτυο LSTM, ένας Apache Kafka Broker, ένας Apache Kafka Consumer, ένας αναλυτής δεδομένων που χρησιμοποιεί το Apache Spark, μια σχεσιακή βάση δεδομένων PostgreSQL και ένα REST API. Όλα αυτά τα συστατικά αποτελούν το backend του συστήματος που υλοποιήθηκε και θα παρουσιαστούν αναλυτικότερα στο υποκεφάλαιο 7.3. Ο σκοπός λειτουργίας του backend είναι η συλλογή δεδομένων που προκύπτουν από τις διαδρομές που πραγματοποιούν οι διάφοροι χρήστες, η εξαγωγή συμπερασμάτων που προκύπτουν από την ανάλυση των δεδομένων που έχουν συλλεχθεί και τελικά η αποθήκευση των νέων δεδομένων που παράγονται. Αρχικά, τα δεδομένα που συλλέγονται για τις διάφορες διαδρομές αποθηκεύονται σε μία βάση δεδομένων MongoDB. Η επιλογή μιας NoSQL βάσης κρίνεται απαραίτητη για την αποτελεσματική αποθήκευση των δεδομένων, καθώς πρόκειται για ροές δεδομένων μεγάλου όγκου. Παράλληλα, καθώς τα δεδομένα εισέρχονται στη βάση MongoDB στέλνονται σε έναν Apache Kafka Broker, όπου και τοποθετούνται σε ένα προκαθορισμένο topic. Τα δεδομένα που υπάρχουν στον Broker διαβάζονται από έναν Apache Kafka Consumer, ο οποίος φροντίζει για την ομαδοποίηση των δεδομένων ανά διαδρομή και την εφαρμογή της τεχνικής του κυλιόμενου παραθύρου πάνω σε αυτά. Στη συνέχεια, τα ομαδοποιημένα αυτά δεδομένα συλλέγονται από έναν αναλυτή δεδομένων, ο οποίος μέσω του Apache Spark εξασφαλίζει την παράλληλη προεπεξεργασία των δεδομένων με σκοπό το μετασχηματισμό τους σε μια μορφή η οποία θα είναι αξιοποιήσιμη από το μοντέλο LSTM. Έπειτα, κάνοντας χρήση του μοντέλου LSTM προβλέπει την κατηγορία στην οποία ανήκουν τα δεδομένα και παράγει νέα δεδομένα, βάσει αυτών των προβλέψεων, με σκοπό την κατηγοριοποίηση της οδηγικής συμπεριφοράς του εκάστοτε χρήστη την εκάστοτε χρονική στιγμή. Τα νέα αυτά δεδομένα αποθηκεύονται σε μια σχεσιακή βάση δεδομένων PostgreSQL με σκοπό την περαιτέρω ανάλυση τους και τον σχηματισμό ενός οδηγικού προφίλ για κάθε χρήστη του συστήματος. Τέλος, η ανάκτηση των δεδομένων που έχουν αποθηκευτεί στη σχεσιακή βάση δεδομένων γίνεται εύκολα μέσω ενός REST API που έχει υλοποιηθεί.

Τέλος, θα πρέπει να αναφερθεί ότι πέρα από το backend του συστήματος έχει υλοποιηθεί και ένα frontend το οποίο επιτρέπει την ανάλυση των δεδομένων που έχουν παραχθεί. Πιο συγκεκριμένα, το frontend επιτρέπει την παρακολούθηση των αποτελεσμάτων που παράγονται από το backend σε πραγματικό χρόνο, ενώ παράλληλα προσφέρει και τη δυνατότητα της ιστορικής αναδρομής των δεδομένων που έχουν αποθηκευτεί στο σύστημα. Για την ανάκτηση των δεδομένων από τη σχεσιακή βάση πραγματοποιούνται κλήσεις στα αντίστοιχα endpoints του REST API. Η αναλυτική παρουσίαση της λειτουργίας του frontend πραγματοποιείται στο υποκεφάλαιο 7.6. Σε αυτό το σημείο θα πρέπει να αναφερθεί ότι ο κώδικας που αναπτύχθηκε στο πλαίσιο υλοποιήσης του συστήματος (backend και frontend) μπορεί να βρεθεί στη διεύθυνση `https://gitlab.telecom.ntua.gr/thesis/serafeim_panagiotidis`.

7.3 Ανάλυση Υλοποίησης του Backend

Για την υλοποίηση του backend του συστήματος χρησιμοποιήθηκε η γλώσσα προγραμματισμού python [74] και πιο συγκεκριμένα οι βιβλιοθήκες pymongo [106], kafka-python [107], pyspark [108], sqlalchemy [109] και flask [100].

7.3.1 Βάση Δεδομένων MongoDB

Στο πλαίσιο υλοποίησης του backend, δημιουργήθηκε μια NoSQL βάση δεδομένων, η οποία ονομάζεται cars και είναι προσβάσιμη μέσω του connection string mongodb://localhost:27017. Η βάση cars αποτελείται από μια μόνο συλλογή, η οποία ονομάζεται routes. Στη συλλογή routes εισάγονται οδηγικά δεδομένα που παράγονται από όλες τις διαδρομές που πραγματοποιούνται και καταγράφονται από το σύστημα. Αξίζει να αναφερθεί ότι τα δεδομένα αυτά τοποθετούνται όλα μαζί, χωρίς να υπάρχει μέριμνα για την ομαδοποίηση των δεδομένων που αφορούν την ίδια διαδρομή, όπως θα γινόταν, δηλαδή, σε μια σχεσιακή βάση δεδομένων. Συνεχίζοντας, στο σχήμα 7.27 παρουσιάζεται η μορφή που έχει ένα έγγραφο που ανήκει στη συλλογή routes.


```
_id: ObjectId("604a06f0c733cd40f15cb66b")
AltitudeVariation: -2.2999878
VehicleSpeedInstantaneous: 25.67051888
VehicleSpeedAverage: 13.22350089
VehicleSpeedVariance: 121.5926897
VehicleSpeedVariation: -2.4769802
LongitudinalAcceleration: 0.3555
EngineLoad: 4.705882549
EngineCoolantTemperature: 68
ManifoldAbsolutePressure: 106
EngineRPM: 1796
MassAirFlow: 15.81000042
IntakeAirTemperature: 24
VerticalAcceleration: -0.1133
FuelConsumptionAverage: 19.49733543
roadSurface: "SmoothCondition"
traffic: "LowCongestionCondition"
drivingStyle: "EvenPaceStyle"
DriverId: 1
Timestamp: 60
RouteId: "f7d134e8fd59491db5b38dde92d09328"
```


Σχήμα 7.27: Οδηγικά δεδομένα στη MongoDB

Παρατηρώντας το σχήμα 7.27 φαίνεται ότι υπάρχουν τρία πεδία τα οποία δεν παρουσιάστηκαν στο υποκεφάλαιο 5.1. Τα πεδία αυτά είναι τα DriverId, Timestamp και RouteId. Τα πεδία αυτά δεν είναι αυθεντικά πεδία του συνόλου δεδομένων που χρησιμοποιήθηκε, αλλά αποτελούν πεδία που προστέθηκαν χειροκίνητα καθώς κρίθηκαν απαραίτητα για τη διεκπεραίωση της παρούσης διπλωματικής εργασίας. Πιο συγκεκριμένα, το DriverId αποτελεί ένα μοναδικό αναγνωριστικό του οδηγού το οχήματος, το RoutedId αποτελεί ένα μοναδικό αναγνωριστικό της διαδρομής που πραγματοποιείται και το Timestamp αποτελεί τη χρονική στιγμή, έπειτα από την εκκίνηση του εκάστοτε οχήματος, κατά την οποία συλλέχθηκαν τα δεδομένα που αφορούν μια συγκεκριμένη διαδρομή. Η χρησιμότητα του πεδίου RoutedId έγκειται στο γεγονός ότι επιτρέπει την ανάκτηση όλων των εγγράφων που αφορούν μια συγκεκριμένη διαδρομή. Επιπλέον, με χρήση του πεδίου Timestamp είναι δυνατή η ταξινομημένη ανάκτηση των εγγράφων μιας διαδρομής, γεγονός που εξασφαλίζει τη σωστή χρονολογική σειρά των δεδομένων που έχουν καταγραφεί. Τέλος, έγινε η υπόθεση ότι όλα τα δεδομένα συλλέγονται για χρήστες που έχουν δώσει τη συναίνεση τους και έχουν προηγουμένως εγγραφτεί στο σύστημα. Για τον λόγο αυτό χρησιμοποιήθηκε το πεδίο DriverId, το οποίο θεωρητικά αποδίδεται σε κάποιον χρήστη κατά την εγγραφή του στο σύστημα με σκοπό την αξιολόγηση της οδηγικής του συμπεριφοράς.

Κλείνοντας, θα πρέπει να αναφερθεί ότι η βάση δεδομένων cars έχει υλοποιηθεί ως ένα pseudo replica set που αποτελείται από δύο κόμβους. Αυτό σημαίνει ότι ενώ στην πραγματικότητα η βάση cars αποτελείται από έναν κόμβο, τα υπόλοιπα συστατικά του συστήματος που επικοινωνούν μαζί της την βλέπουν σαν ένα εικονικό replica set. Η δημιουργία του pseudo replica set ήταν απαραίτητη προκειμένου να επιτευχθεί η επικοινωνία με το Apache Kafka.

7.3.2 Apache Kafka Broker

Το Apache Kafka χρησιμοποιήθηκε για τη δημιουργία ενός μεσίτη (broker). Η προσθήκη ενός μεσίτη στο σύστημα έγινε λόγω της λειτουργικότητας που παρέχει το MongoDB Kafka Connector [110]. Το MongoDB Kafka Connector είναι ένα λογισμικό γραμμένο σε Java, το οποίο επιτρέπει την ανίχνευση των εγγράφων που εισάγονται σε μια βάση MongoDB και την περαιτέρω αποστολή τους σε ένα μεσίτη με αυτόματο τρόπο. Συνεπώς, η βάση MongoDB λειτουργεί έμμεσα σαν Kafka Producer και γράφει τα δεδομένα που εισάγονται σε αυτήν στον μεσίτη του συστήματος. Είναι προφανές ότι με αυτόν τον τρόπο διασύνδεσης επιτυγχάνεται με μεγάλη ευκολία η μεταφορά των δεδομένων κατά μήκος του συστήματος. Προκειμένου να γίνει χρήση του MongoDB Kafka Connector, κατά την εκκίνηση του μεσίτη θα πρέπει να προσδιοριστεί ως παράμετρος ένα αρχείο που σχετίζεται με τη λειτουργία του MongoDB Kafka Connector και το οποίο παρουσιάζεται στο σχήμα 7.28.

```
bootstrap.servers=localhost:9092
plugin.path=/usr/local/share/java,/opt/kafka/plugins
name=mongo-source
connector.class=com.mongodb.kafka.connect.MongoSourceConnector
tasks.max=1

# Connection and source configuration
connection.uri=mongodb://localhost:27017
database=cars
collection=routes

key.converter=org.apache.kafka.connect.json.JsonConverter
key.converter.schemas.enable=false
value.converter=org.apache.kafka.connect.json.JsonConverter
value.converter.schemas.enable=false
publish.full.document.only=true
offset.storage.file.filename=/tmp/connect.offsets

topic.prefix=routes
# topic.suffix=
poll.max.batch.size=10000
poll.await.time.ms=5000

# Change stream options
pipeline=[{"$match": {"operationType": "insert"}}]
batch.size=0
change.stream.full.document=updateLookup
```

Σχήμα 7.28: Παραμετροποίηση του MongoDB Kafka Connector

Παρατηρώντας το σχήμα 7.28, οι πιο σημαντικές παράμετροι που ορίζονται είναι οι εξής:

- bootstrap.servers: Αυτή η παράμετρος προσδιορίζει τις διευθύνσεις των μεσιτών του Kafka cluster. Στα πλαίσια της παρούσης διπλωματικής εργασίας χρησιμοποιήθηκε μόνο ένας μεσίτης ο οποίος βρίσκεται στη διεύθυνση localhost:9092.

- plugin.path: Αυτή η παράμετρος προσδιορίζει την τοποθεσία στην οποία έχει τοποθετηθεί το MongoDB Kafka Connector.

- name, connector.class: Αυτές οι δύο παράμετροι δηλώνουν ότι θα χρησιμοποιηθεί το MongoDB Kafka Connector.

- connection.uri, database, collection: Αυτές οι τρεις παράμετροι προσδιορίζουν το connection string, το όνομα και τη συλλογή της βάσης δεδομένων από την οποία θα λαμβάνονται τα δεδομένα, αντίστοιχα.

- key.converter, value.converter: Αυτές οι δύο παράμετροι φροντίζουν έτσι ώστε τα δεδομένα που λαμβάνονται να αποθηκεύονται σε μορφή JSON.

- topic.prefix: Αυτή η παράμετρος προσδιορίζει το σύνολο χαρακτήρων το οποίο θα περιέχεται στην αρχή του ονόματος του topic που θα δημιουργηθεί. Εξ ορισμού, στο όνομα του topic περιέχεται το όνομα της βάσης και της συλλογής οι οποίες έχουν δηλωθεί. Συνεπώς, εάν ένας καταναλωτής θέλει να διαβάσει τα σχετικά δεδομένα θα πρέπει να κάνει εγγραφή στο topic routes.cars.routes.

- pipeline: Αυτή η παράμετρος προσδιορίζει ότι τα δεδομένα που θα γίνονται publish στον μεσίτη θα είναι δεδομένα που εισάγονται στη βάση.

- poll.await.time: Αυτή η παράμετρος προσδιορίζει το χρονικό διάστημα ανά το οποίο ο Kafka Broker θα ελέγχει αν έχουν εισαχθεί νέα δεδομένα στη βάση. Το χρονικό διάστημα αυτό ορίζεται ίσο με 5 δευτερόλεπτα.

- poll.max.batch.size: Αυτή η παράμετρος προσδιορίζει το μέγιστο πλήθος μηνυμάτων που μπορούν να διαβαστούν σε μία επανάληψη από το λογισμικό MongoDB Kafka Connector. Το μέγιστο πλήθος αυτό ορίζεται ίσο με 10000 εγγραφές.

Συνεπώς, η λειτουργία του Apache Kafka Broker, σε συνδυασμό με τη βάση MongoDB, εξασφαλίζει ότι ανά 5 δευτερόλεπτα νέα δεδομένα που εισάγονται στο σύστημα θα είναι εύκολα διαθέσιμα σε έναν καταναλωτή, καθώς το μόνο προαπαιτούμενο είναι να έχει κάνει εγγραφή στο αντίστοιχο topic.

7.3.3 Apache Kafka Consumer

Το Apache Kafka χρησιμοποιήθηκε για τη δημιουργία ενός καταναλωτή. Η προσθήκη ενός καταναλωτή στο σύστημα έγινε με γνώμονα την εύκολη ανάγνωση των δεδομένων από τον μεσίτη.

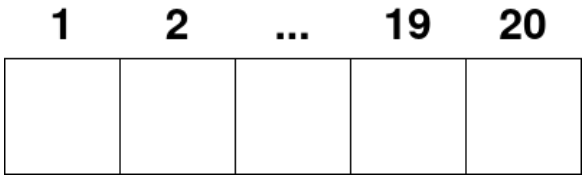
Ξεκινώντας, το πρώτο πράγμα που κάνει ο καταναλωτής είναι να ανοίξει ένα socket το οποίο ακούει στη διεύθυνση localhost:9999. Το socket ανοίγεται με σκοπό να εγκατασταθεί ένας δίαυλος επικοινωνίας με το επόμενο συστατικό του μηχανισμού, το οποίο είναι ο αναλυτής δεδομένων. Μέχρις ότου να συνδεθεί ο αναλυτής δεδομένων, ο καταναλωτής παραμένει αδρανής. Αυτό σημαίνει ότι προκειμένου να μπορεί ο καταναλωτής να διαβάσει δεδομένα από τον μεσίτη του συστήματος, θα πρέπει πρώτα να έχει εξασφαλιστεί η επιτυχής επικοινωνία με τον αναλυτή δεδομένων.

Όταν επιτευχθεί η σύνδεση μεταξύ αυτών των δύο συστατικών, ο καταναλωτής επικοινωνεί με τον μεσίτη και εγγράφεται στο topic routes.cars.routes. Στη συνέχεια, διαβάζει επαναληπτικά όλα τα διαθέσιμα μηνύματα που υπάρχουν στον μεσίτη. Σε κάθε επανάληψη διαβάσματος ο καταναλωτής φροντίζει να διαβάζει καινούρια δεδομένα, έτσι ώστε να μην υπάρχει πιθανότητα να διαβαστεί η ίδια εγγραφή δύο φορές. Τα δεδομένα που διαβάζει ο καταναλωτής από τον μεσίτη του συστήματος έχουν τη μορφή που παρουσιάζεται στο σχήμα 7.29.

```
{'AltitudeVariation': 1.0,
 'DriverId': 4,
 'EngineCoolantTemperature': 48,
 'EngineLoad': 34.11764908,
 'EngineRPM': 796.0,
 'FuelConsumptionAverage': 21.56131935,
 'IntakeAirTemperature': 18,
 'LongitudinalAcceleration': -0.322,
 'ManifoldAbsolutePressure': 102,
 'MassAirFlow': 5.380000114,
 'RouteId': '7ce0f07d33694c8e8bdef957fce0f169',
 'Timestamp': 225,
 'VehicleSpeedAverage': 1.604999928,
 'VehicleSpeedInstantaneous': 0.0,
 'VehicleSpeedVariance': 5.576414812,
 'VehicleSpeedVariation': 0.0,
 'VerticalAcceleration': -0.1986,
 '_id': {'$oid': '604fd33bfd72a69de8eb1d01'},
 'drivingStyle': 'EvenPaceStyle',
 'roadSurface': 'FullOfHolesCondition',
 'traffic': 'HighCongestionCondition'}
```


Σχήμα 7.29: Οδηγικά δεδομένα κατά την εισαγωγή τους στον καταναλωτή

Συνεχίζοντας, ο καταναλωτής είναι υπεύθυνος για την εφαρμογή του κυλιόμενου παραθύρου πάνω στα δεδομένα. Πρόκειται για μια πολύ σημαντική διαδικασία μέσω της οποίας τα δεδομένα θα αποκτήσουν τις διαστάσεις που επιτρέπουν την αποτελεσματική αξιοποίησή τους από το LSTM μοντέλο του συστήματος. Όπως έχει ήδη αναφερθεί στο υποκεφάλαιο 6.2.1, το μέγεθος παραθύρου έχει επιλεχτεί να είναι ίσο με 20. Επομένως, ο καταναλωτής καλείται, για κάθε πιθανή διαδρομή που πρέπει να διαχειριστεί, να οργανώνει τα δεδομένα σε εικοσάδες με την τεχνική του κυλιόμενου παραθύρου και με σεβασμό στη χρονολογική σειρά των δεδομένων της εκάστοτε διαδρομής. Για το σκοπό αυτό έχει οριστεί μια δομή δεδομένων τύπου dictionary στην οποία αποθηκεύονται τα μηνύματα που ο καταναλωτής διαβάζει από τον μεσίτη. Πιο συγκεκριμένα, ως κλειδιά αυτής της δομής ορίζονται οι τιμές του πεδίου RoutedId των δεδομένων, ενώ σε κάθε κλειδί αντιστοιχίζεται μια λίστα μηνυμάτων. Κάθε μήνυμα αυτής της λίστας έχει τη μορφή που παρουσιάστηκε στο σχήμα 7.29. Εάν κάποιο μήνυμα περιέχει έστω και μια null τιμή σε κάποιο πεδίο του, τότε ο καταναλωτής το αγνοεί και συνεχίζει με το επόμενο μήνυμα. Όταν το μέγεθος κάποιας λίστας γίνει ίσο με 20, τότε όλα τα μηνύματα που περιέχονται στη λίστα στέλνονται στον αναλυτή δεδομένων, μέσω του socket, με σκοπό την περαιτέρω ανάλυσή τους. Στη συνέχεια, προκειμένου να γίνει περισσότερο κατανοητό, παρουσιάζεται μέσω ενός παραδείγματος ο τρόπος με τον οποίο ενημερώνεται η λίστα μηνυμάτων που αφορά μια διαδρομή. Αρχικά, η λίστα μηνυμάτων είναι κενή, όπως φαίνεται και στο σχήμα 7.30.

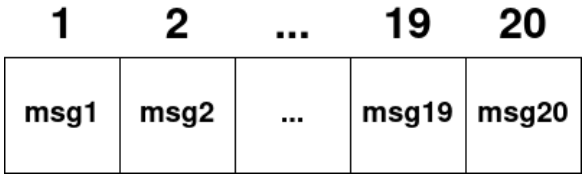


Σχήμα 7.30: Κενή λίστα μηνυμάτων

Έστω ότι στη συνέχεια καταφθάνει το πρώτο μήνυμα δεδομένων που αφορά τη διαδρομή της οποίας η λίστα παρουσιάζεται. Η λίστα μηνυμάτων θα έχει την εξής μορφή:

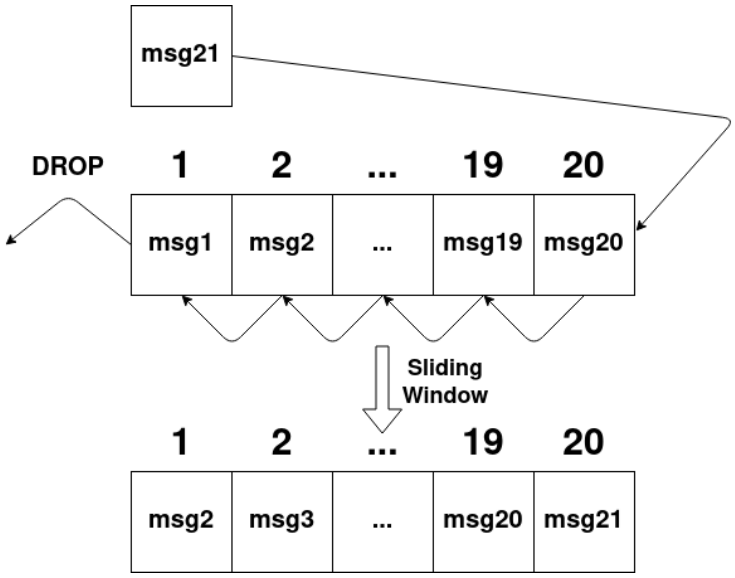
Σχήμα 7.31: Εισαγωγή πρώτου μηνύματος στη λίστα μηνυμάτων

Σε αυτή τη φάση δεν αποστέλλεται τίποτα μέσω του socket ακόμη, καθώς το μέγεθος της λίστας δεν είναι ίσο με 20. Έστω ότι καταφθάνουν 19 μηνύματα ακόμη. Τότε, η λίστα μηνυμάτων θα έχει την εξής μορφή:



Σχήμα 7.32: Γεμάτη λίστα μηνυμάτων

Σε αυτήν την περίπτωση το μήκος της λίστας είναι ίσο με 20, όποτε η λίστα μηνυμάτων αποστέλλεται μέσω του socket στον αναλυτή δεδομένων. Έστω ότι στη συνέχεια καταφθάνει το επόμενο μήνυμα δεδομένων που αφορά τη διαδρομή της οποίας η λίστα παρουσιάζεται. Σε αυτήν την περίπτωση, ο καταναλωτής θα πρέπει να εφαρμόσει την τεχνική του κυλιόμενου παραθύρου προκείμενου να εξασφαλίσει τη σωστή μορφή των δεδομένων που θα σταλούν αμέσως μετά. Επομένως, η λίστα μηνυμάτων θα έχει την εξής μορφή:



Σχήμα 7.33: Εφαρμογή κυλιόμενου παραθύρου στη λίστα μηνυμάτων

Από το σχήμα 7.33 φαίνεται ότι προκειμένου να εφαρμόσει την τεχνική του κυλιόμενου παραθύρου, ο καταναλωτής αφαιρεί το παλαιότερο μήνυμα της λίστας, μετατοπίζει όλα τα άλλα μηνύματα κατά μια θέση αριστερά και τέλος τοποθετεί το νέο μήνυμα στη τελευταία θέση της λίστας. Με αυτόν τον τρόπο εξασφαλίζεται ότι το μήνυμα msg21

θα φέρει μαζί του όλη την απαραίτητη πληροφορία προκειμένου να πραγματοποιηθεί η κατηγοριοποίησή του από το LSTM μοντέλο. Στη συνέχεια, η νέα λίστα που έχει σχηματιστεί αποστέλλεται μέσω του socket. Η διαδικασία που περιγράφηκε παραπάνω πραγματοποιείται για κάθε λίστα που υπάρχει στο dictionary του καταναλωτή. Είναι προφανές ότι κάθε διαδρομή έχει τη δική της λίστα. Προκειμένου ο καταναλωτής να καταλάβει σε ποια λίστα αντιστοιχεί κάποιο μήνυμα που έχει διαβάσει από τον μεσίτη κάνει χρήση του πεδίου RouteId του μηνύματος, έτσι ώστε να εντοπίσει τη σωστή λίστα από τη δομή dictionary που έχει οριστεί.

