



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΙΩΑΝΝΙΝΩΝ

ΣΧΟΛΗ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΠΜΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΚΤΥΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΣΧΕΔΙΑΣΜΟΣ ΚΑΙ ΑΝΑΠΤΥΞΗ ΣΥΣΤΗΜΑΤΟΣ ΠΡΟΒΛΕΨΗΣ
ΔΙΑΘΕΣΙΜΟΤΗΤΑΣ ΧΩΡΟΥ ΣΤΑΘΜΕΥΣΗΣ ΣΕ ΠΛΑΤΦΟΡΜΑ
ΠΑΡΑΛΛΗΛΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ

Φλώρου Αικατερίνη

Επιβλέπων: Στύλιος Χρυσόστομος

ΔΕΠ Καθηγητής

Άρτα, Μάρτιος, 2024

**DESIGN AND DEVELOPMENT OF A PARKING SPACE AVAILABILITY
PREDICTION SYSTEM ON A PARALLEL PROCESSING PLATFORM**

Εγκρίθηκε από τριμελή εξεταστική επιτροπή

Άρτα, 15/03/2024

ΕΠΙΤΡΟΠΗ ΑΞΙΟΛΟΓΗΣΗΣ

1. Επιβλέπων καθηγητής

Στύλιος Χρυσόστομος,

ΔΕΠ Καθηγητής

2. Μέλος επιτροπής

Γκόγκος Χρήστος,

ΔΕΠ Αναπληρωτής Καθηγητής

3. Μέλος επιτροπής

Τσούλος Ιωάννης,

ΔΕΠ Αναπληρωτής Καθηγητής

© Φλώρου, Αικατερίνη, 2024.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Δήλωση μη λογοκλοπής

Δηλώνω υπεύθυνα και γνωρίζοντας τις κυρώσεις του Ν. 2121/1993 περί Πνευματικής Ιδιοκτησίας, ότι η παρούσα μεταπτυχιακή εργασία είναι εκ ολοκλήρου αποτέλεσμα δικής μου ερευνητικής εργασίας, δεν αποτελεί προϊόν αντιγραφής ούτε προέρχεται από ανάθεση σε τρίτους. Όλες οι πηγές που χρησιμοποιήθηκαν (κάθε είδους, μορφής και προέλευσης) για τη συγγραφή της περιλαμβάνονται στη βιβλιογραφία.

Φλώρου, Αικατερίνη



ΠΕΡΙΛΗΨΗ

Στην παρούσα διπλωματική εξετάζεται η πρόβλεψη αστικών χώρων στάθμευσης, ζωτικής σημασίας για τη διαχείριση της κυκλοφορίας, μέσω προηγμένων τεχνολογιών μηχανικής μάθησης, βαθιάς μάθησης και αυτοματοποιημένης μηχανικής μάθησης (AutoML), ενώ ενσωματώνεται και η τεχνολογία federated learning. Οι τεχνικές μηχανικής μάθησης αποκωδικοποιούν την πολυπλοκότητα της πρόβλεψης της διαθεσιμότητας στάθμευσης, διακρίνοντας μοτίβα μέσα σε εκτεταμένα σύνολα δεδομένων για την ενίσχυση της κατανόησης και της πρόβλεψης σε μεγάλες πόλεις.

Η βαθιά μάθηση, ένα υποσύνολο της μηχανικής μάθησης, αυξάνει τις δυνατότητες πρόβλεψης. Η συγκριτική ανάλυση με παραδοσιακές προσεγγίσεις μηχανικής μάθησης αποσκοπεί στην αποκάλυψη μιας βέλτιστης στρατηγικής για ακριβείς προβλέψεις στάθμευσης σε αστικές περιοχές. Η πρακτική εφαρμογή αξιοποιεί το Scikit-Learn και το TensorFlow, όπου το πρώτο διευκολύνει την αρχική ανάπτυξη μοντέλων και το δεύτερο προσφέρει επεκτασιμότητα για περίπλοκα μοντέλα πρόβλεψης.

Η αυτοματοποιημένη μηχανική μάθηση (AutoML), που χρησιμοποιεί εργαλεία όπως το AutoSklearn και το AutoKeras, βοηθά στην ανάπτυξη μοντέλων με την αυτοματοποίηση της βέλτιστης επιλογής μοντέλου και υπερπαραμέτρων. Η μεθοδολογία περιλαμβάνει τη συλλογή δεδομένων, την προεπεξεργασία και την ανάπτυξη μοντέλων.

Στο πλαίσιο των δοκιμών εντάσσεται και το Federated Learning (FL), μια καινοτόμα προσέγγιση στη μηχανική μάθηση. Στόχος είναι η αυξημένη προστασία, καθώς και η ενίσχυση της ακρίβειας των μοντέλων.

Από την προεπεξεργασία των δεδομένων έως τη δημιουργία μοντέλων και την ανάλυση των αποτελεσμάτων, η μεθοδολογία αποτελεί μια ισχυρή βάση για την εξαγωγή ουσιαστικών συμπερασμάτων και την καθοδήγηση μελλοντικών ερευνητικών προσπαθειών.

Λέξεις-κλειδιά: Πρόβλεψη χώρου στάθμευσης, Μηχανική Μάθηση, Βαθιά Μάθηση, Αυτοματοποιημένη μηχανική μάθηση (AutoML), Διαχείριση κυκλοφορίας

ABSTRACT

Prediction of urban parking spaces, vital for traffic management, is addressed in this thesis through advanced machine learning, deep learning and automated machine learning (AutoML) technologies, whereas the technology of federated learning is embedded too. Machine learning techniques decode the complexity of predicting parking availability by discriminating patterns within extensive datasets to enhance understanding and prediction in large cities.

Deep learning, a subset of machine learning, increases prediction capabilities. Comparative analysis with traditional machine learning approaches aims to uncover an optimal strategy for accurate parking forecasts in urban areas. The practical application leverages Scikit-Learn and TensorFlow, where the former facilitates initial model development and the latter offers scalability for complex prediction models.

Automated machine learning (AutoML), using tools such as AutoSklearn and Autokeras, helps model development by automating optimal model and hyperparameter selection. The methodology involves data collection, pre-processing and model development.

The trials include Federated Learning (FL), an innovative approach to machine learning. The aim is increased protection as well as enhanced model accuracy.

From data pre-processing to modelling and analysis of results, the methodology provides a strong basis for drawing meaningful conclusions and guiding future research efforts.

Keywords: Parking space prediction, Machine learning, Deep learning, Automated machine learning (AutoML), Traffic management

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

| | |
|--|-----|
| ΠΕΡΙΛΗΨΗ | ix |
| ABSTRACT | x |
| ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ | xi |
| ΕΙΣΑΓΩΓΗ..... | xiv |
| 1 Πρόβλεψη διαθεσιμότητας χώρων στάθμευσης | 17 |
| 1.1 Η πρόκληση της πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης | 17 |
| 1.2 Οφέλη ενός συστήματος πρόβλεψης διαθεσιμότητας..... | 20 |
| 1.3 Τεχνολογικές λύσεις | 22 |
| 1.4 Προηγούμενη έρευνα και συναφείς εργασίες..... | 24 |
| 2 Τεχνικές Μηχανικής Μάθησης | 28 |
| 2.1 Μέθοδοι Μηχανικής Μάθησης..... | 30 |
| 2.2 Προκλήσεις μηχανικής μάθησης | 32 |
| 2.3 Κοινοί αλγόριθμοι μηχανικής μάθησης | 34 |
| 2.4 Τεχνικές ταξινόμησης | 34 |
| 2.5 Τεχνικές Βαθιάς Μάθησης (Deep Learning)..... | 39 |
| 2.5.1 Βαθιά Μάθηση vs. Μηχανική Μάθηση | 40 |
| 2.5.2 Τρόπος λειτουργίας..... | 41 |
| 2.6 Εργαλεία μηχανικής μάθησης | 42 |
| 2.6.1 Scikit-Learn | 43 |
| 2.6.2 TensorFlow | 44 |
| 2.6.3 Σύγκριση | 46 |
| 3 AutoML..... | 47 |
| 3.1 Εργαλεία AutoML..... | 48 |

| | | |
|-------|--|----|
| 3.1.1 | AutoSklearn | 48 |
| 3.1.2 | Autokeras | 49 |
| 3.2 | Εφαρμογή AutoML για την ανάπτυξη μοντέλων | 50 |
| 4 | Federated Learning | 52 |
| 4.1 | Εφαρμογές federated learning σε μοντέλα πρόβλεψης..... | 54 |
| 4.2 | Προκλήσεις | 55 |
| 5 | Μεθοδολογία και πρακτικό πλαίσιο | 57 |
| 5.1 | Συλλογή και προεπεξεργασία δεδομένων | 57 |
| 5.1.1 | Προεπεξεργασία δεδομένων – 1 ^ο Στάδιο | 57 |
| 5.1.2 | Προεπεξεργασία δεδομένων – 2 ^ο Στάδιο | 59 |
| 5.1.3 | Προεπεξεργασία δεδομένων – 3 ^ο Στάδιο | 60 |
| 5.2 | Περιβάλλοντα λογισμικού μηχανικής μάθησης..... | 61 |
| 5.3 | Εκπαίδευση μοντέλων | 62 |
| 5.4 | Ενσωμάτωση AutoML πρακτικών..... | 63 |
| 5.5 | Διερεύνηση Federated Learning πρακτικών | 65 |
| 6 | Αποτελέσματα και συζήτηση | 66 |
| 6.1 | 1 ^ο Στάδιο Αποτελεσμάτων | 66 |
| 6.2 | 2 ^ο Στάδιο Αποτελεσμάτων | 69 |
| 6.3 | 3 ^ο Στάδιο Αποτελεσμάτων | 70 |
| 6.4 | 4 ^ο Στάδιο Αποτελεσμάτων | 71 |
| 6.5 | 5 ^ο Στάδιο Αποτελεσμάτων | 72 |
| 6.6 | 6 ^ο Στάδιο Αποτελεσμάτων | 73 |
| 6.7 | 7 ^ο Στάδιο Αποτελεσμάτων | 81 |
| 7 | Συμπεράσματα και Μελλοντική Έρευνα | 84 |
| 7.1 | Ανάλυση αποτελεσμάτων..... | 84 |
| 7.2 | Μελλοντικοί στόχοι | 89 |

| | |
|---|-----|
| ΠΑΡΑΡΤΗΜΑ..... | 92 |
| Keras MLP Train | 92 |
| Keras MLP Train (2 levels, Previous State)..... | 96 |
| Keras MLP Train (All days, Sundays, Holidays) | 101 |
| Keras MLP Train (Hours) | 102 |
| Autokeras | 103 |
| Autosklearn | 109 |
| Federated Learning..... | 115 |
| ΒΙΒΛΙΟΓΡΑΦΙΑ | 120 |

ΕΙΣΑΓΩΓΗ

Στο ταχέως εξελισσόμενο αστικό τοπίο, η πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης αποτελεί μια κρίσιμη πρόκληση με εκτεταμένες επιπτώσεις στη ροή της κυκλοφορίας και στις καθημερινές εμπειρίες των μετακινούμενων. Η παρούσα διπλωματική εξετάζει μεθόδους που μπορεί να βοηθήσουν κατά πολύ στην πρόβλεψη στάθμευσης αξιοποιώντας τις δυνατότητες προηγμένων τεχνολογιών, με ιδιαίτερη έμφαση στη μηχανική μάθηση, τη βαθιά μάθηση και την αυτοματοποιημένη μηχανική μάθηση (AutoML).

Η μηχανική μάθηση, ένας ισχυρός τομέας της τεχνητής νοημοσύνης, αποτελεί λύση για την αντιμετώπιση των περιπλοκών της πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης. Επιτρέποντας στα συστήματα να διακρίνουν μοτίβα και ιδέες από μεγάλα σύνολα δεδομένων, οι τεχνικές μηχανικής μάθησης προσφέρουν μια πολλά υποσχόμενη οδό για την κατανόηση και την πρόβλεψη της δυναμικής της στάθμευσης σε μια μεγάλη πόλη.

Η βαθιά μάθηση, ένα υποσύνολο της μηχανικής μάθησης, πηγαίνει τις δυνατότητες πρόβλεψης ένα βήμα παραπέρα. Η ικανότητά της να κατανοεί πολύπλοκες σχέσεις μέσα στα δεδομένα, ιδίως σε αδόμητα περιβάλλοντα, παρέχει ένα εξελιγμένο πλαίσιο για την αντιμετώπιση των πολύπλευρων προκλήσεων της αστικής στάθμευσης. Η παρούσα διπλωματική διερευνά τα συγκριτικά πλεονεκτήματα της βαθιάς μάθησης έναντι των παραδοσιακών προσεγγίσεων μηχανικής μάθησης, με στόχο την αποκάλυψη της βέλτιστης στρατηγικής για ακριβείς προβλέψεις στάθμευσης.

Στο πεδίο της πρακτικής υλοποίησης, δύο σταθερά εργαλεία, το Scikit-Learn και το TensorFlow, αναδεικνύονται κρίσιμα. Το Scikit-Learn, γνωστό για την ευελιξία και την ευκολία χρήσης του, διευκολύνει τα αρχικά στάδια ανάπτυξης μοντέλων και πειραματισμού. Το TensorFlow, μια βιβλιοθήκη μηχανικής μάθησης ανοικτού κώδικα, προσφέρει επεκτασιμότητα και ευελιξία, επιτρέποντας την κατασκευή περίπλοκων μοντέλων πρόβλεψης ικανών να περιηγηθούν στο περίπλοκο αστικό τοπίο.

Η αυτοματοποιημένη μηχανική μάθηση (AutoML) αποτελεί εργαλείο που αλλάζει τα δεδομένα στη διαδικασία ανάπτυξης μοντέλων. Εργαλεία όπως το AutoSklearn και το

Autokeras αυτοματοποιούν την επιλογή των βέλτιστων μοντέλων και υπερπαραμέτρων, απλοποιώντας την προγνωστική μοντελοποίηση. Μειώνοντας τη χειροκίνητη προσπάθεια που απαιτείται για την ανάπτυξη μοντέλων, το AutoML επιταχύνει το ρυθμό της καινοτομίας στην πρόβλεψη στάθμευσης.

Σημαντική προσθήκη αποτελεί η υλοποίηση Federated Learning (FL) πρακτικών. Το FL είναι μια ισχυρή προσέγγιση στον τομέα της μηχανικής μάθησης που επιτρέπει την εκπαίδευση μοντέλων σε αποκεντρωμένες συσκευές ή διακομιστές που κατέχουν τοπικά δείγματα δεδομένων, χωρίς να τα ανταλλάσσουν. Αυτό είναι ιδιαίτερα επωφελές σε σενάρια όπου το απόρρητο των δεδομένων και η ασφάλεια αποτελούν πρωταρχικές ανησυχίες, όπως στην ανάπτυξη ενός συστήματος πρόβλεψης διαθεσιμότητας χώρων στάθμευσης.

Η μεθοδολογία που υιοθετήθηκε στην παρούσα μελέτη ακολουθεί μια δομημένη ακολουθία βημάτων. Ξεκινώντας με τη σχολαστική συλλογή και προεπεξεργασία των δεδομένων, τίθενται τα θεμέλια για την επακόλουθη ανάπτυξη μοντέλων. Οι τεχνικές μηχανικής μάθησης, που περιλαμβάνουν τόσο παραδοσιακές όσο και βαθιές προσεγγίσεις μάθησης, αναπτύσσονται για τον εντοπισμό του πιο αποτελεσματικού μοντέλου για την πρόβλεψη της διαθεσιμότητας στάθμευσης στο πολύπλοκο αστικό περιβάλλον.

Καθώς αποκαλύπτονται τα αποτελέσματα των προγνωστικών μοντέλων, η παρούσα διπλωματική προσπαθεί να εξάγει ουσιαστικά συμπεράσματα. Η αλληλεπίδραση μεταξύ των επιλεγμένων τεχνικών, των συλλεχθέντων δεδομένων και των επιπτώσεων των προβλέψεων στον πραγματικό κόσμο αποτελούν τη βάση για διεισδυτικές παρατηρήσεις. Βασικότερος στόχος είναι μια ολοκληρωμένη κατανόηση του τρόπου με τον οποίο η μηχανική μάθηση, η βαθιά μάθηση και η AutoML μπορούν συλλογικά να αναδιαμορφώσουν το τοπίο της πρόβλεψης στάθμευσης σε μια μεγάλη πόλη.

Τα επόμενα κεφάλαια παρουσιάζουν αρχικά μια επισκόπηση στο πρόβλημα πρόβλεψης διαθεσιμότητας χώρων στάθμευσης, αναφέροντας και σχετικές εργασίες και έρευνες στο συγκεκριμένο τομέα, ενώ ακολουθεί και η παρουσίαση ενός συνόλου τεχνικών μηχανικής και βαθιάς μάθησης, επικεντρώνοντας στα εργαλεία Scikit-Learn,

Tensorflow και στις τεχνικές AutoML. Στη συνέχεια, παρουσιάζεται η μεθοδολογία που ακολουθήθηκε στη συγκεκριμένη περίπτωση, από τα διάφορα στάδια προεπεξεργασίας δεδομένων, έως τα μοντέλα που δημιουργήθηκαν και αποτελέσματα που προέκυψαν. Με βάση αυτά τα αποτελέσματα, προέκυψαν κάποια συμπεράσματα, τα οποία θα μπορούσαν να βοηθήσουν σε μελλοντική έρευνα.

1 Πρόβλεψη διαθεσιμότητας χώρων στάθμευσης

Μέσα στην έντονη ζωή των αστικών κέντρων, παραμένει μια επίμονη πρόκληση, η οποία επηρεάζει καθημερινά τόσο τους ανθρώπους, όσο και τους ρυθμούς και την αστική ζωή γενικότερα. Η διαθεσιμότητα των χώρων στάθμευσης, παίζει καθοριστικό ρόλο στη διαμόρφωση της κινητικότητας και της βιωσιμότητας των πόλεων. Καθώς οι πόλεις μεγαλώνουν και η αστικοποίηση επιταχύνεται, η πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο έχει αναδειχθεί σε κύριο μέλημα.

1.1 Η πρόκληση της πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης

Οι αστικές περιοχές σε όλο τον κόσμο αντιμετωπίζουν μια κοινή πρόκληση που συνεχίζει να κλιμακώνεται - τη διαθεσιμότητα χώρων στάθμευσης. Καθώς οι πόλεις μεγαλώνουν και η αστικοποίηση εντείνεται, η ανάγκη για αποτελεσματική διαχείριση των χώρων στάθμευσης γίνεται όλο και πιο σημαντική. Οι ακριβείς προβλέψεις της διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο έχουν τη δυνατότητα να ανακουφίσουν την κυκλοφοριακή συμφόρηση, να μειώσουν τις περιβαλλοντικές επιπτώσεις και να βελτιώσουν τη συνολική ποιότητα ζωής στα αστικά περιβάλλοντα.

Ιστορικά, η διαχείριση των χώρων στάθμευσης ήταν ένα πολύπλοκο ζήτημα, με πολλούς παράγοντες να παίζουν ρόλο. Η ικανότητα πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης είναι δύσκολη, κυρίως λόγω της δυναμικής φύσης της αστικής ζωής. Παράγοντες όπως η ώρα της ημέρας, η γεωγραφική θέση, τα γεγονότα και τα ιστορικά δεδομένα συμβάλλουν στη δυσκολία πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης.

Η πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης είναι ένα πολύπλευρο πρόβλημα με εκτεταμένες επιπτώσεις στην αστική ζωή. Σε μια πόλη που χαρακτηρίζεται από έντονη δραστηριότητα, κυκλοφοριακή συμφόρηση και περιορισμένη υποδομή στάθμευσης, η ικανότητα πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης είναι υψίστης σημασίας. Η πρόκληση έγκειται στην πρόβλεψη

σε πραγματικό χρόνο, λαμβάνοντας υπόψη τη δυναμική φύση των αστικών κέντρων. Το έργο αυτό απαιτεί μια ολιστική κατανόηση των διαφόρων παραγόντων που επηρεάζουν τη διαθεσιμότητα χώρων στάθμευσης, συμπεριλαμβανομένων, μεταξύ άλλων:

- της ώρας της ημέρας,
- της ημέρας της εβδομάδας,
- την τοποθεσία,
- των ειδικών εκδηλώσεων και
- των ιστορικών δεδομένων.

Οι ακριβείς προβλέψεις είναι ζωτικής σημασίας για την καθοδήγηση των οδηγών σε διαθέσιμες θέσεις στάθμευσης, την ελαχιστοποίηση της περιττής κυκλοφοριακής συμφόρησης και την προώθηση βιώσιμων αστικών περιβαλλόντων.

Το πρόβλημα γίνεται ακόμα πιο σοβαρό όταν δημιουργεί αρνητικές συνέπειες και σε άλλους τομείς. Η αυξημένη κατανάλωση ενέργειας, η υπερθέρμανση του πλανήτη ο πολλαπλασιασμός των αερομεταφερόμενων ασθενειών είναι μερικά παραδείγματα. Σύμφωνα με το Παγκόσμιο Ινστιτούτο Πόρων, το 74% των εκπομπών διοξειδίου του άνθρακα που συμβάλλουν στην υπερθέρμανση του πλανήτη προέρχεται από τα αέρια του θερμοκηπίου, ενώ το 93% αυτών των εκπομπών αποδίδεται στη χρήση ορυκτών καυσίμων στις μεταφορές, την κατασκευή και την κατανάλωση. Με φόντο αυτά τα ανησυχητικά στατιστικά στοιχεία, το έτος 2020, όπως αποκαλύφθηκε από την ανάλυση της NASA, κέρδισε την αμφίβολη διάκριση του θερμότερου έτους που έχει καταγραφεί ποτέ (Inam, 2022).

Η πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης αναδεικνύεται ως ένα βασικό ζήτημα που επηρεάζει βαθιά τόσο τις ατομικές αποφάσεις για ταξίδια με βάση το αυτοκίνητο, όσο και τη συνολική δυναμική της κυκλοφορίας στις αστικές περιοχές. Οι επιλογές των οδηγών είναι εγγενώς χρονικά εξαρτώμενες, επηρεαζόμενες όχι μόνο από τις αντιλήψεις σε πραγματικό χρόνο στο δρόμο, αλλά

και από προηγούμενες εμπειρίες. Στην περίπτωση της στάθμευσης, η προηγούμενη γνώση σχετικά με τις επικρατούσες συνθήκες -όπως η πιθανότητα εύρεσης θέσης στάθμευσης, το κόστος στάθμευσης εκτός δρόμου και άλλοι σχετικοί παράγοντες- ασκεί σημαντική επιρροή στις αποφάσεις των οδηγών για στάθμευση. Ταυτόχρονα, η αναζήτηση ελεύθερων θέσεων στάθμευσης μπορεί να προκαλέσει καταστροφή στις συνθήκες κυκλοφορίας και στο περιβάλλον, καθώς τα οχήματα που κάνουν κύκλους στους δρόμους σε αναζήτηση διαθέσιμων θέσεων επιτείνουν το χάος και συμβάλλουν στην εκπομπή επιβλαβών ρύπων (Feng,2022).

Αυτή η αδιάκοπη αναζήτηση ελεύθερων χώρων στάθμευσης αποτελεί σημαντικό παράγοντα που συμβάλλει στα προβλήματα της οδικής κυκλοφορίας στις αστικές περιοχές, αντιπροσωπεύοντας έως και το 40% της συνολικής κυκλοφοριακής ροής. Η βασική αιτία αυτού του προβλήματος έγκειται στο γεγονός ότι οι οδηγοί συχνά αποτυγχάνουν να λάβουν τις πιο αποτελεσματικές αποφάσεις στάθμευσης. Οι επιλογές τους, τις περισσότερες φορές, καθοδηγούνται από μεροληπτικές αντιλήψεις στο δρόμο και προηγούμενες προσωπικές εμπειρίες, αντί να καθοδηγούνται από πληροφορίες και προβλέψεις που βασίζονται σε δεδομένα (Camero, 2019).

Έρευνα που διεξήχθη υπογράμμισε το ζήτημα της αναζήτησης θέσεων στάθμευσης, αποκαλύπτοντας ότι αντιπροσωπεύει ένα σημαντικό μέρος της αστικής κυκλοφορίας, που κυμαίνεται από 8% έως 74%. Αυτό μεταφράζεται σε μέσο χρόνο αναζήτησης 3,5 έως 14 λεπτών ανά θέση στάθμευσης, ασκώντας βαθιά και επιζήμια επίδραση στο κυκλοφοριακό δίκτυο. Σε μια ξεχωριστή μελέτη, οι συνέπειες αυτής της ευρέως διαδεδομένης πρακτικής γίνονται ολοφάνερες. Η πράξη της αναζήτησης θέσεων στάθμευσης σπαταλά κολοσσιαία 8,37 εκατομμύρια γαλόνια βενζίνης ετησίως και συμβάλλει σε πάνω από 129.000 τόνους εκπομπών διοξειδίου του άνθρακα (CO₂) στο ήδη μολυσμένο αστικό περιβάλλον. Οι επιπτώσεις αυτών των ευρημάτων είναι βαθιές, τονίζοντας ότι η ταχεία και αποτελεσματική χωροθέτηση των χώρων στάθμευσης έχει τεράστια σημασία. Δεν ανακουφίζει μόνο από τα βάρη της κυκλοφοριακής συμφόρησης,

αλλά εξοικονομεί επίσης ενεργειακούς πόρους και συμβάλλει αποφασιστικά στη μείωση των επιβλαβών εκπομπών (Fan, 2020).

Ως εκ τούτου, η επείγουσα ανάγκη για καινοτόμες λύσεις σε αυτό το πρόβλημα είναι προφανής, με την πρόβλεψη της διαθεσιμότητας στάθμευσης να είναι έτοιμη να διαδραματίσει καθοριστικό ρόλο στη αλλαγή του αστικού τοπίου και στην εξομάλυνση των προκλήσεων που θέτουν η αστικοποίηση, η κυκλοφοριακή συμφόρηση και η ρύπανση.

1.2 Οφέλη ενός συστήματος πρόβλεψης διαθεσιμότητας

Στην επιδίωξη της βελτιστοποίησης των αστικών δικτύων μεταφορών και της ανακούφισης των προκλήσεων που σχετίζονται με τη στάθμευση σε πυκνοκατοικημένες περιοχές, η εφαρμογή συστημάτων διαθεσιμότητας στάθμευσης σε πραγματικό χρόνο αναδεικνύεται κρίσιμη. Τέτοια συστήματα έχουν πολύπλευρα οφέλη, τα οποία εκτείνονται πολύ πέρα από την απλή ευκολία εύρεσης μιας θέσης στάθμευσης. Με την απρόσκοπτη ενσωμάτωση δεδομένων από έξυπνα συστήματα πλοήγησης, κάμερες παρακολούθησης και προγνωστικές αναλύσεις, τα προηγμένα αυτά συστήματα όχι μόνο ενισχύουν τη ροή της κυκλοφορίας και μειώνουν την κατανάλωση καυσίμων, αλλά συμβάλλουν και στην προστασία του περιβάλλοντος. Ο μετριασμός των παραβάσεων στάθμευσης και η προώθηση της οδικής ασφάλειας υπογραμμίζουν περαιτέρω τον κοινωνικό αντίκτυπο αυτών των καινοτόμων λύσεων. Πιο αναλυτικά, τα σημαντικότερα οφέλη τέτοιων συστημάτων είναι:

- **Ενισχυμένη ροή κυκλοφορίας:** Ένα υπερσύγχρονο σύστημα διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο παίζει καθοριστικό ρόλο στη βελτιστοποίηση της ροής της κυκλοφορίας, παρέχοντας στους οδηγούς άμεση πληροφόρηση για τις διαθέσιμες θέσεις στάθμευσης. Συγκλωνεύοντας δεδομένα από έξυπνα συστήματα πλοήγησης, κάμερες παρακολούθησης, προγνωστικές αναλύσεις και άλλους μηχανισμούς μεταφορών, το σύστημα αυτό δίνει τη δυνατότητα στους οδηγούς να λαμβάνουν καλά ενημερωμένες αποφάσεις σχετικά με τις θέσεις στάθμευσης. Κατά συνέπεια, αυτή η προσέγγιση έχει ως

αποτέλεσμα τη μείωση των οχημάτων στο δρόμο, μετριάζοντας την πιθανότητα κυκλοφοριακής συμφόρησης. Ο θετικός αντίκτυπος στη ροή της κυκλοφορίας όχι μόνο επιταχύνει τις μετακινήσεις, αλλά συμβάλλει και στη συνολική αποτελεσματικότητα της αστικής κινητικότητας.

- **Αποδοτικότητα καυσίμων και διατήρηση του περιβάλλοντος:** Πέρα από τις ικανότητές του για την ανακούφιση της κυκλοφορίας, ένα σύστημα διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο αναδεικνύεται σε μια περιβαλλοντικά συνειδητή λύση που βοηθά στον περιορισμό της κατανάλωσης καυσίμων. Όταν οι οδηγοί εντοπίζουν αβίαστα τις κατάλληλες θέσεις στάθμευσης χωρίς να καταφεύγουν σε περιττά χιλιόμετρα άσκοπης οδήγησης, η επακόλουθη μείωση της κατανάλωσης καυσίμων επιφέρει οικολογικά οφέλη. Αυτή η φιλική προς το περιβάλλον λύση στάθμευσης δεν εξοικονομεί μόνο καύσιμα αλλά μεταφράζεται επίσης σε εξοικονόμηση κόστους για τους οδηγούς, προωθώντας μια πιο οικονομικά βιώσιμη εμπειρία οδήγησης. Επιπλέον, η μειωμένη κατανάλωση καυσίμων συμβάλλει σε αξιοσημείωτη μείωση των εκπομπών αερίων και επιβλαβών ρύπων, ευθυγραμμιζόμενη με τους ευρύτερους στόχους διατήρησης του περιβάλλοντος.
- **Μετριασμός των παραβάσεων στάθμευσης:** Σε πυκνοκατοικημένες αστικές περιοχές, οι παραβάσεις στάθμευσης αποτελούν σημαντική πρόκληση για τις υπηρεσίες επιβολής του νόμου. Η εφαρμογή ενός συστήματος διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο μειώνει σημαντικά την πιθανότητα παράνομης στάθμευσης παρέχοντας στους οδηγούς ακριβείς πληροφορίες σχετικά με τις διαθέσιμες θέσεις στάθμευσης. Ως αποτέλεσμα, οι οδηγοί είναι λιγότερο πρόθυμοι να παρκάρουν σε μη εξουσιοδοτημένες ή απαγορευμένες περιοχές. Η επακόλουθη μείωση των παραβάσεων στάθμευσης όχι μόνο ελαφρύνει τον φόρτο για τους φύλακες στάθμευσης και τις υπηρεσίες επιβολής του νόμου, αλλά προάγει επίσης την αυξημένη τήρηση των κανόνων και των κανονισμών κυκλοφορίας, συμβάλλοντας σε ένα πιο εύρυθμο αστικό περιβάλλον.

- **Αυξημένα πρότυπα ασφαλείας:** Αντιμετωπίζοντας τις ανησυχίες για την οδική ασφάλεια, το σύστημα διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο εξαλείφει την ανάγκη οι οδηγοί να περιηγούνται άσκοπα ή να κάνουν κύκλους σε οικοδομικά τετράγωνα προς αναζήτηση θέσεων στάθμευσης. Διαθέτοντας καθορισμένες θέσεις στάθμευσης απευθείας στους οδηγούς, το σύστημα ελαχιστοποιεί τον κίνδυνο ατυχημάτων, ενισχύοντας την ασφάλεια για τους πεζούς και τους ποδηλάτες σε πολυσύχναστα αστικά περιβάλλοντα. Αυτή η διαδικασία στάθμευσης όχι μόνο μειώνει τις αβεβαιότητες της κίνησης των οχημάτων, αλλά προάγει επίσης ένα ασφαλέστερο αστικό τοπίο.
- **Ολιστική αστική βελτίωση:** Στο πεδίο των λύσεων διαχείρισης έξυπνων χώρων στάθμευσης στις πόλεις, τα συστήματα διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο αναδεικνύονται ως μετασχηματιστικά εργαλεία που διαθέτουν τεράστιες δυνατότητες για την ενίσχυση των αστικών δικτύων μεταφορών. Αυτές οι έξυπνες λύσεις δίνουν τη δυνατότητα σε κυβερνητικές υπηρεσίες και διαχειριστές χώρων στάθμευσης να βελτιώσουν τις λειτουργίες τους, επιτρέποντάς τους να επικεντρωθούν στη βελτίωση της συνολικής εμπειρίας στάθμευσης για τους οδηγούς. Ως αποτέλεσμα, οι αστικές περιοχές γίνονται πιο αποτελεσματικές, βιώσιμες και αειφόρες. Επιπλέον, καθώς η τεχνολογία εξελίσσεται, τα συστήματα αυτά είναι έτοιμα να γίνουν όλο και πιο εξελιγμένα, υποσχόμενα συνεχείς βελτιώσεις στα οφέλη που προσφέρουν στο δίκτυο αστικών μεταφορών (Joshi, 2023).

1.3 Τεχνολογικές λύσεις

Στο διαρκώς εξελισσόμενο τοπίο του τομέα της στάθμευσης, οι συνεχείς τεχνολογικές καινοτομίες έχουν οδηγήσει σε αξιοσημείωτη πρόοδο, εισάγοντας την αποτελεσματικότητα, την ασφάλεια και την ευκολία. Τα συστήματα διαχείρισης χώρων στάθμευσης έχουν υποστεί αλλαγή, αντιμετωπίζοντας τις προκλήσεις της αστικής κινητικότητας με εφευρετικότητα.

