



ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΧΗΜΕΙΑΣ

ΠΜΣ «ΧΗΜΕΙΑ- ΣΥΧΡΟΝΕΣ ΤΕΧΝΟΛΟΓΙΕΣ ΑΝΑΛΥΤΙΚΗΣ ΚΑΙ
ΠΕΡΙΒΑΛΛΟΝΤΙΚΗΣ ΧΗΜΕΙΑΣ»

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Δημιουργία μοντέλου – λογισμικού για την πρόβλεψη της ατμοσφαιρικής
ρύπανσης βασισμένο σε ιστορικά δεδομένα ατμοσφαιρικής ρύπανσης και
καιρικών συνθηκών με χρήση αλγορίθμων μηχανικής μάθησης**

ΝΤΟΒΑ ΔΗΜΗΤΡΑ

A.M.: 1145

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ

1. **Α. Γκάμας, Αναπλ. Καθηγητής, Τμήμα Χημείας, Πανεπιστήμιο Ιωαννίνων
(Επιβλέπων)**
2. **Κ. Βλάχος, Καθηγητής, Τμήμα Χημείας, Πανεπιστήμιο Ιωαννίνων (Μέλος)**
3. **Α. Καλαμπούνιας, Αναπλ. Καθηγητής, Τμήμα Χημείας, Πανεπιστήμιο Ιωαννίνων
(Μέλος)**



UNIVERSITY OF IOANNINA

DEPARTMENT OF CHEMISTRY

MASTER'S THESIS

Creation of a Model - Software for Predicting Air Pollution, Based on Historical Air Pollution and Weather Data, Using Machine Learning Algorithms.

NTOVA DIMITRA

Ioannina, 2025

ΕΥΧΑΡΙΣΤΙΕΣ

Ευχαριστώ τον επιβλέποντα καθηγητή κ. Απόστολο Γκάμα για την υποστήριξη και την καθοδήγηση, την οικογένεια μου για τη στήριξη και τον αδερφό μου Γιάννη για τις συμβουλές του και την βοήθεια του στις δυσκολίες.

Δημιουργία Μοντέλου – Λογισμικού για την Πρόβλεψη της Ατμοσφαιρικής Ρύπανσης, Βασισμένο σε Ιστορικά Δεδομένα Ατμοσφαιρικής Ρύπανσης και Καιρικών Συνθηκών, με Χρήση Αλγορίθμων Μηχανικής Μάθησης.

ΠΕΡΙΛΗΨΗ

Στη σημερινή εποχή οι όροι ατμοσφαιρική ρύπανση και κλιματική κρίση αποτελούν καθημερινό θέμα συζήτησης στον επιστημονικό κόσμο, την ενημέρωση και το δημόσιο διάλογο. Η ανθρωπότητα βρίσκεται σε ένα κρίσιμο σημείο που πρέπει να βρει λύση χωρίς να υπάρχει το χρονικό περιθώριο. Παρόλα αυτά, σήμερα υπάρχουν τα απαραίτητα τεχνολογικά εργαλεία τα οποία συμβάλουν στην ταχύτερη ανάλυση πληροφοριών και πρόβλεψη μελλοντικών καταστάσεων. Σκοπός της παρούσας εργασίας είναι η ανάπτυξη μοντέλων πρόβλεψης της ατμοσφαιρικής ρύπανσης σε συσχέτιση με τις καιρικές συνθήκες με τη χρήση αλγορίθμων μηχανικής μάθησης.

Μακροπρόθεσμος στόχος, να μπορούν μελλοντικά να χρησιμοποιηθούν για προβλέψεις με μικρότερο κόστος και χρόνο και να συμβάλουν στην επίλυση του προβλήματος. Για την εκπόνηση της διπλωματικής εργασίας χρησιμοποιήθηκαν ελεύθερα μετεωρολογικά δεδομένα και δεδομένα καταγραφής ρύπων ενώ για την ανάπτυξη των μοντέλων πρόγνωσης επιλέχθηκε η γλώσσα προγραμματισμού python και η βιβλιοθήκη Scikit-learn. Επίσης δοκιμάστηκε το εργαλείο prophet της meta καθώς και ένα νευρωνικό δίκτυο RNN (LSTM).

Λέξεις Κλειδιά: *Ατμοσφαιρική Ρύπανση, Αιωρούμενα Σωματίδια, Μοντελοποίηση Πρόγνωσης, Μηχανική Μάθηση.*

Creation of a Model - Software for Predicting Air Pollution, Based on Historical Air Pollution and Weather Data, Using Machine Learning Algorithms.

ABSTRACT

In today's era, the terms air pollution and climate crisis are a daily topic of discussion in the scientific field, information and public debate. Humanity is at a critical point that we must find a solution without any time limit. However, today there are the necessary technological tools that contribute to faster information analysis and prediction of future situations. The purpose of this research is to develop models for predicting air pollution in relation to weather conditions using machine learning algorithms.

The long-term goal is to be able to use forecasts in the future with less cost and time and to contribute to solving the problem. For the preparation of the diploma thesis, free meteorological data and pollutant recording data were used, while for the development of forecast models, the python programming language and the Scikit-learn library were chosen. The meta prophet tool was also tested, as well as an RNN (LSTM) neural network.

Keywords: *Air Pollution, Particulate Matter, Forecast Modeling, Machine Learning.*

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΥΧΑΡΙΣΤΙΕΣ	2
ΠΕΡΙΛΗΨΗ.....	3
ABSTRACT	4
ΠΕΡΙΕΧΟΜΕΝΑ	5
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ	7
ΚΑΤΑΛΟΓΟΣ ΓΡΑΦΗΜΑΤΩΝ	7
ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ	9
ΚΕΦΑΛΑΙΟ 2. ΑΤΜΟΣΦΑΙΡΙΚΗ ΡΥΠΑΝΣΗ	12
2.1 Η ΡΥΠΑΝΣΗ ΤΗΣ ΑΤΜΟΣΦΑΙΡΑΣ	12
2.2 ΕΠΙΠΤΩΣΕΙΣ ΑΤΜΟΣΦΑΙΡΙΚΗΣ ΡΥΠΑΝΣΗΣ.....	20
2.3 ΤΡΟΠΟΙ ΜΕΤΡΗΣΗΣ ΑΤΜΟΣΦΑΙΡΙΚΗΣ ΡΥΠΑΝΣΗΣ.....	21
2.3.1 Δειγματοληψία αέρα	21
2.3.2 Χημικές μέθοδοι μέτρησης.....	22
2.3.3 Παρακολούθηση με συστήματα αισθητήρων	22
2.3.4 Μοντελοποίηση και σύγχρονες μέθοδοι	23
2.4 Η ΑΤΜΟΣΦΑΙΡΙΚΗ ΡΥΠΑΝΣΗ ΣΤΗΝ ΕΥΡΩΠΗ & ΤΟ ΛΟΝΔΙΝΟ	23
2.4.1 Ευρώπη	23
2.4.2 Λονδίνο.....	24
2.5 ΣΧΕΣΗ ΕΠΙΠΕΔΩΝ ΡΥΠΑΝΣΗΣ ΜΕ ΚΛΙΜΑΤΙΚΟΥΣ ΚΑΙ ΚΑΙΡΙΚΟΥΣ ΠΑΡΑΓΟΝΤΕΣ.....	26
2.5.1 Μετεωρολογικές παράμετροι.....	26
2.5.2 Καιρικές συνθήκες και ρύπανση.....	28
ΚΕΦΑΛΑΙΟ 3. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ	31
3.1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ.....	31
3.2 ΚΑΤΗΓΟΡΙΕΣ ΜΟΝΤΕΛΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ.....	33
3.2.1. Μάθηση υπό επίβλεψη (Supervised Learning)	33
3.2.2. Μάθηση χωρίς επίβλεψη (Unsupervised Learning)	34
3.2.3. Ενισχυμένη μάθηση (Reinforcement Learning).....	34
3.2.4. Βαθιά μάθηση (Deep Learning).....	34
3.3 ΧΡΗΣΗ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΣΤΗΝ ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΒΛΕΨΗ ΡΥΠΑΝΣΗΣ ..	35
3.4 ΑΛΓΟΡΙΘΜΟΙ ΠΑΛΙΝΔΡΟΜΗΣΗΣ ΣΤΗΝ ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ.....	37

ΚΕΦΑΛΑΙΟ 4. ΕΡΕΥΝΗΤΙΚΟ ΜΕΡΟΣ.....	40
4.1 ΣΚΟΠΟΣ ΕΡΕΥΝΑΣ – ΜΕΘΟΔΟΛΟΓΙΑ.....	40
4.2 ΕΡΓΑΛΕΙΑ ΚΑΙ ΜΕΘΟΔΟΙ.....	42
4.3 ΑΛΓΟΡΙΘΜΟΙ.....	43
4.3.1 Random Forest	43
4.3.2. Prophet.....	44
4.3.3. RNN/LSTM	45
4.3.4 Σύγκριση.....	46
ΚΕΦΑΛΑΙΟ 5. ΥΛΟΠΟΙΗΣΗ.....	47
5.1 ΜΕΘΟΔΟΙ.....	47
5.2 ΜΕΤΑΒΛΗΤΕΣ.....	48
5.3 ΡΟΗ ΕΚΠΑΙΔΕΥΣΗΣ	50
5.3.1 Στάδια	50
5.3.2 Random Forest - Δημιουργία Χαρακτηριστικών.....	50
5.3.3 Prophet - Ανάλυση Χρονοσειρών	51
5.3.4 LSTM (Long Short-Term Memory).....	52
5.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΚΠΑΙΔΕΥΣΗΣ	53
5.5 ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ.....	58
5.6 ΓΕΝΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ	58
ΚΕΦΑΛΑΙΟ 6. ΔΟΚΙΜΕΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ.....	59
6.1 ΑΝΑΛΥΣΗ ΓΡΑΦΗΜΑΤΩΝ.....	59
6.1.1 Στοιχεία γραφημάτων.....	59
6.1.2 Ερμηνεία αποτελεσμάτων	60
6.2 RANDOM FOREST	62
6.3 ΣΗΜΑΝΤΙΚΟΙ ΠΑΡΑΓΟΝΤΕΣ.....	70
6.4 PROPHET	74
6.5 RNN-LSTM	80
6.6 ΑΛΓΟΡΙΘΜΟΣ ALL	86
ΚΕΦΑΛΑΙΟ 7. ΣΥΜΠΕΡΑΣΜΑΤΑ-ΠΡΟΤΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ	90
ΒΙΒΛΙΟΓΡΑΦΙΑ	93
A. ΞΕΝΟΓΛΩΣΣΗ	93
B. ΕΛΛΗΝΙΚΗ.....	95
ΠΑΡΑΡΤΗΜΑ	97

Π1. ΚΩΔΙΚΑΣ	97
Π2. ΔΟΜΗ & ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ	116

ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1. Σύσταση κατώτερης ατμόσφαιρας της γης. (Σαχσαμάνογλου, 1998)	13
Πίνακας 2. Επίδραση καιρικών μεταβολών στους ρύπους της ατμόσφαιρας.....	29
Πίνακας 3. Σύγκριση μοντέλων.....	46
Πίνακας 4. Συγκριτική Ανάλυση	58
Πίνακας 5. Μετρικές ρύπων στον αλγόριθμο All.	87

ΚΑΤΑΛΟΓΟΣ ΓΡΑΦΗΜΑΤΩΝ

Γράφημα 1. Random Forest PM2.5	62
Γράφημα 2. Random Forest PM10	63
Γράφημα 3. Random Forest O ₃	65
Γράφημα 4. Random Forest NO ₂	66
Γράφημα 5. Random Forest SO ₂	68
Γράφημα 6. Random Forest CO	69
Γράφημα 7. Random Forest Feature Importance PM2.5	70
Γράφημα 8. Random Forest Feature Importance PM10	70
Γράφημα 9. Random Forest Feature Importance O ₃	71
Γράφημα 10. Random Forest Feature Importance NO ₂	71
Γράφημα 11. Random Forest Feature Importance SO ₂	72
Γράφημα 12. Random Forest Feature Importance CO	72
Γράφημα 13. Prophet PM2.5	74
Γράφημα 14. Prophet PM10	75
Γράφημα 15. Prophet O ₃	76
Γράφημα 16. Prophet NO ₂	77
Γράφημα 17. Prophet SO ₂	78
Γράφημα 18. Prophet CO.....	79
Γράφημα 19. RNN-LSTM PM2.5.....	80
Γράφημα 20. RNN-LSTM PM10.....	81
Γράφημα 21. RNN-LSTM O ₃	82
Γράφημα 22. RNN-LSTM NO ₂	83
Γράφημα 23. RNN-LSTM SO ₂	84
Γράφημα 24. RNN-LSTM CO	85

ΚΕΦΑΛΑΙΟ 1. ΕΙΣΑΓΩΓΗ

Ένα από τα σημαντικότερα προβλήματα της σύγχρονης εποχής είναι η περιβαλλοντική ρύπανση που έχει άμεσες και έμμεσες επιπτώσεις στα οικοσυστήματα, τα έμβια όντα, το κλίμα και τις υποδομές. Η βιομηχανική επανάσταση αποτέλεσε ένα σημείο σταθμό στην ιστορία της ανθρωπότητας και οδήγησε σε ραγδαία τεχνολογική και επιστημονική ανάπτυξη καθώς πρόσφερε τα απαραίτητα εργαλεία για την αύξηση της παραγωγικής δυνατότητας. Έτσι οι κοινωνίες ξέφυγαν από τα παραδοσιακά επαγγέλματα για τα οποία χρησιμοποιούσαν απλά εργαλεία και ζώα και μέσα σε μικρό χρονικό διάστημα ήταν σε θέση να παράγουν μεγάλες ποσότητες προϊόντων σε ταχύτερο χρόνο και με λιγότερο κόστος. Το βιοτικό επίπεδο των ανθρώπων βελτιώθηκε σημαντικά, όμως δεν δόθηκε η απαραίτητη σημασία στη μελέτη των επιπτώσεων του «θαύματος» που βίωνε ο πλανήτης.

Οι πρώτες αρνητικές επιδράσεις άρχισαν να παρατηρούνται στις μεγάλες βιομηχανικές πόλεις της Ευρώπης όπως το Λονδίνο, όπου εντοπίστηκε έντονα το φαινόμενο της αιθαλομίχλης που προκαλούσε αναπνευστικά προβλήματα στους κατοίκους της περιοχής. Από τα μέσα του 20^{ου} αιώνα η επιστημονική κοινότητα αντιλήφθηκε τις μακροπρόθεσμες επιπτώσεις στο περιβάλλον, τον άνθρωπο και το κλίμα και ξεκίνησε προσπάθεια περιορισμού και ελέγχου της ρύπανσης που προέρχεται κυρίως από τις βιομηχανίες, την παραγωγή ενέργειας και τις μεταφορές.

Στη σημερινή εποχή παραμένει αυξημένος ο κίνδυνος για τις επιπτώσεις της περιβαλλοντικής ρύπανσης. Τις τελευταίες δεκαετίες φαίνεται πια ξεκάθαρα η βλάβη που προκλήθηκε στο κλίμα, τη βιοποικιλότητα, την υγεία ανθρώπων και άλλων οργανισμών, το φυσικό περιβάλλον, τα υδάτινα οικοσυστήματα αλλά και στο ανθρωπογενές περιβάλλον. Σε δεύτερο βαθμό επηρεάζονται τομείς όπως η οικονομία, η δημόσια υγεία, οι κατασκευές, οι πηγές ενέργειας, οι θέσεις εργασίας. Η κλιματική αλλαγή, στο βαθμό που οφείλεται στον άνθρωπο, έχει ως κύρια αίτια τους αέριους ρύπους που προκαλούν το φαινόμενο του θερμοκηπίου με αποτέλεσμα την αύξηση της μέσης θερμοκρασίας του πλανήτη. Αυτό έχει ως άμεση συνέπεια τα ακραία καιρικά φαινόμενα, την ξηρασία και την ερημοποίηση πολλών περιοχών, τη μετατόπιση ζωνών καλλιέργειας, την αλλαγή του κύκλου των εποχών στις εύκρατες ζώνες, τις αλλαγές στη

βιοποικιλότητα. Μακροπρόθεσμα αναμένονται πιο σοβαρές επιπτώσεις όπως εξαφάνιση ειδών, μαζική μετανάστευση, ανεργία καθώς και επισιτιστική κρίση.

Από το 1970 ισχύει νομοθεσία «περί καθαρού αέρα» όπου ορίζονται τα έξι κριτήρια που αφορούν τους ατμοσφαιρικούς ρύπους και τα επιτρεπόμενα όρια στον ατμοσφαιρικό αέρα ώστε αυτός να κρίνεται ποιοτικά κατάλληλος. Μεταξύ αυτών βρίσκονται και τα αιωρούμενα σωματίδια τα οποία ευθύνονται για προβλήματα στην υγεία του πληθυσμού, κυρίως αναπνευστικά και καρδιολογικά. Ακόμη συμβάλλουν σε μεγάλο βαθμό στο φαινόμενο του θερμοκηπίου και την κλιματική κρίση που αυτή τη στιγμή έχει αναχθεί σε μείζον πρόβλημα για την ανθρώπινη επιβίωση και τη διατήρηση των οικοσυστημάτων.

Τα αέρια του θερμοκηπίου και τα αιωρούμενα σωματίδια εκτός από τις ανθρωπογενείς πηγές ρύπανσης προέρχονται και από φυσικές πηγές όπως σκόνη, γύρη, εκρήξεις ηφαιστειών και άλλα γεωλογικά φαινόμενα καθώς και από αστικές ή δασικές πυρκαγιές. Η συγκέντρωση τους στον ατμοσφαιρικό αέρα εξαρτάται μεταξύ άλλων παραγόντων και από τις κλιματικές συνθήκες. Οι αέριοι ρύποι που έχουν μεγαλύτερο επιστημονικό ενδιαφέρον είναι τα οξείδια όπως το διοξείδιο του αζώτου (nitrogen dioxide), του άνθρακα (carbon dioxide) και του θείου (sulfur dioxide), το μονοξείδιο του άνθρακα (carbon monoxide), το όζον (ozone), καθώς και τα μικροσωματίδια (particulate matter) PM_{2.5} και PM₁₀. Η πρόβλεψη των επιπέδων αιωρούμενων σωματιδίων είναι απαραίτητη για άτομα με προϋπάρχοντα προβλήματα υγείας που μπορούν να επιβαρυνθούν αλλά και για την εύρεση λύσεων ώστε να περιοριστούν.

Στην παρούσα εργασία διερευνάται η δυνατότητα πρόβλεψης των επιπέδων ορισμένων τύπων αιωρούμενων σωματιδίων και αέριων ρύπων με τη χρήση αλγορίθμων μηχανικής μάθησης. Μελετάται η περιοχή του Λονδίνου με προοπτική το μοντέλο να χρησιμοποιηθεί και για περιοχές της Ελλάδας και ιδίως της Ηπείρου, όπου λόγω της μορφολογίας και του μικροκλίματος αντιμετωπίζει κυρίως το χειμώνα προβλήματα ποιότητας ατμοσφαιρικού αέρα. Η ανάπτυξη της μηχανικής μάθησης ως τομέα της πληροφορικής βελτιώνει σημαντικά το χρόνο που χρειάζεται η ανάλυση δεδομένων και προσφέρει αποτελέσματα σε σύντομο χρονικό διάστημα.

Η εργασία δομείται σε δύο μέρη. Στο πρώτο παρουσιάζονται θεωρητικά στοιχεία από την αντίστοιχη επιστημονική βιβλιογραφία για την ατμοσφαιρική ρύπανση και τη μηχανική μάθηση καθώς και αντίστοιχες μελέτες όπου χρησιμοποιήθηκαν σχετικοί αλγόριθμοι. Στο δεύτερο μέρος παρουσιάζονται η μεθοδολογία της έρευνας, η πειραματική διαδικασία, ο τρόπος λειτουργίας των

αλγορίθμων και τα βήματα που εκτελεί ο κώδικας καθώς και τα αποτελέσματα. Κλείνοντας, στο τελευταίο κεφάλαιο παρουσιάζονται τα συμπεράσματα, γίνεται σύγκριση απόδοσης μεταξύ των μοντέλων αλλά και της ευαισθησίας τους ανάλογα με το ρύπο και προτείνονται λύσεις βελτίωσης και επέκτασης της εφαρμογής τους.

ΚΕΦΑΛΑΙΟ 2. ΑΤΜΟΣΦΑΙΡΙΚΗ ΡΥΠΑΝΣΗ

Η ατμόσφαιρα της γης είναι ένα αέριο στρώμα που περιβάλλει τον πλανήτη και εκτείνεται σε ύψος που φτάνει τα 120-150 χιλιόμετρα γύρω από αυτόν. Χωρίζεται σε πέντε βασικά τμήματα κατά τα οποία μεταβάλλεται η θερμοκρασία, η πυκνότητα του αέρα και η σύσταση του. Πιο αναλυτικά το κατώτερο τμήμα είναι η τροπόσφαιρα που φτάνει περίπου στα 10 χιλιόμετρα και περιέχει τα σημαντικά για τη ζωή αέρια, νερό αλλά και σωματίδια που δεν ανήκουν στο φυσιολογικό μείγμα του αέρα. Ακολουθούν η στρατόσφαιρα όπου βρίσκεται η ζώνη του όζοντος και επίσης έχει μεγάλη σημασία καθώς έχει άμεση επίδραση στο κλίμα της γης, η μεσόσφαιρα, η θερμόσφαιρα και τελευταίο στρώμα η εξώσφαιρα. Σε φυσιολογικές συνθήκες η σύσταση της γήινης ατμόσφαιρας παρουσιάζει μικρές διακυμάνσεις που δεν επηρεάζουν σημαντικά τα έμβια όντα, κάτι που συμβαίνει όταν οι συγκεντρώσεις των βασικών στοιχείων αποκλίνουν από τις φυσιολογικές τιμές και αυξάνονται οι συγκεντρώσεις ξένων ουσιών και σωματιδίων. Κάθε αλλαγή στα στοιχεία που αποτελούν την ατμόσφαιρα μπορεί να επιδράσει στον πλανήτη με ποικίλους τρόπους (Γεντεκάκης, 1999).

2.1 Η ΡΥΠΑΝΣΗ ΤΗΣ ΑΤΜΟΣΦΑΙΡΑΣ

Στην ατμόσφαιρα παρατηρούνται και στοιχεία που δεν αποτελούν τη φυσική της σύσταση. Η παρουσία σωματιδίων και ουσιών, θορύβου και ακτινοβολίας που εντοπίζεται μπορεί να έχει τέτοια συγκέντρωση ή διάρκεια που δυνητικά να προκαλεί αρνητικές επιδράσεις στα οικοσυστήματα, τον άνθρωπο και τους ζωντανούς οργανισμούς, το κλίμα αλλά και στο μη έμβιο φυσικό και ανθρωπογενές περιβάλλον, βραχυπρόθεσμα ή μακροπρόθεσμα. Η κατάσταση αυτή ορίζεται ως *ατμοσφαιρική ρύπανση* και τα στοιχεία που την προκαλούν ονομάζονται *ρυπαντές* ή *ρύποι* (Μελάς, 2013). Οι ρύποι προέρχονται από φυσικές διεργασίες του πλανήτη, όπως μια ηφαιστειακή έκρηξη, και σε μεγάλο βαθμό ως απόρροια των δραστηριοτήτων του ανθρώπου και της αλληλεπίδρασης του με το φυσικό περιβάλλον. Πριν τη βιομηχανική επανάσταση οι άνθρωποι ελάχιστα μπορούσαν να επηρεάσουν το περιβάλλον καθώς αλληλοεπιδρούσαν με αυτό χωρίς να

το επιβαρύνουν. Οι συνήθεις ανθρωπογενείς ρύποι ήταν προϊόντα καύσης ξύλου και άλλων φυσικών υλικών. Έπειτα όμως με την άνθηση της βιομηχανίας και τη χρήση των ορυκτών καυσίμων οι ρύποι που κατέληγαν στην ατμόσφαιρα και γενικότερα στο περιβάλλον πολλαπλασιάστηκαν (Καραθανάσης, 2006). Οι μεταφορές και η θέρμανση, τα βιομηχανικά απόβλητα, τα σύγχρονα όπλα που χρησιμοποιούνται στους πολέμους, η παραγωγή ενέργειας, η γεωργία και η κτηνοτροφία αποτελούν τις βασικές πηγές ρύπων. Ακόμη η ποιότητα του ατμοσφαιρικού αέρα επηρεάζεται και από τη συγκέντρωση σωματιδίων όπως σκόνη (Γεντεκάκης, 1999).

Τα κυριότερα αέρια συστατικά της ατμόσφαιρας καθώς και η συνήθης συγκέντρωσή τους σε τροπόσφαιρα και στρατόσφαιρα φαίνονται στον παρακάτω πίνακα:

Αέριο	Μόριο	Συγκέντρωση
Άζωτο	N ₂	78,08%
Οξυγόνο	O ₂	20,95%
Αργό	Ar	0,93%
Διοξείδιο του άνθρακα	CO ₂	330 ppm
Όζον	O ₃	0-12 ppm
Νέο	Ne	18 ppm
Ήλιο	He	5 ppm
Κρυπτό	Kr	1 ppm
Υδρογόνο	H ₂	0,5 ppm

Πίνακας 1. Σύσταση κατώτερης ατμόσφαιρας της γης. (Σαχσαμάνογλου, 1998)

Οι ατμοσφαιρικοί ρύποι ταξινομούνται σε επτά βασικές κατηγορίες ανάλογα με το είδος τους και αναφέρονται αναλυτικά παρακάτω:

i) Ενώσεις Άνθρακα (carbon-containing compounds).

Οι επιβλαβείς ενώσεις του άνθρακα διοχετεύονται στην ατμόσφαιρα κυρίως από τη βιομηχανία, την παραγωγή ενέργειας, τη θέρμανση και τις μεταφορές. Επίσης από πυρκαγιές,

ηφαίστεια και δραστηριότητα ωκεανών (Καραθανάσης, 2006). Τα μόρια των ανθρακούχων ενώσεων είναι απλά όσον αφορά τη χημική τους σύσταση. Συγκεκριμένα το διοξείδιο του άνθρακα (CO_2), το μονοξείδιο του άνθρακα (CO), καθώς και οι πτητικές οργανικές ενώσεις (VOCs). Το μονοξείδιο του άνθρακα είναι εξαιρετικά επικίνδυνο και τοξικό και μπορεί να οδηγήσει ακόμα και σε θάνατο. Απελευθερώνεται στο περιβάλλον σε συνθήκες ατελούς καύσης υδρογονανθράκων σε βιομηχανικές διεργασίες και οχήματα. Σε μοριακό επίπεδο, το CO δεσμεύεται με την αιμοσφαιρίνη του αίματος, σχηματίζοντας καρβοξυ-αιμοσφαιρίνη, μια ένωση που μειώνει τη μεταφορά οξυγόνου στους ιστούς. Αυτή η διαδικασία προκαλεί κυτταρική υποξία, η οποία μπορεί να οδηγήσει σε νευρολογικές βλάβες και καρδιακά προβλήματα. (Ζερεφός, 2009, Καραθανάσης, 2006, Γεντεκάκης, 1999)

Το διοξείδιο του άνθρακα (CO_2), αν και βρίσκεται ως φυσικό συστατικό στην ατμόσφαιρα θεωρείται ρύπος όταν οι συγκεντρώσεις του αυξάνονται πέραν των φυσιολογικών τιμών λόγω ανθρωπογενών δραστηριοτήτων, όπως η καύση ορυκτών καυσίμων. Η αύξηση του CO_2 συνδέεται με την ενίσχυση του φαινομένου του θερμοκηπίου που επηρεάζει αρνητικά τα κλιματικά συστήματα. Σε επίπεδο οικοσυστημάτων οι υψηλές συγκεντρώσεις CO_2 επηρεάζουν την αλκαλικότητα των υδάτων, οδηγώντας σε αύξηση της οξύτητας και κατά συνέπεια επηρεάζει θαλάσσιους οργανισμούς (Καραθανάσης, 2006).

Οι πτητικές οργανικές ενώσεις (VOCs) αποτελούν μια ετερογενή κατηγορία ενώσεων που περιλαμβάνει αλειφατικούς, αρωματικούς και χλωριωμένους υδρογονάνθρακες. Οι VOCs συμβάλλουν στη δημιουργία τροποσφαιρικού όζοντος μέσω φωτοχημικών αντιδράσεων με οξείδια του αζώτου (NO_x). Παράλληλα ορισμένες VOCs, όπως το βενζόλιο και το τολουόλιο, είναι καρκινογόνες ενώσεις που επηρεάζουν άμεσα τη δομή του DNA και προκαλούν κυτταρικές μεταλλάξεις (Lee et al., 2020).

ii) Ενώσεις Αζώτου

Το διοξείδιο του αζώτου είναι ένας από τους κυριότερους αέριους ρύπους που εκλύονται στην ατμόσφαιρα μέσω της καύσης ορυκτών καυσίμων. Οι πηγές του περιλαμβάνουν τις εκπομπές οχημάτων, τις βιομηχανικές διεργασίες και τις μονάδες παραγωγής ενέργειας. Το NO_2 είναι σημαντικός πρόδρομος στη δημιουργία δευτερογενών ρύπων, όπως το όζον και τα αιωρούμενα σωματίδια, μέσω πολύπλοκων φωτοχημικών αντιδράσεων (Ζερεφός, 2009, Καραθανάσης, 2006, Γεντεκάκης, 1999).

Η έκθεση σε υψηλές συγκεντρώσεις NO₂ μπορεί να προκαλέσει σοβαρές αναπνευστικές δυσλειτουργίες για παράδειγμα αυξημένη ευαισθησία στις λοιμώξεις του αναπνευστικού συστήματος, μειωμένη πνευμονική λειτουργία και ανάπτυξη άσθματος σε παιδιά. Ιδιαίτερα ευάλωτες ομάδες είναι τα παιδιά, οι ηλικιωμένοι και τα άτομα με προϋπάρχουσες πνευμονολογικές παθήσεις (Lee et al., 2020). Το NO₂ μετατρέπεται σε νιτρώδη και νιτρικά άλατα εντός του οργανισμού τα οποία αλληλεπιδρούν με τις πρωτεΐνες και το DNA μέσω νιτροποίησης. Αυτές οι τροποποιήσεις επηρεάζουν τη δομή και τη λειτουργία πρωτεϊνών και ενζύμων, με συνέπεια την απώλεια της φυσιολογικής κυτταρικής λειτουργίας. Παράλληλα προάγει τη φλεγμονή στους αεραγωγούς και μειώνει την αντιοξειδωτική ικανότητα του οργανισμού.

Στο περιβάλλον συμβάλλει στην όξινη βροχή που προκαλεί εκτεταμένες βλάβες στα υδάτινα οικοσυστήματα, τα δάση και τις καλλιέργειες (European Environment Agency, 2021). Οι σύγχρονες μελέτες υπογραμμίζουν τη σημασία της υιοθέτησης καθαρών τεχνολογιών στις μεταφορές και τη βιομηχανία, ώστε να μειωθεί η εκπομπή νιτρικών και να βελτιωθεί η ποιότητα του αέρα.

iii) Θειούχες ενώσεις

Το διοξείδιο του θείου εκλύεται κυρίως από την καύση θειούχων ορυκτών καυσίμων, όπως ο άνθρακας και το πετρέλαιο. Οι κύριες πηγές περιλαμβάνουν σταθμούς παραγωγής ενέργειας, βιομηχανικές εγκαταστάσεις και διυλιστήρια πετρελαίου. Το SO₂ προκαλεί έντονο ερεθισμό στο αναπνευστικό σύστημα και έχει συσχετιστεί με αυξημένο κίνδυνο εμφάνισης άσθματος, βρογχίτιδας και ΧΑΠ (Lee et al., 2020).

Επιπλέον, το SO₂ διαδραματίζει κεντρικό ρόλο στη δημιουργία της όξινης βροχής, η οποία μειώνει τη γονιμότητα του εδάφους, προκαλεί διάβρωση στα κτίρια και υποβαθμίζει τα υδάτινα οικοσυστήματα. Στοιχεία από τον Ευρωπαϊκό Οργανισμό Περιβάλλοντος (EEA, 2021) δείχνουν ότι οι αυστηροί κανονισμοί για τον περιορισμό των εκπομπών έχουν οδηγήσει σε σημαντική μείωση των επιπέδων του στην ατμόσφαιρα τα τελευταία χρόνια. Η αντικατάσταση των θειούχων καυσίμων με καθαρότερες εναλλακτικές πηγές ενέργειας, αποτελεί βασική στρατηγική για τη μείωση των εκπομπών SO₂ και την προστασία του περιβάλλοντος (WHO, 2021).

iv) Αιωρούμενα Σωματίδια

Τα σωματίδια ποικίλουν σε μέγεθος και προέλευση, κοινό χαρακτηριστικό τους είναι μπορούν να εισέλθουν στον ανθρώπινο οργανισμό μέσω της αναπνευστικής οδού. Διαχωρίζονται ανάλογα με τη διάμετρο τους σε δυο τύπους, PM_{2.5} και PM₁₀, στους οποίους οι αεροδυναμικές διάμετροι είναι 2,5 και 10 μικρόμετρα αντίστοιχα. Στα συστατικά τους περιλαμβάνονται θειικά, νιτρικά, αμμωνία, άνθρακας, χλωριούχο νάτριο, νερό, καθώς και σκόνη προερχόμενη από το έδαφος (Ζερεφός, 2009). Οι κύριες πηγές των αιωρούμενων σωματιδίων περιλαμβάνουν τις βιομηχανικές δραστηριότητες, την καύση ορυκτών καυσίμων, τις εκπομπές από τα οχήματα και τις εγκαταστάσεις θέρμανσης, καθώς και φυσικές διεργασίες όπως τη διάβρωση του εδάφους, τις δασικές πυρκαγιές και τις ηφαιστειακές εκρήξεις (WHO, 2021). Επιπλέον, τα σωματίδια συχνά φέρουν τοξικές ουσίες όπως βαρέα μέταλλα και πολυκυκλικούς αρωματικούς υδρογονάνθρακες, οι οποίες αυξάνουν την επικινδυνότητά τους για την ανθρώπινη υγεία (Lee et al., 2020).

Η μικρή διάμετρος των PM_{2.5} τους επιτρέπει να διεισδύουν βαθιά στο αναπνευστικό σύστημα και να προκαλούν σοβαρές επιπτώσεις στην υγεία, όπως αναπνευστικές ασθένειες και καρδιαγγειακά προβλήματα. Διεισδύουν στις κυψελίδες των πνευμόνων και από εκεί στην κυκλοφορία του αίματος φτάνοντας σε απομακρυσμένα όργανα. Στο κυτταρικό επίπεδο, τα σωματίδια προκαλούν οξειδωτικό στρες μέσω της παραγωγής αντιδραστικών ειδών οξυγόνου (ROS). Αυτά τα ROS καταστρέφουν τις μεμβράνες των κυττάρων, το DNA και τις πρωτεΐνες, οδηγώντας σε φλεγμονώδεις αποκρίσεις. Επιπλέον, η παρουσία βαρέων μετάλλων στα PM_{2.5} μπορεί να αναστείλει τη λειτουργία ενζύμων, επιδεινώνοντας τις κυτταρικές δυσλειτουργίες (Lee et al., 2020).