Συνεχίζοντας, η λίστα μηνυμάτων που στέλνεται από τον καταναλωτή προς τον αναλυτή δεδομένων έχει τη μορφή που παρουσιάζεται στο σχήμα 7.34. Θα πρέπει να αναφερθεί ότι το μέγεθος των δεδομένων δεν επιτρέπει την απεικόνιση ολόκληρης της λίστας σε μια μόνο εικόνα.


```
'{"4fa47c0b69ce45d789b419ddd6e327ca":[{"_id":{"$oid":"604fddfc48c0ec61295ad914"},"AltitudeVariation":-0.399993896,"Vehicl
eSpeedInstantaneous":36.89999771,"VehicleSpeedAverage":18.35999941,"VehicleSpeedVariance":165.9593783,"VehicleSpeedVariat
ion":0.0,"LongitudinalAcceleration":0.9774,"EngineLoad":41.96078491,"EngineCoolantTemperature":35.0,"ManifoldAbsolutePres
sure":101.0,"EngineRPM":1540.5,"MassAirFlow":15.72000027,"IntakeAirTemperature":16.0,"VerticalAcceleration":-0.3672,"Fuel
ConsumptionAverage":20.48775482,"roadSurface":"SmoothCondition","traffic":"LowCongestionCondition","drivingStyle":"EvenPa
ceStyle","DriverId":3,"Timestamp":91,"RouteId":"4fa47c0b69ce45d789b419ddd6e327ca"},{"_id":{"$oid":"604fddfd48c0ec61295ad9
15"},"AltitudeVariation":0.0,"VehicleSpeedInstantaneous":38.70000076,"VehicleSpeedAverage":19.00499942,"VehicleSpeedVaria
nce":166.8350491,"VehicleSpeedVariation":1.800003052,"LongitudinalAcceleration":0.8745,"EngineLoad":48.23529434,"EngineCo
olantTemperature":35.0,"ManifoldAbsolutePressure":102.0,"EngineRPM":1628.0,"MassAirFlow":16.71999931,"IntakeAirTemperatur
e":16.0,"VerticalAcceleration":-0.4905,"FuelConsumptionAverage":20.20194435,"roadSurface":"SmoothCondition","traffic":"Lo
wCongestionCondition","drivingStyle":"EvenPaceStyle","DriverId":3,"Timestamp":92,"RouteId":"4fa47c0b69ce45d789b419ddd6e32
7ca"},{"_id":{"$oid":"604fddfe48c0ec61295ad916"},"AltitudeVariation":-0.200004578,"VehicleSpeedInstantaneous":38.70000076
,"VehicleSpeedAverage":19.64999944,"VehicleSpeedVariance":166.8645673,"VehicleSpeedVariation":0.0,"LongitudinalAccelerati
on":1.0014,"EngineLoad":15.68627453,"EngineCoolantTemperature":35.0,"ManifoldAbsolutePressure":102.0,"EngineRPM":1628.0,"
MassAirFlow":16.71999931,"IntakeAirTemperature":16.0,"VerticalAcceleration":-0.5749,"FuelConsumptionAverage":20.20194435,
"roadSurface":"SmoothCondition","traffic":"LowCongestionCondition","drivingStyle":"EvenPaceStyle","DriverId":3,"Timestamp
":93,"RouteId":"4fa47c0b69ce45d789b419ddd6e327ca"},{"_id":{"$oid":"604fdff48c0ec61295ad917"},"AltitudeVariation":-0.5,"V
ehicleSpeedInstantaneous":39.59999847,"VehicleSpeedAverage":20.30999941,"VehicleSpeedVariance":166.6229399,"VehicleSpeedV
ariation":0.899997711,"LongitudinalAcceleration":1.1644,"EngineLoad":15.68627453,"EngineCoolantTemperature":35.0,"Manifol
dAbsolutePressure":102.0,"EngineRPM":1627.0,"MassAirFlow":20.86000061,"IntakeAirTemperature":16.0,"VerticalAcceleration":
-0.7365,"FuelConsumptionAverage":20.10399437,"roadSurface":"SmoothCondition","traffic":"LowCongestionCondition","drivingS
tyle":"EvenPaceStyle","DriverId":3,"Timestamp":94,"RouteId":"4fa47c0b69ce45d789b419ddd6e327ca"},{"_id":{"$oid":"604fde004
8c0ec61295ad918"},"AltitudeVariation":-0.400001526,"VehicleSpeedInstantaneous":38.70000076,"VehicleSpeedAverage":20.95499
942,"VehicleSpeedVariance":164.9404751,"VehicleSpeedVariation":-0.899997711,"LongitudinalAcceleration":1.2231,"EngineLoad
":40.39215851,"EngineCoolantTemperature":35.0,"ManifoldAbsolutePressure":102.0,"EngineRPM":1627.0,"MassAirFlow":20.860000
61,"IntakeAirTemperature":16.0,"VerticalAcceleration":-0.7791,"FuelConsumptionAverage":20.10399437,"roadSurface":"SmoothC
ondition","traffic":"LowCongestionCondition","drivingStyle":"EvenPaceStyle","DriverId":3,"Timestamp":95,"RouteId":"4fa47c
0b69ce45d789b419ddd6e327ca"},{"_id":{"$oid":"604fde0148c0ec61295ad919"},"AltitudeVariation":-0.400001526,"VehicleSpeedIns
tantaneous":37.79999924,"VehicleSpeedAverage":21.58499941,"VehicleSpeedVariance":161.9036618,"VehicleSpeedVariation":-0.9
00001526,"LongitudinalAcceleration":1.2792,"EngineLoad":66.27451324,"EngineCoolantTemperature":35.0,"ManifoldAbsolutePres
sure":102.0,"EngineRPM":1565.0,"MassAirFlow":18.44000053,"IntakeAirTemperature":16.0,"VerticalAcceleration":-0.8706,"Fuel
ConsumptionAverage":19.96433449,"roadSurface":"SmoothCondition","traffic":"LowCongestionCondition","drivingStyle":"EvenPa
ceStyle","DriverId":3,"Timestamp":96,"RouteId":"4fa47c0b69ce45d789b419ddd6e327ca"},{"_id":{"$oid":"604fde0248c0ec61295ad9
```


Σχήμα 7.34: Οδηγικά δεδομένα κατά την έξοδο τους από τον καταναλωτή

Παρατηρώντας το σχήμα 7.34 φαίνεται ότι ο καταναλωτής στέλνει στον αναλυτή δεδομένων ένα αντικείμενο JSON με κλειδί κάποιο RoutedId και τιμή έναν πίνακα από 20 αντικείμενα JSON που έχουν τη μορφή που παρουσιάστηκε στο σχήμα 7.29.

Συνεπώς, η λειτουργία του Apache Kafka Consumer εξασφαλίζει ότι τα δεδομένα μιας διαδρομής θα έχουν τις διαστάσεις και την δομή της εισόδου που αναμένει το LSTM μοντέλο προκειμένου να πραγματοποιήσει τις προβλέψεις του.

7.3.4 Νευρωνικό Δίκτυο LSTM

Το νευρωνικό δίκτυο LSTM χρησιμοποιείται για την εξόρυξη γνώσης από τα οδηγικά δεδομένα. Αναλυτικότερα, το μοντέλο LSTM εκπαιδεύεται με όλα τα δεδομένα που είναι αποθηκευμένα στη βάση δεδομένων MongoDB. Η διαδικασία της εκπαίδευσης του μοντέλου και της προεπεξεργασίας των δεδομένων είναι ίδια με αυτή που έχει παρουσιαστεί στο κεφάλαιο 6. Το μοντέλο δέχεται δεδομένα των οποίων οι διαστάσεις είναι ίδιες με αυτές των δεδομένων που παρουσιάστηκαν στο σχήμα 7.34, με την

επιπλέον προσθήκη ότι τα δεδομένα έχουν υποστεί προεπεξεργασία. Για κάθε τέτοια είσοδο το μοντέλο παράγει ως έξοδο μια ετικέτα ,η οποία έχει την τιμή AggressiveStyle ή την τιμή EvenPaceStyle, και η οποία αναφέρεται στη τελευταία εγγραφή της πολυδιάστατης χρονοσειράς που ορίζουν τα δεδομένα. Μόλις ολοκληρωθεί η εκπαίδευση, το μοντέλο αποθηκεύεται με σκοπό να χρησιμοποιηθεί γρήγορα και εύκολα από τον αναλυτή δεδομένων του συστήματος. Καθώς οι προβλέψεις του LSTM μοντέλου χρησιμοποιούνται από τον αναλυτή δεδομένων, η διασύνδεση του LSTM μοντέλου στον συνολικότερο μηχανισμό που έχει υλοποιηθεί θα φανεί καλύτερα στο υποκεφάλαιο 7.3.5, όπου παρουσιάζεται η λειτουργία του αναλυτή δεδομένων.

7.3.5 Apache Spark Αναλυτής Δεδομένων

Το Apache Spark χρησιμοποιήθηκε για τη δημιουργία ενός αναλυτή δεδομένων, ο οποίος αποτελεί ένα από τα σημαντικότερα συστατικά του συστήματος. Η προσθήκη ενός αναλυτή δεδομένων στο σύστημα έγινε με γνώμονα την ταχύτατη και παράλληλη διαχείριση του μεγάλου όγκου δεδομένων που καλείται να αντιμετωπίσει το σύστημα. Θα πρέπει να αναφερθεί ότι η υλοποίηση του συστήματος δεν περιλαμβάνει την κατανεμημένη λειτουργία του Apache Spark, καθώς ο αναλυτής δεδομένων έχει οριστεί να τρέχει σε έναν υπολογιστή και όχι σε ένα cluster κόμβων. Ωστόσο, το γεγονός ότι παρέχεται η δυνατότητα της οριζόντιας κλιμάκωσης του συστήματος, λόγω της ενσωμάτωσης του Apache Spark, είναι πολύ σημαντικό και σίγουρα ένα επιθυμητό χαρακτηριστικό για οποιαδήποτε εφαρμογή διαχείρισης δεδομένων μεγάλου όγκου.

Ξεκινώντας, το πρώτο πράγμα που κάνει ο αναλυτής δεδομένων είναι να συνδεθεί στο socket που έχει ανοίξει ο καταναλωτής του συστήματος. Μόλις επιτευχθεί η σύνδεση ο αναλυτής δεδομένων διαβάζει επαναληπτικά τα δεδομένα που του στέλνει ο καταναλωτής. Κάθε επανάληψη διαβάσματος συμβαίνει ανά 20 δευτερόλεπτα. Θα πρέπει να αναφερθεί ότι η σύνδεση του αναλυτή δεδομένων στο socket του καταναλωτή και η επαναληπτική διαδικασία διαβάσματος επιτυγχάνονται πολύ εύκολα με χρήση της Streaming βιβλιοθήκης που παρέχει το Apache Spark.

Τα δεδομένα που λαμβάνει ο αναλυτής δεδομένων όταν πραγματοποιεί μια ανάγνωση είναι ένα σύνολο εγγραφών, κάθε μια από τις οποίες έχει τη μορφή που παρουσιάστηκε στο σχήμα 7.34. Συνεπώς, η βασική μονάδα δεδομένων που καλείται να διαχειριστεί ο αναλυτής δεδομένων είναι μια πολυδιάστατη χρονοσειρά μήκους 20. Όπως αναφέρθηκε και παραπάνω, ο αναλυτής δεδομένων ελέγχει ανά 20 δευτερόλεπτα αν του έχουν σταλεί νέα δεδομένα από τον καταναλωτή του συστήματος. Αυτό σημαίνει ότι ο αναλυτής καλείται να διαχειριστεί ένα σύνολο πολυδιάστατων χρονοσειρών που έχουν σταλεί μέσα σε αυτό το χρονικό διάστημα των 20 δευτερολέπτων. Σε θεωρητικό επίπεδο, όπου δεν υπάρχουν καθυστερήσεις και η μετάδοση της πληροφορίας πραγματοποιείται ακαριαία, ο αναλυτής δεδομένων θα πρέπει σε κάθε ανάγνωση να διαβάζει 20 χρονοσειρές για κάθε διαδρομή που πραγματοποιείται εκείνη τη χρονική στιγμή. Ωστόσο, σε ένα ρεαλιστικό σύστημα κάτι τέτοιο είναι αρκετά απίθανο να συμβεί, όποτε και αναμένεται ότι για κάθε διαδρομή το πλήθος των χρονοσειρών που θα διαβάζεται θα είναι περίπου 20.

Συνεχίζοντας, ο αναλυτής δεδομένων εκτελεί την προεπεξεργασία δεδομένων που περιγράφτηκε στο κεφάλαιο 5, για κάθε εγγραφή του συνόλου που διάβασε από τον καταναλωτή του συστήματος. Πιο συγκεκριμένα, φροντίζει να κρατήσει τα πεδία που χρειάζονται για την πρόβλεψη του LSTM μοντέλου, καθώς και να κωδικοποιήσει τα

πεδία που πρέπει να κωδικοποιηθούν. Η μορφή των νέων εγγραφών που προκύπτουν παρουσιάζεται στο σχήμα 7.35. Θα πρέπει να αναφερθεί ότι το μέγεθος των δεδομένων δεν επιτρέπει την απεικόνιση ολόκληρης της εγγραφής, ωστόσο διευκρινίζεται ότι το μήκος της χρονοσειράς που αυτή περιέχει είναι ίσο με 20.

```
'ba35e8ed75e9476e80d846feb073ad83', 3, 79, array([[ 6.05999982e+00,  1.15840061e+02, -4.50000000e+00,
        1.05480000e+00,  3.80392151e+01,  3.10000000e+01,
        1.01000000e+02,  7.98000000e+02,  6.21999979e+00,
        1.50000000e+01, -5.98200000e-01,  2.79196968e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 6.34499983e+00,  1.17200816e+02, -3.59999847e+00,
        9.01200000e-01,  3.80392151e+01,  3.10000000e+01,
        1.01000000e+02,  7.97500000e+02,  6.26999998e+00,
        1.50000000e+01, -5.24400000e-01,  2.64583550e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 6.58499982e+00,  1.17559595e+02, -2.70000076e+00,
        9.33100000e-01,  9.05882340e+01,  3.20000000e+01,
        1.01000000e+02,  7.97500000e+02,  6.26999998e+00,
        1.50000000e+01, -5.06800000e-01,  2.64583550e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 6.82499981e+00,  1.17801222e+02,  0.00000000e+00,
        9.87800000e-01,  9.05882340e+01,  3.20000000e+01,
        1.01000000e+02,  7.26000000e+02,  8.63000011e+00,
        1.50000000e+01, -4.88200000e-01,  2.60041256e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 7.06499981e+00,  1.17925697e+02,  0.00000000e+00,
        1.13160000e+00,  9.29411774e+01,  3.20000000e+01,
        1.01000000e+02,  7.84000000e+02,  8.88000011e+00,
        1.50000000e+01, -6.71200000e-01,  2.52954979e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 7.33499979e+00,  1.18419934e+02,  1.79999924e+00,
        1.22100000e+00,  9.29411774e+01,  3.20000000e+01,
        1.01000000e+02,  7.84000000e+02,  8.88000011e+00,
        1.50000000e+01, -6.48700000e-01,  2.52954979e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 7.63499979e+00,  1.19344341e+02,  1.80000114e+00,
        1.16650000e+00,  3.13725495e+00,  3.20000000e+01,
```

Σχήμα 7.35: Προεπεξεργασία οδηγικών δεδομένων από τον αναλυτή δεδομένων

Παρατηρώντας το σχήμα 7.35 φαίνεται ο τρόπος με τον οποίο μετασχηματίζονται τα δεδομένα που έχουν τη μορφή που παρουσιάστηκε στο σχήμα 7.34. Το πρώτο πεδίο της εγγραφής αναφέρεται στο RouteId, το δεύτερο στο DriverId, το τρίτο στο Timestamp και το τέταρτο είναι ένας δισδιάστατος πίνακας ο οποίος αναπαριστά τη πολυδιάστατη χρονοσειρά.

Όπως έχει ήδη αναφερθεί ο αναλυτής δεδομένων δέχεται ένα σύνολο εγγραφών, τις οποίες προεπεξεργάζεται και τελικά τις φέρνει στη μορφή που παρουσιάστηκε στο σχήμα 7.35. Ωστόσο, οι εγγραφές αυτές μπορεί να έχουν διαφορετική τιμή για το πεδίο RoutedId, γεγονός που σημαίνει ότι αναφέρονται σε διαφορετική διαδρομή. Επομένως, η επόμενη ενέργεια που εκτελεί ο αναλυτής δεδομένων είναι η ομαδοποίηση των διάφορων εγγραφών που διάβασε με βάση το πεδίο RouteId. Έπειτα από την ομαδοποίηση, τα δεδομένα έχουν τη μορφή που παρουσιάζεται στις δύο παρακάτω εικόνες. Θα πρέπει να αναφερθεί ότι το μέγεθος των δεδομένων δεν επιτρέπει την απεικόνιση ολόκληρης της εγγραφής που προκύπτει.

```
'f65ceff30e554c549699ea430dd8f60b', [[3, 79, array([[ 6.05999982e+00,  1.15840061e+02, -4.50000000e+00,
        1.05480000e+00,  3.80392151e+01,  3.10000000e+01,
        1.01000000e+02,  7.98000000e+02,  6.21999979e+00,
        1.50000000e+01, -5.98200000e-01,  2.79196968e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 6.34499983e+00,  1.17200816e+02, -3.59999847e+00,
        9.01200000e-01,  3.80392151e+01,  3.10000000e+01,
        1.01000000e+02,  7.97500000e+02,  6.26999998e+00,
        1.50000000e+01, -5.24400000e-01,  2.64583550e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 6.58499982e+00,  1.17559595e+02, -2.70000076e+00,
        9.33100000e-01,  9.05882340e+01,  3.20000000e+01,
        1.01000000e+02,  7.97500000e+02,  6.26999998e+00,
        1.50000000e+01, -5.06800000e-01,  2.64583550e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 6.82499981e+00,  1.17801222e+02,  0.00000000e+00,
        9.87800000e-01,  9.05882340e+01,  3.20000000e+01,
        1.01000000e+02,  7.26000000e+02,  8.63000011e+00,
        1.50000000e+01, -4.88200000e-01,  2.60041256e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 7.06499981e+00,  1.17925697e+02,  0.00000000e+00,
        1.13160000e+00,  9.29411774e+01,  3.20000000e+01,
        1.01000000e+02,  7.84000000e+02,  8.88000011e+00,
        1.50000000e+01, -6.71200000e-01,  2.52954979e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 7.33499979e+00,  1.18419934e+02,  1.79999924e+00,
        1.22100000e+00,  9.29411774e+01,  3.20000000e+01,
        1.01000000e+02,  7.84000000e+02,  8.88000011e+00,
        1.50000000e+01, -6.48700000e-01,  2.52954979e+01,
        0.00000000e+00,  0.00000000e+00],
       [ 7.63499979e+00,  1.19344341e+02,  1.80000114e+00,
        1.16650000e+00,  3.13725495e+00,  3.20000000e+01,
```

(α′) Αρχή λίστας

```
  0.00000000e+00,  0.00000000e+00]])], [3, 83, array([[ 7.06499981e+00,  1.17925697e+02,  0.00000000e+00,
  1.13160000e+00,  9.29411774e+01,  3.20000000e+01,
  1.01000000e+02,  7.84000000e+02,  8.88000011e+00,
  1.50000000e+01, -6.71200000e-01,  2.52954979e+01,
  0.00000000e+00,  0.00000000e+00],
 [ 7.33499979e+00,  1.18419934e+02,  1.79999924e+00,
  1.22100000e+00,  9.29411774e+01,  3.20000000e+01,
  1.01000000e+02,  7.84000000e+02,  8.88000011e+00,
  1.50000000e+01, -6.48700000e-01,  2.52954979e+01,
  0.00000000e+00,  0.00000000e+00],
 [ 7.63499979e+00,  1.19344341e+02,  1.80000114e+00,
  1.16650000e+00,  3.13725495e+00,  3.20000000e+01,
  1.01000000e+02,  9.16000000e+02,  7.44000006e+00,
  1.50000000e+01, -5.96100000e-01,  2.42132855e+01,
  0.00000000e+00,  0.00000000e+00],
 [ 7.94999978e+00,  1.20406265e+02,  8.99999619e-01,
  1.30470000e+00,  3.76470604e+01,  3.20000000e+01,
  1.01000000e+02,  9.16000000e+02,  7.44000006e+00,
  1.50000000e+01, -6.95500000e-01,  2.42132855e+01,
  0.00000000e+00,  0.00000000e+00],
 [ 8.23499979e+00,  1.20671460e+02, -1.79999924e+00,
  1.44740000e+00,  3.76470604e+01,  3.20000000e+01,
  1.01000000e+02,  7.93500000e+02,  6.36000013e+00,
  1.50000000e+01, -8.11500000e-01,  2.33656254e+01,
  0.00000000e+00,  0.00000000e+00],
 [ 8.48999977e+00,  1.20301926e+02, -1.80000114e+00,
  1.58680000e+00,  3.80392151e+01,  3.20000000e+01,
  1.01000000e+02,  7.98000000e+02,  6.23999977e+00,
  1.50000000e+01, -9.46500000e-01,  2.32202797e+01,
  0.00000000e+00,  0.00000000e+00],
 [ 8.68499977e+00,  1.19216205e+02, -3.59999943e+00,
  1.64320000e+00,  3.80392151e+01,  3.20000000e+01,
  1.01000000e+02,  7.98000000e+02,  6.23999977e+00,
```

(β′) Τέλος λίστας

Σχήμα 7.36: Ομαδοποίηση δεδομένων

Παρατηρώντας τις δύο παραπάνω εικόνες σε συνδυασμό με τις όσες διευκρινήσεις έχουν δοθεί ως τώρα, φαίνεται ότι τα δεδομένα ομαδοποιούνται βάσει του RoutedId που αυτά φέρουν και στη συνέχεια τα δεδομένα που έχουν κοινό RoutedId τοποθετούνται σε μια λίστα. Στο παράδειγμα που παρουσιάζεται στις παραπάνω εικόνες φαίνεται ότι ο αναλυτής δεδομένων είχε διαβάσει από τον καταναλωτή του συστήματος πέντε εγγραφές που αφορούν τη διαδρομή με RoutedId

f65ceff30e554c549699ea430dd8f60b. Οι εγγραφές αυτές έχουν ομαδοποιηθεί σε μια μεγάλη λίστα, η οποία ουσιαστικά περιέχει τα δεδομένα που έχουν συλλεχθεί για τη διαδρομή μεταξύ των Timestamps 79 έως 83. Προφανώς για κάθε Timestamp από αυτά, τα δεδομένα έχουν τη μορφή πολυδιάστατης χρονοσειράς. Ακόμη, αξίζει να διευκρινιστεί ότι οι εγγραφές που διαβάζει ο αναλυτής δεδομένων από τον καταναλωτή δεν θα έχουν όλες την ίδια τιμή στο πεδίο RoutedId. Επομένως, μετά τη διαδικασία της ομαδοποίησης θα προκύψουν νέες εγγραφές, οι οποίες θα έχουν τη μορφή που παρουσιάστηκε στις εικόνες 7.36.α και 7.36.β, και το πλήθος των οποίων θα είναι ίσο με το πλήθος των μοναδικών RoutedId που διάβασε ο αναλυτής από τον καταναλωτή.