Μια πρωτοποριακή καινοτομία που μεταμορφώνει τη βιομηχανία στάθμευσης είναι η ευρεία υιοθέτηση των αισθητήρων πληρότητας και των συστημάτων καθοδήγησης. Αυτές οι εξελιγμένες συσκευές αξιοποιούν την τεχνολογία ανίχνευσης αιχμής για την παρακολούθηση της διαθεσιμότητας χώρων στάθμευσης σε πραγματικό χρόνο. Αυτό όχι μόνο επιταχύνει τη διαδικασία εύρεσης θέσεων στάθμευσης για τους οδηγούς, αλλά μειώνει επίσης σημαντικά τον χρόνο που δαπανάται για να κάνει κύκλους γύρω από τα γκαράζ στάθμευσης, ανακουφίζοντας στη συνέχεια την κυκλοφοριακή συμφόρηση. Πέρα από την απλή καθοδήγηση, υπάρχουν αισθητήρες που διευκολύνουν τη βιντεοεπιτήρηση, την ανίχνευση σταθμευμένων οχημάτων και την ακριβή αναγνώριση πινακίδων κυκλοφορίας, προσφέροντας μια ολοκληρωμένη λύση που υπερβαίνει την παραδοσιακή διαχείριση στάθμευσης.

Μια άλλη κομβική τάση στον κλάδο της στάθμευσης αφορά την ενσωμάτωση συστημάτων αναγνώρισης πινακίδων κυκλοφορίας στα σημεία πρόσβασης. Τα συστήματα αυτόματης αναγνώρισης πινακίδων κυκλοφορίας διαδραματίζουν καθοριστικό ρόλο στον εντοπισμό εξουσιοδοτημένων ή μη εξουσιοδοτημένων οχημάτων εντός των εγκαταστάσεων στάθμευσης. Τα συστήματα αυτά προσφέρουν στους φορείς λειτουργίας χώρων στάθμευσης μεγαλύτερο έλεγχο, αυξημένη ασφάλεια και βελτιωμένη αυτοματοποίηση στην υποδομή τους.

Η αυξανόμενη τάση της χρήσης εφαρμογών κινητών τηλεφώνων για την πληρωμή και την κράτηση θέσεων στάθμευσης σηματοδοτεί μια στροφή προς την ενίσχυση της εμπειρίας των χρηστών. Οι κινητές εφαρμογές απλοποιούν τη διαδικασία, επιτρέποντας στους οδηγούς να κάνουν κράτηση και να πληρώνουν άνετα για θέσεις στάθμευσης χρησιμοποιώντας τις κινητές συσκευές τους, προσθέτοντας ένα επιπλέον επίπεδο ευκολίας στη συνολική εμπειρία στάθμευσης.

Η έλευση των συστημάτων χωρίς εισιτήριο προαναγγέλλει την απομάκρυνση από τα παραδοσιακά μοντέλα εισιτηρίων σε χαρτί, επιφέροντας μια σειρά από οφέλη. Πέρα από την αύξηση της αποδοτικότητας με την εξάλειψη της

ανάγκης εκτύπωσης και συλλογής εισιτηρίων, τα συστήματα αυτά βελτιώνουν σημαντικά την εμπειρία του χρήστη. Οι οδηγοί μπορούν να εισέρχονται και να εξέρχονται απρόσκοπτα από τις εγκαταστάσεις στάθμευσης χωρίς την ταλαιπωρία της διαχείρισης φυσικών εισιτηρίων, επιταχύνοντας έτσι τη διαδικασία και μειώνοντας το άγχος. Για τους φορείς, τα συστήματα ανίχνευσης πινακίδων συμβάλλουν στην επαγγελματική διαχείριση των εγκαταστάσεων, αυτοματοποιώντας την καταγραφή της εισόδου και εξόδου των οχημάτων για λεπτομερή έλεγχο πρόσβασης και αποτελεσματική τιμολόγηση.

Μια αναδυόμενη τάση στην αγορά στάθμευσης είναι η εφαρμογή στρατηγικών δυναμικής τιμολόγησης. Αυτό το καινοτόμο σύστημα προσαρμόζει τις τιμές με βάση τη ζήτηση για στάθμευση ανά πάσα στιγμή, αξιοποιώντας αισθητήρες πληρότητας και συστήματα ανάλυσης δεδομένων. Η δυναμική τιμολόγηση όχι μόνο μεγιστοποιεί το εισόδημα για τους φορείς, αλλά διασφαλίζει επίσης την αποτελεσματική αξιοποίηση των χώρων στάθμευσης. Η προσαρμογή των τιμών σε πραγματικό χρόνο επιτρέπει στους φορείς να ανταποκρίνονται γρήγορα στις αλλαγές της ζήτησης, βελτιστοποιώντας ανάλογα τις στρατηγικές τους.

Οι πλατφόρμες διαχείρισης χώρων στάθμευσης αντιπροσωπεύουν το αποκορύφωμα της αποτελεσματικότητας και του ελέγχου για τους φορείς λειτουργίας χώρων στάθμευσης. Αυτές οι πλατφόρμες αξιοποιούν την τεχνολογία έξυπνων χώρων στάθμευσης για τη συγκέντρωση και ανάλυση δεδομένων σε πραγματικό χρόνο σχετικά με τη διαθεσιμότητα χώρων στάθμευσης, τη ροή της κυκλοφορίας και άλλους συναφείς παράγοντες. Προσφέροντας μια ολοκληρωμένη εικόνα των λειτουργιών των εγκαταστάσεων στάθμευσης, οι πλατφόρμες αυτές δίνουν τη δυνατότητα στους φορείς να λαμβάνουν τεκμηριωμένες αποφάσεις, να βελτιώνουν την αποδοτικότητα και να αναβαθμίζουν τη συνολική διαχείριση των εγκαταστάσεών τους (Quercus, n.d.).

1.4 Προηγούμενη έρευνα και συναφείς εργασίες

Πολυάριθμες μελέτες και ερευνητικές προσπάθειες έχουν αφιερωθεί στην πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης σε αστικά περιβάλλοντα. Οι

μελέτες αυτές περιλαμβάνουν ένα ευρύ φάσμα προσεγγίσεων, από μοντέλα που βασίζονται σε κανόνες έως προηγμένες τεχνικές μηχανικής μάθησης.

Προηγούμενες έρευνες έχουν δείξει ότι τα συστήματα που βασίζονται σε κανόνες, τα οποία βασίζονται σε προκαθορισμένες ευρετικές μεθόδους, μπορούν να παρέχουν πολύτιμες πληροφορίες σχετικά με τη διαθεσιμότητα χώρων στάθμευσης. Ωστόσο, τα συστήματα αυτά μπορεί να μην έχουν την προσαρμοστικότητα και την ακρίβεια πρόβλεψης που απαιτούνται για δυναμικά αστικά περιβάλλοντα. Ως αποτέλεσμα, οι ερευνητές έχουν στραφεί σε μεθόδους μηχανικής και βαθιάς μάθησης για να ξεπεράσουν αυτούς τους περιορισμούς.

Μια σχετική μελέτη (Feng, 2022) εισάγει ένα νέο πλαίσιο βαθιάς μάθησης, το dConvLSTM-DCN (dual Convolutional Long Short-Term Memory with Dense Convolutional Network), για να κάνει βραχυπρόθεσμες και μακροπρόθεσμες προβλέψεις σχετικά με τη διαθεσιμότητα VPS. Αξιοποιώντας τις χρονικές και χωρικές συσχετίσεις, αυτό το μοντέλο επιτυγχάνει υψηλή ακρίβεια στην πρόβλεψη της διαθεσιμότητας θέσεων στάθμευσης, όπως αποδεικνύεται μέσω εκτεταμένης αξιολόγησης με πρακτικά δεδομένα από δημόσιους χώρους στάθμευσης.

Στο άρθρο των Bassetti, Berti κ.ά. (Bassetti, 2022) οι συγγραφείς ανέπτυξαν και εκπαίδευσαν τέσσερα μοντέλα μηχανικής μάθησης για την πρόβλεψη των επιπέδων διαθεσιμότητας στάθμευσης για τμήματα αστικών δρόμων, βοηθώντας την πλοήγηση και τις εφαρμογές έξυπνης στάθμευσης. Χρησιμοποίησαν έναν προσαρμοσμένο προσομοιωτή *cruising-for-parking* για να δημιουργήσουν δεδομένα που αναπαράγουν τις συμπεριφορές στάθμευσης εργαζομένων, κατοίκων, αγοραστών και επισκεπτών. Αυτά τα δεδομένα προσομοίωσης, που συλλέχθηκαν από 40 χρήστες επί 200 εβδομάδες στο San Giovanni της Ρώμης, επέτρεψαν την εξαγωγή δέκα χαρακτηριστικών για κάθε οδικό τμήμα σε διαφορετικές χρονικές στιγμές, με αποτέλεσμα ένα σύνολο δεδομένων με 761 δείγματα. Τα μοντέλα μηχανικής μάθησης εκπαιδεύτηκαν χρησιμοποιώντας αυτά τα δεδομένα για να ταξινομήσουν τη διαθεσιμότητα στάθμευσης ως πράσινη, κίτρινη ή κόκκινη. Τα μοντέλα αξιολογήθηκαν αυστηρά σε μια διαφορετική

περιοχή της πόλης, το San Lorenzo, και απέδωσαν υψηλά ποσοστά ακρίβειας που ξεπέρασαν το 97% για οδικά τμήματα με επαρκείς ενέργειες των χρηστών. Αυτό αποδεικνύει την ικανότητα του προσομοιωτή να παράγει ρεαλιστικά δεδομένα στάθμευσης και τις δυνατότητες για έναν ταξινομητή διαθεσιμότητας στάθμευσης (PAC) που βασίζεται σε δεδομένα που συλλέγονται με smartphone.

Μια άλλη μελέτη (Camero, 2019) εισάγει μια καινοτόμο προσέγγιση για την πρόβλεψη των ποσοστών πληρότητας των χώρων στάθμευσης αυτοκινήτων με τη χρήση βαθιάς μάθησης και επαναλαμβανόμενων νευρωνικών δικτύων. Πρόκειται για μια σημαντική πρόκληση στο πεδίο της έξυπνης κινητικότητας και η μέθοδος περιλαμβάνει τον αυτόματο σχεδιασμό ενός βαθύ δικτύου που περικλείει τη συμπεριφορά της πληρότητας των αυτοκινήτων. Αυτό το δίκτυο, με τη σειρά του, προσφέρει τεκμηριωμένες εκτιμήσεις για τις διαθέσιμες θέσεις στάθμευσης σε μεσοπρόθεσμο ορίζοντα. Η μελέτη περίπτωσης αφορά 29 χώρους στάθμευσης αυτοκινήτων στο Μπέρμιγχαμ του Ηνωμένου Βασιλείου και διήρκεσε έντεκα εβδομάδες. Τα αποτελέσματά συγκρίθηκαν με τους πλέον σύγχρονους προβλεπτικούς μηχανισμούς και αποκαλύπτουν ότι η προσέγγιση επιδεικνύει αξιοσημείωτη ακρίβεια.

Άλλο ένα μοντέλο βαθιάς μάθησης (Gao, 2023), το DWT-ConvGRU-BRC, ενσωματώνει διάφορα βασικά στοιχεία για αυξημένη ακρίβεια. Ξεκινά με τη χρήση ενός διακριτού μετασχηματισμού κυματιδίων (DWT) για την αποτελεσματική αποθορυβοποίηση των ιστορικών δεδομένων στάθμευσης. Στη συνέχεια, αποτυπώνει τις χρονικές και χωρικές συσχετίσεις εντός και μεταξύ των χώρων στάθμευσης μέσω ενός συνελκτικού δικτύου επαναλαμβανόμενων μονάδων (ConvGRU). Επιπλέον, το μοντέλο ενσωματώνει διάφορες εισροές, όπως η διαθεσιμότητα, η θερμοκρασία, η υγρασία, η ταχύτητα του ανέμου, οι καθημερινές και τα Σαββατοκύριακα, για την ενίσχυση της ακρίβειας πρόβλεψης.

Μοντέλα μηχανικής μάθησης, όπως τα δέντρα αποφάσεων (Decision Trees), τα Random Forest και οι K-Nearest Neighbors, έχουν χρησιμοποιηθεί για να συλλάβουν σύνθετα μοτίβα σε δεδομένα στάθμευσης. Στη μελέτη των Inam κ.α. (Inam, 2022) αξιοποιούνται η πληρότητα των χώρων στάθμευσης, τα

δεδομένα πεζών, οι καιρικές συνθήκες και τα δεδομένα κυκλοφορίας για να διερευνηθεί η επίδραση των εξωτερικών μεταβλητών στις προβλέψεις στάθμευσης στον δρόμο. Η έρευνα περιλάμβανε μια συγκριτική ανάλυση τεχνικών μηχανικής μάθησης και βαθιάς μάθησης, συμπεριλαμβανομένων των τεχνικών Random Forest (RF), Decision Trees (DT), K-Nearest Neighbors (KNN) και άλλων για προβλέψεις διαθεσιμότητας χώρων στάθμευσης στον δρόμο. Η RF αναδείχθηκε ως η καλύτερη απόδοση, με εντυπωσιακή μέση ακρίβεια 81% και AUC (Area Under Curve) 0,18. Η μελέτη αυτή αναδεικνύει την αποτελεσματικότητα των λιγότερο πολύπλοκων αλγορίθμων, όπως οι RF, DT και KNN, σε σχέση με πιο περίπλοκα μοντέλα, όπως το Multilayer Perceptron (MLP).

Άλλες έρευνες και μελέτες ασχολούνται με την πρόβλεψη ελεύθερων θέσεων πάρκινγκ χρησιμοποιώντας federated learning. Στο πλαίσιο της έρευνας από τους Huang κ.α (Huang, 2021), οι διαχειριστές χώρων στάθμευσης (Parking Lot Operators - PLO) εκπαιδεύουν συνεργατικά ένα μοντέλο μακράς βραχυπρόθεσμης μνήμης για την εκτίμηση των χώρων στάθμευσης χωρίς την ανάγκη ανταλλαγής ευαίσθητων ακατέργαστων δεδομένων. Αυτή η προσέγγιση διατήρησης της ιδιωτικής ζωής επεκτείνεται περαιτέρω στο Parked Vehicle-assisted Edge Computing (PVEC), όπου τα οχήματα λειτουργούν ως κόμβοι υπολογισμού ακμής. Το σύστημα εκφορτώνει αποτελεσματικά τις υπηρεσίες, λαμβάνοντας υπόψη τις απαιτήσεις υπολογισμού και τους περιορισμούς χωρητικότητας στάθμευσης, όλα μέσα σε ένα πλαίσιο βαθιάς ενισχυτικής μάθησης πολλαπλών πρακτόρων. Το FedParking επιδεικνύει μια αποτελεσματική και συνειδητή ως προς την προστασία της ιδιωτικής ζωής μέθοδο για την ενίσχυση της διαχείρισης στάθμευσης και της εκφόρτωσης υπολογισμών.

Ενώ οι προσπάθειες αυτές έχουν προωθήσει τον τομέα, υπάρχουν ευκαιρίες για περαιτέρω βελτίωση και καινοτομία. Η παρακάτω μελέτη εισάγει μια έμφαση στην AutoML (AutoSklearn και Autokeras), το federated learning και την παράλληλη επεξεργασία ως νέες προσεγγίσεις για την ενίσχυση της ακρίβειας και της επεκτασιμότητας της πρόβλεψης. Με βάση προηγούμενες έρευνες και εισάγοντας καινοτόμες μεθοδολογίες, στόχος είναι η συμβολή στη συνεχιζόμενη

εξέλιξη της πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης σε αστικά περιβάλλοντα.

2 Τεχνικές Μηχανικής Μάθησης

Η μηχανική μάθηση, ένας βασικός κλάδος της τεχνητής νοημοσύνης (TN) και της επιστήμης των υπολογιστών, λειτουργεί στη διασταύρωση δεδομένων και αλγορίθμων, φιλοδοξώντας να αναπαράγει τον τρόπο με τον οποίο μαθαίνει ο άνθρωπος και να βελτιώνει σταδιακά την ακρίβειά του. Τις τελευταίες δεκαετίες, τα σημαντικά βήματα στην τεχνολογική πρόοδο, ιδίως όσον αφορά την αποθήκευση και την επεξεργαστική ισχύ, άνοιξαν το δρόμο για πρωτοποριακά προϊόντα και καινοτομίες στη μηχανική μάθηση. Εξέχοντα παραδείγματα αποτελούν η ιδιαίτερα αναγνωρισμένη μηχανή συστάσεων του Netflix, η οποία φαίνεται να κατανοεί τις προτιμήσεις των χρηστών της για την προβολή των ταινιών, και η ανατρεπτική εμφάνιση των αυτοκινούμενων αυτοκινήτων, που φέρνει επανάσταση στην αυτοκινητοβιομηχανία.

Η μηχανική μάθηση παίζει απαραίτητο ρόλο στο εκτεταμένο πεδίο της επιστήμης των δεδομένων. Χρησιμοποιώντας μια σειρά από στατιστικές τεχνικές, οι αλγόριθμοι μηχανικής μάθησης εκπαιδεύονται σχολαστικά για την εκτέλεση εργασιών, όπως η πραγματοποίηση ταξινομήσεων ή προβλέψεων και η εξαγωγή κρίσιμων πληροφοριών σε έργα εξόρυξης δεδομένων. Αυτές οι γνώσεις, μόλις ανακαλυφθούν, γίνονται η κινητήριος δύναμη πίσω από τις διαδικασίες λήψης αποφάσεων σε εφαρμογές και επιχειρήσεις, επηρεάζοντας ιδανικά βασικές μετρήσεις ανάπτυξης και ενισχύοντας τη λειτουργική αποτελεσματικότητα. Καθώς τα μεγάλα δεδομένα αυξάνονται και εξελίσσονται, η αγορά θα ζητά όλο και περισσότερους επιστήμονες δεδομένων. Αυτοί οι εξειδικευμένοι επαγγελματίες είναι επιφορτισμένοι όχι μόνο με την πλοήγηση σε τεράστια σύνολα δεδομένων, αλλά και με τον εντοπισμό των πιο σημαντικών επιχειρηματικών ερωτημάτων και των δεδομένων που απαιτούνται για την αποτελεσματική αντιμετώπισή τους.

Χαρακτηριστικό γνώρισμα της μηχανικής μάθησης είναι η εξάρτησή της από πλαίσια που επιταχύνουν την ανάπτυξη λύσεων. Μεταξύ των πιο γνωστών

είναι το TensorFlow και το PyTorch, ισχυρές πλατφόρμες που δίνουν τη δυνατότητα σε προγραμματιστές και επιστήμονες δεδομένων να δημιουργούν, να βελτιώνουν και να αναπτύσσουν αποτελεσματικά μοντέλα μηχανικής μάθησης. Αυτά τα πλαίσια έχουν συμβάλει καταλυτικά στην καινοτομία σε ένα φάσμα κλάδων, από την υγειονομική περίθαλψη και τη χρηματοδότηση έως το λιανικό εμπόριο και όχι μόνο, καταδεικνύοντας έτσι τις μετασχηματιστικές δυνατότητες της μηχανικής μάθησης στη διαμόρφωση του μέλλοντος της τεχνολογίας και της κοινωνίας. Κάθε μέρα που περνάει, η μηχανική μάθηση συνεχίζει να διευρύνει τα όρια του εφικτού, προσφέροντας μια ματιά σε ένα μέλλον όπου οι ευφυείς αλγόριθμοι θα ενισχύουν απρόσκοπτα την ανθρώπινη λήψη αποφάσεων, φέρνοντας επανάσταση στον τρόπο με τον οποίο αλληλεπιδρούμε με την τεχνολογία και τον κόσμο γύρω μας.

Το UC Berkeley παρέχει μια συνοπτική ανάλυση των βασικών ενεργειών που αποτελούν το σύστημα μάθησης ενός αλγορίθμου μηχανικής μάθησης, τονίζοντας τον υποκείμενο μηχανισμό του. Η ανάλυση αυτή περιλαμβάνει τρεις πρωταρχικές πτυχές:

1. **Διαδικασία λήψης αποφάσεων:** Οι αλγόριθμοι μηχανικής μάθησης, στην ουσία, χρησιμεύουν ως εργαλεία πρόβλεψης ή ταξινόμησης. Αξιοποιούν δεδομένα εισόδου, τα οποία μπορεί να είναι επισημασμένα ή μη επισημασμένα, για να παράγουν εκτιμήσεις σχετικά με τα πρότυπα εντός του συνόλου δεδομένων. Αυτή η κομβική διαδικασία λήψης αποφάσεων αποτελεί το θεμέλιο πάνω στο οποίο λειτουργεί η ευφυΐα του αλγορίθμου.
2. **Συνάρτηση σφάλματος:** Η αποτελεσματικότητα των προβλέψεων του μοντέλου μηχανικής μάθησης εξετάζεται μέσω μιας συνάρτησης σφάλματος. Αυτή η συνάρτηση λειτουργεί όταν είναι διαθέσιμα γνωστά παραδείγματα, διευκολύνοντας τη σύγκριση για τη μέτρηση της ακρίβειας των προβλέψεων του μοντέλου. Χρησιμεύει ως ένας κρίσιμος μηχανισμός ανατροφοδότησης για την αξιολόγηση της ποιότητας των αποτελεσμάτων του αλγορίθμου.
3. **Διαδικασία βελτιστοποίησης μοντέλου:** Η πορεία προς τη βελτίωση της απόδοσης του μοντέλου εξαρτάται από τη διαδικασία βελτιστοποίησης

του μοντέλου. Εάν υπάρχουν περιθώρια βελτίωσης - συγκεκριμένα, εάν το μοντέλο μπορεί να ευθυγραμμιστεί καλύτερα με τα σημεία δεδομένων εντός του συνόλου εκπαίδευσης - γίνονται προσαρμογές των βαρών. Οι προσαρμογές αυτές αποσκοπούν στην ελαχιστοποίηση των αποκλίσεων μεταξύ των εκτιμήσεων του μοντέλου και των γνωστών παραδειγμάτων. Ο αλγόριθμος προχωρά επαναληπτικά μέσω ενός κύκλου "αξιολόγησης και βελτιστοποίησης", ρυθμίζοντας αυτόνομα τα βάρη του μέχρι να επιτευχθεί ένα προκαθορισμένο όριο ακρίβειας.

Η οριοθέτηση αυτών των συστατικών από το UC Berkeley προσφέρει μια θεμελιώδη κατανόηση του τρόπου λειτουργίας των αλγορίθμων μηχανικής μάθησης, οι οποίοι προσπαθούν συνεχώς να βελτιώνουν τις προβλεπτικές τους ικανότητες μέσω μιας ανατροφοδοτούμενης, επαναληπτικής διαδικασίας. Αυτό το θεμελιώδες πλαίσιο υπογραμμίζει την επαναληπτική και αυτοβελτιούμενη φύση των αλγορίθμων μηχανικής μάθησης, αναδεικνύοντας τον καθοριστικό τους ρόλο σε πληθώρα εφαρμογών σε διάφορους τομείς.

2.1 Μέθοδοι Μηχανικής Μάθησης

Η μηχανική μάθηση αναπτύσσεται ως ένα πολύπλευρο πεδίο, το οποίο διακρίνεται από διάφορα παραδείγματα μάθησης, καθένα από τα οποία είναι προσαρμοσμένο σε συγκεκριμένες ανάγκες ανάλυσης δεδομένων και επίλυσης προβλημάτων. Παρακάτω αναλύονται τέσσερις κλάδοι, ο καθένας με τα μοναδικά του χαρακτηριστικά και εφαρμογές:

1. **Επιβλεπόμενη μηχανική μάθηση:** Στο πεδίο της επιβλεπόμενης μάθησης, τα επισημασμένα σύνολα δεδομένων έχουν τον κύριο ρόλο. Αυτά τα σύνολα δεδομένων επιτρέπουν στους αλγορίθμους να εκπαιδεύονται αποτελεσματικά, δίνοντάς τους τη δυνατότητα να ταξινομούν δεδομένα ή να προβλέπουν αποτελέσματα με ακρίβεια. Καθώς τα δεδομένα εισόδου εισάγονται στο μοντέλο, αυτό προσαρμόζει επαναληπτικά τα εσωτερικά του βάρη μέχρι να επιτευχθεί η κατάλληλη προσαρμογή. Αυτή η προσέγγιση μάθησης αποδεικνύεται ανεκτίμητη για την αντιμετώπιση πραγματικών προκλήσεων σε κλίμακα, από το διαχωρισμό των

ανεπιθύμητων μηνυμάτων ηλεκτρονικού ταχυδρομείου από τα εισερχόμενα, μέχρι μια πληθώρα εργασιών ταξινόμησης. Οι δημοφιλείς μέθοδοι περιλαμβάνουν νευρωνικά δίκτυα, naïve bayes, γραμμική παλινδρόμηση, λογιστική παλινδρόμηση, random forest και support vector machine (SVM).

2. **Μη επιβλεπόμενη μηχανική μάθηση:** Από την άλλη πλευρά, η μάθηση χωρίς επίβλεψη δεν χρειάζεται δεδομένα με ετικέτες. Αντ' αυτού, αξιοποιεί αλγορίθμους μηχανικής μάθησης για την ανάλυση και ομαδοποίηση μη επισημασμένων συνόλων δεδομένων, αναδεικνύοντας λανθάνουσες μορφές και ομαδοποιήσεις χωρίς ανθρώπινη παρέμβαση. Αυτή η εγγενής ικανότητα διάκρισης ομοιοτήτων και διαφορών τοποθετεί τη μάθηση χωρίς επίβλεψη ως πολύτιμο εργαλείο για διερευνητική ανάλυση δεδομένων, τμηματοποίηση πελατών, αναγνώριση εικόνων και προτύπων και στρατηγικές πωλήσεων. Η ανάλυση κύριων συνιστωσών (Principal Component Analysis - PCA) και η αποσύνθεση μοναδιαίων τιμών (Singular Value Decomposition - SVD) συγκαταλέγονται στις εξέχουσες τεχνικές, μαζί με τα νευρωνικά δίκτυα, την ομαδοποίηση k-means και τις πιθανοτικές μεθόδους ομαδοποίησης.
3. **Μάθηση με ημιεπίβλεψη:** Λειτουργώντας ως γέφυρα μεταξύ της επιβλεπόμενης και της μη επιβλεπόμενης μάθησης, η μάθηση με ημιεπίβλεψη συγχωνεύει δεδομένα με ετικέτα με μια μεγαλύτερη δεξαμενή δεδομένων χωρίς ετικέτα κατά τη διάρκεια της εκπαίδευσης. Αυτή η υβριδική προσέγγιση συλλέγει πληροφορίες από το σύνολο δεδομένων με ετικέτες για να καθοδηγήσει την ταξινόμηση και την εξαγωγή χαρακτηριστικών στο σύνολο δεδομένων χωρίς ετικέτες. Αποδεικνύεται ιδιαίτερα ικανή στην αντιμετώπιση σεναρίων όπου τα επισημασμένα δεδομένα είναι ελάχιστα ή το κόστος της επισήμανσης εκτεταμένων δεδομένων είναι απαγορευτικό. Η μέθοδος αυτή επεκτείνει τη χρησιμότητά της σε σενάρια όπου η παραδοσιακή μάθηση με επίβλεψη δεν επαρκεί.
4. **Ενισχυτική μηχανική μάθηση:** Η ενισχυτική μηχανική μάθηση, ενώ έχει κάποια κοινά σημεία με την επιβλεπόμενη μάθηση, αποκλίνει σημαντικά

ως προς τη μεθοδολογία εκπαίδευσής της. Σε αυτό το παράδειγμα, οι αλγόριθμοι δεν εκπαιδεύονται με δειγματικά δεδομένα. Αντ' αυτού, η μάθηση εκτυλίσσεται μέσω δοκιμής και σφάλματος. Τα επιτυχή αποτελέσματα σε μια ακολουθία ενισχύουν την ανάπτυξη της καλύτερης σύστασης ή πολιτικής για την επίλυση ενός συγκεκριμένου προβλήματος. Αυτή η προσέγγιση ευθυγραμμίζεται με σενάρια όπου η επαναληπτική μάθηση, η προσαρμοστικότητα και η συνεχής βελτίωση έχουν ύψιστη σημασία.

Αυτοί οι διακριτοί κλάδοι της μηχανικής μάθησης καταδεικνύουν την ευελιξία αυτού του πεδίου, προσφέροντας ένα ευρύ φάσμα εργαλείων και τεχνικών προσαρμοσμένων σε ποικίλες εφαρμογές, από την ταξινόμηση και την πρόβλεψη δεδομένων έως τη μείωση διαστάσεων και την ανάπτυξη πολιτικών. Η κατανόηση αυτών των παραδειγμάτων εφοδιάζει τους επιστήμονες δεδομένων και τους επαγγελματίες με τα μέσα για να επιλέξουν την καταλληλότερη προσέγγιση για την αντιμετώπιση των προκλήσεων που παρουσιάζονται σε πραγματικές καταστάσεις.

2.2 Προκλήσεις μηχανικής μάθησης

Η εξέλιξη της μηχανικής μάθησης έχει αναμφισβήτητα διευκολύνει πολλούς τομείς στη ζωή μας. Ωστόσο, καθώς οι επιχειρήσεις χρησιμοποιούν όλο και περισσότερο αυτή την τεχνολογία, υπάρχουν πολλές ηθικές ανησυχίες γύρω από τις τεχνολογίες τεχνητής νοημοσύνης. Οι ανησυχίες αυτές συνδέονται με τις κοινωνικές επιπτώσεις της μηχανικής μάθησης, που περιλαμβάνουν:

- **Τεχνολογική μοναδικότητα:** Η έννοια της τεχνολογικής μοναδικότητας, η οποία χαρακτηρίζεται από την τεχνητή νοημοσύνη που ξεπερνά την ανθρώπινη νοημοσύνη, αν και δεν αποτελεί επικείμενη πραγματικότητα, προκαλεί κάποια σημαντικά ερωτήματα. Η άνοδος των αυτόνομων συστημάτων, όπως τα αυτοκινούμενα αυτοκίνητα, οδηγεί σε διλήμματα γύρω από την ευθύνη και την υπευθυνότητα σε περίπτωση ατυχημάτων, επηρεάζοντας και αλλάζοντας τις συζητήσεις σχετικά με την έκταση της αυτονομίας και της ανθρώπινης εποπτείας σε τέτοιες τεχνολογίες.

- **Επιπτώσεις της τεχνητής νοημοσύνης στις θέσεις εργασίας:** Η τεχνητή νοημοσύνη μεταβάλλει αναπόφευκτα τη δυναμική της αγοράς εργασίας, δημιουργώντας αλλαγές στη ζήτηση για ορισμένες θέσεις. Η εμφάνιση της τεχνητής νοημοσύνης απαιτεί ένα εργατικό δυναμικό με εξειδίκευση στη διαχείριση συστημάτων τεχνητής νοημοσύνης και στην αντιμετώπιση σύνθετων προβλημάτων σε όλους τους κλάδους που βιώνουν αλλαγές στη ζήτηση θέσεων εργασίας.
- **Ιδιωτικότητα:** Η συζήτηση για την προστασία της ιδιωτικής ζωής επικεντρώνεται κυρίως γύρω από την ιδιωτικότητα, την προστασία και την ασφάλεια των δεδομένων. Νομοθετικές προσπάθειες, όπως το GDPR και CCPA, τονίζουν την αναγνώριση των δικαιωμάτων των ατόμων επί των δεδομένων τους. Κατά συνέπεια, οι επιχειρήσεις αυξάνουν τις επενδύσεις στην ασφάλεια των δεδομένων και την προστασία της ιδιωτικής ζωής.
- **Προκατάληψη και διακρίσεις:** Οι ανησυχίες σχετικά με την προκατάληψη και τις διακρίσεις έχουν αποκτήσει μεγάλη σημασία, καθώς έρχονται στην επιφάνεια περιπτώσεις συστημάτων μηχανικής μάθησης που αναπαράγουν αυτά τα ζητήματα. Η διασφάλιση έναντι της προκατάληψης και των διακρίσεων γίνεται δύσκολη όταν τα δεδομένα εκπαίδευσης μπορεί να κληρονομούν προκαταλήψεις από ανθρώπινες διαδικασίες.

Ως απάντηση σε αυτές τις ηθικές ανησυχίες, οι επιχειρήσεις έχουν αρχίσει να συμμετέχουν ενεργά σε συζητήσεις σχετικά με την ηθική και τις αξίες της τεχνητής νοημοσύνης. Ορισμένες έχουν λάβει συγκεκριμένα μέτρα για να προσαρμόσουν τις πρωτοβουλίες τους για την τεχνητή νοημοσύνη και τις ηθικές αρχές, αντιπροσωπεύοντας την αυξανόμενη συνειδητοποίηση των κινδύνων που συνδέονται με αυτήν. Παρόλο που μπορεί να υπάρξει σημαντική νομοθεσία στο μέλλον, στο παρόν θα πρέπει οργανισμοί και ερευνητές να εξερευνήσουν το ηθικό πεδίο της τεχνητής νοημοσύνης, προσπαθώντας να υποστηρίξουν τις αρχές που διασφαλίζουν την υπεύθυνη ανάπτυξή της.