Προκαλούν αναπνευστικά προβλήματα, βρογχίτιδες, προβλήματα σε έμβρυα και εγκύους, και είναι συχνά αιτία νοσηλείας. Η μακροχρόνια έκθεση σε υψηλά επίπεδα PM_{2.5} έχει συνδεθεί με την αύξηση της θνησιμότητας από ασθένειες όπως η χρόνια αποφρακτική πνευμονοπάθεια (ΧΑΠ), ο καρκίνος των πνευμόνων και το εγκεφαλικό επεισόδιο. Σύμφωνα με τον Παγκόσμιο Οργανισμό Υγείας (WHO, 2021), η έκθεση σε PM_{2.5} ευθύνεται για περίπου 7 εκατομμύρια πρόωρους θανάτους παγκοσμίως κάθε χρόνο, κυρίως σε χώρες μεσαίου και χαμηλού εισοδήματος.

v) Αλογονούχες ενώσεις

Οι αλογονούχες ενώσεις αποτελούν μια σημαντική κατηγορία ανθρωπογενών ρύπων στην ατμόσφαιρα, με σημαντικές επιπτώσεις στο περιβάλλον και την ανθρώπινη υγεία. Στις κυριότερες ενώσεις αυτής της κατηγορίας περιλαμβάνονται οι χλωροφθοράνθρακες (CFCs), οι υδροχλωροφθοράνθρακες (HCFCs), οι υδροφθοράνθρακες (HFCs), καθώς και ουσίες όπως το μεθυλοχλωροφόρμιο και ο τετραχλωράνθρακας (Καραθανάσης, 2006).

Οι CFCs και οι HCFCs χρησιμοποιούνταν ευρέως ως ψυκτικά, προωθητικά σε αεrolύματα και διαλύτες, αλλά η υψηλή σταθερότητά τους επιτρέπει να φτάνουν στη στρατόσφαιρα, όπου διασπώνται υπό την επίδραση της υπεριώδους ακτινοβολίας, απελευθερώνοντας χλώριο. Το χλώριο καταστρέφει το στρατοσφαιρικό όζον, το οποίο προστατεύει τη Γη από τη βλαβερή υπεριώδη ακτινοβολία (World Meteorological Organization, 2018). Αυτή η διαδικασία οδήγησε στο παρελθόν στη δημιουργία της τρύπας του όζοντος, ιδίως πάνω από την Ανταρκτική. Οι HFCs, αν και δεν επηρεάζουν το όζον, συμβάλλουν σημαντικά στην υπερθέρμανση του πλανήτη, καθώς είναι ισχυρά αέρια του θερμοκηπίου με δυναμικό θέρμανσης πολλαπλάσιο του CO₂. Η χρήση αυτών των ενώσεων αυξήθηκε ως αντικατάσταση των CFCs και HCFCs μετά το Πρωτόκολλο του Μόντρεαλ (1987), αλλά η συμβολή τους στην κλιματική αλλαγή έχει οδηγήσει στην ανάγκη περιορισμού τους (Ζερεφός, 2009).

Τα μέτρα για τη μείωση των εκπομπών αλογονούχων ενώσεων περιλαμβάνουν τη σταδιακή κατάργηση των CFCs και HCFCs, την ανάπτυξη εναλλακτικών ψυκτικών μέσων με χαμηλότερο δυναμικό θέρμανσης και την προώθηση πιο φιλικών προς το περιβάλλον τεχνολογιών. Σύμφωνα με τη Συμφωνία που υπογράφηκε στο Κιγκάλι της Ρουάντα το 2016, η σταδιακή μείωση των HFCs προβλέπεται να μειώσει σημαντικά τη συμβολή τους στην κλιματική αλλαγή.

vi) Τοξικοί και επικίνδυνοι ρύποι

Στην κατηγορία αυτή περιλαμβάνονται βαρέα μέταλλα, διοξίνες, φουράνια και πολυχλωριωμένα διφαινύλια (PCBs), καθώς και ραδιενεργά στοιχεία, που δε μπορούν να ταξινομηθούν σε άλλες κατηγορίες. Τα βαρέα μέταλλα όπως ο μόλυβδος (Pb), το κάδμιο (Cd), ο υδράργυρος (Hg) και το αρσενικό (As), εισέρχονται στην ατμόσφαιρα μέσω βιομηχανικών εκπομπών, καύσης απορριμμάτων και ορυκτών καυσίμων (Καραθανάσης, 2006). Σε μοριακό

επίπεδο ο υδράργυρος μπορεί να σχηματίζει σύμπλοκα με θειολικές ομάδες πρωτεϊνών, διαταράσσοντας σημαντικές βιολογικές λειτουργίες. Ο μόλυβδος επηρεάζει το κεντρικό νευρικό σύστημα, δεσμευόμενος με πρωτεΐνες που ρυθμίζουν τη μεταφορά ιόντων ασβεστίου, οδηγώντας σε νευρολογικές διαταραχές. Οι διοξίνες και τα φουράνια, που εκλύονται κυρίως από τη θερμική διάσπαση οργανικών υλικών παρουσία χλωρίου, αποτελούν ισχυρούς ενδοκρινικούς διαταράκτες (Lee et al., 2020). Αυτές οι ενώσεις συνδέονται με υποδοχείς αρωματικών υδρογονανθράκων (AhR) στο κυτταρόπλασμα, πυροδοτώντας την έκφραση γονιδίων που εμπλέκονται στη φλεγμονή και τον καρκινογένεση (Ζερεφός, 2009).

Τέλος οι ραδιενεργές ουσίες αποτελούν μια επικίνδυνη κατηγορία τοξικών ρύπων, οι οποίοι προέρχονται κυρίως από ατυχήματα σε πυρηνικές εγκαταστάσεις, διαχείριση πυρηνικών αποβλήτων και ατμοσφαιρική δοκιμή πυρηνικών όπλων. Στοιχεία όπως το καίσιο-137 (Cs-137), το στρόντιο-90 (Sr-90) και το ιώδιο-131 (I-131) εκλύονται στην ατμόσφαιρα και μπορούν να εισέλθουν στον ανθρώπινο οργανισμό μέσω εισπνοής ή κατανάλωσης μολυσμένων τροφών. Σε κυτταρικό επίπεδο, τα ραδιενεργά στοιχεία εκπέμπουν ιοντίζουσα ακτινοβολία που προκαλεί θραύση του DNA, οδηγώντας σε μεταλλάξεις και πιθανή καρκινογένεση. Επίσης τα ραδιενεργά ισότοπα μπορούν να μιμηθούν φυσικά στοιχεία και να ενσωματωθούν σε μεταβολικές διεργασίες όπως το στρόντιο-90, που συσσωρεύεται στα οστά αντικαθιστώντας το ασβέστιο, προκαλώντας βλάβες στον μυελό των οστών και λευχαιμία (Lee et al., 2020).

vii) Όζον και Φωτοχημικά οξειδωτικά

Το όζον είναι ένας δευτερογενής ρύπος που σχηματίζεται στην τροπόσφαιρα μέσω φωτοχημικών αντιδράσεων οξειδίων του αζώτου (NO_x) και πτητικών οργανικών ενώσεων (VOC) υπό την παρουσία ηλιακής ακτινοβολίας. Αν και το όζον στην ανώτερη ατμόσφαιρα (στρατόσφαιρα) προστατεύει τη ζωή στον πλανήτη από την υπεριώδη ακτινοβολία, στην τροπόσφαιρα αποτελεί σοβαρή απειλή για την υγεία και το περιβάλλον (Καραθανάσης, 2006).

Το όζον αντιδρά με τις λιπιδικές μεμβράνες των κυττάρων, προκαλώντας υπεροξειδωση των λιπιδίων. Αυτό οδηγεί σε διαταραχές της διαπερατότητας των μεμβρανών, οι οποίες επηρεάζουν την ομοιόσταση ιόντων και την επικοινωνία των κυττάρων. Οι φλεγμονώδεις αποκρίσεις που προκύπτουν ενισχύονται μέσω της απελευθέρωσης κυτοκινών και χημειοκινών, οι οποίες προάγουν τη διεύδυση φλεγμονωδών κυττάρων (Jenkin & Clemitshaw, 2000). Η έκθεση σε υψηλές συγκεντρώσεις όζοντος προκαλεί ερεθισμό των ματιών, του λαιμού και των

αεραγωγών, επιδεινώνοντας παθήσεις όπως το άσθμα και η χρόνια βρογχίτιδα. Οι έρευνες (Lee et al., 2020) δείχνουν ότι η παρατεταμένη έκθεση στο όζον μπορεί να οδηγήσει σε πρόωρο θάνατο, ιδιαίτερα σε άτομα με καρδιοαναπνευστικές παθήσεις. Οι VOCs εισέρχονται στον οργανισμό μέσω εισπνοής και μεταβολίζονται στο ήπαρ, όπου παράγουν τοξικούς μεταβολίτες. Αυτοί οι μεταβολίτες ενσωματώνονται στο DNA και προκαλούν μεταλλάξεις. Παράλληλα, αυξάνουν το κυτταρικό οξειδωτικό στρες, συμβάλλοντας στην έναρξη καρκινικών διεργασιών (Jenkin & Clemitshaw, 2000).

Στο περιβάλλον το όζον επηρεάζει αρνητικά τα φυτά μειώνοντας την παραγωγικότητα των καλλιεργειών και καταστρέφοντας τα δάση. Η μείωση των εκπομπών των πρόδρομων ουσιών του όζοντος, όπως τα NO_x και τα VOC, είναι απαραίτητη για τον έλεγχο της ρύπανσης από όζον. Σύμφωνα με τον WHO (2021) οι πολιτικές για την προώθηση καθαρών τεχνολογιών και την αύξηση της ενεργειακής αποδοτικότητας μπορούν να συμβάλουν σημαντικά στη μείωση της συγκέντρωσης του όζοντος στην τροπόσφαιρα (Ζερεφός, 2009).

Εκτός από το είδος των ρύπων μια άλλη μορφή ταξινόμησης είναι η προέλευση τους. Ως πρωτογενείς ρύποι χαρακτηρίζονται οι ουσίες που εκπέμπονται απευθείας στην ατμόσφαιρα ενώ ως δευτερογενείς αυτοί που δημιουργούνται έπειτα από φωτοχημικές αντιδράσεις που συμβαίνουν στην ατμόσφαιρα μεταξύ φυσικών συστατικών αλλά και πρωτογενών ρύπων (Καραθανάσης, 2006) . Επίσης κριτήριο ταξινόμησης είναι η φυσική τους κατάσταση όπου υπάρχουν δυο κατηγορίες για τους ατμοσφαιρικούς ρύπους, οι αέριοι στους οποίους περιλαμβάνονται πρωτογενείς και δευτερογενείς ρύποι σε αέρια μορφή όπως το NO₂, SO₂, όζον και πτητικοί υδρογονάνθρακες, και οι σωματιδιακοί όπου ανήκουν μικροσκοπικά σωματίδια όπως άνθρακα, μέταλλα ελαφρά ή βαρέα, σωματίδια από εκπομπές καύσης, καθώς και μεγαλύτερα πχ σκόνη, άμμο, γύρη. Στους σωματιδιακούς ρύπους ανήκουν και οι μικροοργανισμοί και τα έντομα. (Ανδρίτσος, 2008).

2.2 ΕΠΙΠΤΩΣΕΙΣ ΑΤΜΟΣΦΑΙΡΙΚΗΣ ΡΥΠΑΝΣΗΣ

Η ρύπανση της ατμόσφαιρας προκαλεί πολλά προβλήματα στην υγεία τόσο του ανθρώπου όσο και των υπόλοιπων έμβιων όντων του πλανήτη, στα οικοσυστήματα, τη βιοποικιλότητα, το φυσικό περιβάλλον καθώς και στην ανθρώπινες κατασκευές και την οικονομία. Στον άνθρωπο όταν εκτίθεται σε επιβαρυνμένο με ρύπους αέρα προκαλούνται προβλήματα υγείας κυρίως στο αναπνευστικό και καρδιαγγειακό σύστημα καθώς και ανάπτυξη νεοπλασμάτων ή νευρολογικών νόσων, που μπορεί να οδηγήσουν ακόμα και σε πρόωρο θάνατο. Βάσει εκτιμήσεων του Παγκόσμιου Οργανισμού Υγείας (WHO) οι θάνατοι που οφείλονται στην ατμοσφαιρική ρύπανση ξεπερνούν τους 500.000 ανά έτος και σε ποσοστό 2-5% αιτία είναι η εισπνοή σωματιδίων. Οι υπόλοιποι ζωντανοί οργανισμοί επίσης βρίσκονται αντιμέτωποι με τις επιπτώσεις της ατμοσφαιρικής ρύπανσης και παρατηρείται νοσηρότητα, μείωση βιοποικιλότητας, φαινόμενο βιοσυσσώρευσης μέσω της τροφικής αλυσίδας, απώλειες φυτικών ειδών, μείωση πληθυσμού εντόμων και πτηνών. Σοβαρές συνέπειες παρατηρούνται και στα οικοσυστήματα όπως διάβρωση αλλά και σε δομικά έργα.

Ένα άλλο σημαντικό πρόβλημα είναι το φαινόμενο του θερμοκηπίου που οφείλεται σε αέριους ρύπους, γνωστούς ως «αέρια του θερμοκηπίου» και συμβάλει στην αύξηση της θερμοκρασίας σε περιοχές με υψηλές τιμές ρύπανσης αλλά και γενικότερα στην παρατηρούμενη κλιματική αλλαγή που πλέον μπορεί να χαρακτηριστεί και ως κλιματική κρίση. Η αλλαγή στο κλίμα αναγνωρίστηκε από τις αρχές της δεκαετίας του 1970 όταν παρατηρήθηκε αύξηση της μέσης θερμοκρασίας του πλανήτη σε σχέση με τα δεδομένα πριν την εποχή της βιομηχανοποίησης (Καραθανάσης, 2006). Παρά τις προσπάθειες της επιστημονικής κοινότητας για περιορισμό της εξέλιξης του φαινομένου δεν υπήρξε η ανταπόκριση που θα βελτίωνε την κατάσταση παρά μόνο η μετατόπιση των βιομηχανικών κέντρων από την Ευρώπη και τις χώρες της αμερικάνικης ηπείρου στην Ασία. Η ραγδαία εκβιομηχάνιση των χωρών της νοτιοανατολικής Ασίας είχε άμεσο αντίκτυπο στην ποιότητα αέρα στις μεγάλες πόλεις και σε συνδυασμό με την υψηλή πυκνότητα πληθυσμού παρατηρήθηκαν άμεσα ζητήματα υγείας σε πλήθος πολιτών. Η Ιαπωνία ήταν από τις πρώτες ασιατικές χώρες που έλαβε μέτρα για τον περιορισμό των ρύπων (Ζερεφός, 2009).

Όπως προκύπτει από τα επίσημα στοιχεία η ρύπανση της ατμόσφαιρας επιδρά αρνητικά και στην οικονομία των πληγέντων περιοχών αλλά και όλου του πλανήτη καθώς μεγάλο μέρος

των κρατικών προϋπολογισμών δαπανάται για τη θεραπεία ασθενειών, χάνονται έσοδα από καταστροφές στην αγροτική παραγωγή καθώς και από τις επιπτώσεις ακραίων καιρικών φαινομένων (Καραθανάσης, 2006). Οι ζώνες καλλιέργειας μετατοπίζονται, χωροκατακτητικά είδη εξαπλώνονται απειλώντας τη βιοποικιλότητα, προκαλούνται προβλήματα λειψυδρίας και ερημοποίηση. Έτσι σε δεύτερο χρόνο πλήττεται ακόμα περισσότερο η υγεία του πληθυσμού, οι διασυνοριακές μετακινήσεις και η οικονομία.

2.3 ΤΡΟΠΟΙ ΜΕΤΡΗΣΗΣ ΑΤΜΟΣΦΑΙΡΙΚΗΣ ΡΥΠΑΝΣΗΣ

Η μέτρηση της ατμοσφαιρικής ρύπανσης είναι κρίσιμη για την παρακολούθηση της ποιότητας του αέρα και για την εφαρμογή στρατηγικών μείωσης των ρύπων. Οι τρόποι μέτρησης διαφέρουν ανάλογα με το είδος των ρύπων που εξετάζονται, το επίπεδο ακρίβειας που απαιτείται και το κόστος της τεχνολογίας που χρησιμοποιείται. Κάποιες από τις πιο διαδεδομένες μεθόδους περιλαμβάνουν τις χημικές αναλυτικές τεχνικές, τη δειγματοληψία αέρα, τη χρήση αισθητήρων, καθώς και τη μετεωρολογική παρακολούθηση σε συνδυασμό με μοντελοποίηση της διάδοσης των ρύπων (Ζερεφός, 2009).

2.3.1 Δειγματοληψία αέρα

Η μέθοδος αυτή περιλαμβάνει τη συλλογή αέρα από την ατμόσφαιρα, με στόχο την ανάλυση του περιεχομένου των ρύπων. Οι πιο συνηθισμένες τεχνικές δειγματοληψίας περιλαμβάνουν τη χρήση φίλτρων ή άλλων απορροφητικών μέσων για τη συλλογή των αιωρούμενων σωματιδίων και αερίων. Αυτή η μέθοδος χρησιμοποιείται κυρίως για τη μέτρηση των αιωρούμενων σωματιδίων (PM_{2.5} και PM₁₀) και άλλων αερίων, όπως το CO₂ και το CO. Οι μετρήσεις που πραγματοποιούνται μέσω δειγματοληψίας αέρα παρέχουν αξιόπιστα δεδομένα για την ποιότητα του αέρα σε συγκεκριμένες περιοχές και ώρες της ημέρας.

Η διαδικασία δειγματοληψίας μπορεί να είναι είτε συνεχής είτε περιοδική. Στην περίπτωση της συνεχούς δειγματοληψίας, οι μετρήσεις γίνονται σε πραγματικό χρόνο, επιτρέποντας την άμεση παρακολούθηση των ρύπων. Αντίθετα, η περιοδική δειγματοληψία είναι πιο οικονομική, αλλά δεν παρέχει δεδομένα σε πραγματικό χρόνο, γεγονός που ενδέχεται να περιορίσει τη δυνατότητα άμεσης παρέμβασης σε περίπτωση έκτακτης ανάγκης. Ανάλογα με τις ανάγκες της

περιοχής και το κόστος της εξοπλιστικής υποδομής, οι μέθοδοι αυτές μπορούν να εφαρμοστούν είτε σε αστικά κέντρα είτε σε περιοχές με χαμηλότερο πληθυσμό (Zerefos et al., 2002).

2.3.2 Χημικές μέθοδοι μέτρησης

Η χρήση χημικών αναλυτών είναι μια από τις πιο ακριβείς μεθόδους μέτρησης ατμοσφαιρικών ρύπων. Ειδικότερα, η φασματομετρία ατμών (Atomic Absorption Spectroscopy, AAS) και η φασματοφωτομετρία χρησιμοποιούνται για τον προσδιορισμό συγκεντρώσεων αερίων ρύπων όπως το διοξείδιο του θείου (SO_2), τα οξείδια του αζώτου (NO_x) και το όζον (O_3) (Jenkin & Clemitshaw, 2000). Η μέθοδος αυτή βασίζεται στην απορρόφηση ή εκπομπή φωτός από τα αέρια που εξετάζονται, κάτι που επιτρέπει την ακριβή μέτρηση της συγκέντρωσής τους. Τα όργανα που χρησιμοποιούνται στις χημικές αναλύσεις είναι συνήθως πολύ ακριβή και παρέχουν άμεσα αποτελέσματα, αν και το κόστος της αγοράς τους και η ανάγκη για εξειδικευμένο προσωπικό καθιστούν τη χρήση τους περιορισμένη σε ειδικές περιπτώσεις (Zerefos et al., 2002).

Η φασματομετρία ατμών χρησιμοποιείται ιδιαίτερα για τη μέτρηση αερίων που δεν έχουν μεγάλη οπτική απορρόφηση, όπως το μονοξείδιο του άνθρακα (CO), το οποίο είναι ένα κοινό ρύπος σε αστικές περιοχές λόγω των καυσαερίων. Η υψηλή ακρίβεια των μετρήσεων αυτών καθιστά αυτήν τη μέθοδο κατάλληλη για τη σύγκριση των επιπέδων ρύπανσης με τα όρια που έχουν καθοριστεί από διεθνείς οργανισμούς όπως ο Παγκόσμιος Οργανισμός Υγείας (WHO).

2.3.3 Παρακολούθηση με συστήματα αισθητήρων

Η ανάπτυξη της τεχνολογίας των αισθητήρων έχει διευκολύνει τη μέτρηση της ατμοσφαιρικής ρύπανσης σε πραγματικό χρόνο και σε μεγάλη κλίμακα. Οι αισθητήρες ατμοσφαιρικής ρύπανσης παρέχουν έναν οικονομικό και αποτελεσματικό τρόπο για τη συνεχή παρακολούθηση διαφόρων ρύπων, όπως τα αιωρούμενα σωματίδια ($\text{PM}_{2.5}$ και PM_{10}), το διοξείδιο του άνθρακα (CO_2), το μονοξείδιο του άνθρακα (CO) και τα οξείδια του αζώτου (NO_2). Οι αισθητήρες χρησιμοποιούνται ευρέως για την παρακολούθηση της ποιότητας του αέρα σε αστικές περιοχές, ειδικά σε συνθήκες χαμηλού κόστους και περιορισμένων πόρων.

Οι σύγχρονοι αισθητήρες είναι ικανοί να προσφέρουν δεδομένα σε πραγματικό χρόνο και να συνδεθούν με κεντρικά συστήματα παρακολούθησης, καθιστώντας δυνατή την παρακολούθηση της ατμοσφαιρικής ρύπανσης σε μεγάλες γεωγραφικές περιοχές. Ειδικότερα η τεχνολογία των αισθητήρων αναπτύσσεται διαρκώς επιτρέποντας μεγαλύτερη ακρίβεια και

αξιοπιστία, ενώ ταυτόχρονα μειώνει το κόστος των συστημάτων μέτρησης. Παρόλα αυτά, η ακρίβεια των μετρήσεων μπορεί να ποικίλει, και η χρήση τους απαιτεί τη συντήρηση και την τακτική βαθμονόμησή τους για την εξασφάλιση αξιόπιστων αποτελεσμάτων (Καραθανάσης, 2006).

2.3.4 Μοντελοποίηση και σύγχρονες μέθοδοι

Η ανάγκη για ακριβή και αξιόπιστη παρακολούθηση της ατμοσφαιρικής ρύπανσης έχει οδηγήσει στην ανάπτυξη νέων, πιο σύγχρονων τεχνολογιών ανάλυσης και μοντελοποίησης. Αυτές οι μέθοδοι επιτρέπουν την παρακολούθηση των ρύπων σε πραγματικό χρόνο και την πρόβλεψη της εξάπλωσής τους, γεγονός που είναι κρίσιμο για τη δημόσια υγεία και την εφαρμογή στρατηγικών μείωσης των ρύπων. Στη σύγχρονη εποχή οι μέθοδοι αυτές περιλαμβάνουν τη χρήση προηγμένων αισθητήρων, την ανάλυση δεδομένων μέσω τεχνικών μηχανικής μάθησης και τη χρήση μοντέλων διάδοσης των ρύπων (Bai et al., 2016).

2.4 Η ΑΤΜΟΣΦΑΙΡΙΚΗ ΡΥΠΑΝΣΗ ΣΤΗΝ ΕΥΡΩΠΗ & ΤΟ ΛΟΝΔΙΝΟ

2.4.1 Ευρώπη

Η ατμοσφαιρική ρύπανση στην Ευρώπη έχει μακρά ιστορία που συνδέεται με την εκβιομηχάνιση και την αστικοποίηση του 19ου αιώνα. Η χρήση άνθρακα ως βασική πηγή ενέργειας κατά τη Βιομηχανική Επανάσταση οδήγησε σε εκτεταμένες εκπομπές καπνού και διοξειδίου του θείου, προκαλώντας φαινόμενα όπως το «smog» στο Λονδίνο. Το 1952, το Μεγάλο Νέφος του Λονδίνου ήταν ένα σημείο καμπής, καθώς προκάλεσε χιλιάδες θανάτους και ανέδειξε την ανάγκη για νομοθεσία. Η πρώτη νομοθετική απάντηση ήρθε με τον Νόμο για τον Καθαρό Αέρα του 1956, ο οποίος αποτέλεσε πρότυπο για άλλες ευρωπαϊκές χώρες.

Τις τελευταίες δεκαετίες, η Ευρώπη έχει σημειώσει σημαντική πρόοδο στη μείωση της ατμοσφαιρικής ρύπανσης, κυρίως μέσω της θέσπισης αυστηρών πολιτικών από την Ευρωπαϊκή Ένωση. Η Οδηγία 2008/50/EK για την ποιότητα του αέρα και το Ευρωπαϊκό Πρόγραμμα Clean Air for Europe έχουν μειώσει τις συγκεντρώσεις ρύπων όπως το διοξείδιο του θείου και τα αιωρούμενα σωματίδια (EEA, 2021). Παρόλα αυτά, προβλήματα παραμένουν, ιδιαίτερα σε

αστικές περιοχές, όπου το όζον (O_3), τα οξείδια του αζώτου (NO_x) και τα $PM_{2.5}$ συνεχίζουν να υπερβαίνουν τα όρια ασφαλείας του Παγκόσμιου Οργανισμού Υγείας (WHO, 2021).

Οι πηγές της ρύπανσης στην Ευρώπη διαφοροποιούνται. Ο τομέας της ενέργειας, οι μεταφορές και η γεωργία είναι από τους κύριους υπαίτιους. Η γεωργία, ειδικότερα, συμβάλλει στην έκλυση αμμωνίας (NH_3), που εμπλέκεται στο σχηματισμό δευτερογενών σωματιδίων. Η βελτίωση της τεχνολογίας στις μεταφορές και η σταδιακή υιοθέτηση ανανεώσιμων πηγών ενέργειας έχουν μειώσει τις εκπομπές από οχήματα και βιομηχανίες, αλλά οι συγκεντρώσεις ρύπων παραμένουν υψηλές σε πολλές περιοχές (Kingham et al., 2000). Η έκθεση των Ευρωπαίων πολιτών σε ατμοσφαιρική ρύπανση συνδέεται με σοβαρές επιπτώσεις στην υγεία. Ετήσιες μελέτες υποδεικνύουν ότι περίπου 400.000 πρόωροι θάνατοι οφείλονται στην κακή ποιότητα του αέρα, κυρίως λόγω καρδιαγγειακών και αναπνευστικών νοσημάτων (EEA, 2021).

Κατά τη διάρκεια της πανδημίας COVID-19 η μειωμένη κυκλοφορία οχημάτων και η προσωρινή διακοπή βιομηχανικών δραστηριοτήτων οδήγησαν σε αισθητή μείωση ρύπων όπως τα οξείδια του αζώτου (NO_x) και τα αιωρούμενα σωματίδια ($PM_{2.5}$), ιδίως σε αστικές περιοχές. Ωστόσο, η άρση των περιοριστικών μέτρων και η επιστροφή στις οικονομικές δραστηριότητες επανάφεραν γρήγορα τις εκπομπές σε προ-πανδημικά επίπεδα ή και υψηλότερα. Παράλληλα, παρατηρήθηκε αυξημένη καύση βιομάζας για οικιακή θέρμανση λόγω οικονομικών δυσχερειών, γεγονός που επηρέασε την ποιότητα του αέρα σε συγκεκριμένες περιοχές (EEA, 2021).

2.4.2 Λονδίνο

Ιστορική ανασκόπηση

Από τη Βιομηχανική Επανάσταση τον 18ο αιώνα, το Λονδίνο υπήρξε ένα από τα πρώτα μεγάλα αστικά κέντρα που επηρεάστηκαν σοβαρά από τα φαινόμενα της ατμοσφαιρικής ρύπανσης. Η ταχεία ανάπτυξη της βιομηχανίας, η χρήση άνθρακα για τη θέρμανση και την παραγωγή ενέργειας και η ανάπτυξη του συστήματος μεταφορών ενίσχυσαν την εκπομπή ρύπων στην ατμόσφαιρα. Η πόλη, ήδη τότε πυκνοκατοικημένη, άρχισε να υποφέρει από εκτεταμένες καταστάσεις καπνού και καπνομίχλης, οι οποίες ήταν αρκετά συχνές (Davis, 2002).

Η μεγαλύτερη έκρηξη ατμοσφαιρικής ρύπανσης συνέβη με την αύξηση της βιομηχανοποίησης κατά τη διάρκεια του 19ου αιώνα. Οι εκπομπές διοξειδίου του θείου (SO_2), αιωρούμενων σωματιδίων (PM_{10}) και άλλων ρύπων από τα εργοστάσια και τις βιομηχανικές

εγκαταστάσεις εντάθηκαν προκαλώντας τη δημιουργία ομίχλης στην πόλη. Η ατμοσφαιρική ρύπανση εκείνη την εποχή ήταν τόσο έντονη που το Λονδίνο έγινε διάσημο για τις πυκνές αιθαλομίχλες του, που επηρέαζαν τη ζωή των κατοίκων και τη δημόσια υγεία. Το 1952 η Μεγάλη Ομίχλη του Λονδίνου (Great Smog of London) υπήρξε ένα από τα πιο δύσκολα περιστατικά, με την πόλη να καλύπτεται από μια πυκνή ομίχλη για τέσσερις ημέρες, προκαλώντας περίπου 4.000 θανάτους από αναπνευστικά προβλήματα και επιδεινώνοντας τη γενική κατάσταση της υγείας των πολιτών (Davis, 2002).

Το γεγονός αυτό αποτέλεσε καταλύτη για την υιοθέτηση περιβαλλοντικών κανονισμών. Το 1956 η Βρετανική κυβέρνηση πέρασε τον «Clean Air Act», ο οποίος προέβλεπε περιορισμούς στη χρήση άνθρακα και εισήγαγε τα πρώτα μέτρα κατά της καπνιάς και της αιθαλομίχλης. Αυτή η νομοθεσία είχε θετικά αποτελέσματα, μειώνοντας σταδιακά την ένταση των καπνών και των ομιχλών στην πόλη και βελτιώνοντας την ποιότητα του αέρα. Ωστόσο, η ατμοσφαιρική ρύπανση δεν εξαλείφθηκε πλήρως, καθώς οι εκπομπές διοξειδίου του αζώτου (NO_2) και αιωρούμενων σωματιδίων συνέχισαν να υπάρχουν, κυρίως λόγω της αυξημένης κυκλοφορίας και των αυτοκινήτων, γεγονός που είχε σοβαρές επιπτώσεις στην υγεία των κατοίκων (Davis, 2002).

Με την είσοδο του 20ού αιώνα, η ανάπτυξη της αυτοκίνησης και η χρήση των ορυκτών καυσίμων είχαν σημαντικές συνέπειες για τα επίπεδα ρύπανσης στο Λονδίνο. Παρά τη σταδιακή βελτίωση των τεχνολογιών καύσης και την αντικατάσταση των πιο ρυπογόνων υλών, τα επίπεδα ρύπανσης παρέμειναν υψηλά, και η ρύπανση από τα οχήματα έγινε ο κύριος παράγοντας αύξησης των ρύπων.

Σημερινά δεδομένα

Σήμερα το Λονδίνο εξακολουθεί να αντιμετωπίζει σημαντικά προβλήματα ατμοσφαιρικής ρύπανσης, αν και έχουν γίνει σημαντικά βήματα για την καταπολέμηση της και τη βελτίωση της ποιότητας του αέρα. Παρά τη μείωση των ρύπων από τις βιομηχανικές δραστηριότητες και την εφαρμογή αυστηρότερων κανονισμών για τη βιομηχανία και τις κατασκευές, οι εκπομπές ρύπων από τις μεταφορές παραμένουν η κύρια πηγή ρύπανσης. Οι εκπομπές διοξειδίου του αζώτου (NO_2) και αιωρούμενων σωματιδίων (PM_{10} και $\text{PM}_{2.5}$) εξακολουθούν να είναι σε επικίνδυνα επίπεδα και να προκαλούν αρνητικές συνέπειες για τη δημόσια υγεία (EEA).

Η κυριότερη πηγή των ατμοσφαιρικών ρύπων είναι η κυκλοφορία των οχημάτων, ιδιαίτερα τα οχήματα με κινητήρες εσωτερικής καύσης. Η κίνηση των αυτοκινήτων και των φορτηγών, σε συνδυασμό με την πυκνότητα της αστικής ανάπτυξης, οδηγούν σε υψηλά επίπεδα ρύπανσης, ιδιαίτερα στις κεντρικές περιοχές της πόλης. Το 2016, ο Παγκόσμιος Οργανισμός Υγείας (ΠΟΥ) ανέφερε ότι το Λονδίνο υπερέβαινε τα όρια για τα επίπεδα διοξειδίου του αζώτου (NO₂), ενώ οι ρύποι αυτοί συνδέονται με την εμφάνιση σοβαρών αναπνευστικών και καρδιαγγειακών ασθενειών (WHO, 2021). Εκτιμάται ότι η ατμοσφαιρική ρύπανση προκαλεί 9.000 πρόωρους θανάτους κάθε χρόνο στην πόλη, κυρίως λόγω των παθήσεων του αναπνευστικού και της καρδιαγγειακής συστήματος. Τα παιδιά, οι ηλικιωμένοι και τα άτομα με αναπνευστικά προβλήματα αποτελούν τις πιο ευάλωτες ομάδες.

Αντίστοιχα, η χρήση καυσίμων από τα οχήματα συνεχίζει να είναι πρόβλημα, καθώς τα diesel οχήματα και οι βιομηχανικές διαδικασίες εξακολουθούν να εκλύουν μεγάλες ποσότητες ρύπων στην ατμόσφαιρα. Από το 2014, η πόλη του Λονδίνου έχει καταβάλει προσπάθειες να μειώσει τις εκπομπές με την εισαγωγή του Ultra Low Emission Zone (ULEZ) το 2019, που περιορίζει την κυκλοφορία των πιο ρυπογόνων οχημάτων στο κέντρο της πόλης. Παρά τα αυξημένα μέτρα, τα επίπεδα των ατμοσφαιρικών ρύπων εξακολουθούν να παραμένουν πάνω από τα συνιστάμενα όρια, ιδίως τις ημέρες αυξημένης κίνησης μεταφορικών μέσων ή όταν σημειώνονται υψηλές θερμοκρασίες.