Στη συνέχεια, ο αναλυτής δεδομένων φροντίζει για την κανονικοποίηση των δεδομένων, έτσι ώστε να μπορέσουν να αξιοποιηθούν από το μοντέλο LSTM. Όπως έχει αναφερθεί και στο κεφάλαιο 6, χρησιμοποιείται η στρατηγική της κανονικοποίησης Ελάχιστου-Μέγιστου. Έπειτα από την κανονικοποίηση, τα παραγόμενα δεδομένα θα έχουν την εξής μορφή:

```
'03e0d9d2029f466d95c9b229e0e3bfa5', 3, 79, 89, array([[[-1.00000000e+00, -7.90789758e-01, -1.00000000e+00, ...,
        1.00000000e+00, -1.00000000e+00, -1.00000000e+00],
       [-8.92351270e-01, -5.74974635e-01, -8.18181520e-01, ...,
         4.17696120e-01, -1.00000000e+00, -1.00000000e+00],
       [-8.01699711e-01, -5.18072333e-01, -6.36363812e-01, ...,
         4.17696120e-01, -1.00000000e+00, -1.00000000e+00],
       ...,
       [ 6.26062317e-01, -4.26622280e-01,  8.18181713e-01, ...,
        -7.50459436e-01, -1.00000000e+00, -1.00000000e+00],
       [ 8.01699711e-01,  2.11242450e-01,  4.54545525e-01, ...,
        -1.00000000e+00, -1.00000000e+00, -1.00000000e+00],
       [ 1.00000000e+00,  1.00000000e+00,  4.54545525e-01, ...,
        -1.00000000e+00, -1.00000000e+00, -1.00000000e+00]],

      [[-1.00000000e+00, -6.85148634e-01, -9.99999576e-01, ...,
         1.00000000e+00, -1.00000000e+00, -1.00000000e+00],
       [-9.13279133e-01, -6.42996402e-01, -8.00000106e-01, ...,
         1.00000000e+00, -1.00000000e+00, -1.00000000e+00],
       [-8.26558266e-01, -6.14608162e-01, -2.00000000e-01, ...,
         7.68577223e-01, -1.00000000e+00, -1.00000000e+00],
       ...,
       [ 6.20596196e-01, -1.02732750e-01,  4.00000106e-01, ...,
        -8.12659369e-01, -1.00000000e+00, -1.00000000e+00],
       [ 8.10298098e-01,  4.81565066e-01,  4.00000106e-01, ...,
        -8.12659369e-01, -1.00000000e+00, -1.00000000e+00],
       [ 1.00000000e+00,  1.00000000e+00, -2.00000000e-01, ...,
        -1.00000000e+00, -1.00000000e+00, -1.00000000e+00]],

      [[-1.00000000e+00, -7.00715563e-01, -8.00000106e-01, ...,
         1.00000000e+00, -1.00000000e+00, -1.00000000e+00],
       [-9.17312661e-01, -6.76917039e-01, -2.00000000e-01, ...,
         7.68577223e-01, -1.00000000e+00, -1.00000000e+00],
       [-8.34625323e-01, -6.64657191e-01, -2.00000000e-01, ...,
```

Σχήμα 7.37: Κανονικοποίηση οδηγικών δεδομένων από τον αναλυτή δεδομένων

Παρατηρώντας το σχήμα 7.37 φαίνεται ότι πέρα από την κανονικοποίηση των δεδομένων, ο αναλυτής έχει αλλάξει ελάχιστα και την οργάνωση των δεδομένων. Πιο συγκεκριμένα, το πρώτο πεδίο της εγγραφής αναφέρεται στο RoutedId της διαδρομής που μελετάται, το δεύτερο πεδίο αναφέρεται στο DriverId του οδηγού του οχήματος, το τρίτο πεδίο αναφέρεται στο μικρότερο Timestamp των δεδομένων που μελετώνται, το τέταρτο πεδίο αναφέρεται στο μεγαλύτερο Timestamp των δεδομένων που μελετώνται και το πέμπτο πεδίο περιέχει σε μορφή λίστας και χρονολογικά ταξινομημένες τις πολυδιάστατες χρονοσειρές των αντίστοιχων Timestamps οι τιμές των οποίων είναι ανάμεσα στη τιμή του τρίτου και του τέταρτου πεδίου της εγγραφής. Με απλά λόγια, έχει παραχθεί μια εγγραφή η οποία περιέχει τα οδηγικά δεδομένα που αφορούν μια συγκεκριμένη διαδρομή που πραγματοποιεί κάποιος οδηγός για κάποιο συγκεκριμένο χρονικό διάστημα. Και σε αυτήν την περίπτωση θα πρέπει να τονιστεί ότι το πλήθος

των παραγόμενων εγγραφών είναι ίσο με το πλήθος των μοναδικών RoutedId που διαβάστηκαν κατά την ανάγνωση δεδομένων από τον καταναλωτή του συστήματος.

Μέχρι στιγμής, η λειτουργία του αναλυτή δεδομένων εξασφαλίζει ότι θα διαβάζονται όλα τα δεδομένα από τον καταναλωτή του συστήματος και θα ομαδοποιούνται ανάλογα με τη διαδρομή στην οποία ανήκουν. Επιπλέον, εξασφαλίζει ότι τα δεδομένα θα μετασχηματίζονται σε μορφή κατάλληλη ώστε να τροφοδοτηθούν ως είσοδος στο LSTM μοντέλο του συστήματος. Συνεπώς, η επόμενη ενέργεια την οποία εκτελεί ο αναλυτής δεδομένων σχετίζεται με τη πρόβλεψη της κατηγορίας που ανήκουν τα δεδομένα. Αυτό γίνεται προκειμένου να σχηματιστεί ένα προφίλ για την οδηγική συμπεριφορά που είχε ο εκάστοτε οδηγός για το συγκεκριμένο χρονικό διάστημα που μελετάται. Για το σκοπό αυτό, ο αναλυτής κάνει χρήση του αποθηκευμένου μοντέλου LSTM που αναφέρθηκε στο υποκεφάλαιο 7.3.4. Έπειτα από την πρόβλεψη του μοντέλου, τα δεδομένα έχουν την εξής μορφή:

```
1831b783229c4a0aae7bda7d630c176f, 4, 102, 125, [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1]
2a0538d86cb54301be639a6e3b684766, 3, 112, 135, [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1]
```

Σχήμα 7.38: Παραγόμενα δεδομένα έπειτα από την πρόβλεψη του LSTM μοντέλου

Οι εγγραφές που παρουσιάζονται στο σχήμα 7.38, έχουν τη μορφή που παρουσιάστηκε στο σχήμα 7.37, με τη μόνη διαφορά ότι το τελευταίο πεδίο αντί να περιέχει οδηγικά δεδομένα περιέχει τη κατηγορία που αυτά ανήκουν, όπως αυτή προβλέφθηκε από το μοντέλο LSTM. Πιο συγκεκριμένα, η ύπαρξη άσσου υποδηλώνει την ετικέτα EvenPaceStyle, ενώ η ύπαρξη μηδενικού υποδηλώνει την ετικέτα AggressiveStyle. Στο παράδειγμα που παρουσιάζεται στο σχήμα 7.38 φαίνεται ότι ο χρήστης με DriverId 4 πραγματοποιεί μια διαδρομή για την οποία η συμπεριφορά του όλες τις χρονικές στιγμές ανάμεσα στο χρονικό διάστημα 102 έως 125 έχει κατηγοριοποιηθεί στην ομάδα EvenPaceStyle. Ακόμη, φαίνεται ότι ο χρήστης με DriverId 3 πραγματοποιεί επίσης μια διαδρομή και ότι η συμπεριφορά του τις χρονικές στιγμές 126, 127, 128, 129 και 130 έχει χαρακτηριστεί ως επιθετική.

Συνεπώς, φαίνεται ότι το ζητούμενο έχει επιτευχθεί. Ο αναλυτής δέχεται τα δεδομένα από τον καταναλωτή και στη συνέχεια κατηγοριοποιεί τη συμπεριφορά του οδηγού το οχήματος με βάση αυτά. Ωστόσο, ο πίνακας που περιέχει την πληροφορία σχετικά με τη συμπεριφορά του οδηγού δεν αποτελεί την πιο αποτελεσματική αναπαράσταση αυτής της γνώσης. Αρκεί κανείς να σκεφτεί ότι θα είναι πολύ δύσκολο να ερμηνευθεί από κάποιον άνθρωπο κάποια διαδρομή η οποία, για παράδειγμα, διήρκησε μια ώρα, καθώς θα είναι απλά ένα μεγάλο σύνολο από 0 και 1. Για τον λόγο αυτό αποφασίστηκε να αποδοθεί η συμπεριφορά του οδηγού σε κάποιο χρονικό διάστημα με τη μορφή σκορ. Πιο συγκεκριμένα, σε κάθε οδηγό για κάθε χρονικό διάστημα που μελετάται αποδίδεται ένα σκορ "κανονικής" οδήγησης, το οποίο ονομάζεται normal_score, και ένα σκορ "επιθετικής" οδήγησης, το οποίο ονομάζεται aggressive_score. Το άθροισμα των δύο αυτών σκορ ισούται πάντα με 1. Το aggressive_score ισούται με το πλήθος των στοιχείων που έχουν χαρακτηριστεί με την ετικέτα 0 προς το πλήθος όλων των στοιχείων του διαστήματος που εξετάζεται. Αντίστοιχα, το normal_score ισούται με το πλήθος των στοιχείων που έχουν χαρακτηριστεί με την ετικέτα 1 προς το πλήθος όλων των στοιχείων του διαστήματος που εξετάζεται. Για παράδειγμα, το normal_score που πέτυχε ο οδηγός με DriverId 3 για το χρονικό διάστημα 112 έως 135 είναι ίσο με 0.76, ενώ το aggressive_score που

πέτυχε για το ίδιο διάστημα είναι ίσο με 0.24. Συνεπώς, τα τελικά αποτελέσματα που παράγονται από τη λειτουργία του αναλυτή δεδομένων έχουν την εξής μορφή:

```
1ac66146a2264f299cd5ef81192d353d, 3, 126, 143, 0.2777777777777778, 0.7222222222222222
7e969b6963c64a30bf8ee6a4b695151d, 4, 116, 133, 0.0, 1.0
```

Σχήμα 7.39: Τελικό αποτέλεσμα του αναλυτή δεδομένων

Παρατηρώντας το σχήμα 7.39, φαίνεται ότι οι τελικές εγγραφές περιέχουν ως πεδία το RouteId, το DriverId, το Timestamp της αρχής του χρονικού διαστήματος που μελετάται, το Timestamp του τέλους του χρονικού διαστήματος που μελετάται, το aggressive_score του οδηγού για το συγκεκριμένο χρονικό διάστημα καθώς και το normal_score του οδηγού για το συγκεκριμένο χρονικό διάστημα. Επομένως, αφού τα δεδομένα είναι πλέον στην επιθυμητή μορφή ο αναλυτής φροντίζει να τα εισάγει στη σχεσιακή βάση δεδομένων PostgreSQL, η οποία παρουσιάζεται αναλυτικά στο υποκεφάλαιο 7.3.6.

Συνεπώς, η λειτουργία του Apache Spark αναλυτή δεδομένων εξασφαλίζει ότι θα εξαχθούν τα απαραίτητα συμπεράσματα που σχετίζονται με τη συμπεριφορά του οδηγού και ότι στη συνέχεια θα αποθηκευτούν στη σχεσιακή βάση δεδομένων.

7.3.6 Βάση Δεδομένων PostgreSQL

Στο σύστημα που υλοποιήθηκε έχει οριστεί μια σχεσιακή βάση δεδομένων, η οποία ονομάζεται cars και είναι προσβάσιμη μέσω του connection string postgres://postgres:postgres@localhost/cars. Η βάση cars αποτελείται από 4 πίνακες οι οποίοι είναι οι εξής:

- Πίνακας users: Αυτός ο πίνακας ορίζει την οντότητα user και χρησιμοποιείται για την αποθήκευση των χρηστών του συστήματος, δηλαδή οδηγών που έχουν δώσει τη συναίνεση τους για τη συλλογή δεδομένων από τις διαδρομές που πραγματοποιούν με σκοπό την αξιολόγηση της οδηγικής συμπεριφοράς τους. Ως primary key αυτού του πίνακα έχει οριστεί ένα πεδίο το οποίο ονομάζεται user_id και ουσιαστικά ταυτίζεται με το πεδίο DriverId που παρουσιάστηκε στο υποκεφάλαιο 7.3.1. Επιπλέον, στον πίνακα users ορίζονται τα πεδία firstname και lastname τα οποία περιέχουν το όνομα και το επίθετο του εκάστοτε χρήστη, αντίστοιχα. Προφανώς, σε ένα εμπορικό σύστημα θα κρατούνταν παραπάνω πληροφορίες για κάθε χρήστη, ωστόσο στο πλαίσιο της παρούσης διπλωματικής εργασίας για λόγους απλότητας επιλέχτηκε να κρατηθούν μόνο οι πλήρως απαραίτητες πληροφορίες.

- Πίνακας routes: Αυτός ο πίνακας ορίζει την οντότητα route και χρησιμοποιείται για την αποθήκευση των διαδρομών που έχουν πραγματοποιηθεί από τους διάφορους χρήστες του συστήματος. Ως primary key αυτού του πίνακα έχει οριστεί ένα πεδίο το οποίο ονομάζεται route_id και ουσιαστικά ταυτίζεται με το πεδίο RouteId που παρουσιάστηκε στο υποκεφάλαιο 7.3.1. Επιπλέον, στον πίνακα routes ορίζονται τα πεδία user_id, date και is_active τα οποία περιέχουν το user_id του εκάστοτε χρήστη από τον πίνακα users, την ημερομηνία εκκίνησης της διαδρομής και την πληροφορία αν η διαδρομή είναι ενεργή, αντίστοιχα. Προφανώς, το user_id έχει οριστεί ως foreign key προερχόμενο από τον πίνακα users και χρησιμοποιείται για να προσδιοριστεί ο χρήστης που πραγματοποιεί την εκάστοτε διαδρομή. Τέλος,

εάν η τιμή του πεδίου is_active είναι ίση με μηδέν σημαίνει ότι η διαδρομή έχει ολοκληρωθεί και δεν συλλέγονται πλέον δεδομένα που σχετίζονται με αυτήν, ενώ εάν η τιμή του πεδίου is_active είναι ίση με ένα σημαίνει ότι η διαδρομή βρίσκεται σε εξέλιξη.

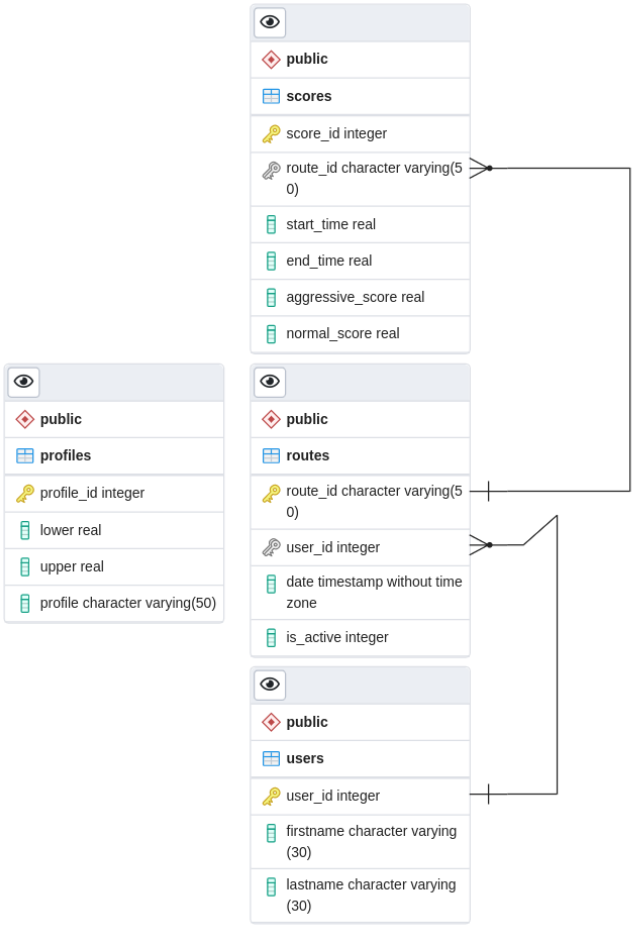
- Πίνακας scores: Αυτός ο πίνακας ορίζει την οντότητα score και χρησιμοποιείται για την αποθήκευση των βαθμολογιών που έχουν πραγματοποιηθεί από τους διάφορους χρήστες του συστήματος κατά την διάρκεια μια διαδρομής. Ως primary key αυτού του πίνακα έχει οριστεί ένα πεδίο το οποίο ονομάζεται score_id. Τα υπόλοιπα πεδία αυτού του πίνακα είναι τα route_id, start_time, end_time, aggressive_score και normal_score, και ταυτίζονται πλήρως με τα πεδία που παρουσιάστηκαν στο σχήμα 7.39. Πιο συγκεκριμένα, το πεδίο route_id έχει οριστεί ως foreign key προερχόμενο από τον πίνακα routes και χρησιμοποιείται για να προσδιοριστεί η διαδρομή στην οποία αναφέρεται η εκάστοτε βαθμολογία. Τα πεδία start_time και end_time χρησιμοποιούνται για να προσδιοριστεί το χρονικό διάστημα στο οποίο αποδόθηκε η εκάστοτε βαθμολογία. Τέλος, τα πεδία aggressive_score και normal_score περιέχουν τη βαθμολογία της συμπεριφοράς του εκάστοτε χρήστη για την εκάστοτε διαδρομή και υποδηλώνουν τον βαθμό επιθετικής και φυσιολογικής συμπεριφοράς, αντίστοιχα. Σε αυτό το σημείο θα πρέπει να αναφερθεί ότι ο αναλυτής παράγει τα αποτελέσματα που παρουσιάστηκαν στο σχήμα 7.39 και στη συνέχεια τα αποθηκεύει σε αυτόν τον πίνακα.