2.3 Κοινοί αλγόριθμοι μηχανικής μάθησης

Η μηχανική μάθηση παρουσιάζει ένα πλούσιο σύνολο αλγορίθμων, ο καθένας από τους οποίους είναι προσεκτικά σχεδιασμένος για να ανταποκρίνεται σε συγκεκριμένες ανάγκες ανάλυσης δεδομένων και πρόβλεψης. Μερικοί από τους πιο συχνά χρησιμοποιούμενους αλγορίθμους μηχανικής μάθησης είναι:

- Νευρωνικά δίκτυα (Neural Networks)
- Γραμμική παλινδρόμηση (Linear Regression)
- Λογιστική παλινδρόμηση (Logistic Regression)
- Συσταδοποίηση (Clustering)
- Δέντρα αποφάσης (Decision Trees)
- Τυχαία δάση (Random Forest)
- Support Vector Machines (SVM)
- Naive Bayes
- K-Nearest Neighbor

Αυτοί οι αλγόριθμοι μηχανικής μάθησης αποτελούν τη βάση αμέτρητων εφαρμογών που βασίζονται σε δεδομένα, αντιπροσωπεύοντας την ποικιλομορφία και την προσαρμοστικότητα που καθορίζουν τον τομέα. Η κατανόηση των μοναδικών χαρακτηριστικών και των πλεονεκτημάτων τους εφοδιάζει τους επιστήμονες δεδομένων και τους επαγγελματίες με τα εργαλεία για να εξιχνιάσουν πολύπλοκα μοτίβα δεδομένων, να κάνουν τεκμηριωμένες προβλέψεις και να αποκαλύπτουν χρήσιμες πληροφορίες σε πολλούς τομείς και βιομηχανίες (IBM, n.d.).

2.4 Τεχνικές ταξινόμησης

Η ταξινόμηση, μια θεμελιώδης τεχνική στην εξόρυξη δεδομένων και τη μηχανική μάθηση, επιτρέπει την πρόβλεψη της συμμετοχής σε ομάδες για μεμονωμένες περιπτώσεις δεδομένων. Ενώ υπάρχει μια σειρά τεχνικών στο πεδίο της μηχανικής μάθησης, η ταξινόμηση αποτελεί την πιο εκτεταμένα εφαρμοζόμενη προσέγγιση. Η σημασία της υπογραμμίζεται από τον καθοριστικό της ρόλο στον μελλοντικό σχεδιασμό και την ανακάλυψη γνώσης, καθιστώντας την ένα ευρέως διερευνημένο έργο στο πεδίο της μηχανικής μάθησης. Η ταξινόμηση

αναδεικνύεται ως μία από τις πιο διεξοδικά εξεταζόμενες προκλήσεις στους τομείς της μηχανικής μάθησης και της εξόρυξης δεδομένων, προσελκύοντας την εστιασμένη προσοχή των ερευνητών στην προσπάθειά τους για καινοτόμες λύσεις και βαθύτερη κατανόηση αυτού του δυναμικού πεδίου.

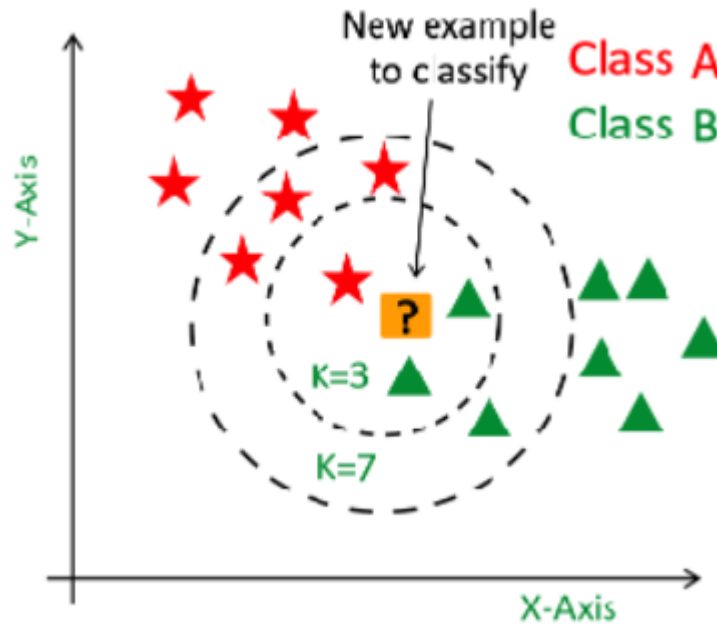
Οι βασικότερες τεχνικές ταξινόμησης με τις κυριότερες λειτουργίες τους παρουσιάζονται παρακάτω:

- **Adaptive Boosting (AdaBoost):** Το Adaptive Boosting (ή AdaBoost), ένας αλγόριθμος μάθησης συνόλου με επίβλεψη, ήταν ο πρώτος αλγόριθμος Boosting που χρησιμοποιήθηκε στην πράξη και αναπτύχθηκε από τους Freund και Schapire το 1995. Με λίγα λόγια, το Adaptive Boosting βοηθά στη μείωση του σφάλματος οποιουδήποτε αλγορίθμου μάθησης ταξινόμησης (όπως τα Δέντρα Αποφάσεων ή οι Μηχανές Διανυσμάτων Υποστήριξης) μετατρέποντας διαδοχικά πολλούς αδύναμους ταξινομητές σε έναν ισχυρό ταξινομητή. Αυτό μπορεί να επιτευχθεί με διαδοχικές προσαρμογές βαρών, μεμονωμένες δυνάμεις ψηφοφορίας και ένα σταθμισμένο άθροισμα των τελικών ταξινομητών του αλγορίθμου (Brugger, 2022).
- **Δέντρα απόφασης:** Οι αλγόριθμοι δέντρων απόφασης είναι τα πιο διαδεδομένα εργαλεία για εργασίες ταξινόμησης. Προσφέρουν μια διαισθητική τεχνική μοντελοποίησης που απλοποιεί τη διαδικασία ταξινόμησης. Τα δέντρα αποφάσεων, γνωστά για τη διαφάνειά τους, επιτρέπουν στους χρήστες να περιηγηθούν σε ένα δομημένο δέντρο, παρέχοντας ένα σαφές μονοπάτι για την κατανόηση της διαδικασίας λήψης αποφάσεων. Ο θεμελιώδης στόχος ενός δέντρου αποφάσεων είναι η δημιουργία ενός μοντέλου που να εξάγει την τιμή μιας επιθυμητής μεταβλητής από ένα πλήθος μεταβλητών εισόδου. Η κατασκευή των αλγορίθμων δέντρων απόφασης τυπικά εκτυλίσσεται σε δύο διακριτές φάσεις: (i) την ανάπτυξη του δέντρου, όπου το σύνολο εκπαίδευσης υφίσταται αναδρομική διάσπαση με βάση τοπικά βέλτιστα κριτήρια έως ότου οι περισσότερες εγγραφές σε ένα τμήμα μοιράζονται την ίδια ετικέτα

κλάσης - και (ii) το κλάδεμα του δέντρου, το οποίο περιορίζει το μέγεθος του δέντρου, καθιστώντας το πιο κατανοητό και φιλικό προς το χρήστη. Αυτή η προσέγγιση δύο φάσεων εξασφαλίζει την ανάπτυξη αποτελεσματικών και ερμηνεύσιμων μοντέλων δέντρων απόφασης.

- **Bayesian Network:** Ένα δίκτυο Bayes (BN) χρησιμεύει ως ένα γραφικό μοντέλο που διευκολύνει την αναπαράσταση πιθανοτικών σχέσεων μεταξύ μιας συλλογής μεταβλητών. Η δομή ενός BN, που συμβολίζεται ως S , ενσωματώνει ένα κατευθυνόμενο ακυκλικό γράφημα (DAG), όπου οι κόμβοι εντός του S δημιουργούν αντιστοιχίες ένα προς ένα με τα χαρακτηριστικά που συμβολίζονται με X . Τα τόξα μεταξύ αυτών των κόμβων απεικονίζουν ζωντανά τις απροσδόκητες επιρροές μεταξύ των μεταβλητών και η σπανιότητα των πιθανών τόξων στο S κωδικοποιεί εξαρτήσεις υπό όρους. Το εγχείρημα της εκμάθησης των Bayesian Networks συνήθως χωρίζεται σε δύο πρωταρχικά επιμέρους καθήκοντα: (α) την εκμάθηση της δομής DAG του δικτύου και (β) τον προσδιορισμό των παραμέτρων. Μια αξιοσημείωτη πρόκληση που σχετίζεται με τους ταξινομητές δικτύων Bayes είναι η εξάρτησή τους από τη διακριτοποίηση συνεχών χαρακτηριστικών. Η μετατροπή των συνεχών χαρακτηριστικών σε διακριτά αντίστοιχα εισάγει προκλήσεις ταξινόμησης, που περιλαμβάνουν ζητήματα όπως ο θόρυβος, τα ελλιπή δεδομένα και η ευαισθησία σε μετασχηματισμούς χαρακτηριστικών που αφορούν μεταβλητές κλάσης (Soofi, 2017).
- **K – Nearest Neighbors (KNN):** Οι K-Nearest Neighbors (KNN) είναι μια καλά αναγνωρισμένη τεχνική ταξινόμησης, ιδιαίτερα πολύτιμη όταν έχουμε να κάνουμε με δεδομένα που δεν έχουν προηγούμενες πληροφορίες σχετικά με την κατανομή τους. Αυτή η μη παραμετρική μέθοδος προσφέρει μια αποτελεσματική λύση που παρακάμπτει τις περιπλοκές της εκτίμησης πυκνότητας πιθανότητας. Ο κανόνας K-Nearest Neighbor ταξινομεί το σημείο δεδομένων x συσχετίζοντάς το με την ετικέτα που παρατηρείται συχνότερα μεταξύ των K πλησιέστερων γειτόνων του. Αυτή η τεχνική σχεδιάστηκε για να διευκολύνει την ανάλυση διάκρισης, ιδίως σε σενάρια όπου η δημιουργία αξιόπιστων παραμετρικών εκτιμήσεων πυκνότητας

πιθανότητας αποδεικνύεται δύσκολο έργο. Η δομή του μοντέλου KNN αναπαρίσταται οπτικά στην παρακάτω εικόνα.

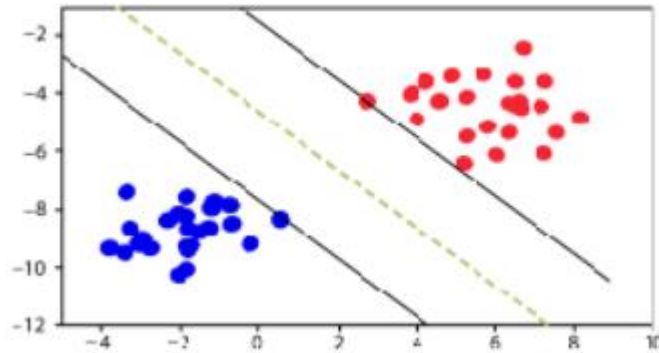


Εικόνα 1: Μοντέλο KNN

Στο πλαίσιο αυτής της οπτικής αναπαράστασης, ο πρωταρχικός στόχος είναι ο προσδιορισμός της κατάλληλης κλάσης στην οποία πρέπει να ανατεθεί ένα νέο παράδειγμα. Για παράδειγμα, εάν το K ρυθμιστεί στο 3, το νέο παράδειγμα θα κατηγοριοποιηθεί ως μέλος της κλάσης B. Αντίθετα, όταν το K ρυθμιστεί στο 7, το νέο παράδειγμα θα βρει τη θέση που του αρμόζει στην κλάση A, όπως απεικονίζεται στην Εικόνα 1. Οι K-Nearest Neighbors προσφέρουν ευελιξία στην ταξινόμηση δεδομένων, καθιστώντας το ένα πολύτιμο εργαλείο στο πεδίο της μηχανικής μάθησης και της ανάλυσης δεδομένων.

- **Support Vector Machines (SVM):** Οι γραμμικοί ταξινομητές, ιδίως οι μηχανές διανυσμάτων υποστήριξης (SVM), μοιράζονται ομοιότητες με τα κλασικά νευρωνικά δίκτυα perceptron πολλαπλών στρωμάτων. Στα SVM, μια μεταβλητή πρόβλεψης αναφέρεται ως χαρακτηριστικό και το μετασχηματισμένο χαρακτηριστικό παίζει καθοριστικό ρόλο στον καθορισμό του υπερεπιπέδου, που αναφέρεται ως χαρακτηριστικό. Η διαδικασία επιλογής χαρακτηριστικών περιλαμβάνει το έργο του

προσδιορισμού της καταλληλότερης αναπαράστασης για ένα δεδομένο πρόβλημα. Ένα διάνυσμα περιλαμβάνει τα χαρακτηριστικά που περιγράφουν συλλογικά μια σειρά τιμών πρόβλεψης. Η έννοια της μηχανής διανυσμάτων υποστήριξης απεικονίζεται οπτικά στην Εικόνα 2.



Εικόνα 2: Παράδειγμα SVM

Στο πλαίσιο αυτής της απεικόνισης, η κίτρινη διακεκομμένη γραμμή συμβολίζει το ιδανικό υπερεπίπεδο, το οποίο διαχωρίζει αποτελεσματικά δύο κλάσεις με το μέγιστο περιθώριο. Αντίθετα, οι δύο άλλες γραμμές καταφέρνουν να διαχωρίσουν τις κλάσεις μόνο με σχετικά στενά περιθώρια, όπως παρουσιάζεται στην Εικόνα 2. Οι Μηχανές Διανυσμάτων Υποστήριξης εκτιμώνται ευρέως για την ικανότητά τους να οριοθετούν τα όρια απόφασης με τρόπο που μεγιστοποιεί το περιθώριο μεταξύ των κλάσεων, καθιστώντας τις πολύτιμα εργαλεία για εργασίες ταξινόμησης στη μηχανική μάθηση και την ανάλυση δεδομένων (Bardab, 2020).

- **Random Forest:** Το Random Forest είναι ένας ευρέως χρησιμοποιούμενος αλγόριθμος μηχανικής μάθησης που αναπτύχθηκε από τους Leo Breiman και Adele Cutler, ο οποίος συνδυάζει την έξοδο πολλαπλών δέντρων απόφασης για να καταλήξει σε ένα ενιαίο αποτέλεσμα. Η ευρεία δημοτικότητα του αλγορίθμου Random Forest οφείλεται στη φιλική προς το χρήστη φύση και την προσαρμοστικότητά του, που του επιτρέπει να αντιμετωπίζει αποτελεσματικά προβλήματα ταξινόμησης και παλινδρόμησης. Η δύναμη του αλγορίθμου έγκειται στην ικανότητά του να χειρίζεται πολύπλοκα σύνολα δεδομένων και να μετριάσει την υπερβολική

προσαρμογή, καθιστώντας τον πολύτιμο εργαλείο για διάφορες προγνωστικές εργασίες στη μηχανική μάθηση. Ένα από τα σημαντικότερα χαρακτηριστικά του αλγορίθμου Random Forest είναι ότι μπορεί να χειριστεί το σύνολο δεδομένων που περιέχει συνεχείς μεταβλητές, όπως στην περίπτωση της παλινδρόμησης, και κατηγορικές μεταβλητές, όπως στην περίπτωση της ταξινόμησης. Αποδίδει καλύτερα για εργασίες ταξινόμησης και παλινδρόμησης (R, 2024).

- **Extreme Gradient Boosting (XGBoost):** Το XGBoost σημαίνει Extreme Gradient Boosting και εφαρμόζει μια τεχνική Gradient Boosting βασισμένη σε δέντρα αποφάσεων. Κατασκευάζει σύντομα, βασικά δέντρα αποφάσεων επαναληπτικά. Κάθε δέντρο χαρακτηρίζεται ως "αδύναμος μαθητής" λόγω της υψηλής προκατάληψής του. Το XGBoost ξεκινά με την κατασκευή του πρώτου βασικού δέντρου που έχει χαμηλή απόδοση. Στη συνέχεια, κατασκευάζει ένα άλλο δέντρο, εκπαιδευμένο να προβλέπει αυτό που δεν μπορεί να κάνει το πρώτο δέντρο, το οποίο είναι ένας αδύναμος μαθητής. Η τεχνική παράγει διαδοχικά ασθενέστερους εκπαιδευόμενους, καθένας από τους οποίους διορθώνει το προηγούμενο δέντρο πριν από την εκπλήρωση της συνθήκης διακοπής, όπως ο αριθμός των δέντρων (εκτιμητών) που πρέπει να δημιουργηθούν. Το XGBoost έχει πρόσθετα πλεονεκτήματα: η εκπαίδευση είναι ταχύτερη και μπορεί να παραλληλιστεί/διανεμηθεί σε συστάδες (Subasi, 2022).

2.5 Τεχνικές Βαθιάς Μάθησης (Deep Learning)

Η βαθιά μάθηση, ένας εξειδικευμένος τομέας της μηχανικής μάθησης, περιλαμβάνει νευρωνικά δίκτυα με τρία ή περισσότερα επίπεδα. Αυτά τα νευρωνικά δίκτυα προσπαθούν να μιμηθούν τη λειτουργία του ανθρώπινου εγκεφάλου και έχουν την ικανότητα να "μαθαίνουν" από εκτεταμένα σύνολα δεδομένων. Ενώ ένα νευρωνικό δίκτυο ενός στρώματος μπορεί να προσφέρει κατά προσέγγιση προβλέψεις, η προσθήκη κρυφών στρωμάτων ενισχύει τις δυνατότητες βελτιστοποίησης και βελτίωσης της ακρίβειας.

Εκτός από τα διάφορα στρώματα (layers), ένα άλλο χαρακτηριστικό των νευρωνικών δικτύων είναι ο ρυθμός μάθησης (learning rate). Ο ρυθμός μάθησης είναι μια υπερπαράμετρος που ελέγχει το μέγεθος του βήματος κατά τη διαδικασία βελτιστοποίησης στην εκπαίδευση ενός νευρωνικού δικτύου. Καθορίζει πόσο προσαρμόζονται οι παράμετροι του μοντέλου σε κάθε επανάληψη.

Η βαθιά μάθηση χρησιμεύει ως κινητήρια δύναμη πίσω από πολυάριθμες εφαρμογές και υπηρεσίες τεχνητής νοημοσύνης (AI) που ενισχύουν την αυτοματοποίηση. Ενισχύει την εκτέλεση τόσο αναλυτικών όσο και φυσικών εργασιών χωρίς την ανάγκη ανθρώπινης παρέμβασης. Η τεχνολογία βαθιάς μάθησης στηρίζει ένα ευρύ φάσμα καθημερινών προϊόντων και υπηρεσιών, όπως ψηφιακούς βοηθούς, τηλεχειριστήρια τηλεόρασης με φωνητική ενεργοποίηση και ανίχνευση απάτης με πιστωτικές κάρτες. Επιπλέον, διαδραματίζει καθοριστικό ρόλο στην προώθηση τεχνολογιών αιχμής, όπως τα αυτοκινούμενα αυτοκίνητα.

2.5.1 Βαθιά Μάθηση vs. Μηχανική Μάθηση

Η βαθιά μάθηση διαφέρει από την παραδοσιακή μηχανική μάθηση μέσω της φύσης των δεδομένων που χειρίζεται και των μεθόδων μάθησης που χρησιμοποιεί.

Οι αλγόριθμοι μηχανικής μάθησης βασίζονται σε δομημένα, επισημασμένα δεδομένα για πρόβλεψη, δηλαδή ορίζουν συγκεκριμένα χαρακτηριστικά από τα δεδομένα εισόδου και τα οργανώνουν σε πίνακες. Αυτό δεν συνεπάγεται αποκλεισμό των μη δομημένων δεδομένων- απλώς σημαίνει ότι, όταν χρησιμοποιούνται μη δομημένα δεδομένα, συχνά υφίστανται κάποια προεπεξεργασία για να διαμορφωθούν σε δομημένη μορφή.

Αντίθετα, η βαθιά μάθηση εξορθολογίζει μεγάλο μέρος της προεπεξεργασίας δεδομένων που συνήθως συνδέεται με τη μηχανική μάθηση. Αυτοί οι αλγόριθμοι μπορούν να προσλαμβάνουν και να επεξεργάζονται μη δομημένα δεδομένα, όπως κείμενο και εικόνες, και να αυτοματοποιούν την εξαγωγή χαρακτηριστικών, μειώνοντας την ανάγκη για ανθρώπινη παρέμβαση.

Για παράδειγμα, σε ένα σύνολο φωτογραφιών διαφόρων κατοικίδιων ζώων υπάρχει η ανάγκη για ταξινόμηση ως "γάτα", "σκύλος", "χάμστερ" κ.ο.κ. Οι αλγόριθμοι βαθιάς μάθησης μπορούν να προσδιορίσουν ποια χαρακτηριστικά (π.χ. αυτιά) είναι πιο σημαντικά για τη διάκριση ενός ζώου από ένα άλλο. Στη μηχανική μάθηση, αυτή η ιεραρχία των χαρακτηριστικών καθορίζεται συνήθως χειροκίνητα από έναν ανθρώπινο εμπειρογνώμονα. Στη συνέχεια, μέσω των διαδικασιών βαθμιαίας καθόδου (gradient descent) και οπισθοδιάδοσης (backpropagation), οι αλγόριθμοι βαθιάς μάθησης ρυθμίζονται με ακρίβεια, ενισχύοντας την ικανότητά τους να κάνουν ακριβείς προβλέψεις για νέες φωτογραφίες ζώων.

Τα μοντέλα μηχανικής μάθησης και βαθιάς μάθησης εμπλέκονται επίσης σε διαφορετικά παραδείγματα μάθησης, τα οποία συχνά κατηγοριοποιούνται ως μάθηση με επίβλεψη, μάθηση χωρίς επίβλεψη και μάθηση ενίσχυσης.

2.5.2 Τρόπος λειτουργίας

Τα νευρωνικά δίκτυα βαθιάς μάθησης, γνωστά και ως τεχνητά νευρωνικά δίκτυα, στοχεύουν στην αναπαραγωγή της περίπλοκης λειτουργίας του ανθρώπινου εγκεφάλου, αξιοποιώντας τις εισροές δεδομένων, τα βάρη και τις προκαταλήψεις σε συνδυασμό. Αυτά τα στοιχεία συνεργάζονται για να επιτύχουν τον ακριβή εντοπισμό, την κατηγοριοποίηση και την περιγραφή των στοιχείων εντός των δεδομένων.

Τα βαθιά νευρωνικά δίκτυα περιλαμβάνουν πολυάριθμα στρώματα διασυνδεδεμένων κόμβων, με κάθε στρώμα να βασίζεται στο προηγούμενο, βελτιώνοντας τις δυνατότητες πρόβλεψης. Αυτό το σύνολο υπολογισμών μέσω του δικτύου αναφέρεται ως προς τα εμπρός διάδοση (forward propagation). Σε ένα βαθύ νευρωνικό δίκτυο, τα στρώματα εισόδου και εξόδου χαρακτηρίζονται ως ορατά στρώματα. Το στρώμα εισόδου χρησιμεύει ως το σημείο εισόδου των δεδομένων στο μοντέλο βαθιάς μάθησης, ενώ στο στρώμα εξόδου αποδίδεται η τελική πρόβλεψη ή ταξινόμηση.

Μια συμπληρωματική διαδικασία, η οπισθοδιάδοση, χρησιμοποιεί αλγόριθμους όπως ο gradient descent για τον υπολογισμό των σφαλμάτων στις προβλέψεις. Στη συνέχεια, οπισθοδρομεί μέσω των στρωμάτων, προσαρμόζοντας τα βάρη και τις προκαταλήψεις της συνάρτησης για την αποτελεσματική εκπαίδευση του μοντέλου. Οι συνδυασμένες δυνάμεις της προς τα εμπρός διάδοσης και της οπισθοδιάδοσης δίνουν στο νευρωνικό δίκτυο τη δυνατότητα να παρέχει προβλέψεις και να διορθώνει τα σφάλματα ανάλογα με τις ανάγκες. Με την πάροδο του χρόνου, ο αλγόριθμος βελτιώνει σταδιακά την ακρίβειά του.

Τα παραπάνω διευκρινίζουν με απλουστευμένους όρους τις θεμελιώδεις λειτουργίες του απλούστερου τύπου βαθύ νευρωνικού δικτύου. Παρ' όλα αυτά, οι αλγόριθμοι βαθιάς μάθησης είναι εξαιρετικά περίπλοκοι και περιλαμβάνουν διάφορους τύπους νευρωνικών δικτύων προσαρμοσμένους σε συγκεκριμένους προβληματικούς τομείς και σύνολα δεδομένων. Για παράδειγμα:

- Τα Convolutional Neural Networks (CNN), που εφαρμόζονται κυρίως στην όραση υπολογιστών και την ταξινόμηση εικόνων, μπορούν να διακρίνουν χαρακτηριστικά και μοτίβα σε μια εικόνα, επιτρέποντας εργασίες όπως η ανίχνευση ή η αναγνώριση αντικειμένων. Το 2015, ένα CNN πέτυχε ένα ιστορικό ορόσημο ξεπερνώντας για πρώτη φορά έναν άνθρωπο σε μια πρόκληση αναγνώρισης αντικειμένων.
- Τα Recurrent Neural Networks (RNN) βρίσκουν τη θέση τους σε εφαρμογές επεξεργασίας φυσικής γλώσσας και αναγνώρισης ομιλίας, αξιοποιώντας διαδοχικά δεδομένα ή δεδομένα χρονοσειρών για να προσφέρουν τις δυνατότητές τους (IBM, n.d).

2.6 Εργαλεία μηχανικής μάθησης

Παρακάτω παρουσιάζονται δύο εργαλεία μηχανικής μάθησης, τα οποία χρησιμοποιούνται στη συγκεκριμένη έρευνα. Πρόκειται για δύο από τα πιο γνωστά εργαλεία και με την μεγαλύτερη απήχηση, όσον αφορά μελέτη συστημάτων και εφαρμογές μηχανικής μάθησης.

2.6.1 Scikit-Learn

Το Scikit-learn είναι μια βιβλιοθήκη Python ανοικτού κώδικα που περιλαμβάνει μια σειρά από τεχνικές μάθησης χωρίς επίβλεψη και με επίβλεψη. Βασίζεται σε τεχνολογίες και βιβλιοθήκες όπως οι Matplotlib, Pandas και NumPy, απλοποιώντας τη διαδικασία κωδικοποίησης.

Βασικά χαρακτηριστικά του Scikit-learn:

- Επιλογή μοντέλου
- Ταξινόμηση
- Προεπεξεργασία
- Ομαδοποίηση
- Παλινδρόμηση (που καλύπτει τη λογιστική και τη γραμμική παλινδρόμηση)

Το Scikit-learn δίνει τη δυνατότητα στους χρήστες να ορίζουν και να συγκρίνουν αλγορίθμους μηχανικής μάθησης, ενώ παράλληλα προσφέρει εργαλεία προεπεξεργασίας δεδομένων. Υποστηρίζει ένα ευρύ φάσμα μοντέλων μηχανικής μάθησης, από την ομαδοποίηση K-means έως τα random forest και support vector machines, προωθώντας την ευέλικτη ανάπτυξη μοντέλων.

Η πραγματική ικανότητα του Scikit-learn φαίνεται μέσα από το πλαίσιο αξιολόγησης και επιλογής μοντέλων, διευκολύνοντας τη διασταυρούμενη επικύρωση και την εκτεταμένη αναζήτηση υπερπαραμέτρων. Αυτή η εκτεταμένη βιβλιοθήκη βοηθά στον εντοπισμό του καταλληλότερου μοντέλου για μια δεδομένη εργασία.

Κάποια πλεονεκτήματα του scikit-learn είναι:

- Η λεπτομερής τεκμηρίωση API στον ιστότοπο της scikit-learn απλοποιεί την ενσωμάτωση του αλγορίθμου σε διάφορες πλατφόρμες.
- Ένα εκτεταμένο δίκτυο συνεργατών, συγγραφέων και μια ζωντανή διεθνής διαδικτυακή κοινότητα εξασφαλίζει συνεχή υποστήριξη και ενημερώσεις.
- Φιλικό προς τον χρήστη και απλό στην εφαρμογή.

- Η βιβλιοθήκη λειτουργεί υπό την άδεια BSD, παρέχοντας ελεύθερη προσβασιμότητα με ελάχιστους αδειοδοτικούς και νομικούς περιορισμούς.
- Το scikit-learn μπορεί να υπερηφανεύεται για την προσαρμοστικότητα και τη δυνατότητα εφαρμογής σε ένα ευρύ φάσμα εργασιών του πραγματικού κόσμου, από τη νευροαπεικόνιση έως την πρόβλεψη της συμπεριφοράς των καταναλωτών και όχι μόνο.

Ωστόσο:

- Δεν αποτελεί τη βέλτιστη επιλογή για όσους αναζητούν δυνατότητες βαθιάς μάθησης.
- Η απλουστευμένη αφαίρεσή του μπορεί να βάλει σε πειρασμό τους αρχάριους επιστήμονες δεδομένων να παρακάμψουν τη θεμελιώδη μάθηση προτού προχωρήσουν σε πιο προηγμένες πρακτικές.

2.6.2 TensorFlow

Από την άλλη μεριά, το TensorFlow, ένα πλαίσιο ανοικτού κώδικα που συντηρείται από τη Google, έχει σχεδιαστεί για την προτυποποίηση και την αξιολόγηση μοντέλων μηχανικής μάθησης, με κύρια έμφαση στα νευρωνικά δίκτυα. Διαθέτει συμβατότητα με πολλές γλώσσες προγραμματισμού, συμπεριλαμβανομένων των Swift, Python, Go, Javascript, Java και C++, και απολαμβάνει υποστήριξη από την κοινότητα για πολλές άλλες.

Το TensorFlow εξορθολογίζει τον αριθμητικό προγραμματισμό χαμηλού επιπέδου μέσω μιας υψηλού επιπέδου και αφηρημένης προσέγγισης. Υποστηρίζει επίσης βιβλιοθήκες που επιτρέπουν την εκτέλεση εφαρμογών σε τυπικές CPU χωρίς να απαιτούνται τροποποιήσεις. Το TensorFlow επεκτείνει την εμβέλειά του σε διάφορα συστήματα, συμπεριλαμβανομένων των Linux, Android, macOS και Windows. Επιπλέον, το Google Cloud Machine Learning Engine είναι εξοπλισμένο για την εκτέλεση μοντέλων TensorFlow χωρίς τους περιορισμούς των παραδοσιακών υπολογιστικών πλατφορμών.

Αν και το TensorFlow συνδέεται συχνά με τα νευρωνικά δίκτυα, είναι άριστα κατάλληλο για ένα ευρύ φάσμα μεθόδων μηχανικής μάθησης που βασίζονται σε κλίσεις, όπως τα Boosted Trees. Το TensorFlow παρουσιάζει επίσης το TensorBoard, ένα εργαλείο οπτικοποίησης για τη σύγκριση και την παρακολούθηση των μαθημένων μοντέλων. Η γοητεία του TensorFlow έγκειται στην ταχύτητα και τη βελτιστοποίησή του για νευρωνικά δίκτυα. Λίγα πλαίσια μπορούν να συγκριθούν με την ικανότητά του να εκτελεί μοντέλα σε GPUs, CPUs και TPUs.

Πλεονεκτήματα του TensorFlow:

- Γρήγοροι και αποδοτικοί υπολογισμοί μαθηματικών εκφράσεων.
- Ικανότητα δημιουργίας ποικίλων μοντέλων ακολουθίας και εκπαίδευσης βαθιών νευρωνικών δικτύων για εργασίες όπως η ταξινόμηση χειρόγραφων ψηφίων.
- Μοναδικό χαρακτηριστικό βελτιστοποίησης μνήμης και δεδομένων.
- Ισχυρή υποστήριξη από την Google, με αποτέλεσμα τακτικές κυκλοφορίες χαρακτηριστικών, γρήγορες αναβαθμίσεις και ομαλές επιδόσεις.
- Συμβατότητα με μια ποικιλία λογισμικού backend, συμπεριλαμβανομένων ASICs και GPUs, με ακραίο παραλληλισμό.
- Υποστηρίζεται από μια ισχυρή και ενεργή κοινότητα.
- Επιτρέπει την εκτέλεση συγκεκριμένων τμημάτων ενός γραφήματος, διευκολύνοντας τον διακριτό χειρισμό δεδομένων.
- Ανώτερες οπτικοποιήσεις υπολογιστικών γράφων σε σύγκριση με τις εγγενείς βιβλιοθήκες όπως το Theano και το Torch.
- Καινοτόμος προσέγγιση για την παρακολούθηση των μετρικών και την παρακολούθηση της προόδου της εκπαίδευσης του μοντέλου.
- Εξαιρετικές επιδόσεις, που το τοποθετούν στο ίδιο επίπεδο με τα καλύτερα του κλάδου.
- Εγκατάσταση σε μηχανές υλικού για επεκτασιμότητα.

