

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ



ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

Ανάλυση και Πρόβλεψη Ατμοσφαιρικών Ρύπων με Αλγορίθμους  
Χρονοσειρών: Μελέτη Περίπτωσης στα Δεδομένα OpenAQ

---

Τμήμα Μαθηματικών

Πρόγραμμα Μεταπτυχιακών Σπουδών

Υπολογιστική και Στατιστική Αναλυτική στην Επιστήμη των Δεδομένων

Συγγραφή:  
Σπυρίδων  
Μπουγιατάκης

Επιβλέπων:  
Σωτήριος  
Κωτσιαντής

Πάτρα, 2026



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ



ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

---

Ανάλυση και Πρόβλεψη Ατμοσφαιρικών Ρύπων με Αλγορίθμους  
Χρονοσειρών: Μελέτη Περίπτωσης στα Δεδομένα OpenAQ

---

Τμήμα Μαθηματικών

Πρόγραμμα Μεταπτυχιακών Σπουδών

Υπολογιστική και Στατιστική Αναλυτική στην Επιστήμη των Δεδομένων

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 22 Μαΐου 2026.

(υπογραφή)

(υπογραφή)

(υπογραφή)

.....

.....

.....

Σωτήριος Κωτσιαντής,  
Αναπληρωτής Καθηγητής

Όμηρος Ράγγος,  
Αναπληρωτής Καθηγητής

Δημήτριος Καββαδίας,  
Επίκουρος Καθηγητής



|                                                                          |    |
|--------------------------------------------------------------------------|----|
| Περιεχόμενα                                                              |    |
| Περίληψη .....                                                           | 3  |
| Κεφάλαιο 1: Εισαγωγή .....                                               | 4  |
| 1.1 Σκοπός και Αντικείμενο της Έρευνας .....                             | 4  |
| 1.2 Σημασία των Χρονοσειρών στην Ανάλυση Περιβαλλοντικών Δεδομένων ..... | 5  |
| 1.3 Δομή της Διπλωματικής Εργασίας .....                                 | 6  |
| Κεφάλαιο 2: Θεωρητικό Υπόβαθρο .....                                     | 7  |
| 2.1 Περιβαλλοντική Ρύπανση και Ατμοσφαιρικοί Ρύποι .....                 | 7  |
| 2.1.1 Ορισμός και Τύποι Ατμοσφαιρικών Ρύπων .....                        | 7  |
| 2.1.2 Επιπτώσεις στην Υγεία και στο Περιβάλλον .....                     | 8  |
| 2.2 Βιβλιογραφική Ανασκόπηση .....                                       | 10 |
| 2.3 Ανάλυση Χρονοσειρών .....                                            | 12 |
| 2.3.1 Ορισμός και Βασικές Έννοιες .....                                  | 12 |
| 2.3.2 Στασιμότητα και Αυτοσυσχέτιση .....                                | 13 |
| 2.3.3 Ολοκληρωμένα αυτοπαλίνδρομα μοντέλα κινητού μέσου (ARIMA) .....    | 14 |
| 2.4 Μοντέλα Μηχανικής Μάθησης .....                                      | 16 |
| 2.4.1 Πολυεπίπεδα τεχνητά νευρωνικά δίκτυα (MLP) .....                   | 17 |
| 2.4.2 Κ-Κοντινότεροι Γείτονες (KNN) .....                                | 20 |
| 2.4.3 Τυχαία Δάση (RF) .....                                             | 21 |
| 2.4.4 Gradient Boosting (GB) .....                                       | 22 |
| 2.4.5 Extreme Gradient Boosting (XGB) .....                              | 24 |
| 2.4.6 Light Gradient Boosting Machine (LGBM) .....                       | 26 |
| 2.5 Μοντέλα Βαθιάς Μάθησης .....                                         | 27 |
| 2.5.1 Long Short-Term Memory (LSTM) .....                                | 28 |
| 2.5.2 Gated Recurrent Unit (GRU) .....                                   | 31 |
| Κεφάλαιο 3: Δεδομένα OpenAQ .....                                        | 33 |
| 3.1 Περιγραφή και Δομή των Δεδομένων OpenAQ .....                        | 34 |

|                                                                              |    |
|------------------------------------------------------------------------------|----|
| 3.2 Διαδικασία Συλλογής Δεδομένων και Προβλήματα Ποιότητας .....             | 35 |
| Κεφάλαιο 4: Μεθοδολογία .....                                                | 36 |
| 4.1 Σύνολο Δεδομένων .....                                                   | 36 |
| 4.2 Προεπεξεργασία Δεδομένων.....                                            | 36 |
| 4.2.1 Επεξεργασία Ακραίων Τιμών.....                                         | 37 |
| 4.2.2 Ελλιπή Δεδομένα και Χειρισμός τους .....                               | 38 |
| 4.3 Επιλογή Αλγορίθμων και Δημιουργία Χαρακτηριστικών .....                  | 40 |
| 4.4 Κατασκευή και Εκπαίδευση Μοντέλων .....                                  | 41 |
| 4.4.1 Μπεϋζιανή Αναζήτηση και Διασταυρούμενη Επικύρωση για Χρονοσειρές ..... | 42 |
| 4.4.2 Μεθοδολογία ARIMA .....                                                | 43 |
| 4.5 Μέθοδοι Αξιολόγησης Προγνωστικών Μοντέλων .....                          | 44 |
| 4.6 Τεχνικές Ρυθμίσεις και Αναπαραγωγιμότητα της Πειραματικής Διαδικασίας .. | 45 |
| 4.7 Περιορισμοί της μελέτης .....                                            | 47 |
| Κεφάλαιο 5: Αποτελέσματα και Ανάλυση .....                                   | 49 |
| 5.1 Ανάλυση των Προγνωστικών Αποτελεσμάτων.....                              | 49 |
| 5.1.1 Προβλέψεις Σύντομης Περιόδου .....                                     | 49 |
| 5.1.2 Προβλέψεις Μακροχρόνιας Περιόδου.....                                  | 51 |
| 5.2 Στατιστική Αξιολόγηση .....                                              | 53 |
| 5.2.1 Προβλέψεις Σύντομης Περιόδου .....                                     | 54 |
| 5.2.2 Προβλέψεις Μακροχρόνιας Περιόδου.....                                  | 55 |
| 5.3 Συζήτηση και Ερμηνεία Αποτελεσμάτων .....                                | 57 |
| Κεφάλαιο 6: Συμπεράσματα και Μελλοντική Έρευνα.....                          | 59 |
| 6.1 Συμπεράσματα .....                                                       | 59 |
| 6.2 Προτάσεις για Μελλοντική Έρευνα .....                                    | 60 |
| Βιβλιογραφία .....                                                           | 61 |
| Παράρτημα.....                                                               | 64 |

## Περίληψη

Η παρούσα διπλωματική εργασία πραγματεύεται την ανάπτυξη και συγκριτική αξιολόγηση αλγορίθμων πρόβλεψης των ημερήσιων συγκεντρώσεων του ατμοσφαιρικού ρύπου PM<sub>2.5</sub> στην πόλη του Νέου Δελχί, με χρήση δεδομένων από την πλατφόρμα OpenAQ. Εξετάστηκαν αλγόριθμοι στατιστικής, μηχανικής μάθησης και βαθιάς μάθησης, συγκεκριμένα οι ARIMA, K-Nearest Neighbors, Random Forest, Gradient Boosting, XGBoost, LightGBM, MLP, LSTM και GRU, καθώς και δύο αλγόριθμοι αναφοράς, το αφελές μοντέλο τελευταίας τιμής και το εποχικά αφελές μοντέλο. Η αξιολόγηση πραγματοποιήθηκε για δύο χρονικούς ορίζοντες πρόβλεψης, δύο και επτά ημερών, με χρήση των μετρικών MAE, MSE, RMSE και  $R^2$ , ενώ για τη διερεύνηση της στατιστικής σημαντικότητας των διαφορών μεταξύ των αλγορίθμων εφαρμόστηκε το τεστ Diebold–Mariano.

Τα αποτελέσματα έδειξαν ότι για τον βραχυπρόθεσμο ορίζοντα των δύο ημερών δεν παρατηρούνται στατιστικά σημαντικές διαφορές στα σφάλματα μεταξύ του αφελούς μοντέλου τελευταίας τιμής (P-Naive), των αλγορίθμων μηχανικής και βαθιάς μάθησης. Το εύρημα αυτό αποδίδεται στην ισχυρή βραχυχρόνια αυτοσυσχέτιση των δεδομένων, όπου οι πρόσφατες τιμές της χρονοσειράς περιέχουν επαρκή πληροφορία για ακριβή πρόβλεψη σε πολύ μικρούς ορίζοντες. Λόγω της απλότητας και της συγκρίσιμης απόδοσής του, το αφελές μοντέλο τελευταίας τιμής αποτελεί την πλέον αποδοτική επιλογή για πρόβλεψη δύο ημερών. Για τον μακροπρόθεσμο ορίζοντα των επτά ημερών, οι αρχιτεκτονικές βαθιάς μάθησης LSTM και GRU αναδείχθηκαν ως οι πλέον αποδοτικές προσεγγίσεις, εμφανίζοντας στατιστικά σημαντικά καλύτερη ακρίβεια πρόβλεψης έναντι των περισσότερων αλγορίθμων μηχανικής μάθησης, λόγω της ικανότητάς τους να μοντελοποιούν μακροχρόνιες χρονικές εξαρτήσεις.

# Κεφάλαιο 1: Εισαγωγή

## 1.1 Σκοπός και Αντικείμενο της Έρευνας

Η ατμοσφαιρική ρύπανση αποτελεί ένα από τα σημαντικότερα περιβαλλοντικά προβλήματα των σύγχρονων κοινωνιών, με άμεσες και έμμεσες επιπτώσεις στην ανθρώπινη υγεία, στο οικοσύστημα και στην ποιότητα ζωής. Η συνεχώς αυξανόμενη αστικοποίηση, η αυξημένη κυκλοφορία οχημάτων, οι βιομηχανικές δραστηριότητες και οι κλιματικές μεταβολές συμβάλλουν στη διαμόρφωση σύνθετων ατμοσφαιρικών συνθηκών, οι οποίες ευνοούν την αύξηση των συγκεντρώσεων επιβλαβών ρύπων. Μεταξύ αυτών, τα αιωρούμενα σωματίδια PM<sub>2.5</sub> συγκαταλέγονται στους πλέον επικίνδυνους ρύπους, καθώς, λόγω του μικρού μεγέθους τους, δύνανται να εισέλθουν βαθιά στο αναπνευστικό σύστημα και να προκαλέσουν σοβαρές επιπτώσεις στην ανθρώπινη υγεία, ιδιαίτερα σε μεγάλες αστικές περιοχές με υψηλή πληθυσμιακή πυκνότητα και έντονη βιομηχανική δραστηριότητα.

Η δυνατότητα αξιόπιστης πρόβλεψης των επιπέδων ατμοσφαιρικής ρύπανσης καθίσταται, επομένως, ιδιαίτερα σημαντική, καθώς μπορεί να συμβάλει στην έγκαιρη λήψη μέτρων προστασίας, στην αποτελεσματικότερη ενημέρωση των πολιτών και στη διαμόρφωση κατάλληλων περιβαλλοντικών πολιτικών. Στο πλαίσιο αυτό, η ανάλυση χρονοσειρών σε συνδυασμό με τη χρήση μεθόδων στατιστικής, μηχανικής μάθησης και βαθιάς μάθησης προσφέρει αξιόπιστες δυνατότητες πρόβλεψης, συμβάλλοντας στην κατανόηση και διαχείριση του φαινομένου.

Σκοπός της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη και αξιολόγηση μοντέλων πρόβλεψης των συγκεντρώσεων του ατμοσφαιρικού ρύπου PM<sub>2.5</sub>, μέσω της αξιοποίησης μεθόδων στατιστικής, μηχανικής μάθησης και βαθιάς μάθησης. Στο πλαίσιο αυτό, επιδιώκεται η συγκριτική αξιολόγηση διαφορετικών αλγορίθμων σε διαφορετικούς χρονικούς ορίζοντες. Η μελέτη εστιάζει τόσο σε βραχυπρόθεσμες όσο και σε μακροπρόθεσμες προβλέψεις, με στόχο την κατανόηση των δυνατοτήτων και των περιορισμών των εξεταζόμενων μοντέλων.

Αντικείμενο της έρευνας αποτελεί η ανάλυση των συγκεντρώσεων PM<sub>2.5</sub> στην πόλη του Νέου Δελχί της Ινδίας, μία από τις πλέον επιβαρυνμένες περιοχές παγκοσμίως ως προς την ποιότητα του αέρα. Στο πλαίσιο της εργασίας, πραγματοποιείται η

επεξεργασία και ανάλυση των δεδομένων, η ανάπτυξη μοντέλων πρόβλεψης, καθώς και η αξιολόγηση της απόδοσής τους μέσω κατάλληλων μετρικών. Επιπλέον, επιχειρείται η σύγκριση των μοντέλων με στόχο την ανάδειξη των πιο αποτελεσματικών προσεγγίσεων για κάθε σενάριο πρόβλεψης.

## **1.2 Σημασία των Χρονοσειρών στην Ανάλυση Περιβαλλοντικών Δεδομένων**

Οι χρονοσειρές αποτελούν βασικό εργαλείο στην ανάλυση περιβαλλοντικών δεδομένων, καθώς επιτρέπουν τη μελέτη της εξέλιξης ενός φαινομένου στον χρόνο. Μέσω της διαχρονικής καταγραφής των δεδομένων, καθίσταται δυνατή η παρακολούθηση των μεταβολών τους και η κατανόηση της συμπεριφοράς τους.

Μια σταδιακή αύξηση των συγκεντρώσεων ρύπων σε βάθος χρόνου μπορεί να υποδηλώνει ανοδική τάση. Αντίστοιχα, η επαναλαμβανόμενη εμφάνιση υψηλών τιμών κατά τους χειμερινούς μήνες και χαμηλότερων κατά τους θερινούς αποτελεί χαρακτηριστικό παράδειγμα εποχικότητας. Όταν είναι γνωστό ότι ένα φαινόμενο παρουσιάζει συγκεκριμένη τάση ή επαναλαμβανόμενα μοτίβα, η πληροφορία αυτή μπορεί να αξιοποιηθεί για την πρόβλεψη των μελλοντικών τιμών του με μεγαλύτερη ακρίβεια.

Η δυνατότητα πρόβλεψης που παρέχει η ανάλυση χρονοσειρών συμβάλλει στην έγκαιρη ενημέρωση των πολιτών και στη λήψη κατάλληλων μέτρων από τις αρμόδιες αρχές, όπως ο περιορισμός της κυκλοφορίας οχημάτων ή η σύσταση προς ευπαθείς ομάδες να αποφεύγουν την έκθεση σε εξωτερικούς χώρους. Με τον τρόπο αυτό, μειώνονται οι αρνητικές επιπτώσεις στην ανθρώπινη υγεία και στο περιβάλλον, ενώ παράλληλα οι πολίτες μπορούν να προσαρμόζουν τη συμπεριφορά τους, περιορίζοντας δραστηριότητες που επιδεινώνουν την κατάσταση.

Ωστόσο, τα δεδομένα που καταγράφονται σε μορφή χρονοσειράς συχνά παρουσιάζουν ιδιαιτερότητες, όπως θόρυβο, ελλειπείς καταγραφές και απότομες μεταβολές. Οι παράγοντες αυτοί δυσχεραίνουν την ανάλυση και την πρόβλεψη, καθώς η συμπεριφορά των δεδομένων δεν είναι πάντα ομαλή ή προβλέψιμη. Για τον λόγο αυτό, απαιτείται η χρήση κατάλληλων μεθόδων ανάλυσης που μπορούν να διαχειριστούν την πολυπλοκότητα των δεδομένων και να αποδώσουν αξιόπιστα αποτελέσματα.

### 1.3 Δομή της Διπλωματικής Εργασίας

Η παρούσα διπλωματική εργασία οργανώνεται σε έξι κύρια κεφάλαια. Το Κεφάλαιο 1 αποτελεί την εισαγωγή και εισάγει τον αναγνώστη στο αντικείμενο της μελέτης, αναδεικνύοντας τη σημασία του προβλήματος και το πλαίσιο της έρευνας. Στη συνέχεια, στο Κεφάλαιο 2 αναπτύσσεται το θεωρητικό υπόβαθρο, όπου παρουσιάζονται βασικές έννοιες που σχετίζονται με την ατμοσφαιρική ρύπανση, την ανάλυση χρονοσειρών, καθώς και τα μοντέλα στατιστικής, μηχανικής και βαθιάς μάθησης. Ακολούθως, το Κεφάλαιο 3 επικεντρώνεται στα δεδομένα της μελέτης, παρουσιάζοντας τη δομή, την προέλευση και τη διαδικασία συλλογής τους, καθώς και τα ζητήματα ποιότητας που προκύπτουν. Έπειτα, στο Κεφάλαιο 4 παρουσιάζεται η μεθοδολογία που ακολουθήθηκε, όπου περιγράφονται η προεπεξεργασία των δεδομένων, η επιλογή των αλγορίθμων και η ανάπτυξη των προγνωστικών μοντέλων. Στο Κεφάλαιο 5 παρουσιάζονται τα αποτελέσματα των προβλέψεων, τόσο σε βραχυπρόθεσμο όσο και σε μακροπρόθεσμο ορίζοντα, και εξετάζεται η απόδοση των μοντέλων μέσω σύγκρισης και στατιστικής ανάλυσης των σφαλμάτων τους. Τέλος, στο Κεφάλαιο 6 συνοψίζονται τα βασικά συμπεράσματα της εργασίας και διατυπώνονται προτάσεις για μελλοντική έρευνα.

## Κεφάλαιο 2: Θεωρητικό Υπόβαθρο

### 2.1 Περιβαλλοντική Ρύπανση και Ατμοσφαιρικοί Ρύποι

Η περιβαλλοντική ρύπανση αποτελεί ένα από τα πιο σοβαρά προβλήματα της σύγχρονης εποχής, καθώς επηρεάζει ταυτόχρονα την υγεία του ανθρώπου, το περιβάλλον και τη βιωσιμότητα του πλανήτη. Με τον όρο περιβαλλοντική ρύπανση αναφερόμαστε στη ρύπανση της ατμόσφαιρας, των υδάτων και του εδάφους, καθώς και στην ηχορύπανση και τη φωτορύπανση. Όμως, στο πλαίσιο της εργασίας αυτής θα εστιάσουμε στους ατμοσφαιρικούς ρύπους, εξετάζοντας τις βασικές κατηγορίες τους, τις πηγές προέλευσής τους και τις επιπτώσεις τους στον άνθρωπο και το περιβάλλον.

#### 2.1.1 Ορισμός και Τύποι Ατμοσφαιρικών Ρύπων

Για την κατανόηση του φαινομένου της ατμοσφαιρικής ρύπανσης είναι απαραίτητος ο σαφής προσδιορισμός της έννοιας των ατμοσφαιρικών ρύπων. Ως ατμοσφαιρικοί ρύποι ορίζονται οι χημικές, φυσικές ή βιολογικές ουσίες και σωματίδια που βρίσκονται στον αέρα σε επίπεδα που τροποποιούν τα φυσικά χαρακτηριστικά της ατμόσφαιρας και μπορούν να έχουν επιπτώσεις στην υγεία των οργανισμών και στο περιβάλλον. Η παρουσία ατμοσφαιρικών ρύπων αποτελεί καθοριστικό παράγοντα για την υποβάθμιση της ποιότητας του αέρα, καθώς συνδέεται άμεσα τόσο με προβλήματα στη δημόσια υγεία όσο και με τη διατάραξη των οικοσυστημάτων. Στο πλαίσιο της προστασίας της δημόσιας υγείας, ο Παγκόσμιος Οργανισμός Υγείας εστιάζει στην αξιολόγηση συγκεκριμένων ατμοσφαιρικών ρύπων, για τους οποίους έχουν τεκμηριωθεί σημαντικές επιπτώσεις στην ανθρώπινη υγεία.

Σύμφωνα με τις κατευθυντήριες οδηγίες του ΠΟΥ για την ατμοσφαιρική ρύπανση (WHO Global Air Quality Guidelines, 2021), οι έξι βασικοί ατμοσφαιρικοί ρύποι που αξιολογούνται λόγω των τεκμηριωμένων επιπτώσεών τους στην ανθρώπινη υγεία είναι τα αιωρούμενα σωματίδια  $PM_{2.5}$  και  $PM_{10}$ , το διοξείδιο του αζώτου ( $NO_2$ ), το όζον ( $O_3$ ), το διοξείδιο του θείου ( $SO_2$ ) και το μονοξείδιο του άνθρακα ( $CO$ ). Για τους ρύπους αυτούς, ο Παγκόσμιος Οργανισμός Υγείας έχει θεσπίσει κατευθυντήριες τιμές (Air Quality Guidelines – AQG), οι οποίες προσδιορίζουν τα επίπεδα συγκέντρωσης που συνδέονται με τον χαμηλότερο δυνατό κίνδυνο για την ανθρώπινη υγεία. Πιο συγκεκριμένα, οι AQG ορίζουν για τα  $PM_{2.5}$  ετήσια μέση τιμή  $5 \mu g/m^3$  και

ημερήσια 15  $\mu\text{g}/\text{m}^3$ , για τα  $\text{PM}_{10}$  ετήσια 15  $\mu\text{g}/\text{m}^3$  και ημερήσια 45  $\mu\text{g}/\text{m}^3$ , για το  $\text{NO}_2$  ετήσια 10  $\mu\text{g}/\text{m}^3$  και ωριαία 25  $\mu\text{g}/\text{m}^3$ , για το  $\text{O}_3$  μέγιστη μέση τιμή οκταώρου 100  $\mu\text{g}/\text{m}^3$ , για το  $\text{SO}_2$  ημερήσια μέση τιμή 40  $\mu\text{g}/\text{m}^3$  και για το CO μέγιστη μέση τιμή 24ώρου 4  $\text{mg}/\text{m}^3$ . Στον Πίνακα 1 συνοψίζονται οι κατευθυντήριες και οι αντίστοιχοι ενδιάμεσοι στόχοι για τους ατμοσφαιρικούς ρύπους. Οι ενδιάμεσοι στόχοι είναι υψηλότεροι από τις τελικές τιμές AQG, αλλά χαμηλότεροι από τα επίπεδα συγκέντρωσης που παρατηρούνται συχνά σε περιοχές με σοβαρή ατμοσφαιρική ρύπανση, και λειτουργούν ως ρεαλιστικά και σταδιακά βήματα προόδου προς την επίτευξη των AQG, οι οποίες αποτελούν τον τελικό στόχο για τη μέγιστη προστασία της ανθρώπινης υγείας.

| Ρύπος                                          | Μέση τιμή       | Ενδιάμεσοι στόχοι |     |      |    | AQG |
|------------------------------------------------|-----------------|-------------------|-----|------|----|-----|
|                                                |                 | 1                 | 2   | 3    | 4  |     |
| $\text{PM}_{2.5}$ ( $\mu\text{g}/\text{m}^3$ ) | Ετήσια          | 35                | 25  | 15   | 10 | 5   |
|                                                | 24ώρου          | 75                | 50  | 37.5 | 25 | 15  |
| $\text{PM}_{10}$ ( $\mu\text{g}/\text{m}^3$ )  | Ετήσια          | 70                | 50  | 30   | 20 | 15  |
|                                                | 24ώρου          | 150               | 100 | 75   | 50 | 45  |
| $\text{O}_3$ ( $\mu\text{g}/\text{m}^3$ )      | Περίοδος αιχμής | 100               | 70  | -    | -  | 60  |
|                                                | 8ώρου           | 160               | 120 | -    | -  | 100 |
| $\text{NO}_2$ ( $\mu\text{g}/\text{m}^3$ )     | Ετήσια          | 40                | 30  | 20   | -  | 10  |
|                                                | 24ώρου          | 120               | 50  | -    | -  | 25  |
| $\text{SO}_2$ ( $\mu\text{g}/\text{m}^3$ )     | 24ώρου          | 125               | 50  | -    | -  | 40  |
| CO ( $\text{mg}/\text{m}^3$ )                  | 24ώρου          | 7                 | -   | -    | -  | 4   |

Πίνακας 1 Οι κατευθυντήριες και οι αντίστοιχοι ενδιάμεσοι στόχοι του ΠΟΥ για τους βασικούς ατμοσφαιρικούς ρύπους.

### 2.1.2 Επιπτώσεις στην Υγεία και στο Περιβάλλον

Η αυξανόμενη επιβάρυνση της ατμόσφαιρας από ρύπους έχει αναδειχθεί ως κρίσιμο ζήτημα δημόσιας υγείας και περιβαλλοντικής πολιτικής, δεδομένου ότι η έκθεση σε υψηλές συγκεντρώσεις ατμοσφαιρικών ρύπων συνδέεται με σοβαρά αναπνευστικά και καρδιαγγειακά νοσήματα, ενώ συμβάλλει και στην ενίσχυση φαινομένων όπως το φωτοχημικό νέφος και η κλιματική αλλαγή. Στο πλαίσιο αυτό, στη συνέχεια εξετάζονται οι έξι βασικοί ατμοσφαιρικοί ρύποι που θεωρούνται ιδιαίτερα επιβαρυντικοί για την ανθρώπινη υγεία.

Τα αιωρούμενα σωματίδια  $\text{PM}_{2.5}$  και  $\text{PM}_{10}$ , δηλαδή σωματίδια με αεροδυναμική διάμετρο μικρότερη από 2,5  $\mu\text{m}$  και 10  $\mu\text{m}$  αντίστοιχα, συνδέονται άμεσα με την εμφάνιση νοσημάτων καθώς και με αυξημένη θνησιμότητα. Ιδιαίτερη ανησυχία προκαλούν τα  $\text{PM}_{2.5}$ , λόγω του μικρού τους μεγέθους, το οποίο επιτρέπει τη βαθιά διείσδυσή τους στο αναπνευστικό σύστημα και τη συστηματική τους επίδραση στον

οργανισμό. Εκτιμάται ότι η μείωση των επιπέδων  $PM_{2.5}$  από  $35 \mu g/m^3$  σε  $10 \mu g/m^3$  θα μπορούσε να επιφέρει μείωση της τάξης του 15% στους θανάτους που σχετίζονται με την ατμοσφαιρική ρύπανση. Σε παγκόσμια κλίμακα, τα  $PM_{2.5}$  κατατάσσονται ως ο πέμπτος σημαντικότερος παράγοντας κινδύνου θνησιμότητας, καθώς η έκθεση σε αυτά εκτιμάται ότι ευθύνεται για το 7,6% του συνόλου των θανάτων. Επιπλέον, σύμφωνα με σχετικές μελέτες, ποσοστό που υπερβαίνει το 70% των θανάτων από SARS-CoV-2 στην Ιταλία έχει συσχετιστεί με τη μακροχρόνια έκθεση στην ατμοσφαιρική ρύπανση και στα  $PM_{2.5}$  (Cazzolla Gatti et al., 2020).

Το όζον  $O_3$  συναντάται τόσο στα ανώτερα στρώματα της ατμόσφαιρας (στρατόσφαιρα) όσο και κοντά στην επιφάνεια της Γης. Το στρατοσφαιρικό όζον είναι ωφέλιμο, καθώς λειτουργεί ως προστατευτικό φίλτρο έναντι της υπερϊώδους ακτινοβολίας. Αντίθετα, στο επίπεδο του εδάφους και στην τροπόσφαιρα, το όζον αποτελεί δευτερογενή ατμοσφαιρικό ρύπο. Σχηματίζεται μέσω σύνθετων φωτοχημικών διεργασιών που περιλαμβάνουν την ηλιακή ακτινοβολία και πρόδρομες ουσίες του όζοντος. Η έκθεση σε αυξημένες συγκεντρώσεις όζοντος ( $O_3$ ) έχει συνδεθεί με ερεθισμό και φλεγμονή των αεραγωγών, συνοδευόμενα από μείωση της πνευμονικής λειτουργίας και εμφάνιση συμπτωμάτων όπως βήχας, δύσπνοια και συριγμός. Επιπλέον, έχει παρατηρηθεί επιδείνωση του άσθματος και αύξηση των επισκέψεων σε υπηρεσίες υγείας ή των νοσοκομειακών εισαγωγών για αναπνευστικά προβλήματα, ιδιαίτερα σε ευάλωτες ομάδες πληθυσμού, όπως τα παιδιά, οι ηλικιωμένοι και τα άτομα με προϋπάρχοντα αναπνευστικά νοσήματα. Πέραν των άμεσων αυτών επιπτώσεων, η μακροχρόνια έκθεση στο όζον φαίνεται να συνδέεται με αυξημένη θνησιμότητα από αναπνευστικά αίτια (Jerrett et al., 2009; Kasdagli et al., 2024).

Το διοξείδιο του αζώτου ( $NO_2$ ) και το διοξείδιο του θείου ( $SO_2$ ) παράγονται κυρίως από την καύση ορυκτών καυσίμων, ιδιαίτερα σε σταθμούς ηλεκτροπαραγωγής και στον τομέα των μεταφορών. Οι οδικές μεταφορές αποτελούν σημαντική πηγή εκπομπών οξειδίων του αζώτου ( $NO_x$ ), καθώς το 2017 ευθύνονταν για το 39% των συνολικών εκπομπών  $NO_x$  στην Ευρώπη. Τα αέρια αυτά αποτελούν βασικούς πρόδρομους ρύπους για τον σχηματισμό όξινης βροχής και συμβάλλουν επίσης στη δημιουργία φωτοχημικού νέφους. Παράλληλα, επιδρούν αρνητικά στην ανθρώπινη υγεία, στη βλάστηση και στα δομικά υλικά.

Το μονοξείδιο του άνθρακα (CO) αποτελεί ένα ιδιαίτερα τοξικό αέριο, η εισπνοή του οποίου μπορεί να αποβεί θανατηφόρα, καθώς ευθύνεται για μεγάλο αριθμό περιστατικών δηλητηρίασης παγκοσμίως και συνιστά την κυριότερη αιτία θανάτων από δηλητηρίαση στις Ηνωμένες Πολιτείες. Σχηματίζεται κατά την ατελή καύση υδρογονανθράκων και γενικότερα καυσίμων που περιέχουν άνθρακα, όταν η παροχή οξυγόνου δεν επαρκεί για την πλήρη μετατροπή του άνθρακα σε διοξείδιο του άνθρακα (CO<sub>2</sub>). Χαρακτηριστικές πηγές εκπομπής του αποτελούν τα καυσαέρια των οχημάτων, οι σταθμοί ηλεκτροπαραγωγής, οι εγκαταστάσεις καύσης αποβλήτων και οι οικιακές θερμαντικές συσκευές υπό συνθήκες κακής λειτουργίας ή ανεπαρκούς αερισμού. Πρόκειται για άχρωμο, άοσμο και άγευστο αέριο, χωρίς ερεθιστικές ιδιότητες, γεγονός που καθιστά τη δηλητηρίαση από αυτό συχνά δύσκολη στη διάγνωση, καθώς τα συμπτώματά της μπορούν εύκολα να αποδοθούν σε άλλες, συνηθισμένες παθήσεις (Essamlali et al., 2024).