2.5 ΣΧΕΣΗ ΕΠΙΠΕΔΩΝ ΡΥΠΑΝΣΗΣ ΜΕ ΚΛΙΜΑΤΙΚΟΥΣ ΚΑΙ ΚΑΙΡΙΚΟΥΣ ΠΑΡΑΓΟΝΤΕΣ

2.5.1 Μετεωρολογικές παράμετροι

Θερμοκρασία αέρα

Με τον όρο αυτό αναφέρεται συνήθως η θερμοκρασία (T) σε σημείο ύψους περίπου 2 μέτρων από την επιφάνεια του εδάφους και η μέτρηση γίνεται με όργανα που βρίσκονται υπό σκιά, εντός μετεωρολογικού κλωβού. Επίσης μετρήσεις μπορούν να πραγματοποιούνται και σε άλλα σημεία πιο κοντά στο έδαφος ή πιο ψηλά στην ατμόσφαιρα. Στη διάρκεια του 24ωρου αποδίδονται η απολύτως μέγιστη και η απολύτως ελάχιστη θερμοκρασία ενός γεωγραφικού

σημείου ως T_{max} και T_{min} αντίστοιχα. Η θερμοκρασία μετράται συνήθως σε βαθμούς Κελσίου ή Κέλβιν (Ζερεφός, 2009).

Ατμοσφαιρική πίεση

Η πίεση της ατμόσφαιρα ή βαρομετρική πίεση είναι η επίδραση που έχει σε ένα σημείο της επιφάνειας της γης η υπερκείμενη αέρια στήλη και μετράται σε hPa, όπου $1 \text{ hPa} = 102 \text{ N/m}^2$. Μέση τιμή που επικρατεί στην επιφάνεια της γης είναι τα 1013 hPa και οι μεταβολές που συμβαίνουν εξηγούν τα καιρικά φαινόμενα και τις ατμοσφαιρικές διαταραχές (Ζερεφός, 2009).

Άνεμος

Οι οριζόντιες μετακινήσεις του αέρα στην επιφάνεια της γης ορίζονται ως άνεμος. Συνήθως η κίνηση του ανέμου πραγματοποιείται σε ρεύματα αέρα που μετακινούνται από ένα σημείο σε ένα άλλο με ορισμένη ταχύτητα και διεύθυνση. Οι άνεμοι προσδίδουν χαρακτηριστικά στο κλίμα και τις καιρικές συνθήκες ενός τόπου και κατατάσσονται σε κλίμακα ανάλογα με την ταχύτητα ή ένταση ενώ επίσης καταγράφεται και η φορά τους. Αιτίες δημιουργίας ανέμων είναι η διαφορά ατμοσφαιρικής πίεσης και θερμοκρασίας, η περιστροφή της γης, η προσλαμβάνουσα ενέργεια από τον ήλιο καθώς και τα γεωμορφολογικά χαρακτηριστικά (Σαχσαμάνογλου & Μακρογιάννης, 1998, Ζερεφός, 2009).

Υετός

Μία ακόμα παράμετρος που χρησιμοποιείται στη μετεωρολογία είναι ο υετός που δείχνει την ποσότητα νερού που πέφτει στη γη από την ατμόσφαιρα σε υγρή μορφή ή σε μορφή πάγου όπως βροχή, χιόνι, χαλάζι κλπ. Αποτελεί βασικό στοιχείο για τον υδρολογικό κύκλο καθώς μέσω του υετού επιστρέφει στην επιφάνεια της γης το νερό που εξατμίζεται ενώ επηρεάζει άμεσα τις καιρικές συνθήκες και το κλίμα μιας περιοχής. Συνήθης μονάδα μέτρησης είναι τα mm (Ζερεφός, 2009).

Ατμοσφαιρική υγρασία

Πρόκειται για την ποσότητα υδρατμών που βρίσκονται στην ατμόσφαιρα και έχει άμεση επιρροή στο κλίμα, επηρεάζει τη θερμοκρασία και τον υετό. Μετράται ως ποσοστό σχετικής υγρασίας, δηλαδή το ποσοστό της μέγιστης δυνατής υγρασίας που μπορεί να υπάρξει σε δεδομένη θερμοκρασία (Ζερεφός, 2009).

Οι παραπάνω παράγοντες συντελούν στη διαμόρφωση του κλίματος μιας περιοχής και καθορίζουν τις καιρικές συνθήκες που επικρατούν. Η ατμοσφαιρική πίεση, η θερμοκρασία και ο άνεμος έχουν τη μεγαλύτερη επίδραση στα καιρικά συστήματα. Τα προγνωστικά μοντέλα καιρού λαμβάνουν υπόψιν τις παραπάνω παραμέτρους και εξετάζουν στοιχεία από μετεωρολογικούς σταθμούς που βρίσκονται σε όλο τον κόσμο για να πραγματοποιούν ακριβείς προβλέψεις (Σαχσαμάνογλου & Μακρογιάννης, 1998) .

2.5.2 Καιρικές συνθήκες και ρύπανση

Η συγκέντρωση ρύπων στην ατμόσφαιρα οφείλεται σε πολλές πηγές, φυσικές ή ανθρωπογενείς, και σε ορισμένες ενώσεις παρατηρείται άμεση ή έμμεση συσχέτιση με τις καιρικές συνθήκες. Ενώ είναι γνωστές εδώ και δεκαετίες οι επιπτώσεις των ρύπων στο κλίμα υπάρχει και η αντίστροφη κατάσταση με τις καιρικές μεταβολές να επιδρούν στα επίπεδα των ατμοσφαιρικών ρύπων (Zerefos et al., 2002). Στη σημερινή εποχή η αλλαγή του κλίματος εκτός από τα έντονα και πολλές φορές ακραία καιρικά φαινόμενα, την ερημοποίηση και άλλες επιπτώσεις συσχετίζεται και με τις μεταβολές των συγκεντρώσεων ρύπων στην ατμόσφαιρα. Ενώ λαμβάνονται αυστηρά μέτρα για τον περιορισμό των εκπομπών ρύπων, στα αστικά περιβάλλοντα η ρύπανση υφίσταται και συχνά προκαλεί προβλήματα στην υγεία των κατοίκων (Καραθανάσης, 2006).

Μια ενδιαφέρουσα περίπτωση είναι το όζον το οποίο όπως συμβαίνει και με άλλους φωτοχημικούς ρύπους μπορεί να σχηματιστεί ως αποτέλεσμα της ηλιακής ακτινοβολίας μικρού μήκους σε αέριους ρύπους και συστατικά της ατμόσφαιρας. Συγκεκριμένα η συγκέντρωση του όζοντος φαίνεται αυξημένη σε καλές καιρικές συνθήκες με ήπιες νεφώσεις και μειώνεται όταν υπάρχει έντονη νέφωση και βροχοπτώσεις (Jenkin & Clemitshaw, 2000).

Επίσης η συγκέντρωση αιωρούμενων σωματιδίων δείχνει να εξαρτάται σε σημαντικό βαθμό από τα καιρικά φαινόμενα που επικρατούν κι αυτό συμβαίνει καθώς οι καιρικές συνθήκες επιδρούν ποικιλοτρόπως στα μικροσωματίδια. Η θερμοκρασία επιδρά στα σωματίδια ανάλογα με τα συστατικά που αυτά περιέχουν. Η άνοδος της θερμοκρασίας προκαλεί μείωση των σωματιδίων που περιέχουν νιτρικά ή οργανικά ημιπτητικά συστατικά καθώς αυτά μετατρέπονται σε αέρια, αντιθέτως αυξάνονται τα θειικά άλατα επειδή οξειδώνεται πιο γρήγορα το διοξείδιο του θείου. Ο άνεμος έχει επίσης άμεση επίδραση είτε μεταφέροντας σωματίδια, κυρίως φυσικής προέλευσης, όπως γύρη και σκόνη (χαρακτηριστική περίπτωση η αφρικάνικη σκόνη που κατακλύζει την Ευρώπη και τις υπόλοιπες περιοχές της Μεσογείου), είτε απομακρύνοντας σωματίδια που ήδη

βρίσκονται σε μια περιοχή. Η αυξημένη σχετική υγρασία επιτρέπει στα σωματίδια να απορροφούν νερό συνήθως μαζί με ημιπτητικά συστατικά (Jenkin & Clemitshaw, 2000).

Τέλος η αύξηση της θερμοκρασία τόσο ως καιρική μεταβολή όσο και γενικότερα ως κλιματικό στοιχείο οδηγεί σε αύξηση συγκεντρώσεων αερίων του θερμοκηπίου. Οι περιοχές που επηρεάζονται περισσότερο από την κλιματική αλλαγή και παρουσιάζουν αύξηση της μέσης θερμοκρασίας ενδέχεται να παρουσιάσουν και αυξημένα επίπεδα ατμοσφαιρικών ρύπων, καθώς προβλέπεται ότι στις επόμενες δεκαετίες η ετήσια θερμοκρασία της γης θα παρουσιάσει αύξηση πάνω από 2 °C σε σχέση με τα σημερινά επίπεδα (Zerefos et al., 2002). Ήδη παρατηρείται τήξη πάγων στους πόλους και απελευθέρωση παγιδευμένων αερίων, όπως μεθάνιο.

Παράγοντας	Ατμοσφαιρικός ρύπος	Μεταβολή ρύπων	Επίδραση
Αύξηση θερμοκρασίας	Όζον	Αύξηση	Επιτάχυνση φωτοχημικών αντιδράσεων
	PM θειικά	Αύξηση	Ταχύτερη οξείδωση θείου
	PM νιτρικά	Μείωση	Μετάβαση σε αέρια φάση
	Φωτοχημικά οξειδωτικά	Αύξηση	Αύξηση πτητικών οργανικών ενώσεων
	Αλλεργιογόνα	Αύξηση	Μεταβολές βιοποικιλότητας φυτών
Μείωση βροχοπτώσεων	Αιωρούμενα σωματίδια, σκόνη	Μεταφορά	Συσσώρευση σωματιδίων
Άπνοια	Όζον	Αύξηση	Εξασθένιση κυκλοφορίας αέρα, μετατόπιση κυκλώνων
	Αιωρούμενα σωματίδια	Αύξηση	
Τήξη πάγων	Όζον	Αύξηση	Απελευθέρωση στην ατμόσφαιρα
	Μεθάνιο	Αύξηση	
	Διοξείδιο άνθρακα	Αύξηση	
	Οργανικοί ρύποι (PCBs, DDT)	Αύξηση	

Πίνακας 2. Επίδραση καιρικών μεταβολών στους ρύπους της ατμόσφαιρας

Συνοψίζοντας λοιπόν φαίνεται ότι υπάρχει μια συνεχής αλληλεπίδραση μεταξύ καιρού και ρύπανσης. Οι ρύποι προκαλούν μεταβολές στο κλίμα και με τη σειρά τους οι καιρικές συνθήκες μπορούν να επηρεάζουν τα επίπεδα ρύπανσης. Έτσι δημιουργείται ένας κύκλος που κλιμακώνει την εξέλιξη του φαινομένου της κλιματικής κρίσης με σοβαρές άμεσες και έμμεσες επιπτώσεις στο περιβάλλον και τον άνθρωπο. Για το λόγο αυτό είναι σημαντική η ανάπτυξη συνδυαστικών μοντέλων πρόβλεψης των επιπέδων των ρύπων και των κλιματικών μεταβολών.

ΚΕΦΑΛΑΙΟ 3. ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Η ραγδαία ανάπτυξη των υπολογιστικών συστημάτων έδωσε στον άνθρωπο τη δυνατότητα να εξερευνήσει την Τεχνητή Νοημοσύνη σε πραγματικές συνθήκες. Δεκαετίες πριν, όταν ξεκίνησε η αλματώδης πορεία της επιστήμης της πληροφορικής, ξεκίνησε και ο σχεδιασμός αλγορίθμων που μπορούσαν να δώσουν ένα είδος νοημοσύνης στα συστήματα. Αυτό αποτέλεσε έναυσμα για περεταίρω έρευνα στον τομέα όμως παράλληλα προκάλεσε και ανησυχία για τα όρια στα οποία μπορεί να φτάσει και κατά πόσο μπορεί να γίνει επικίνδυνη για την ανθρωπότητα.

3.1 ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ

Στην πραγματικότητα η Τεχνητή Νοημοσύνη δεν είναι ανεξάρτητη νοημοσύνη συστημάτων και δεν μπορεί προς το παρόν να αντικαταστήσει την ανθρώπινη. Ο όρος Μηχανική Μάθηση περιγράφει καλύτερα την συγκεκριμένη τεχνολογία και μπορεί να αποδοθεί ως η τεχνολογία με την οποία *«ένα υπολογιστικό σύστημα δημιουργεί μοντέλα ή πρότυπα βασιζόμενο σε ένα σύνολο δεδομένων»* (Βλαχάβας κ.α., 2020). Τα συστήματα αυτά έχουν σκοπό να βελτιστοποιούν τα παραγόμενα αποτελέσματα ως προς την ταχύτητα και την ακρίβεια.

Η μηχανική μάθηση (Machine Learning, ML) είναι ένας τομέας της τεχνητής νοημοσύνης (Artificial Intelligence, AI), ο οποίος αφορά την ανάπτυξη αλγορίθμων και στατιστικών μοντέλων που επιτρέπουν στα υπολογιστικά συστήματα να μαθαίνουν και να βελτιώνονται μέσω της εμπειρίας και των δεδομένων, χωρίς την ανάγκη για ρητή προγραμματιστική παρέμβαση. Η ερευνητική πρόοδος στον τομέα της μηχανικής μάθησης έχει παρουσιάσει ραγδαία ανάπτυξη τις τελευταίες δεκαετίες, επιτυγχάνοντας σημαντικά ορόσημα τόσο στη θεωρητική βάση όσο και στις εφαρμογές του πεδίου (Ertel).

Η θεμελίωση του πεδίου μπορεί να αναχθεί στις δεκαετίες του 1950 και του 1960, με την αρχική ιδέα ότι οι υπολογιστές μπορούσαν να προγραμματιστούν για να λύσουν προβλήματα, χωρίς να απαιτείται η ρητή καθοδήγηση του ανθρώπου. Ο Arthur Samuel είναι συνήθως

αναγνωρισμένος ως ένας από τους πρώτους που ανέπτυξε την έννοια της «μάθησης από την εμπειρία» το 1959, μέσω του διάσημου αλγορίθμου του για το παιχνίδι του πιγκ-πογκ. Στόχος του ήταν να κατασκευάσει αλγορίθμους που επιτρέπουν στον υπολογιστή να βελτιώνει τη συμπεριφορά του με την πάροδο του χρόνου μέσω της αλληλεπίδρασης με το περιβάλλον και της ανατροφοδότησης (Ertel). Τη δεκαετία του 1960 και του 1970, οι πρώτες έννοιες της μηχανικής μάθησης και της αναγνώρισης προτύπων ανέπτυξαν μεθόδους για την ανάλυση δεδομένων μέσω τεχνικών όπως η γραμμική παλινδρόμηση και τα στατιστικά μοντέλα. Ωστόσο, παρά τις θεωρητικές εξελίξεις, η πρακτική εφαρμογή της μηχανικής μάθησης ήταν περιορισμένη, κυρίως λόγω των τεχνολογικών περιορισμών των υπολογιστικών συστημάτων της εποχής (Βλαχάβας κ.α., 2020).

Η δεκαετία του 1980 υπήρξε καθοριστική για την εξέλιξη της μηχανικής μάθησης. Η βελτίωση των υπολογιστικών δυνατοτήτων και η ανακάλυψη πιο αποτελεσματικών μεθόδων εκπαίδευσης για αλγορίθμους, όπως τα νευρωνικά δίκτυα και οι αλγόριθμοι ενισχυμένης μάθησης, επέτρεψαν την εφαρμογή αυτών των τεχνικών σε μεγαλύτερης κλίμακας προβλήματα. Ο Geoffrey Hinton και οι συνεργάτες του στην ανάπτυξη των backpropagation algorithms επέτρεψαν την εκπαίδευση νευρωνικών δικτύων με περισσότερα από ένα επίπεδα, ανοίγοντας το δρόμο για πιο πολύπλοκα μοντέλα μάθησης. Η νέα αυτή προσέγγιση συνέβαλε σημαντικά στην αναγνώριση και ταξινόμηση προτύπων, όπως η αναγνώριση γραφής και ομιλίας (Βλαχάβας κ.α., 2020).

Από τη δεκαετία του 1990 και μετά, η μηχανική μάθηση άρχισε να αποκτά ευρύτερη εφαρμογή, με την ανάπτυξη αλγορίθμων όπως οι Support Vector Machines (SVMs), οι οποίοι προσέφεραν ισχυρά εργαλεία για την ταξινόμηση δεδομένων και την ανάλυση υψηλής διάστασης. Κατά τη διάρκεια αυτής της περιόδου, η ανάπτυξη των τεχνικών κλασμάτων αναγνώρισης δεδομένων και της μαθηματικής βελτιστοποίησης συνέβαλε στην αποτελεσματικότερη και ταχύτερη ανάλυση και επεξεργασία δεδομένων. Επίσης, η εστίαση άρχισε να επεκτείνεται στην εξόρυξη δεδομένων και τη στατιστική μάθηση, με στόχο την ανακάλυψη κρυφών μοτίβων και σχέσεων σε μεγάλα σύνολα δεδομένων (Βλαχάβας κ.α., 2020).

Η πιο πρόσφατη σημαντική εξέλιξη στην περιοχή της μηχανικής μάθησης σημειώθηκε τη δεκαετία του 2010 με την εξάπλωση των βαθιών νευρωνικών δικτύων (Deep Learning), τα οποία έφεραν επανάσταση στον τρόπο ανάλυσης δεδομένων, επιτυγχάνοντας σημαντικά καλύτερα αποτελέσματα σε εφαρμογές όπως η υπολογιστική όραση, η αναγνώριση φωνής και η μετάφραση

γλωσσών. Η ραγδαία πρόοδος των GPU (Graphics Processing Units) και των τεχνικών παράλληλης επεξεργασίας υπήρξε καθοριστική για την εκπαίδευση και τη λειτουργία αυτών των υπολογιστικά εντατικών μοντέλων. Οι μέθοδοι αυτές αναγνώρισαν τα μεγάλα δεδομένα ως έναν από τους κύριους καταλύτες για την πρόοδο της μηχανικής μάθησης, καθώς η εκπαίδευση των μοντέλων απαιτούσε τεράστιες ποσότητες δεδομένων για να διακρίνουν και να κατανοήσουν πρότυπα και μοτίβα με υψηλή ακρίβεια (Ertel, Βλαχάβας κ.α., 2020).

Η πρόοδος της μηχανικής μάθησης έχει συντελέσει στην ανάπτυξη ευρέως εφαρμοζόμενων αλγορίθμων και τεχνικών που έχουν επηρεάσει σημαντικά πεδία όπως η ιατρική, οι χρηματοοικονομικές αγορές, η ρομποτική και η ανάλυση κοινωνικών δικτύων. Οι εξελίξεις αυτές, ωστόσο, συνεχίζουν να προκαλούν προκλήσεις, ιδιαίτερα στην κατανόηση της ερμηνευσιμότητας των μοντέλων και την αντιμετώπιση θεμάτων που σχετίζονται με την ηθική και τις αντιφάσεις δεδομένων (Bishop, 2006).

3.2 ΚΑΤΗΓΟΡΙΕΣ ΜΟΝΤΕΛΩΝ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

3.2.1. Μάθηση υπό επίβλεψη (Supervised Learning)

Η μάθηση με επίβλεψη είναι μία από τις πιο διαδεδομένες και βασικές τεχνικές στη μηχανική μάθηση. Σε αυτήν την προσέγγιση, το μοντέλο εκπαιδεύεται με τη βοήθεια ενός συνόλου δεδομένων που περιλαμβάνει τόσο τις εισόδους όσο και τις εξόδους (ετικέτες). Στόχος της μεθόδου είναι η εκμάθηση μιας συνάρτησης που να συνδέει τις εισόδους με τις εξόδους, επιτρέποντας στο μοντέλο να κάνει σωστές προβλέψεις σε νέα, αόριστα δεδομένα. Τα δεδομένα εκπαίδευσης περιλαμβάνουν ζεύγη (εικόνα, ετικέτα), (κείμενο, κατηγορία), κλπ., και το μοντέλο προσπαθεί να μάθει να προβλέπει την ετικέτα για κάθε δεδομένο εισόδου. Στη συνέχεια, το μοντέλο μπορεί να αξιολογηθεί χρησιμοποιώντας ένα σύνολο δεδομένων ελέγχου, το οποίο περιλαμβάνει δεδομένα για τα οποία γνωρίζουμε την εξωτερική έξοδο (Ertel, Βλαχάβας κ.α., 2020).

Η γραμμική παλινδρόμηση και οι αλγόριθμοι δέντρων απόφασης είναι δύο κλασικά παραδείγματα αλγορίθμων που χρησιμοποιούνται στη μάθηση με εποπτεία. Παρά το γεγονός ότι

η μάθηση με εποπτεία μπορεί να έχει πολύ υψηλή ακρίβεια, απαιτεί μεγάλο όγκο δεδομένων με γνωστά αποτελέσματα για να εκπαιδευτεί το μοντέλο αποτελεσματικά (Bishop, 2006).

3.2.2. Μάθηση χωρίς επίβλεψη (Unsupervised Learning)

Η μάθηση χωρίς επίβλεψη ή μη εποπτευόμενη μάθηση αναφέρεται σε αλγόριθμους που μαθαίνουν από δεδομένα χωρίς να διαθέτουν ετικέτες εξόδου. Το κύριο χαρακτηριστικό αυτής της μεθόδου είναι ότι το μοντέλο δεν έχει συγκεκριμένες κατευθύνσεις για το τι να μάθει από τα δεδομένα. Αντίθετα, αναζητά δομές και μοτίβα μέσα στα δεδομένα, προσπαθώντας να ανιχνεύσει κρυφές σχέσεις ή ομάδες (clusters) που ενδέχεται να υπάρχουν στα δεδομένα (Bishop, 2006).

Οι πιο κοινές μέθοδοι της μάθησης χωρίς εποπτεία είναι οι αλγόριθμοι ομαδοποίησης, όπως το K-means, και οι αλγόριθμοι ανάλυσης κύριων συνιστωσών (PCA). Αυτοί οι αλγόριθμοι χρησιμοποιούνται σε περιπτώσεις όπου τα δεδομένα είναι αδόμητα ή η εξωτερική έξοδος είναι άγνωστη, όπως στην ανάλυση δεδομένων πελατών για την αναγνώριση προτύπων συμπεριφοράς. Η μάθηση χωρίς εποπτεία είναι χρήσιμη για την εξόρυξη κρυφών μοτίβων, την κατηγοριοποίηση δεδομένων σε ομάδες και τη μείωση διάστασης των δεδομένων (Aurelien, 2019).

3.2.3. Ενισχυμένη μάθηση (Reinforcement Learning)

Η μάθηση με ενίσχυση αναφέρεται σε μια μορφή μάθησης όπου το μοντέλο, ή πράκτορας (agent), αλληλεπιδρά με το περιβάλλον του και αναλαμβάνει δράσεις με στόχο την επίτευξη ενός συγκεκριμένου σκοπού. Ο πράκτορας μαθαίνει μέσω της διαδικασίας της ανατροφοδότησης, στην οποία λαμβάνει ανταμοιβές (rewards) ή ποινές (penalties) ανάλογα με το αν οι ενέργειές του είναι θετικές ή αρνητικές για την επίτευξη του στόχου του (Bishop, 2006). Χρησιμοποιείται σε σενάρια όπου η απόφαση πρέπει να γίνει σε διαδοχικά βήματα, όπως στην ρομποτική ή στον προγραμματισμό παιχνιδιών. Το μοντέλο αξιολογεί την απόδοσή του με τη βοήθεια ενός συστήματος ανταμοιβής και χρησιμοποιεί τις εμπειρίες του για να βελτιώσει τις μελλοντικές του ενέργειες μέσω της διαδικασίας της βελτιστοποίησης. Ο αλγόριθμος Q-learning και το deep Q-network (DQN) είναι χαρακτηριστικά παραδείγματα της μάθησης με ενίσχυση, οι οποίοι έχουν επιτυχώς εφαρμοστεί σε τομείς όπως τα παιχνίδια και η ρομποτική αυτονομία (Aurelien, 2019).

3.2.4. Βαθιά μάθηση (Deep Learning)

Αποτελεί έναν υποκλάδο της μηχανικής μάθησης που επικεντρώνεται στην εκπαίδευση πολυεπίπεδων νευρωνικών δικτύων με σκοπό την ανάλυση πολύπλοκων, αδόμητων δεδομένων.

Οι μέθοδοι βαθιάς μάθησης χρησιμοποιούν βαθιά νευρωνικά δίκτυα που περιλαμβάνουν πολλαπλά επίπεδα (layers) με συνδέσεις μεταξύ τους, επιτρέποντας στο μοντέλο να εξάγει χαρακτηριστικά σε διαφορετικά επίπεδα αφαιρετικότητας. Αυτή η προσέγγιση έχει καταφέρει να επιλύσει προβλήματα που ήταν ανέφικτα για παραδοσιακές τεχνικές μηχανικής μάθησης (Bai et al., 2016).

Η εφαρμογή της βαθιάς μάθησης έχει επιφέρει σημαντικά αποτελέσματα σε τομείς όπως η υπολογιστική όραση, η αναγνώριση φωνής, και η μετάφραση γλωσσών. Τα Convolutional Neural Networks (CNNs) και τα Recurrent Neural Networks (RNNs) είναι δύο από τα πιο δημοφιλή μοντέλα βαθιάς μάθησης, τα οποία χρησιμοποιούνται εκτενώς για την αναγνώριση εικόνας και την επεξεργασία φυσικής γλώσσας. Η βαθιά μάθηση έχει αναδείξει την αξία της εκπαίδευσης σε μεγάλα δεδομένα και της χρήσης εξαιρετικά ισχυρών υπολογιστικών πόρων, όπως οι GPU, για την εκπαίδευση αυτών των μοντέλων σε μεγάλες κλίμακες (Aurelien, 2019).

3.3 ΧΡΗΣΗ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ ΣΤΗΝ ΑΝΑΛΥΣΗ ΚΑΙ ΠΡΟΒΛΕΨΗ ΡΥΠΑΝΣΗΣ

Από τις πιο κοινές τεχνικές που χρησιμοποιούνται είναι η μάθηση με εποπτεία, η οποία αναλύει δεδομένα από αισθητήρες ρύπανσης, μετεωρολογικούς σταθμούς και άλλες παραμέτρους, όπως η θερμοκρασία και η υγρασία, για την εκπαίδευση μοντέλων που προβλέπουν την ποιότητα του αέρα σε διαφορετικές χρονικές στιγμές και περιοχές. Αλγόριθμοι όπως η γραμμική παλινδρόμηση, τα νευρωνικά δίκτυα και οι Support Vector Machines (SVMs) εφαρμόζονται για να ταξινομήσουν περιοχές με υψηλά επίπεδα ρύπανσης και να εντοπίσουν τις παραμέτρους που επηρεάζουν την ατμοσφαιρική ποιότητα (Gururaj et al., 2024). Επιπλέον, η μάθηση με εποπτεία χρησιμοποιείται για να εκτιμηθεί η επίδραση των καιρικών συνθηκών στην εξάπλωση των ρύπων, επιτρέποντας την ανάπτυξη πιο ακριβών μοντέλων πρόβλεψης (Masih, 2019).

Η μάθηση χωρίς εποπτεία εφαρμόζεται για την εξόρυξη κρυφών μοτίβων από αδόμητα δεδομένα. Αυτή η τεχνική χρησιμοποιείται για την ομαδοποίηση περιοχών με παρόμοιες συνθήκες ρύπανσης, την ανίχνευση περιοχών υψηλού κινδύνου και την ανάλυση συσχετίσεων μεταξύ διαφορετικών παραμέτρων. Μέθοδοι όπως το K-means και η ανάλυση κύριων συνιστωσών (PCA) βοηθούν στην κατηγοριοποίηση περιοχών με υψηλά επίπεδα ρύπανσης, συμβάλλοντας στη

στρατηγική διαχείρισης της ποιότητας του αέρα και στην εκτίμηση του αντίκτυπου της ρύπανσης στην υγεία. Ειδικότερα, η μείωση της διάστασης των δεδομένων με την PCA επιτρέπει την εξαγωγή σημαντικών χαρακτηριστικών από μεγάλες βάσεις δεδομένων, κάνοντάς τις πιο εύκολες στην ανάλυση και στην αξιοποίηση για περαιτέρω επεξεργασία (Moustris et al., 2010).

Η μάθηση με ενίσχυση, μια τεχνική που επιτρέπει στους αλγόριθμους να μάθουν μέσω αλληλεπίδρασης με το περιβάλλον, χρησιμοποιείται για την ανάπτυξη στρατηγικών που μειώνουν την ρύπανση σε πραγματικό χρόνο. Μοντέλα όπως το Q-learning και τα deep Q-networks (DQN) μπορούν να εφαρμόζονται για τη βελτιστοποίηση των διαδικασιών βιομηχανικής παραγωγής ή της κυκλοφορίας σε αστικές περιοχές, με στόχο τη μείωση των εκπομπών ρύπων. Για παράδειγμα, χρησιμοποιώντας μάθηση με ενίσχυση, τα μοντέλα μπορούν να προσαρμοστούν και να βρουν τις καλύτερες στρατηγικές για τον περιορισμό της ρύπανσης σε περιοχές με έντονη βιομηχανική δραστηριότητα ή κυκλοφοριακή συμφόρηση. Η εφαρμογή αυτής της μεθόδου στην πραγματική ζωή μπορεί να προσφέρει λύσεις για την πρόληψη και τη διαχείριση της ατμοσφαιρικής ρύπανσης (Akanksha et al., 2023).

Η βαθιά μάθηση, η οποία αποτελεί εξέλιξη της παραδοσιακής μηχανικής μάθησης, έχει βρει επίσης εφαρμογές στην ανάλυση της ατμοσφαιρικής ρύπανσης. Τα Convolutional Neural Networks (CNNs) χρησιμοποιούνται για την ανάλυση εικόνας από δορυφόρους και αισθητήρες, ενώ τα Recurrent Neural Networks (RNNs) είναι ιδανικά για την ανάλυση χρονοσειρών, επιτρέποντας την πρόβλεψη της ποιότητας του αέρα και την ανίχνευση μοτίβων ρύπανσης. Η βαθιά μάθηση επιτρέπει τη συνδυαστική ανάλυση διάφορων τύπων δεδομένων, όπως εικόνες και χρονοσειρές, επιτυγχάνοντας την ακριβέστερη πρόβλεψη της ποιότητας του αέρα και την καλύτερη κατανόηση της αλληλεπίδρασης διαφορετικών παραμέτρων (Samad et al., 2023).

Συνολικά, η μηχανική μάθηση επιτρέπει την ανάλυση και πρόβλεψη των ατμοσφαιρικών συνθηκών και ρύπανσης με μεγαλύτερη ακρίβεια και αποδοτικότητα. Μέσω των τεχνικών αυτών, είναι εφικτή η ανάπτυξη και εφαρμογή πιο αποδοτικών στρατηγικών για την παρακολούθηση και τη μείωση της ατμοσφαιρικής ρύπανσης, συμβάλλοντας έτσι στη βελτίωση της δημόσιας υγείας και της ποιότητας του αέρα στις πόλεις (Rahman et al., 2024).

3.4 ΑΛΓΟΡΙΘΜΟΙ ΠΑΛΙΝΔΡΟΜΗΣΗΣ ΣΤΗΝ ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ

Η παλινδρόμηση αποτελεί θεμελιώδη τεχνική στατιστικής και μηχανικής μάθησης, χρησιμοποιούμενη για την ανάλυση δεδομένων και την πρόβλεψη. Η κύρια ιδέα πίσω από τους αλγόριθμους παλινδρόμησης είναι η κατανόηση και μοντελοποίηση των σχέσεων μεταξύ μιας εξαρτημένης μεταβλητής και μίας ή περισσότερων ανεξάρτητων μεταβλητών, με σκοπό τη δημιουργία μοντέλων που μπορούν να κάνουν ακριβείς προβλέψεις για νέα δεδομένα. Οι αλγόριθμοι παλινδρόμησης παρέχουν ένα ισχυρό εργαλείο για την ανάλυση δεδομένων και την πρόβλεψη. Κάθε τύπος παλινδρόμησης έχει τα δικά του πλεονεκτήματα και εφαρμογές, και η επιλογή του κατάλληλου αλγορίθμου εξαρτάται από τη φύση των δεδομένων και την απαιτούμενη ανάλυση (Bishop, 2006).

Γραμμική Παλινδρόμηση: Είναι μία από τις πιο θεμελιώδεις και απλές μεθόδους για την ανάλυση δεδομένων. Αυτή η μέθοδος χρησιμοποιείται για να μοντελοποιήσει την ευθεία γραμμική σχέση μεταξύ μιας εξαρτημένης μεταβλητής και μίας ή περισσότερων ανεξάρτητων μεταβλητών. Ο στόχος της είναι να εκτιμήσει τις παραμέτρους (ή συντελεστές) της εξίσωσης που προσδιορίζει τη γραμμική σχέση.

Η μαθηματική της μορφή μπορεί να εκφραστεί ως εξής:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n + \epsilon$$

όπου Y είναι η εξαρτημένη μεταβλητή, X_i οι ανεξάρτητες μεταβλητές και β_i οι συντελεστές που πρέπει να εκτιμηθούν. Η γραμμική παλινδρόμηση χρησιμοποιείται ευρέως για προβλέψεις σε οικονομία, κοινωνιολογία και άλλους τομείς, λόγω της απλότητας και της ευκολίας στην εφαρμογή (Bishop, 2006).

Πολυωνυμική Παλινδρόμηση: Επεκτείνει τη γραμμική παλινδρόμηση, επιτρέποντας τη μοντελοποίηση πιο σύνθετων, μη γραμμικών σχέσεων. Αντί για μια ευθεία γραμμή, η πολυωνυμική παλινδρόμηση χρησιμοποιεί πολυώνυμα για να περιγράψει τη σχέση μεταξύ των μεταβλητών. Η βασική διαφορά έγκειται στο γεγονός ότι επιτρέπει μεγαλύτερη ευχέρεια στον προσδιορισμό καμπυλών.

Η μαθηματική της μορφή είναι:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2^2 + \beta_3 X_3^3 + \dots + \beta_n X_n^n + \epsilon$$

Χρησιμοποιείται όταν η σχέση μεταξύ των μεταβλητών δεν είναι γραμμική και τα δεδομένα ακολουθούν μια καμπύλη. Εφαρμογές αυτής της τεχνικής είναι συχνές στις φυσικές επιστήμες, την οικονομετρία και την ανάλυση χρόνου .