Μειονεκτήματα του TensorFlow:

- Υστερεί σε σχέση με τους ανταγωνιστές όσον αφορά τη χρηστικότητα και την ταχύτητα.
- Περιορισμένη υποστήριξη GPU (NVIDIA) και πλήρης υποστήριξη γλώσσας (Python), γεγονός που μπορεί να αποτελέσει μειονέκτημα δεδομένης της εμφάνιση εναλλακτικών γλωσσών βαθιάς μάθησης.
- Περιορισμένες δυνατότητες για το χειρισμό ακολουθιών μεταβλητού μήκους, με την πεπερασμένη αναδίπλωση ως συνιστώμενη λύση.
- Ανεπαρκής υποστήριξη για χρήστες των Windows, με εναλλακτικές μεθόδους εγκατάστασης μέσω της βιβλιοθήκης πακέτων Python (pip) ή του conda.
- Απότομη καμπύλη εκμάθησης, ιδίως για αρχάριους.
- Έλλειψη υποστήριξης OpenCL.
- Η μοναδική δομή καθιστά τον εντοπισμό και την αντιμετώπιση σφαλμάτων πιο δύσκολη.
- Η έμφαση στο περιβάλλον παραγωγής έναντι των επιδόσεων μπορεί να οδηγήσει σε περιορισμούς της υπολογιστικής ταχύτητας.
- Απαιτεί στέρεα θεμέλια σε προχωρημένα μαθηματικά, γραμμική άλγεβρα και ολοκληρωμένη κατανόηση της μηχανικής μάθησης, καθιστώντας το λιγότερο φιλικό προς τους αρχάριους.

2.6.3 Σύγκριση

Με βάση τα προηγούμενα, ακολουθεί πίνακας σύγκρισης των Scikit-Learn και TensorFlow (Simplelearn, 2023).

| TensorFlow | Scikit-Learn |
|---|---|
| Ένα νευρωνικό δίκτυο χρησιμοποιείται για τη βελτιστοποίηση του TensorFlow. | Με άλλα πλαίσια όπως το XGBoost, το Scikit-learn είναι πιο ευέλικτο. |
| Το TensorFlow χρησιμοποιείται στη διαδικασία σχεδιασμού για να βοηθήσει τους προγραμματιστές, καθώς και για τη συγκριτική αξιολόγηση νέων μοντέλων. | Το Scikit-learn χρησιμοποιείται επίσης για τη δημιουργία και τη συγκριτική αξιολόγηση του νέου μοντέλου, καθώς και για το σχεδιασμό και την υποβοήθηση των προγραμματιστών. |
| Το TensorFlow είναι μια βιβλιοθήκη χαμηλού επιπέδου που βοηθά στην | Ο αλγόριθμος μηχανικής μάθησης υλοποιείται επίσης με τη χρήση του |

| | |
|--|---|
| υλοποίηση τεχνικών και αλγορίθμων μηχανικής μάθησης. | Scikit-learn, μιας βιβλιοθήκης υψηλότερου επιπέδου. |
| Πρόκειται για μια ενότητα τρίτου μέρους. Ωστόσο, χρησιμοποιείται ευρύτερα. | Πρόκειται επίσης για μια ενότητα τρίτου μέρους, τη Scikit-learn, η οποία είναι λιγότερο δημοφιλής από την TensorFlow. |
| Το TensorFlow χρησιμοποιεί την κλάση base για την υλοποίηση όλων των αλγορίθμων του. | Όλοι οι αλγόριθμοι της Scikit-learn χρησιμοποιούνται ως βασικός εκτιμητής. |
| Το TensorFlow είναι ένα πλαίσιο βαθιάς μάθησης. | Το Scikit-learn χρησιμοποιείται κυρίως σε εφαρμογές μηχανικής μάθησης. |
| Το νευρωνικό δίκτυο χρησιμοποιείται έμμεσα από το TensorFlow. | Στην πράξη, το Scikit-learn χρησιμοποιείται με ένα ευρύ φάσμα μοντέλων. |
| Παρέχει βελτιστοποίηση εξειδίκευσης κάτω από την κουκούλα, διευκολύνοντας τη σύγκριση μοντέλων νευρωνικών δικτύων και μοντέλων TensorFlow. | Είναι δυνατή η σύγκριση εντελώς διαφορετικών παραλλαγών μοντέλων μηχανικής μάθησης με τη χρήση του Scikit-learn. |
| Το TensorFlow είναι μια υλοποίηση νευρωνικών δικτύων χωρίς καθόλου στοιχεία. | Ένα μοντέλο νευρωνικού δικτύου που είναι barebone δεν υλοποιείται στο Scikit-learn. |

Πίνακας 1: Σύγκριση TensorFlow & Scikit-Learn

3 AutoML

Η αυτοματοποιημένη μηχανική μάθηση (Automated Machine Learning - AutoML) προσφέρει τεχνικές και διαδικασίες που έχουν σχεδιαστεί για να εκδημοκρατίσουν τη μηχανική μάθηση, να ενισχύσουν την αποτελεσματικότητά της και να επιταχύνουν την έρευνα στον τομέα.

Η Μηχανική Μάθηση έχει κάνει αξιοσημείωτα βήματα τα τελευταία χρόνια, βρίσκοντας εφαρμογές σε διάφορους τομείς. Ωστόσο, η επιτυχία της εξαρτάται σε μεγάλο βαθμό από τους ανθρώπινους εμπειρογνώμονες στη μηχανική μάθηση, οι οποίοι χειρίζονται τα ακόλουθα καθήκοντα:

- Προεπεξεργασία και καθαρισμός δεδομένων.
- Επιλογή και μηχανική των χαρακτηριστικών.
- Επιλογή της σωστής οικογένειας μοντέλων.

- Ρύθμιση των υπερπαραμέτρων του μοντέλου.
- Σχεδιασμός αρχιτεκτονικών νευρωνικών δικτύων (για βαθιά μάθηση).
- Μετεπεξεργασία μοντέλων μηχανικής μάθησης.
- Διεξοδική ανάλυση των αποτελεσμάτων που λαμβάνονται.

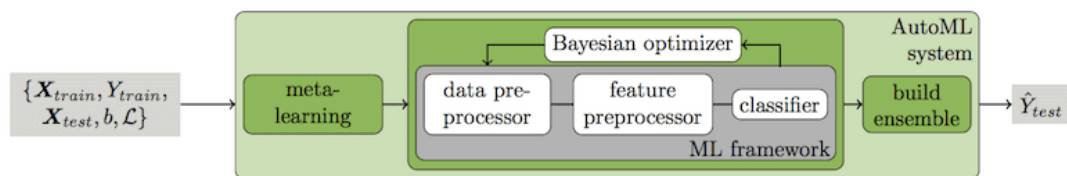
Δεδομένου ότι αυτές οι εργασίες συχνά ξεπερνούν τις δυνατότητες των μη ειδικών στη μηχανική μάθηση, η ταχεία εξάπλωση των εφαρμογών μηχανικής μάθησης έχει δημιουργήσει μια ζήτηση για φιλικές προς το χρήστη, προκατασκευασμένες μεθόδους μηχανικής μάθησης που δεν απαιτούν γνώσεις ειδικών. Αυτός ο αναδυόμενος τομέας, ο οποίος επικεντρώνεται στη σταδιακή αυτοματοποίηση των διαδικασιών μηχανικής μάθησης, αναφέρεται ως αυτοματοποιημένη μηχανική μάθηση ή AutoML (AutoML, n.d.).

3.1 Εργαλεία AutoML

Μια πληθώρα πλαισίων AutoML έχει αναπτυχθεί από διάφορες δομές για την αντιμετώπιση προκλήσεων σε διάφορους τομείς. Δύο από αυτά τα εργαλεία παρουσιάζονται παρακάτω.

3.1.1 AutoSklearn

Το AutoSklearn ξεχωρίζει ως ένα από τα πρωτοποριακά και επιτυχημένα εργαλεία AutoML, που βασίζεται στα θεμέλια του scikit-learn. Κατασκευάζει έναν αγωγό μηχανικής μάθησης από άκρη σε άκρη και προσφέρει πολύτιμες πληροφορίες για μελλοντικές έρευνες. Η δομή του παρουσιάζεται στην Εικόνα 3.



Εικόνα 3: Δομή AutoSklearn

Η ουσία του Auto-sklearn εξαρτάται από τη διαδικασία βελτιστοποίησης Bayesian, η οποία ενισχύεται από δύο σημαντικές βελτιώσεις. Πρώτον, ενσωματώνει μια ενότητα μετα-μάθησης που χρησιμεύει ως προθέρμανση για τον βελτιστοποιητή Bayes, ενισχύοντας έτσι τη συνολική απόδοση στην επιλογή του

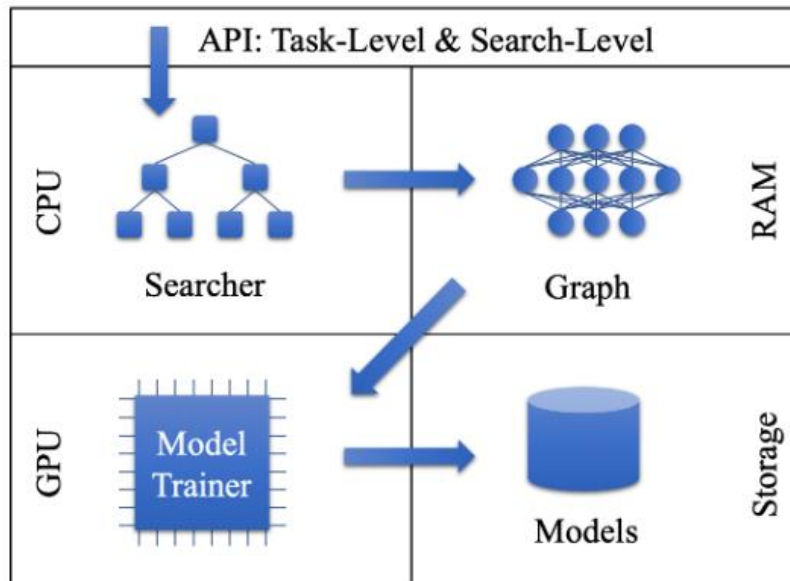
καταλληλότερου πλαισίου ML. Δεύτερον, εισάγεται μια ενότητα κατασκευής συνόλου ως βήμα μετα-επεξεργασίας, με στόχο τη βελτίωση της βελτιστοποίησης των υπερπαραμέτρων και την ενίσχυση της ευρωστίας του πλαισίου.

Το AutoSklearn απελευθερώνει τους χρήστες από την κουραστική βελτιστοποίηση των υπερπαραμέτρων και την επιλογή αλγορίθμων. Με επίκεντρο το scikit-learn, διαθέτει εξαιρετική συμβατότητα με σύνολα δεδομένων που έχουν προηγουμένως ελεγχθεί στην πλατφόρμα scikit-learn. Ωστόσο, αυτή η εξειδίκευση έχει κόστος, καθώς το AutoSklearn είναι λιγότερο ικανό στο χειρισμό εκτεταμένων συνόλων δεδομένων και σύγχρονων συστημάτων βαθιάς μάθησης.

3.1.2 Autokeras

Το AutoKeras, αντίθετα, είναι ένα πλαίσιο AutoML που περιστρέφεται γύρω από το Keras και ενσωματώνει δυνατότητες αναζήτησης νευρωνικών αρχιτεκτονικών. Σχεδιάστηκε με σκοπό να απλοποιήσει τη μηχανική μάθηση για μη ειδικούς, δίνοντας ιδιαίτερη έμφαση σε εργασίες βαθιάς μάθησης. Επιπλέον, είναι βελτιστοποιημένο για τοπικό υπολογισμό, εξυπηρετώντας χρήστες που δεν αισθάνονται άνετα με τον προγραμματισμό στο cloud.

Το AutoKeras χρησιμοποιεί μια καλά δομημένη παράλληλη ροή εργασίας, αξιοποιώντας την υπολογιστική ισχύ τόσο της CPU όσο και της GPU ταυτόχρονα για να παρέχει αξιοσημείωτες επιδόσεις. Όπως απεικονίζεται στην Εικόνα 4, η αρχιτεκτονική του AutoKeras περιλαμβάνει την ενότητα Searcher που εκτελείται στην CPU, χρησιμοποιώντας τη βελτιστοποίηση Bayesian για τη δημιουργία νέων αρχιτεκτονικών. Ταυτόχρονα, η GPU αναλαμβάνει να εκπαιδεύσει το σύνολο δεδομένων χρησιμοποιώντας την καθορισμένη αρχιτεκτονική. Μόλις ολοκληρωθεί η εκπαίδευση, το μοντέλο αποθηκεύεται και τα δεδομένα επιδόσεων αναμεταδίδονται στην CPU για την ενημέρωση της διαδικασίας Gauss.



Εικόνα 4: Δομή AutoKeras

Αυτή η παράλληλη ροή εργασιών ελαχιστοποιεί ουσιαστικά τον χρόνο αδράνειας και αυξάνει την αποδοτικότητα. Η ομάδα AutoKeras υποστηρίζει ότι το πλαίσιο της χρησιμοποιεί τη βελτιστοποίηση Bayesian για την αναζήτηση νευρωνικών δικτύων, προσφέροντας ανώτερες επιδόσεις σε σύγκριση με τις συμβατικές τεχνικές συντονισμού υπερπαραμέτρων και αναζήτησης νευρωνικών αρχιτεκτονικών. Περιγράφουν επίσης τις μελλοντικές κατευθύνσεις για το AutoKeras, οι οποίες περιλαμβάνουν τη διεύρυνση του χώρου αναζήτησης με επαναλαμβανόμενα νευρωνικά δίκτυα, τη συγχώνευση της διαδικασίας συντονισμού και τον σχεδιασμό ενός συστήματος αναζήτησης νευρωνικής αρχιτεκτονικής προσαρμοσμένου στις εργασίες (Ge, 2020).

3.2 Εφαρμογή AutoML για την ανάπτυξη μοντέλων

Το AutoML διαδραματίζει καθοριστικό ρόλο στην ενίσχυση της διαχείρισης της ανάπτυξης μοντέλων για τους οργανισμούς. Διευκολύνει τον πειραματισμό για τους επαγγελματίες της επιστήμης δεδομένων και τους προγραμματιστές, παρέχοντάς τους ταχεία πρόσβαση στα απαιτούμενα εργαλεία για την ταχεία ανάπτυξη μοντέλων. Οι λύσεις αυτές είναι ευέλικτες, φιλοξενούν τεράστια σύνολα δεδομένων και προσφέρουν προσαρμόσιμες λειτουργίες που βελτιώνουν την προετοιμασία των δεδομένων. Υπερέχουν σε δυνατότητες ανάπτυξης στο cloud, υποστηρίζουν ανάλυση βίντεο και ενισχύουν την επεκτασιμότητα του μοντέλου.

Σε διάφορους κλάδους, οι οργανισμοί αναζητούν ενεργά λύσεις τεχνητής νοημοσύνης. Το AutoML αναδεικνύεται ως η πύλη για την προσβασιμότητα στην ΤΝ, εξυπηρετώντας άτομα με διαφορετικά επίπεδα τεχνογνωσίας σχετικά με την ΤΝ και το ML στο επιχειρηματικό οικοσύστημα.

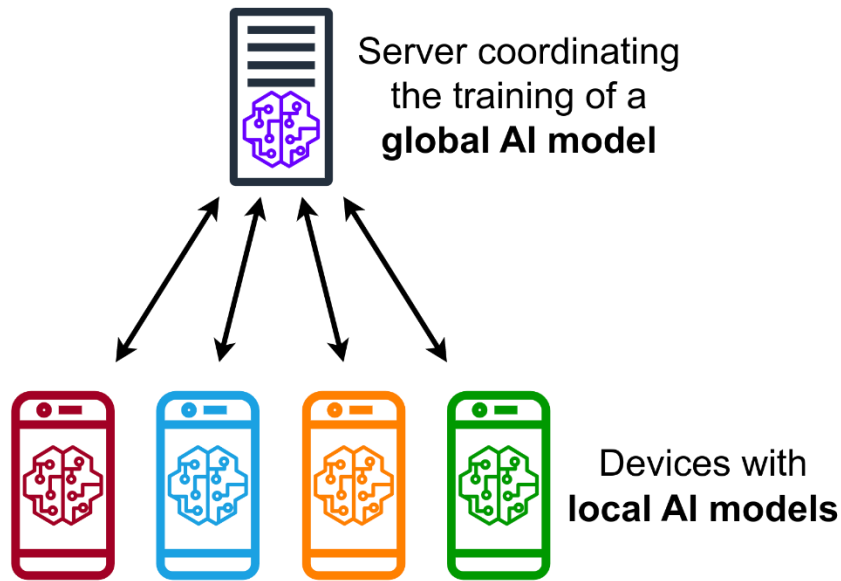
- **Feature engineering:** Η μηχανική χαρακτηριστικών περιλαμβάνει την επιλογή κρίσιμων στοιχείων από τα ακατέργαστα σύνολα δεδομένων για χρήση στην ανάπτυξη μοντέλων ML. Η αυτοματοποιημένη μηχανική χαρακτηριστικών μειώνει δραστικά τον χρόνο που απαιτείται για τη διαδικασία αυτή, η οποία παραδοσιακά καταναλώνει ημέρες καθώς οι επιστήμονες δεδομένων πειραματίζονται με διαφορετικούς συνδυασμούς χαρακτηριστικών. Το AutoML συμβάλλει καθοριστικά στον μετριασμό των προκλήσεων της μηχανικής χαρακτηριστικών, στην απλοποίηση του ορισμού των χαρακτηριστικών και στην παρακολούθηση των εξελισσόμενων χαρακτηριστικών και μοντέλων.
- **Hyperparameter optimization:** Η βελτιστοποίηση των υπερπαραμέτρων περιλαμβάνει τη λεπτομερή ρύθμιση των μοντέλων ML, μια εργασία που μπορεί να εκτελεστεί χειροκίνητα για απλούστερα μοντέλα, αλλά γίνεται εκθετικά πιο δύσκολη για μοντέλα βαθιάς μάθησης. Η αυτοματοποίηση αυτής της διαδικασίας επιτρέπει στους προγραμματιστές να μετατοπίσουν την εστίασή τους από τους μηχανισμούς της δημιουργίας μοντέλων στους λόγους που κρύβονται πίσω από αυτή. Αυτή η μετατόπιση ενισχύει τις αναλυτικές προσπάθειες, επιτρέποντας στους προγραμματιστές και τους επιστήμονες δεδομένων να επικεντρωθούν στη βελτίωση συγκεκριμένων πτυχών των μοντέλων τους. Η προσέγγιση αυτή αποδεικνύεται πολύτιμη σε ένα φάσμα βιομηχανικών περιπτώσεων χρήσης, όπως η ανίχνευση απάτης στον τομέα των χρηματοπιστωτικών υπηρεσιών.
- **Neural Architecture Search (NAS):** Ο σχεδιασμός μιας νευρωνικής αρχιτεκτονικής για εφαρμογές βαθιάς μάθησης είναι ένα τρομερό, υπολογιστικά εντατικό έργο. Ο ρόλος του AutoML σε αυτόν τον τομέα είναι να εκδημοκρατίσει τη βαθιά μάθηση, ενισχύοντας την αποδοτικότητα των μοντέλων και ελαχιστοποιώντας τα σφάλματα κατά την ανάπτυξη

μοντέλων. Οι λειτουργίες του AutoML επιταχύνουν την αξιολόγηση των αρχιτεκτονικών, επιτρέποντας στους προγραμματιστές να αξιοποιήσουν την προσαρμοστικότητα των NAS και να βελτιώσουν τη βελτιστοποίηση και την απόδοση. Το NAS εισάγει ευελιξία σε διάφορους κλάδους και αποκομίζει σημαντικά οφέλη από την αυτοματοποίηση (Loon, 2022).

4 Federated Learning

Το Federated Learning (FL) βρίσκεται στην πρώτη γραμμή των καινοτομιών της μηχανικής μάθησης, σχεδιασμένη για να αντιμετωπίσει την πρόκληση των δεδομένων, διατηρώντας παράλληλα το απόρρητό τους. Το FL περιλαμβάνει τη συνεργασία πολλαπλών πελατών, από κινητές συσκευές έως ιδρύματα και οργανισμούς, σε ένα αποκεντρωμένο παράδειγμα μηχανικής μάθησης. Αυτή η πρωτοποριακή προσέγγιση παρουσιάστηκε για πρώτη φορά από την Google το 2016 ως λύση για την πρόβλεψη της εισαγωγής κειμένου από τον χρήστη σε δεκάδες χιλιάδες συσκευές Android, διατηρώντας παράλληλα το απόρρητο των δεδομένων στις μεμονωμένες συσκευές.

Η βασική διαδικασία του FL, όπως απεικονίζεται στην Εικόνα 5, αναφέρεται συχνά ως "Federated Average" (FedAvg), χρησιμεύοντας ως το θεμελιώδες πλαίσιο για διάφορες ερευνητικές προσπάθειες του FL.



Εικόνα 5: Λειτουργία Federated Learning

(Πηγή: https://en.wikipedia.org/wiki/Federated_learning)

- Αρχικά, κάθε συσκευή κατεβάζει ένα κοινό παγκόσμιο μοντέλο για να ξεκινήσει την τοπική εκπαίδευση.
- Στη συνέχεια, το παγκόσμιο μοντέλο που έχει ληφθεί τελειοποιείται μέσω πολλαπλών τοπικών ενημερώσεων, αξιοποιώντας τοπικά δεδομένα σε διαφορετικές κινητές συσκευές. Αυτές οι τοπικές ενημερώσεις μεταδίδονται με ασφάλεια στο cloud σε κρυπτογραφημένη μορφή.
- Στη συνέχεια, το cloud υπολογίζει μια μέση ενημέρωση χρησιμοποιώντας τα τοπικά μοντέλα και αυτό το βελτιωμένο παγκόσμιο μοντέλο αποστέλλεται πίσω στις μεμονωμένες συσκευές.
- Αυτή η διαδικασία επαναλαμβάνεται μέχρι το μοντέλο να επιτύχει την επιθυμητή απόδοση ή να τηρηθεί η καθορισμένη προθεσμία.

Το FL προσφέρει μια πρωτοποριακή λύση στο συνεχιζόμενο αίνιγμα της εξισορρόπησης της ιδιωτικότητας των δεδομένων με την επιτακτική ανάγκη της κοινής χρήσης δεδομένων σε διασκορπισμένες συσκευές. Ο εγγενής σχεδιασμός του διασφαλίζει ότι τα δεδομένα παραμένουν ασφαλή και ιδιωτικά, θωρακίζοντάς τα από κεντρικούς διακομιστές τρίτων. Αυτό καθιστά το FL ιδανική επιλογή για εφαρμογές όπου η ευαισθησία των δεδομένων είναι υψίστης σημασίας, ιδίως σε τομείς όπως η υγειονομική περίθαλψη και οι κινητές συσκευές, όπου η

συγκέντρωση δεδομένων πρέπει να προσεγγίζεται με υψηλό βαθμό νομικού ελέγχου και συμμόρφωσης.

4.1 Εφαρμογές federated learning σε μοντέλα πρόβλεψης

Το FL έχει αναδειχθεί σε κυρίαρχο παράδειγμα, επιτρέποντας την ανάπτυξη συνεργατικών μοντέλων, ενώ παράλληλα παρακάμπτει τις νομικές περιπλοκές. Παρά την αντιμετώπιση διαφόρων περιορισμών και τρομερών προκλήσεων, οι πρώτοι που το υιοθέτησαν αναγνώρισαν τις σημαντικές δυνατότητες του FL και ξεκίνησαν μια σειρά από διερευνήσεις και πρακτικές εφαρμογές. Οι κυριότεροι κλάδοι στους οποίους εφαρμόζεται το FL είναι (Li, 2020):

- Εφαρμογές για κινητές συσκευές
- Εφαρμογές στη βιομηχανία
- Εφαρμογές στην υγειονομική περίθαλψη

Επιπλέον, αυτή η καινοτόμος τεχνική έχει βρει πρόσφατες εφαρμογές στα δίκτυα οχημάτων, εξυπηρετώντας διάφορους σκοπούς. Για παράδειγμα, μια πρωτοποριακή συγχώνευση του FL με ένα νευρωνικό δίκτυο επαναλαμβανόμενων μονάδων με πύλες έδωσε τη δυνατότητα σε πολλούς οργανισμούς να προβλέψουν συλλογικά τα μελλοντικά πρότυπα ροής της κυκλοφορίας χωρίς κανένα συμβιβασμό στην ιδιωτικότητα. Επιπλέον, οι μεθοδολογίες καταναεμημένης μάθησης έχουν αξιοποιηθεί για την παροχή ισχυρών, διατηρώντας την ιδιωτικότητα προβλέψεων ταχύτητας κυκλοφορίας, διασφαλίζοντας παράλληλα ευαίσθητες τοπολογικές πληροφορίες.

Πέρα από τα ζητήματα προστασίας της ιδιωτικής ζωής, το FL έχει επίσης υιοθετηθεί στο πλαίσιο κινητών δικτύων οχημάτων, υποσχόμενο να βελτιώσει τις επικοινωνίες οχημάτων εξασφαλίζοντας υπερ-αξιοπιστία και χαμηλή καθυστέρηση. Επιπλέον, διαδραματίζει καθοριστικό ρόλο στην υποστήριξη εφαρμογών τεχνητής νοημοσύνης, συμπεριλαμβανομένης της ταξινόμησης εικόνων (Huang, 2021).

4.2 Προκλήσεις

Η αποτελεσματικότητα ενός καθολικού συστήματος FL εξαρτάται από διάφορους κρίσιμους παράγοντες: σφάλμα προσέγγισης (bias), σφάλμα εκτίμησης (διακύμανση) και σφάλμα βελτιστοποίησης (απόκλιση από το επιδιωκόμενο ελάχιστο). Αυτοί οι παράγοντες επηρεάζονται επίσης από το συγκεκριμένο μοντέλο που χρησιμοποιείται, τα χαρακτηριστικά των $Y | X$ και την κατανομή των δεδομένων. Αν και εκκρεμεί μια ολοκληρωμένη διερεύνηση της αλληλεπίδρασής τους σε διάφορα σενάρια, πρόσθετες προκλήσεις του πραγματικού κόσμου περιπλέκουν την εφαρμογή του FL.

- **Περιορισμοί μνήμης και υπολογισμού:** Οι περιορισμοί μνήμης και παίζουν σημαντικό ρόλο. Τα συστήματα FL που έχουν σχεδιαστεί για ακραίες συσκευές πρέπει να λαμβάνουν υπόψη τη διαθέσιμη χωρητικότητα υλικού σε κάθε συσκευή. Πρέπει να ληφθούν υπόψη δύο βασικές συμβιβαστικές λύσεις. Πρώτον, όταν πρόκειται για περιορισμούς μνήμης, τα μοντέλα στη συσκευή πρέπει να διατηρούνται συμπαγή. Αυτό παρουσιάζει ένα μεθοδολογικό δίλημμα για τον διακομιστή, καθώς πρέπει με κάποιο τρόπο να αξιοποιήσει μια εκτεταμένη κλάση λειτουργιών για να χειριστεί εκ των υστέρων τεράστια σύνολα δεδομένων. Οι υποσχόμενες στρατηγικές περιλαμβάνουν την υιοθέτηση της μεταφοράς γνώσης, την απόσταση γνώσης, τη χρήση ποικίλων μοντέλων επί της συσκευής με κοινές νευρικές παραμέτρους ή την αναζήτηση νευρωνικής αρχιτεκτονικής.
- **Περιορισμοί υπολογισμού:** Οι υπολογιστικοί περιορισμοί υπαγορεύουν ότι οι συσκευές μπορούν να πραγματοποιήσουν μόνο έναν περιορισμένο αριθμό ενημερώσεων. Αυτό εγείρει την πρόκληση των παρατεταμένων χρόνων εκπαίδευσης FL, γεγονός που καθιστά αναγκαία την ανάπτυξη ενός αποτελεσματικού επιλύτη βελτιστοποίησης στην πλευρά του πελάτη και ενός αποτελεσματικού μηχανισμού συγκέντρωσης στην πλευρά του διακομιστή. Επιπλέον, οι συσκευές χαμηλότερης ενέργειας μπορεί να είναι σπάνια ενεργές, με αποτέλεσμα χαμηλό ρυθμό δειγματοληψίας στον πελάτη και ενδεχομένως σημαντικές διακυμάνσεις στο συγκεντρωτικό μοντέλο.

- **Ασφάλεια και προστασία:** Η ασφάλεια και η προστασία της ιδιωτικής ζωής αποτελούν επίσης κρίσιμες ανησυχίες. Το FL, το οποίο εξαλείφει την ανάγκη μετάδοσης ακατέργαστων δεδομένων πελατών, όπως εικόνες και ήχος, είναι εγγενώς κατάλληλο για την ενίσχυση του απορρήτου των δεδομένων. Παρ' όλα αυτά, η έρευνα έχει δείξει ότι οι αναγνωρίσιμες πληροφορίες των πελατών μπορούν ακόμη να εξαχθούν από τις κλίσεις (gradients). Αυτή η διαπίστωση οδήγησε στην ανάπτυξη τεχνικών διατήρησης της ιδιωτικότητας, όπως η διαφορικά ιδιωτική SGD. Επιπλέον, μια συσκευή που αποθηκεύει δεδομένα τοπικά γίνεται ελκυστικός στόχος για χάκερ, γεγονός που καθιστά αναγκαία τη λήψη προληπτικών μέτρων για τον μετριασμό αυτών των κινδύνων. Μια προσέγγιση είναι η διατήρηση μόνο περιορισμένου όγκου δεδομένων σε μια συσκευή ανά πάσα στιγμή, η οποία εισάγει πρόσθετες προκλήσεις στην εκπαίδευση FL. Μια συναφής αλλά ξεχωριστή ανησυχία για την προστασία της ιδιωτικής ζωής προκύπτει όταν αναπτύσσεται ένα ιδιόκτητο μοντέλο API στον ανοιχτό κόσμο. Σε τέτοιες περιπτώσεις, πρέπει να εξετάζονται προσεκτικά οι συμβιβασμοί μεταξύ της χρησιμότητας και της ιδιωτικότητας του μοντέλου. Τέλος, στο πεδίο των FL έχουν εμφανιστεί πολυάριθμοι αλγόριθμοι που επικεντρώνονται στην ασφαλή συνάθροιση. Αυτοί οι αλγόριθμοι, οι οποίοι βασίζονται στην αρχή της εκμάθησης συγκεντρωτικών παραμέτρων σε ομάδες συσκευών αντί για μεμονωμένες συσκευές, αποσκοπούν στην ενίσχυση της ανθεκτικότητας του συστήματος έναντι αντιπάλων επιθέσεων.

Συμπερασματικά, καθώς τα συστήματα FL βασίζονται σε πλήθος συμμετεχόντων, μπορεί να είναι ευάλωτα σε απρόβλεπτες προκλήσεις και επιθέσεις που αποκλίνουν από τον προβλεπόμενο σχεδιασμό. Είναι υψίστης σημασίας για τους σχεδιαστές FL να κατανοήσουν τις αντισταθμίσεις μεταξύ της ακρίβειας πρόβλεψης και της ανθεκτικότητας απέναντι σε διάφορες απειλές και να αξιολογήσουν την πρακτική ευπάθεια ενός συστήματος FL (Ding, 2022).

5 Μεθοδολογία και πρακτικό πλαίσιο

Παρακάτω παρουσιάζεται η μεθοδολογία που χρησιμοποιήθηκε για τη συλλογή και την προεπεξεργασία των δεδομένων, καθώς και τα μοντέλα και οι τεχνικές που εφαρμόστηκαν. Η εστίαση έγκειται στην εφαρμογή αισθητήρων στάθμευσης, στρατηγικά τοποθετημένων κατά μήκος τριών κεντρικών δρόμων στην περιοχή της Καλαμαριάς, στη Θεσσαλονίκη, και στη συνέχεια η σωστή αξιοποίηση αυτών των δεδομένων. Ο πρωταρχικός στόχος είναι η μελέτη των δυναμικών καταστάσεων των χώρων στάθμευσης, αποκρυπτογραφώντας τη ροή της πληρότητάς τους.

5.1 Συλλογή και προεπεξεργασία δεδομένων

Για την κατανόηση της δυναμικής της στάθμευσης κατά μήκος των αστικών οδών, υιοθετήθηκε μια μεθοδική προσέγγιση, με τη χρήση αισθητήρων στάθμευσης τοποθετημένων σε τρεις κύριες οδούς στην περιοχή της Καλαμαριάς. Πρόκειται για τις οδούς Πασαλίδη, Χηλής και Μεταμορφώσεως. Ο στόχος είναι να διακρίνουμε τις αλλαγές στην πληρότητα των χώρων στάθμευσης με την πάροδο του χρόνου.

Το σύνολο δεδομένων που λαμβάνεται από αυτούς τους αισθητήρες είναι δομημένο έτσι ώστε να περιλαμβάνει βασικές παραμέτρους:

- Εσωτερικό αναγνωριστικό
- Όνομα του αισθητήρα
- Χρονοσφραγίδα (Timestamp) των αλλαγών κατάστασης
- Νέα κατάσταση της θέσης στάθμευσης: κατειλημμένη ή ελεύθερη

5.1.1 Προεπεξεργασία δεδομένων – 1^ο Στάδιο

Το αρχικό στάδιο της προεπεξεργασίας των δεδομένων είναι ένα κρίσιμο σημείο όπου τα ακατέργαστα δεδομένα υφίστανται έναν μετασχηματισμό για να διευκολυνθεί η ουσιαστική ανάλυση. Η πρωταρχική εστίαση είναι η μετατροπή του timestamp σε μια ολοκληρωμένη μορφή ημερομηνίας, η οποία περιλαμβάνει πληροφορίες για την ημέρα, τον μήνα, το έτος, την ώρα, τα λεπτά και την ημέρα της εβδομάδας.