## 2.2 Βιβλιογραφική Ανασκόπηση

Η πρόβλεψη των συγκεντρώσεων PM<sub>2.5</sub> αποτελεί ένα από τα πλέον δυναμικά ερευνητικά πεδία στην επιστήμη της ατμοσφαιρικής ρύπανσης, με τον αριθμό των σχετικών δημοσιεύσεων να αυξάνεται ραγδαία τα τελευταία χρόνια. Η ανάπτυξη αξιόπιστων μοντέλων πρόβλεψης αποκτά ιδιαίτερη σημασία, καθώς τα αιωρούμενα σωματίδια PM<sub>2.5</sub> συνδέονται άμεσα με σοβαρά προβλήματα δημόσιας υγείας και περιβαλλοντικής υποβάθμισης. Για την αντιμετώπιση αυτής της ανάγκης, η επιστημονική κοινότητα έχει αναπτύξει ένα ευρύ φάσμα μεθοδολογικών προσεγγίσεων, οι οποίες εξελίσσονται σταδιακά από τις παραδοσιακές στατιστικές μεθόδους προς τις σύγχρονες τεχνικές μηχανικής και βαθιάς μάθησης.

Οι πρώιμες ερευνητικές προσπάθειες στηρίχθηκαν κυρίως σε κλασικές στατιστικές μεθόδους, όπως η γραμμική παλινδρόμηση και τα μοντέλα χρονοσειρών τύπου ARIMA. Οι μέθοδοι αυτές, αν και απλές στην εφαρμογή τους, κατόρθωσαν να αποτυπώσουν βασικές τάσεις στα δεδομένα ποιότητας αέρα. Ωστόσο, παρουσίαζαν σαφείς περιορισμούς στη σύλληψη μη γραμμικών σχέσεων και μακροπρόθεσμων εξαρτήσεων, ιδίως σε περιβάλλοντα με έντονη χωροχρονική μεταβλητότητα. Η ανάγκη για ακριβέστερα μοντέλα οδήγησε στην υιοθέτηση αλγόριθμων μηχανικής μάθησης, οι οποίοι κυριάρχησαν στη βιβλιογραφία κατά την επόμενη περίοδο (Wu et al., 2025).

Μεταξύ των αλγόριθμων μηχανικής μάθησης, τα ensemble μοντέλα όπως ο Random Forest (RF) και οι διάφορες παραλλαγές του Gradient Boosting έχουν καταγράψει ιδιαίτερα ενθαρρυντικά αποτελέσματα. Οι Vignesh et al., 2023 αξιολόγησαν εννέα διαφορετικούς αλγόριθμους, μεταξύ των οποίων LR, DT, GBR, AdaBoost, XGBoost, KNN, RF και SVR, για την πρόβλεψη PM2.5 σε σταθμούς μέτρησης σε ολόκληρες τις ΗΠΑ, αναδεικνύοντας τον RF και τον SVR ως τα πλέον αποδοτικά μοντέλα με  $R^2$  έως 0.899. Στο ίδιο πνεύμα, οι Makhdoomi et al., 2025 συνέκριναν LightGBM, XGBoost, RF και Gradient Boosting για την πόλη Μασχάντ του Ιράν, διαπιστώνοντας υψηλή απόδοση και για τους τέσσερις αλγόριθμους με  $R^2$  άνω του 96%, ενώ ο GBR υπερέτρησε ελαφρώς με χαμηλότερο RMSE ίσο με 2.31. Αξίζει να σημειωθεί ότι η απόδοση των μοντέλων μηχανικής μάθησης παρουσιάζει σημαντική γεωγραφική μεταβλητότητα ανάλογα με τα κλιματικά και αστικά χαρακτηριστικά κάθε περιοχής, εύρημα που επισημαίνεται από τους Vignesh et al., 2023.

Παράλληλα με την εξέλιξη των ensemble μεθόδων, τα μοντέλα βαθιάς μάθησης αναδείχθηκαν ως η πλέον υποσχόμενη κατηγορία, ιδίως για δεδομένα υψηλής χρονικής ανάλυσης. Το Long Short-Term Memory (LSTM), χάρη στην ικανότητά του να αποτυπώνει μακροπρόθεσμες χρονικές εξαρτήσεις, κατέγραψε επανειλημμένα την υψηλότερη απόδοση στη βιβλιογραφία. Οι Masood & Ahmad, 2023 ανέδειξαν το LSTM ως το ακριβέστερο μοντέλο για πρόβλεψη PM2.5 στο Δελχί ( $R^2=0.945$ ,  $RMSE=21.510 \mu g/m^3$ ), υπερτερώντας έναντι των MLFFNN, SVM και RF, ενώ η ανάλυση ευαισθησίας που πραγματοποιήσαν κατέδειξε ότι το PM10, η ταχύτητα ανέμου, η NH3 και το βενζόλιο αποτελούν τις πλέον επιδραστικές μεταβλητές εισόδου. Οι Alawi et al., 2024 επίσης συνέκριναν τους αλγόριθμους Bi-LSTM, GRU, XGBoost και MLP, επιβεβαιώνοντας την υπεροχή του LSTM με  $R^2=0.973$  για ωριαία πρόβλεψη PM2.5 κοντά στο Δελχί, ενώ το Bi-LSTM και το GRU ακολούθησαν με οριακά χαμηλότερες επιδόσεις ( $R^2=0.971$  και  $0.969$  αντίστοιχα).

Ένα από τα σημαντικότερα ευρήματα που αναδύεται συστηματικά από τη βιβλιογραφία είναι η επίδραση του χρονικού ορίζοντα πρόβλεψης στη σχετική απόδοση των μοντέλων. Οι Arson et al., 2021, συγκρίνοντας LSTM, CNN και ARIMA για πρόβλεψη PM10 σε τρεις χρονικούς ορίζοντες 6, 12 και 24 ωρών στη Σκόπια της Βόρειας Μακεδονίας, διαπίστωσαν ότι η απόδοση όλων των μοντέλων μειωνόταν με την αύξηση του ορίζοντα, με τα μοντέλα βαθιάς μάθησης να υπερτερούν σταθερά

έναντι του ARIMA. Το εύρημα αυτό επιβεβαιώνεται και από τους (Abuouelezz et al., 2025), οι οποίοι αξιολόγησαν έξι μοντέλα, DT, RF, SVR, CNN, LSTM και Facebook Prophet, για τρεις ορίζοντες πρόβλεψης σε σταθμούς του Αμπού Ντάμπι. Τα αποτελέσματά τους αποκάλυψαν ότι δεν υπάρχει ένα μοντέλο που να υπερτερεί σε όλους τους ορίζοντες: ο SVR αποδείχθηκε βέλτιστος για βραχυπρόθεσμες προβλέψεις PM2.5 (MAPE=18.7%), ενώ το Facebook Prophet αναδείχθηκε ως το πιο αποτελεσματικό για ημερήσιους και εβδομαδιαίους ορίζοντες (MAPE=21.8% και 21.3% αντίστοιχα).

Συνολικά, η βιβλιογραφική επισκόπηση αναδεικνύει μια σαφή τάση: η επιλογή του βέλτιστου μοντέλου πρόβλεψης PM2.5 δεν είναι καθολική, αλλά εξαρτάται από παράγοντες όπως ο χρονικός ορίζοντας πρόβλεψης, η χωρική κλίμακα, η διαθεσιμότητα και η ποιότητα των δεδομένων και τα ιδιαίτερα κλιματικά και αστικά χαρακτηριστικά κάθε περιοχής. Τα μοντέλα βαθιάς μάθησης, και ιδίως το LSTM και οι παραλλαγές του, παρουσιάζουν γενικά ανώτερη απόδοση για δεδομένα υψηλής χρονικής ανάλυσης και μακροπρόθεσμες εξαρτήσεις, ενώ αλγόριθμοι όπως ο SVR και τα ensemble μοντέλα παραμένουν ανταγωνιστικοί, ιδίως σε βραχυπρόθεσμες προβλέψεις ή όταν τα διαθέσιμα δεδομένα είναι περιορισμένα.

## 2.3 Ανάλυση Χρονοσειρών

Η ανάλυση χρονοσειρών αποτελεί κλάδο της στατιστικής που ασχολείται με φαινόμενα τα οποία εξελίσσονται δυναμικά στον χρόνο. Τα εν λόγω φαινόμενα παρουσιάζουν συχνά τάσεις, περιοδικότητα αλλά και συσχετίσεις με προηγούμενες καταστάσεις τους. Χαρακτηριστικά παραδείγματα τέτοιων φαινομένων συναντώνται, μεταξύ άλλων, σε περιβαλλοντικά δεδομένα (θερμοκρασία, βροχόπτωση), οικονομικούς δείκτες (έσοδα και έξοδα μιας επιχείρησης, ποσοστό ανεργίας) και συστήματα μεταφοράς (αριθμός επιβατών, κυκλοφοριακή ροή).

### 2.3.1 Ορισμός και Βασικές Έννοιες

Τυπικά, μια χρονοσειρά ορίζεται ως μια ακολουθία παρατηρήσεων  $\{y_t\}_{t=1}^T$  του υπό μελέτη φαινομένου, καταγεγραμμένων σε διακριτές και ισαπέχουσες χρονικές στιγμές. Με βάση τον συμβολισμό αυτό, το  $y_t$  αντιστοιχεί στην παρατηρούμενη τιμή στη χρονική στιγμή  $t$  και το  $T$  στο πλήθος των παρατηρήσεων. Στο πλαίσιο της θεωρητικής ανάλυσης, το υπό μελέτη φαινόμενο περιγράφεται πιθανοθεωρητικά μέσω των στοχαστικών διαδικασιών, δηλαδή από την ακολουθία τυχαίων μεταβλητών

$\{Y_t\}_{t \in \mathbb{Z}}$ . Συνεπώς, η παρατηρούμενη χρονοσειρά  $\{y_t\}_{t=1}^T$  αποτελεί μία πεπερασμένη πραγματοποίηση αυτής.

Μια χρονοσειρά συχνά εμφανίζει δομικά χαρακτηριστικά που περιγράφουν τη συστηματική της συμπεριφορά στον χρόνο, με σημαντικότερα την τάση και την εποχικότητα. Τάση μιας χρονοσειράς ονομάζεται η μακροχρόνια και συστηματική μεταβολή της μέσης τιμής του υπό μελέτη φαινομένου στον χρόνο. Η τάση αποτυπώνει τη γενική κατεύθυνση της χρονοσειράς (αύξουσα, φθίνουσα ή σταθερή), αγνοώντας τις βραχυχρόνιες και τυχαίες διακυμάνσεις. Εποχικότητα μιας χρονοσειράς ονομάζεται η περιοδική μεταβολή της συμπεριφοράς της, η οποία εμφανίζεται σε σταθερά χρονικά διαστήματα. Η εποχικότητα συνδέεται συνήθως με ημερολογιακούς ή θεσμικούς παράγοντες και επαναλαμβάνεται με παρόμοιο μοτίβο σε κάθε περίοδο. Η παρουσία τάσης και εποχικότητας επηρεάζει καθοριστικά τις στατιστικές ιδιότητες μιας χρονοσειράς και συνδέεται άμεσα με την έννοια της στασιμότητας.

### 2.3.2 Στασιμότητα και Αυτοσυσχέτιση

Στην ανάλυση χρονοσειρών, ιδιαίτερη σημασία αποδίδεται στη στασιμότητα, καθώς αυτή καθορίζει τη συμπεριφορά των στατιστικών ιδιοτήτων της σειράς στον χρόνο. Η έννοια της στασιμότητας αφορά την από κοινού κατανομή των τυχαίων μεταβλητών της στοχαστικής διαδικασίας  $\{Y_t\}_{t \in \mathbb{Z}}$  που διέπουν το φαινόμενο. Πιο συγκεκριμένα, η στασιμότητα περιγράφει την κατάσταση εκείνη κατά την οποία οι από κοινού κατανομές των  $\{Y_{t_1}, \dots, Y_{t_k}\}$  είναι ίδιες με τις από κοινού κατανομές των  $\{Y_{t_1+h}, \dots, Y_{t_k+h}\}$ , για κάθε  $k \in \mathbb{N}$ , κάθε επιλογή χρονικών στιγμών  $t_1, \dots, t_k$  και κάθε χρονική μετατόπιση  $h \in \mathbb{Z}$ . Αυτό μεταφράζεται ως ότι οι αντίστοιχες παρατηρούμενες χρονοσειρές δεν παρουσιάζουν σημαντικές διαφοροποιήσεις σε διαφορετικά χρονικά διαστήματα. Η έννοια της στασιμότητας διακρίνεται σε αυστηρή και ασθενή. Στη συνέχεια παρατίθενται οι ορισμοί κάθε είδους στασιμότητας.

**Ορισμός:** Μία στοχαστική διαδικασία λέγεται **αυστηρά στάσιμη**, αν οι από κοινού κατανομές των  $(Y_{t_1}, \dots, Y_{t_k})$  είναι ίδιες με τις κατανομές των  $(Y_{t_1+h}, \dots, Y_{t_k+h})$ , για κάθε  $k$ , κάθε επιλογή χρονικών στιγμών και κάθε μετατόπιση  $h$ .

**Ορισμός:** Μία στοχαστική διαδικασία λέγεται **ασθενώς στάσιμη**, αν οι ροπές πρώτης και δεύτερης τάξης αυτής είναι σταθερές στον χρόνο, δηλαδή αν

$$\text{η μέση τιμή είναι σταθερή } \mu_t = E(Y_t) = \mu, \forall t \in \mathbb{Z}$$

η αυτοσυνδιακύμανση για  $t_1 = t$  και  $t_2 = t - \tau$  εξαρτάται μόνο από την υστέρηση  $\tau$  και όχι από το χρόνο  $t$ , δηλαδή

$$\gamma(t_1, t_2) = \text{Cov}(Y_{t_1}, Y_{t_2}) = E \left[ (Y_{t_1} - E(Y_{t_1})) (Y_{t_2} - E(Y_{t_2})) \right] = E(Y_{t_1} Y_{t_2}) - \mu^2$$

$$\gamma(t, t - \tau) = \gamma(\tau), \forall t \in Z$$

**Πόρισμα:** Σε μία ασθενώς στάσιμη στοχαστική διαδικασία η διασπορά είναι σταθερή και ορίζεται από τη σχέση  $\sigma_Y^2 = \gamma(0) = \kappa(0) - \mu^2$ , όπου η ροπή δεύτερης τάξης  $\kappa(\tau) = \kappa(t, t - \tau) = E(Y_t Y_{t-\tau})$ , είναι σταθερή ως προς το χρόνο  $t$ .

Επομένως, η αυτοσυνδιακύμανση μιας ασθενούς στάσιμης στοχαστικής διαδικασίας εκφράζει τη συνδιασπορά μεταξύ δύο χρονικών στιγμών και μπορεί να πάρει αρνητικές ή θετικές τιμές, εκφράζοντας την αρνητική ή τη θετική συσχέτισή τους αντίστοιχα (Οικονόμου & Μαλεφάκη, 2023).

Μία ποσότητα που είναι στενά συνδεδεμένη με την έννοια αυτοσυνδιακύμανσης είναι η έννοια της αυτοσυσχέτισης. Η αυτοσυσχέτιση μιας ασθενούς στάσιμης στοχαστικής διαδικασίας για υστέρηση  $\tau$  ορίζεται από τη σχέση

$$\rho(\tau) = \frac{\gamma(\tau)}{\gamma(0)}$$

και αποτελεί έναν καθαρό αριθμό ανάλογο με τον συντελεστή συσχέτισης μεταξύ δύο τυχαίων μεταβλητών  $X$  και  $Y$ , ο οποίος μας επιτρέπει να αξιολογήσουμε το είδος και την ισχύ της συσχέτισης μεταξύ των μεταβλητών που βρίσκονται σε υστέρηση  $\tau$  και αποτελεί μέτρο της «μνήμης» της διαδικασίας. Τέλος αξίζει να σημειωθεί ότι η αυτοσυσχέτιση ικανοποιεί τις παρακάτω ιδιότητες:

- i.  $|\rho_\tau| \leq 1$
- ii.  $\rho_\tau = \rho_{-\tau}$

### 2.3.3 Ολοκληρωμένα αυτοπαλίνδρομα μοντέλα κινητού μέσου (ARIMA)

Τα ολοκληρωμένα αυτοπαλίνδρομα μοντέλα κινητού μέσου ή μοντέλα ARIMA (Autoregressive Integrated Moving Average), αποτελούν μία ιδιαίτερα χρήσιμη οικογένεια στατιστικών μοντέλων, για την ανάλυση ασθενών στάσιμων χρονοσειρών. Η έννοια της στασιμότητας αναφέρεται σε χρονοσειρές των οποίων οι στατιστικές ιδιότητες, δηλαδή η μέση τιμή και η διακύμανση, δεν μεταβάλλονται με τον χρόνο.

Συνεπώς, χρονοσειρές με τάση ή εποχικότητα δεν είναι στάσιμες, καθώς η παρουσία αυτών των δομικών στοιχείων οδηγεί σε μεταβολή των τιμών της χρονοσειράς με την πάροδο του χρόνου. Ως μέσο αντιμετώπισης της μη στασιμότητας μιας χρονοσειράς, εισάγεται η έννοια των διαφορών.

**Διαφορά  $I(d)$**  τάξης  $d$  αντιστοιχεί στον αριθμό εφαρμογών του τελεστή πρώτης διαφοράς και σχετίζεται με την απομάκρυνση στοχαστικής τάσης ή παρουσίας μοναδιαίας ρίζας. Η διαφορίση ορίζεται ως ο επαναληπτικός υπολογισμός διαφορών της χρονοσειράς και αποτελεί βασικό μετασχηματισμό για την επίτευξη στασιμότητας. Αντίστοιχα, η εποχικότητα αντιμετωπίζεται μέσω εποχικής διαφοροποίησης, η οποία ορίζεται ως η διαφορά μεταξύ παρατηρήσεων που απέχουν μία πλήρη εποχική περίοδο  $s$ , οδηγώντας στην εξάλειψη των περιοδικών δομών της χρονοσειράς. Το  $\nabla$  συμβολίζει τον τελεστή διαφοράς και ορίζεται ως  $\nabla Y_t = Y_t - Y_{t-1}$ , ενώ ο τελεστής οπισθοδρόμησης  $B$  μετατοπίζει τη χρονοσειρά κατά μία χρονική μονάδα προς το παρελθόν, δηλαδή  $BY_t = Y_{t-1}$ .

Διαφορά 1<sup>ης</sup> τάξης:  $\nabla Y_t = Y_t - Y_{t-1} = (1 - B)Y_t$

Διαφορά 2<sup>ης</sup> τάξης:  $\nabla^2 Y_t = (1 - B)^2 Y_t = (1 - 2B + B^2)Y_t = Y_t - 2Y_{t-1} + Y_{t-2}$

Διαφορά τάξης  $d$ :  $\nabla^d Y_t = (1 - B)^d Y_t$

Διαφορά υστέρησης  $s$ :  $\nabla_s Y_t = Y_t - Y_{t-s} = (1 - B^s)Y_t$

Συνδυασμός απλής και εποχικής διαφορίσης:  $\nabla^d \nabla_s Y_t = (1 - B)^d (1 - B^s)Y_t$

**Αυτοπαλίνδρομη διαδικασία  $AR(p)$**  ονομάζεται μια στοχαστική διαδικασία της οποίας η τρέχουσα τιμή εκφράζεται ως γραμμικός συνδυασμός πεπερασμένου αριθμού παρελθοντικών τιμών της ίδιας της διαδικασίας.

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + \varepsilon_t \Leftrightarrow Y_t = \sum_{i=1}^p \phi_i Y_{t-i} + \varepsilon_t \Leftrightarrow \Phi(B)Y_t = \varepsilon_t$$

όπου  $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p$  και  $\varepsilon_t$  είναι μια διαδικασία λευκού θορύβου. Μία στοχαστική διαδικασία  $\varepsilon_t$  ονομάζεται λευκός θόρυβος  $\varepsilon_t \sim WN(0, \sigma^2)$  όταν  $E(\varepsilon_t) = 0$ ,  $Var(\varepsilon_t) = \sigma^2$  και  $Cov(\varepsilon_t, \varepsilon_{t-\tau}) = 0$ , για κάθε  $\tau \neq 0$ .

**Διαδικασία κινούμενου μέσου  $MA(q)$**  ονομάζεται μια στοχαστική διαδικασία της οποίας η τρέχουσα τιμή εκφράζεται ως γραμμικός συνδυασμός του τρέχοντος και των  $q$  προηγούμενων στοχαστικών διαταραχών.

$$Y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q} \Leftrightarrow Y_t = \varepsilon_t + \sum_{i=1}^q \theta_i \varepsilon_{t-i} \Leftrightarrow Y_t = \Theta(B) \varepsilon_t$$

όπου  $\Theta(B) = 1 + \theta_1 B + \theta_2 B^2 + \dots + \theta_q B^q$ .

**Μοντέλο αυτοπαλίνδρομου κινούμενου μέσου  $ARMA(p, q)$**  ονομάζεται ένα στοχαστικό μοντέλο στο οποίο η τρέχουσα τιμή της χρονοσειράς εκφράζεται ως συνδυασμός ενός αυτοπαλίνδρομου μέρους τάξης  $p$  και ενός μέρους κινούμενου μέσου τάξης  $q$ .

$$\Phi(B)Y_t = \Theta(B)\varepsilon_t$$

**Ολοκληρωμένο αυτοπαλίνδρομο μοντέλο κινητού μέσου  $ARIMA(p, d, q)$**  ονομάζεται ένα στοχαστικό μοντέλο στο οποίο η χρονοσειρά, αφού υποστεί διαφοροποίηση τάξης  $d$ , ακολουθεί ένα μοντέλο  $ARMA(p, q)$  (Hyndman & Athanasopoulos, 2021)

$$\Phi(B)(1 - B)^d Y_t = \Theta(B)\varepsilon_t$$

## 2.4 Μοντέλα Μηχανικής Μάθησης

Η πρόβλεψη χρονοσειρών μπορεί να διατυπωθεί είτε μέσω στοχαστικών γραμμικών μοντέλων, όπως τα  $ARIMA$ , είτε μέσω εκτίμησης γενικών συναρτήσεων παλινδρόμησης στο πλαίσιο της επιβλεπόμενης μηχανικής μάθησης. Δηλαδή, υπό κατάλληλη αναπαράσταση των δεδομένων μέσω χαρακτηριστικών, η πρόβλεψη χρονοσειρών ανάγεται σε πρόβλημα επιβλεπόμενης μηχανικής μάθησης, όπου οι τιμές-στόχοι είναι οι τιμές της χρονοσειράς.

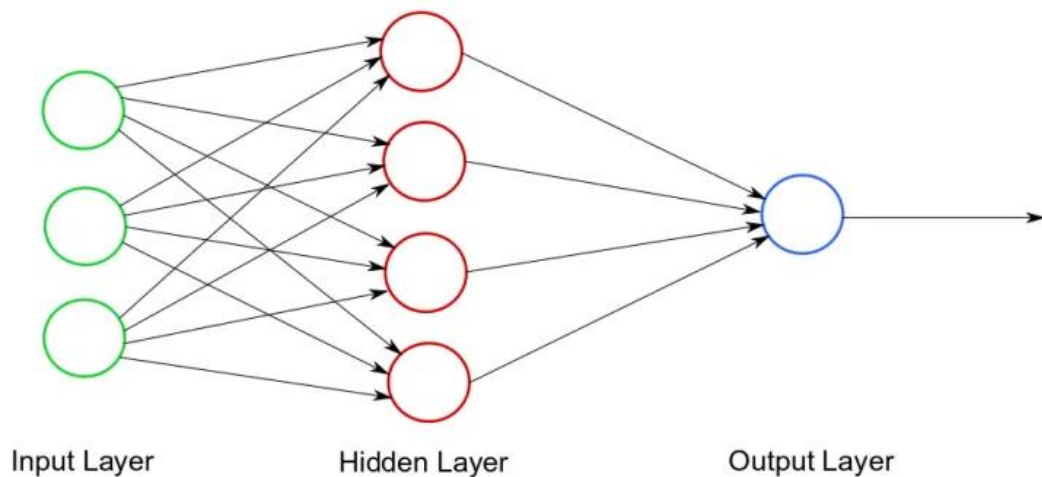
Για την εφαρμογή μεθόδων επιβλεπόμενης μάθησης σε χρονοσειρές, απαιτείται η κατάλληλη μετατροπή της χρονοσειράς σε ζεύγη χαρακτηριστικών και τιμών-στόχων. Η πιο διαδεδομένη προσέγγιση είναι η χρήση ενός παραθύρου ολίσθησης (sliding window), όπου οι προηγούμενες τιμές της χρονοσειράς χρησιμοποιούνται ως χαρακτηριστικά για την πρόβλεψη μελλοντικών τιμών. Το μέγεθος του παραθύρου

καθορίζει πόσες προηγούμενες χρονικές στιγμές λαμβάνονται υπόψη και αποτελεί υπερπαραμέτρο που επηρεάζει σημαντικά την απόδοση του μοντέλου.

Έστω το σύνολο εκπαίδευσης  $T = \{(x_i, y_i)\}_{i=1}^N$  με  $N$  παρατηρήσεις, όπου  $x_i = (x_{i1}, x_{i2}, \dots, x_{id}) \in R^d$  το  $i$ -οστό διάνυσμα χαρακτηριστικών, με  $d$  το πλήθος των χαρακτηριστικών και  $y_i \in R$  η αντίστοιχη τιμή-στόχος.

#### 2.4.1 Πολυεπίπεδα τεχνητά νευρωνικά δίκτυα (MLP)

Τα πολυεπίπεδα τεχνητά νευρωνικά δίκτυα (Multilayer Perceptron, MLP) αποτελούν μία από τις βασικότερες αρχιτεκτονικές τεχνητών νευρωνικών δικτύων και συνιστούν φυσική εξέλιξη του απλού perceptron. Οι νευρώνες είναι οργανωμένοι σε επίπεδα, τα οποία διακρίνονται σε ένα επίπεδο εισόδου, ένα ή περισσότερα κρυφά επίπεδα και ένα επίπεδο εξόδου. Επιπλέον, υπάρχει πλήρης σύνδεση μεταξύ νευρώνων διαδοχικών επιπέδων, χωρίς ωστόσο να υπάρχουν συνδέσεις μεταξύ νευρώνων του ίδιου επιπέδου. Η λειτουργία ενός MLP αποτελείται από δύο μέρη, τη διαδικασία προώθησης προς τα εμπρός (feedforward) για τον υπολογισμό της εξόδου και τον αλγόριθμο οπισθοδιάδοσης του σφάλματος (backpropagation) για την εκπαίδευση των παραμέτρων.



Εικόνα 1 Αρχιτεκτονική MLP με ένα ενδιάμεσο (κρυφό) επίπεδο και αριθμό νευρώνων 3, 4 και 1 στα επίπεδα εισόδου, κρυφό και εξόδου αντίστοιχα (Qaddoura & Younes, 2023).