Λογιστική Παλινδρόμηση: Η λογιστική παλινδρόμηση, αν και ονομάζεται "παλινδρόμηση", δεν χρησιμοποιείται για την πρόβλεψη συνεχών τιμών, αλλά για την εκτίμηση πιθανοτήτων σε δυαδικά προβλήματα (π.χ. ναι/όχι, αληθές/ψευδές). Εφαρμόζεται κυρίως για ταξινομητικά προβλήματα, όπου η εξαρτημένη μεταβλητή είναι κατηγορική (Bishop, 2006).

Η συνάρτηση λογιστικής που χρησιμοποιείται στη λογιστική παλινδρόμηση έχει την εξής μορφή:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_n X_n)}}$$

Αυτό επιτρέπει τη μετατροπή της γραμμικής εξίσωσης σε μια πιθανότητα μεταξύ 0 και 1. Αυτή η τεχνική χρησιμοποιείται ευρέως σε εφαρμογές όπως η ιατρική διάγνωση, η οικονομική ανάλυση και η ανάλυση της συμπεριφοράς των καταναλωτών.

Ριζική Παλινδρόμηση (Ridge Regression): Η ριζική παλινδρόμηση είναι μια μέθοδος κανονικοποίησης που χρησιμοποιείται όταν υπάρχουν ισχυρές συσχετίσεις μεταξύ των ανεξάρτητων μεταβλητών. Εισάγοντας έναν όρο ποινής στο μοντέλο, η ριζική παλινδρόμηση μειώνει την επίδραση των λιγότερο σημαντικών μεταβλητών και αποτρέπει την υπερπροσαρμογή (overfitting). Η ριζική παλινδρόμηση είναι χρήσιμη σε περιπτώσεις όπου τα δεδομένα περιέχουν πολλές συσχετιζόμενες μεταβλητές και μπορεί να χρησιμοποιηθεί για τη βελτίωση της γενίκευσης του μοντέλου (Ertel).

Lasso Παλινδρόμηση: Η Lasso (Least Absolute Shrinkage and Selection Operator) είναι μια παραλλαγή της γραμμικής παλινδρόμησης που προσθέτει έναν όρο κανονικοποίησης τύπου L1. Ο κύριος στόχος της είναι να πραγματοποιήσει αυτόματη επιλογή μεταβλητών, μειώνοντας τις λιγότερο σημαντικές μεταβλητές σε μηδέν. Χρησιμοποιείται σε περιπτώσεις με μεγάλους αριθμούς χαρακτηριστικών, καθώς επιτρέπει την εκτίμηση της σημαντικότητας των μεταβλητών και τη μείωση της διάστασης του μοντέλου (Ertel).

Υποστήριξη με Διανύσματα (SVR): Η υποστήριξη με διανύσματα για παλινδρόμηση (SVR) είναι μια τεχνική που προσαρμόζεται στις απαιτήσεις μη γραμμικών προβλημάτων. Αντί να εστιάζει στην εύρεση μιας γραμμής που να προσαρμόζεται στα δεδομένα, η SVR αναζητά την καλύτερη γραμμή ή υπερπλάνο με στόχο την ελαχιστοποίηση του σφάλματος, με στόχο την καλή γενίκευση στα δεδομένα. Η μορφή του μοντέλου SVR είναι πιο περίπλοκη και συνήθως χρησιμοποιεί τεχνικές πυρηνικής μετατροπής για την εύρεση της καλύτερης λύσης (Zhang et al., 2022).

ΚΕΦΑΛΑΙΟ 4. ΕΡΕΥΝΗΤΙΚΟ ΜΕΡΟΣ

4.1 ΣΚΟΠΟΣ ΕΡΕΥΝΑΣ – ΜΕΘΟΔΟΛΟΓΙΑ

Ο σκοπός της εργασίας είναι η δημιουργία ενός μοντέλου βραχυχρόνιας πρόβλεψης ατμοσφαιρικής ρύπανσης με έμφαση στα μικροσωματίδια ώστε να είναι δυνατή η έγκαιρη ενημέρωση και προειδοποίηση των αρμόδιων φορέων και των πολιτών που αντιμετωπίζουν ευπάθειες. Η ενσωμάτωση των τεχνικών καθυστερημένων τιμών και κινούμενων μέσων όρων προσφέρει μια βελτιωμένη κατανόηση των χρονικών εξαρτήσεων στις χρονοσειρές ρύπανσης, ενώ η χρήση LSTM παρέχει τη δυνατότητα εκμάθησης σύνθετων σχέσεων που δεν μπορούν να προσεγγιστούν από τα παραδοσιακά μοντέλα. Επίσης η υλοποίηση ενισχύεται από τη δυνατότητα ενσωμάτωσης μετεωρολογικών δεδομένων, γεγονός που βελτιώνει τη γενικευσιμότητα του μοντέλου σε διαφορετικές γεωγραφικές περιοχές και περιβαλλοντικές συνθήκες. Επιλέχθηκαν 3 διαφορετικά μοντέλα ανάλυσης ώστε να υπάρξει σύγκριση μεταξύ των αποτελεσμάτων.

Τα δεδομένα συλλέχθηκαν από ανοιχτές βάσεις δεδομένων καιρού και ποιότητας αέρα. Χρησιμοποιήθηκαν αλγόριθμοι για την σωστή ανάκτηση και αποθήκευση τους ώστε να χρησιμοποιηθούν από το εκπαιδευόμενο μοντέλο. Αυτά τα δεδομένα αποτέλεσαν τη βάση για τη δημιουργία ενός συνόλου χαρακτηριστικών που αντικατοπτρίζουν τη σχέση μεταξύ των περιβαλλοντικών συνθηκών και των επιπέδων ρύπανσης.

Δεδομένα καιρού

Ινστιτούτο Copernicus

url: <https://cds.climate.copernicus.eu/api>

Με τη χρήση του terminal του VS_code έγινε σταδιακή λήψη των δεδομένων καιρού καθώς δεν είναι επιτρεπτή η μαζική λήψη σε μικρό διάστημα. Επιλέχθηκαν τα έτη 2014 έως 2024.

Δεδομένα ρύπων:

World's Air Pollution: Real-time Air Quality Index: <https://waqi.info/#/c/4.429/7.909/2.3z>

aqicn.org/widgets/airnet/A71704/

Αρχικά χρησιμοποιήθηκε API για τη λήψη δεδομένων όμως λόγω χρόνου ήταν πιο εύκολη η συγκεντρωτική λήψη δεδομένων σε συμπιεσμένο αρχείο.

Στη συνέχεια, αναλύθηκαν οι απαιτήσεις της έρευνας και επιλέχθηκαν κατάλληλοι αλγόριθμοι μηχανικής μάθησης, και συγκεκριμένα το Random Forest, το Prophet και το LSTM, που επιτρέπουν την πρόβλεψη χρονοσειρών. Κατά τη διάρκεια της υλοποίησης, δόθηκε έμφαση στη χρήση τεχνικών επεξεργασίας δεδομένων, όπως η δημιουργία χαρακτηριστικών (lags, rolling averages), η κλιμάκωση τιμών και η διαχείριση ελλειπόντων δεδομένων. Αυτό ήταν απαραίτητο καθώς κατά το στάδιο δοκιμών με δεδομένα από διάφορες περιοχές όπου υπήρχαν πολλές κενές τιμές υπήρχαν προβλήματα στην εκπαίδευση του μοντέλου τα οποία οδηγούσαν σε εσφαλμένα αποτελέσματα. Μετά την εκπαίδευση των μοντέλων, εφαρμόστηκαν διαδικασίες αξιολόγησης με βάση μετρικές όπως το MAE, το RMSE και το R^2 . Παράλληλα, έγινε χρήση εργαλείων οπτικοποίησης για τη σύγκριση πραγματικών και προβλεπόμενων τιμών.

Η διαδικασία περιλαμβάνει 3 στάδια, επεξεργασία, εκπαίδευση, αξιολόγηση. Για το στάδιο της εκπαίδευσης χρησιμοποιείται το 70% των δεδομένων για εκπαίδευση και για την δοκιμή το 30%. Στη συνέχεια ακολουθεί η δοκιμή πρόβλεψης. Οι ρύποι που ελέγχονται είναι οι εξής: PM2.5, PM10, CO, SO2 και NO2, σε συνάρτηση με μετεωρολογικά στοιχεία και συγκεκριμένα υγρασία, θερμοκρασία, ατμοσφαιρική πίεση, ταχύτητα και διεύθυνση ανέμου. Επίσης εξετάζεται η εποχή του έτους ως επιπλέον παράγοντας.

Η μεγαλύτερη δυσκολία σε όλη τη διάρκεια της μελέτης ήταν η αυξημένη πιθανότητα λανθασμένης εκπαίδευσης γι' αυτό το λόγο πραγματοποιήθηκαν αρκετές αλλαγές στις παραμέτρους του μοντέλου ώστε να καταστεί δυνατή η εκπαίδευση του και η παραγωγή έγκυρων αποτελεσμάτων.

4.2 ΕΡΓΑΛΕΙΑ ΚΑΙ ΜΕΘΟΔΟΙ

Για την υλοποίηση του συστήματος πρόβλεψης ατμοσφαιρικής ρύπανσης, χρησιμοποιήθηκαν σύγχρονα εργαλεία ανάπτυξης λογισμικού και πλατφόρμες, που εξασφάλισαν την αποδοτική ανάπτυξη, εκτέλεση και αξιολόγηση των αλγορίθμων.

1. Γλώσσα Προγραμματισμού Python

Η Python επελέγη ως η βασική γλώσσα ανάπτυξης λόγω της ευελιξίας της και της πλούσιας υποστήριξης βιβλιοθηκών για μηχανική μάθηση και επεξεργασία δεδομένων.

2. Περιβάλλον Ανάπτυξης: Visual Studio Code (VS Code)

Ο κώδικας αναπτύχθηκε στο VS Code, ένα ελαφρύ και ισχυρό περιβάλλον ανάπτυξης που προσφέρει εργαλεία για debugging, διαχείριση αρχείων και υποστήριξη επεκτάσεων για Python. Το ενσωματωμένο terminal διευκολύνει την εκτέλεση του κώδικα και τη διαχείριση των πακέτων.

3. Βιβλιοθήκες

Pandas και NumPy: Χρησιμοποιήθηκαν για την επεξεργασία δεδομένων, τη δημιουργία χαρακτηριστικών (lags, rolling averages) και την ανάλυση (<https://pandas.pydata.org/>, <https://numpy.org/>).

Scikit-learn: Χρησιμοποιήθηκε για την υλοποίηση και τη βελτιστοποίηση του Random Forest, καθώς και για την αξιολόγηση μετρικών όπως το MAE και το RMSE (<https://scikit-learn.org/>).

TensorFlow/Keras: Υποστηρίζει τη δημιουργία και εκπαίδευση του LSTM μοντέλου, με χρήση callbacks για τη βελτιστοποίηση της εκπαίδευσης (www.tensorflow.org).

Prophet: Χρησιμοποιήθηκε για την ανάλυση τάσεων και εποχικότητας στις χρονοσειρές (<https://prophet.com>).

Matplotlib: Εξασφαλίζει την οπτικοποίηση των αποτελεσμάτων με γραφήματα πραγματικών και προβλεπόμενων τιμών (<https://matplotlib.org>).

Joblib: Διευκολύνει την αποθήκευση και τη φόρτωση των εκπαιδευμένων μοντέλων (<https://joblib.readthedocs.io>).

4. Διαχείριση Εξαρτήσεων

Η διαχείριση των βιβλιοθηκών έγινε μέσω του εργαλείου *pip*, εξασφαλίζοντας ότι όλες οι απαραίτητες βιβλιοθήκες εγκαταστάθηκαν στις σωστές εκδόσεις.

5. Λειτουργικό Περιβάλλον

Το λογισμικό αναπτύχθηκε και εκτελέστηκε σε λειτουργικό σύστημα Windows, με δυνατότητα μεταφοράς σε άλλα συστήματα λόγω της χρήσης εργαλείων ανοιχτού κώδικα.

6. Αρχεία Δεδομένων

Τα δεδομένα αποθηκεύτηκαν και επεξεργάστηκαν σε μορφή CSV, που διευκολύνει τη διαχείριση μεγάλων συνόλων δεδομένων και την εισαγωγή τους στα μοντέλα.

4.3 ΑΛΓΟΡΙΘΜΟΙ

Όπως αναφέρθηκε χρησιμοποιήθηκαν 3 διαφορετικές μέθοδοι ανάλυσης. Πιο αναλυτικά, παλινδρόμηση Random forest (τυχαίου δάσους) βασιζόμενη στο SciKit learn, το εργαλείο Prophet της meta καθώς και ένα νευρωνικό δίκτυο βαθιάς μάθησης και συγκεκριμένα RNN όπου χρησιμοποιήθηκε η βιβλιοθήκη TensorFlow.

4.3.1 Random Forest

Το Random Forest είναι ένα ensemble μοντέλο που χρησιμοποιείται για προβλήματα παλινδρόμησης ή ταξινόμησης. Στην περίπτωση αυτή, χρησιμοποιείται για πρόβλεψη της τιμής ενός ρύπου.

Λογική Random Forest:

Δημιουργία Δεδομένων:

Χρησιμοποιείται προεπεξεργασία για τη δημιουργία χαρακτηριστικών από τις τιμές του ρύπου.

Δημιουργούνται "lags" (καθυστερημένες τιμές) και "rolling averages" (κινούμενοι μέσοι όροι), που εισάγουν πληροφορίες για τη χρονική εξάρτηση.

Εκπαίδευση:

Το μοντέλο εκπαιδεύεται με χαρακτηριστικά εισόδου (X) που περιλαμβάνουν lags και rolling averages, και εξόδου (y) που είναι οι τιμές του ρύπου.

Χρησιμοποιείται RandomizedSearchCV για την επιλογή των καλύτερων υπέρ-παραμέτρων του μοντέλου.

Αξιολόγηση:

Αξιολογείται σε δεδομένα δοκιμής με μετρικές όπως MAE (Mean Absolute Error), RMSE (Root Mean Squared Error) και R^2 (Coefficient of Determination).

Εξάγονται γραφήματα για να συγκρίνει τις πραγματικές τιμές με τις προβλεπόμενες.

4.3.2. Prophet

Το Prophet είναι ένα μοντέλο χρονοσειρών, σχεδιασμένο για εύκολη χρήση και αυτόματη ανάλυση εποχικότητας και τάσεων.

Λογική Prophet:

Διαχείριση Δεδομένων:

Απαιτείται τα δεδομένα να είναι σε συγκεκριμένη μορφή: μια στήλη "ds" (timestamps) και μια στήλη "y" (τιμές ρύπου).

Εκπαίδευση:

Το Prophet αναλύει την τάση (trend), εποχικότητα (seasonality) και τυχόν διακοπές (holidays) στα δεδομένα.

Εκπαιδεύεται ώστε να δημιουργήσει μια καμπύλη πρόβλεψης που προσαρμόζεται στις ιδιαιτερότητες της χρονοσειράς.

Αξιολόγηση:

Χρησιμοποιείται για πρόβλεψη τιμών σε δεδομένα δοκιμής.

Οι προβλέψεις συγκρίνονται με τις πραγματικές τιμές και υπολογίζονται οι ίδιες μετρικές όπως και στο Random Forest.

4.3.3. RNN/LSTM

Το LSTM (Long Short-Term Memory) είναι ένα είδος RNN (Recurrent Neural Network) που είναι κατάλληλο για χρονοσειρές, λόγω της ικανότητάς του να "θυμάται" σημαντικές πληροφορίες για μεγάλο χρονικό διάστημα.

Λογική LSTM:

Μετατροπή Δεδομένων

Τα δεδομένα χωρίζονται σε ακολουθίες (sequence_length) που χρησιμοποιούνται για εκπαίδευση.

Για παράδειγμα, αν το sequence_length είναι 10, κάθε "παράθυρο" δεδομένων περιλαμβάνει τις τελευταίες 10 τιμές για να προβλεφθεί η επόμενη.

Αρχιτεκτονική Μοντέλου

Το LSTM αποτελείται από "cells" που χρησιμοποιούν πύλες (gates) για να διατηρούν ή να απορρίπτουν πληροφορίες.

Μπορεί να περιλαμβάνει πολλαπλά στρώματα LSTM και ένα πλήρως συνδεδεμένο (dense) στρώμα εξόδου.

Εκπαίδευση

Το μοντέλο εκπαιδεύεται με Adam Optimizer και Mean Squared Error (MSE).

Χρησιμοποιεί batches για αποδοτική εκπαίδευση.

Αξιολόγηση

Το μοντέλο προβλέπει τις τιμές ρύπων για τα δεδομένα δοκιμής.

Εξάγονται μετρικές και γραφήματα για να συγκριθούν οι προβλέψεις με τις πραγματικές τιμές.

4.3.4 Σύγκριση

Στον παρακάτω πίνακα παρουσιάζονται συγκριτικά τα 3 μοντέλα που αναπτύχθηκαν ως προς τον τρόπο που χρησιμοποιούνται και τα πλεονεκτήματα και μειονεκτήματα καθενός.

Μοντέλο	Χρήση	Πλεονεκτήματα	Μειονεκτήματα
Random Forest	Στατικές σχέσεις μεταξύ δεδομένων	Απλό στην υλοποίηση, γρήγορο, δεν απαιτεί πολλά δεδομένα	Δεν κατανοεί χρονοσειρές, απαιτεί δημιουργία χαρακτηριστικών
Prophet	Ανάλυση χρονοσειρών	Αυτόματη ανάλυση τάσεων/εποχικότητας, εύκολη χρήση	Δεν είναι ευέλικτο σε περίπλοκες δυναμικές
LSTM (RNN)	Σύνθετη ανάλυση χρονοσειρών	Μπορεί να μάθει περίπλοκες χρονικές εξαρτήσεις	Απαιτεί περισσότερους πόρους, δύσκολο να εκπαιδευτεί

Πίνακας 3. Σύγκριση μοντέλων

ΚΕΦΑΛΑΙΟ 5. ΥΛΟΠΟΙΗΣΗ

5.1 ΜΕΘΟΔΟΙ

Βιβλιοθήκες

Ο κώδικας εισάγει μια σειρά από βιβλιοθήκες: Pandas, Numpy (επεξεργασία δεδομένων), Joblib (αποθήκευση και ανάκτηση εκπαιδευμένων μοντέλων), Matplotlib (οπτικοποίηση), Scikit-learn (μοντέλα και μετρικές), Prophet (χρονοσειρές πρόβλεψης), TensorFlow/Keras (LSTM).

Σταθερές

TRAIN_CSV & TEST_CSV: Ορίζουν τα ονόματα των αρχείων δεδομένων για εκπαίδευση και αξιολόγηση.

Κύριες Λειτουργίες

α. Random Forest

-train_and_save_model: Εκπαιδεύει ένα Random Forest μοντέλο για έναν συγκεκριμένο ρύπο. Δημιουργεί νέα χαρακτηριστικά (lags & rolling averages). Χρησιμοποιεί RandomizedSearchCV για βελτιστοποίηση υπερπαραμέτρων. Αποθηκεύει το μοντέλο σε αρχείο .pkl.

-load_model_and_evaluate: Φορτώνει το αποθηκευμένο μοντέλο και το αξιολογεί σε δεδομένα δοκιμής. Υπολογίζει μετρικές (MAE, RMSE, R^2) και δημιουργεί γραφήματα.

β. Prophet

-train_prophet_and_save_model: Εκπαιδεύει ένα μοντέλο Prophet για έναν συγκεκριμένο ρύπο. Αποθηκεύει το μοντέλο σε αρχείο .pkl.

-prophet_evaluate: Φορτώνει και αξιολογεί το Prophet μοντέλο. Παράγει μετρικές και γραφήματα.

γ. RNN (LSTM)

-train_rnn_and_save_model: Δημιουργεί δεδομένα χρονοσειρών (create_tf_dataset). Εκπαιδεύει ένα LSTM-based RNN μοντέλο για έναν ρύπο. Αποθηκεύει το μοντέλο και το scaler.

-evaluate_rnn_model: Αξιολογεί το LSTM μοντέλο σε δεδομένα δοκιμής. Υπολογίζει μετρικές και οπτικοποιεί τα αποτελέσματα.

Βοηθητικές Συναρτήσεις

-create_tf_dataset: Δημιουργεί tf.data.Dataset για εισαγωγή δεδομένων στο LSTM.

Διεπαφή Χρήστη

Είναι σχεδιασμένος να εκτελείται ως script: Εμφανίζει διάφορες λειτουργίες (1-6). Ζητά από τον χρήστη να επιλέξει λειτουργία και ρύπο (π.χ., pm25, o3).

Δομή κώδικα

Προεπεξεργασία δεδομένων: Δημιουργεί χαρακτηριστικά lag και rolling averages. Αποθήκευση και φόρτωση μοντέλων: Χρησιμοποιεί αρχεία .pkl και .keras. Οπτικοποίηση αποτελεσμάτων: Χρησιμοποιεί scatter plots για σύγκριση προβλέψεων και πραγματικών τιμών. Αναλυτικά στο παράρτημα.

5.2 METABΛΗΤΕΣ

Σταθερές

Οι σταθερές αυτές χρησιμοποιούνται για την οργάνωση των αρχείων δεδομένων:

TRAIN_CSV: Το όνομα του αρχείου που περιέχει τα δεδομένα εκπαίδευσης.

TEST_CSV: Το όνομα του αρχείου που περιέχει τα δεδομένα δοκιμής.

Αυτές οι μεταβλητές χρησιμοποιούνται στην αρχική φόρτωση δεδομένων με Pandas (pd.read_csv).

Γενικές μεταβλητές

Ονόματα Ρύπων (Pollutants):

Τα ονόματα που χρησιμοποιούνται για κάθε ρύπο (π.χ., pm25, o3, no2) πιθανόν αποθηκεύονται ή παρέχονται ως είσοδος από τον χρήστη.

Υπέρ-παράμετροι (Hyperparameters):

Για Random Forest:

Στο RandomizedSearchCV, οι υπέρ-παράμετροι του αλγορίθμου όπως:

n_estimators (αριθμός δέντρων).

max_depth (μέγιστο βάθος).

min_samples_split (ελάχιστος αριθμός δειγμάτων για διαίρεση κόμβου).

Για LSTM: Παράμετροι όπως:

batch_size, sequence_length (παράθυρο δεδομένων),

learning_rate, epochs.

Scaler:

Χρησιμοποιείται για την κλιμάκωση των χαρακτηριστικών (π.χ., MinMaxScaler ή StandardScaler).

Αποθηκεύεται και φορτώνεται μαζί με τα μοντέλα για συνέπεια.

Διαδρομές Μοντέλων

model_save_path: Αποθηκεύει τη διαδρομή για κάθε εκπαιδευμένο μοντέλο (π.χ., .pkl για Random Forest, .keras για LSTM, .pkl για Prophet).

Αυτή η διαδρομή είναι χρήσιμη για τη φόρτωση και αποθήκευση μοντέλων.

5.3 ΡΟΗ ΕΚΠΑΙΔΕΥΣΗΣ

5.3.1 Στάδια

Επεξεργασία Δεδομένων:

Καθαρισμός, δημιουργία χαρακτηριστικών (Random Forest).

Δημιουργία χρονικών παραθύρων (LSTM).

Μορφοποίηση δεδομένων για Prophet.

Εκπαίδευση Μοντέλου:

Κάθε μοντέλο εκπαιδεύεται στα δεδομένα εκπαίδευσης και βελτιστοποιείται.

Αξιολόγηση:

Τα αποτελέσματα αξιολογούνται με κοινές μετρικές:

MAE (Mean Absolute Error): Μέση απόλυτη διαφορά.

RMSE (Root Mean Squared Error): Ρίζα μέσης τετραγωνικής απόκλισης.

R² (Coefficient of Determination): Εξήγηση διακύμανσης.

Οπτικοποίηση:

Δημιουργούνται γραφήματα για να συγκριθούν οι προβλέψεις με τις πραγματικές τιμές.

5.3.2 Random Forest - Δημιουργία Χαρακτηριστικών

Το Random Forest είναι ένα στατιστικό μοντέλο που βασίζεται στη δημιουργία πολλαπλών δέντρων απόφασης. Για να λειτουργήσει σωστά σε χρονοσειρές, χρειάζεται τη δημιουργία νέων χαρακτηριστικών (features) από τα δεδομένα εισόδου. Ο κώδικας προσθέτει τα παρακάτω χαρακτηριστικά

Καθυστερημένες τιμές (Lags):

Δημιουργούνται στήλες που περιέχουν τιμές ρύπων από προηγούμενες χρονικές στιγμές.

Παράδειγμα:

- Αν η τρέχουσα τιμή είναι στο t , τότε:

- $\text{pollutant_lag_1} = \text{pollutant}(t-1)$
- $\text{pollutant_lag_2} = \text{pollutant}(t-2)$, κ.ο.κ.

Αυτό βοηθά το Random Forest να μάθει από την ιστορική συμπεριφορά.

Κινούμενος Μέσος Όρος (Rolling Average):

Υπολογίζεται ο μέσος όρος των τιμών ενός ρύπου σε ένα παράθυρο π.χ. 3 ημερών:

$$\text{pollutant_rolling_avg} = \frac{\text{pollutant}(t-1) + \text{pollutant}(t-2) + \text{pollutant}(t-3)}{3}$$

Αυτό εξομαλύνει τις διακυμάνσεις και δίνει στο μοντέλο πληροφορίες για γενικές τάσεις.

Μετεωρολογικά χαρακτηριστικά:

Χρησιμοποιούνται οι παρακάτω παράγοντες:

tp (συνολικός υετός),

t2m (θερμοκρασία),

u10, v10 (συνιστώσες ανέμου),

sp (ατμοσφαιρική πίεση).

Οι παράγοντες καιρού συνδέονται με τη ρύπανση και βοηθούν το μοντέλο να κατανοήσει περιβαλλοντικές συνθήκες.

5.3.3 Prophet - Ανάλυση Χρονοσειρών

Το Prophet είναι ένα μοντέλο ειδικά σχεδιασμένο για χρονοσειρές. Χωρίζει τη χρονοσειρά σε τμήματα:

Trend (τάση):

Η μακροχρόνια αλλαγή στις τιμές όπως η αύξηση της ρύπανσης λόγω αύξησης πληθυσμού.

Seasonality (εποχικότητα):

Περιοδικές αλλαγές στις τιμές όπως π.χ. οι τιμές NO₂ είναι υψηλότερες το χειμώνα λόγω θέρμανσης.

Holidays (Ειδικές χρονικές περίοδοι):

Μοναδικά γεγονότα που επηρεάζουν τη ρύπανση, όπως αργίες ή lockdowns.

Εκπαίδευση Prophet

1. Τα δεδομένα εισόδου πρέπει να έχουν τη μορφή:
 - ο Στήλη ds: Οι χρονικές στιγμές (timestamps).
 - ο Στήλη y: Οι τιμές του ρύπου.
2. Το Prophet αναλύει τις τάσεις και την εποχικότητα χρησιμοποιώντας γραμμικά και μη γραμμικά μοντέλα.
3. Η πρόβλεψη γίνεται προσθέτοντας: $y(t) = \text{trend}(t) + \text{seasonality}(t) + \text{holidays}(t) + \epsilon y(t)$
$$\text{trend}(t) + \text{seasonality}(t) + \text{holidays}(t) + \epsilon y(t)$$

$$\text{trend}(t) + \text{seasonality}(t) + \text{holidays}(t) + \epsilon$$
 Όπου ϵ είναι το θόρυβος (noise).

5.3.4 LSTM (Long Short-Term Memory)

Το LSTM είναι ιδανικό για χρονοσειρές επειδή μπορεί να "θυμάται" σημαντικές πληροφορίες για μεγάλο χρονικό διάστημα.

Αρχιτεκτονική LSTM

Η λογική του LSTM βασίζεται στις κυψέλες μνήμης (memory cells), οι οποίες περιέχουν τρεις κύριες πύλες:

1. Forget Gate (Πύλη διαγραφής):
 - ο Αποφασίζει ποιες πληροφορίες από το παρελθόν να "ξεχάσει".
 - ο Υπολογισμός: $f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$
2. Input Gate (Πύλη εισόδου):

- Αποφασίζει ποιες νέες πληροφορίες να αποθηκεύσει.
- Υπολογισμός: $i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
 $= \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$

3. Output Gate (Πύλη εξόδου):

- Καθορίζει τι θα περάσει στο επόμενο στάδιο.
- Υπολογισμός: $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
 $o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$

Λογική Εκπαίδευσης LSTM

1. Είσοδος:

- Τα δεδομένα οργανώνονται σε ακολουθίες (sequence_length).
- Για κάθε παράθυρο δεδομένων, προβλέπεται η επόμενη τιμή.

2. Εκπαίδευση:

- Ορίζεται ένα μοντέλο με τουλάχιστον ένα LSTM στρώμα.
- Προστίθεται ένα πλήρως συνδεδεμένο (dense) στρώμα εξόδου.

3. Αξιολόγηση:

- Μετά την εκπαίδευση, το μοντέλο ελέγχεται σε δεδομένα δοκιμής.
- Υπολογίζονται οι αποκλίσεις μεταξύ των προβλέψεων και των πραγματικών τιμών.

5.4 ΧΑΡΑΚΤΗΡΙΣΤΙΚΑ ΕΚΠΑΙΔΕΥΣΗΣ

1. Εκπαίδευση Random Forest

Το Random Forest εκπαιδεύεται σε δεδομένα που έχουν προετοιμαστεί με επιπλέον χαρακτηριστικά όπως lags και rolling averages.

Διαδικασία

1. Διαμόρφωση δεδομένων:

- Χαρακτηριστικά Εισόδου (X): Περιλαμβάνουν τις μετεωρολογικές παραμέτρους (tp, t2m, u10, v10, sp) και τα επιπλέον χαρακτηριστικά (lag_1, lag_2, rolling_avg).
- Ετικέτες (y): Η τιμή του ρύπου (π.χ., pm25, no2) που το μοντέλο προσπαθεί να προβλέψει.

2. Κλιμάκωση δεδομένων:

- Χρησιμοποιείται StandardScaler για να φέρει όλα τα χαρακτηριστικά στην ίδια κλίμακα (μέσος όρος = 0, τυπική απόκλιση = 1).

3. Επιλογή μοντέλου:

- Το RandomForestRegressor χρησιμοποιείται ως βασικός αλγόριθμος.
- Ορίζεται ένα πλέγμα υπερπαραμέτρων:
 - Αριθμός δέντρων (n_estimators).
 - Μέγιστο βάθος δέντρων (max_depth).
 - Ελάχιστα δείγματα για διαίρεση κόμβου (min_samples_split).
- Η βέλτιστη ρύθμιση επιλέγεται με τη ρουτίνα RandomizedSearchCV.

4. Εκπαίδευση και αποθήκευση:

- Το μοντέλο εκπαιδεύεται στα δεδομένα εκπαίδευσης με την εντολή:

```
rf_random_search.fit(X_scaled, y)
```

- Το εκπαιδευμένο μοντέλο αποθηκεύεται με τη joblib:

```
joblib.dump((best_model, scaler), MODEL_FILE)
```

2. Εκπαίδευση Prophet

Το Prophet είναι πιο αυτόνομο και απαιτεί ελάχιστη προεπεξεργασία.

Διαδικασία εκπαίδευσης

1. Διαμόρφωση δεδομένων:

- Τα δεδομένα προετοιμάζονται στη μορφή:
 - Στήλη ds: Οι χρονικές στιγμές.
 - Στήλη y: Οι τιμές του ρύπου.

Παράδειγμα

ds	y
2023-01-01	35.5
2023-01-02	36.2

2. Ορισμός μοντέλου:

- Δημιουργείται ένα αντικείμενο Prophet:

```
model = Prophet()
```

3. Εκπαίδευση:

- Το μοντέλο μαθαίνει την τάση και την εποχικότητα μέσω των δεδομένων εκπαίδευσης:

```
model.fit(prophet_df)
```

4. Αποθήκευση:

- ο Αποθηκεύεται με joblib:

```
joblib.dump(model, PROPHET_MODEL_FILE)
```

3. Εκπαίδευση LSTM (RNN)

Η εκπαίδευση του LSTM είναι πιο περίπλοκη λόγω της ανάγκης να μετατραπούν τα δεδομένα σε χρονικές ακολουθίες.

Διαδικασία εκπαίδευσης

1. Προετοιμασία δεδομένων:

- ο Χρονικές ακολουθίες:
- ο Τα δεδομένα οργανώνονται σε παράθυρα (sequence_length).

Παράδειγμα

- ο Δεδομένα: [10,20,30,40,50][10, 20, 30, 40, 50][10,20,30,40,50].
- ο Παράθυρα:
 - Είσοδος (XXX): [10,20,30][10, 20, 30][10,20,30], Έξοδος (yyy): 404040.
 - Είσοδος (XXX): [20,30,40][20, 30, 40][20,30,40], Έξοδος (yyy): 505050.

2. Δημιουργία tf.data.Dataset:

- ο Χρησιμοποιείται η συνάρτηση create_tf_dataset για να παράγει batches δεδομένων:

```
dataset = create_tf_dataset(X_scaled, y, sequence_length, batch_size)
```

3. Αρχιτεκτονική μοντέλου:

- Το LSTM μοντέλο περιλαμβάνει ένα ή περισσότερα LSTM στρώματα:

```
tf.keras.layers.LSTM(64, activation='tanh', return_sequences=True)
```

- Dropout για αποφυγή υπερεκπαίδευσης.

- Πλήρως συνδεδεμένα (Dense) στρώματα για την έξοδο.

4. Ρύθμιση εκπαίδευσης:

- ο Ορίζεται ο βελτιστοποιητής (Adam Optimizer) και η συνάρτηση απώλειας (mse):

```
model.compile(optimizer='adam', loss='mse')
```

5. Callbacks:

- ο Χρησιμοποιούνται για αποδοτικότερη εκπαίδευση:
 - EarlyStopping: Διακόπτει την εκπαίδευση αν η απόδοση δεν βελτιώνεται.
 - ReduceLROnPlateau: Μειώνει το learning rate αν η απώλεια παραμένει σταθερή.