Ο βρόχος επεξεργασίας, κεντρικό στοιχείο αυτού του σταδίου, εξελίσσεται ως εξής:

- **for each sensor:** Η επανάληψη αρχίζει εξασφαλίζοντας την ενδεδειγμένη εξέταση των δεδομένων από κάθε μεμονωμένο αισθητήρα.
- **for a specific period (e.g., year):** Οριοθετείται το χρονικό πεδίο εφαρμογής, επιτρέποντας την προσαρμογή της ανάλυσης σε μια προκαθορισμένη περίοδο, όπως ένα έτος.
- **for every month:** Τα δεδομένα τμηματοποιούνται περαιτέρω σε μηνιαία διαστήματα, επιτρέποντας μια πιο λεπτομερή διερεύνηση των χρονικών προτύπων.
- **for every day:** Στη συνέχεια, η ανάλυση αναλύεται σε ημερήσιο επίπεδο, καταγράφοντας τις διακυμάνσεις και τις τάσεις εντός κάθε ημέρας.
- **for every hour:** Εξετάζεται η ωριαία προοπτική, ρίχνοντας φως στη δυναμική της στάθμευσης κατά τη διάρκεια της ημέρας.
- **for every 10 or 15 minutes (timeslot):** Η χρονική ανάλυση βελτιώνεται σε διαστήματα των 10 ή 15 λεπτών, διευκολύνοντας τη λεπτομερή εξέταση των καταστάσεων των αισθητήρων εντός διακριτών χρονικών διαστημάτων.

Στη συνέχεια, τα επεξεργασμένα δεδομένα αποθηκεύονται συστηματικά, ενσωματώνοντας την κατάσταση κάθε αισθητήρα σε αυτά τα καθορισμένα χρονικά διαστήματα. Ειδικότερα, υπάρχουν 2 περιπτώσεις:

- **Κατάσταση ενός αισθητήρα πριν από το πρώτο αναφερόμενο συμβάν μιας ημέρας:** Η εξέταση των καταστάσεων των αισθητήρων που προηγούνται του αρχικού αναφερόμενου συμβάντος διασφαλίζει την ολοκληρωμένη κατανόηση της δυναμικής του χώρου στάθμευσης στην αρχή κάθε ημέρας.
- **Κατάσταση ενός αισθητήρα όταν δεν υπάρχουν αναφερόμενα συμβάντα:** Οι περιπτώσεις στις οποίες δεν αναφέρονται συμβάντα για έναν αισθητήρα καταγράφονται επιμελώς, αποτρέποντας τα κενά δεδομένων και διασφαλίζοντας την ακεραιότητα της ανάλυσης.

5.1.2 Προεπεξεργασία δεδομένων – 2^ο Στάδιο

Στο δεύτερο στάδιο της προετοιμασίας των δεδομένων, η έμφαση δίνεται στην οργάνωση των αισθητήρων σε διακριτές ομάδες, απλοποιώντας τη διαδικασία ανάλυσης. Ο στόχος είναι να οριστούν αυτές οι ομάδες ως τμήματα του δρόμου, καθένα από τα οποία είναι εξοπλισμένο με αισθητήρες στάθμευσης. Ακολουθεί ο τρόπος με τον οποίο εξελίσσεται η διαδικασία:

- **Ορισμός ομάδας:** Οι αισθητήρες αντιστοιχίζονται συστηματικά σε συγκεκριμένα οδικά τμήματα ή ομάδες.
- **Ελάχιστοι αισθητήρες ανά ομάδα:** Για τη διασφάλιση της αποτελεσματικότητας κάθε ομάδας, καθορίζεται ένα κριτήριο που απαιτεί τουλάχιστον 8 αισθητήρες σε κάθε ομάδα.
- **Διαφορετικά μεγέθη ομάδων:** Αναγνωρίζοντας τις διαφοροποιήσεις στη διαθεσιμότητα χώρων στάθμευσης, ορισμένες ομάδες μπορούν να περιλαμβάνουν περισσότερους αισθητήρες.

Έχουν προσδιοριστεί συνολικά 9 διαφορετικές ομάδες, οι οποίες κατηγοριοποιούνται σύμφωνα με τους αντίστοιχους δρόμους και τους δείκτες τμημάτων:

- Πασαλίδη: 5 ομάδες (0, 1, 2, 3, 4).
- Χηλής: 2 ομάδες (5, 6).
- Μεταμορφώσεως: 2 ομάδες (7, 8).

Οι ομαδοποιήσεις αυτές λαμβάνουν υπόψη την παρουσία θέσεων στάθμευσης και στις δύο πλευρές του δρόμου, προσφέροντας πληροφορίες για τη διαφοροποιημένη δυναμική της αμφίπλευρης στάθμευσης.

Ο παρακάτω πίνακας περιέχει τις ομάδες μαζί με τα ονόματα των διαφόρων τμημάτων και τους αισθητήρες:

| Group | Name | Sensors |
|-------|------------|--|
| 0 | Pasalidi_1 | 1,2,3,4,5,6,7,8,9,10,11,12,13,14 |
| 1 | Pasalidi_2 | 15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33 |
| 2 | Pasalidi_3 | 34,35,36,37,38,39,40,41,42,43,44 |

| | | |
|---|-----------------|--|
| 3 | Pasalidi_4 | 45,46,47,48,49,50,148,149,51,52,53,54,55,56,57,58,150 |
| 4 | Pasalidi_5 | 151,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79 |
| 5 | Chilis_1 | 95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119 |
| 6 | Chilis_2 | 146,147,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,120,121,122,123,124,125,126,127,128 |
| 7 | Metamorfoseos_1 | 144,145,142,143,137,138,139,140,141 |
| 8 | Metamorfoseos_2 | 133,134,135,136,131,132,129,130 |

Πίνακας 2: Ομάδες και αισθητήρες

5.1.3 Προεπεξεργασία δεδομένων – 3^ο Στάδιο

Περνώντας στο τρίτο στάδιο της προεπεξεργασίας των δεδομένων, η προσοχή μετατοπίζεται στη διασφάλιση ότι κάθε ομάδα είναι επαρκώς εξοπλισμένη με αισθητήρες στάθμευσης. Ο στόχος είναι να υπολογιστεί ο λόγος διαθεσιμότητας για κάθε ομάδα κατά τη διάρκεια συγκεκριμένων χρονικών περιόδων, μετρώντας την αναλογία των ελεύθερων αισθητήρων προς το σύνολο των αισθητήρων εντός της ομάδας. Ακολουθεί ο τρόπος με τον οποίο εξελίσσεται αυτό το στάδιο:

- **Επαρκείς αισθητήρες σε κάθε ομάδα:** Εξασφάλιση ότι κάθε ομάδα διαθέτει επαρκή αριθμό αισθητήρων στάθμευσης για ουσιαστική ανάλυση.
- **Υπολογισμός της αναλογίας διαθεσιμότητας:** Για κάθε timeslot, υπολογίζεται ο λόγος διαθεσιμότητας διαιρώντας τον αριθμό των ελεύθερων αισθητήρων με τον συνολικό αριθμό αισθητήρων στην ομάδα

$$\text{Διαθεσιμότητα} = \frac{\text{Αριθμός ελεύθερων αισθητήρων}}{\text{Συνολικός αριθμός αισθητήρων}}$$

- **Ορισμός επιπέδων φορτίου:** Με βάση τον υπολογισμένο λόγο, κατηγοριοποιούμε τον φόρτο σε μια ομάδα σε τρία επίπεδα:
 - “High”: Εάν λόγος διαθεσιμότητας < 0,20.
 - “Medium”: 0.20 <= λόγος διαθεσιμότητας < 0,40.
 - “Low”: Εάν λόγος διαθεσιμότητας >= 0,40.
- **Αποθήκευση πληροφοριών φορτίου:** Το επίπεδο φόρτου κάθε ομάδας αποθηκεύεται τόσο για τα τρέχοντα όσο και για τα προηγούμενα timeslots,

επιτρέποντας τη διαχρονική κατανόηση των αλλαγών στη δυναμική της στάθμευσης.

5.2 Περιβάλλοντα λογισμικού μηχανικής μάθησης

Ο πρωταρχικός στόχος είναι η πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης για μελλοντικές χρονικές περιόδους, ένα έργο που αποδεικνύεται δύσκολο σε επίπεδο μεμονωμένων αισθητήρων. Για να αντιμετωπιστεί αυτή η πολυπλοκότητα, έχουν επινοηθεί δύο λύσεις:

- **Ομαδοποίηση αισθητήρων:** Οι αισθητήρες οργανώνονται συστηματικά σε ομάδες, επιτρέποντας μια πιο ολοκληρωμένη ανάλυση σε συλλογικό επίπεδο.
- **Πρόβλεψη φορτίου:** Η εστίαση μετατοπίζεται στην πρόβλεψη του φορτίου (κατηγοριοποιημένο ως χαμηλό, μεσαίο ή υψηλό) για κάθε ομάδα αισθητήρων. Το σύνολο δεδομένων ενσωματώνει ήδη την κατάσταση φορτίου για κάθε ομάδα κατά τη διάρκεια διαφόρων timeslots.

Αυτή η εργασία πρόβλεψης διαμορφώνεται ως πρόβλημα ταξινόμησης πολλαπλών κατηγοριών, ευθυγραμμιζόμενη καλά με τις μεθοδολογίες μηχανικής μάθησης. Αξιοποιώντας το υπάρχον σύνολο δεδομένων, το μοντέλο στοχεύει να διακρίνει μοτίβα και σχέσεις μέσα στα ομαδοποιημένα δεδομένα αισθητήρων για να προβλέψει με ακρίβεια τις καταστάσεις φορτίου για μελλοντικά timeslots.

Αυτή η προσέγγιση όχι μόνο ξεπερνά την πρόκληση της πρόβλεψης σε επίπεδο μεμονωμένου αισθητήρα, αλλά θέτει επίσης τις βάσεις για έναν πιο ισχυρό και ακριβή μηχανισμό πρόβλεψης, συμβάλλοντας στην κατανόηση της δυναμικής των χώρων στάθμευσης.

Στο πεδίο της μηχανικής μάθησης, ένα περιβάλλον Python χρησιμεύει ως βάση τόσο για την επεξεργασία δεδομένων όσο και για τη δημιουργία μοντέλων πρόβλεψης. Για την κατασκευή μοντέλων για τα δεδομένα στάθμευσης χρησιμοποιήθηκαν δύο εργαλεία:

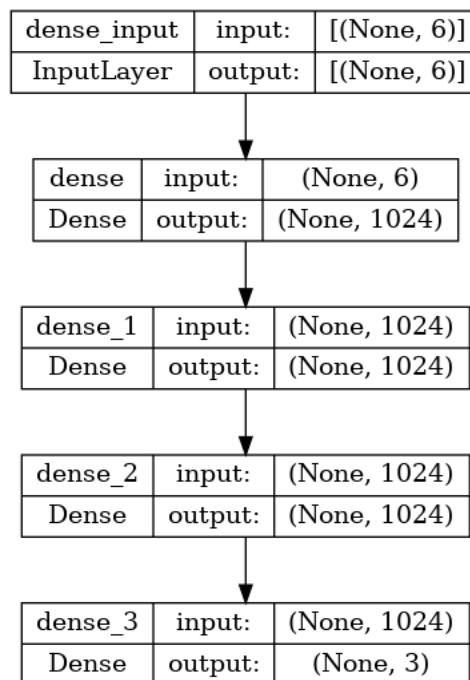
- **Scikit-Learn:** Προσφέρει απλά και αποτελεσματικά εργαλεία αφιερωμένα στην ανάλυση δεδομένων πρόβλεψης. Ο φιλικός προς τον χρήστη χαρακτήρας του διευκολύνει την κατασκευή μοντέλων πρόβλεψης με ευκολία και ακρίβεια.
- **Keras:** Λειτουργώντας ως διεπαφή προγραμματισμού εφαρμογών υψηλού επιπέδου πάνω στην πλατφόρμα TensorFlow, το Keras είναι προσαρμοσμένο για την επίλυση προβλημάτων μηχανικής μάθησης. Δίνει ιδιαίτερη έμφαση στις σύγχρονες μεθοδολογίες βαθιάς μάθησης. Η ενσωμάτωση με το TensorFlow ενισχύει τις δυνατότητές του, παρέχοντας ένα εξελιγμένο πλαίσιο για τη δημιουργία και τη βελτιστοποίηση μοντέλων μηχανικής μάθησης.

5.3 Εκπαίδευση μοντέλων

Στη φάση της εκπαίδευσης μοντέλων, ακολουθείται μια συστηματική προσέγγιση που περιλαμβάνει τα ακόλουθα βήματα:

- **Διαχωρισμός συνόλου δεδομένων:** Το προεπεξεργασμένο σύνολο δεδομένων χωρίζεται σε δύο ξεχωριστά υποσύνολα για σκοπούς εκπαίδευσης και δοκιμής.
- **Κλιμάκωση και κωδικοποίηση χαρακτηριστικών:** Οι έξι αριθμητικές στήλες εισόδου υποβάλλονται σε κλιμάκωση και η στήλη εξόδου κωδικοποιείται για την προετοιμασία των δεδομένων για την αποτελεσματική εκπαίδευση του μοντέλου.
- **Κατασκευή μοντέλου με το Keras:** Χρησιμοποιώντας το πλαίσιο Keras, δημιουργείται ένα μοντέλο Multi-Layer Perceptron (MLP). Η αρχιτεκτονική περιλαμβάνει τρία πυκνά στρώματα, καθένα από τα οποία φιλοξενεί 1024 νευρώνες, προωθώντας μια ισχυρή δομή μάθησης.
- **Εκπαίδευση σε διακομιστή GPU:** Η διαδικασία εκπαίδευσης του μοντέλου εκτελείται σε διακομιστή GPU εξοπλισμένο με NVIDIA GeForce RTX 4090Ti, αξιοποιώντας την παράλληλη επεξεργαστική ισχύ για αποδοτικούς υπολογισμούς.

- **Αξιολόγηση πρόβλεψης:** Η ακρίβεια πρόβλεψης του εκπαιδευμένου μοντέλου αξιολογείται αυστηρά χρησιμοποιώντας το σύνολο δεδομένων επικύρωσης, παρέχοντας πληροφορίες σχετικά με την απόδοσή του.
- **Δημιουργία και αξιολόγηση τμηματοποιημένου μοντέλου:** Τα μοντέλα κατασκευάζονται και αξιολογούνται συστηματικά σε διάφορα επίπεδα και για διαφορετικές παραμέτρους, ανάλογα και με τα αποτελέσματα που προκύπτουν σε κάθε περίπτωση.



Εικόνα 6: Εκπαίδευση μοντέλων

5.4 Ενσωμάτωση AutoML πρακτικών

Στο πλαίσιο διάφορων παραμέτρων, για τη βελτίωση του πολύπλοκου έργου της πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης, διερευνάται επίσης η δύναμη της αυτοματοποιημένης μηχανικής μάθησης (AutoML).

Αναγνωρίζοντας τα πολυάριθμα πλεονεκτήματα της AutoML, συμπεριλαμβανομένης της αυτοματοποιημένης επιλογής μοντέλου, της ρύθμισης υπερπαραμέτρων και της προεπεξεργασίας δεδομένων, η προσέγγιση αυτή δοκιμάστηκε παράλληλα με την επιλογή της χειροκίνητης δημιουργίας ενός μοντέλου. Ο κώδικας διαβάζει και προεπεξεργάζεται το σύνολο δεδομένων στάθμευσης, χρησιμοποιώντας την AutoML για την αυτοματοποίηση των

εργασιών προετοιμασίας δεδομένων και την αποτελεσματική κωδικοποίηση των επιπέδων φόρτωσης για την εκπαίδευση του μοντέλου.

Το AutoKeras χρησιμοποιείται για την αυτόματη διερεύνηση και επιλογή μιας βέλτιστης αρχιτεκτονικής νευρωνικού δικτύου, απλοποιώντας τη διαδικασία επιλογής μοντέλου. Επιπλέον, χειρίζεται τον συντονισμό των υπερπαραμέτρων για τη λεπτομερή ρύθμιση του μοντέλου για τη σύλληψη περίπλοκων μοτίβων στα δεδομένα στάθμευσης. Η υλοποίηση αξιολογεί τόσο το μοντέλο που δημιουργείται από το AutoML όσο και τα μοντέλα που δημιουργούνται χειροκίνητα, παρέχοντας μια ολοκληρωμένη σύγκριση για την εξακρίβωση της πιο αποτελεσματικής προσέγγισης για το συγκεκριμένο έργο της πρόβλεψης. Αυτή η στρατηγική δοκιμών διασφαλίζει την ενδελεχή εξέταση των επιδόσεων του AutoML, διατηρώντας παράλληλα την ευελιξία για τη χειροκίνητη δημιουργία μοντέλων με βάση τα μοναδικά χαρακτηριστικά του συνόλου δεδομένων και του τομέα του προβλήματος.

Η μεθοδολογία αξιοποιεί και το AutoSklearn, ένα εργαλείο αυτοματοποιημένης μηχανικής μάθησης (AutoML), για τη δημιουργία ενός συστήματος πρόβλεψης διαθεσιμότητας θέσεων στάθμευσης. Αρχικά, γίνεται επεξεργασία του συνόλου δεδομένων και κωδικοποιούνται τα επίπεδα φόρτωσης για εκπαίδευση. Στη συνέχεια χρησιμοποιείται το AutoSklearn για την αυτόματη επιλογή και βελτιστοποίηση του καλύτερου μοντέλου μηχανικής μάθησης και των υπερπαραμέτρων εντός συγκεκριμένων χρονικών περιορισμών. Το μοντέλο εκπαιδεύεται και υπολογίζονται μετρικές αξιολόγησης, όπως η ακρίβεια και το confusion matrix. Η ευελιξία του AutoSklearn διευκολύνει τη διαδικασία επιλογής μοντέλου και συντονισμού υπερπαραμέτρων, ενισχύοντας την αποτελεσματικότητα της ανάπτυξης του συστήματος πρόβλεψης χώρων στάθμευσης. Επιπλέον, ο κώδικας περιλαμβάνει διατάξεις για την παραδοσιακή εκπαίδευση μοντέλων με τη χρήση ενός RandomForestClassifier για συγκριτική ανάλυση.

Συνολικά, με την συγκεκριμένη υλοποίηση εξετάζονται τα πλεονεκτήματα του AutoML, μειώνοντας τις χειροκίνητες προσπάθειες, προωθώντας την

αναπαραγωγικότητα και επιταχύνοντας τον κύκλο ανάπτυξης της μηχανικής μάθησης για την πρόβλεψη διαθεσιμότητας χώρων στάθμευσης.

5.5 Διερεύνηση Federated Learning πρακτικών

Το FL, ένα αποκεντρωμένο παράδειγμα μηχανικής μάθησης, επιλέχθηκε ως μεθοδολογία για την αξιολόγηση της αποτελεσματικότητάς του στο πλαίσιο της πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης. Η προσέγγιση αυτή παρακινήθηκε από την ανάγκη σεβασμού των περιορισμών απορρήτου που σχετίζονται με τα δεδομένα θέσης και προώθησης της συνεργατικής ανάπτυξης μοντέλων σε πολλαπλές ομάδες.

Το πείραμα ξεκίνησε με τον διαχωρισμό του συνόλου δεδομένων σε διακριτά κομμάτια, καθένα από τα οποία αντιστοιχούσε σε ένα συγκεκριμένο group. Κάθε group εκτέλεσε ανεξάρτητα το μοντέλο, εκπαιδεύοντάς το αποκλειστικά με τα τοπικά δεδομένα που κατείχε. Στη συνέχεια, τα εκπαιδευμένα βάρη του μοντέλου διαβιβάστηκαν με ασφάλεια σε έναν κεντρικό διακομιστή.

Στην πλευρά του διακομιστή, πραγματοποιήθηκε μια διαδικασία συνάθροισης, κατά την οποία συνδυάστηκαν τα βάρη του μοντέλου από όλα τα groups. Στη συνέχεια, ο διακομιστής υπολόγισε το μέσο όρο αυτών των συγκεντρωτικών βαρών. Αυτό το μέσο σύνολο βαρών μεταδόθηκε στη συνέχεια σε κάθε group.

Κάθε group, αφού έλαβε τα μέσα βάρη, τα χρησιμοποίησε ως τα αρχικά βάρη για τα αντίστοιχα μοντέλα του. Ακολούθησε περαιτέρω εκπαίδευση, βελτιώνοντας τα μοντέλα συνεργατικά. Αυτή η επαναληπτική διαδικασία εκπαίδευσης μοντέλων, συνάθροισης και επαναδιανομής συνεχίστηκε για αρκετά epochs, επιτρέποντας στο συνολικό μοντέλο να ενσωματώσει σταδιακά τις γνώσεις από κάθε ομάδα.

Το πείραμα FL βοήθησε έτσι μια συνεργατική και διατηρώντας την ιδιωτικότητα προσέγγιση, αξιοποιώντας τα δυνατά σημεία των μεμονωμένων groups, ενώ παράλληλα βελτίωνε συλλογικά τις προγνωστικές ικανότητες του μοντέλου. Αυτή η μεθοδολογία αποσκοπούσε στη διερεύνηση των δυνατοτήτων

του FL για την αντιμετώπιση των προκλήσεων που θέτουν τα αποκεντρωμένα σύνολα δεδομένων στο πεδίο της πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης.

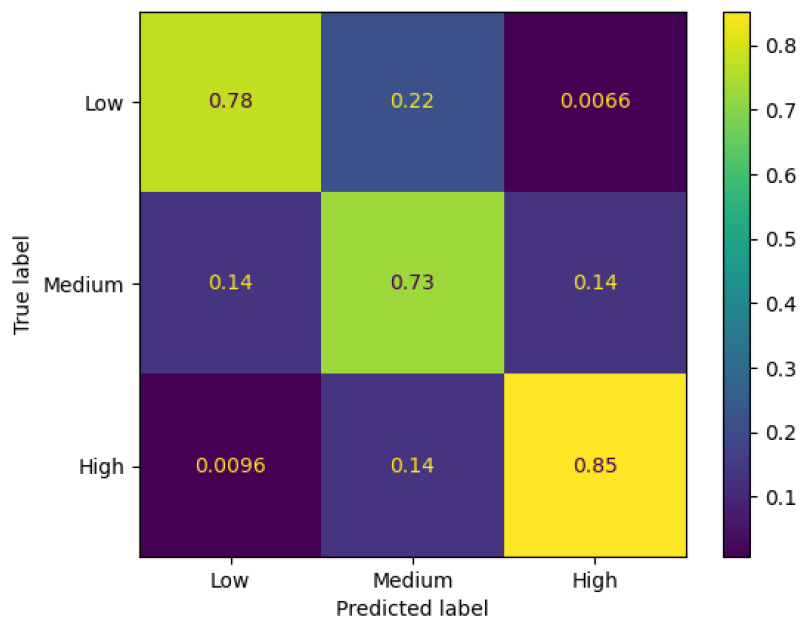
6 Αποτελέσματα και συζήτηση

Παρακάτω παρουσιάζεται ολοκληρωμένη η ανάλυση και ο τρόπος με τον οποίο εξελίχθηκε με βάση τα αποτελέσματα.

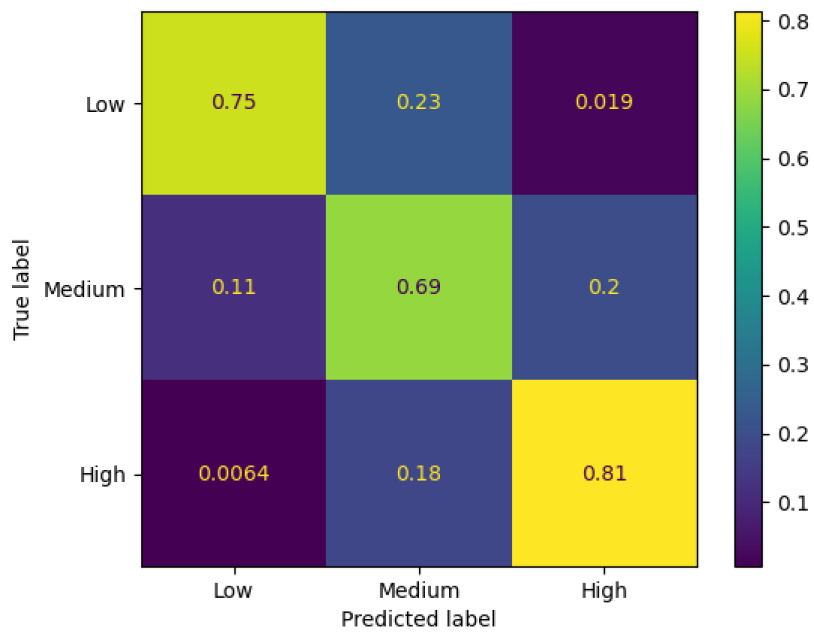
6.1 1^ο Στάδιο Αποτελεσμάτων

Στην αρχική φάση του πειραματισμού, αξιολογήθηκε η απόδοση των μοντέλων Keras MLP και Scikit-Learn για να εκτιμηθούν οι δυνατότητες πρόβλεψής τους για τη διαθεσιμότητα θέσεων στάθμευσης. Στη συνέχεια, χρησιμοποιήθηκαν διάφοροι ταξινομητές, συμπεριλαμβανομένων των Adaboost, K-Nearest Neighbors, Decision Tree, Random Forest και XGBoost, για την επέκταση της αξιολόγησης. Σε αυτή τη συνολική ανάλυση, διερευνήθηκε η επίδραση βασικών παραμέτρων, όπως ο αριθμός των επιπέδων (layers) και οι ρυθμοί μάθησης (Learning rates: 0,001, 0,01 και 0,1). Ο στόχος πίσω από αυτή την προσέγγιση ήταν να εντοπιστεί το βέλτιστο μοντέλο που υπερείχε στην πρόβλεψη της διαθεσιμότητας θέσεων στάθμευσης. Μέσω της ανάλυσης, κατέστη προφανές ότι το μοντέλο με τις καλύτερες επιδόσεις είναι το Keras.

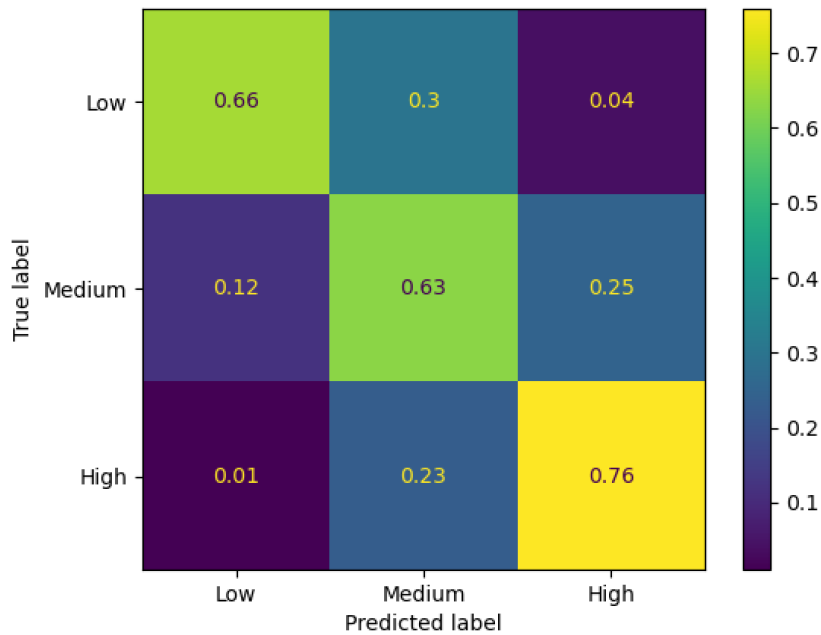
Παρακάτω παρουσιάζονται κάποια αρχικά αποτελέσματα πρόβλεψης με βάση ένα group (Group 0), έναν δρόμο (Πασαλίδη) και όλα τα groups.



Εικόνα 7: Αποτελέσματα πρόβλεψης - Ένα group (Group 0)



Εικόνα 8: Αποτελέσματα πρόβλεψης - Ένας δρόμος (Πασαλίδη)



Εικόνα 9: Αποτελέσματα πρόβλεψης – Όλα τα Groups

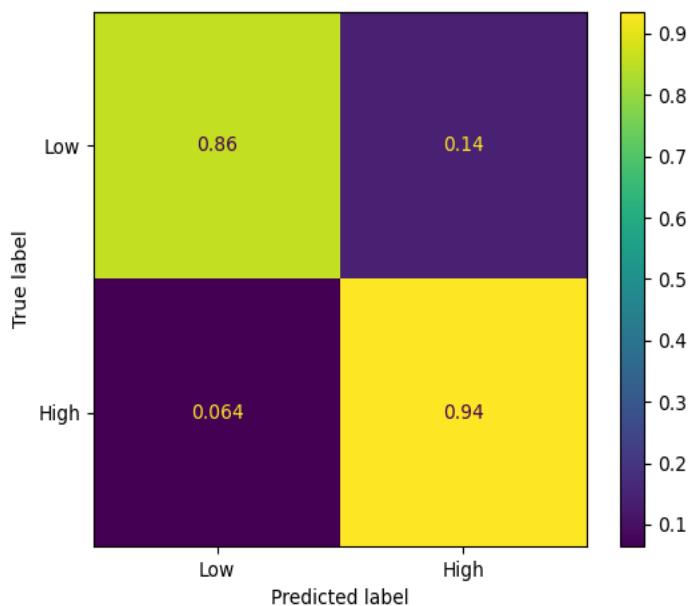
Ο κατακόρυφος άξονας των γραφημάτων αντιπροσωπεύει τις πραγματικές τιμές, ενώ ο οριζόντιος άξονας απεικονίζει τις προβλέψεις του μοντέλου. Για παράδειγμα, στην πρώτη σειρά στην Εικόνα 7, παρατηρείται ότι το μοντέλο πρόβλεψε Low και αυτό ήταν και το πραγματικό δεδομένο, δηλαδή έκανε σωστή πρόβλεψη, και αυτό συνέβη με πιθανότητα 78%, ποσοστό το οποίο αποτελεί το accuracy για κάθε περίπτωση. Το συνολικό accuracy υπολογίζεται ως ο μέσος όρος των τιμών κατά μήκος της διαγώνιου.

Κατά την προσεκτικότερη εξέταση του τελικού γραφήματος, γίνεται εμφανές ότι η απόδοση πρόβλεψης δεν είναι τόσο ισχυρή όσο παρατηρήθηκε στα δύο προηγούμενα γραφήματα. Η διαφορά αυτή μπορεί να αποδοθεί στο γεγονός ότι το μοντέλο εκπαιδεύτηκε σε όλους του δρόμους και για όλες τις ομάδες. Οι δρόμοι, ωστόσο, παρουσιάζουν μοναδικά χαρακτηριστικά και πολυάριθμοι παράγοντες επηρεάζουν δυναμικά τη διαθεσιμότητα στάθμευσης. Η ανομοιογένεια των χαρακτηριστικών σε επίπεδο δρόμου μπορεί να συμβάλει στις διακυμάνσεις της ακρίβειας του μοντέλου.

6.2 2^ο Στάδιο Αποτελεσμάτων

Μια πρόσθετη παρατήρηση στα πρώτα αποτελέσματα υπογραμμίζει την αδύναμη απόδοση του μοντέλου στο μεσαίο επίπεδο. Στις ακραίες τιμές, το μοντέλο έχει μεγαλύτερη προβλεπτική ικανότητα από ότι στις μεσαίες καταστάσεις. Κατά συνέπεια, αποφασίστηκε να επαναπροσδιοριστούν τα επίπεδα, ενοποιώντας τα σε δύο κατηγορίες: Low και High. Ο στόχος πίσω από αυτή την τροποποίηση ήταν να βελτιωθεί η ακρίβεια πρόβλεψης βελτιώνοντας την ικανότητα του μοντέλου να διακρίνει μεταξύ διαφορετικών καταστάσεων διαθεσιμότητας χώρων στάθμευσης.

Μια επακόλουθη ανάλυση περιλάμβανε τη δοκιμή διαφορετικών τιμών κατωφλίου για τον προσδιορισμό των βέλτιστων ορίων που αποδίδουν τα πιο ευνοϊκά αποτελέσματα. Επίσης, ακόμη μια παράμετρος που προστέθηκε είναι το μοντέλο να λαμβάνει πλέον υπόψη και την προηγούμενη κατάσταση της θέσης στην οποία βρίσκεται ο αισθητήρας (ελεύθερη ή κατειλημμένη). Ένα παράδειγμα αποτελέσματος πρόβλεψης με δύο επίπεδα και λαμβάνοντας υπόψη και την προηγούμενη κατάσταση παρουσιάζεται παρακάτω.