- Πίνακας profiles: Αυτός ο πίνακας ορίζει την οντότητα profile και χρησιμοποιείται για την εξαγωγή εύκολα αναγνώσιμης πληροφορίας, τον σχηματισμό των προφίλ των χρηστών καθώς και την κατηγοριοποίησή μιας διαδρομής. Ως primary key αυτού του πίνακα έχει οριστεί ένα πεδίο το οποίο ονομάζεται profile_id. Επιπλέον, στον πίνακα profiles ορίζονται τα πεδία lower, upper και profile. Το πεδίο profile αναφέρεται στην ετικέτα που μπορεί να λάβει μια βαθμολογία από τον πίνακα scores. Η ετικέτα profile μπορεί να λάβει μία από τις τιμές NORMAL, SLIGHTLY AGGRESSIVE, FAIRLY AGGRESSIVE, HIGHLY AGGRESSIVE και EXTREMELY AGGRESSIVE, οι οποίες υποδηλώνουν φυσιολογική συμπεριφορά, χαμηλού βαθμού επιθετικότητα, μεσαίου βαθμού επιθετικότητα, υψηλού βαθμού επιθετικότητα και υπερβολικά υψηλού βαθμού επιθετικότητα, αντίστοιχα. Τα πεδία lower και upper ορίζουν κάποια κατώτατα και ανώτατα thresholds, αντίστοιχα, με βάση τα οποία αποδίδεται η ετικέτα profile σε κάποια εγγραφή του πίνακα scores. Θα πρέπει να αναφερθεί ότι για την απόδοση της ετικέτας profile χρησιμοποιείται το πεδίο aggressive_score του πίνακα scores σε συνδυασμό με τα πεδία lower και upper του πίνακα profiles. Πιο συγκεκριμένα, εάν το aggressive_score μια εγγραφής τύπου score ανήκει σε κάποιο από τα διαστήματα που ορίζεται από τις τιμές των lower και upper του πίνακα profiles, τότε αποδίδεται η αντίστοιχη ετικέτα. Η χρησιμότητα του πίνακα profile θα φανεί καλύτερα στο υποκεφάλαιο 7.3.7, όπου παρουσιάζεται το REST API του συστήματος. Τέλος, θα πρέπει να αναφερθεί ότι ο πίνακας profiles είναι προκαθορισμένος εξ αρχής και παραμένει σταθερός καθ' όλη τη διάρκεια λειτουργίας του συστήματος, σε αντίθεση με τους υπόλοιπους πίνακες της σχεσιακής βάσης όπου εισάγονται δεδομένα συνεχώς.

	profile_id [PK] integer	lower real	upper real	profile character varying (50)
1	1	-1	0.05	NORMAL
2	2	0.05	0.3	SLIGHTLY AGGRESSIVE
3	3	0.3	0.6	FAIRLY AGGRESSIVE
4	4	0.6	0.85	HIGHLY AGGRESSIVE
5	5	0.85	1	EXTREMELY AGGRESSIVE

Σχήμα 7.40: Ο πίνακας profiles

Με βάση τα όσα ειπώθηκαν παραπάνω, το σχήμα της βάσης που δημιουργήθηκε είναι το εξής:



Σχήμα 7.41: Σχήμα της σχεσιακής βάσης δεδομένων

Παρατηρώντας το σχήμα 7.41, φαίνεται ότι οι σχέσεις που ορίζονται μεταξύ των πινάκων users-routes και routes-scores είναι σχέσεις one-to-many. Αυτό σημαίνει ότι ένα user_id μπορεί να αντιστοιχιστεί σε πολλές εγγραφές του πίνακα routes. Ομοίως ένα route_id μπορεί να αντιστοιχιστεί σε πολλές εγγραφές του πίνακα scores. Με απλά λόγια, ένας χρήστης του συστήματος μπορεί να πραγματοποιήσει πολλές διαδρομές και κάθε διαδρομή αποτελείται από χρονικά διαστήματα στα οποία έχουν αποδοθεί βαθμολογίες. Ακόμη, θα πρέπει να τονιστεί ότι ο τρόπος με τον οποίο έχει σχεδιαστεί η βάση επιτρέπει την ολοκληρωτική ανάκτηση της γνώσης μέσω των

αντίστοιχων SQL ερωτημάτων. Για παράδειγμα, αν για μια εγγραφή τύπου score πρέπει να βρεθεί ο χρήστης στον οποίο η βαθμολογία αναφέρεται, μπορεί πρώτα να προσδιοριστεί η διαδρομή της εγγραφής score μέσω του route_id και στη συνέχει ο χρήστης που πραγματοποιεί τη διαδρομή μέσω του user_id. Τέλος, από το σχήμα της σχεσιακής βάσης που δημιουργήθηκε φαίνεται ότι ο δεν ορίζεται καμία σχέση μεταξύ του πίνακα profiles και των άλλων πινάκων.

7.3.7 REST API

Στο πλαίσιο υλοποίησης του backend, δημιουργήθηκε ένα REST API με σκοπό την ανάκτηση των πληροφοριών που έχουν αποθηκευτεί στη σχεσιακή βάση δεδομένων PostgreSQL και το οποίο δέχεται κλήσεις στη διεύθυνση http://localhost:8080/api. Θα πρέπει να επισημανθεί ότι για την ανάκτηση των πληροφοριών έχει χρησιμοποιηθεί η γλώσσα SQL είτε έμμεσα, μέσω της βιβλιοθήκης sqlalchemy, είτε άμεσα. Το REST API αποτελείται από τα εξής endpoints:

- api/routes?active: Αυτό το endpoint υποστηρίζει τις μεθόδους GET και POST. Η μέθοδος POST χρησιμοποιείται για την εκτέλεση της προσομοίωσης λειτουργίας του συστήματος και θα παρουσιαστεί αναλυτικότερα στο υποκεφάλαιο 7.4. Η μέθοδος GET χρησιμοποιείται για την ανάκτηση όλων των διαδρομών που έχει καταγράψει το σύστημα και οι οποίες επιστρέφονται ταξινομημένες κατά αύξουσα σειρά με βάση την ημερομηνία εκκίνησης τους. Το active είναι μια προαιρετική παράμετρος, η οποία αν δοθεί στο αίτημα HTTP που αποστέλλεται, τότε το REST API θα συμπεριλάβει στην JSON απάντηση του μόνο τις διαδρομές που βρίσκονται σε εξέλιξη αυτή τη στιγμή. Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON το οποίο έχει ένα πεδίο, που ονομάζεται routes, και ως τιμή παίρνει μια λίστα από JSON αντικείμενα. Κάθε ένα από αυτά τα JSON αντικείμενα αντικατοπτρίζει μια διαδρομή και έχει τα πεδία data, is_active, route_id, lastname, firstname και user_id, τα οποία ταυτίζονται με τα αντίστοιχα πεδία των πινάκων routes και users της σχεσιακής βάσης του συστήματος.


```
{
    "routes": [
      | {
            "date": "2021-02-18 21:33:02",
            "firstname": "Marios",
            "is_active": 0,
            "lastname": "Lzi",
            "route_id": "33939258fdc5452cbea60ad5866bc5bf",
            "user_id": 2
        },
        {
            "date": "2021-02-17 09:13:27",
            "firstname": "Nikos",
            "is_active": 0,
            "lastname": "Vitas",
            "route_id": "ab56aca9bd2e4ab0a25e2c0f7a668b57",
            "user_id": 1
        }
    ]
}
```


Σχήμα 7.42: HTTP GET στο api/routes

- api/routes/<route_id>?include_labels&include_score: Αυτό το endpoint υποστηρίζει τις μεθόδους GET και POST. Η μέθοδος POST χρησιμοποιείται για την εκτέλεση της προσομοίωσης λειτουργίας του συστήματος και θα

παρουσιαστεί αναλυτικότερα στο υποκεφάλαιο 7.4. Η μέθοδος GET χρησιμοποιείται για την ανάκτηση των πληροφοριών που σχετίζονται με τη διαδρομή που έχει route_id ίσο με <route_id>. Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON το οποίο έχει τα πεδία is_active και route. Το πεδίο is_active ταυτίζεται με το αντίστοιχο πεδίο του πίνακα routes της σχεσιακής βάσης, ενώ το πεδίο route είναι μια λίστα από JSON αντικείμενα που αντικατοπτρίζουν τα σκορ που έχουν επιτευχθεί για τη συγκεκριμένη διαδρομή. Κάθε ένα από αυτά τα JSON αντικείμενα έχει τα πεδία start_time, end_time, score_id, aggressive_score και normal_score, τα οποία ταυτίζονται με τα αντίστοιχα πεδία του πίνακα scores της σχεσιακής βάσης του συστήματος. Συνεχίζοντας, το include_labels είναι μια προαιρετική παράμετρος, η οποία αν δοθεί στο αίτημα HTTP που αποστέλλεται, τότε το REST API για κάθε JSON αντικείμενο της λίστας routes θα συμπεριλάβει και ένα επιπλέον πεδίο το οποίο ονομάζεται profile και ταυτίζεται με το αντίστοιχο πεδίο του πίνακα profiles της σχεσιακής βάσης του συστήματος. Ουσιαστικά, το πεδίο route της απάντησης περιέχει όλες τις εγγραφές του πίνακα scores οι οποίες έχουν route_id ίσο με <route_id> και κάθε εγγραφή, βάσει της τιμής που έχει στο πεδίο aggressive_score, συνοδεύεται από την αντίστοιχη ετικέτα του πίνακα profiles εφαρμόζοντας τη μέθοδο που περιγράφτηκε στο υποκεφάλαιο 7.3.6. Τέλος, το include_score είναι μια προαιρετική παράμετρος, η οποία αν δοθεί στο αίτημα HTTP που αποστέλλεται, τότε το REST API θα συμπεριλάβει στην απάντησή του δύο επιπλέον πεδία που ονομάζονται count και score. Το πεδίο count περιέχει τα πλήθη των διάφορων ετικετών που έχουν αποδοθεί στα χρονικά διαστήματα της διαδρομής ομαδοποιημένα ανά ετικέτα. Το πεδίο score περιέχει την ίδια πληροφορία με το πεδίο count σε μορφή ποσοστών με βάση το 1. Θα πρέπει να σημειωθεί ότι στο σχήμα 7.43 δεν παρουσιάζονται όλα τα δεδομένα της απάντησης του REST API, καθώς ο όγκος τους ήταν ιδιαίτερα μεγάλος.


```
{
    "count": {
        "EXTREMELY AGGRESSIVE": 6,
        "FAIRLY AGGRESSIVE": 43,
        "HIGHLY AGGRESSIVE": 7,
        "NORMAL": 110,
        "SLIGHTLY AGGRESSIVE": 30
    },
    "is_active": 0,
    "score": {
        "EXTREMELY AGGRESSIVE": 0.0306,
        "FAIRLY AGGRESSIVE": 0.2194,
        "HIGHLY AGGRESSIVE": 0.0357,
        "NORMAL": 0.5612,
        "SLIGHTLY AGGRESSIVE": 0.1531
    },
    "route": [
        {
            "aggressive_score": 0.25,
            "end_time": 29.0,
            "normal_score": 0.75,
            "profile": "SLIGHTLY AGGRESSIVE",
            "score_id": 343,
            "start_time": 10.0
        },
        {
            "aggressive_score": 0.1,
            "end_time": 49.0,
            "normal_score": 0.9,
            "profile": "SLIGHTLY AGGRESSIVE",
```


Σχήμα 7.43: HTTP GET στο api/routes/<route_id>

- api/users: Αυτό το endpoint υποστηρίζει τη μέθοδο GET και χρησιμοποιείται για την ανάκτηση όλων των χρηστών του συστήματος. Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON το οποίο έχει ένα πεδίο, που ονομάζεται users, και ως τιμή παίρνει μια λίστα από JSON αντικείμενα. Κάθε ένα από αυτά τα JSON αντικείμενα αντικατοπτρίζει ένα χρήστη και έχει τα πεδία lastname, firstname και user_id, τα οποία ταυτίζονται με τα αντίστοιχα πεδία του πίνακα users της σχεσιακής βάσης του συστήματος. Θα πρέπει να σημειωθεί ότι στο σχήμα 7.44 δεν παρουσιάζονται όλα τα δεδομένα της απάντησης του REST API, καθώς ο όγκος τους ήταν μεγάλος.

Σχήμα 7.44: HTTP GET στο api/users

- api/users/<user_id>?include_routes&include_scores&include_profile: Αυτό το endpoint υποστηρίζει τη μέθοδο GET και χρησιμοποιείται για την ανάκτηση πληροφοριών που σχετίζονται με το χρήστη του συστήματος που έχει user_id ίσο με <user_id>. Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON το οποίο έχει τα πεδία lastname, firstname και user_id, τα οποία ταυτίζονται με τα αντίστοιχα πεδία του πίνακα users της σχεσιακής βάσης του συστήματος. Συνεχίζοντας, το include_routes είναι μια προαιρετική παράμετρος, η οποία αν δοθεί στο αίτημα HTTP που αποστέλλεται, τότε το REST API θα συμπεριλάβει στην απάντησή του ένα επιπλέον πεδίο που ονομάζονται routes και περιέχει πληροφορίες για όλες τις διαδρομές που έχει πραγματοποιήσει ο συγκεκριμένος χρήστης. Το πεδίο routes λαμβάνει ως τιμή μια λίστα από JSON αντικείμενα, κάθε ένα από τα οποία περιέχει τα πεδία date, is_active και route_id, τα οποία ταυτίζονται με τα αντίστοιχα πεδία του πίνακα routes της σχεσιακής βάσης του συστήματος. Εάν, επιπλέον, δοθεί και η προαιρετική παράμετρος include_scores, για κάθε μια από τις διαδρομές της λίστας routes θα συμπεριληφθεί και ένα πεδίο score, το οποίο είναι ίδιο με αυτό που παρουσιάστηκε στο endpoint api/routes/<route_id>. Τέλος, το include_profile είναι μια προαιρετική παράμετρος, η οποία αν δοθεί στο αίτημα HTTP που αποστέλλεται, τότε το REST API θα συμπεριλάβει στην απάντησή του ένα επιπλέον πεδίο που ονομάζονται profile και περιέχει το οδηγικό προφίλ του χρήστη. Το πεδίο έχει παρόμοια δομή με το πεδίο score μιας διαδρομής και προκύπτει ως ο μέσος όρος των πεδίων score όλως των διαδρομών που βρίσκονται στη λίστα routes.

```
{
    "firstname": "Nikos",
    "lastname": "Vitas",
    "profile": {
        "EXTREMELY AGGRESSIVE": 0.0234,
        "FAIRLY AGGRESSIVE": 0.2017,
        "HIGHLY AGGRESSIVE": 0.0614,
        "NORMAL": 0.5819,
        "SLIGHTLY AGGRESSIVE": 0.1316
    },
    "routes": [
        {
            "date": "2021-02-17 09:13:27",
            "is_active": 0,
            "route_id": "ab56aca9bd2e4ab0a25e2c0f7a668b57",
            "score": {
                "EXTREMELY AGGRESSIVE": 0.0234,
                "FAIRLY AGGRESSIVE": 0.2017,
                "HIGHLY AGGRESSIVE": 0.0614,
                "NORMAL": 0.5819,
                "SLIGHTLY AGGRESSIVE": 0.1316
            }
        }
    ],
    "user_id": 1
}
```


Σχήμα 7.45: HTTP GET στο api/users/<users_id>

- api/users/<user_id>/routes?include_scores: Αυτό το endpoint υποστηρίζει τη μέθοδο GET και χρησιμοποιείται για την ανάκτηση πληροφοριών που σχετίζονται με τις διαδρομές που έχει πραγματοποιήσει ο χρήστης του συστήματος που έχει user_id ίσο με <user_id>. Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON, το οποίο περιέχει ένα πεδίο routes και είναι ακριβώς ίδιο με αυτό που παρουσιάστηκε για το endpoint api/user/<user_id>.


```
{
    "routes": [
        {
            "date": "2021-02-17 09:13:27",
            "is_active": 0,
            "route_id": "ab56aca9bd2e4ab0a25e2c0f7a668b57",
            "score": {
                "EXTREMELY AGGRESSIVE": 0.0234,
                "FAIRLY AGGRESSIVE": 0.2017,
                "HIGHLY AGGRESSIVE": 0.0614,
                "NORMAL": 0.5819,
                "SLIGHTLY AGGRESSIVE": 0.1316
            }
        }
    ]
}
```


Σχήμα 7.46: HTTP GET στο api/users/<users_id>/routes

- api/users/<user_id>/routes/<route_id>?include_labels&include_scores: Αυτό το endpoint υποστηρίζει τη μέθοδο GET και χρησιμοποιείται για την ανάκτηση πληροφοριών που σχετίζονται με μια συγκεκριμένη διαδρομή που έχει πραγματοποιήσει ο χρήστης του συστήματος που έχει user_id ίσο με <user_id> και η οποία έχει route_id ίσο με <route_id>. Η απάντηση του REST API είναι ακριβώς ίδια με αυτήν που παρουσιάστηκε στο endpoint api/routes/<route_id>.

```
{
    "count": {
        "EXTREMELY AGGRESSIVE": 8,
        "FAIRLY AGGRESSIVE": 69,
        "HIGHLY AGGRESSIVE": 21,
        "NORMAL": 199,
        "SLIGHTLY AGGRESSIVE": 45
    },
    "is_active": 0,
    "score": {
        "EXTREMELY AGGRESSIVE": 0.0234,
        "FAIRLY AGGRESSIVE": 0.2017,
        "HIGHLY AGGRESSIVE": 0.0614,
        "NORMAL": 0.5819,
        "SLIGHTLY AGGRESSIVE": 0.1316
    },
    "route": [
        {
            "aggressive_score": 0.0,
            "end_time": 29.0,
            "normal_score": 1.0,
            "profile": "NORMAL",
            "score_id": 1,
            "start_time": 10.0
        },
        {
            "aggressive_score": 0.0,
            "end_time": 49.0,
            "normal_score": 1.0,
            "profile": "NORMAL",
```


Σχήμα 7.47: HTTP GET στο api/users/<users_id>/routes/<route_id>

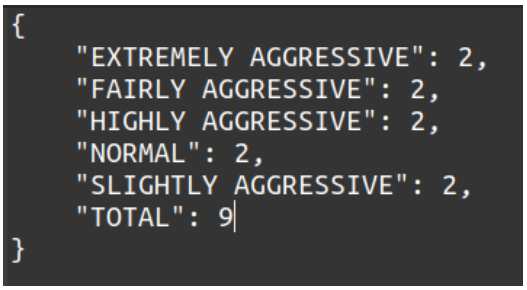
- api/users/<user_id>/routes/<route_id>/<score_id>: Αυτό το endpoint υποστηρίζει τη μέθοδο GET και χρησιμοποιείται για την ανάκτηση πληροφοριών που σχετίζονται με τη βαθμολογία ενός χρονικού διαστήματος για τη διαδρομή που έχει έχει route_id ίσο με <route_id> και την οποία έχει πραγματοποιήσει ο χρήστης που έχει user_id ίσο με <user_id> και η οποία . Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON, το οποίο περιέχει τα πεδία start_time, end_time, score_id, aggressive_score και normal_score, τα οποία ταυτίζονται με τα αντίστοιχα πεδία του πίνακα scores της σχεσιακής βάσης του συστήματος.


```
{
    "aggressive_score": 0.0,
    "end_time": 29.0,
    "normal_score": 1.0,
    "score_id": 1,
    "start_time": 10.0
}
```


Σχήμα 7.48: HTTP GET στο api/users/<users_id>/routes/<route_id>/<score_id>

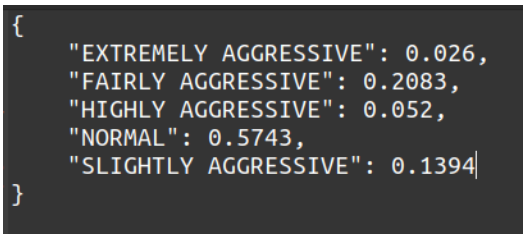
- api/stats/users: Αυτό το endpoint υποστηρίζει τη μέθοδο GET και χρησιμοποιείται για την ανάκτηση στατιστικών που σχετίζονται με τους χρήστες του συστήματος. Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON, το οποίο περιέχει τα πεδία TOTAL, NORMAL, SLIGHTLY AGGRESSIVE, FAIRLY AGGRESSIVE, HIGHLY AGGRESSIVE και EXTREMELY AGGRESSIVE, τα οποία λαμβάνουν ως τιμή έναν ακέραιο αριθμό. Το πεδίο TOTAL υποδηλώνει το πλήθος των χρηστών που

χρησιμοποιούν το σύστημα που έχει υλοποιηθεί, ενώ όλα τα υπόλοιπα πεδία περιέχουν το πλήθος των χρηστών που τους έχει αποδοθεί η αντίστοιχη ετικέτα. Για να έχει αποδοθεί η εκάστοτε ετικέτα σε κάποιον χρήστη θα πρέπει στο προφίλ που έχει σχηματιστεί για αυτόν (πεδίο profile του endpoint /api/users/<user_id>) η αντίστοιχη ετικέτα να έχει θετική τιμή.



Σχήμα 7.49: HTTP GET στο api/stats/users

- api/stats/routes: Αυτό το endpoint υποστηρίζει τη μέθοδο GET και χρησιμοποιείται για την ανάκτηση πληροφοριών που σχετίζονται με τον μέσο χρήστη του συστήματος. Η απάντηση του REST API αποτελείται από ένα αντικείμενο JSON το οποίο έχει ίδια μορφή με το πεδίο profile που περιγράφτηκε στο endpoint api/users/<user_id>. Ουσιαστικά, επιστρέφεται το προφίλ που θα είχε ένας χρήστης εάν είχε πραγματοποιήσει όλες τις διαδρομές που έχουν αποθηκευτεί στη σχεσιακή βάση δεδομένων. Το προφίλ αυτό προκύπτει ως ο μέσος όρος των σκορ που έχουν όλες οι διαδρομές του συστήματος.



Σχήμα 7.50: HTTP GET στο api/stats/routes

Συνεπώς, η λειτουργία του REST API εξασφαλίζει ότι θα δεδομένα που έχουν αποθηκευτεί στη σχεσιακή βάση δεδομένων θα είναι εύκολα διαθέσιμα. Επιπλέον, μέσω στατιστικών υπολογισμών που λαμβάνουν χώρα στο backend δίνεται η δυνατότητα επιστροφής αποτελεσμάτων που εξάγονται από τα αρχικά δεδομένα και τα οποία μπορούν ευκολότερα να ερμηνευθούν και να οδηγήσουν σε συμπεράσματα. Για παράδειγμα, η οδηγική συμπεριφορά ενός χρηστή για μια διαδρομή αντικατοπτρίζεται από τις εγγραφές του πίνακα scores της σχεσιακής βάσης δεδομένων που αφορούν τη συγκεκριμένη διαδρομή. Προφανώς, το ανθρώπινο μάτι είναι δύσκολο να καταναλώσει όλη αυτή τη πληροφορία και να εξάγει συμπεράσματα για την οδηγική συμπεριφορά του χρήστη. Ο τρόπος με τον οποίο έχει υλοποιηθεί το REST API επιτρέπει την επιστροφή ενός συνολικού σκορ για ολόκληρη την διαδρομή το οποίο μπορεί πολύ ευκολότερα να κατανοήσει ένας άνθρωπος.

7.4 Προσομοίωση Λειτουργίας του Συστήματος

Μέχρι στιγμής έχει περιγραφεί πολύ αναλυτικά η πορεία που ακολουθούν τα δεδομένα μέσα στο σύστημα, από τη στιγμή της εισαγωγής του σε αυτό μέχρι τη στιγμή που αποθηκεύονται στη σχεσιακή βάση δεδομένων. Στο υποκεφάλαιο 7.3 περιγράφτηκε λεπτομερώς ο τρόπος με τον οποίο κάθε συστατικό του συστήματος διαχειρίζεται τα δεδομένα και δόθηκαν αρκετά παραδείγματα, μέσω εικόνων, έτσι ώστε να είναι ξεκάθαρο το πως μετασχηματίζονται τα δεδομένα κατά μήκος του συστήματος. Σε αυτό το υποκεφάλαιο θα παρουσιαστεί ο τρόπος με τον οποίο προσομοιώθηκε η λειτουργία του συστήματος ώστε να προσεγγίζει μια πραγματική εφαρμογή η οποία δέχεται δεδομένα από πραγματικά οχήματα.