6. Εκπαίδευση:

Το μοντέλο εκπαιδύεται με τη συνάρτηση:

```
model.fit(dataset, epochs=500, validation_data=dataset, callbacks=[...])
```

7. Αποθήκευση μοντέλου:

Το LSTM και ο scaler αποθηκεύονται ξεχωριστά:

```
model.save(RNN_MODEL_FILE)
```

```
joblib.dump(scaler, SCALER_FILE)
```

5.5 ΣΥΓΚΡΙΤΙΚΗ ΑΝΑΛΥΣΗ

Χαρακτηριστικό	Random Forest	Prophet	LSTM
Τύπος προσέγγισης	Μηχανική μάθηση	Χρονοσειρές	Νευρωνικά δίκτυα
Διαχείριση χρονοσειρών	Απαραίτητα lags και rolling averages	Αυτόματη	Ενσωματωμένη
Απαιτήσεις δεδομένων	Μικρή ποσότητα	Δεδομένα με τάσεις ή εποχικότητα	Μεγάλη ποσότητα, ορθά δομημένα
Ερμηνεία	Εύκολη	Μέτρια	Δύσκολη

Πίνακας 4. Συγκριτική Ανάλυση

5.6 ΓΕΝΙΚΕΣ ΠΑΡΑΤΗΡΗΣΕΙΣ

- Διαφορές στις μεθόδους εκπαίδευσης:
 - ο Το Random Forest βασίζεται σε στατικές σχέσεις μεταξύ χαρακτηριστικών.
 - ο Το Prophet μαθαίνει αυτόματα την τάση και εποχικότητα χωρίς πρόσθετα χαρακτηριστικά.
 - ο Το LSTM είναι δυναμικό και αναλύει τη χρονολογική εξάρτηση με παραθυροποίηση.
- Εργαλεία αξιολόγησης:
 - ο Και τα τρία μοντέλα αξιολογούνται με τις ίδιες μετρικές (MAE, RMSE, R^2), που επιτρέπουν δίκαιη σύγκριση.

ΚΕΦΑΛΑΙΟ 6. ΔΟΚΙΜΕΣ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

6.1 ΑΝΑΛΥΣΗ ΓΡΑΦΗΜΑΤΩΝ

Εκτελέστηκε ο κώδικας επανελλειμένα για όλους τους ρύπους και όλα τα μοντέλα. Στο παράτημα παρατείνονται στιγμιότυπα οθόνης από την εκπαίδευση και την δοκιμή πρόβλεψης. Αφού ολοκληρωθεί η διαδικασία της πρόγνωσης εμφανίζεται στην οθόνη το γράφημα για την απόδοση του μοντέλου, το καθένα δείχνει την απόδοση του εκάστοτε αλγορίθμου για την πρόβλεψη της συγκέντρωσης ενός ρύπου στη δοκιμαστική ομάδα δεδομένων.

6.1.1 Στοιχεία γραφημάτων

Άξονας x (Actual): Αντιπροσωπεύει τις πραγματικές τιμές των ρύπων από τα δεδομένα δοκιμής (ground truth).

Άξονας y (Predicted): Αντιπροσωπεύει τις προβλεπόμενες τιμές από το κάθε μοντέλο που χρησιμοποιήσαμε.

Διακεκομμένη κόκκινη γραμμή: Είναι η διαγώνιος $y=x$ ή $xy=x$, η οποία δείχνει πού οι προβλεπόμενες τιμές είναι ίσες με τις πραγματικές τιμές. Όσο πιο κοντά βρίσκονται τα σημεία στη γραμμή, τόσο καλύτερη είναι η απόδοση του μοντέλου.

Μπλε σημεία: Κάθε σημείο αντιπροσωπεύει ένα δείγμα από τα δεδομένα δοκιμής. Οι συντεταγμένες του δείχνουν την πραγματική τιμή NO_2 (στον άξονα x) και την αντίστοιχη πρόβλεψη του μοντέλου (στον άξονα y).

Όταν τα περισσότερα σημεία είναι κοντά στη διαγώνιο $y=x$ ή $xy=x$, υποδηλώνουν ότι το μοντέλο κάνει καλές προβλέψεις για πολλές περιπτώσεις, όταν όμως υπάρχουν αποκλίσεις

(σημεία μακριά από τη γραμμή), δείχνουν περιπτώσεις όπου το μοντέλο έκανε λάθος στην πρόβλεψη.

6.1.2 Ερμηνεία αποτελεσμάτων

Γενική ακρίβεια: Δείχνει ότι το μοντέλο δίνει αξιόπιστες προβλέψεις όταν τα περισσότερα σημεία ακολουθούν τη διαγώνιο.

Σφάλματα: Οι αποκλίσεις υποδηλώνουν σημεία όπου το μοντέλο υποτίμησε ή υπερεκτίμησε τις τιμές.

Εύρος τιμών: Το μοντέλο λειτουργεί σε ένα εύρος πραγματικών τιμών όπως φαίνεται από τον άξονα x.

Root Mean Squared Error (RMSE):

- Το RMSE μετρά το μέσο σφάλμα μεταξύ των προβλεπόμενων και πραγματικών τιμών.
- Μικρότερο RMSE σημαίνει ότι το μοντέλο κάνει μικρότερα σφάλματα.

Mean Absolute Error (MAE):

- Μετρά τη μέση απόλυτη διαφορά μεταξύ των πραγματικών και προβλεπόμενων τιμών.
- Εύκολη στην κατανόηση, καθώς δίνει τη μέση απόσταση μεταξύ πρόβλεψης και πραγματικότητας.

R-squared (R^2):

- Μετρά το ποσοστό της συνολικής διακύμανσης των δεδομένων που εξηγείται από το μοντέλο. Τιμές κοντά στο 1 δείχνουν ότι το μοντέλο ερμηνεύει σωστά τα δεδομένα.

RMSE και MAE:

- Αν η διαφορά μεταξύ RMSE και MAE είναι μικρή, το μοντέλο αντιμετωπίζει παρόμοια μεγέθη σφαλμάτων (χωρίς μεγάλα outliers).
- Αν η διαφορά είναι μεγάλη, τότε υπάρχουν κάποιες εξαιρετικά μεγάλες αποκλίσεις που πρέπει να εξεταστούν.

R^2 :

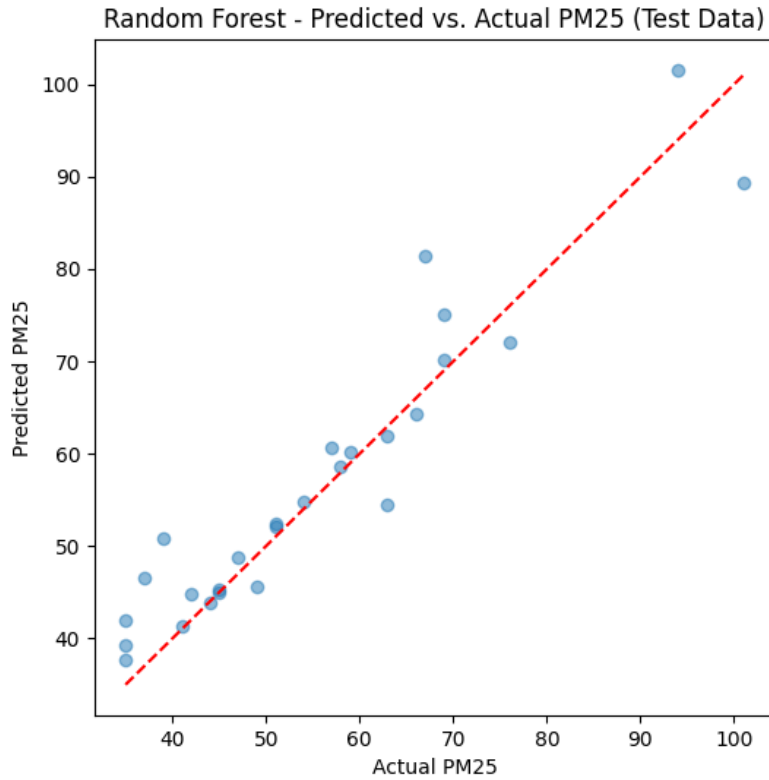
- Αν το R^2 είναι κοντά στο 1, τότε το μοντέλο εξηγεί καλά τη διακύμανση των δεδομένων.

- Αν είναι χαμηλότερο, ίσως χρειάζεται βελτιώσεις (όπως περισσότερα δεδομένα ή προσαρμογή υπερπαραμέτρων).

Τα γραφήματα που ακολουθούν απεικονίζουν την απόδοση των μοντέλων στην πρόβλεψη των συγκεντρώσεων ρύπων, συγκρίνοντας τις προβλεπόμενες τιμές με τις πραγματικές μετρήσεις των δεδομένων δοκιμής. Στον οριζόντιο άξονα εμφανίζονται οι πραγματικές τιμές PM2.5, ενώ στον κατακόρυφο οι αντίστοιχες προβλέψεις του μοντέλου. Όπως αναφέρθηκε προηγουμένως, η κόκκινη γραμμή αντιπροσωπεύει την ιδανική σχέση $y = x$, που σημαίνει ότι οι τέλει προβλέψεις θα έπρεπε να βρίσκονται ακριβώς πάνω σε αυτή τη γραμμή. Όσο πιο κοντά βρίσκονται τα σημεία στην κόκκινη γραμμή, τόσο πιο ακριβείς θεωρούνται οι προβλέψεις. Για κάθε δοκιμή παρουσιάζεται το γράφημα και οι τιμές των RMSE, MAE και R^2 και ακολουθεί αναλυτική περιγραφή.

6.2 RANDOM FOREST

PM2.5



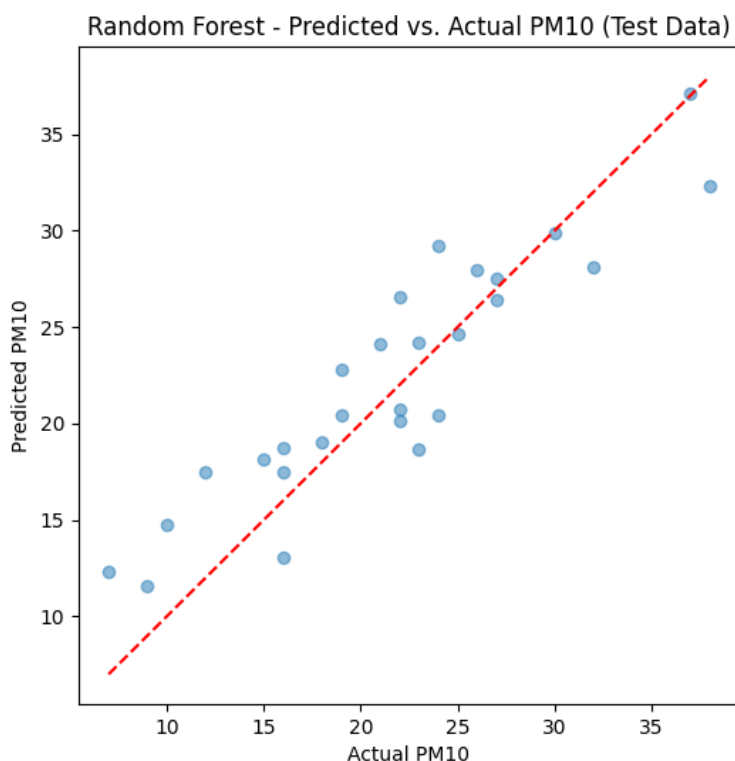
Γράφημα 1. Random Forest PM2.5

MAE	4.0366
RMSE	5.7075
R ²	0.8826

Το πρώτο γράφημα παρουσιάζει την απόδοση του μοντέλου Random Forest στην πρόβλεψη των συγκεντρώσεων PM2.5. Οι περισσότερες προβλέψεις συγκεντρώνονται κοντά στη γραμμή, γεγονός που δείχνει ότι το μοντέλο αποδίδει αρκετά καλά. Ωστόσο, υπάρχουν μερικές αποκλίσεις ιδιαίτερα σε υψηλές τιμές PM2.5, όπου οι προβλέψεις τείνουν να είναι ελαφρώς χαμηλότερες από τις πραγματικές τιμές. Αντίθετα στις χαμηλές και μεσαίες συγκεντρώσεις οι

προβλέψεις είναι πιο ακριβείς και η κατανομή των σημείων είναι πιο ομοιόμορφη γύρω από τη γραμμή $y = x$. Αυτό υποδηλώνει ότι το μοντέλο έχει μεγαλύτερη ευστοχία στις συνηθισμένες τιμές PM2.5, αλλά δυσκολεύεται να γενικεύσει στις ακραίες τιμές. Η τάση αυτή ενδέχεται να οφείλεται σε υπερβολική προσαρμογή στα δεδομένα εκπαίδευσης, μια κατάσταση γνωστή ως overfitting. Επίσης έχει υψηλό R^2 , χαμηλό MAE και μέτριο RMSE. Η διαφορά μεταξύ MAE και RMSE δείχνει καλή ακρίβεια αλλά και ότι υπάρχουν μερικές μεγαλύτερες αποκλίσεις (outliers) που ίσως επηρεάζουν το σφάλμα.

PM10

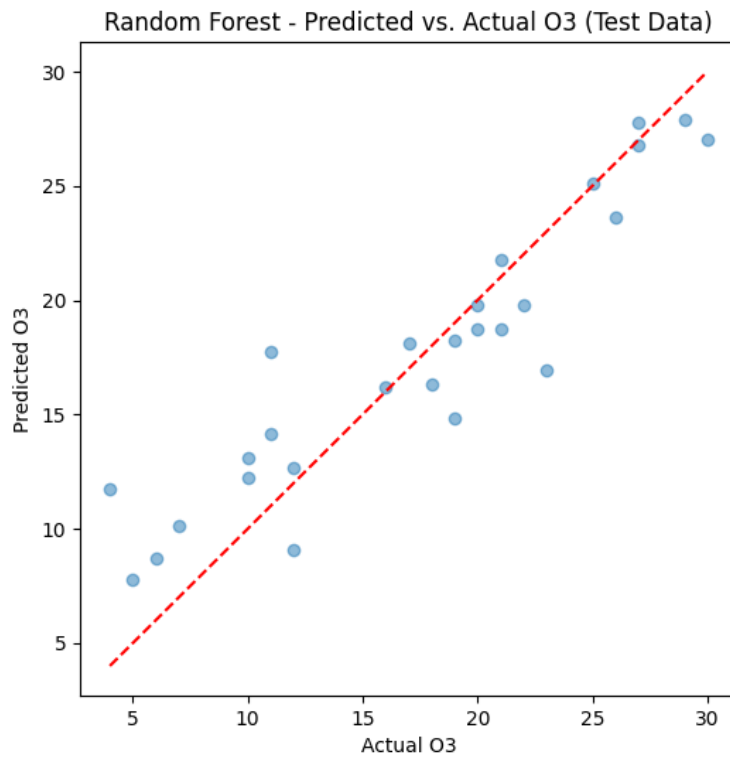


Γράφημα 2. Random Forest PM10

MAE	2.7047
RMSE	3.2243
R^2	0.8192

Συνεχίζοντας, στο επόμενο γράφημα απεικονίζεται η απόδοση του μοντέλου Random Forest για την πρόβλεψη της συγκέντρωσης PM10. Όπως φαίνεται τα περισσότερα σημεία βρίσκονται κοντά στη γραμμή $y=xy = xy=x$, κάτι που δείχνει ότι το μοντέλο αποδίδει αρκετά καλά στις προβλέψεις του. Οι αποκλίσεις είναι μικρότερες σε σχέση με το γράφημα PM2.5, που υποδηλώνει μεγαλύτερη ακρίβεια για τις συγκεντρώσεις PM10. Όσον αφορά τη συμπεριφορά στις ακραίες τιμές παρατηρείται ότι για χαμηλές τιμές (10-20), οι προβλέψεις είναι σχετικά ακριβείς ενώ για υψηλότερες τιμές (30-35), συνεχίζει να αποδίδει ικανοποιητικά, με μικρές αποκλίσεις. Γενικά φαίνεται να διατηρεί καλή ισορροπία και για μικρές και για μεγάλες συγκεντρώσεις. Οι αποκλίσεις για τις χαμηλές τιμές μπορεί να προκύπτουν από ανεπαρκή εκπαίδευση του μοντέλου σε αυτή τη συγκεκριμένη περιοχή ή από τον θόρυβο στα δεδομένα, γενικότερα όμως δεν υπάρχουν σαφείς συστηματικές αστοχίες. Όπως και στο PM2.5 έχει σχετικά υψηλό R^2 , που σημαίνει μεγάλη ακρίβεια πρόβλεψης. Επίσης χαμηλό MAE και μέτριο RMSE με μικρή διαφορά μεταξύ τους, αυτό δείχνει καλή ακρίβεια και μερικές μεγαλύτερες αποκλίσεις.

O₃



Γράφημα 3. Random Forest O₃

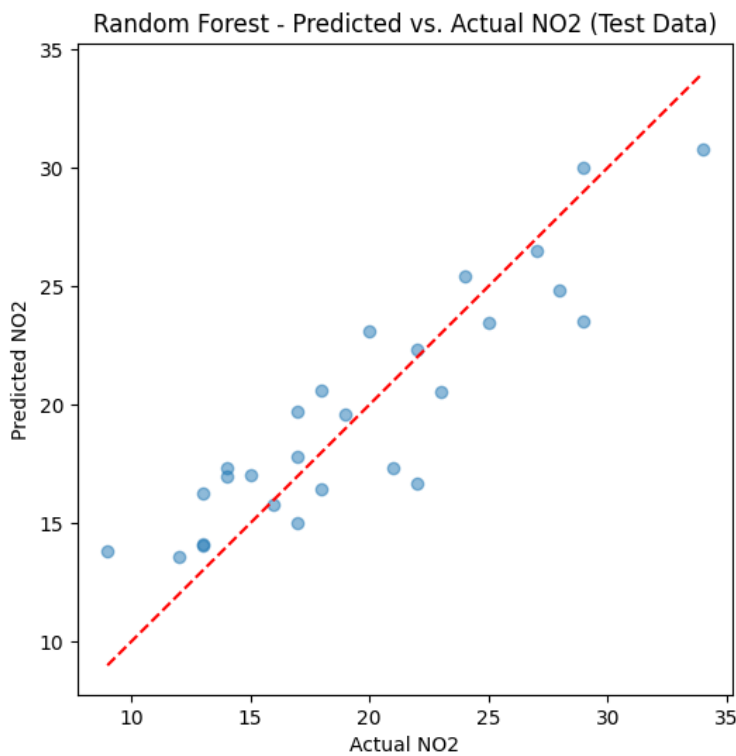
MAE	2.3506
RMSE	3.0464
R ²	0.8362

Εδώ απεικονίζονται οι προβλέψεις των συγκεντρώσεων όζοντος (O₃) και επίσης φαίνεται μια ικανοποιητική ποιότητα πρόβλεψης. Τα περισσότερα σημεία βρίσκονται σχετικά κοντά στην κόκκινη γραμμή και η διασπορά γύρω από την ιδανική γραμμή είναι μικρότερη σε σύγκριση με προηγούμενα γραφήματα. Στις τιμές άνω των 25 οι προβλέψεις βρίσκονται πολύ κοντά στις πραγματικές, αποδεικνύοντας ότι διαχειρίζεται καλά τις υψηλές συγκεντρώσεις O₃ με αποκλίσεις

να εμφανίζονται στις χαμηλές συγκεντρώσεις, όπου παρατηρούνται περιπτώσεις υποεκτίμησης ή υπερεκτίμησης.

Σύμφωνα με το διάγραμμα αποδίδει αξιόπιστες προβλέψεις, με μεγαλύτερη ακρίβεια σε σύγκριση με άλλα ατμοσφαιρικά στοιχεία όπως το PM25 ή το PM10. Επίσης οι αποκλίσεις είναι μικρότερες, γεγονός που υποδηλώνει ότι τα χαρακτηριστικά που χρησιμοποιούνται για την πρόβλεψη του O_3 είναι πιο κατάλληλα ή πιο καλά συσχετισμένα. Όπως και με τους προηγούμενους ρύπους φαίνεται καλή απόκριση του μοντέλου και οι τιμές υποδεικνύουν καλύτερη προσαρμογή για το O_3 .

NO_2

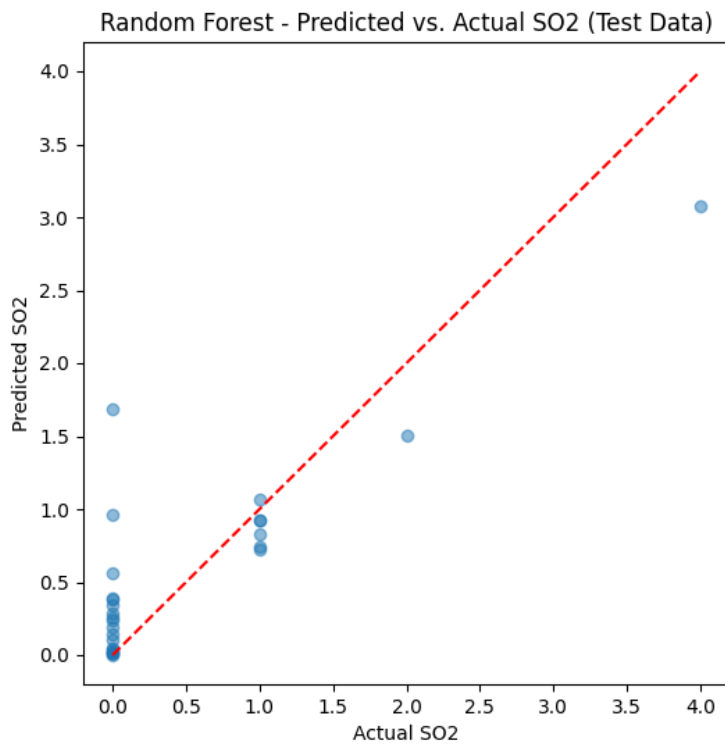


Γράφημα 4. Random Forest NO_2

MAE	2.2932
RMSE	3.0464
R ²	0.8032

Η επόμενη δοκιμή αφορά την απόδοση του μοντέλου για την πρόβλεψη των συγκεντρώσεων διοξειδίου του αζώτου (NO₂). Τα περισσότερα σημεία σχεδόν ευθυγραμμίζονται με την κόκκινη γραμμή, υποδεικνύοντας ότι παρέχει αρκετά αξιόπιστες προβλέψεις. Παρατηρούνται κάποιες αποκλίσεις, κυρίως στα άκρα των δεδομένων αλλά η συνολική διασπορά παραμένει μικρή. Οι μικρές διαφορές γύρω από τη γραμμή υποδεικνύουν ότι το μοντέλο διατηρεί καλή ακρίβεια. Στις χαμηλές τιμές (κάτω από 20), οι προβλέψεις είναι συγκριτικά πιο συνεπείς με τις πραγματικές τιμές ενώ στις υψηλότερες υπάρχουν λίγες περιπτώσεις όπου οι προβλέψεις αποκλίνουν από τις πραγματικές. Φαίνεται πως το Random Forest παρουσιάζει ικανοποιητική απόδοση στην πρόβλεψη συγκεντρώσεων NO₂ ενώ οι αποκλίσεις είναι περιορισμένες και αναμενόμενες.

SO₂

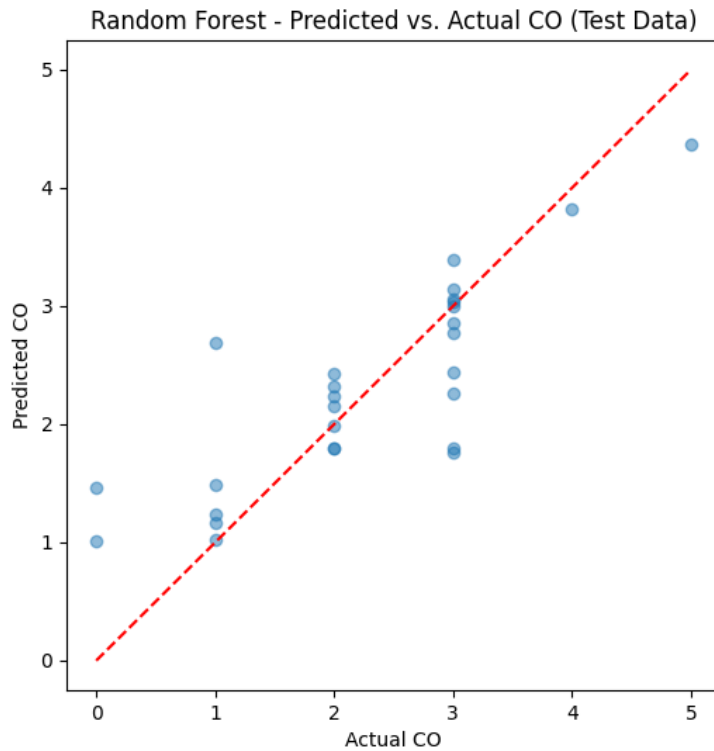


Γράφημα 5. Random Forest SO₂

MAE	0.2978
RMSE	2.7095
R ²	0.7068

Το παραπάνω γράφημα απεικονίζει την προβλεπόμενη συγκέντρωση SO₂ (διοξειδίου του θείου) έναντι της πραγματικής συγκέντρωσης στα δεδομένα ελέγχου. Σε αντίθεση με τις προηγούμενες δοκιμές παρατηρείται έντονη απόκλιση ειδικά στις χαμηλές τιμές και μικρή ταύτιση με τις πραγματικές. Η εκτίμηση του μοντέλου δείχνει ότι πιθανώς το μικρό εύρος τιμών επηρεάζει την πρόβλεψη, επίσης μπορεί να θεωρηθεί ότι η συγκέντρωση του SO₂ δεν σχετίζεται άμεσα με ιστορικά στοιχεία και καιρικές συνθήκες αλλά επηρεάζεται από άλλες παραμέτρους.

CO

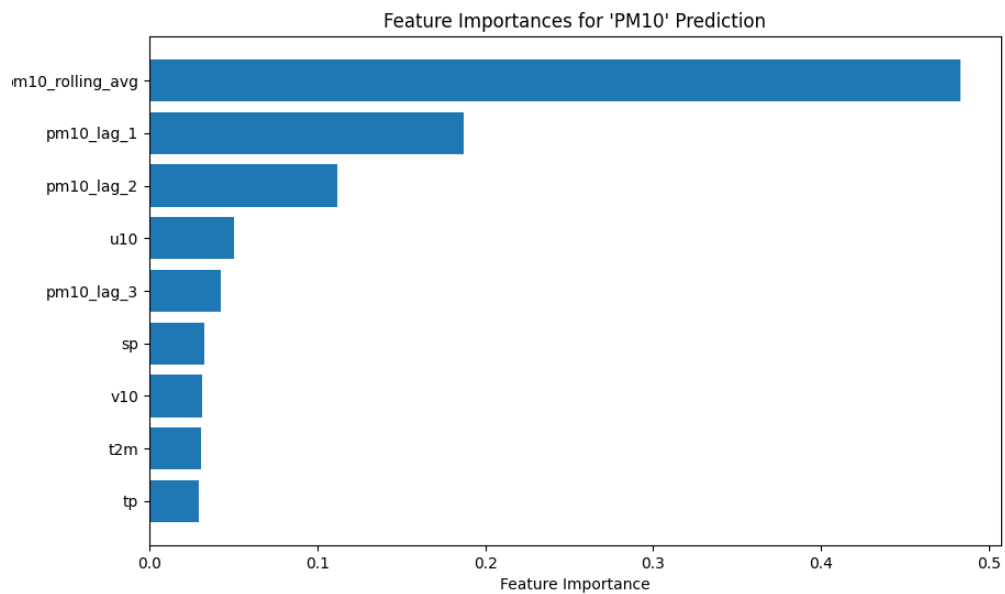


Γράφημα 6. Random Forest CO

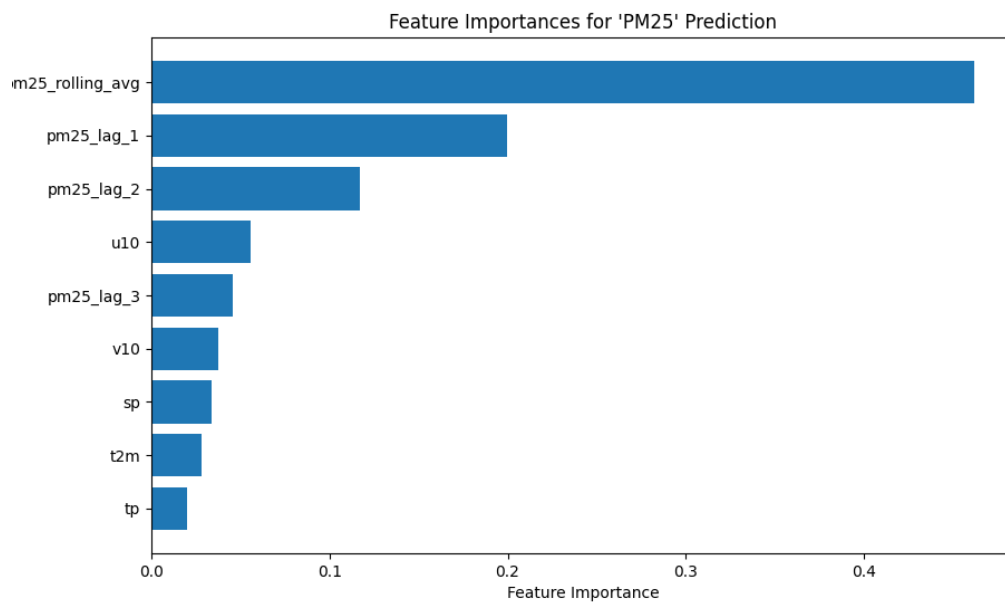
MAE	0.4523
RMSE	0.6485
R ²	0.6773

Για τη δοκιμή του μονοξειδίου του άνθρακα φαίνεται ότι σε κάποιες περιπτώσεις οι προβλέψεις βρίσκονται κοντά στη διαγώνιο, γεγονός που υποδεικνύει μέτρια απόδοση, αλλά υπάρχουν σημαντικές αποκλίσεις, που μπορεί να υποδεικνύουν δυσκολία του μοντέλου σε αξιόπιστη πρόβλεψη. Η γενική τάση δείχνει ότι το μοντέλο δεν προβλέπει ικανοποιητικά και οι αποκλίσεις δείχνουν περιθώρια βελτίωσης, πιθανόν μέσω περαιτέρω βελτίωσης των χαρακτηριστικών (features) ή της ρύθμισης υπερπαραμέτρων.

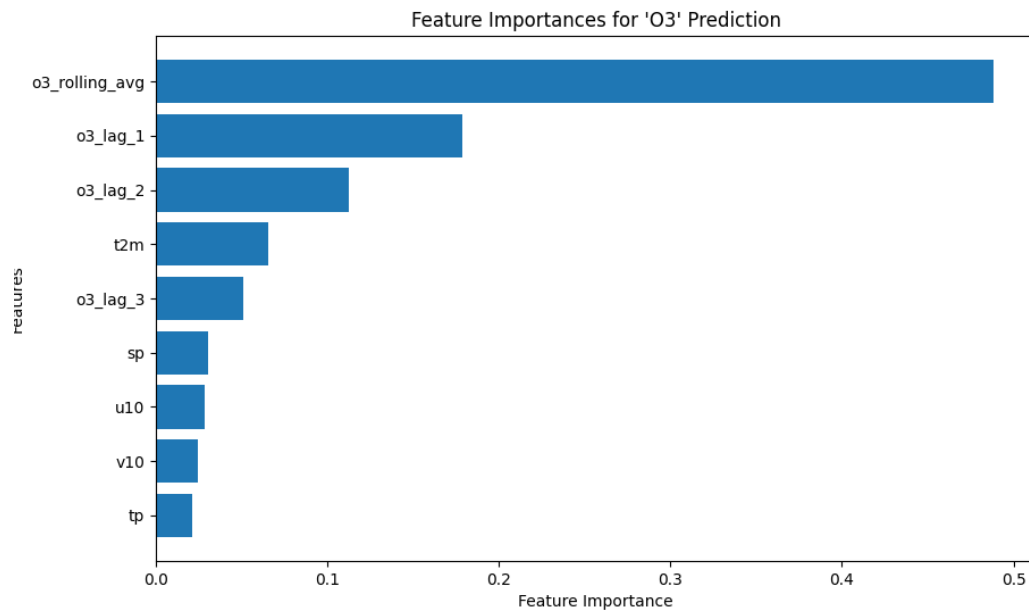
6.3 ΣΗΜΑΝΤΙΚΟΙ ΠΑΡΑΓΟΝΤΕΣ



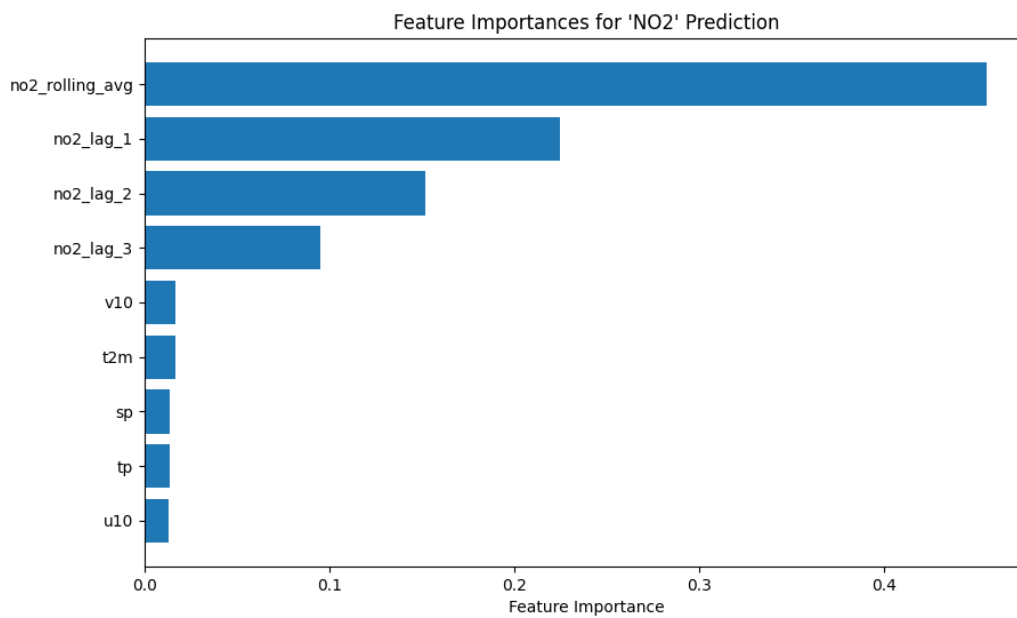
Γράφημα 7. Random Forest Feature Importance PM2.5



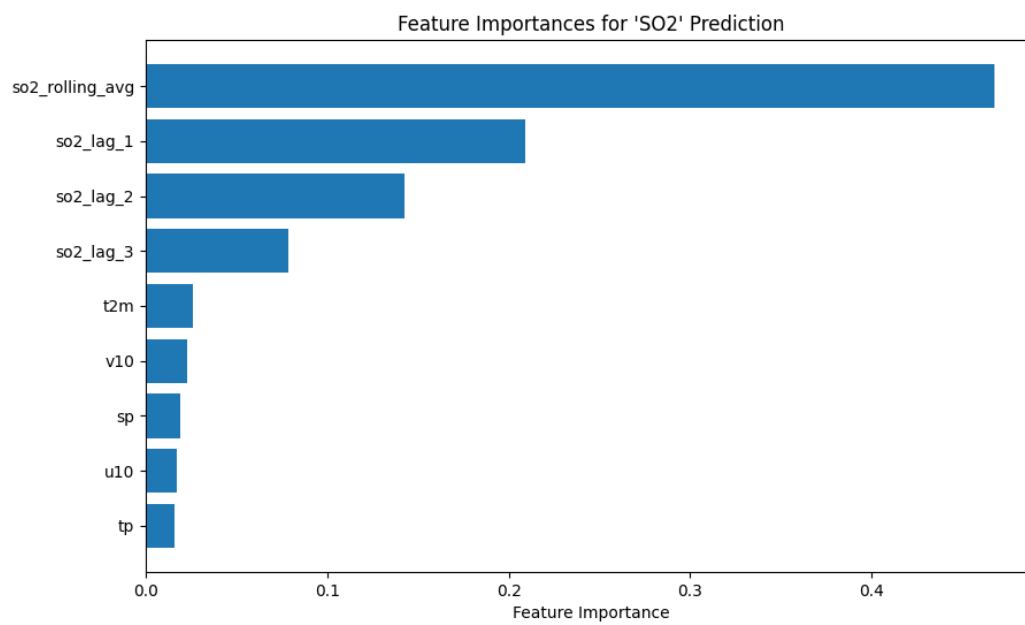
Γράφημα 8. Random Forest Feature Importance PM10



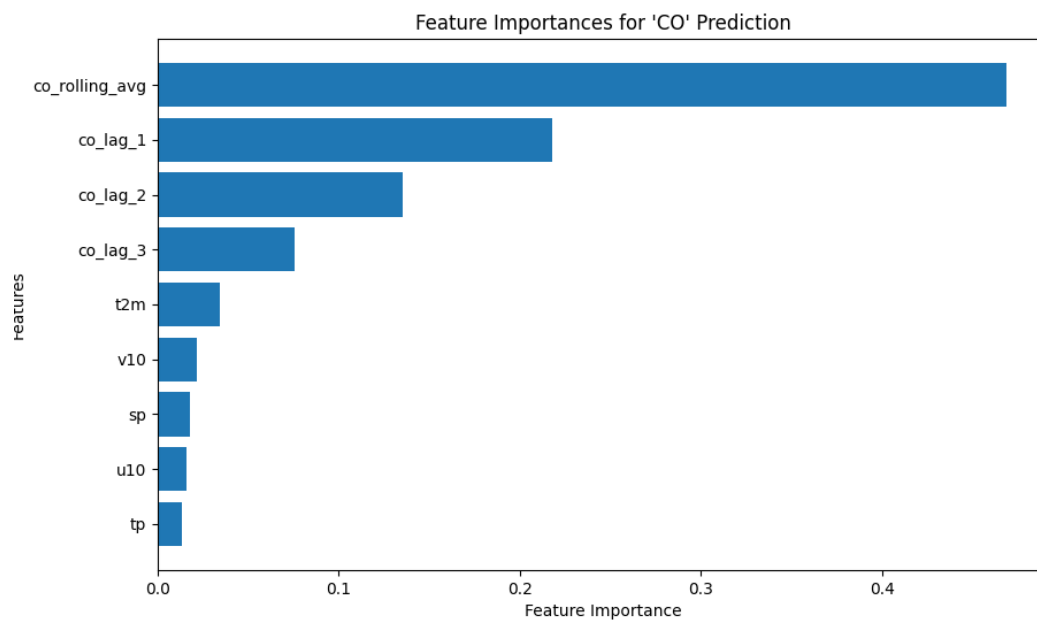
Γράφημα 9. Random Forest Feature Importance O_3



Γράφημα 10. Random Forest Feature Importance NO_2



Γράφημα 11. Random Forest Feature Importance SO_2



Γράφημα 12. Random Forest Feature Importance CO

Τα παραπάνω γραφήματα δείχνουν τη σημασία των χαρακτηριστικών (feature importance) για την πρόβλεψη των συγκεντρώσεων ρύπων στο μοντέλο Random Forest. Στον οριζόντιο άξονα φαίνεται η σημασία κάθε χαρακτηριστικού, ενώ στον κατακόρυφο εμφανίζονται οι μεταβλητές που χρησιμοποιήθηκαν.

