



**Teknoloji Fakültesi**

**MARMARA ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BİTİRME PROJESİ**

İstanbul'un Hava Kalitesinin Tahmini için LSTM, GRU ve Transformer Model Tabanlı  
Derin Öğrenme Yöntemlerinin İncelenmesi

**PROJE YAZARI**

Hamza Gençay  
170422822

Erdem Saçan  
170422826

Çağan Derbent  
170421028

**DANIŞMAN**

Doç. Dr. Buket Doğan

**İL, TEZ YILI**

İstanbul, 2025



**Teknoloji Fakültesi**

**MARMARA UNIVERSITY**

**FACULTY OF TECHNOLOGY**

**DEPARTMENT OF COMPUTER ENGINEERING**

**GRADUATION PROJECT**

Analysis of Deep Learning Methods Based on LSTM, GRU and Transformer Models for Air  
Quality Prediction in Istanbul

**PROJECT AUTHOR**

Hamza Gençay  
170422822

Erdem Saçan  
170422826

Çağan Derbent  
170421028

**ADVISOR**

Assoc. Prof. Buket Doğan

**CITY, THESIS YEAR**

Istanbul, 2025

**MARMARA ÜNİVERSİTESİ**  
**TEKNOLOJİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencileri Hamza Gençay, Erdem Saçan ve Çağan Derbent tarafından gerçekleştirilen “İstanbul’un Hava Kalitesinin Tahmini için LSTM, GRU ve Transformer Model Tabanlı Derin Öğrenme Yöntemlerinin İncelenmesi” başlıklı bitirme projesi çalışması, 19/06/2025 tarihinde sunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyesi**

Doç. Dr. Buket Doğan (Danışman)

Marmara Üniversitesi ..... (İMZA) .....

Prof. Dr. Ali Buldu (Üye)

Marmara Üniversitesi ..... (İMZA) .....

# İÇİNDEKİLER

Sayfa

KISALTMALAR .....	iii
SEMBOLLER .....	iv
ŞEKİL LİSTESİ .....	v
ÖZET .....	1
BÖLÜM 1. GİRİŞ .....	2
1.1. Bitirme Projesinin Amacı .....	3
1.2. Literatür Taraması.....	3
BÖLÜM 2. MATERYAL VE YÖNTEM .....	12
2.1. Verinin Temin Edilmesi.....	12
2.2. Veri Ön İşleme Adımları .....	13
2.3. LSTM Modeli ile PM10 Tahmini .....	14
2.3.1. Gerekli Kütüphanelerin Yüklenmesi.....	15
2.3.2. Verilerin Yüklenmesi ve Düzenlenmesi .....	15
2.3.3. Giriş ve Hedef Sütunlarının Belirlenmesi .....	16
2.3.4. Eğitim ve Test Verilerinin Ayrılması.....	16
2.3.5. Verilerin Normalize Edilmesi .....	16
2.3.6. LSTM Modelinin Tanımlanması .....	17
2.3.7. Modelin Eğitilmesi.....	17
2.3.8. Test Verisi Üzerinde Tahmin .....	17
2.3.9. Performans Metriklerinin Hesaplanması .....	17
2.3.10 Sonuçların Görselleştirilmesi.....	18
2.4. LSTM ile Gelecek 2 Yıl Tahmini.....	20
2.4.1. Yardımcı Kütüphanelerin Aktarılması ve Uyarıların Bastırılması ..	20
2.4.2. Ay ve Gün Bilgilerinin Elde Edilmesi .....	20
2.4.3. Başlangıç Girdisinin Hazırlanması ve Tahmin Zaman Aralığının Belirlenmesi .....	20
2.4.4. Recursive (Adım Adım) Tahmin Döngüsü .....	21
2.4.5. Tahmin Sonuçlarının Veri Çerçevesine Aktarılması .....	21
2.4.6. Tahmin Sonuçlarının CSV Dosyasına Yazdırılması.....	22

2.4.7. 2 Yıllık Tahminin Görselleştirilmesi .....	22
2.5. LSTM Modeli Tahmin Sonuçları .....	22
2.6. Transformer Modeli ile PM10 Tahmini.....	22
2.6.1. Transformer Encoder Katmanı.....	23
2.6.2. Transformer Model Mimarisi.....	23
2.6.3. Model Eğitimi ve Derlenmesi .....	24
2.6.4. Performans Değerlendirmesi.....	24
2.7. Transformer Modeli Tahmin Sonuçları .....	25
2.8. GRU Modeli ile PM10 Tahmini .....	26
2.8.1. GRU Model Mimarisi .....	26
2.8.2. Modelin Derlenmesi ve Eğitilmesi .....	26
2.8.3. Performans Değerlendirmesi .....	27
2.9 GRU Modeli Tahmin Sonuçları.....	28
2.10. Modellerin Karşılaştırılması .....	28
BÖLÜM 3. BULGULAR VE TARTIŞMA.....	30
BÖLÜM 4. SONUÇLAR .....	31
KAYNAKÇA .....	32