Κατά τη διαδικασία προώθησης προς τα εμπρός (feedforward), το νευρωνικό δίκτυο υπολογίζει την έξοδό του διαδίδοντας διαδοχικά την πληροφορία από το επίπεδο εισόδου προς το επίπεδο εξόδου. Έστω  $a^l \in R^{n_l}$  το διάνυσμα εξόδου του επιπέδου  $l$ , όπου  $l = 1, \dots, L$  και  $n_l$  ο αριθμός των νευρώνων του επιπέδου. Στο επίπεδο εισόδου

( $l = 1$ ), το διάνυσμα εισόδου και το διάνυσμα εξόδου ταυτίζονται, και συνεπώς ισχύει  $a^1 = x_i$ . Κάθε νευρώνας ενός επόμενου επιπέδου λαμβάνει ως είσοδο έναν γραμμικό συνδυασμό των εξόδων των νευρώνων του προηγούμενου επιπέδου, στον οποίο προστίθεται ένας όρος πόλωσης (bias). Πιο συγκεκριμένα, η είσοδος του νευρώνα  $j$  επιπέδου  $l$  είναι

$$z_j^l = \sum_{k=1}^{n_{l-1}} w_{jk}^l a_k^{l-1} + b_j^l,$$

όπου  $b_j^l$  η τιμή πόλωσης του νευρώνα  $j$  του επιπέδου  $l$ , με  $a_j = 1, 2, \dots, n_l$ ,  $w_{jk}^l$  είναι το βάρος από τον νευρώνα  $k$  του επιπέδου  $l - 1$  προς τον νευρώνα  $j$  του επιπέδου  $l$  και  $a_k^{l-1}$  είναι η έξοδος του νευρώνα  $k$  του επιπέδου  $l - 1$  και  $b_j^l$ . Επομένως το διάνυσμα εισόδου και το διάνυσμα πόλωσης του επιπέδου  $l$  ορίζεται ως  $z^l = (z_1^l, z_2^l, \dots, z_{n_l}^l)^T$  και  $b^l = (b_1^l, b_2^l, \dots, b_{n_l}^l)^T$  αντίστοιχα. Στη συνέχεια, η είσοδος  $z_j^l$  διέρχεται από τη συνάρτηση ενεργοποίησης  $\sigma^{(l)}(\cdot)$  του επιπέδου  $l$ , για να προκύψει η έξοδος  $a_j^l = \sigma^{(l)}(z_j^l)$  του νευρώνα  $j$  του επιπέδου  $l$ . Άρα, το διάνυσμα εξόδου του επιπέδου  $l$  ορίζεται ως  $a^l = (a_1^l, a_2^l, \dots, a_{n_l}^l)^T \Leftrightarrow a^l = \sigma^{(l)}(z^l)$ , με την  $\sigma^{(l)}(\cdot)$  να εφαρμόζεται κατά στοιχείο. Το  $a^l$  αποτελεί την είσοδο του επόμενου επιπέδου  $l + 1$ . Η διαδικασία επαναλαμβάνεται διαδοχικά μέχρι το επίπεδο εξόδου, όπου το  $a^L$  αντιστοιχεί στην τελική πρόβλεψη του δικτύου. Συνοψίζοντας, η διαδικασία προώθησης προς τα εμπρός (feedforward) σε ένα πολυεπίπεδο perceptron περιγράφεται από τις διανυσματικές σχέσεις  $z^l = W^l a^{l-1} + b^l$ ,  $a^l = \sigma^{(l)}(z^l)$  για κάθε επίπεδο  $l = 2, \dots, L$  και όπου

$$W^l = \begin{pmatrix} w_{11}^l & w_{12}^l & \dots & w_{1n_{l-1}}^l \\ w_{21}^l & w_{22}^l & \dots & w_{2n_{l-1}}^l \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_l 1}^l & w_{n_l 2}^l & \dots & w_{n_l n_{l-1}}^l \end{pmatrix} \in \mathbb{R}^{n_l \times n_{l-1}}, a^{l-1} \in \mathbb{R}^{n_{l-1}} \text{ και } b^l, z^l \in \mathbb{R}^{n_l}.$$

Κατά τη διαδικασία οπισθοδιάδοσης (backpropagation), το νευρωνικό δίκτυο υπολογίζει τις παραγώγους της συνάρτησης κόστους ως προς τις παραμέτρους του μοντέλου (βάρη και όρους πόλωσης), διαδίδοντας το σφάλμα από το επίπεδο εξόδου προς τα προηγούμενα επίπεδα. Η διαδικασία αυτή αποτελεί τον θεμελιώδη μηχανισμό που επιτρέπει την αποδοτική εκπαίδευση πολυεπίπεδων νευρωνικών δικτύων μέσω μεθόδων βελτιστοποίησης βασισμένων στην κλίση. Έστω  $C$  η συνάρτηση κόστους του

δικτύου για μία παρατήρηση  $(x_i, y_i)$ . Ορίζουμε για κάθε επίπεδο  $l$  το διάνυσμα σφάλματος

$$\delta^l = \frac{\partial C}{\partial z^l} \in \mathbb{R}^{n_l},$$

το οποίο εκφράζει τον ρυθμό μεταβολής του κόστους ως προς τις σταθμισμένες εισόδους  $z^l$  του επιπέδου  $l$ . Για το επίπεδο εξόδου  $L$ , το διάνυσμα σφάλματος δίνεται από τη σχέση

$$\delta^L = \nabla_{a^L} C \odot \sigma'^{(L)}(z^L) = \left( \frac{\partial C}{\partial a_1^L}, \frac{\partial C}{\partial a_2^L}, \dots, \frac{\partial C}{\partial a_{n_L}^L} \right)^T \odot \sigma'^{(L)}(z^L),$$

όπου  $\nabla_{a^L} C$  είναι το διάνυσμα μερικών παραγώγων του κόστους ως προς τις εξόδους του δικτύου,  $\sigma'^{(L)}(\cdot)$  η παράγωγος της συνάρτησης ενεργοποίησης του επιπέδου εξόδου και  $\odot$  το κατά στοιχείο γινόμενο (Hadamard product). Για κάθε εσωτερικό επίπεδο  $l = L - 1, \dots, 2$ , το σφάλμα υπολογίζεται αναδρομικά ως

$$\delta^l = ((W^{l+1})^T \delta^{l+1}) \odot \sigma'^{(l)}(z^l).$$

Η παραπάνω σχέση εκφράζει τη διάδοση του σφάλματος από το επίπεδο  $l + 1$  στο επίπεδο  $l$ , προσαρμοσμένη από την τοπική μη γραμμικότητα της συνάρτησης ενεργοποίησης. Αφού υπολογιστούν τα  $\delta^l$ , οι παράγωγοι της συνάρτησης κόστους ως προς τις παραμέτρους του δικτύου δίνονται από:

$$\frac{\partial C}{\partial W^l} = \delta^l (a^{l-1})^T, \frac{\partial C}{\partial b^l} = \delta^l, l = 2, \dots, L.$$

Οι σχέσεις αυτές επιτρέπουν τον άμεσο υπολογισμό των μερικών παραγώγων όλων των βαρών και των όρων πόλωσης. Συνοψίζοντας, η διαδικασία οπισθοδιάδοσης σε ένα πολυεπίπεδο perceptron περιγράφεται από τις σχέσεις

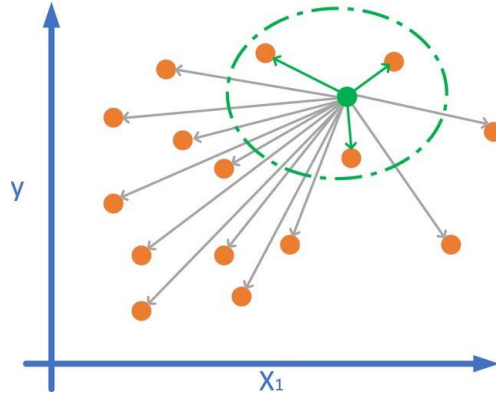
$$\begin{aligned} \delta^L &= \nabla_{a^L} C \odot \sigma'^{(L)}(z^L), \\ \delta^l &= (W^{l+1})^T \delta^{l+1} \odot \sigma'^{(l)}(z^l), l = L - 1, \dots, 2, \\ \frac{\partial C}{\partial W^l} &= \delta^l (a^{l-1})^T, \frac{\partial C}{\partial b^l} = \delta^l, l = 2, \dots, L. \end{aligned}$$

Οι παραπάνω εξισώσεις καθορίζουν πλήρως τον μηχανισμό μέσω του οποίου το σφάλμα διαδίδεται προς τα πίσω και υπολογίζονται οι απαραίτητες παράγωγοι για την ενημέρωση των παραμέτρων του δικτύου (Nielsen, 2015).

Στο πλαίσιο της παρούσας εργασίας, το MLP επιλέχθηκε λόγω της ικανότητάς του να μοντελοποιεί πολύπλοκες και μη γραμμικές σχέσεις μεταξύ των χαρακτηριστικών εισόδου και της μεταβλητής στόχου, καθώς και της ευελιξίας της αρχιτεκτονικής του ως προς τον αριθμό των στρωμάτων και των νευρώνων.

#### 2.4.2 K-Κοντινότεροι Γείτονες (KNN)

Ο K-Nearest Neighbors (KNN) αποτελεί έναν από τους απλούστερους και ευρέως χρησιμοποιούμενους αλγορίθμους στη μηχανική μάθηση, καθώς βασίζεται στην ομοιότητα μεταξύ των δεδομένων για την πραγματοποίηση προβλέψεων. Η θεμελιώδης παραδοχή του είναι ότι παρατηρήσεις με παρόμοια χαρακτηριστικά τείνουν να παρουσιάζουν και παρόμοιες τιμές εξόδου. Επιπλέον, δεν προϋποθέτει την κατασκευή μαθηματικού μοντέλου κατά το στάδιο της εκπαίδευσης, αλλά αποθηκεύει το σύνολο των δεδομένων και λαμβάνει αποφάσεις κατά τη φάση της πρόβλεψης.



Εικόνα 2: Απεικόνιση kNN παλινδρόμησης για  $k=3$  με Ευκλείδεια απόσταση (Qaddoura & Younes, 2023).

Για μια νέα είσοδο  $x \in \mathbb{R}^d$ , υπολογίζεται η απόστασή της από όλα τα σημεία  $x_i$  του συνόλου εκπαίδευσης, σύμφωνα με μια επιλεγμένη μετρική απόστασης  $d(x, x_i)$ . Συνήθεις επιλογές αποτελούν η Ευκλείδεια, η Manhattan και η Minkowski απόσταση.

Ευκλείδεια μετρική: 
$$d(x, x_i) = \sqrt{\sum_{j=1}^d (x_j - x_{ij})^2}$$

Manhattan μετρική: 
$$d(x, x_i) = \sum_{j=1}^d |x_j - x_{ij}|$$

Minkowski μετρική: 
$$d(x, x_i) = \left( \sum_{j=1}^d |x_j - x_{ij}|^p \right)^{\frac{1}{p}}, \quad p \geq 1$$

Με βάση τις αποστάσεις αυτές, προσδιορίζεται το σύνολο  $N_k(x)$ , το οποίο περιλαμβάνει τις  $k$  κοντινότερες παρατηρήσεις προς το  $x$ . Τέλος, η εκτίμηση της εξόδου  $\hat{y}(x)$  προκύπτει συνήθως ως ο μέσος όρος των τιμών εξόδου των

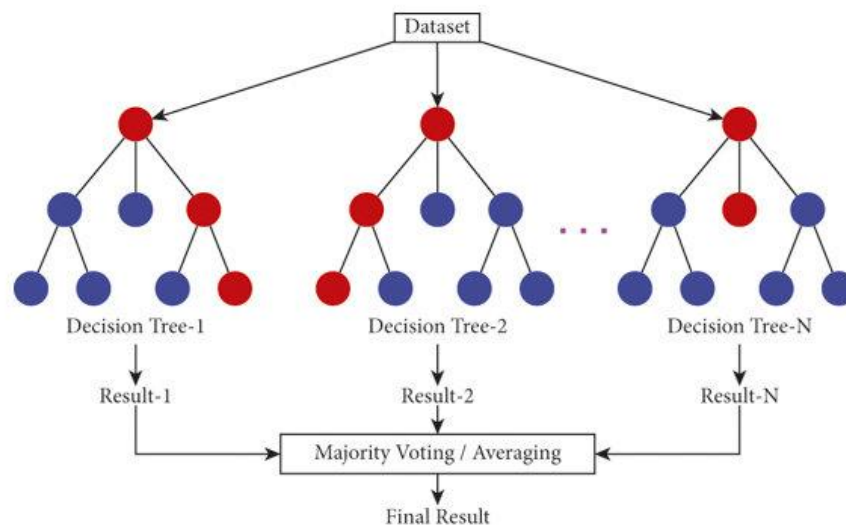
παρατηρήσεων που ανήκουν στο  $N_k(x)$  και δίνεται μαθηματικά από τη σχέση (Song et al., 2017):

$$\hat{y}(x) = \frac{1}{k} \sum_{x_i \in N_k(x)} y_i$$

Στο πλαίσιο της παρούσας εργασίας, ο αλγόριθμος KNN επιλέχθηκε λόγω της απλότητάς του και της ικανότητάς του να αποτυπώνει μη γραμμικές σχέσεις στα δεδομένα, καθιστώντας τον ένα εύχρηστο και αποτελεσματικό εργαλείο για προβλήματα παλινδρόμησης χρονοσειρών.

### 2.4.3 Τυχαία Δάση (RF)

Τα Τυχαία Δάση (Random Forest) αποτελούν μέθοδο συνδυασμού μοντέλων μάθησης, η οποία βασίζεται στη χρήση πολλαπλών δέντρων απόφασης. Κάθε δέντρο εκπαιδεύεται ξεχωριστά και λειτουργεί ως επιμέρους μοντέλο, το οποίο όταν εξετάζεται μεμονωμένα, εμφανίζει συνήθως περιορισμένη προβλεπτική ικανότητα. Όταν όμως τα επιμέρους αυτά μοντέλα συνδυαστούν μέσω λήψης του μέσου όρου ή σταθμισμένου μέσου όρου των προβλέψεών τους, προκύπτει ένα τελικό μοντέλο με σημαντικά βελτιωμένη προβλεπτική ακρίβεια και αυξημένη ικανότητα γενίκευσης.



Εικόνα 3 Ενδεικτική γραφική απεικόνιση δέντρου απόφασης (Khan et al., 2021).

Η επιτυχία της μεθόδου βασίζεται σε δύο βασικούς μηχανισμούς τυχαιοποίησης κατά τη διαδικασία της εκπαίδευσης. Ο πρώτος μηχανισμός είναι η μέθοδος Bagging, σύμφωνα με την οποία για κάθε δέντρο δημιουργείται ένα σύνολο εκπαίδευσης ίδιου μεγέθους με το αρχικό, το οποίο προκύπτει μέσω δειγματοληψίας με

επανατοποθέτηση. Αυτό σημαίνει ότι ορισμένες παρατηρήσεις μπορεί να εμφανίζονται περισσότερες από μία φορές στο ίδιο δείγμα, ενώ άλλες ενδέχεται να μην εμφανίζονται καθόλου. Κατά μέσο όρο, περίπου το 37% των αρχικών δεδομένων δεν περιλαμβάνεται σε κάθε επιμέρους δείγμα. Η εισαγωγή τυχαιότητας μέσω του Bagging οδηγεί σε μεγαλύτερη διαφοροποίηση μεταξύ των δέντρων, γεγονός που συμβάλλει στη βελτίωση της ικανότητας γενίκευσης. Ο δεύτερος μηχανισμός τυχαιοποίησης είναι η δειγματοληψία χαρακτηριστικών. Σε κάθε κόμβο ενός δέντρου, αντί να εξετάζονται όλα τα διαθέσιμα χαρακτηριστικά για την επιλογή του βέλτιστου διαχωρισμού, επιλέγεται τυχαία ένα υποσύνολο χαρακτηριστικών και ο καλύτερος διαχωρισμός αναζητείται μόνο μέσα σε αυτό το υποσύνολο. Κατά τον Leo Breiman, εισηγητή του αλγορίθμου Random Forest, ο αριθμός χαρακτηριστικών που προτείνεται να εξετάζεται σε κάθε κόμβο ισούται με  $\lfloor \log_2(p) + 1 \rfloor$ , όπου  $p$  είναι ο συνολικός αριθμός χαρακτηριστικών. Η διαδικασία αυτή μειώνει τη συσχέτιση μεταξύ των δέντρων και ενισχύει τη διαφοροποίησή τους, στοιχείο κρίσιμο για τη βελτίωση της συνολικής απόδοσης του συνόλου (Bentéjac et al., 2021).

Όπως αναφέρθηκε προηγουμένως, σε προβλήματα παλινδρόμησης, για μια νέα παρατήρηση  $x$ , κάθε δέντρο  $T_i$  παράγει μια επιμέρους πρόβλεψη  $f_i(x)$ . Η τελική πρόβλεψη του Random Forest προκύπτει ως ο αριθμητικός ή σταθμισμένος μέσος των προβλέψεων όλων των δέντρων και δίνεται αντίστοιχα από

$$\hat{y} = \frac{1}{N} \sum_{i=1}^N f_i(x) \text{ ή } \hat{y} = \sum_{i=1}^N w_i f_i(x), \text{ όπου } \sum_{i=1}^N w_i = 1.$$

Στο πλαίσιο της παρούσας εργασίας, ο αλγόριθμος RF επιλέχθηκε λόγω της αποδεδειγμένης αποτελεσματικότητάς του σε προβλήματα παλινδρόμησης, της ανθεκτικότητάς του στον θόρυβο και στις ακραίες τιμές, καθώς και της μειωμένης τάσης υπερπροσαρμογής που τον χαρακτηρίζει.

#### 2.4.4 Gradient Boosting (GB)

Οι αλγόριθμοι της κατηγορίας Boosting βασίζονται στην ιδέα ότι μοντέλα τα οποία από μόνα τους διαθέτουν περιορισμένη προβλεπτική ικανότητα μπορούν, όταν συνδυαστούν κατάλληλα, να σχηματίσουν ένα ισχυρό τελικό μοντέλο. Η διαδικασία είναι επαναληπτική και σε κάθε βήμα προστίθεται ένα νέο μοντέλο, το οποίο επιδιώκει να διορθώσει τα σφάλματα των προηγούμενων.

Πιο συγκεκριμένα, ο αλγόριθμος Gradient Boosting, για ένα σύνολο δεδομένων  $T = \{(x_i, y_i)\}_{i=1}^N$ , έχει ως στόχο να υπολογίσει μία προσέγγιση  $\hat{F}(x)$  της άγνωστης συνάρτησης  $F^*(x)$ , η οποία αντιστοιχίζει το διάνυσμα χαρακτηριστικών μιας παρατήρησης  $x_i$  στην αντίστοιχη τιμή στόχο  $y_i$ . Η συνάρτηση  $\hat{F}(x)$  επιλέγεται έτσι ώστε να ελαχιστοποιεί μία συνάρτηση σφάλματος  $L(y, F(x))$ , η οποία μετρά την απόκλιση μεταξύ της πραγματικής τιμής  $y$  και της προβλεπόμενης τιμής  $F(x)$ . Η συνολική ποσότητα που ελαχιστοποιείται ονομάζεται αντικειμενική συνάρτηση και στην περίπτωση του Gradient Boosting ταυτίζεται με το άθροισμα της απώλειας σε όλες τις παρατηρήσεις:

$$\mathcal{L} = \sum_{i=1}^N L(y_i, F_m(x_i)).$$

Για την επίτευξη του παραπάνω στόχου, το Gradient Boosting κατασκευάζει τη συνάρτηση  $\hat{F}(x)$  με προσθετικό τρόπο, δηλαδή το μοντέλο δεν υπολογίζεται μονομιάς, αλλά χτίζεται σταδιακά ως άθροισμα απλούστερων μοντέλων. Αν συμβολίσουμε με  $F_m(x)$  το μοντέλο μετά από  $m$  επαναλήψεις, τότε αυτό γράφεται ως

$$F_m(x) = F_{m-1}(x) + \rho_m h_m(x),$$

όπου  $h_m(x)$  είναι το νέο επιμέρους μοντέλο που προστίθεται στην επανάληψη  $m$ , και  $\rho_m$  είναι ένας συντελεστής που καθορίζει τη συμβολή του στο τελικό μοντέλο. Με άλλα λόγια, κάθε νέο μοντέλο λειτουργεί ως διόρθωση του προηγούμενου. Η διαδικασία ξεκινά με έναν αρχικό εκτιμητή  $F_0(x)$ , ο οποίος επιλέγεται ως η σταθερά που ελαχιστοποιεί τη συνολική συνάρτηση σφάλματος στο σύνολο εκπαίδευσης

$$F_0(x) = \arg \min_{\alpha} \sum_{i=1}^N L(y_i, \alpha)$$

Στη συνέχεια, για κάθε επανάληψη  $m = 1, 2, \dots, M$ , ο αλγόριθμος υπολογίζει τα λεγόμενα ψευδο-υπόλοιπα (pseudo-residuals), τα οποία ορίζονται ως το αρνητικό της παραγώγου της συνάρτησης απώλειας ως προς την τρέχουσα πρόβλεψη:

$$r_{mi} = - \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}$$

Τα  $r_{mi}$  υποδεικνύουν, για κάθε παρατήρηση, πώς πρέπει να διορθωθεί η τρέχουσα πρόβλεψη ώστε να μειωθεί το σφάλμα του μοντέλου. Συγκεκριμένα, αν  $r_{mi} > 0$ , η πρόβλεψη πρέπει να αυξηθεί, ενώ αν  $r_{mi} < 0$ , η πρόβλεψη πρέπει να μειωθεί. Ακολούθως, το μοντέλο  $h_m(x)$  εκπαιδεύεται στο σύνολο, όπου ως τιμές-στόχος χρησιμοποιούνται τα  $r_{mi}$ . Δηλαδή, το  $h_m(x)$  προσαρμόζεται ώστε να προσεγγίζει όσο το δυνατόν καλύτερα την κατεύθυνση μείωσης της απώλειας. Με τον τρόπο αυτό, το νέο μοντέλο επιχειρεί να διορθώσει τα σφάλματα που παρατηρούνται στο  $F_{m-1}(x)$ . Αφού προσδιοριστεί το  $h_m(x)$ , υπολογίζεται ο βέλτιστος συντελεστής βάρους  $\rho_m$  επιλύοντας ένα πρόβλημα γραμμικής αναζήτησης:

$$\rho_m = \arg \min_{\rho} \sum_{i=1}^N L(y_i, F_{m-1}(x_i) + \rho h_m(x_i))$$

Τέλος, το μοντέλο ενημερώνεται σύμφωνα με τη  $F_m(x) = F_{m-1}(x) + \rho_m h_m(x)$ , και η διαδικασία επαναλαμβάνεται. Με τον τρόπο αυτό, το τελικό μοντέλο  $F_M(x)$  αποτελεί άθροισμα διαδοχικών διορθωτικών όρων, καθένας από τους οποίους στοχεύει στη σταδιακή μείωση της συνάρτησης σφάλματος (Bentéjac et al., 2021; Friedman, 2001).

Στο πλαίσιο της παρούσας εργασίας, ο αλγόριθμος Gradient Boosting επιλέχθηκε λόγω της υψηλής προβλεπτικής του ικανότητας σε προβλήματα παλινδρόμησης και της ευελιξίας του ως προς την επιλογή της συνάρτησης κόστους, γεγονός που τον καθιστά κατάλληλο για την πρόβλεψη χρονοσειρών περιβαλλοντικών δεδομένων.

#### 2.4.5 Extreme Gradient Boosting (XGB)

Ο αλγόριθμος Extreme Gradient Boosting αποτελεί μία βελτιωμένη εκδοχή του Gradient Boosting και βασίζεται στην ίδια βασική προσθετική λογική κατασκευής του μοντέλου, στην περίπτωση του XGBoost τα επιμέρους μοντέλα είναι δέντρα απόφασης. Αρχικά υπολογίζεται ο εκτιμητής  $F_0(x)$ , στη συνέχεια, σε κάθε επανάληψη προστίθεται ένα νέο δέντρο  $f_m(x)$ , έτσι ώστε το μοντέλο μετά από  $m$  επαναλήψεις να γράφεται ως

$$F_m(x) = F_{m-1}(x) + f_m(x)$$

Σε αντίθεση με το κλασικό GB, στο XGB δεν ελαχιστοποιείται μόνο η συνάρτηση απώλειας, αλλά μία κανονικοποιημένη αντικειμενική συνάρτηση, η οποία

περιλαμβάνει τόσο το σφάλμα πρόβλεψης όσο και έναν όρο που ελέγχει την πολυπλοκότητα των δέντρων. Η αντικειμενική συνάρτηση γράφεται ως

$$\mathcal{L} = \sum_{i=1}^N L(y_i, F_m(x_i)) + \sum_{m=1}^M \Omega(f_m)$$

όπου  $L(y_i, F_m(x_i))$  είναι η συνάρτηση απώλειας και  $\Omega(f_m)$  είναι ο όρος κανονικοποίησης που περιορίζει την πολυπλοκότητα των δέντρων. Ο όρος αυτός ορίζεται ως  $\Omega(f_m) = \gamma T_m + \frac{1}{2} \lambda |w|^2$ , όπου  $T_m$  είναι ο αριθμός των φύλλων του δέντρου,  $w = (w_1, \dots, w_T)$  το διάνυσμα των τιμών πρόβλεψης των φύλλων, ενώ οι παράμετροι  $\gamma$  και  $\lambda$  χρησιμοποιούνται για τον έλεγχο της πολυπλοκότητας του μοντέλου. Για την ελαχιστοποίηση της αντικειμενικής συνάρτησης, η συνάρτηση απώλειας προσεγγίζεται μέσω ανάπτυξης Taylor δεύτερης τάξης γύρω από την τρέχουσα πρόβλεψη  $F_{m-1}(x_i)$ . Με τον τρόπο αυτό η αντικειμενική συνάρτηση μπορεί να γραφεί ως

$$\tilde{\mathcal{L}} = \sum_{i=1}^N \left[ g_i f_m(x_i) + \frac{1}{2} h_i f_m(x_i)^2 \right] + \Omega(f_m)$$

όπου  $g_i$  και  $h_i$  είναι αντίστοιχα η πρώτη και η δεύτερη παράγωγος της συνάρτησης απώλειας

$$g_i = \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}, h_i = \frac{\partial^2 L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)^2}.$$

Θεωρούμε το σύνολο  $I_j = \{i: q(x_i) = j\}$  των παρατηρήσεων  $x_i$  που ανήκουν στο φύλλο  $j$ , και τα αθροίσματα των πρώτων και δεύτερων παραγώγων

$$G_j = \sum_{i \in I_j} g_i, H_j = \sum_{i \in I_j} h_i.$$

Με τον τρόπο αυτό η προσεγγιστική αντικειμενική συνάρτηση μπορεί να εκφραστεί ως άθροισμα όρων που εξαρτώνται από τις τιμές πρόβλεψης των φύλλων

$$\tilde{\mathcal{L}} = \sum_{j=1}^T \left[ G_j w_j + \frac{1}{2} (H_j + \lambda) w_j^2 \right] + \gamma T$$

Η ελαχιστοποίηση της παραπάνω συνάρτησης οδηγεί στη βέλτιστη τιμή  $w_j^* = -\frac{G_j}{H_j + \lambda}$ , την οποία αντικαθιστώντας την στην αντικειμενική συνάρτηση προκύπτει

$$\tilde{\mathcal{L}}(q) = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T$$

Η εξίσωση αυτή μπορεί να χρησιμοποιηθεί ως συνάρτηση αξιολόγησης (scoring function) για τη μέτρηση της ποιότητας μιας δομής δέντρου  $q$ . Η τιμή αυτή είναι παρόμοια με το μέτρο impurity που χρησιμοποιείται για την αξιολόγηση των κόμβων στα κλασικά δέντρα απόφασης, με τη διαφορά ότι εδώ προκύπτει από μια πιο γενική μορφή αντικειμενικής συνάρτησης.

Στην πράξη δεν είναι εφικτό να εξεταστούν όλες οι πιθανές δομές δέντρων  $q$ , για τον λόγο αυτό χρησιμοποιείται ένας άπληστος αλγόριθμος, ο οποίος ξεκινά από ένα δέντρο που αποτελείται από ένα μόνο φύλλο και επεκτείνεται σταδιακά μέσω της προσθήκης διαχωρισμών στους κόμβους. Έστω ότι ένας κόμβος του δέντρου διαχωρίζεται σε δύο νέους κόμβους. Οι παρατηρήσεις που ανήκουν στον αρχικό κόμβο σχηματίζουν το σύνολο  $I$ , το οποίο μετά τον διαχωρισμό χωρίζεται σε δύο υποσύνολα το  $I_L$  και το  $I_R$ , που αντιστοιχεί στον αριστερό και δεξιό κόμβο αντίστοιχα. Επιπλέον για κάθε κόμβο υπολογίζονται τα αθροίσματα  $G_j$  και  $H_j$ , και με βάση τα μεγέθη αυτά μπορεί να υπολογιστεί η μείωση της αντικειμενικής συνάρτησης που προκαλεί ο διαχωρισμός ενός κόμβου. Η ποσότητα αυτή εκφράζει πόσο βελτιώνεται το μοντέλο μετά το διαχωρισμό και δίνεται από

$$L_{split} = \frac{1}{2} \left( \frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{G_P^2}{H_P + \lambda} \right).$$

Ο παραπάνω τύπος χρησιμοποιείται κατά την εκπαίδευση του μοντέλου για την αξιολόγηση των υποψήφιων διαχωρισμών των κόμβων. Ο διαχωρισμός που επιλέγεται σε κάθε βήμα είναι εκείνο που οδηγεί στη μεγαλύτερη μείωση της αντικειμενικής συνάρτησης (Bentéjac et al., 2021; Chen & Guestrin, 2016).

#### 2.4.6 Light Gradient Boosting Machine (LGBM)

Ο αλγόριθμος Light Gradient Boosting Machine (LightGBM) αποτελεί μία αποδοτική υλοποίηση των Gradient Boosting Decision Trees (GBDT), σχεδιασμένη ώστε να αντιμετωπίζει αποτελεσματικά προβλήματα μεγάλης κλίμακας τόσο ως προς

τον αριθμό των παρατηρήσεων όσο και ως προς τον αριθμό των χαρακτηριστικών. Όπως και στο Gradient Boosting και στο XGBoost, το μοντέλο κατασκευάζεται προσθετικά ως άθροισμα διαδοχικών δέντρων παλινδρόμησης.

Η βασική διαφορά του LightGBM σε σχέση με τις προηγούμενες υλοποιήσεις boosting έγκειται στον τρόπο κατασκευής των δέντρων. Ενώ οι περισσότερες υλοποιήσεις χρησιμοποιούν ανάπτυξη δέντρου ανά επίπεδο (level-wise tree growth), το LightGBM χρησιμοποιεί ανάπτυξη δέντρου ανά φύλλο (leaf-wise tree growth), δηλαδή επεκτείνει σε κάθε βήμα το φύλλο που οδηγεί στη μεγαλύτερη μείωση της αντικειμενικής συνάρτησης. Η στρατηγική αυτή επιτρέπει την ταχύτερη μείωση του σφάλματος και την παραγωγή βαθύτερων και πιο ακριβών δέντρων.

Επιπλέον, το LightGBM εισάγει δύο σημαντικές τεχνικές για τη βελτίωση της αποδοτικότητας:

- Gradient-based One-Side Sampling (GOSS): τεχνική δειγματοληψίας που διατηρεί τις παρατηρήσεις με μεγάλα gradients και δειγματοληπτεί τυχαία εκείνες με μικρά gradients, μειώνοντας το υπολογιστικό κόστος.
- Exclusive Feature Bundling (EFB): τεχνική που συνδυάζει αραιά χαρακτηριστικά σε μικρότερο αριθμό χαρακτηριστικών χωρίς σημαντική απώλεια πληροφορίας.

Οι τεχνικές αυτές επιτρέπουν στο LightGBM να εκπαιδεύεται σημαντικά ταχύτερα και να απαιτεί λιγότερη μνήμη, διατηρώντας ταυτόχρονα υψηλή ακρίβεια πρόβλεψης (Ke et al., n.d.).

## 2.5 Μοντέλα Βαθιάς Μάθησης

Τα αναδρομικά νευρωνικά δίκτυα (Recurrent Neural Networks, RNN) αποτελούν μια κατηγορία νευρωνικών δικτύων σχεδιασμένων για την επεξεργασία ακολουθιακών δεδομένων, όπως οι χρονοσειρές. Σε αντίθεση με τα κλασικά νευρωνικά δίκτυα, τα οποία επεξεργάζονται κάθε είσοδο ανεξάρτητα, τα RNN διαθέτουν εσωτερική μνήμη, καθώς η έξοδος κάθε χρονικής στιγμής εξαρτάται τόσο από την τρέχουσα είσοδο όσο και από την κρυφή κατάσταση της προηγούμενης χρονικής στιγμής. Με τον τρόπο αυτό, το δίκτυο είναι σε θέση να αξιοποιεί πληροφορία από προηγούμενες χρονικές στιγμές, καθιστώντας τα ιδιαίτερα κατάλληλα για την αποτύπωση χρονικών εξαρτήσεων σε ακολουθιακά δεδομένα. Ωστόσο, παρά τη

θεωρητική τους ικανότητα να μοντελοποιούν μακροχρόνιες εξαρτήσεις, τα RNN αντιμετωπίζουν στην πράξη σοβαρά προβλήματα κατά την εκπαίδευση μέσω οπισθοδιάδοσης στο χρόνο. Συγκεκριμένα, κατά την οπισθοδιάδοση οι κλίσεις περνούν από πολλά χρονικά βήματα και ο επαναληπτικός πολλαπλασιασμός τους οδηγεί είτε σε εκθετική αύξηση (exploding gradient) είτε σε εκθετική μείωση το (vanishing gradient) με αποτέλεσμα το δίκτυο να αδυνατεί να μάθει μακροχρόνιες εξαρτήσεις. Για την αντιμετώπιση αυτών των περιορισμών αναπτύχθηκαν οι αρχιτεκτονικές LSTM και GRU, οι οποίες παρουσιάζονται στις επόμενες ενότητες.

### 2.5.1 Long Short-Term Memory (LSTM)

Ο αλγόριθμος LSTM (Long Short-Term Memory) αναπτύχθηκε ως λύση στα προβλήματα που χαρακτηρίζουν τα απλά RNN (Hochreiter & Schmidhuber, 1997). Το βασικό στοιχείο που το διαφοροποιεί είναι η μονάδα μνήμης (cell state), η οποία διατρέχει ολόκληρη την αλυσίδα χρονικών βημάτων με ελάχιστες παρεμβάσεις, επιτρέποντας τη διατήρηση πληροφορίας για μεγάλα χρονικά διαστήματα. Η ροή αυτής της πληροφορίας ελέγχεται από τρεις πύλες (gates), οι οποίες καθορίζουν πόση και ποια πληροφορία θα διέλθει σε κάθε χρονικό βήμα.