Παρατηρούμε ότι η πιο σημαντική μεταβλητή είναι το `rolling_avg`, δηλαδή ο κινούμενος μέσος όρος των τιμών. Αυτό σημαίνει ότι οι προηγούμενες τιμές του ρύπου αποτελούν κρίσιμο παράγοντα στην πρόβλεψη των μελλοντικών συγκεντρώσεων. Ακολουθεί το `lag_1`, δηλαδή η τιμή του PM2.5 την προηγούμενη ημέρα, το οποίο επίσης έχει σημαντική συμβολή στο μοντέλο και στη συνέχεια τα `lag_2` και `lag_3`, οι τιμές δηλαδή που είχαν καταγραφεί δύο και τρεις ημέρες πριν αντίστοιχα, οι οποίες εξακολουθούν να έχουν επιρροή, αλλά μειωμένη σε σχέση με τις πιο πρόσφατες τιμές. Επιπλέον, παρατηρούμε ότι μεταβλητές που σχετίζονται με μετεωρολογικά δεδομένα, όπως `u10` (ταχύτητα ανέμου), `v10` (συνιστώσα ανέμου), `sp` (ατμοσφαιρική πίεση), `t2m` (θερμοκρασία 2 μέτρων) και `tp` (συνολική βροχόπτωση) έχουν μικρότερη αλλά όχι αμελητέα σημασία.

Πιο αναλυτικά, εξετάζοντας τα στοιχεία καιρού, για τα PM10 φαίνεται να υπάρχει επίδραση κυρίως της ταχύτητας του ανέμου και σε μικρότερο βαθμό αλλά με ίση συνεισφορά τα υπόλοιπα στοιχεία, ενώ στα PM2.5 επιδρούν περισσότερο η ταχύτητα και διεύθυνση ανέμου και στη συνέχεια υετός, ατμοσφαιρική πίεση και θερμοκρασία. Στη συγκέντρωση όζοντος μεγαλύτερη επιρροή δείχνει η θερμοκρασία και στη συνέχεια η ατμοσφαιρική πίεση, στο διοξείδιο του αζώτου η επίδραση είναι πολύ μικρή για όλα τα δεδομένα καιρού και τέλος στο διοξείδιο του θείου και το μονοξείδιο του άνθρακα διακρίνεται να υπερισχύει ελαφρώς η θερμοκρασία αλλά γενικά υπάρχει χαμηλή επίδραση όλων των παραγόντων.

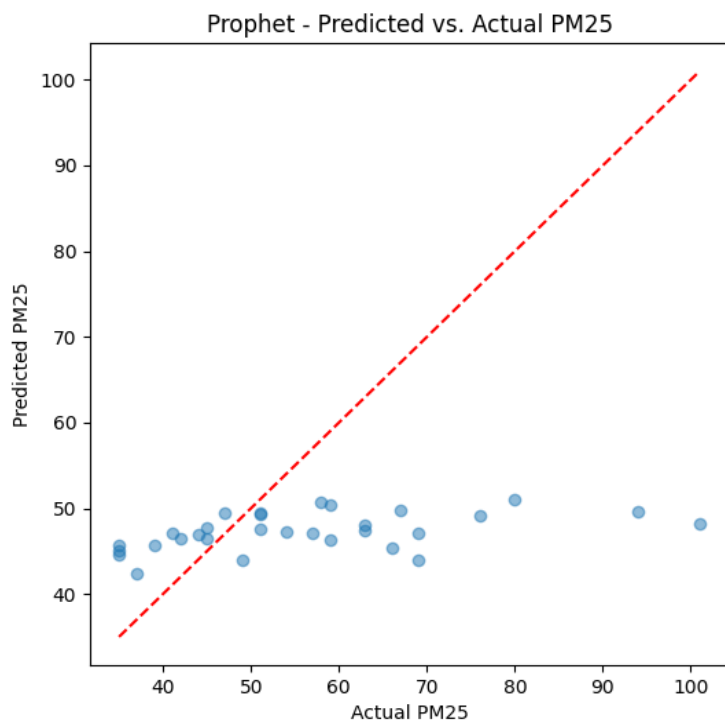
Αυτό σημαίνει ότι οι μετεωρολογικές συνθήκες επηρεάζουν τις συγκεντρώσεις αλλά σε μικρότερο βαθμό σε σχέση με το ιστορικό των μετρήσεων. Η υψηλή σημασία των μεταβλητών καθυστέρησης και του κινούμενου μέσου όρου υποδηλώνει ότι το μοντέλο χρησιμοποιεί παρελθοντικά δεδομένα για να ανιχνεύσει τάσεις, γεγονός που μπορεί να είναι ιδιαίτερα χρήσιμο για τη βελτίωση της ακρίβειας των προβλέψεων. Ανάλογα με την επιθυμητή ακρίβεια, θα μπορούσε να εξεταστεί η ενσωμάτωση περισσότερων μετεωρολογικών ή άλλων παραμέτρων καθώς και η βελτιστοποίηση των υπαρχόντων. Συμπερασματικά, τα γραφήματα δείχνουν ότι το

μοντέλο βασίζεται κυρίως σε ιστορικά δεδομένα για να κάνει προβλέψεις ενώ τα μετεωρολογικά δεδομένα έχουν δευτερεύοντα αλλά όχι ασήμαντο ρόλο.

6.4 PROPHET

To Prophet

PM2.5

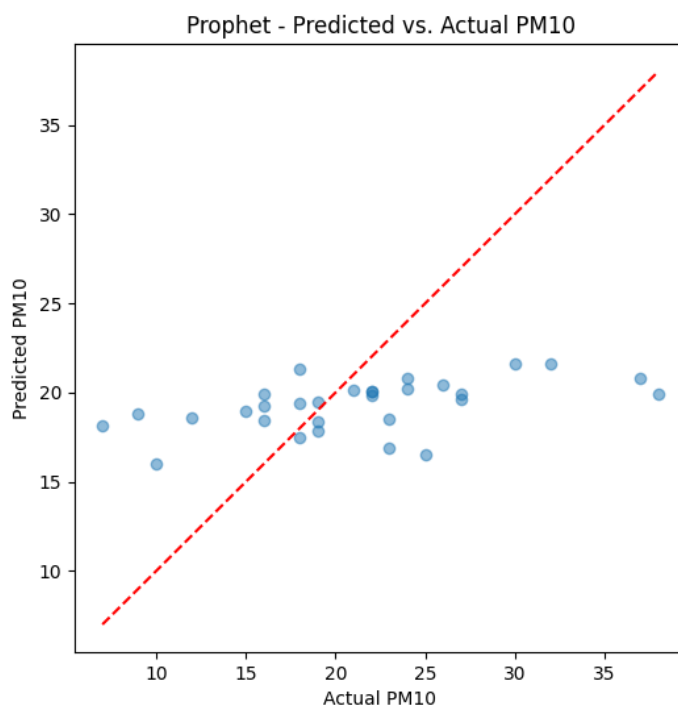


Γράφημα 13. Prophet PM2.5

MAE	12.9272
RMSE	17.8415
R ²	-0.1769

Σύμφωνα με το γράφημα και τις μετρικές αξιολόγησης ο αλγόριθμος Prophet δε δίνει αποτελέσματα που να ταυτίζονται με τις πραγματικές τιμές για τις συγκεντρώσεις των PM2.5 παρά μόνο σε ελάχιστα σημεία. Η απόκλιση είναι πολύ μεγάλη και τα σημεία πρόγνωσης βρίσκονται όλα μεταξύ των τιμών 40-50. Ενδεχομένως, η απόδοση αυτή να βελτιωθεί αν ληφθούν υπόψη περισσότεροι παράγοντες ή αν ρυθμιστεί καλύτερα το μοντέλο.

PM10



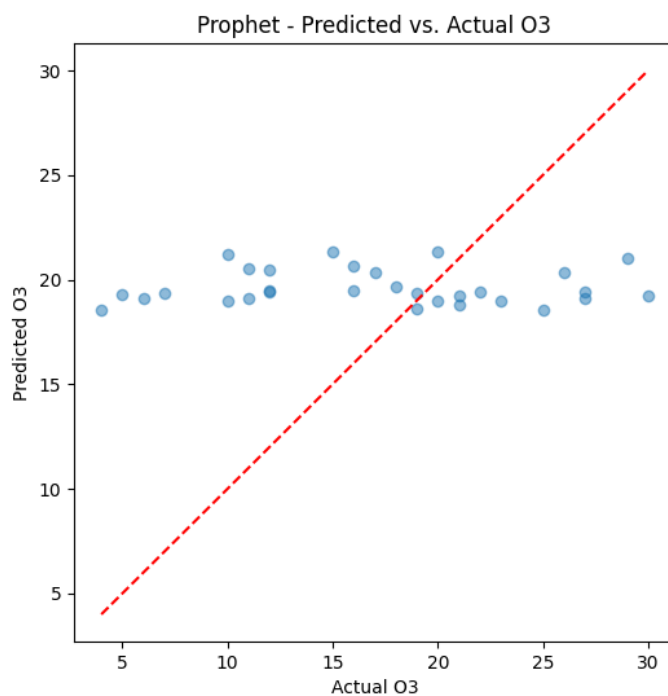
Γράφημα 14. Prophet PM10

MAE	5.3485
RMSE	6.9014
R ²	0.0957

Στο παραπάνω γράφημα παρουσιάζεται η σύγκριση των προβλέψεων του μοντέλου Prophet με τις πραγματικές τιμές για τα PM10. Τα περισσότερα σημεία συγκεντρώνονται γύρω

από χαμηλότερες τιμές μεταξύ 15 και 25. Ωστόσο, υπάρχουν αρκετές αποκλίσεις από τη γραμμή, ιδιαίτερα για υψηλότερες τιμές του. Επίσης υπάρχει μια συγκέντρωση σημείων κοντά στην περιοχή μεταξύ 15-20 μονάδων, γεγονός που δείχνει ότι το μοντέλο έχει μικρή διακύμανση στις προβλέψεις του και δε θεωρείται αξιόπιστο.

O₃



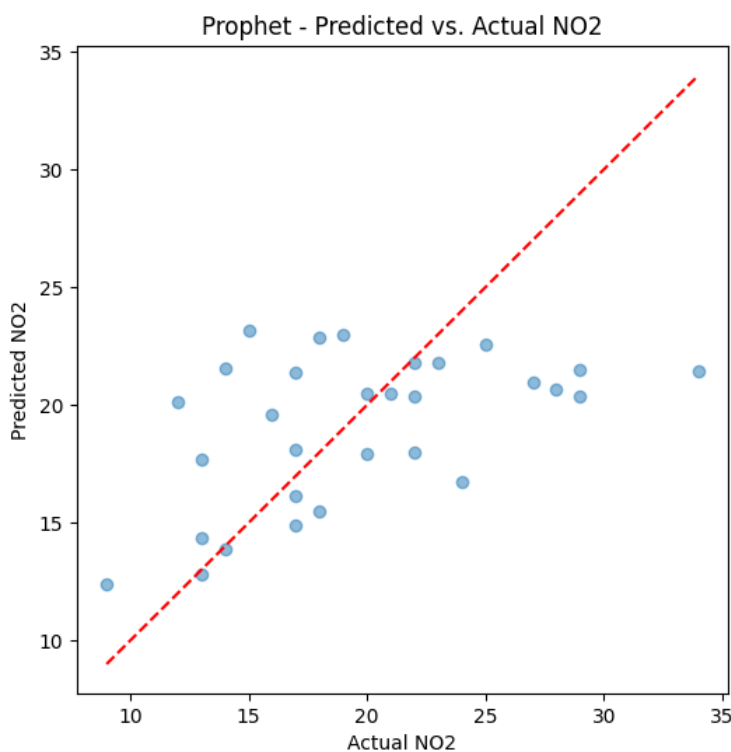
Γράφημα 15. Prophet O₃

MAE	6.5029
RMSE	7.7060
R ²	-0.1398

Όπως και με τα αιωρούμενα σωματίδια το μοντέλο παρουσιάζει και εδώ μεγάλες αποκλίσεις και πρόγνωση που ταυτίζεται μόνο με ορισμένες τιμές κοντά στο 20. Οι μετρικές

επίσης δείχνουν μη αξιόπιστη πρόγνωση, επομένως δεν μπορεί να θεωρηθεί ότι ανταποκρίνεται σωστά στα δεδομένα που του δίνονται.

NO₂



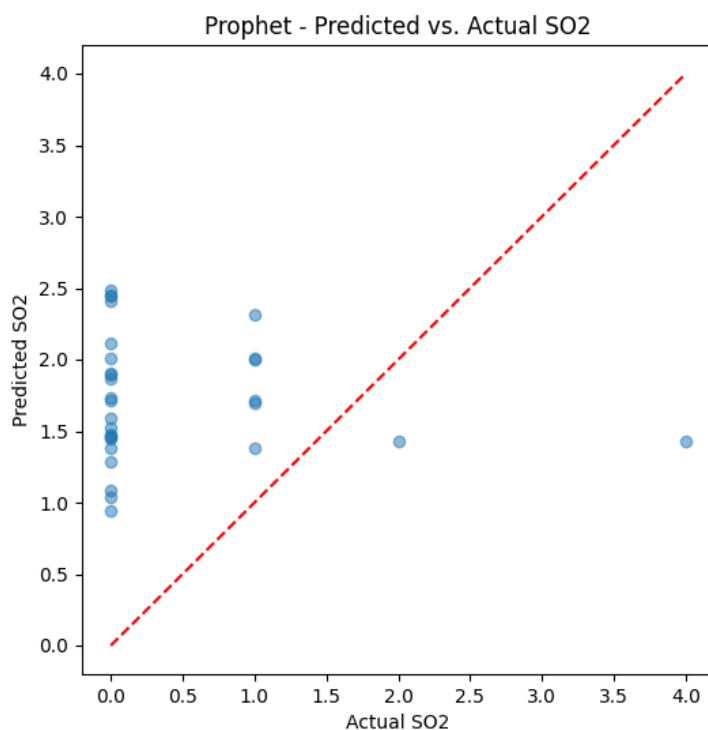
Γράφημα 16. Prophet NO₂

MAE	3.9164
RMSE	5.0723
R ²	0.2427

Το γράφημα δείχνει την πρόβλεψη έναντι των πραγματικών τιμών για το NO₂, χρησιμοποιώντας το μοντέλο Prophet. Τα σημεία είναι αρκετά διάσπαρτα γύρω από τη γραμμή και οι μεγάλες αποκλίσεις παρατηρούνται κυρίως για τιμές πάνω από 25. Αυτό υποδηλώνει ότι το Prophet δεν καταφέρνει να προβλέψει με μεγάλη ακρίβεια τις τιμές του NO₂ και μπορεί να

χαρακτηριστεί οριακά αποδεκτό. Το Prophet έχει περιορισμένη ακρίβεια για την πρόβλεψη του NO_2 , και η απόδοσή του μπορεί να επηρεάζεται από παραμέτρους που δεν έχουν ληφθεί επαρκώς υπόψη. Η δοκιμή για το NO_2 είναι η μοναδική στην οποία το μοντέλο Prophet δείχνει μια μέτρια απόδοση.

SO_2



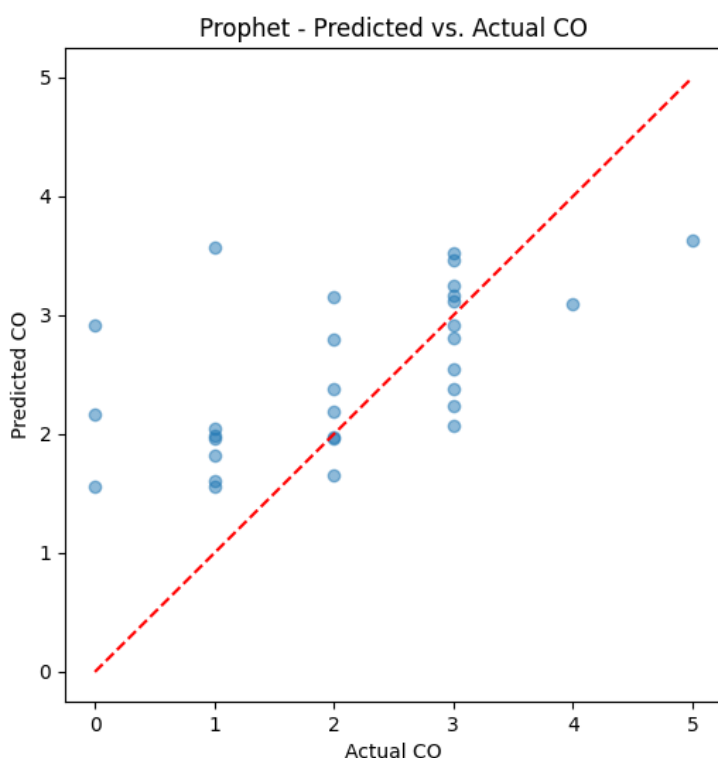
Γράφημα 17. Prophet SO_2

MAE	1.5334
RMSE	1.6442
R^2	-2.8257

Στο παραπάνω γράφημα απεικονίζεται η σχέση μεταξύ των προβλεπόμενων και των πραγματικών τιμών του SO_2 χρησιμοποιώντας το μοντέλο Prophet. Υπάρχουν σοβαρές

αποκλίσεις μεταξύ των πραγματικών τιμών και των προβλέψεων, με τις περισσότερες τιμές να μην ευθυγραμμίζονται με τη γραμμή αναφοράς. Το μοντέλο δεν φαίνεται να έχει ακρίβεια, ειδικά για υψηλότερες τιμές και δυσκολεύεται να αποδώσει ακριβείς προβλέψεις για το SO₂.

CO



Γράφημα 18. Prophet CO

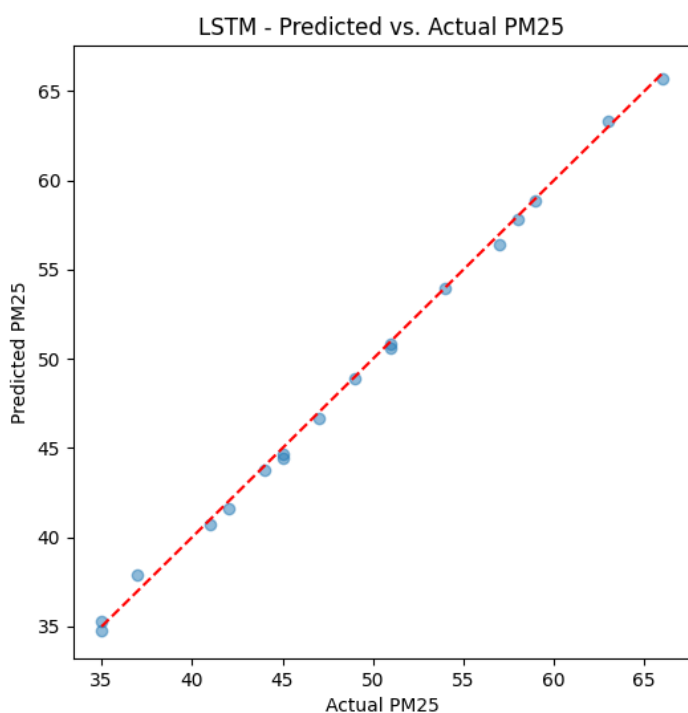
MAE	0.7981
RMSE	1.0674
R ²	0.1995

Σε αυτό το γράφημα, παρουσιάζεται η σύγκριση των προβλέψεων του μοντέλου Prophet με τις πραγματικές τιμές για το CO. Φαίνεται πως τα σημεία παρουσιάζουν αρκετά μεγάλη διασπορά, κάποια είναι κοντά στη γραμμή αναφοράς ενώ αρκετά άλλα βρίσκονται μακριά,

υποδηλώνοντας αποκλίσεις. Παρατηρείται έντονη συγκέντρωση σημείων γύρω από τις πραγματικές τιμές 2-3, υποδεικνύοντας ότι το Prophet τείνει να προβλέπει παρόμοιες τιμές σε αυτό το εύρος. Επομένως η απόδοση του κρίνεται μη ικανοποιητική.

6.5 RNN-LSTM

PM2.5

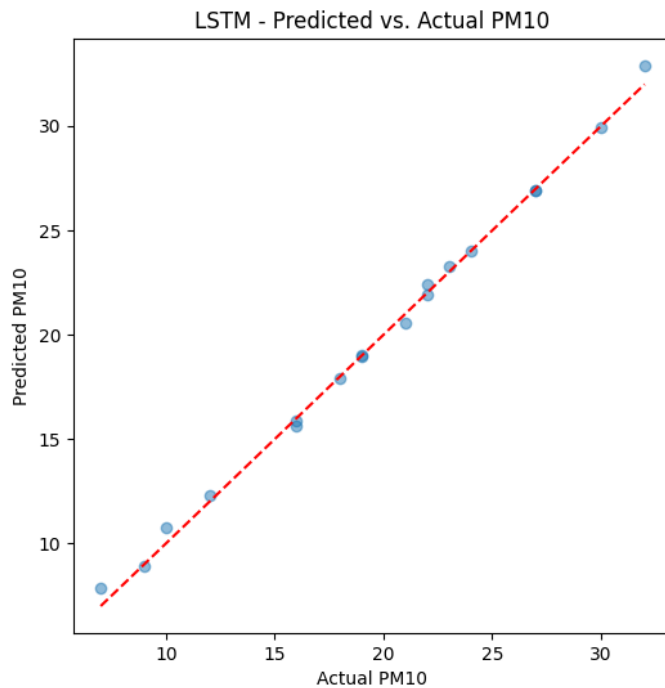


Γράφημα 19. RNN-LSTM PM2.5

MAE	0.3210
RMSE	0.3747
R ²	0.9983

Το συγκεκριμένο διάγραμμα παρουσιάζει τις προβλεπόμενες τιμές PM2.5 από το μοντέλο LSTM σε σχέση με τις πραγματικές τιμές. Παρατηρείται ευθυγράμμιση με τη διαγώνιο, οι περισσότερες μπλε κουκκίδες βρίσκονται πολύ κοντά στη διακεκομμένη κόκκινη γραμμή ($y = x$), γεγονός που δείχνει υψηλή ακρίβεια του μοντέλου. Η εξαιρετικά στενή κατανομή των σημείων υποδεικνύει ότι είναι ιδιαίτερα αποδοτικό στην πρόβλεψη των τιμών PM2.5 και οι ελάχιστες αποκλίσεις φανερώνουν ότι το μοντέλο είναι σχεδόν τέλειο για αυτό το σύνολο δεδομένων. Οι μετρικές αξιολόγησης επαληθεύουν την αξιοπιστία του αλγορίθμου.

PM10

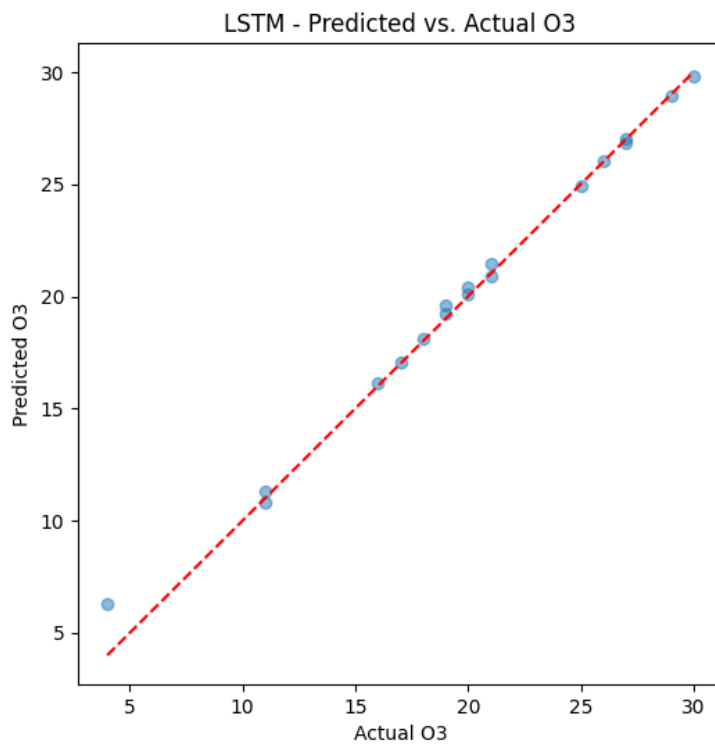


Γράφημα 20. RNN-LSTM PM10

MAE	0.2800
RMSE	0.3968
R ²	0.9967

Στο παραπάνω διάγραμμα παρουσιάζονται οι προβλεπόμενες τιμές PM10 από το μοντέλο LSTM. Όπως και προηγουμένως φαίνεται ευθυγράμμιση με τη διαγώνιο, και επιβεβαιώνεται η αξιόπιστη πρόγνωση και από τις μετρικές αξιολόγησης. Η ακρίβεια του μοντέλου φαίνεται και από την εξαιρετικά στενή κατανομή των σημείων. Δεν παρατηρούνται σημαντικές αποκλίσεις.

O₃

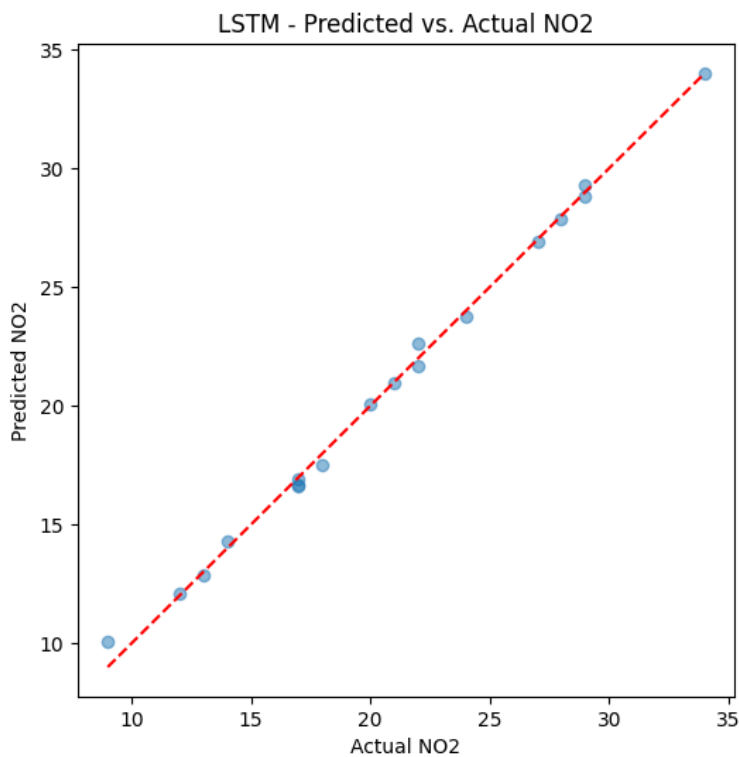


Γράφημα 21. RNN-LSTM O₃

MAE	0.3147
RMSE	0.5866
R ²	0.9922

Στο γράφημα αυτό παρατηρείται επίσης υψηλή ακρίβεια πρόγνωση για το O_3 . Το μοντέλο LSTM έχει πολύ καλή απόκριση στα δεδομένα και οι τιμές πρόβλεψης αγγίζουν σχεδόν τις πραγματικές. Παρατηρείται απόκλιση μόνο στις πολύ μικρές συγκεντρώσεις. Έτσι μπορεί να θεωρηθεί ως αξιόπιστο, κάτι που επαληθεύεται από τα μετρικές αξιολόγησης που έχουν σχεδόν ιδανικές τιμές.

NO_2

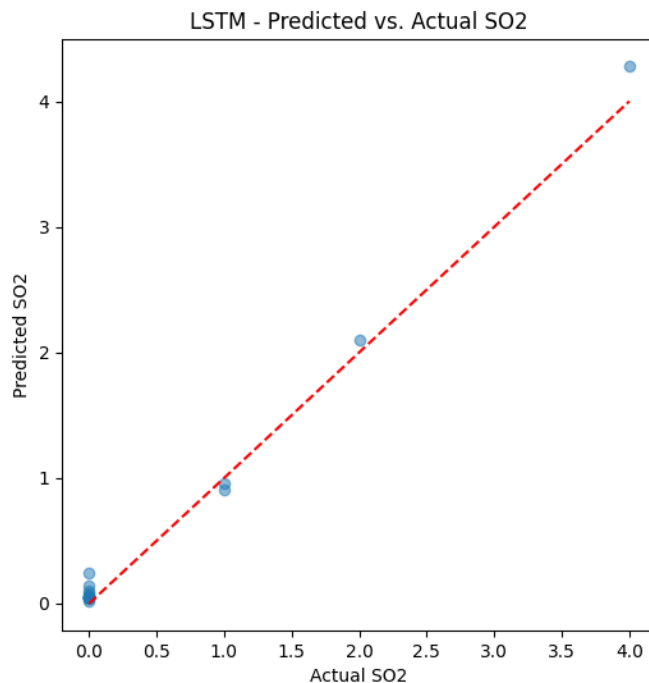


Γράφημα 22. RNN-LSTM NO_2

MAE	0.2687
RMSE	0.3704
R^2	0.9969

Το γράφημα παρουσιάζει τη σύγκριση των προβλέψεων του μοντέλου LSTM με τις πραγματικές τιμές για το NO₂. Και εδώ ο αλγόριθμος δείχνει πολύ μεγάλη ακρίβεια με τις τιμές πρόβλεψης να ταυτίζονται με τις πραγματικές και οι ελάχιστες αποκλίσεις είναι αμελητέας σημασίας. Σε συνδυασμό με τις μετρικές που έχουν τιμές κοντά στο ιδανικό για την περίπτωση μπορεί να θεωρηθεί αρκετά αποτελεσματικός και αξιόπιστος.