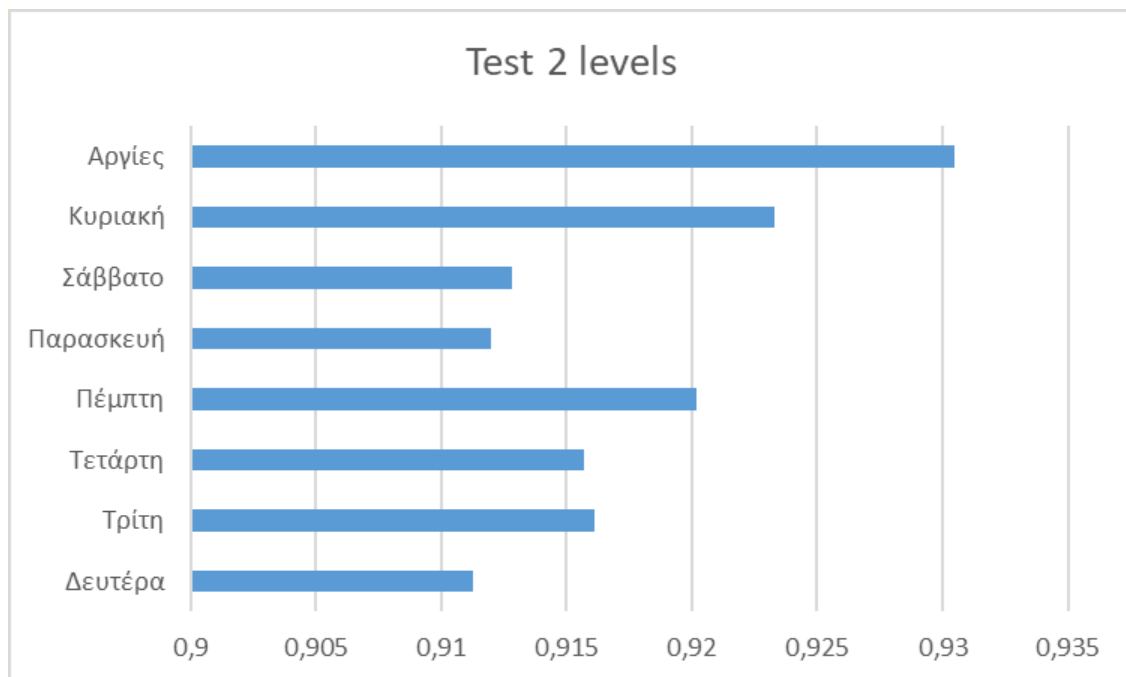
Εικόνα 10: Αποτελέσματα πρόβλεψης – 2 Επίπεδα

Παρατηρείται ότι με δύο επίπεδα το μοντέλο προσφέρει πολύ καλύτερα αποτελέσματα, με πιο ακριβείς προβλέψεις.

6.3 3^ο Στάδιο Αποτελεσμάτων

Με το μοντέλο Keras, τα δύο καθορισμένα επίπεδα και λαμβάνοντας υπόψη και την προηγούμενη κατάσταση, η προσοχή στράφηκε σε υπό περιπτώσεις ώστε να παρατηρηθεί η συμπεριφορά του μοντέλου και να κατανοηθεί για ποιες παραμέτρους το μοντέλο μπορεί να μην έχει τόσο καλή απόδοση. Οι περιπτώσεις εξετάστηκαν μεμονωμένα και έγιναν προσπάθειες να κατανοηθούν οι λόγοι που κρύβονται πίσω από τις αδυναμίες του μοντέλου.

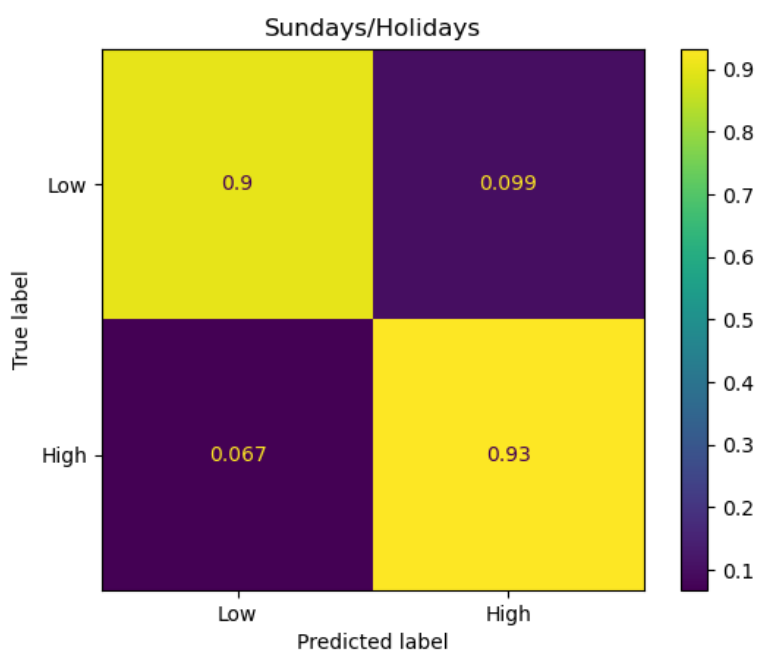
Ο αρχικός έλεγχος περιλάμβανε την αξιολόγηση της απόδοσης του μοντέλου για κάθε ημέρα της εβδομάδας ξεχωριστά. Στόχος ήταν να εντοπιστούν τυχόν συγκεκριμένα πρότυπα ή παράγοντες σε ορισμένες ημέρες που θα μπορούσαν να επηρεάσουν την ακρίβεια του μοντέλου.



Εικόνα 11: Αποτελέσματα πρόβλεψης - Όλες οι μέρες

Παρατηρείται ότι για τις Κυριακές, καθώς και για τις ημέρες που είναι αργίες, το μοντέλο δίνει τα καλύτερα αποτελέσματα.

Η επακόλουθη δοκιμή περιλάμβανε την εξέταση των μοντέλων σε ιδιαίτερες περιστάσεις, όπως οι Κυριακές, που χαρακτηρίζονται από ενδεχομένως διαφορετικά πρότυπα κυκλοφορίας (λιγότερη κίνηση) και οι αργίες. Σε αυτό το σενάριο, το μοντέλο εκπαιδεύτηκε συμβατικά σε ολόκληρο το σύνολο των δεδομένων, αλλά οι αξιολογήσεις πραγματοποιήθηκαν ειδικά σε καθορισμένες ημέρες. Η προσέγγιση αυτή αποσκοπούσε στην ενδελεχή εξέταση της συμπεριφοράς του μοντέλου, αξιολογώντας την προσαρμοστικότητα του στην κανονική εκπαίδευση και την απόδοσή του σε συνθήκες διακριτών ημερών. Η πρόθεση ήταν να διακριθεί αν το μοντέλο μάθαινε επιτυχώς από το σύνολο των δεδομένων ή αντιμετώπιζε περιορισμούς στην ακριβή πρόβλεψη της διαθεσιμότητας θέσεων στάθμευσης σε συγκεκριμένες ημέρες.



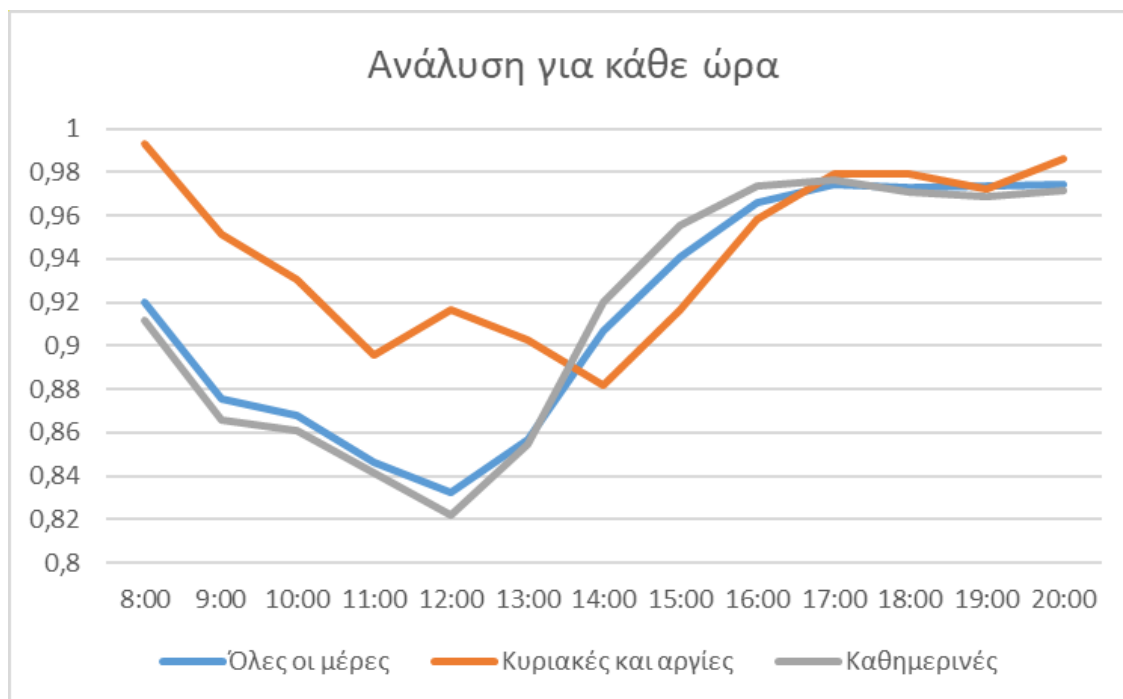
Εικόνα 12: Αποτελέσματα πρόβλεψης - Κυριακές & αργίες

Και πάλι είναι εμφανές ότι τα αποτελέσματα για Κυριακές και αργίες είναι πολύ καλά και με πολύ υψηλά ποσοστά.

6.4 4^ο Στάδιο Αποτελεσμάτων

Ένα άλλο τεστ περιλάμβανε την εξέταση των μοντέλων για κάθε ώρα της ημέρας, από τις 8:00 έως τις 20:00. Η ανάλυση αυτή αποσκοπούσε στην αξιολόγηση της απόδοσης και της ακρίβειας του μοντέλου σε διαφορετικά ωριαία

διαστήματα εντός μιας ημέρας. Με τη διενέργεια αυτής της δοκιμής επιδιώχθηκε να κατανοηθεί πόσο καλά το μοντέλο προσαρμόζεται στις διακυμάνσεις της διαθεσιμότητας χώρων στάθμευσης σε όλο το καθορισμένο χρονικό διάστημα και να εντοπιστούν τυχόν χρονικά πρότυπα ή προκλήσεις που αντιμετωπίζει το μοντέλο κατά τη διάρκεια διαφορετικών ωρών της ημέρας.



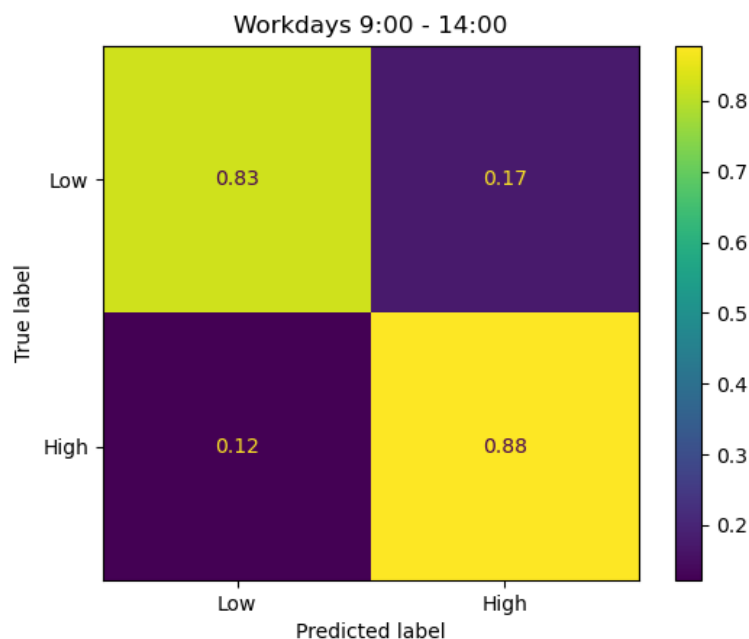
Εικόνα 13: Αποτελέσματα πρόβλεψης - Ανάλυση για κάθε ώρα

Στο παραπάνω διάγραμμα παρουσιάζονται τα αποτελέσματα πρόβλεψης για τις ώρες 8:00 με 20:00 για όλες τις μέρες, για τις Κυριακές και αργίες, καθώς και για τις καθημερινές. Σε όλες τις περιπτώσεις παρατηρείται αρχικά ότι υπάρχει ένα διάστημα ανάμεσα στις ώρες 9:00 και 14:00 περίπου, όπου μειώνεται η ακρίβεια πρόβλεψης του μοντέλου, ενώ και πάλι παρατηρείται ότι το μοντέλο δίνει τα καλύτερα αποτελέσματα για Κυριακές και αργίες.

6.5 5^ο Στάδιο Αποτελεσμάτων

Στα παραπάνω αποτελέσματα παρατηρήθηκαν σταθερά μη βέλτιστα αποτελέσματα για τις καθημερινές, ιδίως κατά τις ώρες από 9:00 έως 14:00. Ως απάντηση σε αυτό το μοτίβο, το επόμενο και καταληκτικό βήμα περιλάμβανε μια στοχευμένη εξέταση που επικεντρώθηκε ειδικά στις καθημερινές κατά τις ώρες 9:00 έως 14:00. Αυτή η συγκεκριμένη δοκιμή είχε ως στόχο να εμβαθύνει στις

προκλήσεις που αντιμετώπιζε το μοντέλο κατά τη διάρκεια αυτών των συγκεκριμένων χρονικών πλαισίων τις καθημερινές και να εξακριβώσει εάν υπήρχαν μοναδικοί παράγοντες που συνέβαλαν στους παρατηρούμενους περιορισμούς στην ακρίβεια πρόβλεψης.

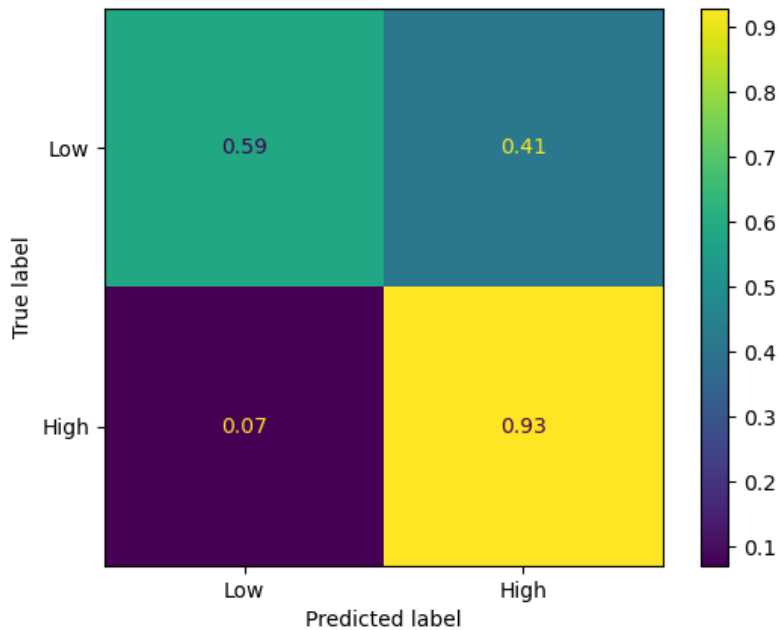


Εικόνα 14: Αποτελέσματα πρόβλεψης - Ανάλυση για συγκεκριμένες ώρες και μέρες

Στη συγκεκριμένη περίπτωση επιβεβαιώνεται ότι στις συγκεκριμένες μέρες και ώρες, το μοντέλο δίνει ναι μεν αρκετά καλά αποτελέσματα, αλλά όχι εξίσου καλά με τις υπόλοιπες μέρες και ώρες.

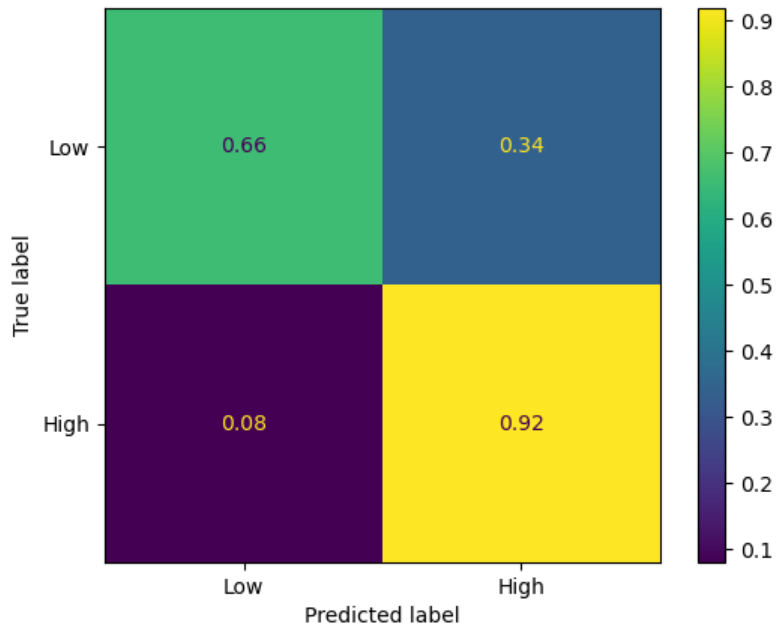
6.6 6^ο Στάδιο Αποτελεσμάτων

Στην περίπτωση των AutoML πρακτικών, ύστερα από τη δοκιμή διάφορων παραμέτρων, όπως επιπέδων, layers, αριθμό δοκιμών, τα αποτελέσματα του εφαρμοζόμενου μοντέλου για την πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης δείχνουν μια αξιόπαινη ακρίβεια περίπου 84,5%, στην περίπτωση που δεν λαμβάνονται υπόψη οι προηγούμενες καταστάσεις. Αυτή η μετρική ακρίβειας αντικατοπτρίζει το ποσοστό των σωστά προβλεπόμενων περιπτώσεων επί του συνολικού συνόλου δεδομένων. Στο πλαίσιο της διαθεσιμότητας χώρων στάθμευσης, όπου ο στόχος είναι ο ακριβής προσδιορισμός της κατάστασης των θέσεων στάθμευσης, η επίτευξη ακρίβειας 84,5% αποτελεί ένα καλό αποτέλεσμα.



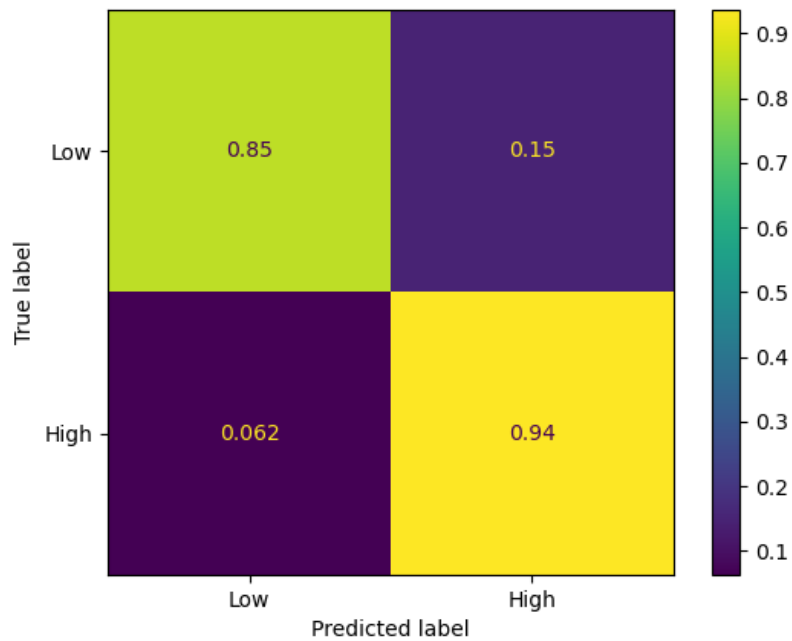
Εικόνα 15: Αποτελέσματα πρόβλεψης Autokeras

Παρά την επίτευξη αυτής της ακρίβειας είναι σημαντικό να σημειωθεί ότι τα αποτελέσματα που προκύπτουν από την προσέγγιση AutoML, χρησιμοποιώντας τη βιβλιοθήκη Autokeras, δεν υπερτερούν σημαντικά σε σχέση με εκείνα του μοντέλου που δημιουργήθηκε χειροκίνητα. Στην πραγματικότητα, υπάρχουν περιπτώσεις όπου το χειροκίνητα δημιουργημένο μοντέλο παρουσιάζει καλύτερες επιδόσεις. Το φαινόμενο αυτό θα μπορούσε να αποδοθεί σε διάφορους παράγοντες. Παρακάτω παρουσιάζεται το confusion matrix του χειροκίνητα δημιουργημένου μοντέλου, με τις ίδιες παραμέτρους με το αυτοματοποιημένο μοντέλο.



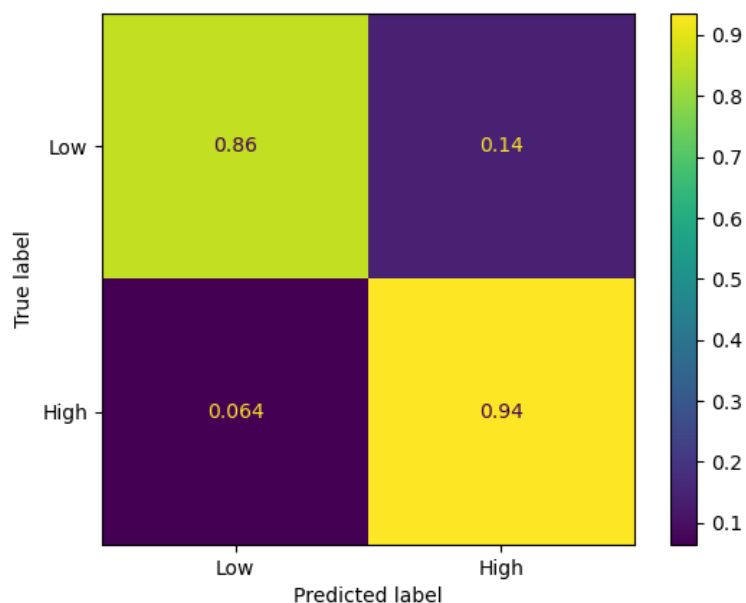
Εικόνα 16: Αποτελέσματα πρόβλεψης χειροκίνητα δημιουργημένου μοντέλου

Στην περίπτωση στην οποία λαμβάνονται υπόψη οι προηγούμενες καταστάσεις, δοκιμάζοντας και πάλι διάφορες παραμέτρους ώστε να βρεθούν οι βέλτιστες τιμές, το ποσοστό ακρίβειας είναι σχεδόν 7% πάνω, σε ποσοστό 91,6%. Παρατηρείται ότι και σε αυτήν την περίπτωση των AutoML πρακτικών, τα αποτελέσματα είναι πολύ καλύτερα δίνοντας στο μοντέλο και τις προηγούμενες καταστάσεις των αισθητήρων.



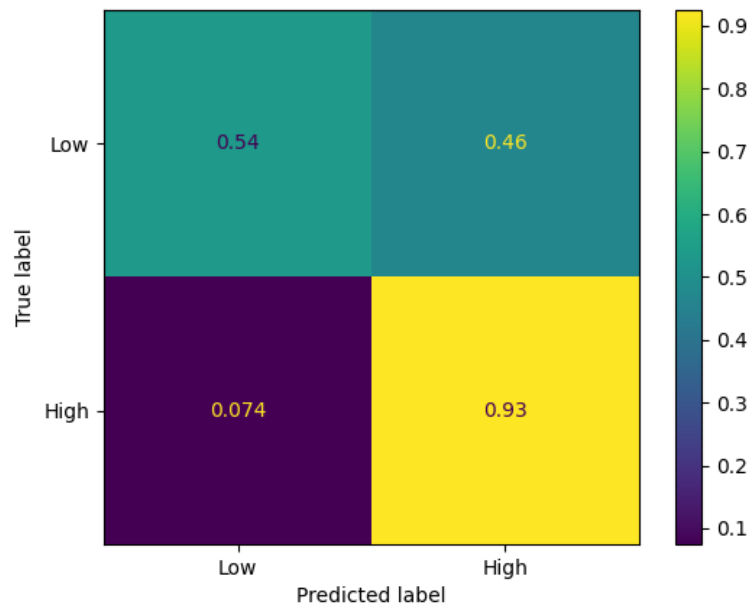
Εικόνα 17: Αποτελέσματα πρόβλεψης Autokeras - Previous State

Αξίζει να σημειωθεί ότι και σε αυτήν την περίπτωση, το χειροκίνητα δημιουργημένο μοντέλο δίνει ελαφρώς καλύτερα αποτελέσματα σε σχέση με το αυτοματοποιημένο, έχοντας πάντα τις ίδιες παραμέτρους. Ωστόσο, οι διαφορές στην τελική ακρίβεια του μοντέλου είναι σχεδόν μηδαμινές. Παρακάτω παρουσιάζεται το confusion matrix του χειροκίνητα δημιουργημένου μοντέλου σε αυτήν την περίπτωση.



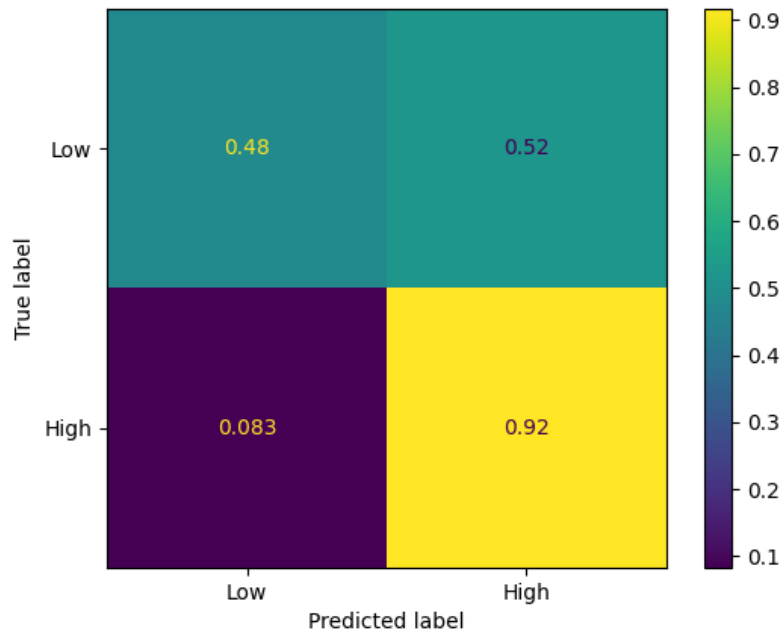
Εικόνα 18: Αποτελέσματα πρόβλεψης χειροκίνητα δημιουργημένου μοντέλου - Previous State

Χρησιμοποιώντας τη βιβλιοθήκη Autoklearn και διεξάγοντας επαναληπτικά πειράματα με ποικίλες διαμορφώσεις παραμέτρων, συμπεριλαμβανομένων των επιπέδων και του αριθμού των δοκιμών, το μοντέλο πρόβλεψης για τη διαθεσιμότητα θέσεων στάθμευσης παρουσιάζει εντυπωσιακή ακρίβεια 82,9%. Αξίζει να σημειωθεί ότι αυτό το επίτευγμα επιτυγχάνεται χωρίς να λαμβάνονται υπόψη οι ιστορικές καταστάσεις των αισθητήρων. Για να υπάρξει μια ολοκληρωμένη αξιολόγηση της αποτελεσματικότητας του Autoklearn, το χειροκίνητα δημιουργημένο μοντέλο υποβάλλεται σε δοκιμές χρησιμοποιώντας τις ίδιες παραμέτρους.



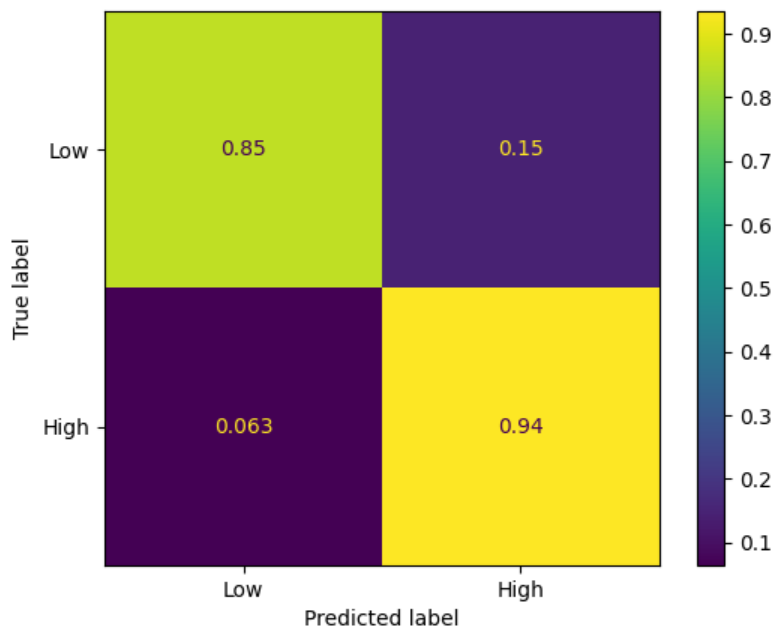
Εικόνα 19: Αποτελέσματα πρόβλεψης Autoklearn

Σε αυτή τη συγκριτική ανάλυση, το χειροκίνητα παραγόμενο μοντέλο, σε αντίθεση με το αντίστοιχο του Autokeras, παρουσιάζει οριακά χαμηλότερη ακρίβεια, η οποία κυμαίνεται γύρω στο 80,9%. Αυτό το ενδιαφέρον αποτέλεσμα υποδηλώνει ότι η αυτοματοποιημένη παραγωγή μοντέλων που διευκολύνεται από το Autoklearn υπερτερεί του χειροκίνητα κατασκευασμένου μοντέλου στο συγκεκριμένο πλαίσιο. Η διαφοροποιημένη διερεύνηση διαφόρων ρυθμίσεων παραμέτρων αναδεικνύει την προσαρμοστικότητα και την αποτελεσματικότητα του Autoklearn, ενισχύοντας τη χρησιμότητά του στη βελτιστοποίηση των μοντέλων πρόβλεψης για τη διαθεσιμότητα χώρων στάθμευσης.



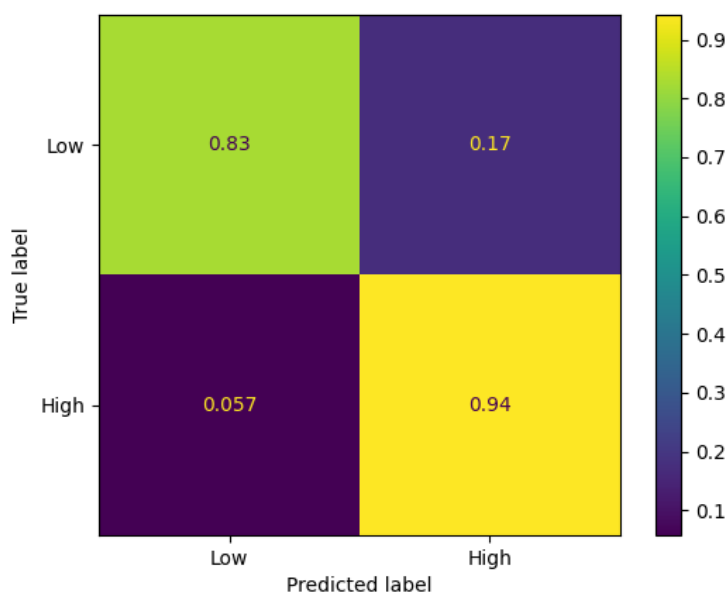
Εικόνα 20: Αποτελέσματα πρόβλεψης χειροκίνητα δημιουργημένου μοντέλου

Ενσωματώνοντας την εξέταση προηγούμενων καταστάσεων και διεξάγοντας διεξοδική διερεύνηση των παραμέτρων για τον εντοπισμό των βέλτιστων τιμών, το ποσοστό ακρίβειας του μοντέλου πρόβλεψης σημειώνει αξιοσημείωτη αύξηση κατά σχεδόν 9% σε ένα εντυπωσιακό 91,6%. Αυτή η αξιοσημείωτη βελτίωση υπογραμμίζει τον καθοριστικό ρόλο των πρακτικών AutoML, ιδίως όταν είναι εφοδιασμένες με τις ιστορικές καταστάσεις των αισθητήρων. Η εμπλουτισμένη επίγνωση του πλαισίου που παρέχεται στο μοντέλο συμβάλλει προφανώς στην ανώτερη απόδοσή του.



Εικόνα 21: Αποτελέσματα πρόβλεψης Autosklearn - Previous State

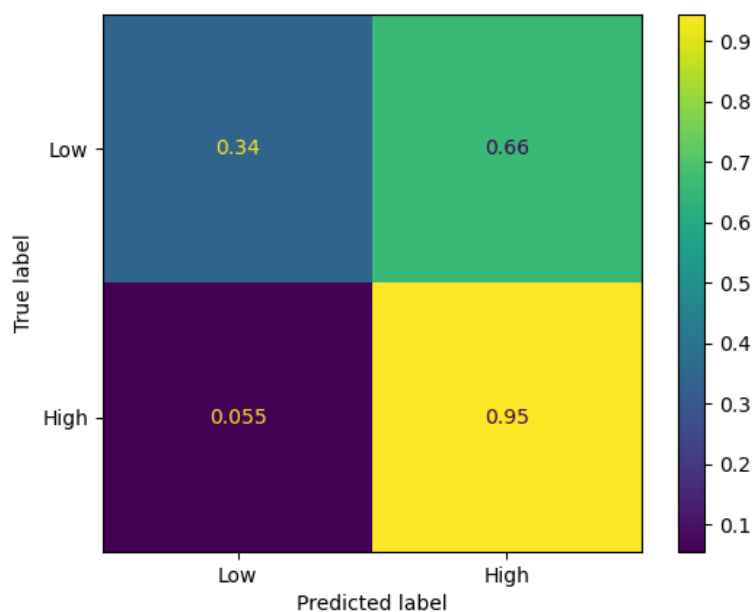
Αξίζει να σημειωθεί ότι, ακόμη και με πανομοιότυπες παραμέτρους, το χειροκίνητα κατασκευασμένο μοντέλο υστερεί ελαφρώς σε σχέση με το αντίστοιχο αυτοματοποιημένο. Παρά την απόκλιση αυτή, οι διαφορές στην τελική ακρίβεια είναι οριακές, πιστοποιώντας την ανθεκτικότητα και των δύο προσεγγίσεων. Ως σχετική προσθήκη, ο πίνακας σύγχυσης για το χειροκίνητα δημιουργημένο μοντέλο παρατίθεται παρακάτω, προσφέροντας ένα λεπτομερές στιγμιότυπο των χαρακτηριστικών επιδόσεών του για ολοκληρωμένη ανάλυση και σύγκριση.