Αρχικά, η πρώτη ενέργεια που εκτελέστηκε ήταν η εκπαίδευση του μοντέλου LSTM. Όπως έχει αναφερθεί και στο κεφάλαιο 5, το σύνολο δεδομένων αποτελείται από 4 αρχεία csv. Συνεπώς, δημιουργήθηκε ένα python script το οποίο διαβάζει τα 4 αρχεία csv και εισάγει στη βάση MongoDB τα δεδομένα που περιέχονται σε αυτά. Στη συνέχεια, μέσω ενός άλλου python script δημιουργείται το μοντέλο LSTM το οποίο εκπαιδεύεται με σύνολο δεδομένων όλα τα δεδομένα που βρίσκονται στη βάση MongoDB και τελικά αποθηκεύεται στο σύστημα, προκειμένου να χρησιμοποιηθεί στο μέλλον από τον αναλυτή δεδομένων του συστήματος. Η επιλογή να γίνει με αυτόν τον τρόπο η εκπαίδευση του μοντέλου, και όχι απευθείας από τα αρχεία csv, δεν ήταν καθόλου τυχαία. Σε ένα πραγματικό σύστημα το μοντέλο θα πρέπει να έχει τη δυνατότητα επανεκπαίδευσης στο μέλλον και καθώς το σύνολο δεδομένων αποτελείται από μεγάλα δεδομένα κάτι τέτοιο δεν θα μπορεί να συμβεί μέσω αρχείων csv, αλλά μέσα από μια NoSQL βάση δεδομένων στην οποία θα είναι αποθηκευμένα όλα τα δεδομένα με αποτελεσματικό τρόπο.

Καθώς το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία αποτελούταν από μικρό πλήθος (4 διαδρομές) κρίθηκε απαραίτητο η εκπαίδευση του μοντέλου LSTM να πραγματοποιηθεί με ολόκληρο το σύνολο δεδομένων, έτσι ώστε να εξασφαλιστεί η υψηλή απόδοση του μοντέλου. Ωστόσο, στα πλαίσια της προσομοίωσης λειτουργίας του συστήματος θα πρέπει τουλάχιστον μια διαδρομή να μην έχει εισαχθεί στη βάση MongoDB, προκειμένου να ελεγχθεί η λειτουργία του συστήματος όταν τα δεδομένα της γίνονται streaming στο σύστημα σε πραγματικό χρόνο. Επομένως, αποφασίστηκε να αρχικοποιηθεί εκ νέου η βάση MongoDB με τα δεδομένα των δύο διαδρομών που πραγματοποίησε το Opel Corsa 1.3 HDi. Τα δεδομένα των διαδρομών που πραγματοποίησε το Peugeot 207 1.4 HDi χρησιμοποιήθηκαν για να προσομοιώσουν με ρεαλιστικό τρόπο την αποστολή δεδομένων μέσω κάποιου οχήματος. Συνεπώς, αφού πρώτα διαγράφηκαν όλα τα δεδομένα που προηγουμένως είχαν τοποθετηθεί στη βάση MongoDB, δημιουργήθηκε ένα python script το οποίο διαβάζει τα 2 αρχεία csv που αφορούν τις διαδρομές του Opel Corsa 1.3 HDi και εισάγει τα αντίστοιχα δεδομένα στη βάση. Σε αυτό το σημείο αξίζει να αναφερθεί ότι μέσω του script αυτού δημιουργούνται και ορίζονται οι τιμές των πεδίων DriverId, RouteId και Timestamp. Για τις δύο αυτές διαδρομές παράχθηκαν δύο μοναδικά RoutedId, ενώ το DriverId της μιας διαδρομής ορίστηκε ίσο με 1 και της άλλης ίσο με 2.

Αντίστοιχα, δημιουργήθηκε ένα python script το οποίο εισάγει τα δεδομένα των δύο αυτών διαδρομών στη σχεσιακή βάση δεδομένων PostgreSQL. Αρχικά, γίνεται αρχικοποίηση του πίνακα users, ο οποίος πληρώνεται με 9 χρήστες. Από αυτούς τους

χρήστες, οι χρήστες που έχουν DriverId ίσα με 1 και 2 έχουν θεωρητικά πραγματοποιήσει τις δύο διαδρομές που υπάρχουν στη βάση MongoDB, ενώ οι υπόλοιποι 7 χρήστες του συστήματος δεν έχουν πραγματοποιήσει καμία διαδρομή. Στη συνέχεια, γεμίζεται ο πίνακας routes με τα αντίστοιχα δεδομένα των διαδρομών που έχουν εισαχθεί στη βάση MongoDB. Τέλος, κάνοντας χρήση του μοντέλου LSTM παράγονται οι βαθμολογίες για τα διάφορα χρονικά διαστήματα των διαδρομών, που ορίζονται ανά 20 δευτερόλεπτα, και αποθηκεύονται στον πίνακα scores. Ο λόγος που επιλέχτηκε να εισαχθούν με αυτόν τον τρόπο τα δεδομένα και όχι μέσω της λειτουργίας του αναλυτή δεδομένων του συστήματος ήταν η εξοικονόμηση χρόνου, καθώς ο αναλυτής δεδομένων έχει σχεδιαστεί να λειτουργεί με βάση δεδομένα που γίνονται streaming ανά 1 δευτερόλεπτο.

Με βάση τα όσα έχουν ειπωθεί παραπάνω, μέχρι στιγμής το σύστημα έχει αρχικοποιηθεί επιτυχώς με το μισό σύνολο δεδομένων, δηλαδή τις δύο διαδρομές του Opel Corsa 1.3 HDi. Επομένως, το μόνο που απομένει είναι να παρουσιαστεί ο τρόπος με τον οποίον γίνονται streaming τα δεδομένα των διαδρομών που πραγματοποίησε το Peugeot 207 1.4 HDi. Όπως έχει αναφερθεί και στο κεφάλαιο 5, τα δεδομένα του dataset που χρησιμοποιήθηκε στη παρούσα διπλωματική εργασία συλλέχθηκαν με συχνότητα 1 Hertz. Στο πλαίσιο, λοιπόν, της ρεαλιστικής προσομοίωσης της λειτουργίας του συστήματος, κάθε εγγραφή δεδομένων των διαδρομών που πραγματοποίησε το Peugeot 207 1.4 HDi θα πρέπει να εισάγεται στη βάση δεδομένων MongoDB ανά 1 δευτερόλεπτο, έτσι ώστε τα δεδομένα να παράγονται σαν να πραγματοποιούταν η διαδρομή εκείνη τη στιγμή. Για το σκοπό αυτό δημιουργήθηκε ένα python script το οποίο διαβάζει ένα αρχείο csv και εισάγει τις εγγραφές του αρχείου αυτού στη βάση δεδομένων MongoDB ανά 1 δευτερόλεπτο. Συνεπώς, υποθέτοντας ότι όλα τα συστατικά του συστήματος επικοινωνούν μεταξύ τους και λειτουργούν ορθά, η προσομοίωση της λειτουργίας του συστήματος επιτυγχάνεται με την παράλληλη εκτέλεση αυτού του script δύο φορές. Μια φορά για τη μία διαδρομή του Peugeot 207 1.4 HDi και μια φορά για την άλλη διαδρομή του Peugeot 207 1.4 HDi.

Τέλος, ένα ακόμη σημείο το οποίο αξίζει να διευκρινιστεί είναι ο τρόπος με τον οποίο ενημερώνεται η σχεσιακή βάση δεδομένων για την πραγματοποίηση μιας καινούριας διαδρομής. Προκειμένου αυτό να γίνει πλήρως κατανοητό, θα πρέπει πρώτα να περιγράφει το πως ακριβώς ορίζεται ένας χρήστης ο οποίος χρησιμοποιεί το σύστημα που υλοποιήθηκε. Έστω, λοιπόν, ένας χρήστης ο οποίος εγγράφεται στο σύστημα για πρώτη φορά με σκοπό την αξιολόγηση της οδηγικής συμπεριφοράς του. Αρχικά, ο χρηστής εισάγεται στη σχεσιακή βάση δεδομένων στον πίνακα users και επομένως του αποδίδεται ένα μοναδικό DriverId. Προκειμένου το σύστημα να μπορεί να λάβει δεδομένα από το όχημα αυτού του χρήστη θα πρέπει να έχει προηγηθεί η εγκατάσταση κάποιου λογισμικού στο όχημα του. Το λογισμικό αυτό θα έχει αποθηκευμένο το DriverId του οδηγού του οχήματος, ενώ για κάθε νέα διαδρομή που αυτός ξεκινάει (έστω ότι ως νέα διαδρομή ορίζεται κάθε διαδρομή που ξεκινάει με σβηστή τη μηχανή του κινητήρα) θα πρέπει να παράγει ένα μοναδικό RoutedId. Χάρη στη λειτουργία ενός τέτοιου λογισμικού τα δεδομένα θα έχουν τη μορφή που παρουσιάστηκε στο σχήμα 7.27 και ανά 1 δευτερόλεπτο θα στέλνονται στη βάση MongoDB. Η λειτουργικότητα ενός τέτοιου λογισμικού έχει ενσωματωθεί στο script που σχετίζεται με το streaming των δεδομένων. Συνεχίζοντας, προκειμένου να ενημερωθεί η σχεσιακή βάση δεδομένων για την εκκίνηση μιας νέας διαδρομής, το script που σχετίζεται με το streaming των δεδομένων, πριν ξεκινήσει την αποστολή

των οδηγικών δεδομένων που συλλέγονται από τους διάφορους αισθητήρες, στέλνει ένα αίτημα POST στο REST API στο endpoint api/routes. Το μήνυμα POST που αποστέλλεται περιέχει ένα μήνυμα JSON στο οποίο δηλώνονται τα πεδία route_id, driver_id, date και is_active της διαδρομής που πρόκειται να ξεκινήσει. Προφανώς το πεδίο is_active θα έχει τιμή ίση με 1 ενώ οι τιμές των υπόλοιπων πεδίων έχουν προκύψει από τη λειτουργία του υποτιθέμενου λογισμικού που είναι ενσωματωμένο στο όχημα του χρήστη. Το REST API αναλαμβάνει να αποθηκεύσει αυτές τις τιμές στον πίνακα routes. Ομοίως, όταν τελειώσει μία διαδρομή (έστω ότι μια διαδρομή θεωρείται ότι έχει τελειώσει όταν σβήσει η μηχανή του κινητήρα του οχήματος) το υποτιθέμενο λογισμικό του οχήματος στέλνει ένα κενό αίτημα POST στο REST API στο endpoint api/routes/<route_id>, όπου <route_id> το RoutedId που έχει παράγει το λογισμικό για την τρέχουσα διαδρομή. Σε αυτήν την περίπτωση, το REST API αναλαμβάνει να ενημερώσει την τιμή του πεδίου is_active από 1 σε 0 για τη διαδρομή που έχει route_id ίσο με <route_id>, γεγονός που επισημαίνει ότι η πραγματοποίηση της συγκεκριμένης διαδρομής έλαβε τέλος.

7.5 Αλγόριθμος Profiling Οδηγικής Συμπεριφοράς

Σε αυτό το σημείο, όπου έχει παρουσιαστεί αναλυτικά ο τρόπος με τον οποίο λειτουργεί και επεξεργάζεται τα δεδομένα κάθε συστατικό του συστήματος, κρίνεται απαραίτητο να παρουσιαστεί πιο εμπεριστατωμένα ο τρόπος με τον οποίο πραγματοποιείται το profiling των χρηστών που συμμετέχουν στην πλατφόρμα που υλοποιήθηκε.

Έστω μια διαδρομή Δ η οποία έχει την εξής μορφή:

$$\Delta = \langle (r, d, t_1, \vec{x}_{t_1}), (r, d, t_2, \vec{x}_{t_2}), ..., (r, d, t_n, \vec{x}_{t_n}) \rangle = \langle (r, d, 1, \vec{x}_1), (r, d, 2, \vec{x}_2), ..., (r, d, n, \vec{x}_n) \rangle \tag{7.1}$$

όπου $\vec{x_i} = \{x^{AltitudeVariation}(i), ..., x^{traffic}(i)\}$, όπου το διάνυσμα x περιέχει όλα τα πεδία που παρουσιάστηκαν στο υποκεφάλαιο 5.1, εκτός από το πεδίο drivingStyle το οποίο θα προβλεφθεί από το μοντέλο LSTM του συστήματος, για κάποια συγκεκριμένη χρονική στιγμή t_i. Με r συμβολίζεται το RouteId της διαδρομής, ενώ με d το DriverId του οδηγού του οχήματος.

Έπειτα από την επεξεργασία του καταναλωτή του συστήματος και την εφαρμογή του κυλιόμενου παραθύρου, το οποίο έχει μέγεθος 20, τα δεδομένα της διαδρομής Δ έχουν την εξής μορφή:

$$\Delta = \langle (r, d, 20, \vec{y}_{20}), (r, d, 21, \vec{y}_{21}), ..., (r, d, n, \vec{y}_n) \rangle \tag{7.2}$$

όπου $\vec{y_i} = \langle \vec{x}_{i-19}, \vec{x}_{i-18}, ..., \vec{x}_i \rangle$, όπου το διάνυσμα y περιέχει όλα τα διανύσματα εγγραφών x για το χρονικό διάστημα $[i-19, i]$. Δεδομένου ενός τέτοιου διανύσματος y το μοντέλο μηχανικής μάθησης του συστήματος καλείται να προβλέψει την σωστή κλάση (EvenPaceStyle ή AggressiveStyle) για τη χρονική στιγμή i.

Συνεχίζοντας, έπειτα από την πρόβλεψη του LSTM μοντέλου τα δεδομένα της διαδρομής Δ έχουν την εξής μορφή:

$$\Delta = \langle (r, d, 20, \vec{y}_{20}, label_{20}), (r, d, 21, \vec{y}_{21}, label_{21}), ..., (r, d, n, \vec{y}_n, label_n) \rangle \tag{7.3}$$

όπου με $label_i$ συμβολίζεται η ετικέτα που προβλέφθηκε από το μοντέλο LSTM για το διάνυσμα y_i. Προφανώς, το $label_i$ μπορεί να πάρει ως τιμή μόνο μία από τις ετικέτες EvenPaceStyle και AggressiveStyle.

Όπως έχει αναφερθεί και στο υποκεφάλαιο 7.3.5 ο αναλυτής δεδομένων του συστήματος διαβάζει τα δεδομένα που υπάρχουν στον καταναλωτή ανά 20 δευτερόλεπτα και τα ομαδοποιεί σε σύνολα ανάλογα με του RoutedId που αυτά έχουν. Υποθέτοντας, λοιπόν, ότι ο αναλυτής δεδομένων σε κάθε επανάληψη διαβάσματος j διαβάζει το σύνολο N_j, των εγγραφών που φέρουν το RoutedId r της διαδρομής Δ, τα δεδομένα που προκύπτουν έχουν την εξής μορφή:

$$\begin{aligned}\Delta = \langle (r, d, tLeft_{N_1}, tRight_{N_1}, \langle label_i{:}\forall \vec{y}_i \in N_1 \rangle), ..., (r, d, tLeft_{N_m}, tRight_{N_m}, \langle label_i{:}\forall \vec{y}_i \in N_m \rangle) \rangle \\ \Rightarrow \langle (r, d, tLeft_{N_1}, tRight_{N_1}, LABEL_1), ..., (r, d, tLeft_{N_m}, tRight_{N_m}, LABEL_m) \rangle\end{aligned} \tag{7.4}$$

όπου όλα τα σύνολα N_j είναι ξένα μεταξύ τους και με $tLeft_{N_j}$ και $tRight_{N_j}$ συμβολίζεται το μικρότερο και το μεγαλύτερο Timestamp από τις εγγραφές που διαβάστηκαν από τον αναλυτή δεδομένων την επανάληψη j. Ουσιαστικά η παραπάνω εξίσωση εκφράζει το γεγονός ότι τα δεδομένα της αρχικής διαδρομής Δ έχουν χωριστεί σε επιμέρους χρονικά διαστήματα (τα οποία έχουν περίπου μήκος 20 το κάθε ένα) και για κάθε ένα από αυτά τα διαστήματα έχουν προβλεφθεί και κρατηθεί σε μια λίστα οι ετικέτες των εγγραφών y που ανήκουν στο εκάστοτε διάστημα.

Σε αυτό το σημείο πραγματοποιείται το πρώτο profiling της οδηγικής συμπεριφοράς των χρηστών που συμμετέχουν στο σύστημα. Πιο συγκεκριμένα, οι ετικέτες ενός συνόλου ετικετών $LABEL_i$ (το σύνολο $LABEL_i$ αναφέρεται στο σύνολο των ετικετών των εγγραφών που ανήκουν στο σύνολο N_i) μετατρέπονται σε βαθμολογίες που αντικατοπτρίζουν τον βαθμό της επιθετικής συμπεριφοράς του οδηγού του οχήματος που πραγματοποιεί τη διαδρομή Δ για το εκάστοτε χρονικό διάστημα. Αυτό επιτυγχάνεται ως εξής:

1. Για κάθε σύνολο $LABEL_i$ υπολογίζεται το πλήθος των ετικετών που έχουν τιμή AggressiveStyle, έστω A, και το πλήθος των ετικετών που έχουν τιμή EvenPaceStyle, έστω E. Ισχύει:

$$A + E = |LABEL_i| \tag{7.5}$$

2. Με βάση τις τιμές A και E υπολογίζονται δύο δείκτες οι οποίοι ονομάζονται $aggressive\_score$ και $normal\_score$ και οι οποίοι αναφέρονται στο χρονικό διάστημα το οποίο ορίζεται από το σύνολο $LABEL_i$. Οι δείκτες $aggressive\_score$ και $normal\_score$ υπολογίζονται ως εξής:

$$\begin{aligned}aggressive\_score = \frac{A}{|LABEL_i|}, normal\_score = \frac{E}{|LABEL_i|}, \\ aggressive\_score + normal\_score = 1\end{aligned} \tag{7.6}$$

Όσο περισσότερες είναι οι εγγραφές του συνόλου N_i (το αντίστοιχο σύνολο εγγραφών του συνόλου ετικετών $LABEL_i$) οι οποίες έχουν ετικέτα AggressiveStyle τόσο μεγαλύτερο θα είναι το $aggressive\_score$ του οδηγού του οχήματος που πραγματοποιεί τη διαδρομή Δ για το αντίστοιχο χρονικό διάστημα. Συνεπώς, σε ένα πρώτο στάδιο το σύστημα εκτελεί ένα βαθμολογικό profiling μιας διαδρομής που εκτελεί κάποιος χρήστης. Έπειτα από αυτό το profiling τα δεδομένα της διαδρομής Δ θα έχουν την εξής μορφή:

$$\begin{aligned}\Delta = \langle &(r, d, tLeft_{N_1}, tRight_{N_1}, aggressive\_score_{N_1}, normal\_score_{N_1}), ..., \\ &(r, d, tLeft_{N_m}, tRight_{N_m}, aggressive\_score_{N_m}, normal\_score_{N_m})\rangle\end{aligned} \tag{7.7}$$

Σε αυτό το σημείο πραγματοποιείται το δεύτερο profiling της οδηγικής συμπεριφοράς των χρηστών που συμμετέχουν στο σύστημα. Για το δεύτερο profiling χρησιμοποιείται ο πίνακας profiles της σχεσιακής βάσης δεδομένων PostgreSQL, ο οποίος παρουσιάστηκε στο σχήμα 7.40. Ουσιαστικά ο πίνακας profiles ορίζει μια συνάρτηση $f(\cdot)$ η οποία αντιστοιχίζει τον δείκτη $aggressive\_score$ σε μια εκ των ετικετών NORMAL, SLIGHTLY AGGRESSIVE, FAIRLY AGGRESSIVE, HIGHLY AGGRESSIVE και EXTREMELY AGGRESSIVE. Η εικονική αυτή συνάρτηση έχει την εξής μορφή:

$$f(aggressive\_score) = \begin{cases} NORMAL & -1 < aggressive\_score \leq 0.05 \\ SLIGHTLY AGGRESSIVE & 0.05 < aggressive\_score \leq 0.3 \\ FAIRLY AGGRESSIVE & 0.3 < aggressive\_score \leq 0.6 \\ HIGHLY AGGRESSIVE & 0.6 < aggressive\_score \leq 0.85 \\ EXTREMELY AGGRESSIVE & 0.85 < aggressive\_score \leq 1 \end{cases} \tag{7.8}$$

Επομένως, τα δεδομένα της διαδρομής Δ θα έχουν την εξής μορφή:

$$\begin{aligned}\Delta = \langle &(r, d, tLeft_{N_1}, tRight_{N_1}, f(aggressive\_score_{N_1}), normal\_score_{N_1}), ..., \\ &(r, d, tLeft_{N_m}, tRight_{N_m}, f(aggressive\_score_{N_m}), normal\_score_{N_m})\rangle\end{aligned} \tag{7.9}$$

Ουσιαστικά η παραπάνω εξίσωση εκφράζει το γεγονός ότι τα δεδομένα της αρχικής διαδρομής Δ έχουν χωριστεί σε επιμέρους χρονικά διαστήματα κάθε ένα από τα οποία έχει αξιολογηθεί ως NORMAL ή SLIGHTLY AGGRESSIVE ή FAIRLY AGGRESSIVE ή HIGHLY AGGRESSIVE ή EXTREMELY AGGRESSIVE. Το τελικό profiling της διαδρομής Δ προκύπτει ως εξής:

1. Υπολογίζεται το πλήθος των ετικετών που έχουν αποδοθεί στο δεύτερο profiling ανά κατηγορία. Το πλήθος των NORMAL ετικετών συμβολίζεται με N, το πλήθος των SLIGHTLY AGGRESSIVE ετικετών συμβολίζεται με SA, το πλήθος των FAIRLY AGGRESSIVE ετικετών συμβολίζεται με FA, το πλήθος των HIGHLY AGGRESSIVE ετικετών συμβολίζεται με HA και το πλήθος των EXTREMELY AGGRESSIVE ετικετών συμβολίζεται με EA. Η διαδρομή Δ αποτελείται από m χρονικά διαστήματα και επομένως ισχύει:

$$N + SA + FA + HA + EA = m \tag{7.10}$$

2. Υπολογίζονται οι στατιστικές μετρήσεις:

$$n = \frac{N}{m},\ sa = \frac{SA}{m},\ fa = \frac{FA}{m},\ ha = \frac{HA}{m},\ ea = \frac{EA}{m}, \\ n + sa + fa + ha + ea = 1 \tag{7.11}$$

Τελικά, έπειτα από την εφαρμογή του profiling αλγορίθμου η διαδρομή Δ ορίζεται ως εξής:

$$\Delta = (r, d, n, sa, fa, ha, ea) \tag{7.12}$$

Η παραπάνω εξίσωση ουσιαστικά εκφράζει ότι η διαδρομή με RoutedId r την οποία πραγματοποίησε ο οδηγός με DriverId d ήταν $n \cdot 100\%$ NORMAL, $sa \cdot 100\%$ SLIGHTLY AGGRESSIVE, $fa \cdot 100\%$ FAIRLY AGGRESSIVE, $ha \cdot 100\%$ HIGHLY AGGRESSIVE και $ea \cdot 100\%$ EXTREMELY AGGRESSIVE.