SO₂



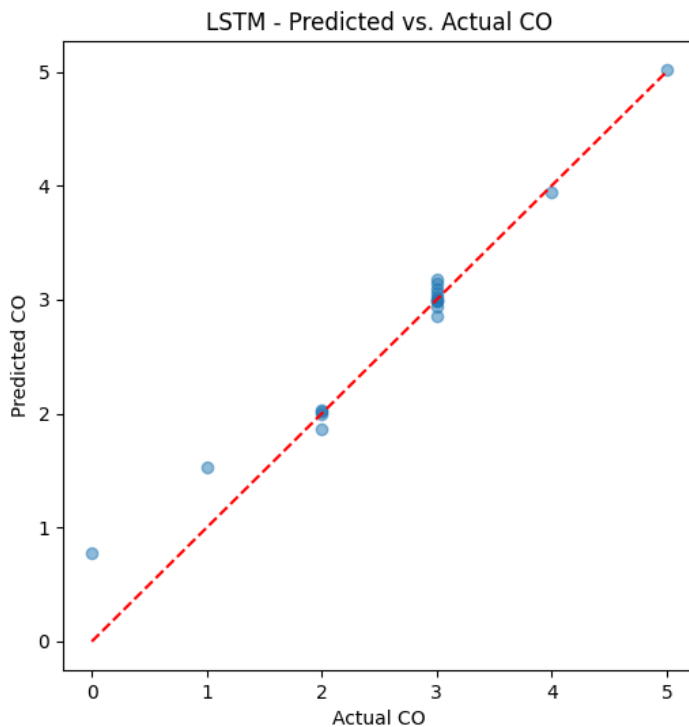
Γράφημα 23. RNN-LSTM SO₂

MAE	0.0851
RMSE	0.1092
R ²	0.9884

Σε αυτό το γράφημα φαίνεται η πρόγνωση για το SO₂ από το μοντέλο LSTM. Οι περισσότερες τιμές είναι κοντά στη διαγώνιο με κάποιες μικρές αποκλίσεις στις τιμές κοντά στο

0 και πάνω από 4. Μπορεί να θεωρηθεί αρκετά αξιόπιστο και οι μετρικές αξιολόγησης το επιβεβαιώνουν. Σε σχέση με τα προηγούμενα μοντέλα, για το διοξείδιο του θείου έχει αρκετά μεγαλύτερη ακρίβεια αλλά όχι απόλυτη.

CO



Γράφημα 24. RNN-LSTM CO

MAE	0.1278
RMSE	0.2354
R ²	0.9501

Το παραπάνω γράφημα δείχνει τη σύγκριση των προβλέψεων του μοντέλου LSTM με τις πραγματικές τιμές για το CO. Τα περισσότερα σημεία συγκεντρώνονται κοντά στη διαγώνια

γραμμή αναφοράς και το μοντέλο φαίνεται να αποδίδει σχετικά καλά με μερικές αποκλίσεις, ιδιαίτερα για τιμές κοντά στο μέσο εύρος. Γενικά, το LSTM παρουσιάζει ικανοποιητική ακρίβεια για την πρόβλεψη του CO όχι όμως σε τέλειο επίπεδο.

Λαμβάνοντας υπόψιν την αποτελεσματικότητα του και την υψηλή απόδοση σε ορισμένους ρύπους η μη ακριβής πρόγνωση στο διοξείδιο του θείου και το μονοξείδιο του άνθρακα μπορεί να οφείλεται στην εξάρτηση των συγκεκριμένων ρύπων από άλλους παράγοντες που δεν περιλαμβάνονται στη συγκεκριμένη ανάλυση,

6.6 ΑΛΓΟΡΙΘΜΟΣ ALL

Ο αλγόριθμος αυτός είναι επέκταση του μοντέλου LSTM και δοκιμάζει πρόβλεψη για όλους τους εξεταζόμενους ρύπους ταυτόχρονα, δίνοντας ξεχωριστά διαγράμματα. Όπως φαίνεται από τα γραφήματα παρόλο που χρησιμοποιεί τη λογική του RNN μοντέλου σημειώνονται μικρές διαφορές στην επίδοση του. Παρατηρείται λοιπόν μικρότερη αξιοπιστία σε σχέση με την εφαρμογή του μοντέλου χωριστά σε κάθε ρύπο όχι όμως σε βαθμό που να είναι ακατάλληλος.

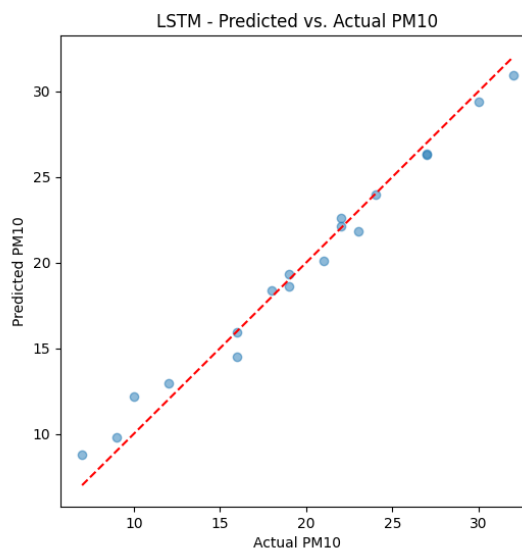
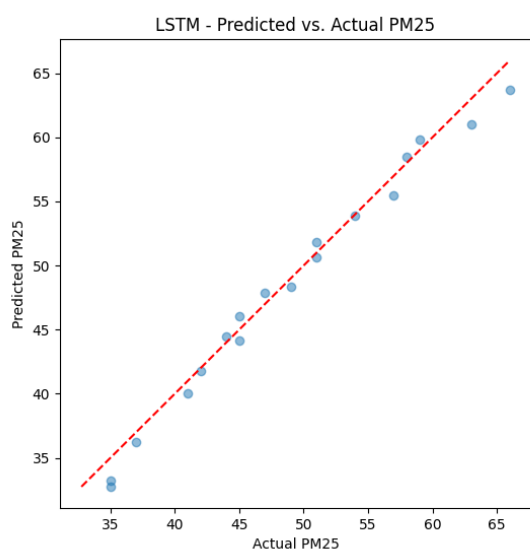
Ο πίνακας 5 δείχνει τις μετρικές για κάθε ρύπο όπου παρατηρούνται διαφορές κυρίως στη δοκιμή πρόβλεψης του διοξειδίου του θείου και του μονοξειδίου του άνθρακα. Ενδεχομένως μπορούν να εφαρμοστούν πρακτικές βελτιστοποίησης για καλύτερη απόδοση.

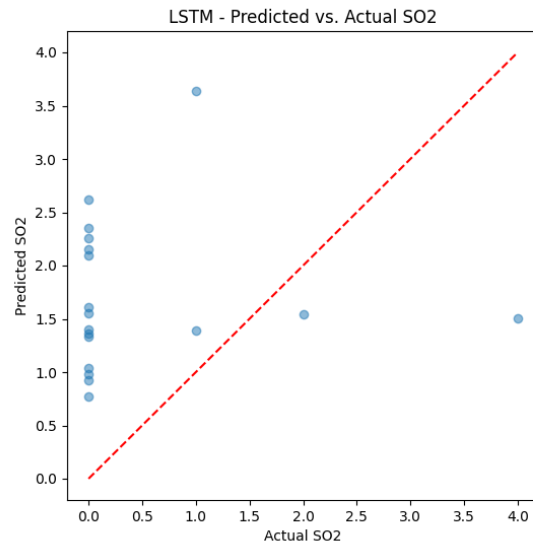
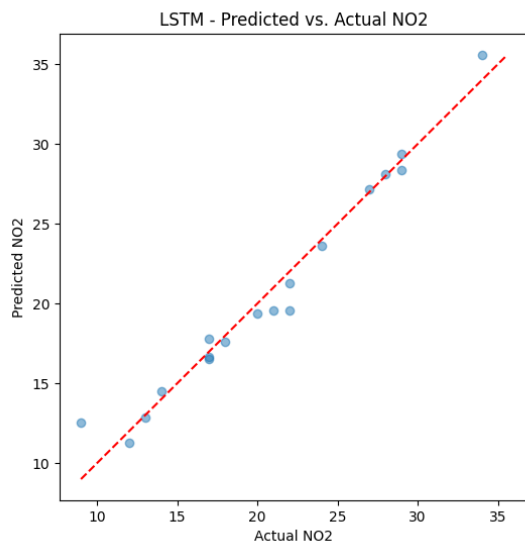
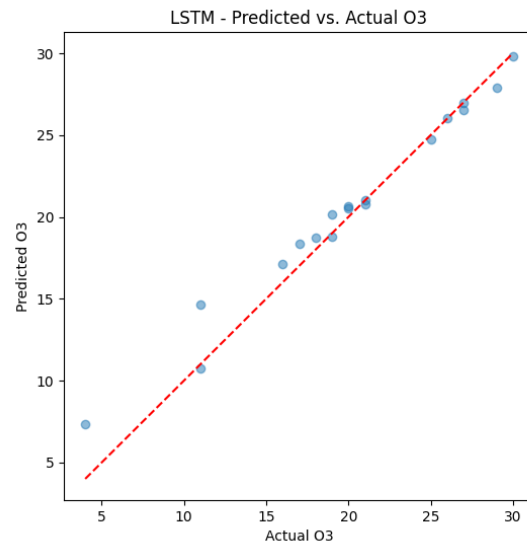
Ρύπος	MAE	RMSE	R2
Pm2.5	1.0131	1.2101	0.9821
Pm10	0.7924	0.9783	0.9801
O3	0.8514	1.3327	0.9596
NO2	0.8601	1.2229	0.9658

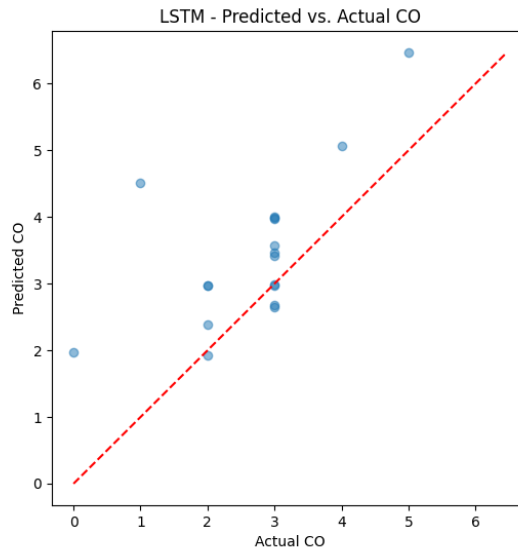
SO2	1.5805	1.7351	-1.9379
CO	0.8672	1.1914	-0.2274

Πίνακας 5. Μετρικές ρύπων στον αλγόριθμο All.

Ακολουθούν τα γραφήματα όπως προέκυψαν από την εκτέλεση του αλγορίθμου που συγκρίνοντας τα με τα αποτελέσματα από τη διαχωρισμένη δοκιμή φαίνονται οι αποκλίσεις πιο έντονα κυρίως στους δυο τελευταίους ρύπους.







Όπως αναφέρθηκε παραπάνω, στα γραφήματα 1-6 φαίνεται η απόκριση του αλγορίθμου All-LSTM για την πρόγνωση επιπέδων ρύπων. Το συγκεντρωτικό μοντέλο λοιπόν δείχνει ότι πλησιάζει αρκετά στις πραγματικές τιμές για τα PM10, PM2.5, O₃ και NO₂ αλλά φαίνονται σημαντικές αποκλίσεις για το SO₂ και το CO. Σε σύγκριση με τις ξεχωριστές δοκιμές ανά ρύπο με το LSTM υστερεί και πιθανόν αυτό συμβαίνει επειδή η εκπαίδευση γίνεται μία φορά για όλους του παράγοντες ρύπων που μελετώνται. Παρόλα αυτά παρουσιάζει μεγαλύτερη ακρίβεια σε σχέση με το Random Forest και το Prophet. Επίσης είναι ταχύτερος και μπορεί να εφαρμοστεί για μία γρήγορη πρόβλεψη ή σε περιπτώσεις που δεν απαιτείται ιδιαίτερα υψηλή ακρίβεια.

ΚΕΦΑΛΑΙΟ 7. ΣΥΜΠΕΡΑΣΜΑΤΑ-ΠΡΟΤΑΣΕΙΣ ΒΕΛΤΙΩΣΗΣ

Κατά τη διάρκεια της εκπαίδευσης και των δοκιμών του αλγορίθμου παρατηρείται ότι με τη χρήση του Random forest τα αποτελέσματα πρόβλεψης είναι ικανοποιητικά για τα $\text{pm}_{2.5}$, pm_{10} , O_3 , NO_2 αλλά όχι για τα SO_2 , CO . Η ακρίβεια παρόλο που δεν είναι τέλεια βρίσκεται σε αποδεκτά όρια και η ταχύτητα του αλγορίθμου μπορεί να θεωρηθεί ικανοποιητική. Το Prophet από την άλλη δεν ανταποκρίνεται επαρκώς και μόνο στη συγκέντρωση του NO_2 πλησιάζει κοντά στις πραγματικές τιμές έχοντας όμως αρκετές αποκλίσεις. Τα αποτελέσματα που δίνει δεν μπορούν να θεωρηθούν αξιόπιστα. Τέλος το RNN-LSTM νευρωνικό δίκτυο δείχνει μεγαλύτερη ικανότητα πρόβλεψης αγγίζοντας σχεδόν ακριβώς τις πραγματικές τιμές για τα $\text{pm}_{2.5}$, pm_{10} , O_3 , NO_2 και με μικρή απόκλιση τους άλλους δύο ρύπους. Το μειονέκτημα είναι ο μεγαλύτερος χρόνος που απαιτεί για την εκπαίδευση του. Επίσης το συγκεντρωτικό του μοντέλο All-LSTM δίνει μια πιο γρήγορη αλλά μικρότερης ακρίβειας πρόβλεψη.

Εξετάζοντας προσεκτικά τα αποτελέσματα των γραφημάτων μπορούμε να συμπεράνουμε ότι οι ρύποι για τους οποίους παρατηρείται άμεση συσχέτιση με τα δεδομένα του καιρού είναι τα μικροσωματίδια, το όζον και το NO_2 . Όπως αναφέρθηκε και στο θεωρητικό μέρος η συγκέντρωση μικροσωματιδίων εξαρτάται από τον άνεμο και την υγρασία, ενώ του όζοντος και του διοξειδίου του αζώτου από τη θερμοκρασία. Το διοξείδιο του θείου επίσης σχετίζεται με την αυξημένη θερμοκρασία με την προϋπόθεση όμως να βρίσκονται ήδη στον αέρα θειούχες ενώσεις. Γενικά όμως ισχύει ότι η συγκέντρωση ρύπων εξαρτάται από πολλούς παράγοντες και για άριστη πρόβλεψη θα πρέπει να λαμβάνονται υπόψιν κι άλλα δεδομένα.

Οι βελτιώσεις που μπορούν να εφαρμοστούν στην παρούσα μεθοδολογία περιλαμβάνουν την υιοθέτηση πιο προηγμένων αλγορίθμων μηχανικής μάθησης, την αξιοποίηση τεχνικών βαθιάς μάθησης (deep learning), καθώς και την περαιτέρω βελτιστοποίηση των υπέρ-παραμέτρων.

Μία πιθανή παρέμβαση είναι η βελτιστοποίηση των υπέρ-παραμέτρων με πιο προηγμένες τεχνικές. Ο παραδοσιακός Grid Search, αν και εξαντλητικός, μπορεί να αντικατασταθεί ή να συνδυαστεί με τον Random Search, που εξετάζει τυχαίους συνδυασμούς υπέρ-παραμέτρων, μειώνοντας τον υπολογιστικό χρόνο. Επίσης ενδιαφέρουσα είναι η χρήση της Bayesian Optimization, η οποία βασίζεται στη δημιουργία ενός πιθανοθεωρητικού μοντέλου για να

εντοπίζει περιοχές υπέρ-παραμέτρων με υψηλό δυναμικό. Αυτή η προσέγγιση επικεντρώνεται σε υποσχόμενους συνδυασμούς, μειώνοντας τον αριθμό των απαιτούμενων αξιολογήσεων και επιταχύνοντας τη διαδικασία. Γενικά, η υιοθέτηση αυτών των τεχνικών και εργαλείων μπορεί να βελτιώσει σημαντικά την ακρίβεια και την αποδοτικότητα των προβλέψεων, καθιστώντας το λογισμικό πιο ισχυρό και προσαρμόσιμο σε διαφορετικές περιβαλλοντικές συνθήκες και γεωγραφικές περιοχές.

Επίσης η λειτουργικότητα του λογισμικού μπορεί να επεκταθεί με την ενσωμάτωση νέων χαρακτηριστικών και εργαλείων που θα ενισχύσουν τη χρηστικότητα και την εφαρμοσιμότητά του. Η προσθήκη κοινωνικοοικονομικών δεδομένων, όπως η κυκλοφοριακή κίνηση, οι βιομηχανικές δραστηριότητες και τα επίπεδα κατανάλωσης ενέργειας, μπορεί να εμπλουτίσει το μοντέλο, επιτρέποντάς του να προβλέπει με μεγαλύτερη ακρίβεια τη ρύπανση. Επιπλέον, η ανάπτυξη μιας διαδικτυακής διεπαφής (web-based πλατφόρμα) θα παρέχει στους χρήστες τη δυνατότητα να ανεβάζουν δεδομένα, να λαμβάνουν προβλέψεις σε πραγματικό χρόνο και να παρακολουθούν τις τάσεις της ρύπανσης. Ακόμη χρήσιμη προσθήκη θα ήταν η δημιουργία ενός συστήματος αυτοματοποιημένων ειδοποιήσεων, το οποίο θα ενημερώνει τους χρήστες για επικίνδυνα επίπεδα ρύπανσης ή προβλεπόμενα επεισόδια υψηλής συγκέντρωσης ρύπων. Τέτοιες επεκτάσεις θα ενισχύσουν την πρακτική αξία του λογισμικού και θα το καταστήσουν ένα χρήσιμο εργαλείο για περιβαλλοντική διαχείριση και λήψη αποφάσεων.

Το παραπάνω σύστημα ειδοποιήσεων μπορεί να υλοποιηθεί με διάφορους τρόπους, ανάλογα με τις απαιτήσεις και τη φύση των χρηστών. Αρκετά εύχρηστη για το ευρύ κοινό είναι η χρήση μηνυμάτων κινητής τηλεφωνίας ή ηλεκτρονικού ταχυδρομείου, όπου οι χρήστες μπορούν να λαμβάνουν ειδοποιήσεις όταν οι προβλεπόμενες συγκεντρώσεις ρύπων υπερβαίνουν προκαθορισμένα όρια, ανάλογες με τις προειδοποιήσεις καιρικών φαινομένων. Εναλλακτικά ένα σύστημα αναδυόμενων ειδοποιήσεων σε μία δικτυακή εφαρμογή θα μπορούσε να ειδοποιεί άμεσα τους χρήστες μέσω περιηγητών ή εφαρμογών. Επίσης μπορεί να ενσωματωθεί ένας πίνακας ελέγχου σε πραγματικό χρόνο, το οποίο θα εμφανίζει δυναμικά τις προβλέψεις και θα ενεργοποιεί οπτικές ειδοποιήσεις (για παράδειγμα κόκκινες ενδείξεις για υψηλά επίπεδα ρύπανσης).

Για πιο προηγμένες υλοποιήσεις η ενσωμάτωση τεχνητής νοημοσύνης θα μπορούσε να προσφέρει προσωποποιημένες ειδοποιήσεις, λαμβάνοντας υπόψη τις προτιμήσεις και τη γεωγραφική τοποθεσία του χρήστη. Η υποδομή μπορεί να επεκταθεί με τη χρήση υπηρεσιών

νέφους, όπως το AWS ή το Google Cloud Platform, για κλιμάκωση και αξιοπιστία. Η αποθήκευση δεδομένων, η φιλοξενία του μοντέλου και η διαχείριση των ειδοποιήσεων μπορούν να υλοποιηθούν σε αυτές τις πλατφόρμες για μεγαλύτερη ευελιξία. Τέλος για τη διασφάλιση της ακρίβειας και της αποδοτικότητας του συστήματος, απαιτείται η συνεχής παρακολούθηση και καταγραφή της απόδοσης των ειδοποιήσεων, μέσω εργαλείων όπως το Grafana ή το Prometheus, που επιτρέπουν τη δημιουργία πίνακα για παρακολούθηση σε πραγματικό χρόνο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Α. ΞΕΝΟΓΛΩΣΣΗ

A. Gururaj, V. Nagaraj, R. A S, A. K N, S. M U and A. D. B, "Air Quality Prediction and Analysis using Machine Learning," *2024 5th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, Lalitpur, Nepal, 2024, pp. 543-550, doi: 10.1109/ICMCSI61536.2024.00086.

Akanksha, N. Maurya, M. Jain and S. Arya, "Prediction And Analysis of Air Pollution Using Machine Learning Algorithms," *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2023, pp. 1-6

Atmospheric Environment, Vol. 310 1 October 2023, 119987 Air pollution prediction using machine learning techniques – An approach to replace existing monitoring stations with virtual monitoring stations A. Samad , S. Garuda, U. Vogt, B. Yang,
<https://doi.org/10.1016/j.atmosenv.2023.119987>

Davis, D. L. (2002). A look back at the London smog of 1952 and the half century since. Environmental health perspectives, 110(12), A734-A735

Ertel Wolfgang, χ.χ., Εισαγωγή στην Τεχνητή Νοημοσύνη, 2^η εκδ., Εκδόσεις Fountas Books

Rahman, M.M., Nayeem, .E.H., Ahmed, .S. *et al.* AirNet: predictive machine learning model for air quality forecasting using web interface. *Environ Syst Res* **13**, 44 (2024).
<https://doi.org/10.1186/s40068-024-00378-z>

Bo Zhang, Yi Rong, Ruihan Yong, Dongming Qin, Maozhen Li, Guojian Zou, Jianguo Pan Atmospheric Environment, Vol. 290 1 December 2022, 119347 Deep learning for air pollutant concentration prediction: A review. Author links open overlay. <https://doi.org/10.1016/j.atmosenv.2022.119347>

Lee, K., Xue, J., Geyh, A. S., Ozkaynak, H., Leaderer, B. P., Weschler, C. J. & Spengler, J. D. 2002. Nitrous acid, nitrogen dioxide, and ozone concentrations in residential environments. Environ Health Perspect, 110, 145-50.

Zerefos, C. S., Kourtidis, K. A., Melas, D., Balis, D., Zanis, P., Katsaros, L., Mantis, H. T., Repapis, C., Isaksen, I., Sundet, J., Herman, J., Bhartia, P. K. & Calpini, B. 2002. Photochemical Activity and Solar Ultraviolet Radiation (PAUR) Modulation Factors: An overview of the project. *Journal of Geophysical Research*.

Kingham S., B. D., Elliott P., Fischer P., & Lebret E. 2000. Spatial variations in the concentrations of traffic-related pollutants in indoor and outdoor air in Huddersfield, England. *Atmospheric Environment*, 34, 905-916.

WHO 2006a. Air quality guidelines for particulate matter, ozone, nitrogen dioxide and sulfur dioxide. Global update 2005. Summary of risk assessment, World Health Organization, Geneva, Switzerland.

World Health Organization (WHO) (2021). Review of evidence on health aspects of air pollution – REVIHAAP Project

K. P. Moustris, I. C. Ziomas, and A. G. Paliatsos, “3-Day-Ahead Forecasting of Regional Pollution Index for the Pollutants NO₂, CO, SO₂, and O₃ Using Artificial Neural Networks in Athens, Greece,” *Water Air Soil Pollut*, vol. 209, no. 1–4, pp. 29–43, Jun. 2010, doi:10.1007/s11270-0090179-5.

Y. Bai, Y. Li, X. Wang, J. Xie, and C. Li, “Air pollutants concentrations forecasting using back propagation neural network based on wavelet decomposition with meteorological conditions,” *Atmospheric Pollution Research*, vol. 7, no. 3, pp. 557–566, May 2016, doi: 10.1016/j.apr.2016.01.004.

World Health Organization, WHO global air quality guidelines. Particulate matter (PM_{2.5} and PM₁₀), ozone, nitrogen dioxide, sulfur dioxide and carbon monoxide.

M. E. Jenkin and K. C. Clemitshaw, “Ozone and other secondary photochemical pollutants: chemical processes governing their formation in the planetary boundary layer,” *Atmospheric Environment*, vol. 34, no. 16, pp. 2499–2527, Jan. 2000, doi: 10.1016/S1352-2310(99)00478-1

C. M. Bishop, Pattern recognition and machine learning. in *Information science and statistics*. New York: Springer, 2006.

Aur' elien G'eron. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems. O'Reilly Media, 2019.

A Masih. "Application of ensemble learning techniques to model the atmospheric concentration of SO₂". Global Journal of Environmental Science and Management 5.3 (2019), pp. 309–318.

M Zickus, AJ Greig, and M Niranjana. "Comparison of four machine learning methods for predicting PM 10 concentrations in Helsinki, Finland". In: Water, Air and Soil Pollution: Focus 2.5 (2002), pp. 717–729.

Keras, Documentation. Keras: The Python Deep Learning library. s.d. <https://keras.io/>.

<https://pandas.pydata.org/>,

<https://numpy.org/>.

<https://scikit-learn.org/>

www.tensorflow.org

<https://prophet.com>

<https://matplotlib.org>

B. ΕΛΛΗΝΙΚΗ

Ανδρίτσος, 2008, ΑΠΘ, σημειώσεις μαθήματος

Γεντεκάκης Ι. Β., 1999, Ατμοσφαιρική Ρύπανση, Επιπτώσεις, Έλεγχος και Εναλλακτικές τεχνολογίες. Εκδόσεις Τζιόλα, Θεσσαλονίκη.

Ζερεφός, Χ. (2009), Εισαγωγικά Μαθήματα στη Φυσική της Ατμόσφαιρας, σ.σ 34-44, Αθήνα, Παπασωτηρίου

Καραθανάσης Σ., 2006. Ατμοσφαιρική ρύπανση: ορισμός επιπτώσεις, πηγές από βιομηχανικές και βιοτεχνικές δραστηριότητες, τεχνολογία αντιμετώπισης, νομοθεσία. Εκδόσεις Τζιόλα, Θεσσαλονίκη.

Ι. Βλαχάβας, Π. Κεφαλάς, Ν. Βασιλειάδης, Φ. Κόκκορας και Η. Σακελλαρίου, 2020. Τεχνητή Νοημοσύνη, 4^η έκδοση.

Σαχσαμάνογλου, Χ., Μακρογιάννης Τ. (1998), Γενική Μετεωρολογία, Τομέας Μετεωρολογίας και Κλιματολογίας Α.Π.Θ, σ.σ 89, 97-99, Θεσσαλονίκη, ΖΗΤΗ

European Environment Agency, 2021. <https://www.eea.europa.eu/el/highlights/ta-epipeda-atmosfairikis-rypansis-eksakoloythoyn>

ΠΑΡΑΡΤΗΜΑ

Π1. ΚΩΔΙΚΑΣ

main

```
#!/usr/bin/env python

import pandas as pd
import numpy as np
import joblib
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler
from prophet import Prophet
import tensorflow as tf
from tensorflow import keras
from sklearn.model_selection import RandomizedSearchCV

# -----
TRAIN_CSV = 'data/daily_averages.csv'
TEST_CSV = 'data/predict.csv'

def train_and_save_model(pollutant):
    """
    Train and save a Random Forest model for a specific pollutant.
    """
    target_col = pollutant # e.g., 'pm10', 'pm25', ...
    MODEL_FILE = f'{pollutant}_forecast_model.pkl'

    df = pd.read_csv(TRAIN_CSV, parse_dates=['time'])

    # Weather features
    feature_cols = ['tp', 't2m', 'u10', 'v10', 'sp']

    # Add lag and rolling average features for the chosen pollutant
    for lag in [1, 2, 3]:
        df[f'{pollutant}_lag_{lag}'] = df[target_col].shift(lag)
        df[f'{pollutant}_rolling_avg'] = df[target_col].rolling(window=3).mean()
    df = df.dropna()

    feature_cols += [f'{pollutant}_lag_{lag}' for lag in [1, 2, 3]] + [f'{pollutant}_rolling_avg']
    X = df[feature_cols].copy()
    y = df[target_col].copy()
```

```

# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Hyperparameter tuning
param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [None, 10, 20],
    'min_samples_split': [2, 5, 10],
    'min_samples_leaf': [1, 2, 4],
    'max_features': ['auto', 'sqrt']
}
rf = RandomForestRegressor(random_state=42)
rf_random_search = RandomizedSearchCV(rf, param_distributions=param_grid, n_iter=50, cv=3,
n_jobs=-1)
rf_random_search.fit(X_scaled, y)

best_model = rf_random_search.best_estimator_
joblib.dump((best_model, scaler), MODEL_FILE)
print(f"Tuned Random Forest model for '{pollutant}' saved to '{MODEL_FILE}'")

def load_model_and_evaluate(pollutant, plot_results=True):
    """
    Load a Random Forest model for a specific pollutant and evaluate on test data.
    """
    target_col = pollutant
    MODEL_FILE = f'{pollutant}_forecast_model.pkl'

    model, scaler = joblib.load(MODEL_FILE)
    df_test = pd.read_csv(TEST_CSV, parse_dates=['time'])

    # Weather features
    feature_cols = ['tp', 't2m', 'u10', 'v10', 'sp']

    # Add lag and rolling average features
    for lag in [1, 2, 3]:
        df_test[f'{pollutant}_lag_{lag}'] = df_test[target_col].shift(lag)
    df_test[f'{pollutant}_rolling_avg'] = df_test[target_col].rolling(window=3).mean()
    df_test = df_test.dropna() # Drop rows with NaN due to lag and rolling mean

    feature_cols += [f'{pollutant}_lag_{lag}' for lag in [1, 2, 3]] + [f'{pollutant}_rolling_avg']
    X_test = df_test[feature_cols].copy()
    y_test = df_test[target_col].copy()

    # Scale the features using the same scaler from training
    X_test_scaled = scaler.transform(X_test)
    predictions = model.predict(X_test_scaled)

    mae = mean_absolute_error(y_test, predictions)
    rmse = np.sqrt(mean_squared_error(y_test, predictions))
    r2 = r2_score(y_test, predictions)

```

```

print(f"Evaluation for '{pollutant}' on test dataset:\n"
      f" MAE: {mae:.4f}\n"
      f" RMSE: {rmse:.4f}\n"
      f" R^2: {r2:.4f}")

if plot_results:
    plt.figure(figsize=(6, 6))
    plt.scatter(y_test, predictions, alpha=0.5)
    plt.xlabel(f"Actual {pollutant.upper()}")
    plt.ylabel(f"Predicted {pollutant.upper()}")
    plt.title(f"Random Forest - Predicted vs. Actual {pollutant.upper()} (Test Data)")
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
    plt.show()

feature_importances = model.feature_importances_
sorted_idx = np.argsort(feature_importances)[::-1]
sorted_importances = feature_importances[sorted_idx]
sorted_features = [feature_cols[i] for i in sorted_idx]

plt.figure(figsize=(10, 6))
plt.barh(sorted_features, sorted_importances, align='center')
plt.xlabel("Feature Importance")
plt.ylabel("Features")
plt.title(f"Feature Importances for '{pollutant.upper()}' Prediction")
plt.gca().invert_yaxis()
plt.show()

def train_prophet_and_save_model(pollutant):
    """
    Train and save a Prophet model for a specific pollutant.
    """
    target_col = pollutant
    PROPHET_MODEL_FILE = f'{pollutant}_forecast_prophet.pkl'

    df = pd.read_csv(TRAIN_CSV, parse_dates=['time'])
    prophet_df = df[['time', target_col]].copy()
    prophet_df.columns = ['ds', 'y']

    model = Prophet()
    model.fit(prophet_df)

    joblib.dump(model, PROPHET_MODEL_FILE)
    print(f"Prophet model for '{pollutant}' saved to '{PROPHET_MODEL_FILE}'")

def prophet_evaluate(pollutant, plot_results=True):
    """
    Evaluate a previously trained Prophet model for a specific pollutant.
    """

```

```

target_col = pollutant
PROPHET_MODEL_FILE = f'{pollutant}_forecast_prophet.pkl'

model = joblib.load(PROPHET_MODEL_FILE)
df_test = pd.read_csv(TEST_CSV, parse_dates=['time'])
prophet_test = df_test[['time']].copy()
prophet_test.columns = ['ds']

y_test = df_test[target_col].values
forecast = model.predict(prophet_test)
df_test[f'{pollutant}_predicted'] = forecast['yhat']
predictions = df_test[f'{pollutant}_predicted'].values

mae = mean_absolute_error(y_test, predictions)
rmse = np.sqrt(mean_squared_error(y_test, predictions))
r2 = r2_score(y_test, predictions)

print(f"Prophet Evaluation for '{pollutant}' on test dataset:\n"
      f" MAE: {mae:.4f}\n"
      f" RMSE: {rmse:.4f}\n"
      f" R^2: {r2:.4f}")

if plot_results:
    plt.figure(figsize=(6, 6))
    plt.scatter(y_test, predictions, alpha=0.5)
    plt.xlabel(f"Actual {pollutant.upper()}")
    plt.ylabel(f"Predicted {pollutant.upper()}")
    plt.title(f"Prophet - Predicted vs. Actual {pollutant.upper()}")
    plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
    plt.show()

return df_test

def create_tf_dataset(X, y, sequence_length, batch_size):
    dataset = tf.data.Dataset.from_tensor_slices((X, y))
    dataset = dataset.window(sequence_length, shift=1, drop_remainder=True)
    dataset = dataset.flat_map(lambda x, y: tf.data.Dataset.zip((x.batch(sequence_length),
y.batch(sequence_length))))
    dataset = dataset.map(lambda x, y: (tf.reshape(x, (sequence_length, -1)), y[-1])) # Keep the last y value
    dataset = dataset.batch(batch_size).prefetch(tf.data.AUTOTUNE)
    return dataset

def train_rnn_and_save_model(pollutant):
    """
    Train and save an LSTM-based RNN model for a specific pollutant.
    """
    target_col = pollutant
    RNN_MODEL_FILE = f'{pollutant}_rnn_model.keras'
    SCALER_FILE = f'{pollutant}_rnn_model_scaler.pkl'

    df = pd.read_csv(TRAIN_CSV, parse_dates=['time'])

```

```

feature_cols = ['tp', 't2m', 'u10', 'v10', 'sp']

# Add lag and rolling average features
for lag in [1, 2, 3]:
    df[f'{pollutant}_lag_{lag}'] = df[target_col].shift(lag)
df[f'{pollutant}_rolling_avg'] = df[target_col].rolling(window=3).mean()
df = df.dropna()

feature_cols += [f'{pollutant}_lag_{lag}' for lag in [1, 2, 3]] + [f'{pollutant}_rolling_avg']
X = df[feature_cols].values
y = df[target_col].values

# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Create tf.data.Dataset for sequences
sequence_length = 10
batch_size = 16
dataset = create_tf_dataset(X_scaled, y, sequence_length, batch_size)

model = tf.keras.Sequential([
    tf.keras.layers.LSTM(64, activation='tanh', return_sequences=True,
        input_shape=(sequence_length, len(feature_cols))),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.LSTM(32, activation='tanh'),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(16, activation='relu',
        kernel_regularizer=tf.keras.regularizers.l2(0.01)),
    tf.keras.layers.Dense(1)
])

model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=0.001), loss='mse')

early_stopping = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=20,
restore_best_weights=True)
lr_scheduler = tf.keras.callbacks.ReduceLROnPlateau(monitor='val_loss', factor=0.5, patience=10)

model.fit(dataset, epochs=500, validation_data=dataset, verbose=1,
    callbacks=[early_stopping, lr_scheduler])
model.save(RNN_MODEL_FILE)
joblib.dump(scaler, SCALER_FILE)
print(f"LSTM model for '{pollutant}' saved to '{RNN_MODEL_FILE}' and scaler saved to '{SCALER_FILE}'")

def evaluate_rnn_model(pollutant, plot_results=True):
    """
    Evaluate the saved RNN model for a specific pollutant.
    """
    target_col = pollutant
    RNN_MODEL_FILE = f'{pollutant}_rnn_model.keras'

```

```

SCALER_FILE = f'{pollutant}_rnn_model_scaler.pkl'

df_test = pd.read_csv(TEST_CSV, parse_dates=['time'])
feature_cols = ['tp', 't2m', 'u10', 'v10', 'sp']

# Add lag and rolling average features
for lag in [1, 2, 3]:
    df_test[f'{pollutant}_lag_{lag}'] = df_test[target_col].shift(lag)
df_test[f'{pollutant}_rolling_avg'] = df_test[target_col].rolling(window=3).mean()
df_test = df_test.dropna()

feature_cols += [f'{pollutant}_lag_{lag}' for lag in [1, 2, 3]] + [f'{pollutant}_rolling_avg']
X_test = df_test[feature_cols].values
y_test = df_test[target_col].values

scaler = joblib.load(SCALER_FILE)
X_test_scaled = scaler.transform(X_test)

# Create tf.data.Dataset for evaluation
sequence_length = 10
batch_size = 1
test_dataset = create_tf_dataset(X_test_scaled, y_test, sequence_length, batch_size)

model = tf.keras.models.load_model(RNN_MODEL_FILE)
predictions = model.predict(test_dataset).flatten()

# Align y_test and predictions
y_test_aligned = y_test[sequence_length - 1:]

if len(y_test_aligned) != len(predictions):
    print(f"Length mismatch! y_test: {len(y_test_aligned)}, predictions: {len(predictions)}")
    return

mae = mean_absolute_error(y_test_aligned, predictions)
rmse = np.sqrt(mean_squared_error(y_test_aligned, predictions))
r2 = r2_score(y_test_aligned, predictions)

print(f"RNN Evaluation for '{pollutant}' on test dataset:\n"
      f" MAE: {mae:.4f}\n"
      f" RMSE: {rmse:.4f}\n"
      f" R^2: {r2:.4f}")

if plot_results:
    plt.figure(figsize=(6, 6))
    plt.scatter(y_test_aligned, predictions, alpha=0.5)
    plt.xlabel(f"Actual {pollutant.upper()}")
    plt.ylabel(f"Predicted {pollutant.upper()}")
    plt.title(f"LSTM - Predicted vs. Actual {pollutant.upper()}")
    plt.plot([y_test_aligned.min(), y_test_aligned.max()],
             [y_test_aligned.min(), y_test_aligned.max()], 'r--')
    plt.show()

```

```

if __name__ == "__main__":
    # List available modes
    print("\nAvailable modes:")
    print("1) train")
    print("2) evaluate")
    print("3) prophet_train")
    print("4) prophet_evaluate")
    print("5) rnn_train")
    print("6) rnn_evaluate")

    try:
        mode_number = int(input("\nPlease select a mode (1-6): ").strip())
    except ValueError:
        print("Invalid input. Please enter a number from 1 to 6.")
        exit()

    # Ask user which pollutant they want to work with
    print("\nAvailable pollutants: pm25, pm10, o3, no2, so2, co")
    pollutant = input("Please select a pollutant: ").strip().lower()
    valid_pollutants = ['pm25', 'pm10', 'o3', 'no2', 'so2', 'co']
    if pollutant not in valid_pollutants:
        print(f"Invalid pollutant '{pollutant}'. Exiting.")
        exit()

    # Toggle to True/False if you want plots
    plot_results = True

    if mode_number == 1:
        train_and_save_model(pollutant)
    elif mode_number == 2:
        load_model_and_evaluate(pollutant, plot_results=plot_results)
    elif mode_number == 3:
        train_prophet_and_save_model(pollutant)
    elif mode_number == 4:
        prophet_evaluate(pollutant, plot_results=plot_results)
    elif mode_number == 5:
        train_rnn_and_save_model(pollutant)
    elif mode_number == 6:
        evaluate_rnn_model(pollutant, plot_results=plot_results)
    else:
        print("Invalid number selected.")