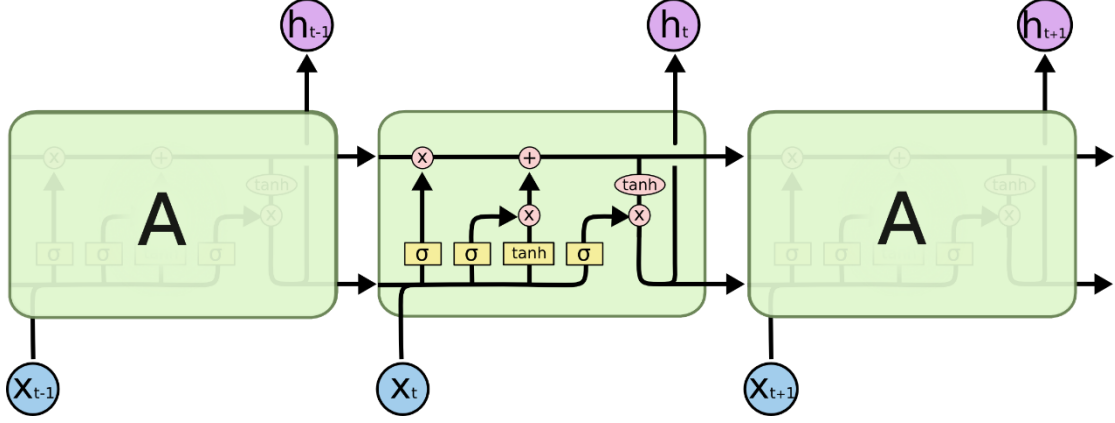
Η εκπαίδευση ενός LSTM ξεκινά από τη μετατροπή της αρχικής χρονοσειράς σε ένα σύνολο ακολουθιών εισόδου-εξόδου. Συγκεκριμένα, επιλέγεται ένας αριθμός υστερήσεων  $T$ , ο οποίος ορίζει πόσες συνεχόμενες παρελθοντικές τιμές θα χρησιμοποιηθούν κάθε φορά ως είσοδος στο μοντέλο, καθώς και ένας ορίζοντας πρόβλεψης  $h$ , ο οποίος ορίζει πόσες μελλοντικές τιμές επιδιώκουμε να προβλέψουμε. Με ολίσθηση ενός παραθύρου μεγέθους  $T$  κατά μήκος της χρονοσειράς, δημιουργείται ένα σύνολο παραδειγμάτων εκπαίδευσης. Σε κάθε παράδειγμα, η είσοδος αποτελείται από  $T$  διαδοχικές χρονικές στιγμές, όπου σε κάθε στιγμή αντιστοιχεί ένα διάνυσμα  $n$  χαρακτηριστικών που περιέχει την τιμή της χρονοσειράς και τυχόν επιπλέον χαρακτηριστικά. Η έξοδος που επιδιώκουμε να προβλέψουμε αποτελείται από τις  $h$  επόμενες τιμές της χρονοσειράς που ακολουθούν την ακολουθία εισόδου.

Έστω  $y_t \in \mathbb{R}$  η τιμή της χρονοσειράς στη χρονική στιγμή  $t$  και  $f_{j,t} \in \mathbb{R}$ , για  $j = 1, \dots, d$ , τα  $d$  επιπλέον χαρακτηριστικά (features) που είναι διαθέσιμα στη χρονική στιγμή  $t$ . Ορίζεται το διάνυσμα εισόδου

$$x_t = (y_t, f_{1,t}, \dots, f_{d,t}) \in \mathbb{R}^{d+1}$$

το οποίο περιγράφει πλήρως την πληροφορία στη χρονική στιγμή  $t$ . Έστω επίσης  $T \in \mathbb{N}$  το μήκος της ακολουθίας εισόδου (sequence length). Τότε η είσοδος του μοντέλου ορίζεται ως η ακολουθία

$$X = (x_{t-T+1}, \dots, x_t) \in \mathbb{R}^{T \times (d+1)}.$$



Εικόνα 4 Αρχιτεκτονική μονάδας LSTM με τις τρεις πύλες (λήθης, εισόδου, εξόδου) και τη μονάδα μνήμης (Olah, n.d.).

Η ακολουθία αυτή επεξεργάζεται από το LSTM, το οποίο σε κάθε χρονικό βήμα  $t$  λαμβάνει ως είσοδο το  $x_t$ , καθώς και την προηγούμενη κρυφή κατάσταση  $h_{t-1} \in R^m$  και την προηγούμενη κατάσταση μνήμης  $C_{t-1} \in R^m$ , όπου  $m \in \mathbb{N}$  είναι ο αριθμός των μονάδων του LSTM (hidden size). Σε κάθε χρονικό βήμα  $t$ , το LSTM εκτελεί μια σειρά από υπολογισμούς μέσω των πυλών του, οι οποίες ελέγχουν τη ροή της πληροφορίας και ενημερώνουν τις καταστάσεις  $h_t$  και  $C_t$ .

Η πύλη λήθης (forget gate) εξετάζει την προηγούμενη κρυφή κατάσταση  $h_{t-1}$  και την τρέχουσα είσοδο  $x_t$  και αποφασίζει ποια πληροφορία που ήδη βρίσκεται στη μονάδα μνήμης  $C_{t-1}$  δεν είναι πλέον χρήσιμη και πρέπει να απορριφθεί. Συγκεκριμένα, ο γραμμικός συνδυασμός των  $h_{t-1}$  και  $x_t$  διέρχεται από σιγμοειδή συνάρτηση ενεργοποίησης, παράγοντας το διάνυσμα

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

Αφού αποφασιστεί ποια πληροφορία θα απορριφθεί, το επόμενο βήμα αφορά την απόφαση για το ποια νέα πληροφορία θα αποθηκευτεί στη μονάδα μνήμης. Η διαδικασία αυτή αποτελείται από δύο παράλληλα υποβήματα. Πρώτον, η πύλη εισόδου (input gate) εξετάζει τις  $h_{t-1}$  και  $x_t$  και αποφασίζει ποιες τιμές της μονάδας μνήμης θα ενημερωθούν, παράγοντας το διάνυσμα

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), i_t \in R^m.$$

Δεύτερον, ένα επίπεδο με συνάρτηση ενεργοποίησης  $\tanh$  δημιουργεί ένα διάνυσμα υποψήφίων νέων τιμών που θα μπορούσαν να προστεθούν στη μονάδα μνήμης

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \tilde{C}_t \in R^m.$$

Έχοντας υπολογίσει το  $f_t$  από την πύλη λήθης και τα  $i_t, \tilde{C}_t$  από την πύλη εισόδου, η μονάδα μνήμης ενημερώνεται συνδυάζοντας δύο συνιστώσες. Η πρώτη είναι ο κατά στοιχείο πολλαπλασιασμός της προηγούμενης κατάστασης μνήμης  $C_{t-1}$  με το  $f_t$ , ο οποίος καθορίζει πόση από την υπάρχουσα πληροφορία διατηρείται. Η δεύτερη είναι ο κατά στοιχείο πολλαπλασιασμός των υποψήφίων νέων τιμών  $\tilde{C}_t$  με το  $i_t$ , ο οποίος καθορίζει πόση νέα πληροφορία προστίθεται. Η νέα κατάσταση μνήμης υπολογίζεται ως:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t, C_t \in R^m$$

Το τελευταίο βήμα αφορά τον υπολογισμό της νέας κρυφής κατάστασης  $h_t$ , η οποία αποτελεί την έξοδο του LSTM για τη χρονική στιγμή  $t$ . Η πύλη εξόδου (output gate) εξετάζει τις  $h_{t-1}$  και  $x_t$  και αποφασίζει ποια πληροφορία από τη μονάδα μνήμης  $C_t$  θα εξαχθεί, παράγοντας το διάνυσμα

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), o_t \in R^m$$

Στη συνέχεια, η μονάδα μνήμης  $C_t$  μετασχηματίζεται μέσω συνάρτησης  $\tanh$ , και πολλαπλασιάζεται κατά στοιχείο με το  $o_t$ , παράγοντας τη νέα κρυφή κατάσταση:

$$h_t = o_t \odot \tanh(C_t), h_t \in R^m.$$

Μετά την επεξεργασία όλων των χρονικών βημάτων της ακολουθίας, η τελική κρυφή κατάσταση  $h_T \in \mathbb{R}^m$  αποτελεί συμπυκνωμένη αναπαράσταση ολόκληρης της εισόδου και χρησιμοποιείται για την παραγωγή της πρόβλεψης. Συγκεκριμένα, η  $h_T$  περνά από ένα γραμμικό επίπεδο (dense layer), το οποίο παράγει την πρόβλεψη:

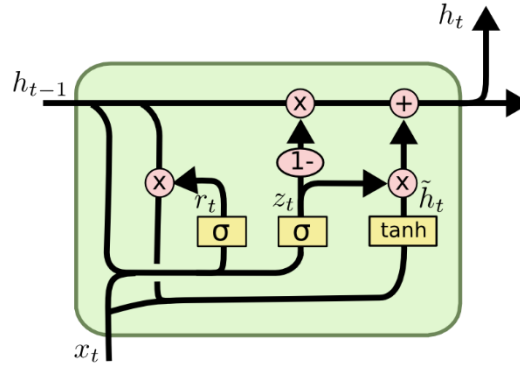
$$\hat{y} = W_y \cdot h_T + b_y, \hat{y} \in R^h$$

όπου  $h$  είναι ο ορίζοντας πρόβλεψης. Η πρόβλεψη αυτή συγκρίνεται με την πραγματική τιμή  $y$  μέσω μιας συνάρτησης κόστους, όπως το μέσο τετραγωνικό σφάλμα:  $L = \|y - \hat{y}\|^2$ , και το σφάλμα που προκύπτει χρησιμοποιείται για την ενημέρωση των βαρών του μοντέλου μέσω της διαδικασίας Backpropagation Through Time (BPTT). Η

διαδικασία επαναλαμβάνεται για όλα τα δείγματα εκπαίδευσης και για πολλαπλές επαναλήψεις (epochs), μέχρι το μοντέλο να συγκλίνει.

### 2.5.2 Gated Recurrent Unit (GRU)

Το GRU (Gated Recurrent Unit) προτάθηκε από τους (Cho et al., 2014) ως μια απλούστερη εναλλακτική του LSTM. Σε αντίθεση με το LSTM, το GRU δεν διαχωρίζει την κρυφή κατάσταση από τη μονάδα μνήμης, αλλά διατηρεί μόνο μία κρυφή κατάσταση  $h_t \in \mathbb{R}^m$ , η οποία επιτελεί και τους δύο ρόλους. Επιπλέον, χρησιμοποιεί δύο πύλες αντί για τρεις, μειώνοντας τον αριθμό των εκπαιδευσιμων παραμέτρων και την υπολογιστική πολυπλοκότητα, διατηρώντας παράλληλα συγκρίσιμη προβλεπτική ικανότητα.



Εικόνα 5 Αρχιτεκτονική μονάδας GRU με τις δύο πύλες (επαναφοράς, ενημέρωσης) (Olah, n.d.).

Η πύλη επαναφοράς (reset gate) εξετάζει την προηγούμενη κρυφή κατάσταση  $h_{t-1}$  και την τρέχουσα είσοδο  $x_t$  και αποφασίζει πόση από την προηγούμενη κρυφή κατάσταση θα συμμετέχει στον υπολογισμό των νέων υποψήφιων τιμών. Συγκεκριμένα, ο γραμμικός συνδυασμός των  $h_{t-1}$  και  $x_t$  διέρχεται από σιγμοειδή συνάρτηση ενεργοποίησης, παράγοντας το διάνυσμα

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r), r_t \in \mathbb{R}^m.$$

Το διάνυσμα αυτό στη συνέχεια πολλαπλασιάζεται κατά στοιχείο με την προηγούμενη κρυφή κατάσταση  $h_{t-1}$ , καθορίζοντας πόση από αυτήν θα συμμετέχει στον υπολογισμό των νέων υποψήφιων τιμών.

Ο γραμμικός συνδυασμός του  $r_t \odot h_{t-1}$  και του  $x_t$  διέρχεται από συνάρτηση ενεργοποίησης  $\tanh$ , παράγοντας το διάνυσμα

$$\tilde{h}_t = \tanh(W_h[r_t \odot h_{t-1}, x_t] + b_h), \tilde{h}_t \in \mathbb{R}^m.$$

Έχοντας υπολογίσει το  $r_t$  από την πύλη επαναφοράς και το  $\tilde{h}_t$ , η πύλη ενημέρωσης (update gate) εξετάζει τις  $h_{t-1}$  και  $x_t$  και αποφασίζει πόση από την προηγούμενη κρυφή κατάσταση θα διατηρηθεί και πόση από τις νέες υποψήφιες τιμές θα ενσωματωθεί. Συγκεκριμένα, ο γραμμικός συνδυασμός των  $h_{t-1}$  και  $x_t$  διέρχεται από σιγμοειδή συνάρτηση ενεργοποίησης, παράγοντας το διάνυσμα

$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z), z_t \in R^m.$$

Η νέα κρυφή κατάσταση υπολογίζεται συνδυάζοντας δύο συνιστώσες μέσω κατά στοιχείο πολλαπλασιασμού. Η πρώτη είναι ο κατά στοιχείο πολλαπλασιασμός της προηγούμενης κρυφής κατάστασης  $h_{t-1}$  με το  $(1 - z_t)$ , ο οποίος καθορίζει πόση από την υπάρχουσα πληροφορία διατηρείται. Η δεύτερη είναι ο κατά στοιχείο πολλαπλασιασμός των νέων υποψήφιων τιμών  $\tilde{h}_t$  με το  $z_t$ , ο οποίος καθορίζει πόση νέα πληροφορία ενσωματώνεται

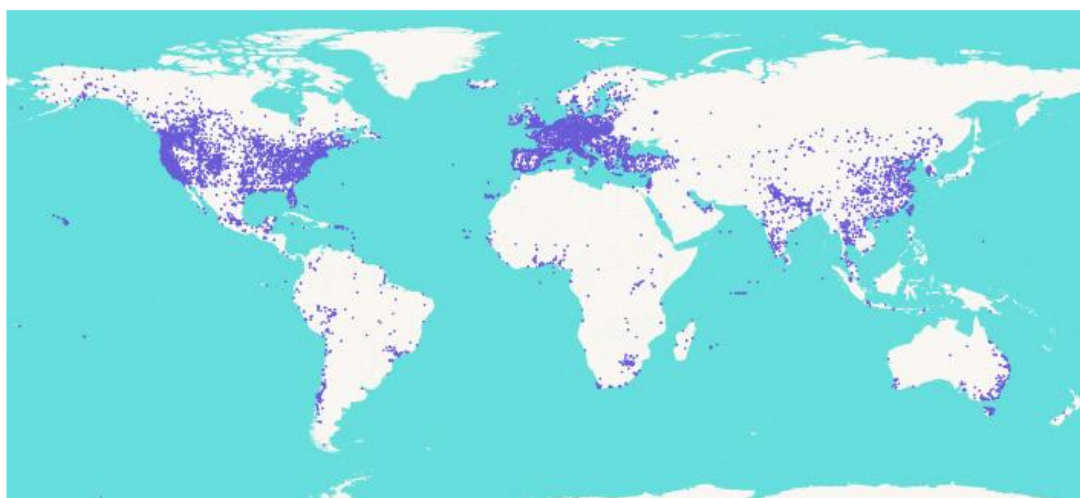
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, h_t \in R^m.$$

Μετά την επεξεργασία όλων των χρονικών βημάτων της ακολουθίας, η τελική κρυφή κατάσταση  $h_T$  χρησιμοποιείται για την παραγωγή της πρόβλεψης μέσω ενός γραμμικού επιπέδου

$$\hat{y} = W_y \cdot h_T + b_y, \hat{y} \in R^h.$$

## Κεφάλαιο 3: Δεδομένα OpenAQ

Η πλατφόρμα OpenAQ αποτελεί μια πλατφόρμα ανοιχτών δεδομένων η οποία συγκεντρώνει δεδομένα ατμοσφαιρικής ρύπανσης, τα οργανώνει σε ενιαία μορφή και τα καθιστά διαθέσιμα στους χρήστες για ανάλυση και επεξεργασία. Τα δεδομένα αυτά αφορούν μετρήσεις συγκεντρώσεων ατμοσφαιρικών ρύπων και προέρχονται από σταθμούς παρακολούθησης της ποιότητας του αέρα που λειτουργούν παγκοσμίως.



Εικόνα 6 Μέχρι τον Δεκέμβριο του 2022, η πλατφόρμα OpenAQ συγκεντρώνει περισσότερες από 25 δισεκατομμύρια μετρήσεις ποιότητας αέρα από περίπου 48.000 τοποθεσίες σε όλο τον κόσμο (Open+Air+Quality+Data+Global+Landscape+2022, n.d.).

Η πλατφόρμα δεν πραγματοποιεί η ίδια τις μετρήσεις, αλλά συγκεντρώνει δεδομένα που έχουν ήδη καταγραφεί από διάφορες επίσημες πηγές. Οι πηγές αυτές περιλαμβάνουν κυβερνητικούς οργανισμούς, εθνικά δίκτυα παρακολούθησης της ποιότητας του αέρα και άλλους αξιόπιστους φορείς. Με την ενοποίηση αυτών των δεδομένων σε μια κοινή πλατφόρμα, το OpenAQ διευκολύνει την ανάλυση και αξιοποίησή τους από ερευνητές, προγραμματιστές, δημοσιογράφους και πολίτες. Επιπλέον ένα βασικό χαρακτηριστικό του OpenAQ είναι ότι λειτουργεί ως πλατφόρμα ανοιχτών δεδομένων, δηλαδή τα δεδομένα που παρέχει είναι ελεύθερα διαθέσιμα σε όλους, χωρίς περιορισμούς πρόσβασης. Η πρόσβαση στα δεδομένα του OpenAQ μπορεί να πραγματοποιηθεί είτε μέσω της ιστοσελίδας της πλατφόρμας, όπου παρέχεται η δυνατότητα λήψης των μετρήσεων σε μορφή αρχείου CSV για συγκεκριμένους ρύπους, ωστόσο με περιορισμό στον αριθμό των δεδομένων που μπορούν να εξαχθούν, είτε μέσω προγραμματιστικής διεπαφής εφαρμογών (API –

Application Programming Interface), η οποία επιτρέπει την ανάκτηση μεγαλύτερου όγκου δεδομένων με μεγαλύτερη ευελιξία. Το API παρέχει προγραμματιστική πρόσβαση στα δεδομένα της πλατφόρμας, επιτρέποντας την αναζήτηση και ανάκτηση μετρήσεων ποιότητας αέρα μέσω διαδικτυακών αιτημάτων και επιστρέφοντας τα αποτελέσματα σε δομημένη μορφή δεδομένων (JSON). Με τον τρόπο αυτό, οι χρήστες μπορούν να ενσωματώνουν απευθείας τα δεδομένα του OpenAQ σε εφαρμογές ανάλυσης δεδομένων και υπολογιστικά εργαλεία, δηλαδή λογισμικό που χρησιμοποιείται για την επεξεργασία και ανάλυση δεδομένων.

### 3.1 Περιγραφή και Δομή των Δεδομένων OpenAQ

Τα δεδομένα που παρέχονται μέσω της πλατφόρμας OpenAQ είναι οργανωμένα σε μορφή πινάκων, όπου κάθε γραμμή αντιστοιχεί σε μία μέτρηση ποιότητας αέρα από έναν συγκεκριμένο σταθμό. Κάθε καταγραφή συνοδεύεται από πληροφορίες που περιγράφουν τόσο τη μέτρηση όσο και την τοποθεσία στην οποία πραγματοποιήθηκε. Πιο συγκεκριμένα, τα δεδομένα περιλαμβάνουν πεδία που αφορούν την ταυτότητα και το όνομα της τοποθεσίας (`location_id`, `location_name`), τον τύπο του ρύπου που μετρήθηκε (`parameter`), την τιμή της μέτρησης και τη μονάδα μέτρησης (`value`, `unit`), καθώς και τον χρόνο καταγραφής σε μορφή UTC και τοπικής ώρας (`datetimeUtc`, `datetimeLocal`). Επιπλέον, παρέχονται γεωγραφικές πληροφορίες για τον σταθμό μέτρησης, όπως οι γεωγραφικές συντεταγμένες (`latitude`, `longitude`), η ζώνη ώρας (`timezone`) και η χώρα στην οποία βρίσκεται ο σταθμός (`country_iso`). Στα δεδομένα περιλαμβάνονται επίσης πληροφορίες σχετικά με τον τύπο της συσκευής μέτρησης, όπως αν πρόκειται για κινητό αισθητήρα ή σταθερό σταθμό (`isMobile`, `isMonitor`), καθώς και στοιχεία σχετικά με τον οργανισμό που διαχειρίζεται ή παρέχει τα δεδομένα (`owner_name`, `provider`).

Σε κάθε σταθμό παρακολούθησης της ποιότητας του αέρα είναι εγκατεστημένοι αισθητήρες, καθένας από τους οποίους μετρά τη συγκέντρωση ενός συγκεκριμένου ατμοσφαιρικού ρύπου. Οι μετρήσεις αυτές αφορούν συνήθως τους έξι βασικούς ρύπους που χρησιμοποιούνται για την αξιολόγηση της ποιότητας του αέρα, ενώ σε ορισμένες περιπτώσεις καταγράφονται και επιπλέον ρύποι. Η καταγραφή των μετρήσεων γίνεται σε τακτά χρονικά διαστήματα μέσα στην ημέρα, συνήθως σε ωριαία βάση, με αποτέλεσμα τα δεδομένα να έχουν τη μορφή χρονοσειρών, αφού πρόκειται για καταγραφές που πραγματοποιούνται σε διαδοχικές χρονικές στιγμές.

### 3.2 Διαδικασία Συλλογής Δεδομένων και Προβλήματα Ποιότητας

Παρότι η πλατφόρμα OpenAQ παρέχει μεγάλο όγκο δεδομένων σχετικά με την ποιότητα του αέρα, τα δεδομένα αυτά ενδέχεται να παρουσιάζουν ορισμένα ζητήματα ποιότητας. Ένα συχνό πρόβλημα είναι η ύπαρξη ελλιπών τιμών, καθώς οι αισθητήρες δεν καταγράφουν πάντοτε μετρήσεις σε όλα τα χρονικά διαστήματα. Τα κενά αυτά μπορεί να οφείλονται σε τεχνικές βλάβες των αισθητήρων, σε διαδικασίες συντήρησης των σταθμών μέτρησης ή σε προβλήματα κατά τη μετάδοση των δεδομένων. Επιπλέον, τα δεδομένα που συγκεντρώνονται μέσω της πλατφόρμας προέρχονται από διαφορετικά δίκτυα παρακολούθησης και οργανισμούς, γεγονός που μπορεί να οδηγεί σε διαφοροποιήσεις ως προς τη συχνότητα καταγραφής, τη διαθεσιμότητα των μετρήσεων και τον τρόπο συλλογής των δεδομένων. Παράλληλα, η ποιότητα των μετρήσεων μπορεί να επηρεάζεται από παράγοντες όπως ο τύπος των αισθητήρων, οι διαδικασίες βαθμονόμησης και οι περιβαλλοντικές συνθήκες στις οποίες λειτουργούν οι σταθμοί, με αποτέλεσμα να εμφανίζονται σε ορισμένες περιπτώσεις ακραίες ή μη ρεαλιστικές τιμές.



Εικόνα 7 Αισθητήρες μέτρησης ατμοσφαιρικών ρύπων σε σταθμό παρακολούθησης της ποιότητας του αέρα (Pixel\_B/Adobe Stock, 2024).

## Κεφάλαιο 4: Μεθοδολογία

### 4.1 Σύνολο Δεδομένων

Τα δεδομένα που χρησιμοποιήθηκαν στην παρούσα εργασία προέρχονται από την πλατφόρμα OpenAQ και συλλέχθηκαν μέσω της προγραμματιστικής διεπαφής εφαρμογών (API) που παρέχει η πλατφόρμα. Η χρήση του API επιτρέπει την ανάκτηση των μετρήσεων ποιότητας του αέρα και τη δυνατότητα επιλογής συγκεκριμένων παραμέτρων. Για τις ανάγκες της παρούσας ανάλυσης επιλέχθηκαν η τοποθεσία, η χρονική περίοδος και ο τύπος του ατμοσφαιρικού ρύπου, ώστε να δημιουργηθεί η μονομεταβλητή χρονοσειρά που χρησιμοποιείται στη συνέχεια της μελέτης.

Η περιοχή που εξετάζεται είναι η πόλη του Νέου Δελχί στην Ινδία, μία από τις περιοχές που παρουσιάζουν υψηλά επίπεδα ατμοσφαιρικής ρύπανσης. Η ανάλυση επικεντρώνεται στον ατμοσφαιρικό ρύπο PM<sub>2.5</sub>, έναν από τους σημαντικότερους δείκτες αξιολόγησης της ποιότητας του αέρα λόγω των επιπτώσεών του στην ανθρώπινη υγεία. Το σύνολο δεδομένων περιλαμβάνει ωριαίες παρατηρήσεις της συγκέντρωσης των αιωρούμενων μικροσωματιδίων PM<sub>2.5</sub>, εκφρασμένες σε μικρογραμμάρια ανά κυβικό μέτρο αέρα ( $\mu\text{g}/\text{m}^3$ ), οι οποίες καλύπτουν χρονική περίοδο 8,5 ετών, από 01/01/2017 έως 30/06/2025.

| station_locality | station_country | station_name | location_id | pm25_sensor_id |
|------------------|-----------------|--------------|-------------|----------------|
| India            | India           | New Delhi    | 8118        | 23534          |

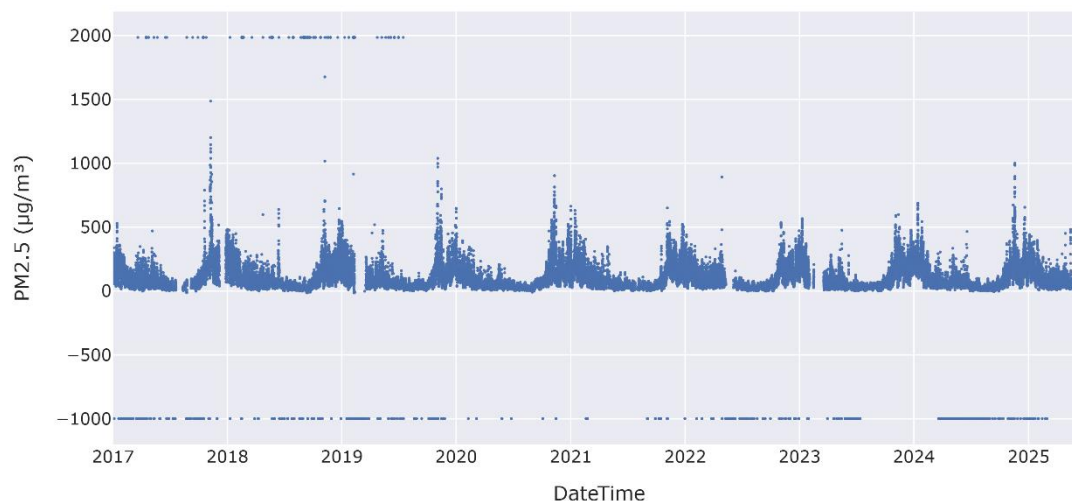
Πίνακας 2 Παράμετροι ανάκτησης δεδομένων από το API OpenAQ.

### 4.2 Προεπεξεργασία Δεδομένων

Πριν από την εφαρμογή των μεθόδων ανάλυσης και πρόβλεψης, το σύνολο δεδομένων υποβλήθηκε σε διαδικασία προεπεξεργασίας. Η προεπεξεργασία των δεδομένων αποτελεί σημαντικό στάδιο στην ανάλυση χρονοσειρών, καθώς τα δεδομένα που προέρχονται από συστήματα αισθητήρων ενδέχεται να περιλαμβάνουν σφάλματα καταγραφής, ακραίες τιμές ή ελλειπείς παρατηρήσεις. Στο πλαίσιο της παρούσας εργασίας πραγματοποιήθηκε έλεγχος και καθαρισμός των δεδομένων, με στόχο τη βελτίωση της ποιότητάς τους και τη διασφάλιση της αξιοπιστίας της περαιτέρω ανάλυσης. Οι επιμέρους διαδικασίες που εφαρμόστηκαν παρουσιάζονται στις επόμενες υποενότητες.

#### 4.2.1 Επεξεργασία Ακραίων Τιμών

Στην παρακάτω γραφική παράσταση απεικονίζονται οι ωριαίες μετρήσεις της συγκέντρωσης των PM<sub>2.5</sub>, όπως είναι καταγεγραμμένες στην πλατφόρμα OpenAQ, χωρίς προηγούμενη επεξεργασία. Στα δεδομένα εντοπίζονται ορισμένες μη ρεαλιστικές τιμές συγκέντρωσης, οι οποίες αφαιρούνται από το σύνολο δεδομένων, καθώς θεωρούνται αποτέλεσμα σφαλμάτων μέτρησης ή καταγραφής.

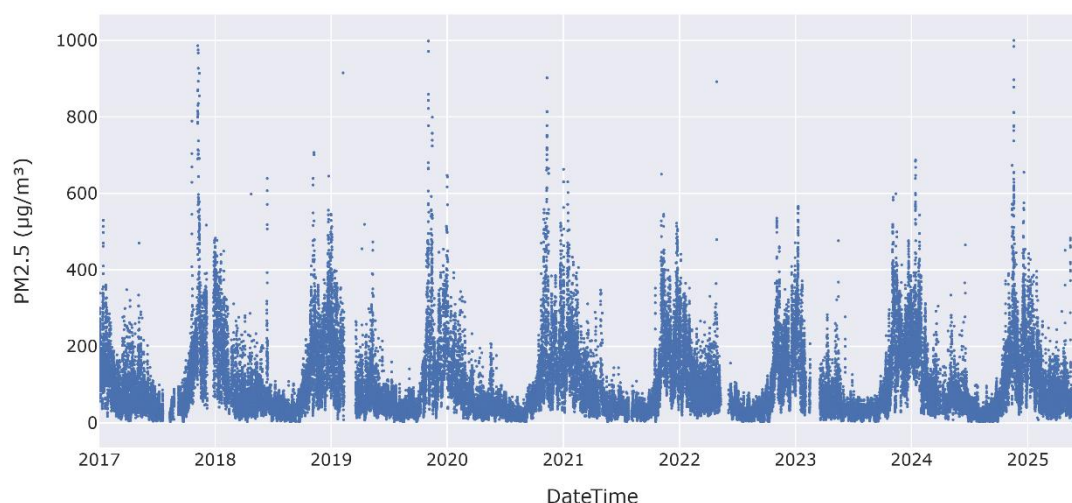


Εικόνα 8 Ωριαίες μετρήσεις συγκέντρωσης PM<sub>2.5</sub> στο Νέο Δελχί όπως είναι καταγεγραμμένες στην πλατφόρμα OpenAQ πριν από τον καθαρισμό των δεδομένων.