Θα πρέπει να σημειωθεί ότι όλη η διαδικασία που περιγράφηκε μέχρι τώρα αφορά το profiling μιας διαδρομής και όχι ένος χρήστη του συστήματος. Το profiling ενός χρήστη του συστήματος πραγματοποιείται με αρκετά παρόμοιο τρόπο. Έστω D το σύνολο των διαδρομών που έχει πραγματοποιήσει ένας χρήστης U:

$$D = \{\Delta_1, \Delta_2, ..., \Delta_k\} = \{(N_1, SA_1, FA_1, HA_1, EA_1), (N_2, SA_2, FA_2, HA_2, EA_2), ... \\ (N_k, SA_k, FA_k, HA_k, EA_k)\} \tag{7.13}$$

Ισχύει:

$$N_1 + SA_1 + FA_1 + HA_1 + EA_1 = m_1, ..., N_k + SA_k + FA_k + HA_k + EA_k = m_k \tag{7.14}$$

Σε αυτήν την περίπτωση οι δείκτες n, sa, fa, ha και ea υπολογίζονται ως εξής:

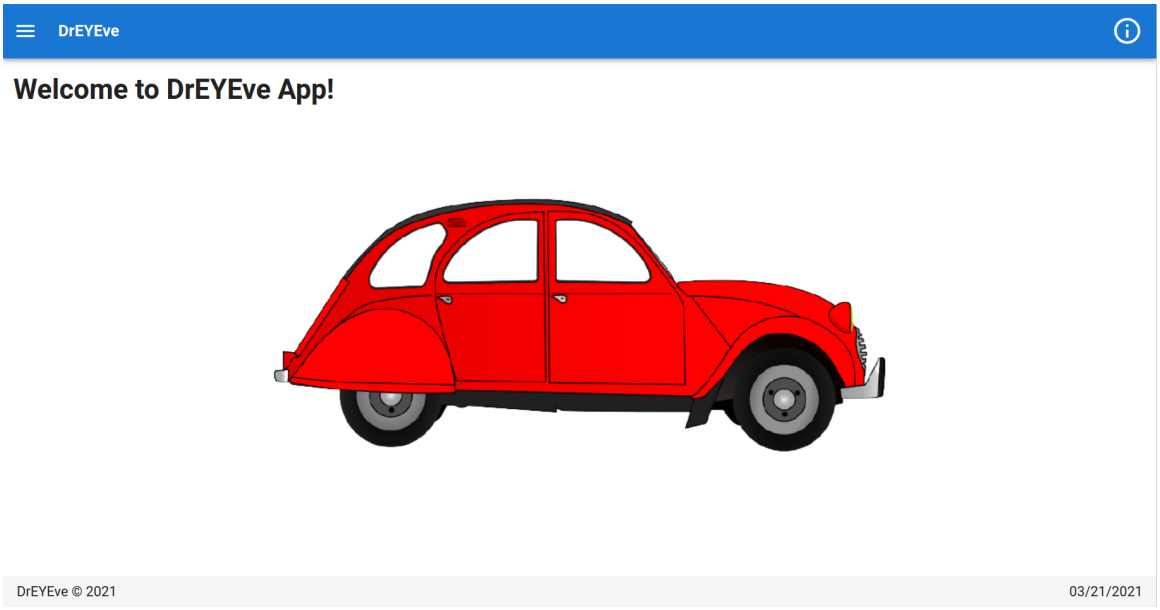
$$n = \frac{N_1 + N_2... + N_k}{m_1 + m_2 + ... + m_k},\ sa = \frac{SA_1 + SA_2... + SA_k}{m_1 + m_2 + ... + m_k},\ fa = \frac{FA_1 + FA_2... + FA_k}{m_1 + m_2 + ... + m_k}, \\ ha = \frac{HA_1 + HA_2... + HA_k}{m_1 + m_2 + ... + m_k},\ ea = \frac{EA_1 + EA_2... + EA_k}{m_1 + m_2 + ... + m_k},\ n + sa + fa + ha + ea = 1 \tag{7.15}$$

Έπειτα από την εφαρμογή του profiling αλγορίθμου ο χρήστης U ορίζεται ως εξής:

$$U = (d, n, sa, fa, ha, ea) \tag{7.16}$$

Η παραπάνω εξίσωση ουσιαστικά εκφράζει ότι ο χρήστης με DriverId d είναι $n \cdot 100\%$ NORMAL, $sa \cdot 100\%$ SLIGHTLY AGGRESSIVE, $fa \cdot 100\%$ FAIRLY AGGRESSIVE, $ha \cdot 100\%$ HIGHLY AGGRESSIVE και $ea \cdot 100\%$ EXTREMELY AGGRESSIVE.

7.6 Παρουσίαση του Frontend

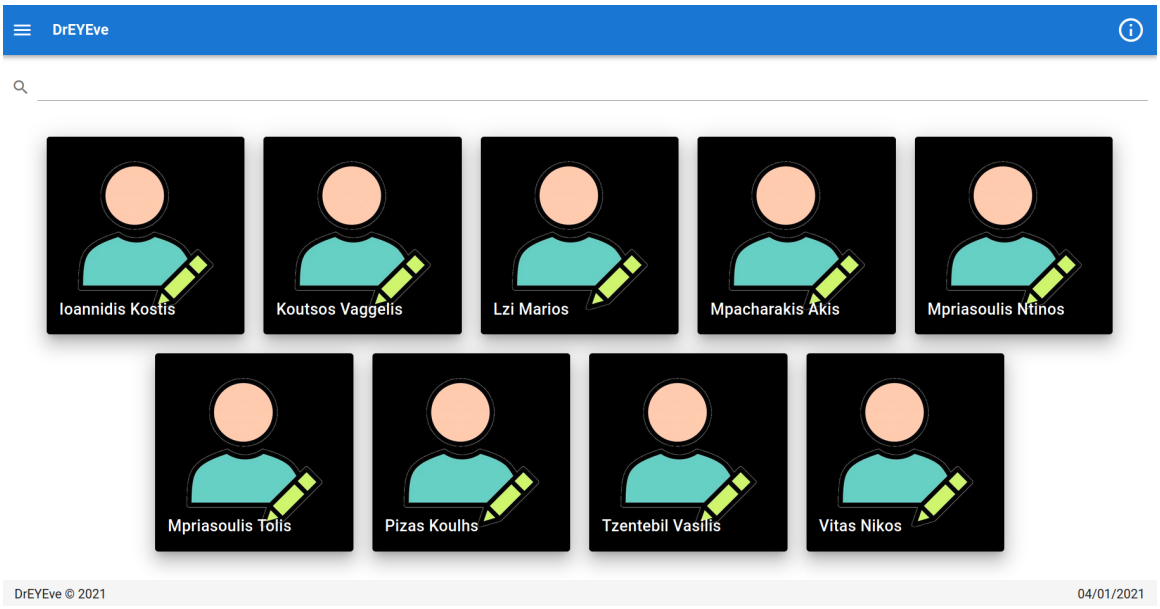


Σχήμα 7.51: Αρχική σελίδα του frontend

Μέχρι στιγμής στα υποκεφάλαια 7.3, 7.4 και 7.5 έχει περιγραφεί αναλυτικότατα ο τρόπος με τον οποίο το σύστημα χρησιμοποιεί τα οδηγικά δεδομένα που συλλέγονται προκειμένου να παράγει βαθμολογίες οι οποίες αντιπροσωπεύουν την κανονική ή την επιθετική συμπεριφορά που έχει ο εκάστοτε οδηγός του οχήματος. Σε αυτό το σημείο, λοιπόν, κρίθηκε απαραίτητη η δημιουργία ενός frontend, που συνοδεύει το σύστημα που έχει υλοποιηθεί, και επιτρέπει την αποτελεσματική οπτικοποίηση των δεδομένων που αποθηκεύονται στη σχεσιακή βάση δεδομένων σε πραγματικό χρόνο καθώς και την ιστορική αναδρομή αυτών. Ουσιαστικά, το frontend προσφέρει έναν απλό και ευανάγνωστο τρόπο αναπαράστασης των δεδομένων και δίνει τη δυνατότητα της αποδοτικής ερμηνείας τους από κάποιον άνθρωπο. Για την ανάπτυξη του frontend χρησιμοποιήθηκε εξ ολοκλήρου το Vue.js framework και οι βιβλιοθήκες που αυτό παρέχει, ενώ για την δημιουργία των διαφόρων γραφικών παραστάσεων που αυτό περιλαμβάνει χρησιμοποιήθηκε η βιβλιοθήκη Javascript Chart.js [111]. Στα υποκεφάλαια που ακολουθούν παρουσιάζονται οι βασικές σελίδες που παρέχει το frontend που υλοποιήθηκε.

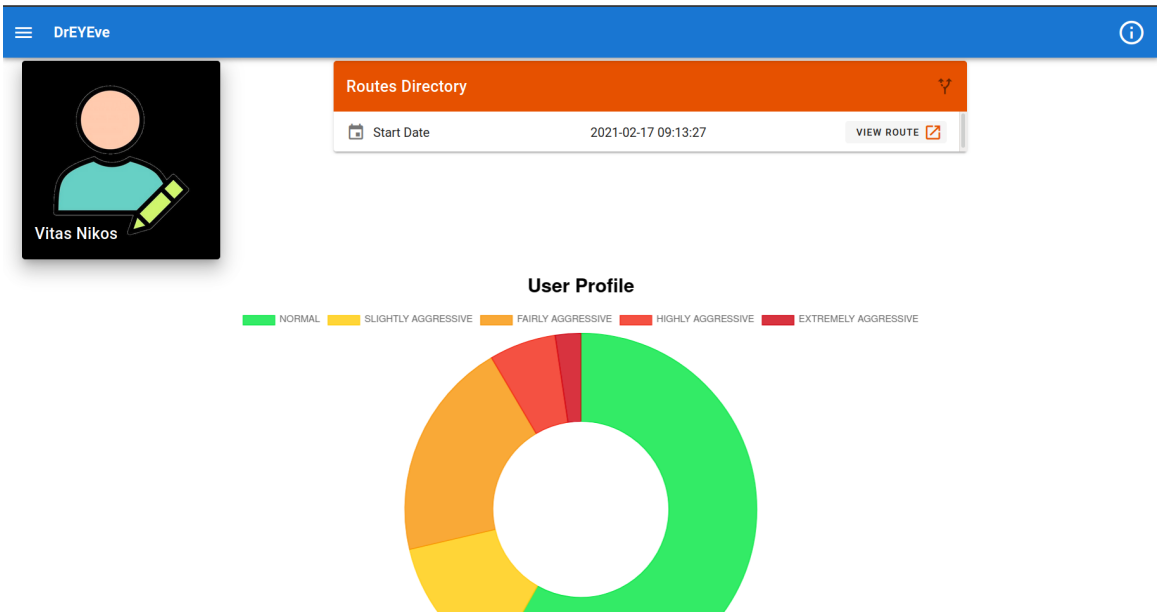
7.6.1 Η Σελίδα Users

Στη σελίδα Users απεικονίζονται όλοι οι χρήστες που έχουν γραφτεί στην εφαρμογή. Αυτό επιτυγχάνεται με την αποστολή ενός HTTP GET αιτήματος στο endpoint api/users του REST API του συστήματος. Όπως φαίνεται και από το σχήμα 7.52, στη σελίδα Users προσφέρεται η αναζήτηση κάποιου χρήστη με βάση το ονοματεπώνυμο.



Σχήμα 7.52: Η σελίδα Users του frontend

Συνεχίζοντας, κάνοντας κλικ στην εικόνα προφίλ κάποιου χρήστη δίνεται η δυνατότητα της ανάλυσης του οδηγικού προφίλ του συγκεκριμένου χρήστη. Τα δεδομένα που απαρτίζουν το προφίλ ενός χρήστη αποκτώνται μέσω της αποστολής ενός HTTP GET αιτήματος στο endpoint api/users/<user_id> του REST API του συστήματος.



Σχήμα 7.53: Οδηγικό προφίλ ενός χρήστη

Όπως φαίνεται και από το σχήμα 7.53, η σελίδα στην οποία παρουσιάζεται το οδηγικό προφίλ ενός χρήστη αποτελείται από δύο βασικά συστατικά, τα οποία είναι τα εξής:

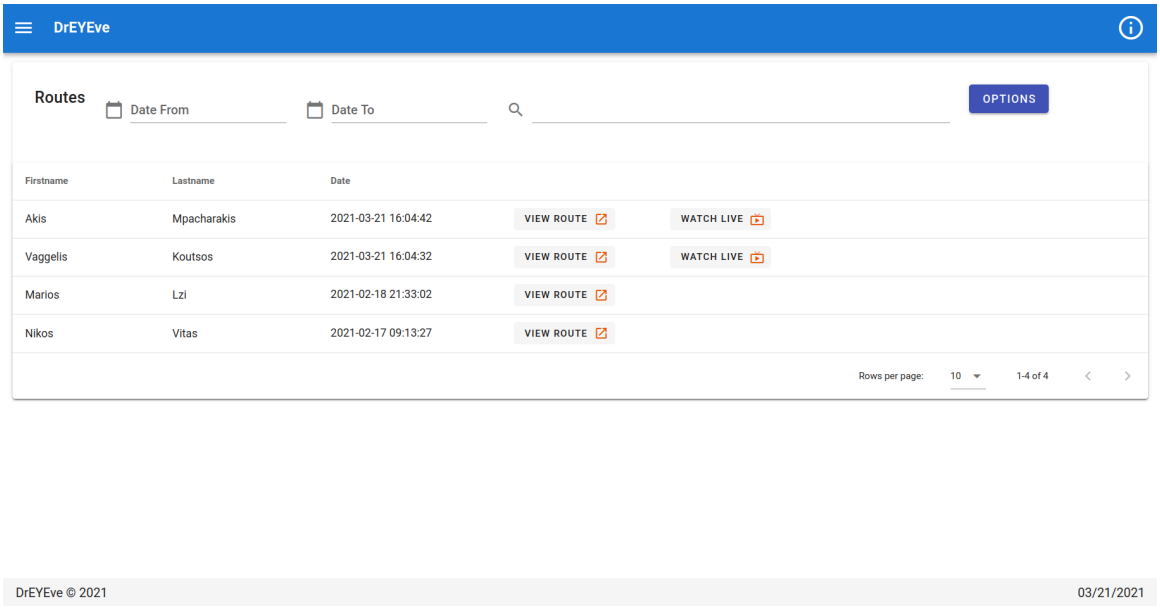
- Ένας πίνακας ο οποίος περιέχει όλες τις διαδρομές που έχει πραγματοποιήσει ο συγκεκριμένος χρήστης. Για κάθε μια από αυτές τις διαδρομές αναφέρεται η ημερομηνία κατά την οποία πραγματοποιήθηκε η εκκίνηση του οχήματος, ενώ

επίσης προσφέρεται και η δυνατότητα μετάβασης στη σελίδα της εκάστοτε διαδρομής, στην οποία παρουσιάζονται αναλυτικές πληροφορίες που αφορούν την διαδρομή. Η σελίδα μιας διαδρομής παρουσιάζεται αναλυτικά στο υποκεφάλαιο 7.6.2.

- Ένα γράφημα σε μορφή πίτας, στο οποίο παρουσιάζεται το οδηγικό προφίλ που έχει σχηματιστεί για το συγκεκριμένο χρήστη. Η πίτα ενός χρήστη αποτελεί έναν εύκολα αναγνώσιμο τρόπο για την αναπαράσταση των ποσοστών των διάφορων ετικετών που έχει συλλέξει ο χρήστης από το σύνολο των διαδρομών που έχει πραγματοποιήσει και οι οποίες συνιστούν το συνολικό προφίλ του χρήστη. Υπενθυμίζεται ότι οι ετικέτες αυτές είναι οι NORMAL, SLIGHTLY AGGRESSIVE, FAIRLY AGGRESSIVE, HIGHLY AGGRESSIVE και EXTREMELY AGGRESSIVE.

7.6.2 Η Σελίδα Routes

Στη σελίδα routes απεικονίζονται όλοι οι διαδρομές που έχουν πραγματοποιηθεί και έχουν αποθηκευτεί στο σύστημα, ανεξαρτήτως χρήστη. Αυτό επιτυγχάνεται με την αποστολή ενός HTTP GET αιτήματος στο endpoint api/routes του REST API του συστήματος. Όπως φαίνεται και από το σχήμα 7.54, στη σελίδα Routes προσφέρεται η δυνατότητα αναζήτησης μιας διαδρομής με βάση την ημερομηνία που αυτή πραγματοποιήθηκε καθώς και το όνομα του χρήστη που την πραγματοποίησε. Επιπλέον, μέσω ενός κουμπιού, που ονομάζεται OPTIONS, δίνεται η δυνατότητα της εύρεσης των διαδρομών που βρίσκονται σε εξέλιξη αυτή τη στιγμή.

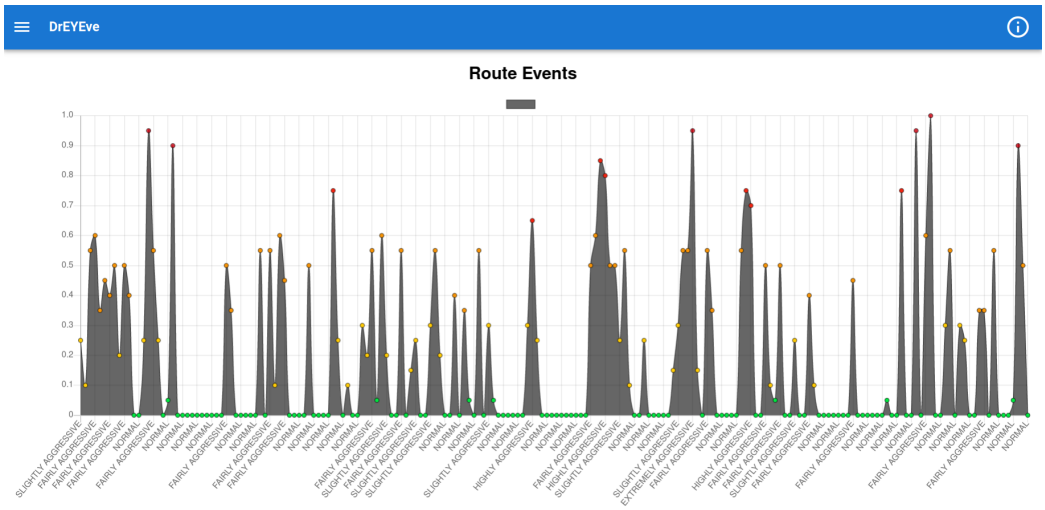


Σχήμα 7.54: Η σελίδα Routes του frontend

Συνεχίζοντας, στο σχήμα 7.54 φαίνεται ότι κάθε διαδρομή συνοδεύεται από ένα κουμπί, το οποίο ονομάζεται VIEW ROUTE. Κάνοντας κλικ σε αυτό το κουμπί, πραγματοποιείται η μετάβαση στη σελίδα που αφορά τη συγκεκριμένη διαδρομή και στην οποία παρουσιάζονται περισσότερες λεπτομέρειες για αυτή. Τα δεδομένα που αφορούν μια συγκεκριμένη διαδρομή αποκτώνται μέσω της αποστολής ενός HTTP GET αιτήματος στο endpoint api/routes/<route_id> του REST API του

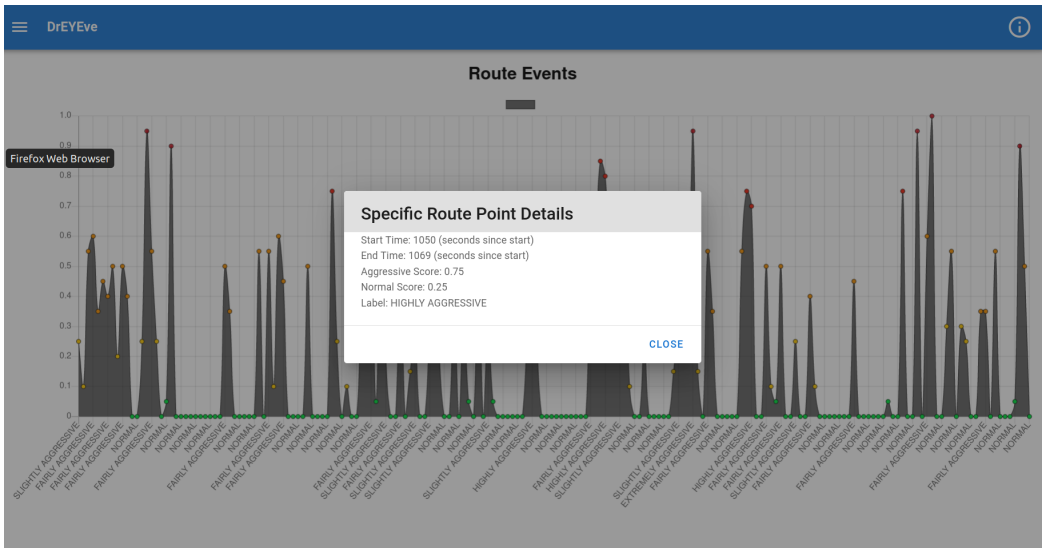
συστήματος. Η σελίδα στην οποία παρουσιάζεται αναλυτικά μια διαδρομή αποτελείται από τρία βασικά συστατικά, τα οποία είναι τα εξής:

- Μια γραφική παράσταση στην οποία απεικονίζονται οι βαθμολογίες που έχουν επιτευχθεί για τη συγκεκριμένη διαδρομή. Ουσιαστικά, μέσω αυτής της γραφικής παράστασης οπτικοποιούνται τα αποτελέσματα που παράγονται από τον αναλυτή δεδομένων. Στον άξονα x απεικονίζονται οι ετικέτες του πίνακα profiles της σχεσιακής βάσης δεδομένων που έχουν αποδοθεί στα διάφορα χρονικά διαστήματα της διαδρομής, ενώ στον άξονα y απεικονίζεται η τιμή του aggressive_score που αντιστοιχεί στο εκάστοτε χρονικό διάστημα. Προφανώς, όσο υψηλότερη είναι η τιμή του aggressive_score τόσο πιο επιθετική είναι η συμπεριφορά του οδηγού του οχήματος για εκείνο το διάστημα.



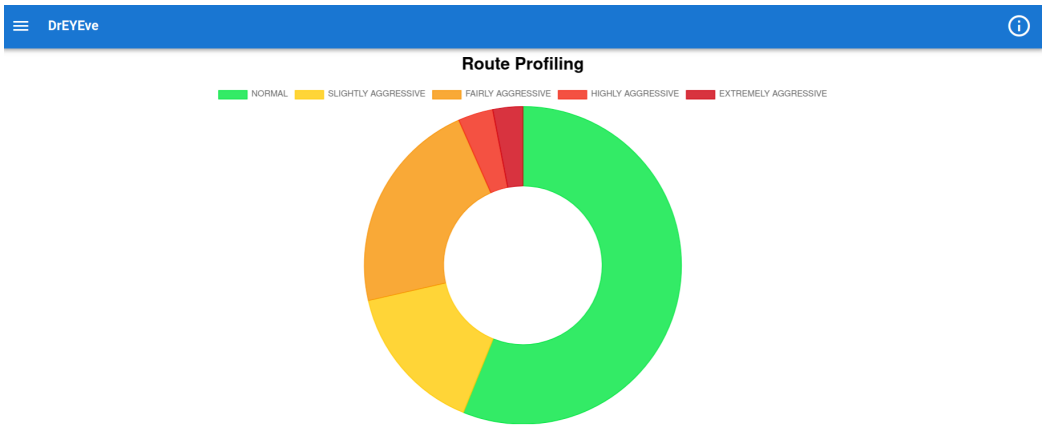
Σχήμα 7.55: Βαθμολογική απεικόνιση μιας διαδρομής

Ακόμη, κάνοντας κλικ σε κάποιο σημείο της παραπάνω γραφικής παράστασης δίνονται περισσότερες πληροφορίες σχετικά με τη συμπεριφορά του οδηγού για το χρονικό διάστημα που επιλέχτηκε.