```

rnn_all.py


```

#!/usr/bin/env python

import pandas as pd
import numpy as np
import joblib
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

TRAIN_CSV = 'data/daily_averages.csv'
TEST_CSV = 'data/predict.csv'

RNN_MODEL_FILE = 'all_pollutants_rnn_model.keras'
SCALER_FILE = 'all_pollutants_rnn_model_scaler.pkl'

POLLUTANT_COLS = ['pm25', 'pm10', 'o3', 'no2', 'so2', 'co']

WEATHER_COLS = ['tp', 't2m', 'u10', 'v10', 'sp']

def create_tf_dataset(X, y, sequence_length, batch_size):
    """
    Creates a tf.data.Dataset for RNN training/evaluation where each sample
    is a sequence of length 'sequence_length'.
    """
    dataset = tf.data.Dataset.from_tensor_slices((X, y))
    dataset = dataset.window(sequence_length, shift=1, drop_remainder=True)
    dataset = dataset.flat_map(
        lambda x, y: tf.data.Dataset.zip((x.batch(sequence_length), y.batch(sequence_length)))
    )
    # For each sequence window, we feed the entire X window into the model
    # but only use the final row's y as the label.
    dataset = dataset.map(
        lambda x, y: (tf.reshape(x, (sequence_length, -1)), y[-1])
    )
    dataset = dataset.batch(batch_size).prefetch(tf.data.AUTOTUNE)
    return dataset

def train_rnn_for_all_pollutants():
    """
    Train and save a single LSTM-based RNN model that predicts
    all pollutants simultaneously.
    """
    df = pd.read_csv(TRAIN_CSV, parse_dates=['time'])

    # For each pollutant, add lag and rolling average features
    for pollutant in POLLUTANT_COLS:
        for lag in [1, 2, 3]:
            df[f'{pollutant}_lag_{lag}'] = df[pollutant].shift(lag)

```

```

df[f'{pollutant}_rolling_avg'] = df[pollutant].rolling(window=3).mean()

# Drop rows with NaN from shifting/rolling
df = df.dropna()

# Construct feature columns:
# - Weather features
# - Lags and rolling averages for each pollutant
feature_cols = []
feature_cols.extend(WEATHER_COLS)
for pollutant in POLLUTANT_COLS:
    feature_cols += [f'{pollutant}_lag_{lag}' for lag in [1, 2, 3]]
    feature_cols.append(f'{pollutant}_rolling_avg')

# Inputs (X) and multi-output targets (Y)
X = df[feature_cols].values
Y = df[POLLUTANT_COLS].values # shape: (num_samples, 6)

# Scale all features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Create tf.data.Dataset for sequences
sequence_length = 10
batch_size = 16
train_dataset = create_tf_dataset(X_scaled, Y, sequence_length, batch_size)

model = keras.Sequential([
    keras.layers.LSTM(128, activation='tanh', return_sequences=True,
                      input_shape=(sequence_length, X_scaled.shape[1])),
    keras.layers.Dropout(0.3),
    keras.layers.LSTM(64, activation='tanh', return_sequences=True),
    keras.layers.Dropout(0.3),
    keras.layers.LSTM(64, activation='tanh'),
    keras.layers.Dropout(0.3),
    keras.layers.Dense(32, activation='relu',
                       kernel_regularizer=keras.regularizers.l2(0.01)),
    keras.layers.Dense(len(POLLUTANT_COLS))
])

model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.001), loss='mse')

early_stopping = keras.callbacks.EarlyStopping(monitor='val_loss',
                                                patience=20,
                                                restore_best_weights=True)
lr_scheduler = keras.callbacks.ReduceLROnPlateau(monitor='val_loss',
                                                  factor=0.5,
                                                  patience=10)

model.fit(train_dataset,
          epochs=500,

```

```

        validation_data=train_dataset,
        callbacks=[early_stopping, lr_scheduler],
        verbose=1)

model.save(RNN_MODEL_FILE)
joblib.dump(scaler, SCALER_FILE)

print(f"Trained multi-pollutant LSTM model saved to '{RNN_MODEL_FILE}'")
print(f"Scaler saved to '{SCALER_FILE}'")

def evaluate_rnn_for_all_pollutants(plot_results=True):
    """
    Evaluate the saved LSTM model on test data for all pollutants simultaneously.
    Computes MAE, RMSE, and R^2 for each pollutant.
    """
    df_test = pd.read_csv(TEST_CSV, parse_dates=['time'])

    for pollutant in POLLUTANT_COLS:
        for lag in [1, 2, 3]:
            df_test[f'{pollutant}_lag_{lag}'] = df_test[pollutant].shift(lag)
            df_test[f'{pollutant}_rolling_avg'] = df_test[pollutant].rolling(window=3).mean()

    df_test = df_test.dropna()

    feature_cols = []
    feature_cols.extend(WEATHER_COLS)
    for pollutant in POLLUTANT_COLS:
        feature_cols += [f'{pollutant}_lag_{lag}' for lag in [1, 2, 3]]
        feature_cols.append(f'{pollutant}_rolling_avg')

    X_test = df_test[feature_cols].values
    Y_test = df_test[POLLUTANT_COLS].values

    scaler = joblib.load(SCALER_FILE)
    X_test_scaled = scaler.transform(X_test)

    sequence_length = 10
    batch_size = 1
    test_dataset = create_tf_dataset(X_test_scaled, Y_test, sequence_length, batch_size)

    model = keras.models.load_model(RNN_MODEL_FILE)
    predictions = model.predict(test_dataset) # shape: (num_samples, 6)

    Y_test_aligned = Y_test[sequence_length - 1:] # shape should match predictions

    if len(Y_test_aligned) != len(predictions):
        print(f"Length mismatch! Y_test_aligned: {len(Y_test_aligned)}, predictions: {len(predictions)}")
        return

    for i, pollutant in enumerate(POLLUTANT_COLS):
        y_true = Y_test_aligned[:, i]

```

```

y_pred = predictions[:, i]

mae = mean_absolute_error(y_true, y_pred)
rmse = np.sqrt(mean_squared_error(y_true, y_pred))
r2 = r2_score(y_true, y_pred)
print(f"\nEvaluation for '{pollutant}':")
print(f" MAE: {mae:.4f}")
print(f" RMSE: {rmse:.4f}")
print(f" R^2: {r2:.4f}")

if plot_results:
    plt.figure(figsize=(6, 6))
    plt.scatter(y_true, y_pred, alpha=0.5)
    plt.xlabel(f"Actual {pollutant.upper()}")
    plt.ylabel(f"Predicted {pollutant.upper()}")
    plt.title(f"LSTM - Predicted vs. Actual {pollutant.upper()}")
    min_val, max_val = min(y_true.min(), y_pred.min()), max(y_true.max(), y_pred.max())
    plt.plot([min_val, max_val], [min_val, max_val], 'r--')
    plt.show()

if __name__ == "__main__":
    print("\nAvailable modes for multi-pollutant RNN:")
    print("1) train")
    print("2) evaluate")

    try:
        mode_number = int(input("\nPlease select a mode (1 or 2): ").strip())
    except ValueError:
        print("Invalid input. Please enter 1 or 2.")
        exit()

    if mode_number == 1:
        train_rnn_for_all_pollutants()
    elif mode_number == 2:
        evaluate_rnn_for_all_pollutants(plot_results=True)
    else:
        print("Invalid selection. Exiting.")

```

merge day.py

```

import pandas as pd

def create_daily_averages(input_csv, output_csv):
    df = pd.read_csv(input_csv)

    df['time'] = pd.to_datetime(df['time'])

    df['date'] = df['time'].dt.date

```

```

daily_averages = df.groupby('date', as_index=False).mean()

if 'time' in daily_averages.columns:
    daily_averages.drop(columns=['time'], inplace=True)

daily_averages = daily_averages.round(8)

daily_averages.to_csv(output_csv, index=False)

if __name__ == "__main__":
    input_file = "data/all_weather_pollution.csv"
    output_file = "data/daily_averages.csv"

    create_daily_averages(input_file, output_file)
    print(f"Daily averages saved to {output_file}.")

```

merge.py

```

import os
import json
import pandas as pd
from datetime import datetime

CSV_PATH = "data/air_quality/london-air-quality-normalized.csv"

JSON_FOLDER = "data/split_json"

OUTPUT_FOLDER = "data/merged_json"

os.makedirs(OUTPUT_FOLDER, exist_ok=True)

df = pd.read_csv(CSV_PATH)

pollution_by_date = {}
for _, row in df.iterrows():
    dt = datetime.strptime(row["date"], "%Y/%m/%d")
    date_str = dt.strftime("%Y-%m-%d")

    pollution_by_date[date_str] = {
        "pm25": row["pm25"],
        "pm10": row["pm10"],
        "o3": row["o3"],
        "no2": row["no2"],
        "so2": row["so2"],
        "co": row["co"]
    }

for date_str, pollution_data in pollution_by_date.items():
    json_filename = f"{date_str}.json"

```

```

json_path = os.path.join(JSON_FOLDER, json_filename)

if os.path.exists(json_path):
    with open(json_path, "r", encoding="utf-8") as f:
        data = json.load(f)

    data["pollution"] = pollution_data
    output_path = os.path.join(OUTPUT_FOLDER, json_filename)
    with open(output_path, "w", encoding="utf-8") as out_f:
        json.dump(data, out_f, ensure_ascii=False, indent=2)

    print(f"Saved merged JSON for {date_str} -> {output_path}")
else:
    print(f"No JSON file found for {date_str}, skipping...")

```

normalize.py

```

import csv
import os

def normalize_air_quality_data(
    input_path='data/air_quality/london-air-quality.csv',
    output_path='data/air_quality/london-air-quality-normalized.csv'
):
    with open(input_path, mode='r', newline="", encoding='utf-8') as infile:
        reader = csv.DictReader(infile)
        fieldnames = reader.fieldnames

        cleaned_rows = []

        for row in reader:
            if (row['pm25'].strip() == " or
                row['pm10'].strip() == " or
                row['o3'].strip() == " or
                row['no2'].strip() == "):
                continue

            if row['so2'].strip() == ":
                row['so2'] = '0'
            if row['co'].strip() == ":
                row['co'] = '0'

            cleaned_rows.append(row)

        with open(output_path, mode='w', newline="", encoding='utf-8') as outfile:
            writer = csv.DictWriter(outfile, fieldnames=fieldnames)
            writer.writeheader()
            writer.writerows(cleaned_rows)

def main():

```

```

input_csv = 'data/air_quality/london-air-quality.csv'
output_csv = 'data/air_quality/london-air-quality-normalized.csv'

os.makedirs(os.path.dirname(output_csv), exist_ok=True)

normalize_air_quality_data(input_csv, output_csv)
print(f'Normalized air quality data has been saved to {output_csv}')

if __name__ == '__main__':
    main()

```

open_weather.py

```

import os
import json
import xarray as xr
import datetime

BASE_DIR = "data/weather"
OUTPUT_DIR = "data/weather_json"

def default_converter(obj):
    if isinstance(obj, (datetime.datetime, datetime.date)):
        return obj.isoformat()

    raise TypeError(f"Type {type(obj)} not serializable")

def convert_era5_nc_to_json():
    os.makedirs(OUTPUT_DIR, exist_ok=True)

    for day_dir in os.listdir(BASE_DIR):
        full_day_path = os.path.join(BASE_DIR, day_dir)

        if not os.path.isdir(full_day_path):
            continue
        if not day_dir.startswith("era5_weather_"):
            continue

        accum_file = os.path.join(full_day_path, "data_stream-oper_stepType-accum.nc")
        instant_file = os.path.join(full_day_path, "data_stream-oper_stepType-instant.nc")

        if not (os.path.isfile(accum_file) and os.path.isfile(instant_file)):
            print(f"Skipping {day_dir}: missing accum/instant files.")
            continue

        print(f"Processing {day_dir}...")

        ds_accum = xr.open_dataset(accum_file)
        ds_instant = xr.open_dataset(instant_file)

```

```

ds_merged = xr.merge([ds_accum, ds_instant])

data_dict = ds_merged.to_dict(data=True)

output_filename = f"{day_dir}.json"
output_path = os.path.join(OUTPUT_DIR, output_filename)

with open(output_path, "w") as f:
    json.dump(data_dict, f, indent=2, default=default_converter)

ds_accum.close()
ds_instant.close()
print(f"Finished writing {output_path}.")

if __name__ == "__main__":
    convert_era5_nc_to_json()

```

scrape_weather.py

```

import cdsapi
import pandas as pd
import os
import time
from tqdm import tqdm

c = cdsapi.Client()

weather_output_dir = 'data/weather'
os.makedirs(weather_output_dir, exist_ok=True)

start_date = '2024-01-01'
end_date = '2024-11-30'

dates = pd.date_range(start=start_date, end=end_date, freq='D')

batch_size = 7

# Define the geographic area for London
london_area = [
    51.70, -0.5, 51.30, 0.3
]

def chunk_dates(date_list, batch_size):
    for i in range(0, len(date_list), batch_size):
        yield date_list[i:i + batch_size]

def format_dates(batch):
    years = sorted(list(set([date.strftime('%Y') for date in batch])))

```



```

months = sorted(list(set([date.strftime('%m') for date in batch])))
days = sorted(list(set([date.strftime('%d') for date in batch])))
return years, months, days

def download_weather(batch, batch_output_filename):
    years, months, days = format_dates(batch)
    try:
        c.retrieve(
            'reanalysis-era5-single-levels',
            {
                'product_type': 'reanalysis',
                'format': 'netcdf',
                'variable': [
                    '2m_temperature',
                    'total_precipitation',
                    '10m_u_component_of_wind',
                    '10m_v_component_of_wind',
                    'surface_pressure',
                    'weather_symbol_1h',
                ],
                'year': years,
                'month': months,
                'day': days,
                'time': [
                    '08:00', '12:00', '18:00',
                ],
                'area': london_area,
            },
            os.path.join(weather_output_dir, batch_output_filename))
        print(f"Weather data downloaded for dates: {batch[0].strftime('%Y-%m-%d')} to {batch[-1].strftime('%Y-%m-%d')}")
    except Exception as e:
        print(f"Failed to download weather data for dates: {batch[0].strftime('%Y-%m-%d')} to {batch[-1].strftime('%Y-%m-%d')}")
        print(f"Error: {e}")

for batch in tqdm(list(chunk_dates(dates, batch_size)), desc="Downloading Data"):
    batch_start = batch[0].strftime('%Y%m%d')
    batch_end = batch[-1].strftime('%Y%m%d')

    weather_filename = f'era5_weather_{batch_start}.nc'

    download_weather(batch, weather_filename)

    time.sleep(1)

print("Data download completed.")

```

split_data.py

```

import os
import json
from datetime import datetime
from collections import defaultdict

def split_weather_json_by_day(input_filepath, output_dir):
    with open(input_filepath, 'r') as f:
        weather_data = json.load(f)

    valid_times_str = weather_data["coords"]["valid_time"]["data"]
    valid_times_dt = [datetime.fromisoformat(ts) for ts in valid_times_str]

    day_to_indices = defaultdict(list)
    for idx, dt_obj in enumerate(valid_times_dt):
        day_str = dt_obj.strftime("%Y-%m-%d")
        day_to_indices[day_str].append(idx)

    data_vars = weather_data.get("data_vars", {})
    coords = weather_data.get("coords", {})

    for day_str, indices in day_to_indices.items():
        indices_sorted = sorted(indices)

        new_json = {
            "coords": {},
            "data_vars": {}
        }

        for coord_key, coord_val in coords.items():
            data_field = coord_val.get("data", None)
            if data_field is None:
                continue

            if isinstance(data_field, list) and len(data_field) == len(valid_times_str):
                new_data = [data_field[i] for i in indices_sorted]
            else:
                new_data = data_field

            new_json["coords"][coord_key] = {
                "data": new_data
            }

        for var_name, var_info in data_vars.items():
            var_data = var_info.get("data", None)
            if var_data is None:
                continue

            if len(var_data) == len(valid_times_str):
                sliced_data = [var_data[i] for i in indices_sorted]
            else:
                sliced_data = var_data

```

```

    new_json["data_vars"][var_name] = {
        "data": sliced_data
    }

    os.makedirs(output_dir, exist_ok=True)
    output_filename = f"{day_str}.json"
    output_path = os.path.join(output_dir, output_filename)
    with open(output_path, 'w') as out_f:
        json.dump(new_json, out_f, indent=2)

def main():
    input_folder = "data/weather_json"
    output_folder = "data/split_json"

    for filename in os.listdir(input_folder):
        if filename.endswith(".json"):
            in_file = os.path.join(input_folder, filename)
            split_weather_json_by_day(in_file, output_folder)

    print("Done splitting JSON files by day ")

if __name__ == "__main__":
    main()

```

to_csv.py

```

import os
import glob
import json
import numpy as np
import pandas as pd

def parse_weather_json(file_path, agg_func=np.mean):
    with open(file_path, 'r') as f:
        data = json.load(f)

    times = data["coords"]["valid_time"]["data"]

    pm25 = data["pollution"].get("pm25", None)
    pm10 = data["pollution"].get("pm10", None)
    o3 = data["pollution"].get("o3", None)
    no2 = data["pollution"].get("no2", None)
    so2 = data["pollution"].get("so2", None)
    co = data["pollution"].get("co", None)

    def get_variable(var_name):
        return np.array(data["data_vars"][var_name]["data"], dtype=np.float32)

    tp_3d = get_variable("tp")

```

```

t2m_3d = get_variable("t2m")
u10_3d = get_variable("u10")
v10_3d = get_variable("v10")
sp_3d = get_variable("sp")

records = []
for t_idx, time_val in enumerate(times):
    tp_val = agg_func(tp_3d[t_idx])
    t2m_val = agg_func(t2m_3d[t_idx])
    u10_val = agg_func(u10_3d[t_idx])
    v10_val = agg_func(v10_3d[t_idx])
    sp_val = agg_func(sp_3d[t_idx])

    record = {
        "time": time_val,
        "tp": tp_val,
        "t2m": t2m_val,
        "u10": u10_val,
        "v10": v10_val,
        "sp": sp_val,
        "pm25": pm25,
        "pm10": pm10,
        "o3": o3,
        "no2": no2,
        "so2": so2,
        "co": co
    }
    records.append(record)

return records

def create_csv_from_json(folder_path, output_csv_path, agg_func=np.mean):
    """
    Reads all JSON files from folder_path, extracts aggregated weather+pollution data,
    compiles into a single CSV (omitting lat/lon).
    Also shows an example correlation analysis at the end.
    """
    all_records = []

    json_files = glob.glob(os.path.join(folder_path, "*.json"))

    for jf in json_files:
        records = parse_weather_json(jf, agg_func=agg_func)
        all_records.extend(records)

    df = pd.DataFrame(all_records)

    df['time'] = pd.to_datetime(df['time'], errors='coerce')

    df.sort_values(by='time', inplace=True)

```

```

df.to_csv(output_csv_path, index=False)
print(f"CSV created: {output_csv_path}")

if __name__ == "__main__":
    input_folder = "data/merged_json"
    output_csv = "data/all_weather_pollution.csv"

    create_csv_from_json(input_folder, output_csv, agg_func=np.mean)

```

unzip.py

```

import os
import subprocess

folder = "data/weather"

for file_name in os.listdir(folder):
    if not file_name.endswith(".nc"):
        continue

    nc_path = os.path.join(folder, file_name)
    zip_name = file_name[:-3] + ".zip"
    zip_path = os.path.join(folder, zip_name)
    os.rename(nc_path, zip_path)

    output_folder = os.path.splitext(zip_path)[0]
    os.makedirs(output_folder, exist_ok=True)
    subprocess.run(["unzip", zip_path, "-d", output_folder])

    os.remove(zip_path)

```

Π2. ΔΟΜΗ & ΑΝΑΛΥΣΗ ΚΩΔΙΚΑ

Η δομή του κώδικα είναι ιεραρχική και οργανωμένη γύρω από τις εξής αρχές:

1. Εισαγωγή Βιβλιοθηκών. Όλες οι απαραίτητες βιβλιοθήκες εισάγονται στην αρχή. Αυτές περιλαμβάνουν τις βιβλιοθήκες για δεδομένα και ανάλυση (pandas, numpy, joblib), για machine learning (sklearn, tensorflow/keras, prophet), και για γραφήματα (matplotlib).
2. Κύριες Συναρτήσεις. Ο κώδικας είναι οργανωμένος σε συναρτήσεις που εκτελούν συγκεκριμένες εργασίες, οι οποίες είναι:

α. Εκπαίδευση και Αποθήκευση Μοντέλων:

- train_and_save_model(pollutant) (Random Forest),
- train_prophet_and_save_model(pollutant) (Prophet) και
- train_rnn_and_save_model(pollutant) (LSTM).

β. Αξιολόγηση Μοντέλων

- load_model_and_evaluate(pollutant) (Random Forest):
 - Φορτώνει το αποθηκευμένο μοντέλο.
 - Υπολογίζει μετρικές (MAE, RMSE, R^2).
 - Δημιουργεί γραφήματα για πραγματικές vs προβλεπόμενες τιμές.
- prophet_evaluate(pollutant) (Prophet):
 - Παρόμοιο με το παραπάνω, αλλά ειδικά για χρονοσειρές.
- evaluate_rnn_model(pollutant) (LSTM):
 - Φορτώνει το αποθηκευμένο LSTM μοντέλο και scaler.
 - Παράγει μετρικές και γραφήματα.

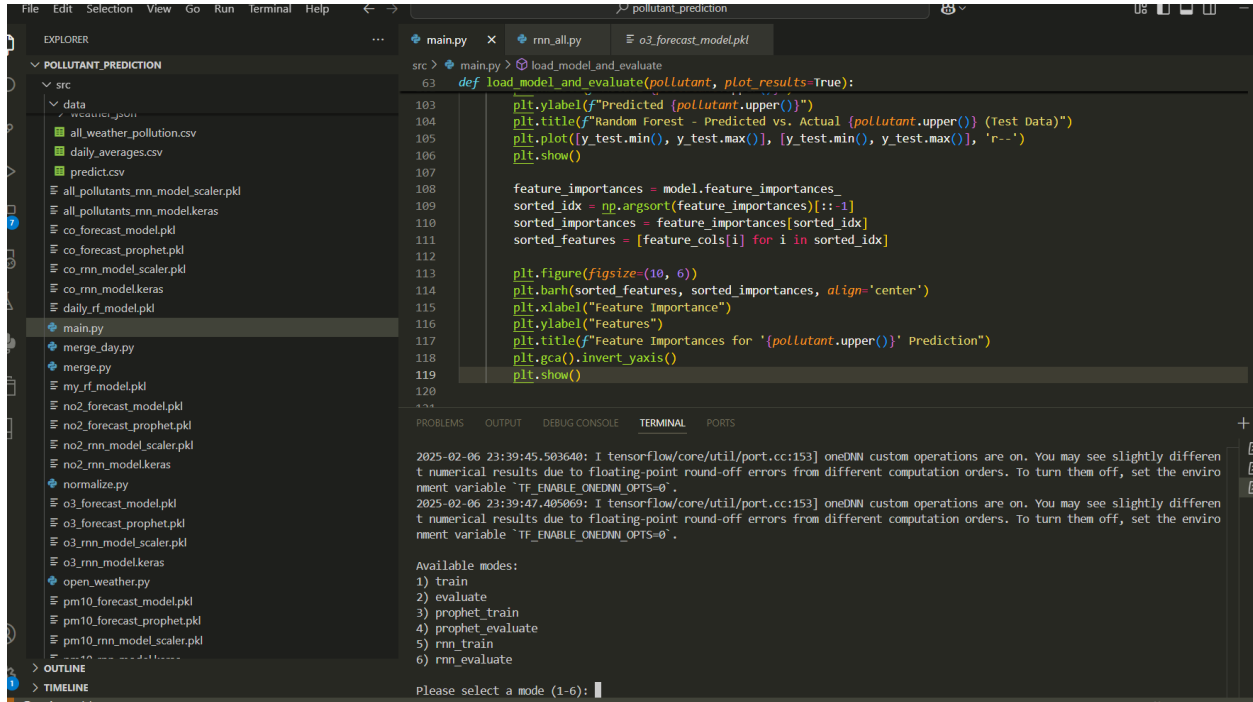
3. Βοηθητικές Συναρτήσεις

- create_tf_dataset(data, sequence_length):
 - Μετατρέπει δεδομένα σε tf.data.Dataset για LSTM εκπαίδευση.
- Άλλες βοηθητικές: Συναρτήσεις για χειρισμό δεδομένων (π.χ., δημιουργία lags).

4. Κύριος Βρόχος (Main Loop)

Ο κώδικας περιέχει έναν κύριο βρόχο, ο οποίος ζητά από τον χρήστη να:

- Επιλέξει λειτουργία (1-6): π.χ., εκπαίδευση/αξιολόγηση μοντέλων.
- Επιλέξει ρύπο (π.χ., pm25, o3, no2).



```

src > main.py > load_model_and_evaluate
63 def load_model_and_evaluate(pollutant, plot_results=True):
103     plt.ylabel(f"Predicted {pollutant.upper()}")
104     plt.title(f"Random Forest - Predicted vs. Actual {pollutant.upper()} (Test Data)")
105     plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
106     plt.show()
107
108     feature_importances = model.feature_importances_
109     sorted_idx = np.argsort(feature_importances)[::-1]
110     sorted_importances = feature_importances[sorted_idx]
111     sorted_features = [feature_cols[i] for i in sorted_idx]
112
113     plt.figure(figsize=(10, 6))
114     plt.barh(sorted_features, sorted_importances, align='center')
115     plt.xlabel("Feature Importance")
116     plt.ylabel("Features")
117     plt.title(f"Feature Importances for '{pollutant.upper()}' Prediction")
118     plt.gca().invert_yaxis()
119     plt.show()
120
121
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
2025-02-06 23:39:45.503640: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-02-06 23:39:47.405069: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.

Available modes:
1) train
2) evaluate
3) prophet_train
4) prophet_evaluate
5) rnn_train
6) rnn_evaluate

Please select a mode (1-6):

```

```

2025-02-06 23:39:47.405069: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.

Available modes:
1) train
2) evaluate
3) prophet_train
4) prophet_evaluate
5) rnn_train
6) rnn_evaluate

Please select a mode (1-6): 1

Available pollutants: pm25, pm10, o3, no2, so2, co
Please select a pollutant: pm10

```

```

C:\Users\dovad\AppData\Local\Programs\Python\Python312\Lib\site-packages\numpy\ma\core.py:2846: RuntimeWarning: invalid value
encountered in cast
_data = np.array(data, dtype=dtype, copy=copy,
C:\Users\dovad\AppData\Local\Programs\Python\Python312\Lib\site-packages\sklearn\model_selection\_search.py:1103: UserWarning:
One or more of the test scores are non-finite: [0.82546017 0.80538632 0.80874443 nan 0.81949164 nan
0.80253649 nan 0.80030899 nan 0.80693919 nan
nan nan 0.82228828 0.82497625 nan 0.79881145
nan nan nan 0.79204478 nan 0.81013455
0.82622629 nan nan 0.82119932 nan nan
nan 0.82903026 0.80796943 0.81335337 0.79862079 0.80902194
nan nan nan 0.80801427 0.80954159 0.80857033
0.80045514 0.8190168 0.79442294 0.82535236 nan 0.82467641
0.80857245 0.79512117]
warnings.warn(

```

```

Please select a pollutant: pm25
2025-02-06 23:44:30.741614: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use av
ailable CPU instructions in performance-critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
1/Unknown 0s 275ms/step2025-02-06 23:44:31.373056: I tensorflow/core/framework/local_rendezvous.cc:405] Local rendezvous
is aborting with status: OUT_OF_RANGE: End of sequence
C:\Users\dovad\AppData\Local\Programs\Python\Python312\Lib\site-packages\keras\src\trainers\epoch_iterator.py:151: UserWarning
: Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps_per
_epoch * epochs` batches. You may need to use the `.repeat()` function when building your dataset.
self._interrupted_warning()
18/18 ————— 0s 2ms/step
RNN Evaluation for 'pm25' on test dataset:
MAE: 0.3737
RMSE: 0.5290
R^2: 0.9966
PS C:\git\pollutant prediction\src>

```

```

Epoch 21/500
242/242 ————— 2s 7ms/step - loss: 72.1306 - val_loss: 14.0403 - learning_rate: 0.0010
Epoch 22/500
242/242 ————— 2s 6ms/step - loss: 69.4635 - val_loss: 18.3007 - learning_rate: 0.0010
Epoch 23/500
242/242 ————— 1s 6ms/step - loss: 60.9023 - val_loss: 12.8806 - learning_rate: 0.0010
Epoch 24/500
242/242 ————— 1s 5ms/step - loss: 66.3113 - val_loss: 14.3982 - learning_rate: 0.0010
Epoch 25/500
242/242 ————— 1s 5ms/step - loss: 64.0902 - val_loss: 12.2984 - learning_rate: 0.0010
Epoch 26/500
242/242 ————— 1s 5ms/step - loss: 58.0983 - val_loss: 16.0626 - learning_rate: 0.0010
Epoch 27/500
242/242 ————— 2s 7ms/step - loss: 61.4069 - val_loss: 13.0320 - learning_rate: 0.0010
Epoch 28/500
239/242 ————— 0s 5ms/step - loss: 58.5107

```



```
epoch 239/500
242/242 ██████████ 1s 4ms/step - loss: 12.5927 - val_loss: 1.6791 - learning_rate: 7.8125e-06
epoch 240/500
242/242 ██████████ 2s 8ms/step - loss: 12.6274 - val_loss: 1.6861 - learning_rate: 7.8125e-06
epoch 241/500
242/242 ██████████ 2s 8ms/step - loss: 11.7936 - val_loss: 1.6744 - learning_rate: 7.8125e-06
epoch 242/500
242/242 ██████████ 2s 8ms/step - loss: 13.2422 - val_loss: 1.6833 - learning_rate: 7.8125e-06
epoch 243/500
242/242 ██████████ 2s 8ms/step - loss: 12.9106 - val_loss: 1.6475 - learning_rate: 7.8125e-06
epoch 244/500
242/242 ██████████ 2s 7ms/step - loss: 11.6051 - val_loss: 1.6127 - learning_rate: 7.8125e-06
epoch 245/500
242/242 ██████████ 2s 7ms/step - loss: 13.0385 - val_loss: 1.6510 - learning_rate: 7.8125e-06
STM model for 'pm25' saved to 'pm25_rnn_model.keras' and scaler saved to 'pm25_rnn_model_scaler.pkl'
PS C:\git\pollutant prediction\src>
```