Ειδικότερα, οι συγκεντρώσεις με τιμές  $-999$  και  $1999 \mu\text{g}/\text{m}^3$  παρατηρούμε ότι εμφανίζονται συστηματικά σε διάφορα χρονικά σημεία της χρονοσειράς για το λόγο αυτό δεν μπορούν να θεωρηθούν έγκυρες μετρήσεις. Η επαναλαμβανόμενη παρουσία τους υποδηλώνει ότι πιθανότατα χρησιμοποιούνται ως ειδικές τιμές για την ένδειξη ελλিপών ή λανθασμένων μετρήσεων, οι οποίες μπορεί να προκύπτουν από σφάλματα των αισθητήρων ή της διαδικασίας καταγραφής των δεδομένων. Ακόμα, αφαιρούνται οι αρνητικές τιμές συγκέντρωσης, καθώς η συγκέντρωση ενός ατμοσφαιρικού ρύπου δεν μπορεί να λάβει αρνητικές τιμές. Επιπλέον, παρατηρούνται και ορισμένες συγκεντρώσεις μεγαλύτερες από  $1000 \mu\text{g}/\text{m}^3$  που, παρότι θεωρητικά θα μπορούσαν να εμφανιστούν σε συνθήκες έντονης ατμοσφαιρικής ρύπανσης στο Νέο Δελχί, κρίνονται ως ιδιαίτερα ακραίες για τους σκοπούς της παρούσας ανάλυσης και αντιμετωπίζονται ως ακραίες τιμές. Παράλληλα, παρατηρούνται και τιμές πολύ κοντά στο μηδέν, όπως συγκεντρώσεις μεταξύ  $0$  και  $2 \mu\text{g}/\text{m}^3$ , οι οποίες θεωρούνται μη ρεαλιστικά χαμηλές για το συγκεκριμένο αστικό περιβάλλον.

#### 4.2.2 Ελλιπή Δεδομένα και Χειρισμός τους

Αρχικά, στο σύνολο δεδομένων υπήρχαν ήδη ορισμένες ελλιπείς τιμές, ενώ η αφαίρεση των μη ρεαλιστικών και ακραίων τιμών συγκέντρωσης δημιούργησε επιπλέον κενά στη χρονοσειρά, με αποτέλεσμα το ποσοστό των ελλিপών τιμών να ανέρχεται περίπου στο 17% των παρατηρήσεων.

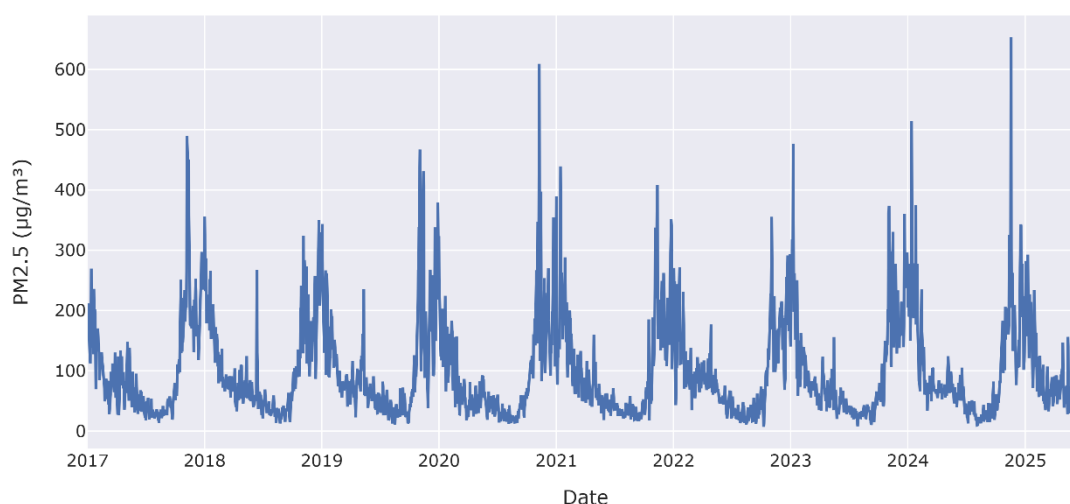


Εικόνα 9 Χρονοσειρά ωριαίων συγκεντρώσεων PM2.5 μετά την αφαίρεση ακραίων τιμών, όπου εμφανίζονται κενά λόγω ελλিপών παρατηρήσεων.

Από την εξέταση της χρονοσειράς διαπιστώνεται η παρουσία έντονης ετήσιας εποχικότητας στις συγκεντρώσεις PM2.5. Κατά την περίοδο των μουσώνων, δηλαδή περίπου από τον Ιούνιο έως τον Σεπτέμβριο, παρατηρείται σημαντική μείωση των συγκεντρώσεων. Η μείωση αυτή αποδίδεται κυρίως στις έντονες βροχοπτώσεις της περιόδου, οι οποίες συμβάλλουν στον καθαρισμό της ατμόσφαιρας από τα αιωρούμενα σωματίδια. Αντίθετα, κατά τους χειμερινούς μήνες, κυρίως από τον Νοέμβριο έως τον Ιανουάριο, εμφανίζονται οι υψηλότερες τιμές συγκέντρωσης PM2.5. Το φαινόμενο αυτό σχετίζεται με τις δυσμενείς μετεωρολογικές συνθήκες που επικρατούν την περίοδο αυτή, όπως η θερμοκρασιακή αναστροφή, κατά την οποία οι ψυχρότερες αέριες μάζες παγιδεύονται κοντά στην επιφάνεια του εδάφους και εμποδίζεται η κατακόρυφη μεταφορά του αέρα. Ως αποτέλεσμα, οι ρύποι δεν διαχέονται αποτελεσματικά στην ατμόσφαιρα και συσσωρεύονται κοντά στο έδαφος, οδηγώντας σε αυξημένες συγκεντρώσεις. Επιπλέον, σημαντικό ρόλο παίζει και η καύση γεωργικών υπολειμμάτων σε γειτονικές περιοχές, κυρίως στα κρατίδια Παντζάμπ και Χαριάνα, κατά τους μήνες Οκτώβριο και Νοέμβριο, γεγονός που συμβάλλει περαιτέρω στην αύξηση των συγκεντρώσεων στο Νέο Δελχί. Τέλος, παρατηρείται και μια

πρόσθετη αύξηση των συγκεντρώσεων κατά τον μήνα Μάιο, η οποία αποδίδεται κυρίως σε επεισόδια μεταφοράς σκόνης, καθώς και στις ξηρές και θερμές συνθήκες που επικρατούν πριν από την έναρξη της περιόδου των μουσώνων.

Για τη συμπλήρωση των ελλιπών τιμών εφαρμόστηκε μέθοδος συμπλήρωσης με βάση τον εποχικό μέσο όρο. Η μέθοδος αυτή βασίζεται στην υπόθεση ότι οι τιμές μιας χρονοσειράς παρουσιάζουν επαναλαμβανόμενα μοτίβα μέσα στο έτος. Στην περίπτωση των συγκεντρώσεων PM<sub>2.5</sub>, οι τιμές τείνουν να εμφανίζουν παρόμοια επίπεδα κατά την ίδια χρονική περίοδο σε διαφορετικά έτη. Συνεπώς, μια ελλιπής παρατήρηση μπορεί να εκτιμηθεί χρησιμοποιώντας τον μέσο όρο των διαθέσιμων τιμών που αντιστοιχούν στην ίδια χρονική περίοδο του έτους σε άλλα έτη της χρονοσειράς. Με τον τρόπο αυτό διατηρούνται τα βασικά εποχικά και ημερήσια πρότυπα μεταβολής των συγκεντρώσεων, αποφεύγοντας την αλλοίωση της δομής της χρονοσειράς.



Εικόνα 10 Χρονοσειρά ημερήσιων συγκεντρώσεων PM<sub>2.5</sub> στο Νέο Δελχί μετά την προεπεξεργασία των δεδομένων.

Στη συνέχεια, οι ωριαίες μετρήσεις συγκέντρωσης PM<sub>2.5</sub> μετατράπηκαν σε ημερήσιες τιμές μέσω του υπολογισμού του ημερήσιου μέσου όρου. Η μετατροπή αυτή επιτρέπει την άμεση σύγκριση των αποτελεσμάτων με διεθνείς κατευθυντήριες οδηγίες για την ποιότητα του αέρα, όπως αυτές που έχουν θεσπιστεί από τον ΠΟΥ και βασίζονται σε ημερήσιες συγκεντρώσεις. Επιπλέον, συμβάλλει στη μείωση του θορύβου που εμφανίζεται στις ωριαίες μετρήσεις και επιτρέπει την καλύτερη αποτύπωση των βασικών τάσεων και των εποχικών προτύπων της χρονοσειράς.

### 4.3 Επιλογή Αλγορίθμων και Δημιουργία Χαρακτηριστικών

Στην παρούσα εργασία μελετάται η πρόβλεψη των ημερήσιων συγκεντρώσεων του ρύπου PM<sub>2.5</sub> για δύο διαφορετικούς χρονικούς ορίζοντες πρόβλεψης, έναν βραχυπρόθεσμο δύο ημερών και έναν μακροπρόθεσμο επτά ημερών. Για κάθε χρονικό ορίζοντα πρόβλεψης αξιολογείται η απόδοση των μοντέλων, ώστε να εντοπιστεί το μοντέλο που επιτυγχάνει την καλύτερη απόδοση σε κάθε περίπτωση.

Για την πρόβλεψη των συγκεντρώσεων PM<sub>2.5</sub> χρησιμοποιήθηκε ένα σύνολο μοντέλων που περιλαμβάνει στατιστικές μεθόδους χρονοσειρών, αλγορίθμους μηχανικής μάθησης και μοντέλα βαθιάς μάθησης. Η επιλογή των αλγορίθμων βασίστηκε στη βιβλιογραφία του πεδίου, όπου οι συγκεκριμένες μέθοδοι εμφανίζονται επανειλημμένα ως οι πλέον αποδοτικές για προβλήματα πρόβλεψης συγκεντρώσεων PM<sub>2.5</sub> και ατμοσφαιρικών ρύπων γενικότερα.

Πιο συγκεκριμένα, χρησιμοποιήθηκε το στατιστικό μοντέλο ARIMA, το οποίο αποτελεί μια καθιερωμένη μέθοδο για την ανάλυση και πρόβλεψη χρονοσειρών και αντιπροσωπεύει την παραδοσιακή στατιστική προσέγγιση που κυριάρχησε στη βιβλιογραφία πριν την ευρεία υιοθέτηση των μεθόδων μηχανικής και βαθιάς μάθησης.

Στην κατηγορία των αλγορίθμων μηχανικής μάθησης περιλαμβάνονται οι K-Nearest Neighbors (KNN), Random Forest (RF), Gradient Boosting (GB), XGBoost, LightGBM και το νευρωνικό δίκτυο Multi-Layer Perceptron (MLP). Οι μέθοδοι αυτές είναι ιδιαίτερα αποτελεσματικές σε προβλήματα παλινδρόμησης και μπορούν να μοντελοποιήσουν πολύπλοκες μη γραμμικές σχέσεις μεταξύ των μεταβλητών. Ο αλγόριθμος KNN συμπεριλήφθηκε ως ένας απλός, μη παραμετρικός αλγόριθμος που δεν απαιτεί φάση εκπαίδευσης με την παραδοσιακή έννοια, καθώς η πρόβλεψη βασίζεται άμεσα στις τιμές των πλησιέστερων παρατηρήσεων στον χώρο των χαρακτηριστικών. Η συμπερίληψή του επιτρέπει την αξιολόγηση μιας μεθόδου χαμηλής πολυπλοκότητας ως σημείο σύγκρισης για τους πιο σύνθετους αλγόριθμους.

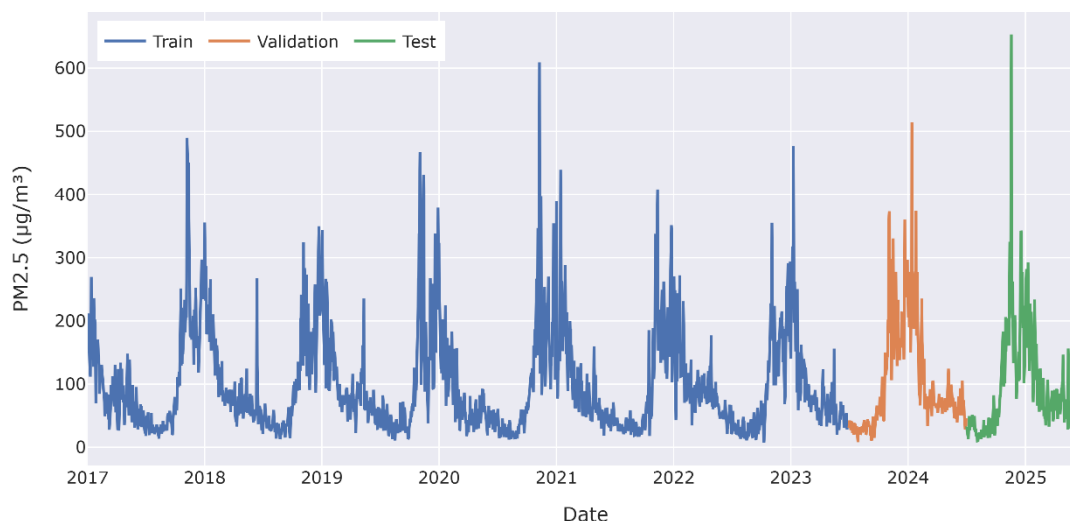
Τέλος, χρησιμοποιήθηκαν μοντέλα βαθιάς μάθησης και συγκεκριμένα τα αναδρομικά νευρωνικά δίκτυα Long Short-Term Memory (LSTM) και Gated Recurrent Unit (GRU), τα οποία είναι κατάλληλα για την ανάλυση χρονοσειρών, καθώς μπορούν να αξιοποιούν πληροφορία από προηγούμενα χρονικά βήματα και να αποτυπώνουν χρονικές εξαρτήσεις στα δεδομένα.

Στα μοντέλα μηχανικής μάθησης και βαθιάς μάθησης, η πρόβλεψη χρονοσειρών απαιτεί τη δημιουργία κατάλληλων χαρακτηριστικών, ώστε το πρόβλημα να διατυπωθεί ως πρόβλημα παλινδρόμησης. Για τον σκοπό αυτό χρησιμοποιήθηκαν υστερήσεις της μεταβλητής στόχου ως είσοδοι των μοντέλων, με τον αριθμό των υστερήσεων να αποτελεί υπερπαραμέτρο που καθορίστηκε κατά τη διαδικασία βελτιστοποίησης.

Συγκεκριμένα, για τον ορίζοντα πρόβλεψης των 2 ημερών εξετάστηκαν ως υποψήφιες τιμές του μεγέθους παραθύρου υστερήσεων τα 4, 7 και 14 προηγούμενα χρονικά βήματα. Αντίστοιχα, για τον ορίζοντα πρόβλεψης των 7 ημερών εξετάστηκαν τα 7, 14, 21 και 28 προηγούμενα χρονικά βήματα. Επιπλέον, υπολογίστηκαν κινητοί μέσοι όροι τριών και επτά ημερών για την αποτύπωση βραχυπρόθεσμων τάσεων της χρονοσειράς, καθώς και κυκλικά χαρακτηριστικά που βασίζονται στις συναρτήσεις ημιτόνου και συνημιτόνου της ημέρας του έτους, ώστε να αναπαρασταθεί η ετήσια εποχικότητα που εμφανίζεται στο σύνολο δεδομένων.

#### 4.4 Κατασκευή και Εκπαίδευση Μοντέλων

Για την ανάπτυξη των μοντέλων τα δεδομένα χωρίστηκαν σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου, διατηρώντας τη χρονική σειρά των παρατηρήσεων. Τα πρώτα



Εικόνα 11 Διαχωρισμός της χρονοσειράς σε σύνολα εκπαίδευσης, επικύρωσης και ελέγχου.

6,5 έτη (01/01/2017 – 30/06/2023) χρησιμοποιήθηκαν ως σύνολο εκπαίδευσης (training set), το επόμενο έτος (01/07/2023 – 30/06/2024) ως σύνολο επικύρωσης (validation set) για τη ρύθμιση των υπερπαραμέτρων των μοντέλων και το τελευταίο

έτος (01/07/2024 – 30/06/2025) ως σύνολο ελέγχου (test set) για την τελική αξιολόγηση της απόδοσής τους. Ο διαχωρισμός αυτός εξασφαλίζει ότι η αξιολόγηση πραγματοποιείται σε δεδομένα που δεν έχουν χρησιμοποιηθεί κατά την εκπαίδευση ή τη ρύθμιση των μοντέλων.

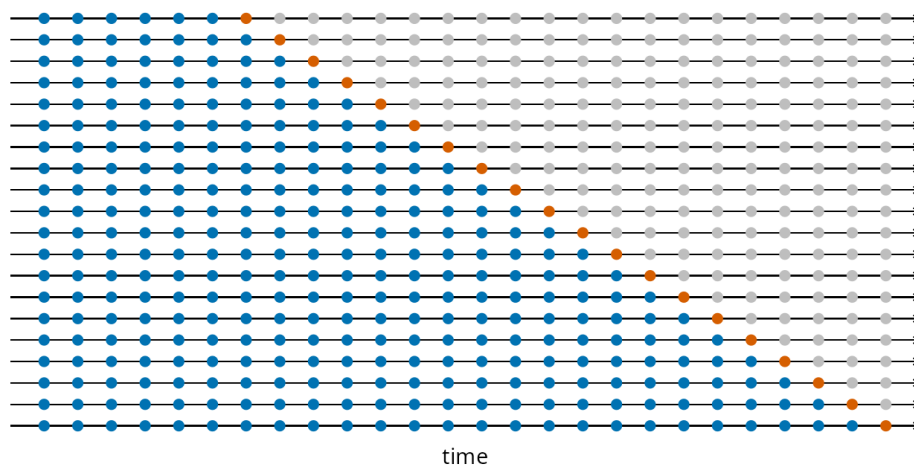
#### **4.4.1 Μπεϋζιανή Αναζήτηση και Διασταυρούμενη Επικύρωση για Χρονοσειρές**

Η διαδικασία επιλογής υπερπαραμέτρων βασίζεται στον συνδυασμό Μπεϋζιανής αναζήτησης και διασταυρούμενης επικύρωσης για χρονοσειρές με επεκτεινόμενο σύνολο εκπαίδευσης (expanding window cross-validation).

Συγκεκριμένα, η Μπεϋζιανή αναζήτηση χρησιμοποιείται για την αποδοτική εξερεύνηση του χώρου των υπερπαραμέτρων κάθε μοντέλου και την επιλογή των συνδυασμών που ελαχιστοποιούν το σφάλμα πρόβλεψης. Για κάθε προτεινόμενο συνδυασμό υπερπαραμέτρων, εφαρμόζεται διαδικασία διασταυρούμενης επικύρωσης με επεκτεινόμενο παράθυρο. Αρχικά, το μοντέλο εκπαιδεύεται στο αρχικό σύνολο εκπαίδευσης και στη συνέχεια πραγματοποιεί προβλέψεις για τον καθορισμένο ορίζοντα πρόβλεψης στο αμέσως επόμενο τμήμα του συνόλου επικύρωσης. Έπειτα, το σύνολο εκπαίδευσης επεκτείνεται ώστε να συμπεριλάβει τις παρατηρήσεις που μόλις χρησιμοποιήθηκαν για την αξιολόγηση και το μοντέλο εκπαιδεύεται εκ νέου στα διευρυμένα δεδομένα. Η διαδικασία αυτή επαναλαμβάνεται διαδοχικά μέχρι να καλυφθεί πλήρως το σύνολο επικύρωσης.

Σε κάθε βήμα, οι προβλέψεις συγκρίνονται με τις πραγματικές τιμές και υπολογίζεται το αντίστοιχο σφάλμα πρόβλεψης. Η συνολική απόδοση κάθε συνδυασμού υπερπαραμέτρων προκύπτει ως ο μέσος όρος των σφαλμάτων σε όλα τα folds του συνόλου επικύρωσης. Η πληροφορία αυτή επιστρέφεται στη Μπεϋζιανή αναζήτηση, η οποία, αξιοποιώντας τα αποτελέσματα των προηγούμενων δοκιμών, προτείνει νέους και βελτιωμένους συνδυασμούς υπερπαραμέτρων. Η διαδικασία συνεχίζεται έως την εύρεση του συνδυασμού υπερπαραμέτρων που επιτυγχάνει τη βέλτιστη απόδοση.

Μετά την ολοκλήρωση της διαδικασίας βελτιστοποίησης, το μοντέλο με τις βέλτιστες υπερπαραμέτρους αξιολογείται στο σύνολο ελέγχου (test set) με αντίστοιχη διαδικασία walk-forward validation, προκειμένου να εκτιμηθεί η γενικευτική του ικανότητα σε μη ορατά δεδομένα.



Εικόνα 12 Διαδικασία διασταυρούμενης επικύρωσης για χρονοσειρές (time series cross-validation) με επεκτεινόμενο σύνολο εκπαίδευσης (Hyndman & Athanasopoulos, 2021).

#### 4.4.2 Μεθοδολογία ARIMA

Ένα από τα βασικά χαρακτηριστικά των ημερήσιων συγκεντρώσεων PM2.5 είναι η παρουσία ετήσιας εποχικότητας περιόδου 365 ημερών. Η ορθή μοντελοποίηση της εποχικότητας αυτής θα απαιτούσε τη χρήση του μοντέλου SARIMA με εποχική περίοδο  $S=365$ . Ωστόσο, ένα τέτοιο μοντέλο είναι υπολογιστικά εξαιρετικά απαιτητικό, καθώς η εκτίμηση εποχικών παραμέτρων για περίοδο 365 ημερών καθιστά την εκπαίδευση πρακτικά ανέφικτη.

Για τον λόγο αυτό, εφαρμόστηκε εποχική διαφοροποίηση (seasonal differencing) με υστέρηση 365 ημερών, δηλαδή υπολογίστηκαν οι διαφορές  $y_t - y_{t-365}$ , αφαιρώντας έτσι την ετήσια εποχική συνιστώσα από τη χρονοσειρά. Οι έλεγχοι στασιμότητας Dickey-Fuller (ADF) και Kwiatkowski-Phillips-Schmidt-Shin (KPSS) έδειξαν ότι η χρονοσειρά είναι στάσιμη, χωρίς παρουσία συστηματικής τάσης, επομένως η παράμετρος  $d$  του ARIMA τίθεται ίση με 0.

Στη συνέχεια, για τον προσδιορισμό των παραμέτρων  $p$  και  $q$  του μοντέλου, αξιοποιήθηκαν οι συναρτήσεις αυτοσυσχέτισης (ACF) και μερικής αυτοσυσχέτισης (PACF) της διαφοροποιημένης σειράς, οι οποίες παρείχαν ενδεικτικά εύρη αναζήτησης για τις παραμέτρους. Βάσει αυτών πραγματοποιήθηκε αναζήτηση πλέγματος (grid search) για την επιλογή του βέλτιστου συνδυασμού  $(p, q)$ , σε συνδυασμό με χρονική διασταυρούμενη επικύρωση με επεκτεινόμενο παράθυρο (expanding window).

Για κάθε συνδυασμό παραμέτρων, το ARIMA μοντέλο προσαρμόζεται (fit) στο εκάστοτε training window, όπου εκτιμώνται οι συντελεστές του με βάση τα διαθέσιμα

δεδομένα, και στη συνέχεια πραγματοποιούνται προβλέψεις στον καθορισμένο ορίζοντα πρόβλεψης. Το σφάλμα πρόβλεψης υπολογίζεται σε κάθε fold και το τελικό score κάθε συνδυασμού προκύπτει ως ο μέσος όρος των σφαλμάτων όλων των folds του validation set. Επιλέγεται ο συνδυασμός παραμέτρων με τη βέλτιστη απόδοση.

Στη συνέχεια, το επιλεγμένο μοντέλο αξιοποιείται ως τελικό και αξιολογείται στο test set με walk-forward validation και επεκτεινόμενο παράθυρο, όπου το μοντέλο επαναπροσαρμόζεται σε κάθε βήμα με χρήση όλων των διαθέσιμων δεδομένων έως εκείνη τη χρονική στιγμή και πραγματοποιεί πρόβλεψη στον αντίστοιχο ορίζοντα. Οι προβλέψεις επαναφέρονται στην αρχική κλίμακα με την προσθήκη της τιμής  $y_{t-365}$ , αποκαθιστώντας έτσι την αφαιρεθείσα εποχική συνιστώσα.

#### 4.5 Μέθοδοι Αξιολόγησης Προγνωστικών Μοντέλων

Μετά την ολοκλήρωση της διαδικασίας ρύθμισης των υπερπαραμέτρων, πραγματοποιήθηκε η αξιολόγηση της απόδοσης των μοντέλων. Η διαδικασία αυτή εφαρμόστηκε με μηχανισμό αντίστοιχο με αυτόν της διασταυρούμενης επικύρωσης για χρονοσειρές. Σε κάθε χρονικό σημείο το μοντέλο εκπαιδεύεται χρησιμοποιώντας τις διαθέσιμες παρατηρήσεις μέχρι εκείνη τη στιγμή και στη συνέχεια παράγει προβλέψεις για τον προκαθορισμένο ορίζοντα πρόβλεψης. Έπειτα η διαδικασία μετακινείται χρονικά προς τα εμπρός, ενσωματώνοντας τις νέες παρατηρήσεις στα διαθέσιμα δεδομένα και επαναλαμβάνοντας την πρόβλεψη για τον επόμενο χρονικό ορίζοντα.

Για την ποσοτική εκτίμηση της απόδοσης των προγνωστικών μοντέλων χρησιμοποιήθηκαν μετρικές σφάλματος που συγκρίνουν τις προβλεπόμενες τιμές με τις πραγματικές παρατηρήσεις. Το Μέσο Απόλυτο Σφάλμα (MAE) ορίζεται ως ο μέσος όρος των απόλυτων διαφορών μεταξύ των πραγματικών τιμών και των προβλεπόμενων τιμών.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Το Μέσο Τετραγωνικό Σφάλμα (MSE) υπολογίζει τον μέσο όρο των τετραγώνων των σφαλμάτων που έχει ως αποτέλεσμα οι μεγάλες αποκλίσεις να επηρεάζουν περισσότερο την τιμή της μετρικής σε σχέση με μικρότερες αποκλίσεις. Η Ρίζα του Μέσου Τετραγωνικού Σφάλματος (RMSE) προκύπτει από την τετραγωνική ρίζα του MSE και εκφράζει το σφάλμα στην ίδια κλίμακα με τη μεταβλητή στόχο.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

Τέλος, ο συντελεστής προσδιορισμού  $R^2$  μετρά το ποσοστό της διακύμανσης των πραγματικών τιμών που εξηγείται από το μοντέλο και ορίζεται ως:

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2},$$

όπου  $y_i$  είναι η πραγματική τιμή,  $\hat{y}_i$  η προβλεπόμενη τιμή,  $\bar{y}$  ο μέσος όρος των πραγματικών τιμών και  $n$  ο αριθμός των παρατηρήσεων.

#### 4.6 Τεχνικές Ρυθμίσεις και Αναπαραγωγιμότητα της Πειραματικής Διαδικασίας

Στην παρούσα υποενότητα συνοψίζονται οι βασικές τεχνικές ρυθμίσεις που ακολουθήθηκαν κατά την υλοποίηση της πειραματικής διαδικασίας, με στόχο τη διαφάνεια και τη δυνατότητα αναπαραγωγής των αποτελεσμάτων.

Η υλοποίηση πραγματοποιήθηκε στη γλώσσα προγραμματισμού Python (έκδοση 3.10.18), με χρήση των βιβλιοθηκών numpy (2.2.6), pandas (2.3.1), scikit-learn (1.7.1), statsmodels (0.14.5), skforecast (0.18.0), tensorflow (2.20.0), keras (3.11.3), optuna (4.5.0), lightgbm (4.6.0), xgboost (3.0.4) και matplotlib (3.10.5).

Για τα μοντέλα που περιλαμβάνουν διαδικασίες με τυχαιότητα κατά την εκπαίδευση, ορίστηκε σταθερή τιμή της παραμέτρου `random_state`, η οποία καθορίζει το σημείο εκκίνησης της διαδικασίας παραγωγής τυχαίων αριθμών. Με τον τρόπο αυτό, όλες οι τυχαίες επιλογές που πραγματοποιούνται κατά την εκπαίδευση, όπως η αρχικοποίηση των παραμέτρων ή η δειγματοληψία δεδομένων, ακολουθούν την ίδια ακολουθία, οδηγώντας στα ίδια αποτελέσματα κάθε φορά που εκτελείται το μοντέλο.