Σχήμα 7.56: Πληροφορίες μεμονωμένου χρονικού διαστήματος μιας διαδρομής

- Ένα γράφημα σε μορφή πίτας, στο οποίο παρουσιάζεται το ποσοστό των διάφορων ετικετών κατηγοριοποίησης οδηγικής συμπεριφοράς που έχουν αποδοθεί στη συγκεκριμένη διαδρομή. Πρόκειται για ένα διάγραμμα το οποίο είναι παρόμοιο με αυτό που παρουσιάστηκε στη σελίδα του προφίλ ενός χρήστη, με τη μόνη διαφορά ότι προκύπτει μόνο από μια διαδρομή.

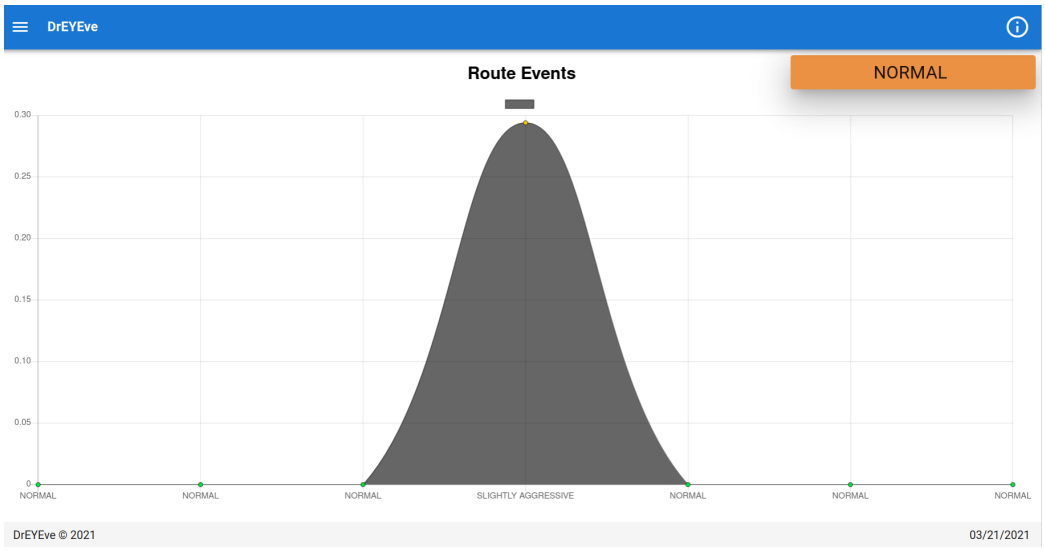


Σχήμα 7.57: Profiling μιας διαδρομής

- Ένα ραβδόγραμμα, στο οποίο παρουσιάζονται τα πλήθη των διάφορων ετικετών κατηγοριοποίησης οδηγικής συμπεριφοράς που έχουν αποδοθεί στη συγκεκριμένη διαδρομή.

Σχήμα 7.58: Πλήθη ετικετών κατηγοριοποίησης μιας διαδρομής

Επιστρέφοντας και πάλι στη σελίδα Routes του frontend, από το σχήμα 7.54 φαίνεται ότι κάποιες διαδρομές έχουν ένα κουμπί που ονομάζεται WATCH LIVE. Αυτό το κουμπί το έχουν μόνο οι διαδρομές οι οποίες βρίσκονται σε εξέλιξη αυτή τη στιγμή. Κάνοντας κλικ σε αυτό το κουμπί, πραγματοποιείται η μετάβαση στη σελίδα που αφορά τη παρακολούθηση της συγκεκριμένης διαδρομής σε πραγματικό χρόνο.



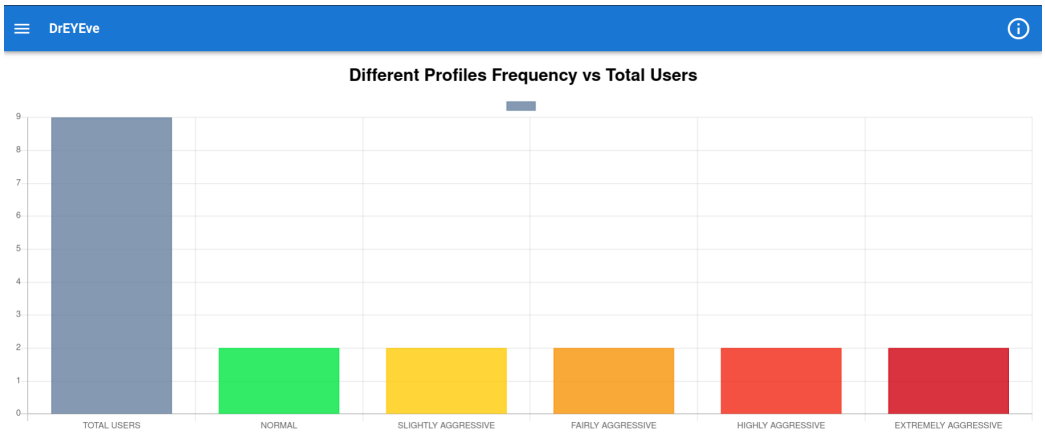
Σχήμα 7.59: Παρακολούθηση διαδρομής σε πραγματικό χρόνο

Η σελίδα αυτή αποτελείται από μια γραφική παράσταση παρόμοια με αυτή που παρουσιάστηκε παραπάνω. Η βασική διαφορά έγκειται στο γεγονός ότι η προηγούμενη γραφική παράσταση ήταν στατική ενώ αυτή σχηματίζεται δυναμικά καθώς τα δεδομένα εισέρχονται στο σύστημα. Με αυτόν τον τρόπο καθίστανται δυνατές τόσο η παρακολούθηση της εξέλιξης μιας διαδρομής όσο και η κατηγοριοποίηση της συμπεριφοράς του οδηγού σε πραγματικό χρόνο. Επιπλέον, στο πάνω δεξί μέρος της σελίδας παρέχεται μια ταμπέλα στην οποία κατονομάζεται η ετικέτα που έχει αποδοθεί στον οδηγό για τη συμπεριφορά που αυτός είχε για το τελευταίο χρονικό διάστημα που εξετάστηκε από το σύστημα. Προφανώς, όπως η γραφική παράσταση, έτσι και η ταμπέλα ενημερώνεται δυναμικά με βάση τα εισερχόμενα δεδομένα.

7.6.3 Η Σελίδα Statistics

Στη σελίδα Statistics απεικονίζονται διάφορα στατιστικά που προκύπτουν από το σύνολο των δεδομένων που έχουν αποθηκευτεί στο σύστημα. Πρόκειται για τα δεδομένα όλων των διαδρομών όλων των χρηστών του συστήματος. Αυτό επιτυγχάνεται με την αποστολή ενός HTTP GET αιτήματος στα endpoints api/stats/users και api/stats/routes του REST API του συστήματος. Η σελίδα Statistics αποτελείται από δύο βασικά συστατικά, τα οποία είναι τα εξής:

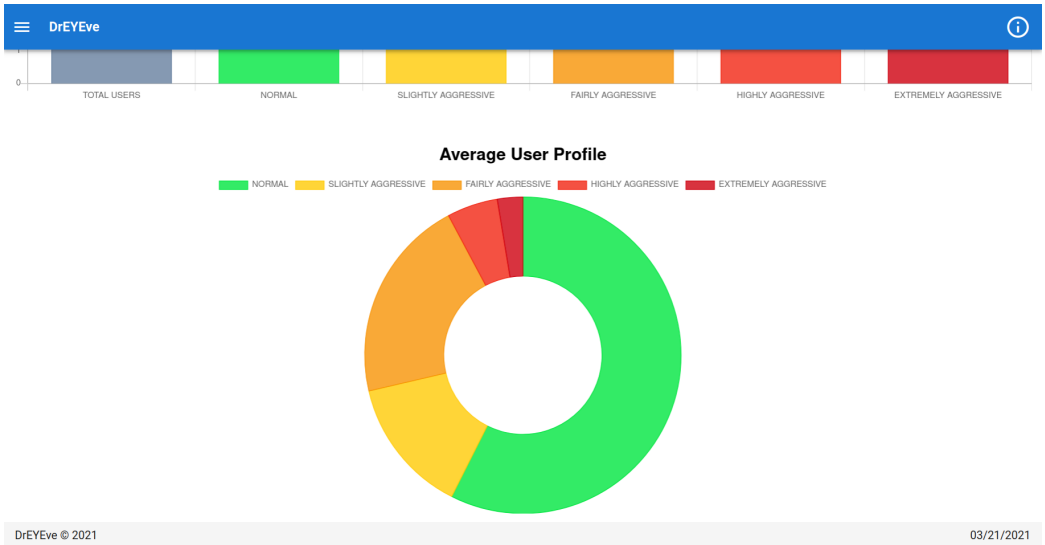
- Ένα ραβδόγραμμα, στο οποίο παρουσιάζονται τα πλήθη των διάφορων ετικετών κατηγοριοποίησης οδηγικής συμπεριφοράς που έχουν αποδοθεί στους διάφορους χρήστες του συστήματος σε σύγκριση με το συνολικό πλήθος αυτών. Πρόκειται για οπτικοποίηση των δεδομένων που επιστρέφονται από την κλήση στο endpoint api/stats/users του REST API.



Σχήμα 7.60: Στατιστικά χρηστών του συστήματος

Για παράδειγμα, στο σχήμα 7.60 φαίνεται ότι στο σύστημα έχουν κάνει εγγραφή 9 χρήστες συνολικά, ενώ για κάθε δυνατή ετικέτα οδηγικής συμπεριφοράς το πλήθος των χρηστών που τους έχει αποδοθεί αυτή η ετικέτα, στο πλαίσιο του συνολικότερου οδηγικού προφίλ των χρηστών, είναι ίσο με 2.

- Ένα γράφημα σε μορφή πίτας, στο οποίο παρουσιάζεται η συνολική αξιολόγηση όλων των χρηστών του συστήματος. Πρόκειται για ένα διάγραμμα το οποίο είναι παρόμοιο με αυτό που παρουσιάστηκε στη σελίδα του προφίλ ενός χρήστη, με τη μόνη διαφορά ότι προκύπτει από όλες τις διαδρομές που έχουν αποθηκευτεί στο σύστημα, ανεξαρτήτως χρήστη.



Σχήμα 7.61: Συνολική αξιολόγηση χρηστών του συστήματος

Κεφάλαιο 8: Επίλογος

8.1 Σύνοψη και Συμπεράσματα

Στόχος της παρούσης διπλωματικής εργασίας ήταν η ανάπτυξη ενός συστήματος συλλογής και ανάλυσης κυκλοφοριακών δεδομένων με σκοπό την εξαγωγή του οδηγικού προφίλ χρηστών που συμμετέχουν σε αυτό. Το σύστημα που υλοποιήθηκε αποτελείται από εργαλεία τα οποία εξειδικεύονται στην αποτελεσματική αποθήκευση και διαχείριση δεδομένων μεγάλου όγκου, ενώ με τη χρήση σύγχρονων μεθόδων μηχανικής μάθησης πραγματοποιείται η εξόρυξη γνώσης από τα κυκλοφοριακά δεδομένα που εισέρχονται στο σύστημα και τελικά αξιολογείται η οδηγική συμπεριφορά των χρηστών που είναι εγγεγραμμένοι σε αυτό.

Προκειμένου να γίνει πλήρως κατανοητή η χρησιμότητα του συστήματος που σχεδιάστηκε, στο κεφάλαιο 2 παρουσιάστηκε η έννοια των ευφυών συστημάτων μεταφορών, στο πλαίσιο λειτουργίας των οποίων εντάσσεται η δράση του συστήματος που υλοποιήθηκε σε αυτή τη διπλωματική εργασία. Τα ευφυή συστήματα μεταφορών, κάνοντας χρήση νέων τεχνολογιών και καινοτομιών, στοχεύουν στη βελτίωση του κυκλοφοριακού συστήματος. Σε αυτό το πλαίσιο, το σύστημα που υλοποιήθηκε προσφέρει τη δυνατότητα λήψης και επεξεργασίας κυκλοφοριακών δεδομένων σε πραγματικό χρόνο καθώς και την ιστορική αναδρομή αυτών, με σκοπό την ανάλυση της οδηγικής συμπεριφοράς και τον σχηματισμό οδηγικού προφίλ. Η ανάλυση των δεδομένων αυτών συμβάλλει άμεσα στην κατανόηση τόσο της συμπεριφοράς του οδηγού όσο και της κατάστασης του οχήματος και επομένως αποτελεί βασικό παράγοντα για τη συνολικότερη βελτίωση του οδικού κυκλοφοριακού συστήματος. Τα οδηγικά προφίλ που σχηματίζονται από το σύστημα δίνουν τη δυνατότητα στους χρήστες που συμμετέχουν σε αυτό να αξιολογήσουν τις ενέργειες που πραγματοποίησαν κατά την εξέλιξη μιας διαδρομής, να εντοπίσουν πιθανά λάθη και τελικά να τα διορθώσουν. Είναι προφανές ότι μια πιο σωστή οδηγική συμπεριφορά μπορεί να οδηγήσει όχι μόνο στην αποτροπή και τον περιορισμό των τροχαίων ατυχημάτων, αλλά και στη χαμηλότερη κατανάλωση καυσίμου.

Στη συνέχεια, στα κεφάλαια 3 και 4 παρουσιάστηκαν γνώσεις οι οποίες είναι απαραίτητες για την κατανόηση του τρόπου με τον οποίο το σύστημα επεξεργάζεται και διαχειρίζεται τα κυκλοφοριακά δεδομένα που συλλέγονται. Πιο συγκεκριμένα, στο κεφάλαιο 3 παρουσιάστηκε σε ένα θεωρητικό επίπεδο τόσο ο ορισμός της έννοιας των μεγάλων δεδομένων όσο και η μεθοδολογία που ακολουθείται για την ανακάλυψη και την εξόρυξη γνώσης από αυτά. Ιδιαίτερη έμφαση δόθηκε στην περιγραφή των μεθόδων που σχετίζονται με την προεπεξεργασία των δεδομένων με σκοπό την μετατροπή τους σε μορφή που εξυπηρετεί τους σκοπούς ανάλυσης τους. Αυτές οι μέθοδοι χρησιμοποιούνται και από το σύστημα που υλοποιήθηκε προκειμένου τα δεδομένα να μετασχηματιστούν σε μορφή που επιτρέπει την εξαγωγή γνώσης από αυτά και τελικά τον σχηματισμό οδηγικών προφίλ για τους χρήστες που συμμετέχουν στο σύστημα. Συνεχίζοντας, στο κεφάλαιο 4 παρουσιάστηκαν θεωρητικές γνώσεις που σχετίζονται με τον επιστημονικό κλάδο της μηχανικής μάθησης και τη διαχείριση χρονοσειρών. Το

σύστημα που υλοποιήθηκε χρησιμοποιεί μεθόδους μηχανικής μάθησης προκειμένου να εξάγει οδηγικά προφίλ από τα κυκλοφοριακά δεδομένα που συλλέγονται και για αυτό κρίθηκε απαραίτητη τόσο η επεξήγηση των βασικών εννοιών της μηχανικής μάθησης όσο και η παρουσίαση των πιο ευρέως χρησιμοποιούμενων μοντέλων κατηγοριοποίησης. Επιπλέον, καθώς τα κυκλοφοριακά δεδομένα που διαχειρίζεται το σύστημα ορίζουν πολυδιάστατες χρονοσειρές, κρίθηκε απαραίτητο να παρουσιαστούν οι βασικές έννοιες που σχετίζονται με τις χρονοσειρές καθώς και οι πιο δημοφιλείς τεχνικές που χρησιμοποιούνται για τη διαχείριση χρονοσειρών από μοντέλα μηχανικής μάθησης.

Στο κεφάλαιο 5 παρουσιάστηκε το σύνολο δεδομένων που χρησιμοποιήθηκε στην παρούσα διπλωματική εργασία, το οποίο αποτελείται από τέσσερις διαδρομές που πραγματοποίησαν δύο διαφορετικά οχήματα. Το σύνολο δεδομένων που χρησιμοποιήθηκε είναι κατάλληλο για την κατηγοριοποίηση της συμπεριφοράς του οδηγού ενός οχήματος βάσει του βαθμού επιθετικότητας των ενεργειών που αυτός πραγματοποιεί κατά την εξέλιξη μιας διαδρομής. Πιο συγκεκριμένα, το πεδίο βάσει του οποίου πραγματοποιείται η κατηγοριοποίηση της οδηγικής συμπεριφοράς ονομάζεται drivingStyle και μπορεί να λάβει δύο τιμές οι οποίες υποδηλώνουν επιθετική ή κανονική οδήγηση. Συνεχίζοντας, περιγράφηκαν τα υπόλοιπα πεδία των εγγραφών που απαρτίζουν μια διαδρομή έτσι ώστε να γίνει πλήρως κατανοητή η σημασία κάθε οδηγικής μέτρησης που έχει συλλεχθεί. Ακολούθως, παρουσιάστηκαν οι μέθοδοι προεπεξεργασίας που εφαρμόστηκαν στο σύνολο δεδομένων, οι οποίες διαφέρουν ανάλογα με το μοντέλο μηχανικής μάθησης που έχει επιλεχτεί να χρησιμοποιηθεί. Πιο συγκεκριμένα, πραγματοποιήθηκαν ενέργειες που αφορούσαν τη διαχείριση των απουσιάζουσων τιμών, την κωδικοποίηση και την κανονικοποίηση των δεδομένων καθώς και την επιλογή των σημαντικότερων πεδίων για την εξαγωγή αξιοποιήσιμης πληροφορίας από τα δεδομένα. Συνεχίζοντας, κάνοντας χρήση του πίνακα συσχετίσεων που ορίζουν τα πεδία του συνόλου δεδομένων εξάχθηκε το συμπέρασμα ότι τα πεδία AltitudeVariation και VehicleSpeedInstantaneous δεν επηρεάζουν άμεσα την κατηγοριοποίηση της οδηγικής συμπεριφοράς και για αυτό αποφασίστηκε να αφαιρεθούν από όλες τις εγγραφές του συνόλου δεδομένων. Τέλος, παρουσιάστηκε η τεχνική του κυλιόμενου παραθύρου η οποία χρησιμοποιήθηκε για την ομαδοποίηση των οδηγικών δεδομένων έτσι ώστε να εξασφαλιστεί η ταχύτατη και αποτελεσματική ανάλυση των δεδομένων που έχουν συλλεχθεί στο σύστημα.

Στο πλαίσιο υλοποίησης του συστήματος αυτής της διπλωματικής εργασίας αναζητήθηκε το αποτελεσματικότερο μοντέλο ανάλυσης κυκλοφοριακών δεδομένων και εξαγωγής συμπερασμάτων από αυτά. Για το σκοπό αυτό στο κεφάλαιο 6 δοκιμάστηκαν όλα τα μοντέλα μηχανικής μάθησης που παρουσιάστηκαν στο κεφάλαιο 4, ενώ στη συνέχεια αξιολογήθηκαν και συγκρίθηκαν τα αποτελέσματα που επιτεύχθηκαν από αυτά. Αρχικά, τα μοντέλα που εξετάστηκαν διακρίθηκαν σε δύο μεγάλες κατηγορίες οι οποίες ήταν τα απλά μοντέλα και τα μοντέλα χρονοσειρών. Για τα μοντέλα χρονοσειρών το σύνολο δεδομένων διαμορφώθηκε με εφαρμογή της τεχνικής του κυλιόμενου παραθύρου, για το οποίο βρέθηκε ότι, έπειτα από δοκιμή πολλών διαφορετικών τιμών, η βέλτιστη τιμή για το μέγεθος του είναι ίση με 20. Για όλα τα μοντέλα που εξετάστηκαν αναζητήθηκε η βέλτιστη παραμετροποίηση καθώς και ο συνδυασμός των μεθόδων προεπεξεργασίας που επιφέρουν τα καλύτερα αποτελέσματα. Συνεχίζοντας, ένα από τα σημαντικότερα συμπεράσματα που εξάχθηκε από τη σύγκριση των επιδόσεων των μοντέλων μηχανικής μάθησης που δοκιμάστηκαν ήταν

ότι η απόδοση των μοντέλων χρονοσειρών ήταν αρκετά υψηλότερη από αυτή που επιτεύχθηκε από τα απλά μοντέλα. Αυτό οφείλεται στη φύση του προβλήματος κατηγοριοποίησης που πρέπει να επιλυθεί το οποίο απαρτίζεται από δεδομένα τα οποία δεν είναι ανεξάρτητα μεταξύ τους αλλά ορίζουν χρονοσειρές δεδομένων μεταξύ των οποίων υπάρχουν σχέσεις εξάρτησης. Συνεπώς, αποφασίστηκε ότι το μοντέλο που θα ενσωματωθεί στο σύστημα θα είναι ένα μοντέλο χρονοσειρών. Τα μοντέλα χρονοσειρών που πέτυχαν τις υψηλότερες τιμές για τις μετρικές αξιολόγησης που χρησιμοποιήθηκαν ήταν τα LSTM, SVM με DTW και kNN με DTW. Ωστόσο, βρέθηκε ότι το μοντέλο LSTM μπορεί να πραγματοποιήσει περισσότερες προβλέψεις στην μονάδα του χρόνου από ότι τα μοντέλα SVM και kNN με DTW, γεγονός το οποίο είναι απαραίτητο σε οποιαδήποτε εφαρμογή καλείται να διαχειριστεί δεδομένα μεγάλου όγκου. Στη συνέχεια εξετάστηκαν οι πιο ευρέως χρησιμοποιούμενες παραλλαγές του LSTM μοντέλου προκειμένου να ελεγχθεί αν η διαφοροποίηση του τρόπου λειτουργίας του LSTM κελιού επηρεάζει σημαντικά την απόδοση του μοντέλου. Η σύγκριση των αποτελεσμάτων των διάφορων παραλλαγών του LSTM με το κλασσικό LSTM έδειξε ότι το κλασσικό LSTM παραμένει το αποτελεσματικότερο μοντέλο. Επομένως, η σύγκριση και η αξιολόγηση των αποτελεσμάτων των διάφορων μοντέλων που εξετάστηκαν ανέδειξαν το LSTM ως το αποτελεσματικότερο μοντέλο και έτσι αποφασίστηκε να χρησιμοποιηθεί αυτό για την υλοποίηση του συστήματος.