Εικόνα 22: Αποτελέσματα πρόβλεψης χειροκίνητα δημιουργημένου μοντέλου - Previous State

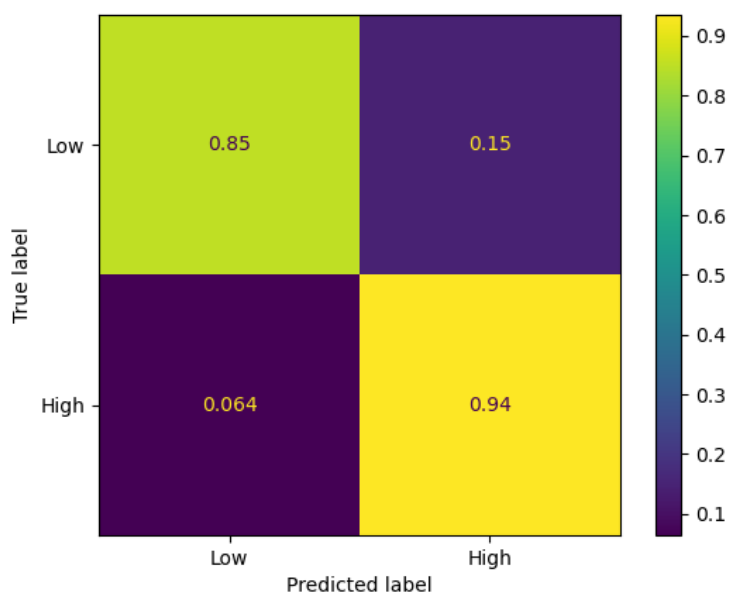
6.7 7^ο Στάδιο Αποτελεσμάτων

Τα λεπτομερή ευρήματα που απορρέουν από την ολοκληρωμένη διερεύνηση των πρακτικών Federated Learning προσφέρουν αρκετές γνώσεις στην απόδοση του προγνωστικού μοντέλου. Στην αρχική φάση, ο πειραματισμός επικεντρώθηκε σε ένα σύνολο δεδομένων χωρίς ιστορικές καταστάσεις αισθητήρων. Αυτό το πλαίσιο απέδωσε μια αξιοσημείωτη ακρίβεια 79,5%, αποτέλεσμα που επιτεύχθηκε μετά από σχολαστική ρύθμιση των παραμέτρων για 5 εποχές και την ενσωμάτωση 2 επιπέδων. Μια πιο προσεκτική εξέταση της επαναληπτικής διαδικασίας εκπαίδευσης αποκαλύπτει μια συνεπή άνοδο της ακρίβειας του μοντέλου σε κάθε εποχή, με τελική κατάληξη τη βέλτιστη απόδοση. Ενδιαφέρον παρουσιάζει μια παρατήρηση σχετικά με τις προβλέψεις χαμηλού επιπέδου, η οποία υποδεικνύει περιθώρια βελτίωσης σε αυτή τη συγκεκριμένη κατηγορία, δίνοντας κίνητρο για περαιτέρω διερεύνηση και πιθανή βελτίωση του μοντέλου.



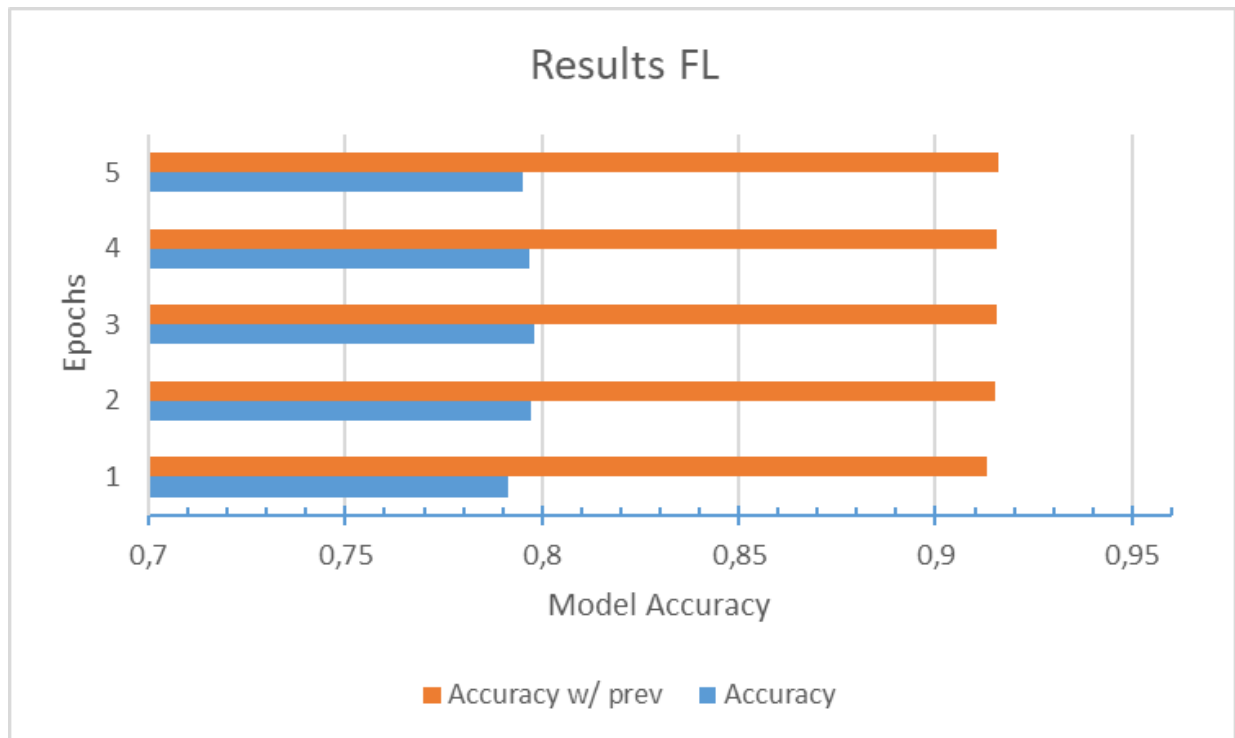
Εικόνα 23: Αποτελέσματα Federated Learning

Στην περίπτωση αλλαγής των πειραμάτων ώστε να συμπεριλάβουν σύνολα δεδομένων που περιλαμβάνουν προηγούμενες καταστάσεις αισθητήρων, εμφανίζεται ένα σημαντικό άλμα στην ακρίβεια πρόβλεψης. Ανεβαίνοντας σημαντικά στο αξιόπαινο 91,6%, αυτή η αξιοσημείωτη βελτίωση είναι ενδεικτική του σημαίνοντος ρόλου που διαδραματίζουν οι ιστορικές καταστάσεις των αισθητήρων στη βελτίωση των προβλεπτικών ικανοτήτων του μοντέλου. Αυτή η αύξηση δεν περιορίζεται μόνο σε μια συνολική αύξηση της ακρίβειας, αλλά επεκτείνεται και σε ανώτερες επιδόσεις που παρατηρούνται τόσο σε χαμηλά όσο και σε υψηλά επίπεδα. Η συμπερίληψη των ιστορικών δεδομένων συμβάλλει προφανώς σε ένα πιο ισχυρό και διαφοροποιημένο πλαίσιο πρόβλεψης. Αυτό υπογραμμίζει την καθοριστική σημασία της αξιοποίησης ολοκληρωμένων συνόλων δεδομένων σε σενάρια FL για αυξημένη αποτελεσματικότητα του μοντέλου. Η βαθιά αντίθεση μεταξύ των δύο πειραματικών διατάξεων υπογραμμίζει τη δυναμική επίδραση της σύνθεσης των συνόλων δεδομένων στην απόδοση των μοντέλων Federated Learning.



Εικόνα 24: Αποτελέσματα Federated Learning - Previous State

Στη συνέχεια παρουσιάζεται ένα συγκριτικό διάγραμμα ανάμεσα στα δύο πειράματα federated learning με τις ακρίβειες για κάθε εποχή. Είναι φανερό ότι το μοντέλο που έχει εκπαιδευτεί με τα δεδομένα που περιέχουν και τις προηγούμενες καταστάσεις, δίνει πολύ καλύτερα αποτελέσματα.



Εικόνα 25: Συνολικά αποτελέσματα FL

7 Συμπεράσματα και Μελλοντική Έρευνα

Η παραπάνω μελέτη και τα αποτελέσματά της βοηθούν στην καλύτερη και βαθύτερη κατανόηση των δεδομένων parking, καθώς και στην στοχευμένη αξιοποίησή τους. Με τη διεξαγωγή ποικίλων πειραμάτων, τα αποτελέσματα που προκύπτουν παρέχουν χρήσιμες πληροφορίες, η αξιοποίηση των οποίων συμβάλλει στην καλύτερη οργάνωση για μελλοντική έρευνα.

7.1 Ανάλυση αποτελεσμάτων

Η επιλογή του μοντέλου Keras αναδείχθηκε ως η βέλτιστη επιλογή με βάση την ανώτερη ακρίβεια που προσφέρει σε διάφορες μετρήσεις, που περιλαμβάνουν διάφορους ταξινομητές, learning rates και layers. Ωστόσο, όταν το μοντέλο εκπαιδεύτηκε αρχικά για όλους τους δρόμους και τις ομάδες, η απόδοσή του ήταν μη βέλτιστη. Αυτή η ασυμφωνία προκάλεσε μια καίρια προσαρμογή: την ενοποίηση των επιπέδων σε κατηγορίες "Low" και "High", λαμβάνοντας παράλληλα υπόψη την προηγούμενη κατάσταση της τοποθεσίας. Αυτή η τροποποίηση είχε ως αποτέλεσμα σημαντικά βελτιωμένη ακρίβεια πρόβλεψης,

γεγονός που υποδηλώνει ότι η συνεκτίμηση των ιστορικών δεδομένων ενισχύει την ικανότητα μάθησης του μοντέλου.

Η επακόλουθη εξέταση, που πραγματοποιήθηκε με τη δοκιμή του μοντέλου σε κάθε ημέρα ξεχωριστά, αποκάλυψε μια πολύ καλή ακρίβεια που παρατηρήθηκε τις Κυριακές και τις αργίες. Οι ακριβείς λόγοι πίσω από αυτό το φαινόμενο μπορεί οφείλονται σε διάφορες παραμέτρους όπως:

- **Μειωμένη κυκλοφορία και δραστηριότητα:** Τις Κυριακές και τις αργίες παρατηρείται συνήθως μείωση της συνολικής κυκλοφορίας και της αστικής δραστηριότητας. Με λιγότερα οχήματα στο δρόμο και μειωμένη συνολική κίνηση, το μοντέλο μπορεί να συναντήσει ένα λιγότερο πολύπλοκο περιβάλλον, καθιστώντας ευκολότερη την ακριβή πρόβλεψη της διαθεσιμότητας στάθμευσης.
- **Συνεπής συμπεριφορά στάθμευσης:** Τις Κυριακές και τις αργίες, η συμπεριφορά στάθμευσης μπορεί να ακολουθεί πιο προβλέψιμα πρότυπα. Οι άνθρωποι μπορεί να ασχολούνται με δραστηριότητες αναψυχής ή να μένουν στο σπίτι, οδηγώντας σε πιο συνεπείς επιλογές στάθμευσης. Αυτή η συνέπεια επιτρέπει στο μοντέλο να μαθαίνει και να προσαρμόζεται αποτελεσματικότερα, με αποτέλεσμα τη βελτίωση της ακρίβειας.
- **Μειωμένη μεταβλητότητα:** Οι καθημερινές ημέρες μπορεί να εισάγουν μεταβλητότητα λόγω παραγόντων όπως οι ώρες αιχμής, οι μετακινήσεις που σχετίζονται με την εργασία και τα σχολικά προγράμματα. Οι Κυριακές και οι αργίες, που χαρακτηρίζονται από έναν πιο ομοιόμορφο ρυθμό δραστηριότητας, παρουσιάζουν ένα λιγότερο μεταβλητό περιβάλλον για την πλοήγηση του μοντέλου, συμβάλλοντας ενδεχομένως στη βελτίωση της ακρίβειας.

Μια άλλη αποκαλυπτική δοκιμή αφορούσε την αξιολόγηση των επιδόσεων του μοντέλου σε διαφορετικές ώρες της ημέρας. Ειδικότερα, το μοντέλο παρουσίασε μειωμένη ακρίβεια μεταξύ 9:00 και 14:00. Οι υποδείξεις για τις

μειωμένες επιδόσεις του μοντέλου αυτές τις μέρες και ώρες, μπορεί να οφείλονται σε αρκετούς παράγοντες:

- **Αυξημένη Κίνηση και Δραστηριότητα:** Οι εργασιακές ώρες Δευτέρας έως Παρασκευής χαρακτηρίζονται συχνά από αυξημένη κίνηση και δραστηριότητα σε αστικές περιοχές. Η υψηλή κίνηση μπορεί να δημιουργεί περιορισμούς στην ακριβή πρόβλεψη της διαθεσιμότητας χώρων στάθμευσης.
- **Μεταβλητότητα της Ζήτησης:** Κατά τις εργασιακές ώρες, οι άνθρωποι ενδέχεται να αναζητούν χώρους στάθμευσης για εργασία, ενώ ταυτόχρονα υπάρχει και η κίνηση των εργαζομένων και των μαθητών. Η μεταβλητή ζήτηση κατά τις εργασιακές ώρες μπορεί να δυσκολεύει το μοντέλο στην πρόβλεψη.
- **Συχνή Αλλαγή του Περιβάλλοντος:** Οι αντίξοες καιρικές συνθήκες, οι κατασκευαστικές εργασίες και άλλοι παράγοντες που επηρεάζουν το περιβάλλον κατά τις εργασιακές ώρες ενδέχεται να επιδρούν στην ακρίβεια των προβλέψεων.
- **Κατανομή του Πληθυσμού:** Ο πληθυσμός διαφοροποιείται κατά τις διάφορες ημέρες της εβδομάδας. Κατά τις εργασιακές ώρες, οι περιοχές που χρησιμοποιούνται πιο εντατικά για εργασία ενδέχεται να εμφανίζουν διαφορετικά μοτίβα στάθμευσης.

Η συνδυασμένη επίδραση αυτών των παραγόντων μπορεί να επηρεάσει αρνητικά την ικανότητα του μοντέλου να προβλέπει ακριβώς τη διαθεσιμότητα χώρων στάθμευσης κατά τις εν λόγω ώρες.

Αφού αναγνωρίστηκε ότι το μοντέλο αντιμετώπιζε προκλήσεις για τις καθημερινές και συγκεκριμένες ώρες (9:00 έως 14:00), διεξήχθη μια στοχευμένη δοκιμή. Τα αποτελέσματα επιβεβαίωσαν την όχι τόσο καλή ακρίβεια του μοντέλου κατά τις καθημερινές και τις συγκεκριμένες ώρες, προτρέποντας σε μια βαθύτερη διερεύνηση.

Στις δοκιμές AutoML, χρησιμοποιώντας τη βιβλιοθήκη Autokeras, παρόλο που η ακρίβεια είναι αρκετά υψηλή, δεν ξεπερνά την ακρίβεια των προβλέψεων όπου χρησιμοποιούνται τα προηγούμενα μοντέλα. Μια πιθανή εξήγηση είναι ότι η πολυπλοκότητα της εργασίας πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης μπορεί να μην ωφελείται σημαντικά από την αυτοματοποιημένη επιλογή μοντέλου και τη ρύθμιση υπερπαραμέτρων που εκτελείται από το Autokeras. Σε περιπτώσεις όπου το σύνολο δεδομένων ή τα χαρακτηριστικά του προβλήματος είναι σχετικά απλά, η χειροκίνητη δημιουργία μοντέλου μπορεί να αρκεί ή και να υπερτερεί των προσεγγίσεων του AutoML. Στην περίπτωση του Autosklearn βέβαια, τα αποτελέσματα στις περισσότερες περιπτώσεις είναι καλύτερα με την αυτόματη δημιουργία μοντέλων, σε αρκετά μικρό βαθμό βέβαια.

Επιπλέον, η επιλογή του χώρου αναζήτησης και ο αριθμός δοκιμών (`max_trials`) στη ρύθμιση AutoML ενδέχεται να περιορίσει τη διερεύνηση πιθανών αρχιτεκτονικών και συνδυασμών υπερπαραμέτρων. Η αύξηση του αριθμού των δοκιμών ή η προσαρμογή του χώρου αναζήτησης μπορεί να αποφέρει πιο ανταγωνιστικά αποτελέσματα.

Ακόμη, η αποτελεσματικότητα του AutoML μπορεί να επηρεαστεί από τα ειδικά χαρακτηριστικά του συνόλου δεδομένων και του τομέα του προβλήματος. Η αυτοματοποιημένη προσέγγιση μπορεί να δυσκολεύεται να ξεπεράσει τα χειροκίνητα δημιουργημένα μοντέλα όταν η εργασία απαιτεί ειδικές γνώσεις του τομέα ή όταν το σύνολο δεδομένων δεν είναι αρκετά μεγάλο ώστε να αξιοποιηθεί η πολυπλοκότητα που εισάγουν οι αυτοματοποιημένες αρχιτεκτονικές.

Το AutoML μπορεί να είναι ένα ισχυρό εργαλείο για διάφορες εργασίες μηχανικής μάθησης, η απόδοσή του μπορεί να ποικίλλει ανάλογα με την πολυπλοκότητα του προβλήματος και τα χαρακτηριστικά του συνόλου δεδομένων. Τα συγκρίσιμα ή ελαφρώς κατώτερα αποτελέσματα σε αυτό το σενάριο υποδηλώνουν ότι, για το συγκεκριμένο έργο πρόβλεψης της διαθεσιμότητας χώρων στάθμευσης, η χειροκίνητη δημιουργία μοντέλων παραμένει μια ανταγωνιστική και βιώσιμη επιλογή.

Τα αποτελέσματα των πειραμάτων FL παρέχουν πολύτιμες πληροφορίες σχετικά με την αποτελεσματικότητα της προσέγγισης για την πρόβλεψη της διαθεσιμότητας θέσεων στάθμευσης. Ειδικότερα, κατά τη διεξαγωγή δοκιμών στο σύνολο δεδομένων χωρίς προηγούμενες καταστάσεις αισθητήρων, το μοντέλο πέτυχε ακρίβεια 79,5% μετά από 5 εποχές και 2 επίπεδα. Ο επαναληπτικός χαρακτήρας της διαδικασίας εκπαίδευσης αποκάλυψε μια σταθερή αύξηση της ακρίβειας σε όλες τις εποχές, φτάνοντας τελικά στη βέλτιστη απόδοση. Ωστόσο, προέκυψε μια αξιοσημείωτη παρατήρηση σχετικά με τις προβλέψεις χαμηλού επιπέδου, υποδεικνύοντας περιθώρια βελτίωσης στη συγκεκριμένη κατηγορία.

Αντίθετα, η συμπερίληψη προηγούμενων καταστάσεων αισθητήρων στο σύνολο δεδομένων απέδωσε σημαντικά βελτιωμένα αποτελέσματα. Η ακρίβεια εκτοξεύθηκε στο 91,6%, αναδεικνύοντας τη θετική επίδραση της ενσωμάτωσης ιστορικών δεδομένων στις επιδόσεις πρόβλεψης. Η βελτίωση αυτή δεν ήταν εμφανής μόνο στη συνολική ακρίβεια, αλλά εκδηλώθηκε επίσης ως ανώτερα αποτελέσματα τόσο σε χαμηλό όσο και σε υψηλό επίπεδο. Η παρουσία ιστορικού πλαισίου συνέβαλε προφανώς στην ικανότητα του μοντέλου να κάνει ακριβέστερες προβλέψεις, υπογραμμίζοντας τη σημασία των πληροφοριών πλαισίου στο παράδειγμα του FL.

Οι πρακτικές FL παρουσίασαν πολλά υποσχόμενα αποτελέσματα, με τη συμπερίληψη των ιστορικών καταστάσεων των αισθητήρων να αποδεικνύεται βασικός παράγοντας για την ενίσχυση της ακρίβειας πρόβλεψης. Οι βελτιώσεις που παρατηρήθηκαν υπογραμμίζουν τις δυνατότητες του FL στην αξιοποίηση αποκεντρωμένων δεδομένων για πιο ισχυρές και συνειδητοποιημένες ως προς το πλαίσιο προβλέψεις διαθεσιμότητας χώρων στάθμευσης.

Συμπερασματικά, ενώ το μοντέλο Keras επέδειξε εξαιρετικές δυνατότητες πρόβλεψης σε πολλά πειράματα και πρακτικές, υπάρχουν περιπτώσεις που απαιτούν περαιτέρω διερεύνηση. Τα παρατηρούμενα μοτίβα, ιδίως στο πλαίσιο συγκεκριμένων ημερών και ωρών, χρήζουν βαθύτερης κατανόησης της υποκείμενης δυναμικής.

7.2 Μελλοντικοί στόχοι

Μελλοντικά, ο στόχος είναι να ενισχυθεί η ακρίβεια πρόβλεψης του μοντέλου, ιδίως σε σενάρια στα οποία η απόδοσή του δεν ήταν η βέλτιστη δυνατή. Με την ενσωμάτωση πρόσθετων παραμέτρων και τη διεύρυνση του πεδίου των δεδομένων εκπαίδευσης, η προσαρμοστικότητα και η αξιοπιστία του μοντέλου μπορούν να βελτιωθούν περαιτέρω.

Ένας κρίσιμος τομέας για μελλοντική διερεύνηση περιλαμβάνει την επέκταση του συνόλου δεδομένων εκπαίδευσης ώστε να περιλαμβάνει ένα ευρύτερο φάσμα παραγόντων που επηρεάζουν τη διαθεσιμότητα χώρων στάθμευσης. Η ενσωμάτωση δεδομένων σχετικά με τα πρότυπα οδικής κυκλοφορίας, την εγγύτητα σε σχολεία, τις καιρικές συνθήκες σε πραγματικό χρόνο, τις τρέχουσες οδικές εργασίες και την παρουσία χώρων πρασίνου μπορεί να προσφέρει μια πιο ολοκληρωμένη κατανόηση του δυναμικού αστικού περιβάλλοντος.

Η ενσωμάτωση δεδομένων κυκλοφορίας σε πραγματικό χρόνο στην εκπαίδευση του μοντέλου μπορεί να ενισχύσει την ικανότητά του να πλοηγείται στην αστική συμφόρηση. Η κατανόηση της κυκλοφοριακής ροής σε συγκεκριμένους δρόμους και η συσχέτισή της με τη ζήτηση στάθμευσης μπορεί να συμβάλει σε ακριβέστερες προβλέψεις, ιδίως κατά τις ώρες αιχμής.

Η εξέταση της εγγύτητας των δρόμων σε εκπαιδευτικά ιδρύματα εισάγει ένα άλλο επίπεδο πολυπλοκότητας. Το μοντέλο μπορεί να τελειοποιηθεί ώστε να αναγνωρίζει μοτίβα που σχετίζονται με τα σχολικά προγράμματα και τις δραστηριότητες, παρέχοντας ακριβέστερες προβλέψεις κατά τις κρίσιμες ώρες.

Οι καιρικές συνθήκες διαδραματίζουν καθοριστικό ρόλο στην αστική κινητικότητα. Με την ενσωμάτωση δεδομένων καιρού σε πραγματικό χρόνο στο μοντέλο, αυτό μπορεί να προσαρμοστεί στις επιπτώσεις των δυσμενών καιρικών συνθηκών στη συμπεριφορά στάθμευσης. Για παράδειγμα, η αυξημένη ζήτηση στάθμευσης κατά τη διάρκεια κακοκαιρίας ή η μειωμένη ζήτηση σε ευχάριστες ημέρες μπορεί να ληφθεί υπόψη στις προβλέψεις.

Δυναμικοί παράγοντες όπως οι οδικές ή κατασκευαστικές εργασίες μπορούν να επηρεάσουν σημαντικά τη διαθεσιμότητα στάθμευσης. Η ενσωμάτωση πληροφοριών σχετικά με το κλείσιμο δρόμων, τις δραστηριότητες συντήρησης ή τα τοπικά γεγονότα μπορεί να βελτιώσει την ανταπόκριση του μοντέλου σε ξαφνικές αλλαγές στο αστικό τοπίο.

Η παρουσία χώρων πρασίνου μπορεί να επηρεάσει τις προτιμήσεις στάθμευσης, καθώς οι οδηγοί μπορεί να έχουν την τάση να επιλέγουν θέσεις κοντά σε πάρκα ή χώρους αναψυχής. Η συνεκτίμηση της διαθεσιμότητας και της προσβασιμότητας χώρων πρασίνου κοντά σε δρόμους μπορεί να προσθέσει ένα διαφοροποιημένο επίπεδο στις προβλέψεις του μοντέλου.

Η συνεργασία με τις τοπικές αρχές, τους πολεοδόμους και τα ενδιαφερόμενα μέρη μπορεί να προσφέρει πολύτιμες ποιοτικές πληροφορίες. Οι συνεντεύξεις και οι διαβουλεύσεις μπορούν να αποκαλύψουν συγκεκριμένους παράγοντες της περιοχής που μπορεί να μην είναι άμεσα εμφανείς μόνο από τα δεδομένα.

Σχετικά με την καινοτόμα τεχνολογία του Federated Learning, μια διαφοροποιημένη διερεύνηση των παραμέτρων, συμπεριλαμβανομένων των γύρων επικοινωνίας και των ρυθμών μάθησης, έχει τη δυνατότητα να βελτιώσει τη σύγκλιση του μοντέλου και τη συνολική απόδοση. Ο εμπλουτισμός του συνόλου δεδομένων με πρόσθετα χαρακτηριστικά του πλαισίου, όπως αναφέρθηκαν παραπάνω, θα μπορούσε να ενισχύσει περαιτέρω την κατανόηση του μοντέλου. Οι δυναμικές στρατηγικές σχηματισμού ομάδων υπόσχονται την προσαρμογή των μοντέλων στις εξελισσόμενες περιβαλλοντικές συνθήκες. Οι προηγμένες τεχνικές διατήρησης της ιδιωτικότητας στο πλαίσιο της ομοσπονδιακής μάθησης, συμπεριλαμβανομένης της ασφαλούς συνάθροισης και της διαφορικής ιδιωτικότητας, θα μπορούσαν να αντιμετωπίσουν τις ανησυχίες για την ασφάλεια των δεδομένων. Ξεπερνώντας τις μεμονωμένες τοποθεσίες, οι εφαρμογές μάθησης μεταφοράς σε διάφορα γεωγραφικά πλαίσια μπορούν να ενισχύσουν τη γενίκευση του μοντέλου. Οι προσεγγίσεις FL σε πραγματικό χρόνο

και οι τυποποιημένες μετρικές αξιολόγησης προσαρμοσμένες στην πρόβλεψη χώρων στάθμευσης θα συμβάλουν στην πρόοδο του τομέα.

Γενικότερα, η επαναληπτική βελτίωση του μοντέλου με βάση την ανατροφοδότηση από τον πραγματικό κόσμο και τη συνεχή μάθηση από τα τρέχοντα δεδομένα μπορεί να διασφαλίσει την προσαρμοστικότητά του στις εξελισσόμενες αστικές δυναμικές. Οι τακτικές ενημερώσεις και προσαρμογές θα είναι απαραίτητες για τη διατήρηση της συνέπειας και της αποτελεσματικότητας του μοντέλου με την πάροδο του χρόνου.

ΠΑΡΑΡΤΗΜΑ

Keras MLP Train

```
import os
import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import ConfusionMatrixDisplay

from tensorflow import keras
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from tensorflow.keras.utils import to_categorical
#%%

np.set_printoptions(precision=2)

config = {
    'epochs': 20,
    'batch_size': 32,
    'scale_input': True,
    'plot_cm': False,
    'levels': [0.00, 0.20, 0.40, 1.01],
# 'levels': [0.00, 0.40, 1.01]
}
#%%

def avail2load(l):
    levels = config['levels']
    lvl = -1
    n = len(levels)-1
    for i in range(n):
        if levels[i] <= l and l < levels[i+1]:
            lvl = n-1-i

    return lvl

def data_reading():
    df = pd.read_csv('preprocessed_data/grouped_2022_23.csv',
encoding='iso8859_7')
    # print(df)
```

```

    # df = df[df['group'] == 0] # if we want to work with a specific group
of sensors
    # df = df.drop(columns=['year'])
    # df = df.drop(columns=['month'])
    df = df.drop(columns=['day'])
    # df = df.drop(columns=['weekday'])
    # df = df[df['weekday'] == 5]

    num_free = df['free'].values
    num_total = df['total'].values

    availability = num_free/num_total
    load = np.copy(availability)
    for i in range(availability.shape[0]):
        load[i] = avail2load(availability[i])

    levels = np.unique(load)
    for level in levels:
        ct = np.count_nonzero(load == level)
        print(level, ct)

    # sys.exit(1)

    df['load'] = load

    df = df.drop(columns=['free'])
    df = df.drop(columns=['occupied'])
    df = df.drop(columns=['total'])

    """
    df = df.drop(columns=['prev_free'])
    df = df.drop(columns=['prev_occupied'])
    df = df.drop(columns=['prev_total'])
    """

    print(df)

    return df

def data_preparation(df):
    df_tmp = df

    y = df_tmp['load']
    X = df_tmp.drop(columns=['load'])

    y_un = np.unique(y)
    print('y_un=', y_un)

```

```

# print(y)

print("X=")
print(X)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
X_train = X_train.values
X_test = X_test.values
y_train = y_train.values
y_test = y_test.values

scale_input = config['scale_input']
if scale_input:
    sc = StandardScaler()
    # sc = MinMaxScaler()
    sc.fit(X_train)
    X_train = sc.transform(X_train)
    X_test = sc.transform(X_test)

le = LabelEncoder()
le.fit(y_train)
print("le.classes_", le.classes_)

y_train = le.transform(y_train)
y_test = le.transform(y_test)

y_train = to_categorical(y_train)
y_test = to_categorical(y_test)

return X_train, X_test, y_train, y_test

def create_model(input_dim, output_dim):
    model = keras.Sequential()
    model.add(keras.layers.Dense(512*2, input_dim=input_dim,
activation='relu'))
    model.add(keras.layers.Dense(512*2, activation='relu'))
    model.add(keras.layers.Dense(512*2, activation='relu'))
    model.add(keras.layers.Dense(output_dim, kernel_initializer='normal',
activation="softmax"))

    opt = keras.optimizers.Adam()
    loss = 'categorical_crossentropy'

    metrics = []
    metrics.append('accuracy')

    model.compile(loss=loss, optimizer=opt, metrics=metrics)

```

```
model.summary()

return model

if __name__ == '__main__':

    df = data_reading()
    X_train, X_test, y_train, y_test = data_preparation(df)

    # model creation
    input_dim = X_train.shape[1]
    output_dim = y_train.shape[1]

    model = create_model(input_dim, output_dim)

    # model training
    epochs = config['epochs']
    batch_size = config['batch_size']

    checkpoint_filepath = './tmp/checkpoint'
    model_checkpoint_callback = keras.callbacks.ModelCheckpoint(
        filepath=checkpoint_filepath,
        save_weights_only=True,
        monitor='val_accuracy',
        mode='max',
        save_best_only=True)

    callbacks=[model_checkpoint_callback]

    history = model.fit(X_train, y_train,
                        validation_data=(X_test, y_test),
                        epochs=epochs,
                        batch_size=batch_size,
                        verbose=2,
                        workers=4,
                        callbacks=callbacks)

    model.load_weights(checkpoint_filepath)

    train_loss = history.history['loss']
    train_acc = history.history['accuracy']
    val_loss = history.history['val_loss']
    val_acc = history.history['val_accuracy']

    print("training done")

    print(train_acc)
```

```

print(val_acc)

# model evaluation
y_pred = model.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)
y_test = np.argmax(y_test, axis=1)

# evaluation metrics
acc = accuracy_score(y_test, y_pred)
print("accuracy=\n", acc)

cm = confusion_matrix(y_test, y_pred)
print("confusion matrix=\n", cm)

cm = confusion_matrix(y_test, y_pred, normalize='true')
print("normalized confusion matrix=\n", cm)

if output_dim == 3:
    target_names = ['Low', 'Medium', 'High']
else:
    target_names = ['Low', 'High']

plot_cm = config['plot_cm']
if plot_cm:
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=target_names)
    disp.plot()
    plt.show()

report = classification_report(y_test, y_pred,
target_names=target_names)
print('classification report=\n', report)

# save model
pid = os.getpid()
model.save('model_{}_dev.h5'.format(pid))

```

Keras MLP Train (2 levels, Previous State)

```

import os
import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder

```

```

from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import ConfusionMatrixDisplay

from tensorflow import keras
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from tensorflow.keras.utils import to_categorical

np.set_printoptions(precision=2)

config = {
    'epochs': 20,
    'batch_size': 32,
    'scale_input': True,
    'plot_cm': True,
    'levels': [0.00, 0.40, 1.01]
}

# levels = [0.00, 0.20, 0.40, 1.01]
# levels = [0.00, 0.40, 1.01]

###
def avail2load(l):
    levels = config['levels']
    lvl = -1
    n = len(levels)-1
    for i in range(n):
        if levels[i] <= l and l < levels[i+1]:
            lvl = n-1-i

    return lvl

###

def data_reading():
    df = pd.read_csv('preprocessed_data/grouped_2022_23_v3.csv',
encoding='iso8859_7')
    # print(df)

    # df = df[df['group'] == 0] # if we want to work with a specific group
of sensors
    df = df.drop(columns=['year'])
    df = df.drop(columns=['month'])
    # df = df.drop(columns=['day'])
    # df = df.drop(columns=['weekday'])
    # df = df[df['weekday']==5]

    num_free = df['free'].values

```

```

num_total = df['total'].values

num_prev_free = df['prev_free'].values
num_prev_total = df['prev_total'].values

availability = num_free/num_total
prev_availability = num_prev_free/num_prev_total

load = np.copy(availability)
prev_load = np.copy(prev_availability)
for i in range(availability.shape[0]):
    load[i] = avail2load(availability[i])
    prev_load[i] = avail2load(prev_availability[i])

levels = np.unique(load)
for level in levels:
    ct = np.count_nonzero(load == level)
    print(level, ct)

# sys.exit(1)

df['load'] = load
df['prev_load'] = prev_load

df = df.drop(columns=['free'])
df = df.drop(columns=['occupied'])
df = df.drop(columns=['total'])

df = df.drop(columns=['prev_free'])
df = df.drop(columns=['prev_occupied'])
df = df.drop(columns=['prev_total'])

print(df)

return df

def data_preparation(df):
    df_tmp = df

    y = df_tmp['load']
    X = df_tmp.drop(columns=['load'])

    y_un = np.unique(y)
    print('y_un=', y_un)
    # print(y)

    print("X=")

```

```

print(X)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
X_train = X_train.values
X_test = X_test.values
y_train = y_train.values
y_test = y_test.values

scale_input = config['scale_input']
if scale_input:
    sc = StandardScaler()
    # sc = MinMaxScaler()
    sc.fit(X_train)
    X_train = sc.transform(X_train)
    X_test = sc.transform(X_test)

le = LabelEncoder()
le.fit(y_train)
print("le.classes_=", le.classes_)

y_train = le.transform(y_train)
y_test = le.transform(y_test)

y_train = to_categorical(y_train)
y_test = to_categorical(y_test)

return X_train, X_test, y_train, y_test

def create_model(input_dim, output_dim):
    model = keras.Sequential()
    model.add(keras.layers.Dense(512*2, input_dim=input_dim,
activation='relu'))
    model.add(keras.layers.Dense(512*2, activation='relu'))
    model.add(keras.layers.Dense(512*2, activation='relu'))
    model.add(keras.layers.Dense(output_dim, kernel_initializer='normal',
activation="softmax"))

    opt = keras.optimizers.Adam(learning_rate=0.01)
    loss = 'categorical_crossentropy'

    metrics = []
    metrics.append('accuracy')

    model.compile(loss=loss, optimizer=opt, metrics=metrics)
    model.summary()

    return model

```



```
if __name__ == '__main__':

    df = data_reading()
    X_train, X_test, y_train, y_test = data_preparation(df)

    # model creation
    input_dim = X_train.shape[1]
    output_dim = y_train.shape[1]

    model = create_model(input_dim, output_dim)

    # model training
    epochs = config['epochs']
    batch_size = config['batch_size']

    checkpoint_filepath = './tmp/checkpoint'
    model_checkpoint_callback = keras.callbacks.ModelCheckpoint(
        filepath=checkpoint_filepath,
        save_weights_only=True,
        monitor='val_accuracy',
        mode='max',
        save_best_only=True)

    callbacks=[model_checkpoint_callback]

    history = model.fit(X_train, y_train,
                        validation_data=(X_test, y_test),
                        epochs=epochs,
                        batch_size=batch_size,
                        verbose=2,
                        workers=4,
                        callbacks=callbacks)

    model.load_weights(checkpoint_filepath)

    train_loss = history.history['loss']
    train_acc = history.history['accuracy']
    val_loss = history.history['val_loss']
    val_acc = history.history['val_accuracy']

    print("training done")

    print(train_acc)
    print(val_acc)

    # model evaluation
```

```

y_pred = model.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)
y_test = np.argmax(y_test, axis=1)

# evaluation metrics
acc = accuracy_score(y_test, y_pred)
print("accuracy=\n", acc)

cm = confusion_matrix(y_test, y_pred)
print("confusion matrix=\n", cm)

cm = confusion_matrix(y_test, y_pred, normalize='true')
print("normalized confusion matrix=\n", cm)

if output_dim == 3:
    target_names = ['Low', 'Medium', 'High']
else:
    target_names = ['Low', 'High']

plot_cm = config['plot_cm']
if plot_cm:
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=target_names)
    disp.plot()
    plt.show()
    plt.savefig(os.path.join(os.getcwd()
+' /results_katerina/keras_all_data_2_levels.png'))

report = classification_report(y_test, y_pred,
target_names=target_names)
print('classification report=\n', report)

# save model
pid = os.getpid()
model.save('model_{}_dev.h5'.format(pid))

```

Keras MLP Train (All days, Sundays, Holidays)

- Επιλογή συγκεκριμένης ημέρας:

```
df = df[df['weekday']!=5]
```

- Εισαγωγή βιβλιοθήκης για αργίες και dataframe 'hol':

```
import holidays
```

```
gr_holidays = holidays.country_holidays('GR')
df['hol'] = df.apply(lambda row: datetime.datetime(row['year'],
row['month'], row['day']) in gr_holidays, axis=1)
df['hol'] = df.apply(lambda row: datetime.datetime(row['year'],
row['month'], row['day']).weekday() == 6, axis=1)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
X_train = X_train.values
X_test = df_tmp[df_tmp['hol']== True].drop(columns=['load'])
y_train = y_train.values
y_test = df_tmp[df_tmp['hol']== True]['load']
```

Keras MLP Train (Hours)

- Ανάλυση κάθε ώρας για όλες τις μέρες:

```
def hourwise_testing(df_test, sc, le, model):
    results = {'accuracy': [], 'loss': []}

    for hour in np.unique(df['hour']):
        print(f"\nTesting for Hour {hour}:")

        # Select data for the current hour
        X_test_hour = df_test[df_test['hour'] ==
hour].drop(columns=['load'])
        y_test_hour = df_test[df_test['hour'] == hour]['load']

        X_test_hour = sc.transform(X_test_hour)

        y_test_hour = le.transform(y_test_hour)

        y_test_hour = to_categorical(y_test_hour)

        # Evaluate the model on the current hour's test set
        loss, accuracy = model.evaluate(X_test_hour, y_test_hour,
verbose=0)
        print('Loss:', loss)
        print('Accuracy:', accuracy)

        results['accuracy'].append(accuracy)
        results['loss'].append(loss)
        # results['confusion_matrices'].append(confusion_matrix_hour)
```

```
return results
```

- Ανάλυση κάθε ώρας για τις καθημερινές:

```
df['work'] = df.apply(lambda row: (datetime.datetime(row['year'], row['month'], row['day']) not in gr_holidays) and (datetime.datetime(row['year'], row['month'], row['day']).weekday() not in [5, 6]), axis=1)
```

- Ανάλυση κάθε ώρας για Κυριακές και αργίες:

```
df = df[df.apply(lambda row: (datetime.datetime(row['year'], row['month'], row['day']) in gr_holidays), axis=1)]
```

- Ανάλυση ωρών 9:00 – 14:00 για τις καθημερινές:

```
df['work'] = df.apply(lambda row: (datetime.datetime(row['year'], row['month'], row['day']) not in gr_holidays) and (datetime.datetime(row['year'], row['month'], row['day']).weekday() not in [5, 6]) and (9 <= row['hour'] < 15), axis=1)
```

Autokeras

```
import os
import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import ConfusionMatrixDisplay

from tensorflow import keras
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from tensorflow.keras.utils import to_categorical
import autokeras as ak
from autokeras import StructuredDataClassifier
```

```

np.set_printoptions(precision=2)

config = {
    'epochs': 20,
    'batch_size': 32,
    'scale_input': True,
    'plot_cm': True,
    'levels': [0.00, 0.40, 1.01]
    #'levels' : [0.00, 0.20, 0.40, 1.01],
}

def avail2load(l):
    levels = config['levels']
    lvl = -1
    n = len(levels)-1
    for i in range(n):
        if levels[i] <= l and l < levels[i+1]:
            lvl = n-1-i

    return lvl

def data_reading():
    df = pd.read_csv('preprocessed_data/grouped_2022_23_v3.csv',
encoding='iso8859_7')
    # print(df)

    # df = df[df['group'] == 0] # if we want to work with a specific group
of sensors
    df = df.drop(columns=['year'])
    df = df.drop(columns=['month'])
    # df = df.drop(columns=['day'])
    # df = df.drop(columns=['weekday'])
    # df = df[df['weekday']==5]

    num_free = df['free'].values
    num_total = df['total'].values

    num_prev_free = df['prev_free'].values
    num_prev_total = df['prev_total'].values

    availability = num_free/num_total
    prev_availability = num_prev_free/num_prev_total

    load = np.copy(availability)
    prev_load = np.copy(prev_availability)
    for i in range(availability.shape[0]):

```

```

    load[i] = avail2load(availability[i])
    prev_load[i] = avail2load(prev_availability[i])

levels = np.unique(load)
for level in levels:
    ct = np.count_nonzero(load == level)
    print(level, ct)

# sys.exit(1)

df['load'] = load
df['prev_load'] = prev_load

df = df.drop(columns=['free'])
df = df.drop(columns=['occupied'])
df = df.drop(columns=['total'])

df = df.drop(columns=['prev_free'])
df = df.drop(columns=['prev_occupied'])
df = df.drop(columns=['prev_total'])

print(df)

return df

def data_preparation(df):
    df_tmp = df

    y = df_tmp['load']
    X = df_tmp.drop(columns=['load'])

    y_un = np.unique(y)
    print('y_un=', y_un)
    # print(y)

    print("X=")
    print(X)

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
    X_train = X_train.values
    X_test = X_test.values
    y_train = y_train.values
    y_test = y_test.values

    scale_input = config['scale_input']
    if scale_input:

```

```

    sc = StandardScaler()
    # sc = MinMaxScaler()
    sc.fit(X_train)
    X_train = sc.transform(X_train)
    X_test = sc.transform(X_test)

    le = LabelEncoder()
    le.fit(y_train)
    print("le.classes_=", le.classes_)

    y_train = le.transform(y_train)
    y_test = le.transform(y_test)

    y_train = to_categorical(y_train)
    y_test = to_categorical(y_test)

    return X_train, X_test, y_train, y_test

def create_model(input_dim, output_dim):
    model = keras.Sequential()
    model.add(keras.layers.Dense(512*2, input_dim=input_dim,
activation='relu'))
    model.add(keras.layers.Dense(512*2, activation='relu'))
    model.add(keras.layers.Dense(512*2, activation='relu'))
    model.add(keras.layers.Dense(output_dim, kernel_initializer='normal',
activation="softmax"))

    opt = keras.optimizers.Adam(learning_rate=0.01)
    loss = 'categorical_crossentropy'

    metrics = []
    metrics.append('accuracy')

    model.compile(loss=loss, optimizer=opt, metrics=metrics)
    model.summary()

    return model

def create_automodel(input_dim=None, output_dim=None):
    # define search
    tmp_path = "./tmp"
    os.system('rm -rf ./tmp')

    """
    search = StructuredDataClassifier(overwrite=True,
                                     max_trials=10,

```

```

        directory=tmp_path,
        # loss="mean_absolute_error",
        objective="val_accuracy",
        # metrics=["mean_absolute_error"],
        multi_label=True,
        tuner="bayesian")

return search
"""

input_node = ak.Input()
output_node = input_node
output_node = ak.DenseBlock(num_layers=2, dropout=0.0,
use_batchnorm=False)(output_node)
output_node = ak.ClassificationHead(multi_label=True,
dropout=0.0)(output_node)

search = ak.AutoModel(inputs=input_node,
                      outputs=output_node,
                      overwrite=True,
                      max_trials=5,
                      directory=tmp_path,
                      # loss="categorical_crossentropy",
                      # loss="mean_absolute_error",
                      objective="val_accuracy",
                      # metrics=["mean_absolute_error", "val_acc"],
                      tuner="bayesian") # random, hyperband

return search

if __name__ == '__main__':

df = data_reading()
X_train, X_test, y_train, y_test = data_preparation(df)

input_dim = X_train.shape[1]
output_dim = y_train.shape[1]

# model training
epochs = config['epochs']
batch_size = config['batch_size']

# model creation
if True:

    model = create_model(input_dim, output_dim)
    checkpoint_filepath = './tmp/checkpoint'
    model_checkpoint_callback = keras.callbacks.ModelCheckpoint(

```



```

        filepath=checkpoint_filepath,
        save_weights_only=True,
        monitor='val_accuracy',
        mode='max',
        save_best_only=True)

callbacks=[model_checkpoint_callback]

history = model.fit(X_train, y_train,
                    validation_data=(X_test, y_test),
                    epochs=epochs,
                    batch_size=batch_size,
                    verbose=2,
                    workers=4,
                    callbacks=callbacks)

model.load_weights(checkpoint_filepath)

train_loss = history.history['loss']
train_acc = history.history['accuracy']
val_loss = history.history['val_loss']
val_acc = history.history['val_accuracy']

print("training done")

print(train_acc)
print(val_acc)
else:
    search = create_automodel() # input_dim, output_dim
    # perform the search
    search.fit(x=X_train, y=y_train, epochs=epochs)

    # summarize
    model = search.export_model()

    # model = search
    model.summary()

# model evaluation
y_pred = model.predict(X_test)
print(y_pred)
y_pred = np.argmax(y_pred, axis=1)
y_test = np.argmax(y_test, axis=1)

# evaluation metrics
acc = accuracy_score(y_test, y_pred)
print("accuracy=\n", acc)

```

```

cm = confusion_matrix(y_test, y_pred)
print("confusion matrix=\n", cm)

cm = confusion_matrix(y_test, y_pred, normalize='true')
print("normalized confusion matrix=\n", cm)

levels = len(config['levels'])-1
if levels==3:
    target_names = ['Low', 'Medium', 'High']
else:
    target_names = ['Low', 'High']

plot_cm = config['plot_cm']
if plot_cm:
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=target_names)
    disp.plot()
    plt.show()
    plt.savefig(os.path.join(os.getcwd()
+'results_katerina/kerasMlpTrain_v3.png'))

report = classification_report(y_test, y_pred,
target_names=target_names)
print('classification report=\n', report)

# save model
pid = os.getpid()
model.save('model_{}_dev.h5'.format(pid))

```

Autosklearn

```

import sys
import os
os.environ['OPENBLAS_NUM_THREADS'] = '1'

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.metrics import ConfusionMatrixDisplay

# from tensorflow import keras

```

```

# from tensorflow.keras.utils import to_categorical
# import autokeras as ak
# from autokeras import StructuredDataClassifier
from sklearn.ensemble import RandomForestClassifier
# from sklearn.ensemble import AdaBoostClassifier
# from sklearn.neighbors import KNeighborsClassifier
# from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from autosklearn.classification import AutoSklearnClassifier
np.set_printoptions(precision=2)

config = {
    'epochs': 10,
    'batch_size': 32,
    'scale_input': True,
    'plot_cm': False,
    'levels': [0.00, 0.20, 0.40, 1.01]
}

# levels = [0.00, 0.20, 0.40, 1.01]
# levels = [0.00, 0.40, 1.01]

def avail2load(l):
    levels = config['levels']
    lvl = -1
    n = len(levels)-1
    for i in range(n):
        if levels[i] <= l and l < levels[i+1]:
            lvl = n-1-i

    return lvl

def data_reading():
    df = pd.read_csv('preprocessed_data/grouped_2022_23_v3.csv',
encoding='iso8859_7')
    # print(df)

    # df = df[df['group'] == 0] # if we want to work with a specific group
of sensors
    df = df.drop(columns=['year'])
    df = df.drop(columns=['month'])
    # df = df.drop(columns=['day'])
    # df = df.drop(columns=['weekday'])
    # df = df[df['weekday']==5]

    num_free = df['free'].values

```

```

num_total = df['total'].values

num_prev_free = df['prev_free'].values
num_prev_total = df['prev_total'].values

availability = num_free/num_total
prev_availability = num_prev_free/num_prev_total

load = np.copy(availability)
prev_load = np.copy(prev_availability)
for i in range(availability.shape[0]):
    load[i] = avail2load(availability[i])
    prev_load[i] = avail2load(prev_availability[i])

levels = np.unique(load)
for level in levels:
    ct = np.count_nonzero(load == level)
    print(level, ct)

# sys.exit(1)

df['load'] = load
df['prev_load'] = prev_load

df = df.drop(columns=['free'])
df = df.drop(columns=['occupied'])
df = df.drop(columns=['total'])

df = df.drop(columns=['prev_free'])
df = df.drop(columns=['prev_occupied'])
df = df.drop(columns=['prev_total'])

print(df)

return df

def data_preparation(df):
    df_tmp = df

    y = df_tmp['load']
    X = df_tmp.drop(columns=['load'])

    y_un = np.unique(y)
    print('y_un=', y_un)
    # print(y)

    print("X=")

```

```

print(X)

X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)
X_train = X_train.values
X_test = X_test.values
y_train = y_train.values
y_test = y_test.values

scale_input = config['scale_input']
if scale_input:
    sc = StandardScaler()
    # sc = MinMaxScaler()
    sc.fit(X_train)
    X_train = sc.transform(X_train)
    X_test = sc.transform(X_test)

le = LabelEncoder()
le.fit(y_train)
print("le.classes_=", le.classes_)

y_train = le.transform(y_train)
y_test = le.transform(y_test)

# y_train = to_categorical(y_train)
# y_test = to_categorical(y_test)

return X_train, X_test, y_train, y_test

def create_model(input_dim, output_dim):
    model = RandomForestClassifier() # n_estimators=50,
class_weight="balanced")
    # model = AdaBoostClassifier()
    # model = KNeighborsClassifier(n_neighbors=10)
    # model = DecisionTreeClassifier()

    return model

def create_automodel(input_dim=None, output_dim=None):
    # automl = AutoSklearnClassifier()

    all_estimators = ["adaboost", "bernoulli_nb", "decision_tree",
"extra_trees", "gaussian_nb",
"gradient_boosting", "k_nearest_neighbors", "lda",
"liblinear_svc",
"libsvm_svc", "mlp", "multinomial_nb",
"passive_aggressive",

```

```

        "qda", "random_forest", "sgd"]

estimators_to_use = ["random_forest"]
time_left_for_this_task = 60*2
per_run_time_limit = 60
memory_limit = 65536
n_jobs = 4
ensemble_size = 1
cv_folds = None
if cv_folds is not None and cv_folds > 1:
    kwargs = {'ensemble_size': ensemble_size,
              'cv_folds': cv_folds}
else:
    kwargs = {}

print("e = ", estimators_to_use)
print("time_left_for_this_task=", time_left_for_this_task)
print("n_jobs=", n_jobs)
print("kwargs=", kwargs)

automl = AutoSklearnClassifier(
    time_left_for_this_task=time_left_for_this_task,
    per_run_time_limit=per_run_time_limit,
    # memory_limit=memory_limit,
    n_jobs=n_jobs,
    delete_tmp_folder_after_terminate=True,
    ensemble_size=ensemble_size,
    **kwargs)
    #tmp_folder=os.path.expanduser('~/.auto-sklearn-tmp'))

return automl

def main():

    df = data_reading()
    X_train, X_test, y_train, y_test = data_preparation(df)

    input_dim = X_train.shape[1]
    output_dim = 1 # y_train.shape[1]

    # model training
    epochs = config['epochs']
    batch_size = config['batch_size']

    # model creation
    if False:

        model = create_model(input_dim, output_dim)

```

```

    model.fit(X_train, y_train)
    print("training done")
else:
    model = create_automodel() # input_dim, output_dim
    # perform the search
    model.fit(X_train, y_train)

    # summarize
    print('autosklearn/stats:', model.sprint_statistics())
    print('autosklearn/models:', model.show_models())

# model evaluation
y_pred = model.predict(X_test)
print(y_pred)
# y_pred = np.argmax(y_pred, axis=1)
# y_test = np.argmax(y_test, axis=1)

# evaluation metrics
acc = accuracy_score(y_test, y_pred)
print("accuracy=\n", acc)

cm = confusion_matrix(y_test, y_pred)
print("confusion matrix=\n", cm)

cm = confusion_matrix(y_test, y_pred, normalize='true')
print("normalized confusion matrix=\n", cm)

# if output_dim == 3:
levels = len(config['levels'])-1
if levels==3:
    target_names = ['Low', 'Medium', 'High']
else:
    target_names = ['Low', 'High']

plot_cm = config['plot_cm']
if plot_cm:
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=target_names)
    disp.plot()
    plt.show()

report = classification_report(y_test, y_pred,
target_names=target_names)
print('classification report=\n', report)

# save model
# pid = os.getpid()
# model.save('model_{}_dev.h5'.format(pid))

```

```
if __name__ == '__main__':  
    main()
```

Federated Learning

```
import pandas as pd  
import numpy as np  
import tensorflow as tf  
import keras  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense, Activation  
from sklearn.model_selection import train_test_split  
from sklearn.metrics import accuracy_score  
from sklearn.model_selection import train_test_split  
from sklearn.preprocessing import StandardScaler, MinMaxScaler  
from sklearn.preprocessing import LabelEncoder  
from tensorflow.keras.utils import to_categorical  
import matplotlib.pyplot as plt  
from sklearn.metrics import accuracy_score  
from sklearn.metrics import confusion_matrix  
from sklearn.metrics import classification_report  
from sklearn.metrics import ConfusionMatrixDisplay  
import os  
  
###  
# Load CSV data  
csv_file = 'preprocessed_data/grouped_2022_23_v3.csv'  
  
config = {  
    'epochs': 10,  
    'batch_size': 32,  
    'scale_input': True,  
    'plot_cm': True,  
    'levels': [0.00, 0.40, 1.01]  
}  
  
def avail2load(l):  
    levels = config['levels']  
    lvl = -1  
    n = len(levels)-1  
    for i in range(n):  
        if levels[i] <= l and l < levels[i+1]:  
            lvl = n-1-i  
    return lvl  
  
def data_read():
```



```

df = pd.read_csv('preprocessed_data/grouped_2022_23_v3.csv',
encoding='iso8859_7')

df = df.drop(columns=['year'])

num_free = df['free'].values
num_total = df['total'].values

num_prev_free = df['prev_free'].values
num_prev_total = df['prev_total'].values

availability = num_free/num_total
prev_availability = num_prev_free/num_prev_total

load = np.copy(availability)
prev_load = np.copy(prev_availability)
for i in range(availability.shape[0]):
    load[i] = avail2load(availability[i])
    prev_load[i] = avail2load(prev_availability[i])

levels = np.unique(load)
for level in levels:
    ct = np.count_nonzero(load == level)
    print(level, ct)

df['load'] = load
df['prev_load'] = prev_load

df = df.drop(columns=['free'])
df = df.drop(columns=['occupied'])
df = df.drop(columns=['total'])

df = df.drop(columns=['prev_free'])
df = df.drop(columns=['prev_occupied'])
df = df.drop(columns=['prev_total'])

print(df)
return df

def preprocess_data(df):
df_tmp = df

y = df_tmp['load']
X = df_tmp.drop(columns=['load'])

scale_input = config['scale_input']
if scale_input:
    sc = MinMaxScaler()

```

```

    X = sc.fit_transform(X)

    le = LabelEncoder()
    y = le.fit_transform(y)

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.20, random_state=42)

    return X_train, X_test, y_train, y_test

df = data_read()

#%%
# Separate data into clients based on the 'group' column
def create_clients(df, group_column='group'):
    clients = {}
    unique_groups = df[group_column].unique()
    for group in unique_groups:
        group_data = df[df[group_column] == group]
        clients[group] = group_data
    return clients

# Create clients
clients = create_clients(df)

#%%
# Create model
def create_model(input_shape):
    model = keras.Sequential()
    model.add(keras.layers.Dense(512*2, input_dim=input_shape,
activation='relu'))
    # model.add(keras.layers.Dense(512*2, input_dim=input_shape,
activation='relu'))
    model.add(keras.layers.Dense(1, kernel_initializer='normal',
activation="sigmoid")) # Changed softmax to sigmoid
    return model

# Define optimizer and compile the model
optimizer = tf.keras.optimizers.legacy.Adam(learning_rate=0.001)
loss = 'binary_crossentropy'
metrics = ['accuracy']

#%%
# Evaluate global model
def evaluate_global_model(global_model, X_test, y_test):
    y_pred = global_model.predict(X_test)
    y_pred = np.where(y_pred >= 0.5, 1, 0)
    print(y_pred)

acc = accuracy_score(y_test, y_pred)

```

```

print("accuracy=\n", acc)

cm = confusion_matrix(y_test, y_pred)
print("confusion matrix=\n", cm)

cm = confusion_matrix(y_test, y_pred, normalize='true')
print("normalized confusion matrix=\n", cm)

levels = len(config['levels'])-1
if levels==3:
    target_names = ['Low', 'Medium', 'High']
else:
    target_names = ['Low', 'High']

plot_cm = config['plot_cm']
if plot_cm:
    disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=target_names)
    disp.plot()
    plt.show()
    plt.savefig(os.path.join(os.getcwd()
+' /results_katerina/fed_test_2.png'))

report = classification_report(y_test, y_pred,
target_names=target_names)
print('classification report=\n', report)

def evaluate_epoch(global_model, X_test, y_test):
y_pred = global_model.predict(X_test)
y_pred = np.where(y_pred >= 0.5, 1, 0)
print(y_pred)
# evaluation metrics
acc = accuracy_score(y_test, y_pred)
print("accuracy=", acc)

cm = confusion_matrix(y_test, y_pred)
print("confusion matrix=\n", cm)

cm = confusion_matrix(y_test, y_pred, normalize='true')
print("normalized confusion matrix=\n", cm)

levels = len(config['levels'])-1
if levels==3:
    target_names = ['Low', 'Medium', 'High']
else:
    target_names = ['Low', 'High']

```

```

    report = classification_report(y_test, y_pred,
target_names=target_names)
    print('classification report=\n', report)

###
def train_federated_model(clients, epochs):
    global_model = create_model(input_shape=len(df.columns) - 1)
    client_weights_history = list()
    for epoch in range(epochs):
        print(f"Epoch {epoch+1}/{epochs}")
        for client, data in clients.items():
            X_train, X_test, y_train, y_test = preprocess_data(data)
            model = create_model(input_shape=len(df.columns) - 1)
            model.compile(optimizer=optimizer, loss=loss, metrics=metrics)
            model.set_weights(global_model.get_weights())
            model.fit(X_train, y_train, epochs=config['epochs'], verbose=0)
            client_weights_history.append(model.get_weights()) # Store the
weights of the client's model

        # At the end of each epoch, calculate the average weights of all
clients
        average_weights = list()
        for weights_list_tuple in zip(*client_weights_history):
            average_weights.append(
                np.array([np.array(w).mean(axis=0) for w in
zip(*weights_list_tuple)]))
        # Update the global model with the average weights
        global_model.set_weights(average_weights)

        # Clear weights before next epoch
        average_weights.clear()
        client_weights_history.clear()

        # Evaluate the eepoch
        X_train, X_test, y_train, y_test = preprocess_data(df)
        print("Evaluating epoch ...")
        evaluate_epoch(global_model, X_test, y_test)

    X_train_global, X_test_global, y_train_global, y_test_global =
preprocess_data(df)
    print("Evaluating global model...")
    evaluate_global_model(global_model, X_test_global, y_test_global)
###
# Train federated model
train_federated_model(clients, epochs=config['epochs'])

```

BIBΛΙΟΓΡΑΦΙΑ

- AutoML.org (no date) *AutoML*. Available at: <https://www.automl.org/automl/> (Accessed: 30 October 2023).
- Bardab, S.N., Ahmed, T.M. and Mohammed, T.A.A. (2020) *Data Mining Classification Algorithms: An overview*, Semantic Scholar. Available at: <https://www.semanticscholar.org/reader/742ea548cb3e6957022171284852cd576065767d> (Accessed: 19 October 2023).
- Bassetti, E. et al. (2022) *Exploiting user behavior to predict parking availability through machine learning*, MDPI. Available at: <https://www.mdpi.com/2624-6511/5/4/64> (Accessed: 31 October 2023).
- Brugger, B. (2022) *Adaptive boosting: A stepwise explanation of the algorithm*, Medium. Available at: <https://towardsdatascience.com/adaptive-boosting-a-stepwise-explanation-of-the-algorithm-50b75c3729c1> (Accessed: 05 February 2024).
- Camero, A. et al. (2019) *Evolutionary deep learning for car park occupancy prediction in smart ...*, ResearchGate. Available at: https://www.researchgate.net/profile/Andres-Camero/publication/330014757_Evolutionary_Deep_Learning_for_Car_Park_Occupancy_Prediction_in_Smart_Cities_12th_International_Conference_LION_12_Kalamata_Greece_June_10-15_2018_Revised_Selected_Papers/links/5c348a5492851c22a363c1a9/Evolutionary-Deep-Learning-for-Car-Park-Occupancy-Prediction-in-Smart-Cities-12th-International-Conference-LION-12-Kalamata-Greece-June-10-15-2018-Revised-Selected-Papers.pdf (Accessed: 18 October 2023).
- Ding, J. et al. (2022) *Federated Learning Challenges and Opportunities: An Outlook*, ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Singapore, pp. 8752-8756, DOI: 10.1109/ICASSP43922.2022.9746925.
- Eleni I. Vlahogianni, Konstantinos Kepaptsoglou, Vassileios Tsetos & Matthew G. Karlaftis (2016) *A Real-Time Parking Prediction System for Smart Cities*, Journal of Intelligent Transportation Systems, 20:2, 192-204, DOI: 10.1080/15472450.2015.1037955.
- Fan, J. et al. (2020) *Predicting Vacant Parking Space Availability: A Long Short-Term Memory Approach*, IEEE Xplore. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9495808> (Accessed: 18 October 2023).
- Feng, Y. et al. (2022) *Predicting vacant parking space availability zone-wisely: A hybrid deep learning approach - complex & intelligent systems*, SpringerLink. Available at: <https://link.springer.com/article/10.1007/s40747-022-00700-1> (Accessed: 17 October 2023).
- Gao, L. et al. (2023) *Prediction of vacant parking spaces in multiple parking lots: A DWT-CONVGRU-BRC model*, MDPI. Available at: <https://www.mdpi.com/2076-3417/13/6/3791> (Accessed: 18 October 2023).

- Ge, P. (2020) *Analysis on Approaches and Structures of Automated Machine Learning Frameworks*, IEEE Xplore. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9349624> (Accessed: 30 October 2023).
- Huang, X. et al. (2021) *FedParking: A Federated Learning Based Parking Space Estimation With Parked Vehicle Assisted Edge Computing*, IEEE Xplore. Available at: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9495808> (Accessed: 18 October 2023).
- IBM (no date) *What is deep learning?*. Available at: <https://www.ibm.com/topics/deep-learning> (Accessed: 27 October 2023).
- IBM (no date) *What is machine learning?*. Available at: <https://www.ibm.com/topics/machine-learning> (Accessed: 18 October 2023).
- Inam, S. et al. (2022) *Machine learning models for on-street parking prediction*, Encyclopedia. Available at: <https://encyclopedia.pub/entry/24490> (Accessed: 17 October 2023).
- Joshi, N. (2023) *How a real-time parking availability system benefits urban transportation*, Smart City Parking Management. Available at: <https://www.linkedin.com/pulse/how-real-time-parking-availability-system-benefits-urban-naveen-joshi/> (Accessed: 06 February 2024).
- Li, L. et al. (2020) *A review of applications in federated learning*, ScienceDirect.
- Loon, R.V. (2022) *Role of automl in assisting model development*, Simplilearn.com. Available at: <https://www.simplilearn.com/role-of-automl-in-assisting-model-development-article> (Accessed: 30 October 2023).
- Quercus (no date) *Current trends in the parking industry*. Available at: <https://quercus-technologies.com/news/current-trends-in-the-parking-industry> (Accessed: 06 February 2024).
- R, S.E. (2024) *Understand random forest algorithms with examples* (updated 2024), Analytics Vidhya. Available at: <https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/> (Accessed: 05 February 2024).
- Soofi, A.A. and Awan, A. (2017) *Classification Techniques in Machine Learning: Applications and Issues*, Semantic Scholar. Available at: <https://pdfs.semanticscholar.org/2678/36b4044b2e8d4c2a0b74a05d661b1efdfad.pdf> (Accessed: 19 October 2023).
- Simplilearn (2023) *Scikit-learn VS TENSORFLOW: A detailed comparison*: Simplilearn, Simplilearn.com. Available at: <https://www.simplilearn.com/scikit-learn-vs-tensorflow-article> (Accessed: 30 October 2023).
- Subasi, A. (2022) *Advanced pattern recognition tools for disease diagnosis*, Science Direct. Available at: <https://www.sciencedirect.com/science/article/abs/pii/B9780323905480000115> (Accessed: 05 February 2024).