Η επιλογή των υπερπαραμέτρων πραγματοποιήθηκε μέσω Bayesian optimization, με στόχο την αποδοτική αναζήτηση βέλτιστων ρυθμίσεων για κάθε μοντέλο. Για κάθε αλγόριθμο ορίστηκε συγκεκριμένη υπερπαραμετρική περιοχή, η οποία περιλάμβανε παραμέτρους που σχετίζονται με τη δομή και τη διαδικασία εκπαίδευσης των μοντέλων. Επιπλέον, ο αριθμός των χρονικών υστερήσεων (lags) αντιμετωπίστηκε ως υπερπαραμέτρος, με τις τιμές 7, 14 και 21 ημερών να εξετάζονται

στη διαδικασία βελτιστοποίησης. Οι υπερπαραμετρικές περιοχές που εξετάστηκαν παρουσιάζονται συνοπτικά στον παρακάτω πίνακα.

| Μοντέλο           | Υπερπαραμετρική περιοχή                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| MLP               | hidden_layer_sizes: (16), (32), (64), (128), (32,16), (64,32), (128,64); activation: relu, tanh; solver: adam; learning_rate: adaptive; early_stopping: True; alpha: $10^{-6}$ έως $10^{-3}$ (λογαριθμική κλίμακα); learning_rate_init: $3 \times 10^{-4}$ έως $5 \times 10^{-3}$ (λογαριθμική κλίμακα); batch_size: 32, 64, 128; max_iter: 200 έως 600; tol: $10^{-4}$ έως $10^{-3}$ (λογαριθμική κλίμακα); n_iter_no_change: 10 έως 20; validation_fraction: 0.1 έως 0.2; lags: 7, 14, 21.                                                                                        |
| KNN               | n_neighbors: 1 έως 100; weights: uniform, distance; algorithm: auto, ball_tree, kd_tree, brute; leaf_size: 15 έως 60; metric: minkowski; p: 1, 2; lags: 7, 14, 21.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Random Forest     | n_estimators: 300 έως 1500, με βήμα 100; max_depth: 4 έως 24; max_features: sqrt, log2; min_samples_split: 2 έως 20; min_samples_leaf: 1 έως 30; bootstrap: True; max_samples: 0.5, 0.7, 0.9, None; criterion: squared_error, absolute_error; lags: 7, 14, 21.                                                                                                                                                                                                                                                                                                                      |
| Gradient Boosting | n_estimators: 100 έως 1500; learning_rate: 0.01 έως 0.3 (λογαριθμική κλίμακα); max_depth: 2 έως 6; min_samples_split: 2 έως 30; min_samples_leaf: 1 έως 30; subsample: 0.5 έως 1.0; max_features: None, sqrt, log2; loss: absolute_error, squared_error; min_impurity_decrease: $10^{-9}$ έως $10^{-3}$ (λογαριθμική κλίμακα); lags: 7, 14, 21.                                                                                                                                                                                                                                     |
| XGBoost           | booster: gbtrees, dart; n_estimators: 300 έως 1200, με βήμα 100; learning_rate: $10^{-2}$ έως 0.2 (λογαριθμική κλίμακα); grow_policy: depthwise, lossguide; min_child_weight: 0.5 έως 50.0 (λογαριθμική κλίμακα); gamma: 0.0 έως 10.0; reg_alpha: $10^{-8}$ έως 100.0 (λογαριθμική κλίμακα); reg_lambda: $10^{-8}$ έως 100.0 (λογαριθμική κλίμακα); subsample: 0.75 έως 1.0; colsample_bytree: 0.75 έως 1.0; tree_method: hist; max_depth: 3 έως 12; max_leaves: 16 έως 256; sample_type: uniform, weighted; normalize_type: tree, forest; rate_drop: 0.0 έως 0.3; lags: 7, 14, 21. |

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| LightGBM | boosting_type: gbd, dart; n_estimators: 500 έως 3000, με βήμα 250; learning_rate: $10^{-3}$ έως 0.2 (λογαριθμική κλίμακα); max_depth: -1, 4, 5, 7, 9, 12; num_leaves: 31 έως 255 όταν max_depth = -1, διαφορετικά 15 έως $\min(255, 2^{\max\_depth})$ ; min_child_samples: 10 έως 200; reg_alpha: $10^{-8}$ έως 10.0 (λογαριθμική κλίμακα); reg_lambda: $10^{-8}$ έως 10.0 (λογαριθμική κλίμακα); subsample: 0.6 έως 1.0; colsample_bytree: 0.6 έως 1.0; lags: 7, 14, 21. Για boosting_type = dart εξετάστηκαν επιπλέον οι drop_rate: 0.01 έως 0.3 και skip_drop: 0.0 έως 0.8. |
| LSTM     | lr: $10^{-4}$ έως $3 \times 10^{-3}$ (λογαριθμική κλίμακα); batch_size: 32, 64, 128, 256; recurrent_units: 16, 32, 48, 64, 96, 128; dense_units: [32], [64], [128], [128,64], [256,128]; dropout: 0.0 έως 0.3, με βήμα 0.05; recurrent_dropout: 0.0 έως 0.3, με βήμα 0.05; lags: 7, 14, 21.                                                                                                                                                                                                                                                                                    |
| GRU      | lr: $10^{-4}$ έως $3 \times 10^{-3}$ (λογαριθμική κλίμακα); batch_size: 32, 64, 128, 256; recurrent_units: 16, 32, 48, 64, 96, 128; dense_units: [32], [64], [128], [128,64], [256,128]; dropout: 0.0 έως 0.3, με βήμα 0.05; recurrent_dropout: 0.0 έως 0.3, με βήμα 0.05; lags: 7, 14, 21.                                                                                                                                                                                                                                                                                    |

Πίνακας 3 Περιοχές τιμών υπερπαραμέτρων που χρησιμοποιήθηκαν στη διαδικασία Μπεϋζιανής βελτιστοποίησης για κάθε μοντέλο.

#### 4.7 Περιορισμοί της μελέτης

Η παρούσα ενότητα παρουσιάζει τους βασικούς περιορισμούς της μελέτης, με στόχο την ορθή ερμηνεία των αποτελεσμάτων και την κατανόηση των ορίων εντός των οποίων αυτά ισχύουν. Ένας σημαντικός περιορισμός αφορά το ποσοστό των ελλειπών τιμών, το οποίο ανήλθε περίπου στο 17% μετά τον καθαρισμό των δεδομένων, ποσοστό που θεωρείται σχετικά υψηλό. Για την αντιμετώπισή τους εφαρμόστηκε συμπλήρωση με βάση τον μέσο όρο των διαθέσιμων τιμών που αντιστοιχούν στην ίδια χρονική στιγμή μέσα στο έτος σε διαφορετικά έτη της χρονοσειράς. Η προσέγγιση αυτή διατηρεί την εποχικότητα, ωστόσο οδηγεί στο να αποτελεί σημαντικό μέρος της χρονοσειράς εκτιμήσεις αντί πραγματικών μετρήσεων και να ενισχύονται τα επαναλαμβανόμενα εποχικά μοτίβα της σειράς.

Επιπλέον, τα μοντέλα που χρησιμοποιήθηκαν βασίστηκαν αποκλειστικά στις ιστορικές τιμές της χρονοσειράς και σε παράγωγα χαρακτηριστικά αυτών, χωρίς την ενσωμάτωση εξωγενών μεταβλητών. Η βιβλιογραφία καταδεικνύει ότι οι

μετεωρολογικές συνθήκες συνδέονται στενά με τις συγκεντρώσεις PM<sub>2.5</sub>, ιδίως στο Δελχί, με την ταχύτητα ανέμου, τη θερμοκρασία και τη βροχόπτωση να εμφανίζουν ισχυρή αρνητική συσχέτιση, την ατμοσφαιρική πίεση θετική συσχέτιση και τη σχετική υγρασία μη γραμμική σχέση που εντείνεται όταν υπερβαίνει το 50%. Ωστόσο, η συλλογή και εναρμόνιση δεδομένων από πολλαπλές και ετερογενείς πηγές, όπως μετεωρολογικοί σταθμοί, δορυφορικές παρατηρήσεις και βάσεις δεδομένων εκπομπών, συνιστά από μόνη της σημαντική πρόκληση, καθώς απαιτεί εκτεταμένη προεπεξεργασία και αυξάνει σημαντικά την πολυπλοκότητα της μεθοδολογίας. Η μη συμπερίληψη των παραγόντων αυτών αποτελεί περιορισμό της παρούσας εργασίας, καθώς η ενσωμάτωσή τους αναμένεται να βελτιώσει σημαντικά την ακρίβεια των μοντέλων.

Τέλος, η παρούσα μελέτη επικεντρώνεται αποκλειστικά στην πρόβλεψη συγκεντρώσεων PM<sub>2.5</sub> για την περιοχή του Νέου Δελχί, με τους αλγορίθμους να εκπαιδεύονται αρχικά στο χρονικό διάστημα 01/01/2017 – 30/06/2023, να επαναεκπαιδεύονται σταδιακά κατά τη φάση επικύρωσης (01/07/2023 – 30/06/2024) μέσω Walk-Forward Validation με expanding window, και να αξιολογούνται τελικά στην περίοδο 01/07/2024 – 30/06/2025. Κατά συνέπεια, τα αποτελέσματα που προκύπτουν αντικατοπτρίζουν τις συνθήκες της συγκεκριμένης περιοχής και χρονικής περιόδου και δεν γενικεύονται σε άλλους ατμοσφαιρικούς ρύπους ούτε σε άλλες γεωγραφικές περιοχές ή χρονικές περιόδους, όπου η εξέλιξη των συγκεντρώσεων διαμορφώνεται από διαφορετικούς κλιματικούς, αστικούς και βιομηχανικούς παράγοντες.

## Κεφάλαιο 5: Αποτελέσματα και Ανάλυση

Στο παρόν κεφάλαιο παρουσιάζονται και αναλύονται τα αποτελέσματα των αλγορίθμων που αναπτύχθηκαν για την πρόβλεψη των ημερήσιων συγκεντρώσεων PM<sub>2.5</sub>. Η αξιολόγηση πραγματοποιείται για δύο διαφορετικούς χρονικούς ορίζοντες, δύο και επτά ημερών, με στόχο να εκτιμηθεί η προβλεπτική ικανότητα των αλγορίθμων τόσο σε βραχυπρόθεσμες όσο και σε μακροπρόθεσμες προβλέψεις. Η σύγκριση της ακρίβειας πρόβλεψής τους βασίζεται στα μέτρα MAE, MSE, RMSE και R<sup>2</sup>. Επιπλέον, τα αποτελέσματα συγκρίνονται με απλά μοντέλα αναφοράς (baseline models), τα οποία λειτουργούν ως σημείο αναφοράς για την αξιολόγηση της προβλεπτικής ικανότητας των πιο σύνθετων αλγορίθμων. Για τη διερεύνηση του κατά πόσο οι διαφορές που παρατηρούνται μεταξύ των αλγορίθμων είναι στατιστικά σημαντικές, χρησιμοποιείται το τεστ Diebold–Mariano.

### 5.1 Ανάλυση των Προγνωστικών Αποτελεσμάτων

Ως μοντέλα αναφοράς (baseline models) χρησιμοποιήθηκαν το αφελές μοντέλο τελευταίας τιμής (naive persistence model) και το εποχικά αφελές μοντέλο (seasonal naive). Το αφελές μοντέλο τελευταίας τιμής βασίζεται στην παραδοχή ότι η μελλοντική τιμή της χρονοσειράς είναι ίση με την πιο πρόσφατη διαθέσιμη παρατήρηση. Για ορίζοντα πρόβλεψης δύο ημερών και επτά ημερών, η πρόβλεψη παραμένει σταθερή και ισούται με την τελευταία διαθέσιμη τιμή για όλες τις μελλοντικές χρονικές στιγμές του αντίστοιχου ορίζοντα. Αντίστοιχα, το εποχικά αφελές μοντέλο προβλέπει ότι η μελλοντική τιμή ισούται με την τιμή της αντίστοιχης χρονικής στιγμής του προηγούμενου έτους. Η επιλογή τους αιτιολογείται από την ευρεία χρήση τους ως σημείων αναφοράς στη βιβλιογραφία.

#### 5.1.1 Προβλέψεις Σύντομης Περιόδου

Στην ενότητα αυτή παρουσιάζονται τα αποτελέσματα των αλγορίθμων για την πρόβλεψη των συγκεντρώσεων PM<sub>2.5</sub> σε βραχυπρόθεσμο χρονικό ορίζοντα δύο ημερών. Ο συγκεκριμένος ορίζοντας είναι ιδιαίτερης πρακτικής σημασίας, καθώς επιτρέπει την έγκαιρη εκτίμηση πιθανών επεισοδίων αυξημένης ατμοσφαιρικής ρύπανσης και μπορεί να συμβάλει στη λήψη προληπτικών μέτρων για την προστασία της δημόσιας υγείας.

|         | <i>MAE</i> | <i>MSE</i> | <i>RMSE</i> | <i>R</i> <sup>2</sup> |
|---------|------------|------------|-------------|-----------------------|
| S-NAIVE | 36,24      | 4080       | 63,88       | 0,354                 |
| P-NAIVE | 24,05      | 1407       | 37,52       | 0,777                 |
| ARIMA   | 27,27      | 2107       | 45,91       | 0,666                 |
| KNN     | 22,87      | 1445       | 38,01       | 0,771                 |
| RF      | 21,96      | 1414       | 37,61       | 0,776                 |
| GB      | 22,05      | 1405       | 37,49       | 0,777                 |
| XGB     | 22,25      | 1398       | 37,39       | 0,779                 |
| LGBM    | 22,28      | 1486       | 38,55       | 0,765                 |
| MLP     | 21,70      | 1233       | 35,12       | 0,805                 |
| LSTM    | 21,55      | 1258       | 35,47       | 0,801                 |
| GRU     | 21,79      | 1295       | 35,99       | 0,795                 |

Πίνακας 4 Αποτελέσματα αξιολόγησης των μοντέλων για προβλέψεις σε χρονικό ορίζοντα 2 ημερών.

Με μια επισκόπηση των αποτελεσμάτων, παρατηρείται ότι οι αλγόριθμοι μπορούν να διακριθούν σε τρεις βασικές κατηγορίες με βάση την ακρίβεια πρόβλεψής τους ως προς τα σφάλματα και την τιμή του συντελεστή προσδιορισμού.

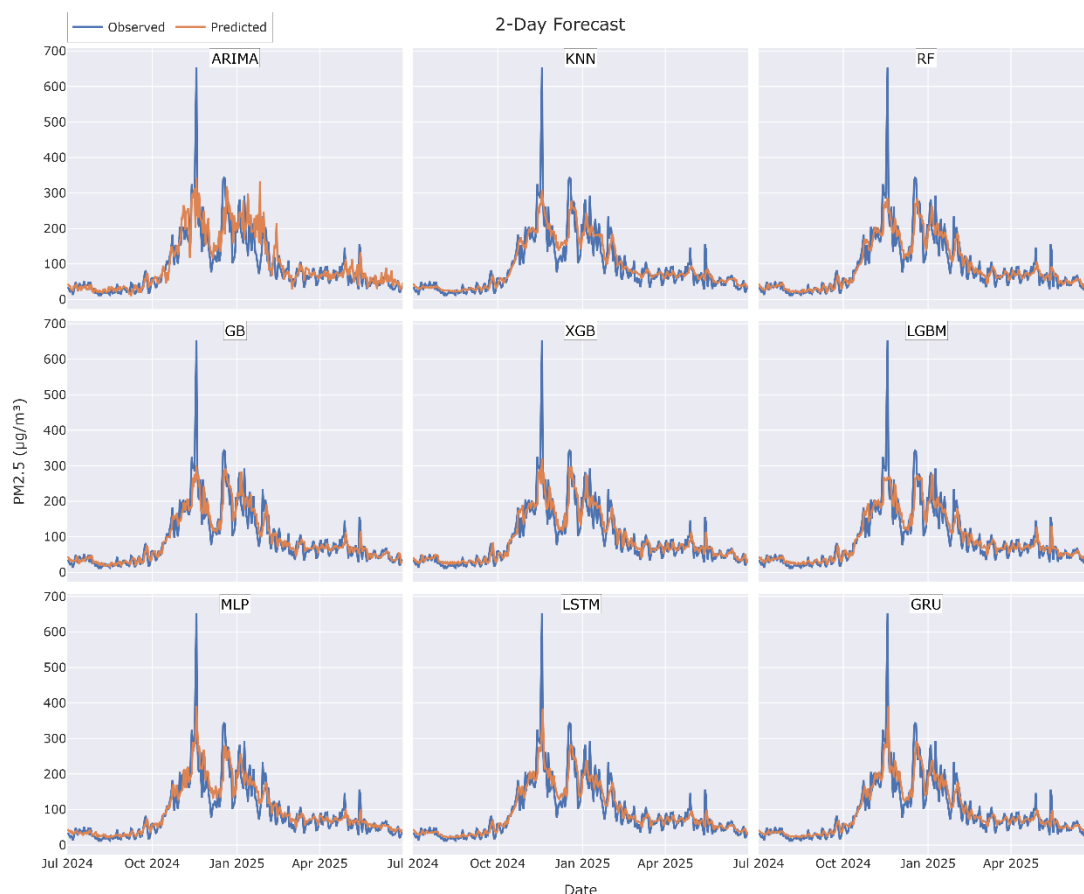
Ο αλγόριθμος ARIMA φαίνεται να υστερεί σε σχέση με τους υπόλοιπους ως προς την απόδοσή του, γεγονός που υποδηλώνει ότι αδυνατεί να αποτυπώσει αποτελεσματικά τη δυναμική και την πολυπλοκότητα των δεδομένων. Η συμπεριφορά αυτή πιθανόν σχετίζεται με τη γραμμική φύση της μεθόδου, η οποία περιορίζει την ικανότητά της να προσαρμοστεί στη μη γραμμική συμπεριφορά των δεδομένων.

Σε αντίθεση με την περιορισμένη ακρίβεια πρόβλεψης του ARIMA, οι αλγόριθμοι μηχανικής μάθησης παρουσιάζουν σαφώς βελτιωμένα αποτελέσματα. Συγκεκριμένα, τα K-Nearest Neighbors, Random Forest, Gradient Boosting, XGBoost και LightGBM επιτυγχάνουν χαμηλότερα σφάλματα και υψηλότερες τιμές  $R^2$ , επισημαίνοντας την ικανότητά τους να αποτυπώνουν πιο αποτελεσματικά τη μη γραμμική δυναμική των δεδομένων. Αξιοσημείωτο είναι ότι οι διαφορές μεταξύ των αλγορίθμων αυτής της κατηγορίας παραμένουν σχετικά μικρές, γεγονός που υποδηλώνει παρόμοια ικανότητα γενίκευσης και πρόβλεψης.

Περαιτέρω βελτίωση της ακρίβειας επιτυγχάνεται με τη χρήση των αλγορίθμων MLP, LSTM και GRU, οι οποίοι υπερτερούν έναντι όλων των εξεταζόμενων προσεγγίσεων. Μεταξύ αυτών, το MLP διακρίνεται ελαφρώς με βάση το MSE, ενώ τα LSTM και GRU ακολουθούν με εξαιρετικά μικρές αποκλίσεις.

Για να αξιολογηθεί η σημαντικότητα των παραπάνω αποτελεσμάτων, είναι απαραίτητη η σύγκρισή τους με τα μοντέλα αναφοράς, τα οποία αποκαλύπτουν ενδιαφέρουσες διαφορές. Το εποχικά αφελές μοντέλο παρουσιάζει τη χαμηλότερη

ακρίβεια πρόβλεψης συνολικά, επιβεβαιώνοντας ότι η απλή επανάληψη της αντίστοιχης τιμής του προηγούμενου έτους δεν επαρκεί για την ακριβή πρόβλεψη της χρονοσειράς. Αντίθετα, το αφελές μοντέλο τελευταίας τιμής εμφανίζει αξιοσημείωτη επίδοση, συγκρίσιμη με εκείνη των αλγορίθμων μηχανικής μάθησης, φανερώνοντας τη σημασία της βραχυχρόνιας εξάρτησης στη χρονοσειρά. Το εύρημα αυτό υπενθυμίζει ότι ακόμα και ένα απλό μοντέλο αναφοράς μπορεί να αποτελεί ισχυρό κριτήριο αξιολόγησης για πιο σύνθετες προσεγγίσεις.



Εικόνα 13 Σύγκριση των πραγματικών τιμών και των προβλέψεων των μοντέλων σε χρονικό ορίζοντα 2 ημερών.

### 5.1.2 Προβλέψεις Μακροχρόνιας Περιόδου

Για τον χρονικό ορίζοντα των επτά ημερών, παρατηρείται γενική επιδείνωση της ακρίβειας πρόβλεψης όλων των αλγορίθμων σε σχέση με τον βραχυπρόθεσμο ορίζοντα των δύο ημερών, γεγονός που είναι αναμενόμενο καθώς η αβεβαιότητα της πρόβλεψης αυξάνεται με τον χρονικό ορίζοντα. Παράλληλα, διακρίνεται και πάλι μια άτυπη κατηγοριοποίηση με βάση την ακρίβεια πρόβλεψής τους, η οποία δίνει μια πρώτη εικόνα της συμπεριφοράς τους.

|         | <i>MAE</i> | <i>MSE</i> | <i>RMSE</i> | <i>R</i> <sup>2</sup> |
|---------|------------|------------|-------------|-----------------------|
| S-NAIVE | 36,24      | 4080       | 63,88       | 0,354                 |
| P-NAIVE | 35,80      | 3797       | 61,62       | 0,399                 |
| ARIMA   | 32,64      | 2914       | 53,98       | 0,538                 |
| KNN     | 27,17      | 2075       | 45,55       | 0,671                 |
| RF      | 24,61      | 1746       | 41,78       | 0,723                 |
| GB      | 24,71      | 1624       | 40,30       | 0,743                 |
| XGB     | 26,58      | 2001       | 44,73       | 0,683                 |
| LGBM    | 24,55      | 1711       | 41,36       | 0,729                 |
| MLP     | 25,29      | 1789       | 42,30       | 0,717                 |
| LSTM    | 22,44      | 1466       | 38,29       | 0,768                 |
| GRU     | 22,70      | 1489       | 38,59       | 0,764                 |

Πίνακας 5 Αποτελέσματα αξιολόγησης των μοντέλων για προβλέψεις σε χρονικό ορίζοντα 7 ημερών.

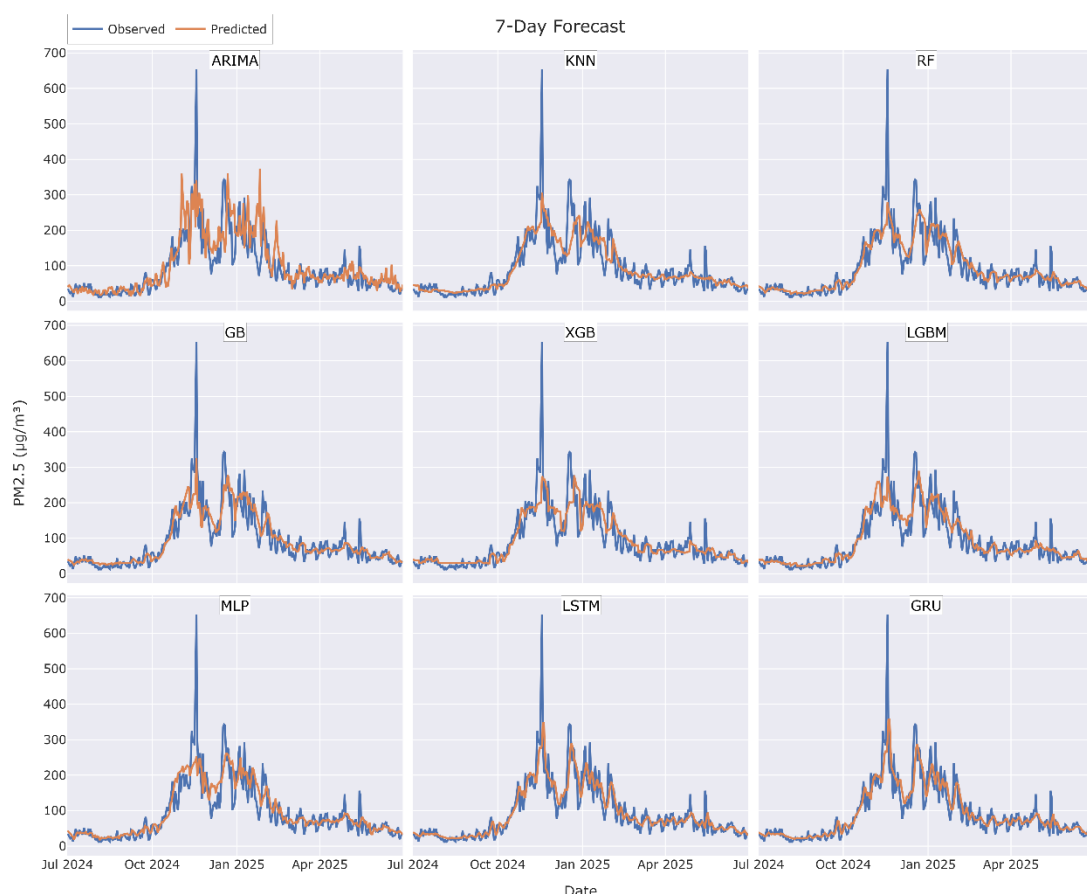
Ο αλγόριθμος ARIMA εξακολουθεί να εμφανίζει τη χαμηλότερη ακρίβεια πρόβλεψης, παρουσιάζοντας σημαντικά υψηλότερα σφάλματα σε σύγκριση με τους υπόλοιπους. Το γεγονός αυτό υποδηλώνει ότι η ακρίβειά του επιδεινώνεται περαιτέρω καθώς αυξάνεται ο χρονικός ορίζοντας, επιβεβαιώνοντας τους περιορισμούς της μεθόδου στην αποτύπωση της πολυπλοκότητας των δεδομένων.

Σε ένα ενδιάμεσο επίπεδο ακρίβειας πρόβλεψης εντάσσονται οι αλγόριθμοι K-Nearest Neighbors και XGBoost, οι οποίοι παρουσιάζουν σαφώς βελτιωμένα αποτελέσματα σε σχέση με το ARIMA, αλλά υστερούν σε σύγκριση με τις πιο αποδοτικές προσεγγίσεις.

Υψηλότερη ακρίβεια πρόβλεψης επιτυγχάνουν οι αλγόριθμοι Random Forest, Gradient Boosting και LightGBM, καθώς και ο αλγόριθμος MLP, οι οποίοι εμφανίζουν χαμηλότερα σφάλματα και υψηλότερες τιμές  $R^2$ , γεγονός που υποδηλώνει μεγαλύτερη ικανότητα πρόβλεψης για τον συγκεκριμένο χρονικό ορίζοντα.

Την υψηλότερη συνολική ακρίβεια πρόβλεψης επιτυγχάνουν τα LSTM και GRU, τα οποία καταγράφουν τις χαμηλότερες τιμές σφαλμάτων και τις υψηλότερες τιμές  $R^2$ , φανερώνοντας ότι οι αρχιτεκτονικές αυτές διατηρούν την ικανότητά τους να αποτυπώνουν χρονικές εξαρτήσεις ακόμη και σε μεγαλύτερους ορίζοντες πρόβλεψης.

Αντίθετα, τα μοντέλα αναφοράς παρουσιάζουν σαφώς χαμηλότερη ακρίβεια πρόβλεψης σε σχέση με όλους τους υπόλοιπους αλγορίθμους. Συγκεκριμένα, τόσο το εποχικά αφελές μοντέλο (S-Naive) όσο και το αφελές μοντέλο τελευταίας τιμής (P-Naive) εμφανίζουν χειρότερα αποτελέσματα, επισημαίνοντας ότι οι απλές αυτές προσεγγίσεις δεν επαρκούν για προβλέψεις σε μεγαλύτερους χρονικούς ορίζοντες.



Εικόνα 14 Σύγκριση των πραγματικών τιμών και των προβλέψεων των μοντέλων σε χρονικό ορίζοντα 7 ημερών.

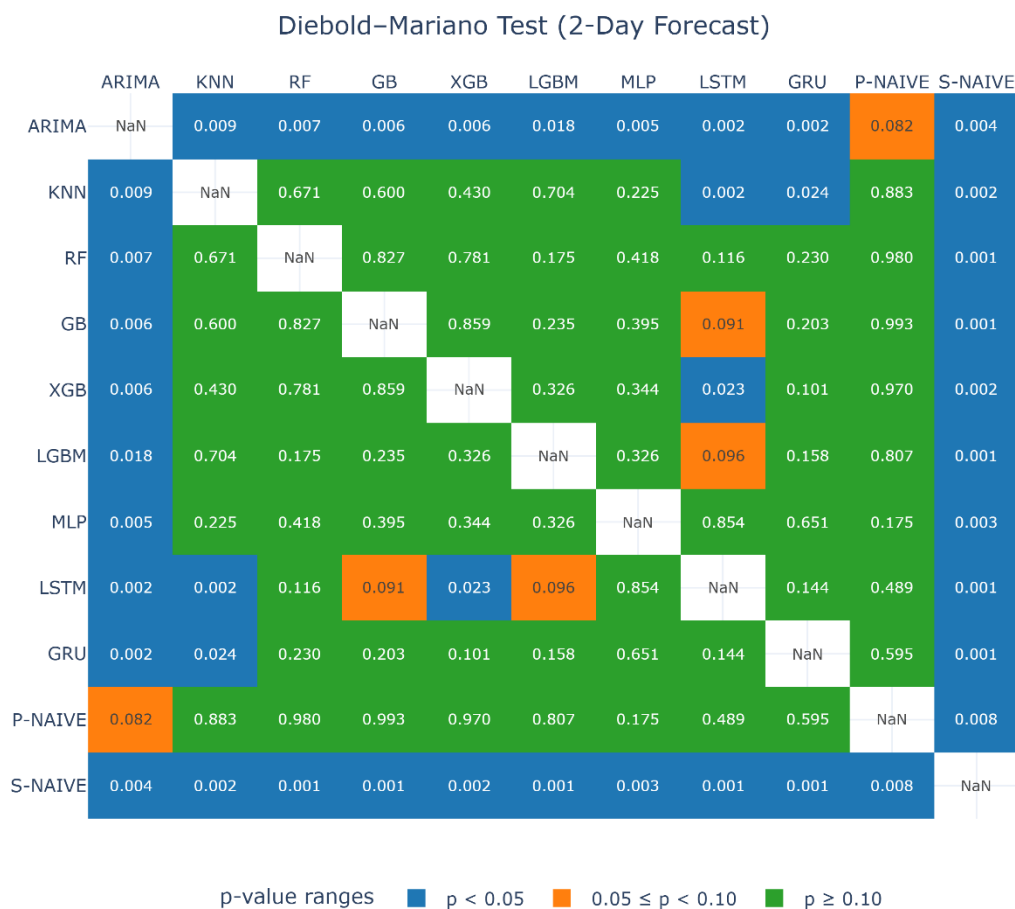
Συνολικά, η παραπάνω κατηγοριοποίηση βασίζεται σε μια αρχική, περιγραφική σύγκριση των μετρικών αξιολόγησης. Για τον λόγο αυτό, οι παρατηρούμενες διαφορές θα υποβληθούν σε στατιστικό έλεγχο στην επόμενη ενότητα, ώστε να διαπιστωθεί αν είναι και στατιστικά σημαντικές.

## 5.2 Στατιστική Αξιολόγηση

Η στατιστική ανάλυση πραγματοποιείται προκειμένου να διερευνηθεί κατά πόσο οι διαφορές που παρατηρούνται στα σφάλματα πρόβλεψης μεταξύ των αλγορίθμων είναι στατιστικά σημαντικές ή αποτέλεσμα τυχαίας διακύμανσης. Αν και από την περιγραφική ανάλυση προκύπτουν ενδείξεις για την καλύτερη ακρίβεια πρόβλεψης ορισμένων προσεγγίσεων, η απλή σύγκριση των μετρικών δεν είναι επαρκής για την εξαγωγή ασφαλών συμπερασμάτων. Για τον λόγο αυτό, εφαρμόζεται το τεστ Diebold–Mariano, το οποίο επιτρέπει τη σύγκριση των σφαλμάτων μεταξύ ζευγών αλγορίθμων και τον έλεγχο της στατιστικής σημαντικότητας των διαφορών τους.

Για την ερμηνεία των αποτελεσμάτων του τεστ Diebold–Mariano, εξετάζεται αρχικά η τιμή του p-value. Όταν το p-value είναι μικρότερο από 0,05, η διαφορά μεταξύ των σφαλμάτων πρόβλεψης των δύο αλγορίθμων θεωρείται στατιστικά σημαντική. Στην περίπτωση αυτή, λαμβάνεται υπόψη και το πρόσημο της στατιστικής τιμής (DM statistic) όπου θετική τιμή υποδηλώνει ότι ο αλγόριθμος της γραμμής εμφανίζει μεγαλύτερο σφάλμα σε σχέση με τον αλγόριθμο της στήλης, ενώ αρνητική τιμή υποδηλώνει το αντίθετο.

### 5.2.1 Προβλέψεις Σύντομης Περιόδου



Εικόνα 15 Αποτελέσματα του ελέγχου Diebold–Mariano για προβλέψεις σε χρονικό ορίζοντα 2 ημερών, με απεικόνιση των p-values για κάθε ζεύγος μοντέλων.

Από τα αποτελέσματα του τεστ Diebold–Mariano για τον χρονικό ορίζοντα των δύο ημερών προκύπτει ότι ο αλγόριθμος ARIMA διαφοροποιείται στατιστικά σημαντικά από όλους τους υπόλοιπους αλγορίθμους, εμφανίζοντας μεγαλύτερα σφάλματα πρόβλεψης. Το εύρημα αυτό επιβεβαιώνει τα συμπεράσματα της

περιγραφικής ανάλυσης και αναδεικνύει την αδυναμία της μεθόδου να ανταγωνιστεί τις υπόλοιπες προσεγγίσεις για τον συγκεκριμένο χρονικό ορίζοντα.