Κλείνοντας, στο κεφάλαιο 7 παρουσιάστηκε αναλυτικά ο τρόπος με τον οποίο διαμορφώνεται και λειτουργεί το σύστημα αξιολόγησης που υλοποιήθηκε σε αυτήν την διπλωματική εργασία. Αρχικά παρουσιάστηκαν τα προγραμματιστικά εργαλεία πάνω στα οποία βασίζεται η υλοποίηση του συστήματος και τα οποία εξειδικεύονται στην αποθήκευση, στην διαχείριση και στην επεξεργασία δεδομένων μεγάλου όγκου. Αυτά είναι τα Apache Spark, Apache Kafka και MongoDB. Η χρήση των εργαλείων αυτών αποδείχτηκε εξαιρετικά χρήσιμη καθώς δίνει στο σύστημα τη δυνατότητα να λειτουργήσει κατανεμημένα και να κλιμακωθεί εύκολα. Στη συνέχεια παρουσιάστηκαν τα προγραμματιστικά εργαλεία πάνω στα οποία βασίζεται η αποθήκευση, η ανάκτηση και η αναπαράσταση των δεδομένων που προκύπτουν από τη λειτουργία του συστήματος. Αυτά είναι τα PostgreSQL, Flask framework και Vue.js framework. Η χρήση των εργαλείων αυτών αποδείχτηκε εξαιρετικά χρήσιμη καθώς ήταν δυνατό να αναπτυχθεί μια πλήρης εφαρμογή εύκολα και με απλό τρόπο. Συνεχίζοντας, στο κεφάλαιο 7 περιγράφηκε ο τρόπος με τον οποίο το σύστημα ενσωματώνει τα εργαλεία αυτά και διαχειρίζεται τα κυκλοφοριακά δεδομένα που έχουν συλλεχθεί από τους αισθητήρες διάφορων οχημάτων. Παράλληλα, μελετήθηκε η πορεία που ακολουθούν τα δεδομένα από τη συλλογή τους και την αποθήκευσή τους σε βάσεις δεδομένων μεγάλου όγκου, την προεπεξεργασία και τον μετασχηματισμό τους σε μορφή που εξυπηρετεί τις ανάγκες ανάλυσης τους με παράλληλο και κατανεμημένο τρόπο, την εξαγωγή και την αποθήκευση των συμπερασμάτων που προκύπτουν από αυτά μέχρι και την αποδοτική ανάκτηση και αναπαράσταση αυτών. Τέλος, εκτελέστηκε προσομοίωση της λειτουργίας του συστήματος σε ρεαλιστικές συνθήκες με δύο από τις διαδρομές του συνόλου δεδομένων γεγονός το οποίο επαλήθευσε την επιτυχή διασύνδεση των συστατικών που το απαρτίζουν καθώς και την ορθή λειτουργία καθενός από αυτά ξεχωριστά.

8.2 Μελλοντικές Προεκτάσεις

Στο υποκεφάλαιο αυτό παρουσιάζονται μερικές μελλοντικές προεκτάσεις του συστήματος που υλοποιήθηκε οι οποίες πιστεύεται ότι θα μπορούσαν να οδηγήσουν σε ενδιαφέροντα αποτελέσματα. Οι προτάσεις αυτές σχετίζονται με τα μοντέλα μηχανικής μάθησης, με την προεπεξεργασία του συνόλου δεδομένων και τη δομή του ίδιου του συστήματος.

Η πρώτη πιθανή προέκταση που προτείνεται αφορά τη διερεύνηση και την ανάπτυξη ακόμη πιο σύνθετων μοντέλων για την ανάλυση των κυκλοφοριακών δεδομένων που συλλέγονται. Στην παρούσα διπλωματική εργασία εξετάστηκαν συνολικά δέκα μοντέλα, πέντε από τα οποία δέχονταν ως είσοδο δεδομένα που είχαν μορφή χρονοσειράς και πέντε τα οποία τροφοδοτούνταν με είσοδο μεμονωμένες εγγραφές δεδομένων. Επιπλέον, εξετάστηκαν και τρεις από τις πιο δημοφιλείς παραλλαγές του LSTM μοντέλου. Από την εκπόνηση αυτής της διπλωματικής εργασίας προέκυψε το συμπέρασμα ότι τα μοντέλα χρονοσειρών είναι πιο κατάλληλα για την επίλυση του προβλήματος της κατηγοριοποίησης της οδηγικής συμπεριφοράς των χρηστών που συμμετέχουν στο σύστημα που υλοποιήθηκε. Συνεπώς, θα παρουσίαζε ιδιαίτερο ενδιαφέρον να δοκιμαστούν κάποιες ακόμη μέθοδοι και μοντέλα τα οποία εξειδικεύονται σε δεδομένα που έχουν μορφή χρονοσειράς. Πιο συγκεκριμένα, προτείνεται η δοκιμή της μεθόδου COTE (Collection of Transformation Ensembles) [112] και ενός LSTM-FCN (Long Short Term Memory - Fully Convolutional Network) [113] μοντέλου. Επιπλέον, προτείνεται η δοκιμή της συνδυαστικής λειτουργίας των αποτελεσματικότερων μοντέλων που παρουσιάστηκαν στο κεφάλαιο 6. Για παράδειγμα, θα μπορούσαν να χρησιμοποιηθούν τα μοντέλα LSTM, SVM με DTW και kNN με DTW, τα οποία είχαν πετύχει τις υψηλότερες τιμές για τις μετρικές αξιολόγησης που χρησιμοποιήθηκαν, προκειμένου να δημιουργήσουν ένα σύνθετο ενισχυτικό μοντέλο το οποίο μπορεί να οδηγήσει σε προβλέψεις μεγαλύτερης ακρίβειας.

Η δεύτερη πιθανή προέκταση που προτείνεται σχετίζεται με την προσθήκη ενός φίλτρου Kalman από το οποίο θα περνάνε τα δεδομένα που εισέρχονται στο σύστημα. Αν και το θεωρητικό υπόβαθρο που απαιτείται για την κατανόηση της λειτουργίας του φίλτρου Kalman παρουσιάστηκε με μεγάλη λεπτομέρεια στο υποκεφάλαιο 4.2.3, ένα τέτοιο φίλτρο δεν ενσωματώθηκε στη λειτουργία του τελικού συστήματος που υλοποιήθηκε. Αυτή η επιλογή έγινε συνειδητά, καθώς οι τιμές των πεδίων του συνόλου δεδομένων που σχετίζονται με την επιτάχυνση του οχήματος είχαν ήδη περάσει από ένα βαθυπερατό φίλτρο έτσι ώστε να αφαιρεθεί πιθανός θόρυβος που αυτές μπορεί να περιέχουν. Ωστόσο, η χρήση του φίλτρου Kalman είναι ευρέως διαδεδομένη στις εφαρμογές που σχετίζονται με τα ευφυή συστήματα μεταφορών και θα είχε ιδιαίτερο ενδιαφέρον να συγκριθούν τα αποτελέσματα που παράγει το σύστημα με τη χρήση μόνο ένος βαθυπερατού φίλτρου με τα αποτελέσματα που θα παράγονταν αν χρησιμοποιούταν επιπλέον και ένα φίλτρο Kalman για την εξομάλυνση των δεδομένων της επιτάχυνσης. Καθώς όλος ο κώδικας που χρησιμοποιήθηκε για την υλοποίηση του συστήματος είναι γραμμένος σε python προτείνεται η χρήση της βιβλιοθήκης pykalman [114].

Η τρίτη και τελευταία πιθανή προέκταση που προτείνεται σχετίζεται με την κατανεμημένη λειτουργία του συστήματος. Στο πλαίσιο υλοποίησης της παρούσης διπλωματικής εργασίας όλα τα συστατικά που απαρτίζουν το σύστημα φιλοξενήθηκαν

στον ίδιο υπολογιστικό κόμβο. Ωστόσο, χρησιμοποιήθηκαν προγραμματιστικά εργαλεία, όπως τα Apache Spark, Apache Kafka και MongoDB, τα οποία επιτρέπουν τόσο την εύκολη κλιμάκωση όσο και την κατανεμημένη λειτουργία του συστήματος. Προφανώς, σε πραγματικές συνθήκες λειτουργίας ένα σύστημα αξιολόγησης οδηγικής συμπεριφοράς, σαν αυτό που υλοποιήθηκε σε αυτή τη διπλωματική εργασία, θα καλούταν να ανταποκριθεί και να διαχειριστεί τεράστιο πλήθος δεδομένων. Για παράδειγμα, αν υπήρχαν 1000 χρήστες στο σύστημα και όλοι εκτελούσαν μια διαδρομή ταυτόχρονα, στο σύστημα θα κατέφθαναν 1000 εγγραφές οδηγικών δεδομένων κάθε δευτερόλεπτο. Συνεπώς, ιδιαίτερο ενδιαφέρον θα παρουσίαζε να πραγματοποιηθεί μια προσαρμογή του κώδικα του συστήματος που υλοποιήθηκε έτσι ώστε να τρέχει σε συστάδες (clusters) κόμβων και στη συνέχεια να πραγματοποιηθεί μια σύγκριση των χρόνων απόκρισης του κατανεμημένου αυτού συστήματος με το σύστημα που υλοποιήθηκε στην παρούσα διπλωματική εργασία, εάν και τα δύο συστήματα τροφοδοτούνταν με μεγάλο όγκο κυκλοφοριακών δεδομένων. Επιπλέον, ιδιαίτερα χρήσιμη θα μπορούσε να αποδειχθεί και η ανάπτυξη κώδικα που σχετίζεται με την κατανεμημένη εκπαίδευση του μοντέλου μηχανικής μάθησης που χρησιμοποιείται από το σύστημα. Στην περίπτωση του συγκεκριμένου συστήματος δεν ήταν δυνατό να χρησιμοποιηθεί η βιβλιοθήκη MLlib του Apache Spark για αυτό το σκοπό, η οποία παρουσιάστηκε στο υποκεφάλαιο 7.1.2.1, καθώς το μοντέλο LSTM δεν περιέχεται σε αυτήν. Συνεπώς, για την εκπαίδευση του μοντέλου LSTM με κατανεμημένο τρόπο μέσω του Apache Spark προτείνεται η χρήση της πλατφόρμας Elephas [115] η οποία επιτρέπει την παράλληλη εκπαίδευση μοντέλων της βιβλιοθήκης tensorflow σε ένα κατανεμημένο σύστημα που χρησιμοποιεί το Apache Spark.

Βιβλιογραφία

[1] "Road traffic injuries." https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 07/02/2020.

[2] "Cars, planes, trains: where do co2 emissions from transport come from." https://ourworldindata.org/co2-emissions-from-transport, 06/10/2020.

[3] A. Broggi, A. Zelinsky, M. Parent, and C. E. Thorpe, "Intelligent vehicles," pp. 1175–1198, 2008.

[4] M. Osman, A. Hussein, and A. Al-Kaff, "Intelligent vehicles localization approaches between estimation and information: A review," pp. 1–8, 09 2019.

[5] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," pp. 1094–1099, 06 2015.

[6] L. Baskar, B. De Schutter, and H. Hellendoorn, "Model predictive control for intelligent speed adaptation in intelligent vehicle highway systems," pp. 468 – 473, 10 2008.

[7] S. Cheon, "An Overview of Automated Highway Systems (AHS) and the Social and Institutional Challenges They Face," University of California Transportation Center, Working Papers qt8j86h0cj, University of California Transportation Center, Mar. 2003.

[8] J. Hedrick, M. Tomizuka, and P. Varaiya, "Control issues in automated highway systems," vol. 14, pp. 21 – 32, 01 1995.

[9] D. N. Godbole, F. H. Eskafi, and P. P. Varaiya, "Automated highway systems*," vol. 29, no. 1, pp. 5506–5511, 1996. 13th World Congress of IFAC, 1996, San Francisco USA, 30 June - 5 July.

[10] "A brief history of database management." https://www.dataversity.net/brief-history-database-management, 23/03/2017.

[11] "How charles bachman invented the dbms, a foundation of our digital world." https://cacm.acm.org/magazines/2016/7/204036-how-charles-bachman-invented-the-dbms-a-foundation-of-our-digital-world/fulltext, 07/2016.

[12] "Information management system." https://www.ibm.com/ibm/history/ibm100/us/en/icons/ib

[13] "Relational database." https://www.ibm.com/ibm/history/ibm100/us/en/icons/reldb/.

[14] "What is the relationship between iot and big data." https://www.soracom.io/blog/what-is-the-relationship-between-iot-and-big-data/, 17/09/2017.

[15] "Number of facebook users worldwide from 2015 to 2020." https://www.statista.com/statistics/490424/number-of-worldwide-facebook-users/, 15/11/2019.

[16] "Wild and interesting facebook statistics and facts." https://kinsta.com/blog/facebook-statistics/, 06/11/2020.

[17] "A btief history of nosql." http://blog.knuthaugen.no/2010/03/a-brief-history-of-nosql.html, 16/03/2010.

[18] "How do hadoop and spark stack up?." https://logz.io/blog/hadoop-vs-spark/, 16/01/2020.

[19] "Gartner." https://www.gartner.com/en/information-technology/glossary/big-data/.

[20] A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, and U. Saxena, "Nosql databases: Critical analysis and comparison," pp. 293–299, 10 2017.

[21] "Cap theorem." https://www.ibm.com/cloud/learn/cap-theorem.

[22] R. Cheruvu, "Big data applications in self-driving cars," 12 2015.

[23] S. Arumugam and R. Bhargavi, "A survey on driving behavior analysis in usage based insurance using big data," vol. 6, 09 2019.

[24] "Tesla." https://www.tesla.com/autopilot.

[25] O. Maimon and L. Rokach, "Chapter 1 - introduction to knowledge discovery in databases, the data mining and knowledge discovery handbook," pp. 1–17, 01 2005.

[26] "Data preprocessing." http://www.itu.dk/, 13/11/2015.

[27] "Principal component analysis." https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html, 05/09/2016.

[28] M. Bharati and B. Ramageri, "Data mining techniques and applications, indian journal of computer science and engineering," vol. 1, 12 2010.

[29] "Convolutional neural networks for visual recognition." https://cs231n.github.io/neural-networks-1, 2020.

[30] "The intellectual excitement of computer science.." https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Architecture/feedforward.html, 2000.

[31] "Recurrent neural networks." https://stanford.edu/ shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks, 2019.

[32] "What is deep learning." https://machinelearningmastery.com/what-is-deep-learning/, 14/08/2020.

[33] "Neural networks - learning." https://doc.plob.org/machine_learning/09_Neural_Networks_L 2011.

[34] "Regularization." http://ethen8181.github.io/machine-learning/regularization/regularization.html, 19/12/2017.

[35] "Gradient descent." https://ml-cheatsheet.readthedocs.io/en/latest/gradient_descent.html.

[36] "Difference between a batch and an epoch in a neural network." https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/, 26/10/2019.

[37] "Optimizers." https://ml-cheatsheet.readthedocs.io/en/latest/optimizers.html.

[38] "K-nearest-neighbor." https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/, 13/07/2016.

[39] "A top machine learning algorithm explained: Support vector machines (svm)." https://www.kdnuggets.com/2020/03/machine-learning-algorithm-svm-explained.html, 03/2020.

[40] "Scikit-learn." https://scikit-learn.org/stable/modules/svm.html.

[41] "Soft margin svm." http://fourier.eng.hmc.edu/e161/lectures/svm/node5.html, 19/08/2016.

[42] "Random forests." https://jakevdp.github.io/PythonDataScienceHandbook/05.08-random-forests.html, 05/08/2016.

[43] "Decision tree algorithm." https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html, 01/2020.

[44] "Xgboost." https://xgboost.readthedocs.io/en/latest/.

[45] T. Chen and C. Guestrin, "Xgboost, proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining," Aug 2016.

[46] "How to explain gradient boosting." https://explained.ai/gradient-boosting/index.html, 06/2018.

[47] "Understanding lstm networks." https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 27/08/2015.

[48] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning." https://d2l.ai/chapter_recurrent-modern/bi-rnn.html, 2020.

[49] H. Jaeger, "The" echo state" approach to analysing and training recurrent neural networks-with an erratum note', bonn, germany: German national research center for information technology gmd technical report," vol. 148, 01 2001.

[50] A. Goudarzi and C. Teuscher, "Reservoir computing: Quo vadis?," pp. 1–6, 09 2016.

[51] A. Abdelrahman, H. Hassanein, and N. Abu Ali, "Data-driven robust scoring approach for driver profiling applications," 12 2018.

[52] D. Alvarez-Coello, B. Klotz, D. Wilms, S. Fejji, J. Marx-Gomez, and R. Troncy, "Modeling dangerous driving events based on in-vehicle data using random forest and recurrent neural network," 06 2019.

[53] "Driver behavior dataset." https://github.com/jair-jr/driverBehaviorDataset.

[54] J. Ferreira, E. Carvalho, B. Ferreira, C. Souza, Y. Suhara, A. Pentland, and G. Pessin, "Driver behavior profiling: An investigation with different smartphone sensors and machine learning, plos one," vol. 12, pp. 1–16, 04 2017.

[55] N. Karginova, S. Byttner, and M. Svensson, "Data-driven methods for classification of driving styles in buses," 04 2012.

[56] J. Grandell, "Time series analysis." http://www.math.kth.se/matstat/gru/5b1545/ts.pdf.

[57] J. Large, P. Southam, and A. Bagnall, "Can automated smoothing significantly improve benchmark time series classification algorithms?," pp. 50–60, 08 2019.

[58] Z. z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification, sigkdd explorations," vol. 12, pp. 40–48, 11 2010.

[59] G. A. Susto, A. Cenedese, and M. Terzi, "Time-series classification methods: Review and applications to power systems data, big data application in power systems," pp. 179–220, 01 2018.

[60] B. Esmael, A. Arnaout, R. Fruhwirth, and G. Thonhauser, "Multivariate time series classification by combining trend-based and value-based approximations," vol. 7336, 06 2012.

[61] A. Nanopoulos, R. Alcock, and Y. Manolopoulos, "Feature-based classification of time-series data, international journal of computer research," vol. 10, pp. 49–61, 01 2001.

[62] "Dynamic time warping." https://www.audiolabs-erlangen.de/resources/MIR/FMP/C3/C3S2_DTWbasic.html, 2015.

[63] S. Seto, W. Zhang, and Y. Zhou, "Multivariate time series classification using dynamic time warping template selection for human activity recognition," pp. 1399–1406, 12 2015.

[64] R. Kate, "Using dynamic time warping distances as features for improved time series classification, data mining and knowledge discovery," vol. 30, 05 2015.

[65] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," pp. 2–11, 01 2003.

[66] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining," pp. 947–956, 06 2009.

[67] A. Bostrom and A. Bagnall, "A shapelet transform for multivariate time series classification," 2017.

[68] M. Lin, J. Yoon, and B. Kim, "Self-driving car location estimation based on a particle-aided unscented kalman filter," vol. 20, p. 2544, 04 2020.

[69] C. Kownacki, "Optimization approach to adapt kalman filters for the real-time application of accelerometer and gyroscope signal' filtering," vol. 21, pp. 131–140, 01 2011.

[70] Y. Kim and H. Bang, "Introduction to kalman filter and its applications," 11 2018.

[71] "Gaussian distribution function." http://hyperphysics.phy-astr.gsu.edu/hbase/Math/gaufcn.html, 30/06/2020.

[72] M. Ruta, F. Scioscia, G. Loseto, A. Pinto, and E. Di Sciascio, "Machine learning in the internet of things: A semantic-enhanced approach," vol. 10, pp. 1–22, 08 2018.

[73] "Traffic, driving style and road surface condition dataset kaggle." https://www.kaggle.com/gloseto/traffic-driving-style-road-surface-condition.

[74] "Python." https://www.python.org/.

[75] "Python pandas." https://pandas.pydata.org/.

[76] "Python seaborn." https://seaborn.pydata.org/.

[77] "Accuracy." https://developers.google.com/machine-learning/crash-course/classification/accuracy, 10/02/2020.

[78] "Classification: True vs false and positive vs negative." https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative, 10/02/2020.

[79] T. Fawcett, "Introduction to roc analysis, pattern recognition letters," vol. 27, pp. 861–874, 06 2006.

[80] "Python xgboost." https://xgboost.readthedocs.io/en/latest/python/python_intro.html.

[81] "Python easyesn." https://github.com/kalekiu/easyesn.

[82] "Python tensorflow." https://www.tensorflow.org/learn.

[83] "Python scikit-learn." https://scikit-learn.org/stable/.

[84] "Python tslearn." https://tslearn.readthedocs.io/en/stable/.

[85] "Python sklearn grid search." https://scikit-learn.org/stable/modules/generated/sklearn.model_Fselection.GridSearchCV.html.

[86] V. Apostolidis-Afentoulis, "Svm classification with linear and rbf kernels," 07 2015.

[87] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui, "A kernel for time series based on global alignments," pp. II–413, 05 2007.

[88] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets," p. 392–401, 2014.

[89] "Python matplotlib." https://matplotlib.org/.

[90] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014.

[91] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," pp. 338–342, 01 2014.

[92] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," 2017.

[93] "Mongodb." https://docs.mongodb.com/manual/.

[94] "Apache spark." https://spark.apache.org/docs/latest.

[95] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," pp. 2–2, 04 2012.

[96] "Apache kafka." https://kafka.apache.org/documentation/.

[97] "Apache zookeeper." https://zookeeper.apache.org/doc/r3.6.2/index.html.

[98] "Postgresql." https://www.postgresql.org/docs/current/.

[99] E. F. Codd, "A relational model of data for large shared data banks," vol. 13, p. 377–387, June 1970.

[100] "Python flask." https://flask.palletsprojects.com/en/1.1.x/.

[101] "Restful api." https://restfulapi.net/.

[102] "Http." https://tools.ietf.org/html/rfc2616.

[103] "Json." https://www.json.org/json-en.html.

[104] "Vue.js." https://vuejs.org/v2/guide/.

[105] "Vuetify." https://vuetifyjs.com/en/getting-started/installation/.

[106] "Python pymongo." https://pymongo.readthedocs.io/en/stable/.

[107] "Python kafka-python." https://kafka-python.readthedocs.io/en/master/compatibility.html.

[108] "Python pyspark." https://spark.apache.org/docs/latest/api/python/.

[109] "Python sqlalchemy." https://www.sqlalchemy.org/.

[110] "Mongodb kafka connector." https://docs.mongodb.com/kafka-connector/current/.

[111] "Javascript chart.js." https://www.chartjs.org/.

[112] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with cote: The collective of transformation-based ensembles," vol. 27, pp. 1–1, 09 2015.

[113] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate lstm-fcns for time series classification," vol. 116, 01 2018.

[114] "Python pykalman." https://pykalman.github.io/.

[115] "Elephas." https://github.com/maxpumperla/elephas.