Μεταξύ των αλγορίθμων μηχανικής και βαθιάς μάθησης, η στατιστική ανάλυση αποκαλύπτει γενική απουσία σημαντικών διαφορών, υποδηλώνοντας ότι οι επιδόσεις τους είναι σε μεγάλο βαθμό συγκρίσιμες. Ωστόσο, εντοπίζονται ορισμένες εξαιρέσεις, με το LSTM να εμφανίζει στατιστικά σημαντικά καλύτερη ακρίβεια πρόβλεψης έναντι των KNN και XGBoost, ενώ το GRU υπερέρχει στατιστικά σημαντικά έναντι του KNN. Επιπλέον, το LSTM παρουσιάζει οριακά στατιστικά σημαντική υπεροχή έναντι των GB και LGBM. Για τους αλγορίθμους με την υψηλότερη συνολική ακρίβεια πρόβλεψης, MLP, LSTM και GRU, οι μεταξύ τους διαφορές δεν κρίνονται στατιστικά σημαντικές, επισημαίνοντας ότι αποδίδουν σε παρόμοιο επίπεδο για τον βραχυπρόθεσμο ορίζοντα.

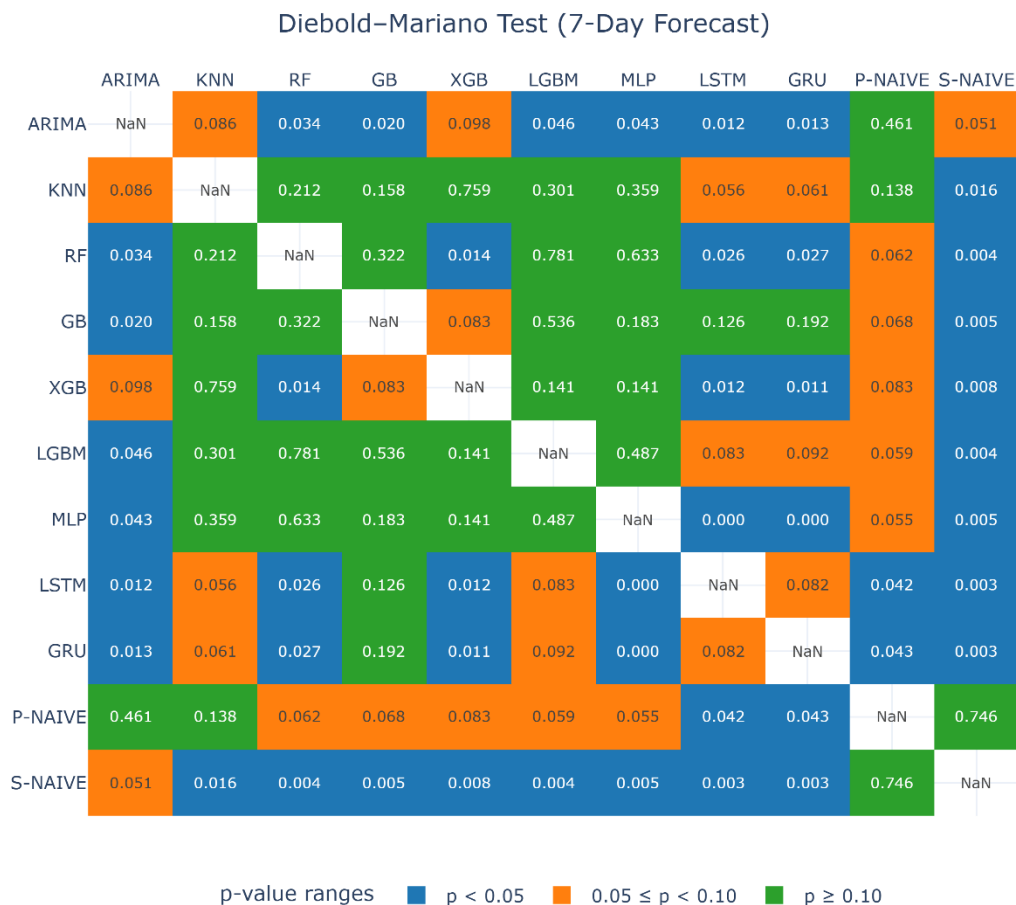
Ιδιαίτερο ενδιαφέρον παρουσιάζουν τα αποτελέσματα για τα μοντέλα αναφοράς. Το εποχικά αφελές μοντέλο (S-Naive) εμφανίζει στατιστικά σημαντικά χαμηλότερη ακρίβεια πρόβλεψης έναντι όλων των αλγορίθμων, επιβεβαιώνοντας την περιορισμένη καταλληλότητά του για τον συγκεκριμένο ορίζοντα. Αντίθετα, το αφελές μοντέλο τελευταίας τιμής (P-Naive) παρουσιάζει ένα αξιοσημείωτο εύρημα, καθώς δεν εμφανίζει στατιστικά σημαντικές διαφορές έναντι κανενός αλγορίθμου μηχανικής ή βαθιάς μάθησης, ενώ υπερέρχει στατιστικά σημαντικά έναντι του ARIMA. Το εύρημα αυτό φανερώνει ότι, για τον βραχυπρόθεσμο ορίζοντα των δύο ημερών, η απλή επανάληψη της τελευταίας διαθέσιμης τιμής αποτελεί εξίσου ανταγωνιστική προσέγγιση με τους πιο σύνθετους αλγορίθμους.

### **5.2.2 Προβλέψεις Μακροχρόνιας Περιόδου**

Για τον χρονικό ορίζοντα των επτά ημερών, από το τεστ Diebold–Mariano προκύπτει ότι ο αλγόριθμος ARIMA εμφανίζει στατιστικά σημαντικά μεγαλύτερα σφάλματα πρόβλεψης σε σύγκριση με τους περισσότερους αλγορίθμους, επιβεβαιώνοντας την περιορισμένη προβλεπτική του ικανότητα για τον συγκεκριμένο ορίζοντα. Εξαίρεση αποτελούν οι αλγόριθμοι KNN και LightGBM, για τους οποίους προκύπτει οριακή στατιστικά σημαντική διαφορά.

Σε αντίθεση με τον βραχυπρόθεσμο ορίζοντα, στον χρονικό ορίζοντα των επτά ημερών παρατηρούνται περισσότερες στατιστικά σημαντικές διαφορές μεταξύ των αλγορίθμων μηχανικής και βαθιάς μάθησης, γεγονός που υποδηλώνει μεγαλύτερη

διαφοροποίηση στην προγνωστική τους ικανότητα. Πιο συγκεκριμένα, τα LSTM και GRU εμφανίζουν στατιστικά σημαντικά καλύτερη ακρίβεια πρόβλεψης σε σύγκριση με αλγορίθμους μηχανικής μάθησης όπως το RF, το XGBoost και το MLP, ενώ στην περίπτωση των KNN και LightGBM τα p-values κυμαίνονται μεταξύ 0,05 και 0,10, υπονοώντας οριακά στατιστικά σημαντικές διαφορές.



Εικόνα 16 Αποτελέσματα του ελέγχου Diebold–Mariano για προβλέψεις σε χρονικό ορίζοντα 7 ημερών, με απεικόνιση των p-values για κάθε ζεύγος μοντέλων.

Συγκρίνοντας τους δύο κορυφαίους αλγορίθμους, LSTM και GRU, διαπιστώνεται ότι το LSTM εμφανίζει ελαφρώς υψηλότερη ακρίβεια πρόβλεψης, με τα αντίστοιχα p-values να κυμαίνονται κοντά στο επίπεδο στατιστικής σημαντικότητας, υπονοώντας οριακή υπεροχή του έναντι του GRU. Ιδιαίτερο ενδιαφέρον παρουσιάζουν και πάλι τα αποτελέσματα για τα μοντέλα αναφοράς. Το εποχικά αφελές μοντέλο (S-Naive) εμφανίζει στατιστικά σημαντικά χαμηλότερη ακρίβεια πρόβλεψης έναντι όλων των αλγορίθμων, με εξαίρεση το P-Naive, επιβεβαιώνοντας την περιορισμένη καταλληλότητά του για μακροπρόθεσμες προβλέψεις. Αντίθετα, το αφελές μοντέλο τελευταίας τιμής (P-Naive) παρουσιάζει πιο διαφοροποιημένη εικόνα σε σχέση με τον

βραχυπρόθεσμο ορίζοντα, καθώς εμφανίζει στατιστικά σημαντικές διαφορές έναντι των LSTM και GRU, ενώ οριακά στατιστικά σημαντικές διαφορές παρατηρούνται έναντι των RF, MLP, GB, XGBoost και LightGBM. Αντίθετα, δεν προκύπτουν στατιστικά σημαντικές διαφορές έναντι των ARIMA και KNN. Το εύρημα αυτό φανερώνει ότι, σε αντίθεση με τον βραχυπρόθεσμο ορίζοντα, η αύξηση του χρονικού ορίζοντα αποκαλύπτει σταδιακά τους περιορισμούς του P-Naive, το οποίο πλέον υστερεί στατιστικά σημαντικά έναντι των πιο αποδοτικών αλγορίθμων.

### 5.3 Συζήτηση και Ερμηνεία Αποτελεσμάτων

Τα αποτελέσματα της ανάλυσης αναδεικνύουν σαφείς διαφορές στην προβλεπτική ικανότητα των εξεταζόμενων προσεγγίσεων, τόσο μεταξύ των κατηγοριών αλγορίθμων όσο και μεταξύ των δύο χρονικών οριζόντων πρόβλεψης. Γενικά, η αύξηση του ορίζοντα από τις δύο στις επτά ημέρες συνοδεύεται από επιδείνωση της ακρίβειας πρόβλεψης για όλους τους αλγορίθμους, γεγονός που είναι σύμφωνο με την αναμενόμενη αύξηση της αβεβαιότητας σε μεγαλύτερους χρονικούς. Παράλληλα, η στατιστική ανάλυση μέσω του τεστ Diebold–Mariano επιβεβαιώνει ότι ορισμένες από τις παρατηρούμενες διαφορές είναι στατιστικά σημαντικές, ενισχύοντας την αξιοπιστία των συμπερασμάτων που προκύπτουν από την περιγραφική ανάλυση.

Ο αλγόριθμος ARIMA εμφανίζει σταθερά τη χαμηλότερη ακρίβεια πρόβλεψης και στους δύο χρονικούς ορίζοντες, με την απόκλισή του από τους υπόλοιπους αλγορίθμους να γίνεται πιο έντονη καθώς ο ορίζοντας αυξάνεται. Η συμπεριφορά αυτή οφείλεται σε δύο βασικούς παράγοντες. Πρώτον, η γραμμική φύση του ARIMA το καθιστά ανεπαρκές για τη μοντελοποίηση χρονοσειρών όπου η μεταβλητότητα δεν είναι σταθερή στο χρόνο. Στο Νέο Δελχί, οι συγκεντρώσεις PM2.5 παρουσιάζουν απότομες αιχμές τους χειμερινούς μήνες με διακύμανση πολύ μεγαλύτερη από τους υπόλοιπους μήνες, συμπεριφορά που ένα γραμμικό μοντέλο αδυνατεί να αποτυπώσει. Δεύτερον, παρά τη χρήση εποχικής διαφοροποίησης με υστέρηση 365 ημερών, το μοντέλο δεν αποτυπώνει πλήρως την πολυεπίπεδη εποχική δομή των δεδομένων.

Οι αλγόριθμοι μηχανικής μάθησης παρουσιάζουν σαφώς βελτιωμένη ακρίβεια πρόβλεψης σε σχέση με το ARIMA, επιβεβαιώνοντας την ικανότητά τους να αποτυπώνουν μη γραμμικές σχέσεις στα δεδομένα. Ωστόσο, οι αλγόριθμοι βαθιάς μάθησης, και ειδικότερα τα LSTM και GRU, υπερέχουν έναντι των περισσότερων αλγορίθμων μηχανικής μάθησης, με τη διαφορά αυτή να γίνεται στατιστικά σημαντική

στον 7ήμερο ορίζοντα. Η υπεροχή αυτή ερμηνεύεται από την αρχιτεκτονική των δύο μοντέλων, το LSTM και το GRU διαθέτουν μηχανισμούς πύλης που τους επιτρέπουν να διατηρούν και να αξιοποιούν πληροφορία από απομακρυσμένες χρονικές στιγμές, αποτυπώνοντας μακροχρόνιες εξαρτήσεις που οι αλγόριθμοι μηχανικής μάθησης δεν μπορούν να μοντελοποιήσουν εξίσου αποτελεσματικά. Καθώς ο ορίζοντας πρόβλεψης αυξάνεται στις επτά ημέρες, οι πληροφορίες από απομακρυσμένες χρονικές στιγμές καθίστανται πιο κρίσιμες, με αποτέλεσμα το πλεονέκτημα των αρχιτεκτονικών αυτών να αναδεικνύεται πλήρως.

Ιδιαίτερου ενδιαφέροντος είναι η συμπεριφορά του αφελούς μοντέλου τελευταίας τιμής (P-Naive), το οποίο παρουσιάζει αξιοσημείωτη ακρίβεια πρόβλεψης για τον βραχυπρόθεσμο ορίζοντα των δύο ημερών, χωρίς να διαφέρει στατιστικά σημαντικά από κανέναν αλγόριθμο μηχανικής ή βαθιάς μάθησης. Το εύρημα αυτό υποδηλώνει ότι οι συγκεντρώσεις PM2.5 εμφανίζουν ισχυρή βραχυχρόνια αυτοσυσχέτιση, με αποτέλεσμα η πρόσφατη τιμή της χρονοσειράς να αποτελεί από μόνη της ικανοποιητική πρόβλεψη για τις επόμενες δύο ημέρες. Με άλλα λόγια, οι συγκεντρώσεις PM2.5 τείνουν να μεταβάλλονται αργά βραχυπρόθεσμα, διατηρώντας παρόμοια επίπεδα από ημέρα σε ημέρα, γεγονός που μειώνει την προστιθέμενη αξία των πολύπλοκων μοντέλων για μικρούς ορίζοντες. Αντίθετα, στον 7ήμερο ορίζοντα το P-Naive υστερεί σαφώς, γεγονός που επιβεβαιώνει ότι η βραχυχρόνια αυτοσυσχέτιση εξασθενεί σε μεγαλύτερους ορίζοντες, καθιστώντας απαραίτητη τη χρήση πιο σύνθετων προσεγγίσεων που μπορούν να αξιοποιήσουν βαθύτερες χρονικές δομές στα δεδομένα.

Τα παραπάνω ευρήματα έχουν σημαντικές πρακτικές προεκτάσεις για την πρόβλεψη των συγκεντρώσεων PM2.5. Για βραχυπρόθεσμες προβλέψεις ορίζοντα δύο ημερών, αν και τα LSTM και GRU εμφανίζουν την υψηλότερη ακρίβεια πρόβλεψης, η διαφορά τους από τους υπόλοιπους αλγορίθμους δεν είναι στατιστικά σημαντική. Επομένως, όταν η υπολογιστική πολυπλοκότητα αποτελεί περιοριστικό παράγοντα, υπολογιστικά απλούστεροι αλγόριθμοι μπορούν να αποτελέσουν εξίσου αποδοτική εναλλακτική. Για μακροπρόθεσμες προβλέψεις ορίζοντα επτά ημερών, τα LSTM και GRU αναδεικνύονται ως οι πλέον κατάλληλες προσεγγίσεις, καθώς η ικανότητά τους να αποτυπώνουν μακροχρόνιες χρονικές εξαρτήσεις τους προσδίδει σαφές και στατιστικά σημαντικό πλεονέκτημα έναντι των υπόλοιπων μεθόδων.

## Κεφάλαιο 6: Συμπεράσματα και Μελλοντική Έρευνα

### 6.1 Συμπεράσματα

Η παρούσα διπλωματική εργασία είχε ως στόχο την ανάπτυξη και συγκριτική αξιολόγηση αλγορίθμων πρόβλεψης των ημερήσιων συγκεντρώσεων PM2.5 στην πόλη του Νέου Δελχί, με χρήση μεθόδων στατιστικής, μηχανικής μάθησης και βαθιάς μάθησης. Η αξιολόγηση πραγματοποιήθηκε για δύο χρονικούς ορίζοντες πρόβλεψης, δύο και επτά ημερών, με στόχο την κατανόηση των δυνατοτήτων και των περιορισμών των εξεταζόμενων προσεγγίσεων τόσο σε βραχυπρόθεσμο όσο και σε μακροπρόθεσμο επίπεδο. Για την αξιολόγηση της απόδοσης των αλγορίθμων χρησιμοποιήθηκαν τόσο μετρικές αξιολόγησης όσο και στατιστικός έλεγχος μέσω του τεστ Diebold–Mariano, επιτρέποντας την εξαγωγή αξιόπιστων συμπερασμάτων για τη συγκριτική αποτελεσματικότητα των μεθόδων.

Για τον βραχυπρόθεσμο ορίζοντα των δύο ημερών, τα αποτελέσματα αναδεικνύουν ότι η πλειονότητα των αλγορίθμων μηχανικής και βαθιάς μάθησης επιτυγχάνει συγκρίσιμη ακρίβεια πρόβλεψης, χωρίς στατιστικά σημαντικές διαφορές μεταξύ τους. Εξάιρεση αποτελεί ο αλγόριθμος ARIMA, ο οποίος υστερεί στατιστικά σημαντικά έναντι όλων των υπόλοιπων προσεγγίσεων, γεγονός που αποδίδεται στη γραμμική φύση της μεθόδου και στην αδυναμία της να αποτυπώσει τις μη γραμμικές εξαρτήσεις της χρονοσειράς. Ιδιαίτερο ενδιαφέρον παρουσιάζει η συμπεριφορά του αφελούς μοντέλου τελευταίας τιμής (P-Naive), το οποίο δεν διαφέρει στατιστικά σημαντικά από κανέναν αλγόριθμο μηχανικής ή βαθιάς μάθησης, επισημαίνοντας την υψηλή αυτοσυσχέτιση της χρονοσειράς σε μικρές υστερήσεις που χαρακτηρίζει τις συγκεντρώσεις PM2.5. Κατά συνέπεια, για τον ορίζοντα των δύο ημερών, το P-Naive αναδεικνύεται ως η πλέον αποδοτική επιλογή, καθώς επιτυγχάνει ισοδύναμη ακρίβεια με τους πολύπλοκους αλγορίθμους χωρίς υπολογιστικό κόστος εκπαίδευσης.

Για τον μακροπρόθεσμο ορίζοντα των επτά ημερών, η εικόνα διαφοροποιείται σημαντικά. Η επιδείνωση της ακρίβειας πρόβλεψης με την αύξηση του χρονικού ορίζοντα είναι αναμενόμενη. Ωστόσο, οι αρχιτεκτονικές βαθιάς μάθησης LSTM και GRU, χάρη στην ικανότητά τους να αποτυπώνουν μακροχρόνιες χρονικές εξαρτήσεις, αναδεικνύουν στατιστικά σημαντικό πλεονέκτημα έναντι των υπόλοιπων αλγορίθμων.

Αντίθετα, το αφελές μοντέλο τελευταίας τιμής (P-Naive) υστερεί πλέον σαφώς, επιβεβαιώνοντας ότι η βραχυχρόνια εξάρτηση εξασθενεί σε μεγαλύτερους ορίζοντες και η χρήση πιο σύνθετων προσεγγίσεων καθίσταται απαραίτητη.

Συνολικά, η παρούσα εργασία καταδεικνύει ότι η επιλογή του κατάλληλου αλγορίθμου πρόβλεψης εξαρτάται σε μεγάλο βαθμό από τον χρονικό ορίζοντα που εξετάζεται. Για τον ορίζοντα των δύο ημερών προτείνεται το αφελές μοντέλο τελευταίας τιμής (P-Naive) λόγω της απλότητάς του και της ισοδύναμης ακρίβειάς του με τους υπόλοιπους αλγορίθμους, ενώ για τον ορίζοντα των επτά ημερών προτείνονται τα LSTM και GRU ως οι αλγόριθμοι με την υψηλότερη ακρίβεια πρόβλεψης. Τα ευρήματα αυτά συμβάλλουν στην καλύτερη κατανόηση των δυνατοτήτων και των περιορισμών των εξεταζόμενων μεθόδων, ενώ παράλληλα προσφέρουν χρήσιμα εργαλεία για την έγκαιρη πρόβλεψη επεισοδίων ατμοσφαιρικής ρύπανσης και την υποστήριξη της λήψης αποφάσεων για την προστασία της δημόσιας υγείας.

## **6.2 Προτάσεις για Μελλοντική Έρευνα**

Η παρούσα εργασία παρουσιάζει ορισμένους περιορισμούς που πρέπει να ληφθούν υπόψη κατά την ερμηνεία των συμπερασμάτων. Η ανάλυση βασίζεται σε μονομεταβλητή χρονοσειρά, αξιοποιώντας αποκλειστικά τις ιστορικές τιμές των συγκεντρώσεων PM<sub>2.5</sub>, χωρίς να ενσωματώνει εξωγενείς παράγοντες που επηρεάζουν τη διαμόρφωσή τους. Επιπλέον, τα δεδομένα αφορούν αποκλειστικά την πόλη του Νέου Δελχί, μία περιοχή με ιδιαίτερα χαρακτηριστικά ως προς την ατμοσφαιρική ρύπανση, γεγονός που περιορίζει τη δυνατότητα γενίκευσης των αποτελεσμάτων σε άλλες γεωγραφικές περιοχές με διαφορετικά κλιματικά και αστικά χαρακτηριστικά.

Με βάση τους παραπάνω περιορισμούς, η πιο ουσιαστική κατεύθυνση για μελλοντική έρευνα αφορά την ενσωμάτωση εξωγενών μεταβλητών στην ανάλυση. Συγκεκριμένα, η προσθήκη μετεωρολογικών δεδομένων όπως θερμοκρασία, υγρασία και ταχύτητα ανέμου, ημερολογιακών χαρακτηριστικών όπως αργίες και εποχικά γεγονότα, καθώς και δεικτών κυκλοφοριακής συμφόρησης ή βιομηχανικής δραστηριότητας, αναμένεται να βελτιώσει σημαντικά την ακρίβεια πρόβλεψης. Οι παράγοντες αυτοί επηρεάζουν άμεσα τις συγκεντρώσεις PM<sub>2.5</sub> και η απουσία τους από την παρούσα ανάλυση αποτελεί έναν από τους βασικούς περιορισμούς της.

## Βιβλιογραφία

- Abuouelezz, W., Ali, N., Aung, Z., Altunaiji, A., Shah, S. B., & Gliddon, D. (2025). Exploring PM2.5 and PM10 ML forecasting models: a comparative study in the UAE. *Scientific Reports*, 15(1), 9797. <https://doi.org/10.1038/s41598-025-94013-1>
- Alawi, O. A., Kamar, H. M., Alsuwaiyan, A., & Yaseen, Z. M. (2024). Temporal trends and predictive modeling of air pollutants in Delhi: a comparative study of artificial intelligence models. *Scientific Reports*, 14(1), 30957. <https://doi.org/10.1038/s41598-024-82117-z>
- Arsov, M., Zdravevski, E., Lameski, P., Corizzo, R., Koteli, N., Gramatikov, S., Mitreski, K., & Trajkovik, V. (2021). Multi-Horizon Air Pollution Forecasting with Deep Neural Networks. *Sensors*, 21(4), 1235. <https://doi.org/10.3390/s21041235>
- Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, 54(3), 1937–1967. <https://doi.org/10.1007/s10462-020-09896-5>
- Cazzolla Gatti, R., Velichevskaya, A., Tateo, A., Amoroso, N., & Monaco, A. (2020). Machine learning reveals that prolonged exposure to air pollution is associated with SARS-CoV-2 mortality and infectivity in Italy. *Environmental Pollution*, 267(6), 115471. <https://doi.org/10.1016/j.envpol.2020.115471>
- Chen, T., & Guestrin, C. (2016). XGBoost. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. <https://doi.org/10.1145/2939672.2939785>
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734. <https://doi.org/10.3115/v1/D14-1179>

- Essamlali, I., Nhaila, H., & El Khaili, M. (2024). Supervised Machine Learning Approaches for Predicting Key Pollutants and for the Sustainable Enhancement of Urban Air Quality: A Systematic Review. *Sustainability*, 16(3), 976. <https://doi.org/10.3390/su16030976>
- Friedman, J. H. (2001). 999 REITZ LECTURE GREEDY FUNCTION APPROXIMATION: A GRADIENT BOOSTING MACHINE 1. In *The Annals of Statistics* (Vol. 29, Number 5).
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hyndman, R. J. ., & Athanasopoulos, George. (2021). *Forecasting : principles and practice*. OTexts. <https://otexts.com/fpp3/index.html>
- Jerrett, M., Burnett, R. T., Pope, C. A., Ito, K., Thurston, G., Krewski, D., Shi, Y., Calle, E., & Thun, M. (2009). Long-Term Ozone Exposure and Mortality. *New England Journal of Medicine*, 360(11), 1085–1095. <https://doi.org/10.1056/nejmoa0803894>
- Kasdagli, M. I., Orellano, P., Pérez Velasco, R., & Samoli, E. (2024). Long-Term Exposure to Nitrogen Dioxide and Ozone and Mortality: Update of the WHO Air Quality Guidelines Systematic Review and Meta-Analysis. *International Journal of Public Health*, 69, 1607676. <https://doi.org/10.3389/ijph.2024.1607676>
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (n.d.). *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*. Retrieved <https://github.com/Microsoft/LightGBM>.
- Khan, M. Y., Qayoom, A., Nizami, M. S., Siddiqui, M. S., Wasi, S., & Raazi, S. M. K.-R. (2021). Automated Prediction of Good Dictionary EXamples (GDEX): A Comprehensive Experiment with Distant Supervision, Machine Learning, and Word Embedding-Based Deep Learning Techniques. *Complexity*, 2021(1). <https://doi.org/10.1155/2021/2553199>
- Makhdoomi, A., Sarkhosh, M., & Ziaei, S. (2025). PM2.5 concentration prediction using machine learning algorithms: an approach to virtual monitoring stations. *Scientific Reports*, 15(1), 8076. <https://doi.org/10.1038/s41598-025-92019-3>

- Masood, A., & Ahmad, K. (2023). Data-driven predictive modeling of PM<sub>2.5</sub> concentrations using machine learning and deep learning techniques: a case study of Delhi, India. *Environmental Monitoring and Assessment*, 195(1), 60. <https://doi.org/10.1007/s10661-022-10603-w>
- Nielsen, M. A. (2015). *Neural Networks and Deep Learning*, Determination Press.
- Olah, C. (n.d.). *Understanding LSTM Networks*. Retrieved April 21, 2026, from <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- Open+Air+Quality+Data+Global+Landscape+2022*. (n.d.).
- Pixel\_B/Adobe Stock. (2024). *Detector for Air Quality*. World Health Organization. <https://www.who.int/data/gho/data/themes/air-pollution/submitting-official-data>
- Qaddoura, R., & Younes, M. B. (2023). Temporal prediction of traffic characteristics on real road scenarios in Amman. *Journal of Ambient Intelligence and Humanized Computing*, 14(7), 9751–9766. <https://doi.org/10.1007/s12652-022-03708-0>
- Song, Y., Liang, J., Lu, J., & Zhao, X. (2017). An efficient instance selection algorithm for k nearest neighbor regression. *Neurocomputing*, 251, 26–34. <https://doi.org/10.1016/J.NEUCOM.2017.04.018>
- Vignesh, P. P., Jiang, J. H., & Kishore, P. (2023). Predicting PM<sub>2.5</sub> Concentrations Across USA Using Machine Learning. *Earth and Space Science*, 10(10). <https://doi.org/10.1029/2023EA002911>
- WHO global air quality guidelines : particulate matter (PM<sub>2.5</sub> and PM<sub>10</sub>), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide*. (2021). WHO European Centre for Environment and Health.
- Wu, C., Wang, R., Lu, S., Tian, J., Yin, L., Wang, L., & Zheng, W. (2025). Time-Series Data-Driven PM<sub>2.5</sub> Forecasting: From Theoretical Framework to Empirical Analysis. *Atmosphere*, 16(3), 292. <https://doi.org/10.3390/atmos16030292>
- Οικονόμου, Π., & Μαλεφάκη, Σ. (2023). *Στατιστική Ανάλυση Δεδομένων*. <https://doi.org/http://dx.doi.org/10.57713/kallipos-321>

# Παράρτημα

## Κώδικας Υλοποίησης Μοντέλων

```
# General utilities
import os
import warnings
import numpy as np
import pandas as pd

# Visualization
import matplotlib.pyplot as plt

# Statistical tests and plots for time series analysis
from statsmodels.tsa.stattools import adfuller, kpss
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# Statistical time series models
from skforecast.sarimax import Sarimax
from skforecast.recursive import ForecasterSarimax
from skforecast.model_selection import backtesting_sarimax,
grid_search_sarimax
from statsmodels.tsa.statespace.sarimax import SARIMAX

# Machine learning preprocessing and evaluation
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score

# Machine learning models
from sklearn.neural_network import MLPRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.ensemble import RandomForestRegressor,
GradientBoostingRegressor
from lightgbm import LGBMRegressor
from xgboost import XGBRegressor

# Skforecast utilities for ML forecasting
from skforecast.recursive import ForecasterRecursive
from skforecast.model_selection import TimeSeriesFold,
backtesting_forecaster, bayesian_search_forecaster

# Deep learning and hyperparameter optimization
import optuna
```

```

import tensorflow as tf
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

# Skforecast utilities for deep learning forecasting
from skforecast.deep_learning import ForecasterRnn,
create_and_compile_model
from skforecast.model_selection import
backtesting_forecaster_multiseries

# Time series
df = pd.read_csv('pm25_new_delhi.csv', index_col='datetime',
parse_dates=True)
df = df.asfreq('D')

# Train and validation split dates
end_train = pd.Timestamp('2023-06-30 23:59:59')
end_validation = pd.Timestamp('2024-06-30 23:59:59')

# Train-Validation-Test Split
y = df['pm25']
y_train = y.loc[:end_train]
y_validation = y.loc[end_train:end_validation]
y_test = y.loc[end_validation:]
y_train_and_validation = y.loc[:end_validation]

# Stationarity assessment of the original series
adf_result = adfuller(y)
kpss_result = kpss(y)
print("Original series stationarity tests")
print(f"ADF p-value: {adf_result[1]}")
print(f"KPSS p-value: {kpss_result[1]}")

# Autocorrelation plot of the original series
fig, ax = plt.subplots(figsize=(10, 4))
plot_acf(y, ax=ax, lags=800, marker=None)
plt.show()

# Stationarity assessment of the training series
adf_result_train = adfuller(y_train)
kpss_result_train = kpss(y_train)
print("\nTraining series stationarity tests")
print(f"ADF p-value: {adf_result_train[1]}")
print(f"KPSS p-value: {kpss_result_train[1]}")

# Autocorrelation plot of the training series
fig, ax = plt.subplots(figsize=(10, 4))
plot_acf(y_train, ax=ax, lags=800, marker=None)

```

```

plt.show()

# Seasonal differencing
y_diff_365 = y.diff(365).dropna()

# Split the seasonally differenced series
y_train_diff_365 = y_diff_365.loc[:end_train]
y_validation_diff_365 = y_diff_365.loc[end_train:end_validation]
y_test_diff_365 = y_diff_365.loc[end_validation:]
y_train_and_validation_diff_365 = y_diff_365.loc[:end_validation]

# Stationarity assessment after seasonal differencing
adf_result_diff = adfuller(y_train_diff_365)
kpss_result_diff = kpss(y_train_diff_365)
print("\nSeasonally differenced training series stationarity tests")
print(f"ADF p-value: {adf_result_diff[1]}")
print(f"KPSS p-value: {kpss_result_diff[1]}")

# Autocorrelation plot of the seasonally differenced training series
fig, ax = plt.subplots(figsize=(10, 4))
plot_acf(y_train_diff_365, ax=ax, lags=60, marker=None)
plt.title("ACF of the seasonally differenced training series")
plt.show()

# Partial autocorrelation plot of the seasonally differenced training series
fig, ax = plt.subplots(figsize=(10, 4))
plot_pacf(y_train_diff_365, ax=ax, lags=60, method='ywm', marker=None)
plt.title("PACF of the seasonally differenced training series")
plt.show()

# Cross-Validation and Evaluation Setup for ARIMA Model
cv = TimeSeriesFold(
    steps = 2,
    initial_train_size = len(y_train_diff_365),
    refit = True,
    fixed_train_size = False,
)

evaluation_scheme_cv = TimeSeriesFold(
    steps = 2,
    initial_train_size = len(y_train_and_validation_diff_365),
    refit = True,
)

# Grid Search
forecaster_arima = ForecasterSarimax(
    regressor = Sarimax(

```

```

        order    = (1, 1, 1), # Placeholder
        maxiter  = 500
    )
)

param_grid = {
    'order': [(p, 0, q) for p in range(6) for q in range(6)],
    'seasonal_order': [(0, 0, 0, 0)],
    'trend': [None]
}

results_grid = grid_search_sarimax(
    forecaster      = forecaster_arima,
    y               = y_train_and_validation_diff_365,
    cv              = cv,
    param_grid      = param_grid,
    metric          = 'mean_absolute_error',
    return_best     = False,
    suppress_warnings_fit = True,
)

results_grid.head(5)

# Model evaluation on the test set
forecaster_arima = ForecasterSarimax(
    regressor=Sarimax(order=(3, 0, 0), seasonal_order=(0,
0, 0, 0), maxiter=500),
)

metric_arima, predictions_arima = backtesting_sarimax(
    forecaster      = forecaster_arima,
    y               = y_diff_365,
    cv              = evaluation_scheme_cv,
    metric          = 'mean_absolute_error',
    suppress_warnings_fit = True,
)

metric_arima

# Reconstruction of predictions on the original scale
y1 = predictions_arima["pred"]
y2 = y_validation.copy()
y2.index = y2.index + pd.DateOffset(years=1)
y2 = y2.groupby(level=0).mean()

```

```

arima_2d_preds = y2 + y1

# Feature engineering for ML and DL models
df['day_of_year'] = df.index.dayofyear
df['cos_doy'] = np.cos(2 * np.pi * df['day_of_year'] / 365.25)
df['sin_doy'] = np.sin(2 * np.pi * df['day_of_year'] / 365.25)
df.drop(columns=['day_of_year'], inplace=True)

df['rm_3'] = df['pm25'].rolling(3).mean().shift(1)
df['rm_7'] = df['pm25'].rolling(7).mean().shift(1)
df = df.dropna()

# Cross-Validation and Evaluation Setup for ML and DL Models
cv = TimeSeriesFold(
    steps = 2,
    initial_train_size = len(y_train),
    refit = True,
    fixed_train_size = False,
)

evaluation_scheme_cv = TimeSeriesFold(
    steps = 2,
    initial_train_size = len(y_train_and_validation),
    refit = True,
)

lags_grid = [4, 7, 14]

# Train-Validation-Test Split for ML models
y_train = df.loc[:end_train, "pm25"]
y_validation = df.loc[end_train:end_validation, "pm25"]
y_test = df.loc[end_validation:, "pm25"]
y_train_and_validation = df.loc[:end_validation, "pm25"]

exog_cols = ['cos_doy', 'sin_doy', 'rm_3', 'rm_7']
exog_train = df.loc[:end_train, exog_cols]
exog_validation = df.loc[end_train:end_validation, exog_cols]
exog_test = df.loc[end_validation:, exog_cols]
exog_train_and_validation = df.loc[:end_validation, exog_cols]

# MLP Pipeline
mlp_pipeline = Pipeline(steps=[
    ("scaler", StandardScaler()),
    ("mlp", MLPRegressor(
        random_state=42,
        early_stopping=False,

```

```

        shuffle=False
    ))
])

# MLP Forecaster
forecaster_mlp = ForecasterRecursive(
    regressor=mlp_pipeline,
    lags=10 # placeholder
)

# Bayesian search space
def search_space_mlp_small_fast(trial):

    return {
        "mlp__hidden_layer_sizes": trial.suggest_categorical(
            "mlp__hidden_layer_sizes",
            [
                (16,),
                (32,),
                (64,),
                (128,),
                (32, 16),
                (64, 32),
                (128, 64),
            ],
        ),
        "mlp__activation": trial.suggest_categorical("mlp__activation",
["relu", "tanh"]),

        "mlp__solver": trial.suggest_categorical("mlp__solver",
["adam"]),
        "mlp__learning_rate":
trial.suggest_categorical("mlp__learning_rate", ["adaptive"]),
        "mlp__early_stopping":
trial.suggest_categorical("mlp__early_stopping", [True]),

        "mlp__alpha": trial.suggest_float("mlp__alpha", 1e-6, 1e-3,
log=True),
        "mlp__learning_rate_init":
trial.suggest_float("mlp__learning_rate_init", 3e-4, 5e-3, log=True),
        "mlp__batch_size": trial.suggest_categorical("mlp__batch_size",
[32, 64, 128]),

        "mlp__max_iter": trial.suggest_int("mlp__max_iter", 200, 600),
        "mlp__tol": trial.suggest_float("mlp__tol", 1e-4, 1e-3,
log=True),
        "mlp__n_iter_no_change":
trial.suggest_int("mlp__n_iter_no_change", 10, 20),

```

```

        "mlp__validation_fraction":
trial.suggest_float("mlp__validation_fraction", 0.1, 0.2),

        "lags": trial.suggest_categorical("lags", lags_grid),
    }

# Hyperparameter tuning
results_search_mlp, frozen_trial_mlp = bayesian_search_forecaster(
    forecaster = forecaster_mlp,
    y = y_train_and_validation,
    exog = exog_train_and_validation,
    cv = cv,
    metric = 'mean_absolute_error',
    search_space = search_space_mlp_small_fast,
    n_trials = 30,
    return_best = True
)

best_params_mlp = results_search_mlp['params'].iat[0]
results_search_mlp.head()

# Model evaluation on the test set
metric_mlp, predictions_mlp = backtesting_forecaster(
    forecaster = forecaster_mlp,
    y = df['pm25'],
    exog = df[exog_cols],
    cv = evaluation_scheme_cv,
    metric = 'mean_absolute_error'
)

display(metric_mlp)
mlp_preds_2df = predictions_mlp["pred"]

# KNN Pipeline
knn_pipeline = Pipeline(steps=[
    ('scaler', StandardScaler()),
    ('knn', KNeighborsRegressor(n_jobs=-1))
])

# KNN Forecaster
forecaster_knn = ForecasterRecursive(
    regressor = knn_pipeline,
    lags = 10 # Placeholder
)

# Bayesian Search Space
def search_space_knn(trial):

```

```

    return {
        'knn__n_neighbors' : trial.suggest_int('knn__n_neighbors', 1,
100),
        'knn__weights'      : trial.suggest_categorical('knn__weights',
['uniform', 'distance']),
        'knn__algorithm'    :
trial.suggest_categorical('knn__algorithm', ['auto', 'ball_tree',
'kd_tree', 'brute']),
        'knn__leaf_size'    : trial.suggest_int('knn__leaf_size', 15,
60),
        'knn__metric'       : trial.suggest_categorical('knn__metric',
['minkowski']),
        'knn__p'            : trial.suggest_categorical('knn__p', [1,
2]),
        'lags'              : trial.suggest_categorical('lags',
lags_grid),
    }

```

#### # Hyperparameter Tuning

```

results_search_knn, frozen_trial_knn = bayesian_search_forecaster(
    forecaster = forecaster_knn,
    y          = y_train_and_validation,
    exog       = exog_train_and_validation,
    cv         = cv,
    metric     = 'mean_absolute_error',
    search_space = search_space_knn,
    n_trials   = 50,
    return_best = True
)

```

```

best_params_knn = results_search_knn['params'].iat[0]
results_search_knn.head()

```

#### # Model Evaluation On The Test Set

```

metric_knn, predictions_knn = backtesting_forecaster(
    forecaster = forecaster_knn,
    y          = df['pm25'],
    exog       = df[exog_cols],
    cv         = evaluation_scheme_cv,
    metric     = 'mean_absolute_error'
)

```

```

display(metric_knn)
knn_preds_2df = predictions_knn["pred"]

```

#### # Random Forest Forecaster

```

forecaster_rf = ForecasterRecursive(

```

```

        regressor = RandomForestRegressor(random_state=15926, n_jobs=-1),
        lags = 10
    )

# Bayesian Search Space
def search_space_rf(trial):
    return {
        "n_estimators": trial.suggest_int("n_estimators", 300, 1500,
step=100),
        "max_depth": trial.suggest_int("max_depth", 4, 24),
        "max_features": trial.suggest_categorical("max_features",
["sqrt", "log2"]),
        "min_samples_split": trial.suggest_int("min_samples_split", 2,
20),
        "min_samples_leaf": trial.suggest_int("min_samples_leaf", 1,
30),
        "bootstrap": trial.suggest_categorical("bootstrap", [True]),
        "max_samples": trial.suggest_categorical("max_samples", [0.5,
0.7, 0.9, None]),

        "criterion": trial.suggest_categorical("criterion",
["squared_error", "absolute_error"]),
        "lags": trial.suggest_categorical("lags", lags_grid),
    }

# Hyperparameter Tuning
results_search_rf, frozen_trial_rf = bayesian_search_forecaster(
    forecaster = forecaster_rf,
    y = y_train_and_validation,
    exog = exog_train_and_validation,
    cv = cv,
    metric = 'mean_absolute_error',
    search_space = search_space_rf,
    n_trials = 10,
    return_best = True
)

best_params_rf = results_search_rf['params'].iat[0]
results_search_rf.head()

# Model Evaluation On The Test Set
metric_rf, predictions_rf = backtesting_forecaster(
    forecaster = forecaster_rf,
    y = df['pm25'],
    exog = df[exog_cols],
    cv = evaluation_scheme_cv,
    metric = 'mean_absolute_error'
)

```

```

display(metric_rf)
rf_preds_2df = predictions_rf["pred"]

# Gradient Boosting Forecaster
forecaster_gb = ForecasterRecursive(
    regressor = GradientBoostingRegressor(random_state=15926),
    lags      = 10
)

# Bayesian Search Space
def search_space_gb(trial):
    return {
        'n_estimators'      : trial.suggest_int('n_estimators', 100,
1500),
        'learning_rate'     : trial.suggest_float('learning_rate',
0.01, 0.3, log=True),
        'max_depth'         : trial.suggest_int('max_depth', 2, 6),
        'min_samples_split' : trial.suggest_int('min_samples_split',
2, 30),
        'min_samples_leaf'  : trial.suggest_int('min_samples_leaf', 1,
30),
        'subsample'         : trial.suggest_float('subsample', 0.5,
1.0),
        'max_features'      :
trial.suggest_categorical('max_features', [None, 'sqrt', 'log2']),
        'loss'              : trial.suggest_categorical('loss',
['absolute_error', 'squared_error']),
        'min_impurity_decrease':
trial.suggest_float('min_impurity_decrease', 1e-9, 1e-3, log=True),

        'lags'              : trial.suggest_categorical('lags',
lags_grid),
    }

# Hyperparameter Tuning
results_search_gb, frozen_trial_gb = bayesian_search_forecaster(
    forecaster = forecaster_gb,
    y          = y_train_and_validation,
    exog       = exog_train_and_validation,
    cv         = cv,
    metric     = 'mean_absolute_error',
    search_space = search_space_gb,
    n_trials   = 10,
    return_best = True
)

```

```

best_params_gb = results_search_gb['params'].iat[0]
results_search_gb.head()

# Model Evaluation On The Test Set
metric_gb, predictions_gb = backtesting_forecaster(
    forecaster = forecaster_gb ,
    y          = df['pm25'],
    exog       = df[exog_cols],
    cv         = evaluation_scheme_cv,
    metric     = 'mean_absolute_error'
)

display(metric_gb)
gb_preds_2df = predictions_gb['pred']

# XGBoost Forecaster
forecaster_xgb = ForecasterRecursive(
    regressor = XGBRegressor(random_state=15926, verbosity=0),
    lags      = 10
)

# Bayesian Search Space
def search_space_xgb(trial):

    booster      = trial.suggest_categorical('booster', ['gbtree',
    'dart'])
    grow_policy  = trial.suggest_categorical('grow_policy',
    ['depthwise', 'lossguide'])

    tree_method  = trial.suggest_categorical('tree_method', ['hist'])

    params = {
        'booster': booster,
        'n_estimators': trial.suggest_int('n_estimators', 300, 1200,
step=100),
        'learning_rate': trial.suggest_float('learning_rate', 1e-2,
0.2, log=True),

        'grow_policy': grow_policy,
        'min_child_weight': trial.suggest_float('min_child_weight',
0.5, 50.0, log=True),
        'gamma': trial.suggest_float('gamma', 0.0, 10.0),

        'reg_alpha': trial.suggest_float('reg_alpha', 1e-8, 100.0,
log=True),
        'reg_lambda': trial.suggest_float('reg_lambda', 1e-8, 100.0,
log=True),

```

```

        'subsample': trial.suggest_float('subsample', 0.75, 1.0),
        'colsample_bytree': trial.suggest_float('colsample_bytree',
0.75, 1.0),

        'tree_method': tree_method,

        'lags': trial.suggest_categorical('lags', lags_grid),

        'max_depth': trial.suggest_int('max_depth', 3, 12),
        'max_leaves': trial.suggest_int('max_leaves', 16, 256),

        'sample_type': trial.suggest_categorical('sample_type',
['uniform', 'weighted']),
        'normalize_type': trial.suggest_categorical('normalize_type',
['tree', 'forest']),
        'rate_drop': trial.suggest_float('rate_drop', 0.0, 0.3),
    }

    if grow_policy == 'depthwise':
        params['max_leaves'] = 10_000
    else:
        params['max_depth'] = 100

    if booster != 'dart':
        params['rate_drop'] = 0.0
        params['sample_type'] = 'uniform'
        params['normalize_type'] = 'tree'

    return params

# Hyperparameter Tuning
results_search_xgb, frozen_trial_xgb = bayesian_search_forecaster(
    forecaster = forecaster_xgb,
    y = y_train_and_validation,
    exog = exog_train_and_validation,
    cv = cv,
    metric = 'mean_absolute_error',
    search_space = search_space_xgb,
    n_trials = 10,
    return_best = True
)

best_params_xgb = results_search_xgb['params'].iat[0]
results_search_xgb.head()

# Model Evaluation On The Test Set
metric_xgb, predictions_xgb = backtesting_forecaster(

```

```

    forecaster = forecaster_xgb ,
    y           = df['pm25'],
    exog        = df[exog_cols],
    cv          = evaluation_scheme_cv,
    metric      = 'mean_absolute_error'
)

display(metric_xgb)
xgb_preds_2df = predictions_xgb['pred']

# LightGBM Forecaster
forecaster_lgbm = ForecasterRecursive(
    regressor = LGBMRegressor(random_state=15926, verbose=-1),
    lags      = 10
)

# Bayesian Search Space
def search_space_lgbm(trial):

    boosting_type = trial.suggest_categorical('boosting_type', ['gbdt',
'dart'])

    max_depth = trial.suggest_categorical('max_depth', [-1, 4, 5, 7, 9,
12])

    if max_depth == -1:
        num_leaves = trial.suggest_int('num_leaves', 31, 255)
    else:
        num_leaves = trial.suggest_int(
            'num_leaves', 15, min(255, 2 ** max_depth)
        )

    params = {
        'boosting_type': boosting_type,
        'n_estimators' : trial.suggest_int('n_estimators', 500, 3000,
step=250),
        'learning_rate': trial.suggest_float('learning_rate', 1e-3,
0.2, log=True),

        'max_depth'    : max_depth,
        'num_leaves'   : num_leaves,
        'min_child_samples': trial.suggest_int('min_child_samples', 10,
200),

        'reg_alpha'    : trial.suggest_float('reg_alpha', 1e-8, 10.0,
log=True),

```

```

        'reg_lambda' : trial.suggest_float('reg_lambda', 1e-8, 10.0,
log=True),

        'subsample'      : trial.suggest_float('subsample', 0.6,
1.0),
        'colsample_bytree' : trial.suggest_float('colsample_bytree',
0.6, 1.0),

        'lags' : trial.suggest_categorical('lags', lags_grid),
    }

    if boosting_type == 'dart':
        params['drop_rate'] = trial.suggest_float('drop_rate', 0.01,
0.3)
        params['skip_drop'] = trial.suggest_float('skip_drop', 0.0,
0.8)

    return params

# Hyperparameter Tuning
results_search_lgbm, frozen_trial_lgbm = bayesian_search_forecaster(
    forecaster = forecaster_lgbm,
    y          = y_train_and_validation,
    exog       = exog_train_and_validation,
    cv         = cv,
    metric     = 'mean_absolute_error',
    search_space = search_space_lgbm,
    n_trials   = 10,
    return_best = True
)

best_params_lgbm = results_search_lgbm['params'].iat[0]
results_search_lgbm.head()

# Model Evaluation On The Test Set
metric_lgbm, predictions_lgbm = backtesting_forecaster(
    forecaster = forecaster_lgbm,
    y          = df['pm25'],
    exog       = df[exog_cols],
    cv         = evaluation_scheme_cv,
    metric     = 'mean_absolute_error'
)

display(metric_lgbm)
lgbm_preds_2df = predictions_lgbm["pred"]

# Train-Validation-Test Split for DL models
y_train = df.loc[:end_train, ["pm25"]]

```

```

y_validation = df.loc[end_train:end_validation, ["pm25"]]
y_test = df.loc[end_validation:, ["pm25"]]
y_train_and_validation = df.loc[:end_validation, ["pm25"]]

exog_cols = ['cos_doy', 'sin_doy', 'rm_3', 'rm_7']
exog_train = df.loc[:end_train, exog_cols]
exog_validation = df.loc[end_train:end_validation, exog_cols]
exog_test = df.loc[end_validation:, exog_cols]
exog_train_and_validation = df.loc[:end_validation, exog_cols]

# Reproducibility Setup
SEED = 123
np.random.seed(SEED)
tf.random.set_seed(SEED)
tf.keras.backend.clear_session()

# Custom LSTM Objective Function For Bayesian Search
def objective_lstm_2df(trial):

    # Search Space
    lr = trial.suggest_float("lr", 1e-4, 3e-3, log=True)
    batch_size = trial.suggest_categorical("batch_size", [32, 64, 128,
256])

    recurrent_units_val = trial.suggest_categorical("recurrent_units",
[16, 32, 48, 64, 96, 128])
    recurrent_units = [recurrent_units_val]

    dense_units_str = trial.suggest_categorical(
        "dense_units",
        ["[32]", "[64]", "[128]", "[128,64]", "[256,128]"]
    )
    dense_units = eval(dense_units_str)

    dropout = trial.suggest_float("dropout", 0.0, 0.3, step=0.05)
    recurrent_dropout = trial.suggest_float("recurrent_dropout", 0.0,
0.3, step=0.05)

    lags = trial.suggest_categorical("lags", [7, 14, 21])

    # Fixed Parameters
    steps = 2
    loss_name = "mae"

    np.random.seed(SEED)
    tf.random.set_seed(SEED)
    tf.keras.backend.clear_session()

```

```

# LSTM Model
lstm_model_2df = create_and_compile_model(
    series=df[["pm25"]], levels=["pm25"], lags=lags, steps=steps,
    exog=df[exog_cols],
    recurrent_layer="LSTM",
    recurrent_units=recurrent_units,
    recurrent_layers_kwargs={
        "activation": "tanh",
        "dropout": float(dropout),
        "recurrent_dropout": float(recurrent_dropout)
    },
    dense_units=dense_units,
    dense_layers_kwargs={"activation": "relu"},
    output_dense_layer_kwargs={"activation": "linear"},
    compile_kwargs={"optimizer": Adam(lr), "loss": loss_name}
)

# LSTM Forecaster
lstm_forecaster_2df = ForecasterRnn(
    regressor=lstm_model_2df, levels=["pm25"], lags=lags,
    transformer_series=StandardScaler(),
    fit_kwargs={
        "epochs": 200, "batch_size": batch_size, "shuffle": False,
        "callbacks": [
            EarlyStopping(monitor="val_loss", patience=6,
restore_best_weights=True),
            ReduceLROnPlateau(monitor="val_loss", factor=0.5,
patience=3, min_lr=1e-5),
        ],
        "series_val": y_validation,
        "exog_val": exog_validation
    },
)

# Model Fitting
lstm_forecaster_2df.fit(
    series=y_train,
    exog=exog_train
)

# Model Evaluation On The Validation Set
metrics_val, preds_val = backtesting_forecaster_multiseries(
    forecaster=lstm_forecaster_2df,
    series=y_train_and_validation[["pm25"]],
    exog=exog_train_and_validation[exog_cols],
    cv=cv,
    levels=lstm_forecaster_2df.levels,

```

```

        metric="mean_absolute_error",
        suppress_warnings=True,
        verbose=False
    )

    pred_lstm = preds_val["pred"]
    y_true = y_validation

    mae_lstm = mean_absolute_error(y_true, pred_lstm)
    return float(mae_lstm)

# Bayesian Search
study_lstm_2df = optuna.create_study(direction="minimize")
study_lstm_2df.optimize(objective_lstm_2df, n_trials=30)

print("Best MAE (LSTM 2df):", study_lstm_2df.best_value)
print("Best params (LSTM 2df):", study_lstm_2df.best_params)

# Final LSTM Model
best_lstm_2df = study_lstm_2df.best_params

np.random.seed(SEED)
tf.random.set_seed(SEED)
tf.keras.backend.clear_session()

best_recurrent_units_lstm_2df = [best_lstm_2df["recurrent_units"]]
best_dense_units_lstm_2df = eval(best_lstm_2df["dense_units"])

lstm_model_2df = create_and_compile_model(
    series=df[["pm25"]], levels=["pm25"], lags=best_lstm_2df["lags"],
    steps=2,
    exog=df[exog_cols],
    recurrent_layer="LSTM",
    recurrent_units=best_recurrent_units_lstm_2df,
    recurrent_layers_kwargs={
        "activation": "tanh",
        "dropout": float(best_lstm_2df["dropout"]),
        "recurrent_dropout": float(best_lstm_2df["recurrent_dropout"])
    },
    dense_units=best_dense_units_lstm_2df,
    dense_layers_kwargs={"activation": "relu"},
    output_dense_layer_kwargs={"activation": "linear"},
    compile_kwargs={"optimizer": Adam(best_lstm_2df["lr"]), "loss":
"mae"}
)

# Final LSTM Forecaster
lstm_forecaster_2df = ForecasterRnn(

```

```

    regressor=lstm_model_2df, levels=["pm25"],
lags=best_lstm_2df["lags"],
    transformer_series=StandardScaler(),
    fit_kwargs={
        "epochs":200, "batch_size":best_lstm_2df["batch_size"],
"shuffle":False,
        "callbacks":[
            EarlyStopping(monitor="val_loss", patience=6,
restore_best_weights=True),
            ReduceLROnPlateau(monitor="val_loss", factor=0.5,
patience=3, min_lr=1e-5),
        ],
        "series_val": y_validation,
        "exog_val": exog_validation
    },
)

# Final Model Fitting
lstm_forecaster_2df.fit(series=y_train, exog=exog_train)

# Final Training History Plot
fig, ax = plt.subplots(figsize=(8, 3))
_ = lstm_forecaster_2df.plot_history(ax=ax)
plt.title(f"Training History - BEST LSTM 2df, seed={SEED}")
plt.show()

# Model Evaluation On The Test Set
metrics_lstm_2df, predictions_2df = backtesting_forecaster_multiseries(
    forecaster      = lstm_forecaster_2df,
    series          = df[["pm25"]],
    exog            = df[exog_cols],
    cv              = evaluation_scheme_cv,
    levels          = lstm_forecaster_2df.levels,
    metric          = "mean_absolute_error",
    suppress_warnings = True,
    verbose         = False
)

metrics_lstm_2df
lstm_preds_2df = predictions_2df["pred"]

# Reproducibility Setup
import optuna
import tensorflow as tf

SEED = 123
np.random.seed(SEED)

```

```

tf.random.set_seed(SEED)
tf.keras.backend.clear_session()

# Custom GRU Objective Function For Bayesian Search
def objective_gru_2df(trial):

    # Search Space
    lr = trial.suggest_float("lr", 1e-4, 3e-3, log=True)
    batch_size = trial.suggest_categorical("batch_size", [32, 64, 128,
256])

    recurrent_units_val = trial.suggest_categorical("recurrent_units",
[16, 32, 48, 64, 96, 128])
    recurrent_units = [recurrent_units_val]

    dense_units_str = trial.suggest_categorical(
        "dense_units",
        ["[32]", "[64]", "[128]", "[128,64]", "[256,128]"]
    )
    dense_units = eval(dense_units_str)

    dropout = trial.suggest_float("dropout", 0.0, 0.3, step=0.05)
    recurrent_dropout = trial.suggest_float("recurrent_dropout", 0.0,
0.3, step=0.05)

    lags = trial.suggest_categorical("lags", [7, 14, 21])

    # Fixed Parameters
    steps = 2
    loss_name = "mae"

    np.random.seed(SEED)
    tf.random.set_seed(SEED)
    tf.keras.backend.clear_session()

    # GRU Model
    gru_model_2df = create_and_compile_model(
        series=df[["pm25"]], levels=["pm25"], lags=lags, steps=steps,
        exog=df[exog_cols],
        recurrent_layer="GRU",
        recurrent_units=recurrent_units,
        recurrent_layers_kwargs={
            "activation": "tanh",
            "dropout": float(dropout),
            "recurrent_dropout": float(recurrent_dropout)
        },
        dense_units=dense_units,
        dense_layers_kwargs={"activation": "relu"},

```

```

        output_dense_layer_kwarg={"activation": "linear"},
        compile_kwarg={"optimizer": Adam(lr), "loss": loss_name}
    )

    # GRU Forecaster
    gru_forecaster_2df = ForecasterRnn(
        regressor=gru_model_2df, levels=["pm25"], lags=lags,
        transformer_series=StandardScaler(),
        fit_kwarg={
            "epochs": 200, "batch_size": batch_size, "shuffle": False,
            "callbacks": [
                EarlyStopping(monitor="val_loss", patience=6,
restore_best_weights=True),
                ReduceLROnPlateau(monitor="val_loss", factor=0.5,
patience=3, min_lr=1e-5),
            ],
            "series_val": y_validation,
            "exog_val": exog_validation
        },
    )

    # Model Fitting
    gru_forecaster_2df.fit(
        series=y_train,
        exog=exog_train
    )

    # Model Evaluation On The Validation Set
    metrics_val, preds_val = backtesting_forecaster_multiseries(
        forecaster=gru_forecaster_2df,
        series=y_train_and_validation[["pm25"]],
        exog=exog_train_and_validation[exog_cols],
        cv=cv,
        levels=gru_forecaster_2df.levels,
        metric="mean_absolute_error",
        suppress_warnings=True,
        verbose=False
    )

    pred_gru = preds_val["pred"]
    y_true = y_validation

    mae_gru = mean_absolute_error(y_true, pred_gru)
    return float(mae_gru)

# Bayesian Search
study_gru_2df = optuna.create_study(direction="minimize")
study_gru_2df.optimize(objective_gru_2df, n_trials=30)

```

```

print("Best MAE (GRU 2df):", study_gru_2df.best_value)
print("Best params (GRU 2df):", study_gru_2df.best_params)

# Final GRU Model
best_gru_2df = study_gru_2df.best_params

np.random.seed(SEED)
tf.random.set_seed(SEED)
tf.keras.backend.clear_session()

best_recurrent_units_gru_2df = [best_gru_2df["recurrent_units"]]
best_dense_units_gru_2df = eval(best_gru_2df["dense_units"])

gru_model_2df = create_and_compile_model(
    series=df[["pm25"]], levels=["pm25"], lags=best_gru_2df["lags"],
    steps=2,
    exog=df[exog_cols],
    recurrent_layer="GRU",
    recurrent_units=best_recurrent_units_gru_2df,
    recurrent_layers_kwargs={
        "activation": "tanh",
        "dropout": float(best_gru_2df["dropout"]),
        "recurrent_dropout": float(best_gru_2df["recurrent_dropout"])
    },
    dense_units=best_dense_units_gru_2df,
    dense_layers_kwargs={"activation": "relu"},
    output_dense_layer_kwargs={"activation": "linear"},
    compile_kwargs={"optimizer": Adam(best_gru_2df["lr"]), "loss":
"mae"}
)

# Final GRU Forecaster
gru_forecaster_2df = ForecasterRnn(
    regressor=gru_model_2df, levels=["pm25"],
    lags=best_gru_2df["lags"],
    transformer_series=StandardScaler(),
    fit_kwargs={
        "epochs": 200, "batch_size": best_gru_2df["batch_size"],
"shuffle": False,
        "callbacks": [
            EarlyStopping(monitor="val_loss", patience=6,
restore_best_weights=True),
            ReduceLROnPlateau(monitor="val_loss", factor=0.5,
patience=3, min_lr=1e-5),
        ],
        "series_val": y_validation,
        "exog_val": exog_validation
    )

```

```

    },
)

# Final Model Fitting
gru_forecaster_2df.fit(series=y_train, exog=exog_train)

# Final Training History Plot
fig, ax = plt.subplots(figsize=(8, 3))
_ = gru_forecaster_2df.plot_history(ax=ax)
plt.title(f"Training History - BEST GRU 2df, seed={SEED}")
plt.show()

# Model Evaluation On The Test Set
metrics_gru_2df, predictions_2df = backtesting_forecaster_multiseries(
    forecaster      = gru_forecaster_2df,
    series          = df[["pm25"]],
    exog            = df[exog_cols],
    cv              = evaluation_scheme_cv,
    levels          = gru_forecaster_2df.levels,
    metric          = "mean_absolute_error",
    suppress_warnings = True,
    verbose         = False
)

metrics_gru_2df
gru_preds_2df = predictions_2df["pred"]

```