## KISALTMALAR/ABBREVIATIONS

**LSTM** : Long Short-Term Memory

**RMSE** : Root Mean Square Error (Kök Ortalama Kare Hata)

**MAE** : Mean Absolute Error (Ortalama Mutlak Hata)

**R<sup>2</sup>** : Determination Coefficient (Belirleme Katsayısı)

**AQI** : Air Quality Index (Hava Kalitesi İndeksi)

**API** : Application Programming Interface

**DL** : Deep Learning (Derin Öğrenme)

**ANN** : Artificial Neural Network (Yapay Sinir Ağı)

**ARIMA** : AutoRegressive Integrated Moving Average

**CSV** : Comma-Separated Values (Virgülle Ayrılmış Değerler)

**KNN** : K-Nearest Neighbors (K-En Yakın Komşu)

**ML** : Machine Learning (Makine Öğrenmesi)

**GRU** : Gated Recurrent Unit (Geçitli Tekrarlayan Birim)

## **SEMBOLLER/SYMBOLS**

**PM<sub>2.5</sub>** : 2.5 mikrondan küçük partikül madde

**PM<sub>10</sub>** : 10 mikrondan küçük partikül madde

**SO<sub>2</sub>** : Kükürt dioksit

**NO<sub>x</sub>** : Azot oksitleri (NO + NO<sub>2</sub>)

**CO** : Karbon monoksit

**O<sub>3</sub>** : Ozon

**T** : Sıcaklık (°C)

**RH** : Bağıl nem (%)

**WS** : Rüzgar hızı (m/s)

## ŞEKİL LİSTESİ

	Sayfa
Şekil 1 LSTM Model Performansı .....	18
Şekil 2 LSTM Model Gerçek – Tahmin Grafiği.....	18
Şekil 3 LSTM Model Tahmin vs Gerçek Dağılımı (Scatter Plot).....	18
Şekil 4 LSTM Model ile Gelecek 2 Yıl için PM10 Tahmin Grafiği .....	21
Şekil 5 Transformer Model Performansı .....	23
Şekil 6 Transformer Model Gerçek – Tahmin Grafiği .....	23
Şekil 7 Transformer Model Tahmin vs Gerçek Dağılımı (Scatter Plot).....	24
Şekil 8 Transformer Model ile Gelecek 2 Yıl için PM10 Tahmin Grafiği.....	24
Şekil 9 GRU Model Performansı.....	26
Şekil 10 GRU Model Gerçek – Tahmin Grafiği.....	26
Şekil 11 GRU Model Tahmin vs Gerçek Dağılımı (Scatter Plot) .....	27
Şekil 12 GRU Model ile Gelecek 2 Yıl için PM10 Tahmin Grafiği .....	27
Şekil 13 Performans Metrikleri Tablosu.....	28



## ÖZET

Bu çalışmada, İstanbul'un hava kalitesinin tahmini amacıyla Uzun Kısa Süreli Bellek (LSTM), Geçitli Tekrarlayan Birimler (GRU) ve Transformer tabanlı derin öğrenme yöntemleri kullanılmaktadır. Projede, PM10, SO<sub>2</sub>, NO<sub>x</sub> gibi hava kirliliği parametreleri ile sıcaklık, nem ve rüzgar hızı gibi meteorolojik değişkenler temel alınarak tahmin modelleri geliştirilmektedir. Başakşehir İstasyonu'na ait veriler, T.C. Çevre, Şehircilik ve İklim Değişikliği Bakanlığı ile Meteoroloji Genel Müdürlüğü'nün Meteorolojik Veri Bilgi ve Sunum ve Satış Sistemi'ne bağlı <https://mevbis.mgm.gov.tr/> web adresinden satın alınmıştır ve eksik ya da hatalı değerlerden arındırılarak analiz için uygun hale getirilmektedir. Zamansal ve mekansal veri bütünlüğü sağlamak amacıyla veri birleştirme (data fusion) yöntemi uygulanmakta ve sensörler arası mekansal korelasyon dikkate alınmaktadır. Derin öğrenme modelleri, Google Colab ortamında eğitilmekte ve performansları MAE, RMSE, R<sup>2</sup> gibi metriklerle değerlendirilmektedir. Çalışmanın sonunda elde edilen model sonuçları karşılaştırılarak en başarılı tahmin yaklaşımı belirlenmekte, İstanbul'un hava kalitesi üzerindeki etkili faktörler analiz edilmektedir. Bu proje, hava kalitesi tahmininde doğruluk ve güvenilirliği artırmayı hedefleyerek çevresel analizlere katkı sağlamayı amaçlamaktadır.

## ABSTRACT

In this study, deep learning methods based on Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), and Transformer architectures are utilized to forecast the air quality in Istanbul. The project develops prediction models using air pollution parameters such as PM10, SO<sub>2</sub>, and NO<sub>x</sub>, along with meteorological variables including temperature, humidity, and wind speed. The data were purchased from <https://mevbis.mgm.gov.tr/>, the Meteorological Data Information, Presentation and Sales System affiliated with the Ministry of Environment, Urbanization and Climate Change of the Republic of Türkiye and the Turkish State Meteorological Service. The data are preprocessed by eliminating missing or erroneous values to ensure suitability for analysis. To maintain temporal and spatial data integrity, a data fusion approach is applied, taking into account the spatial correlation among sensors. The deep learning models are trained in the Google Colab environment, and their performances are evaluated using metrics such as MAE, RMSE, and R<sup>2</sup>. At the end of the study, the results of the models are compared to determine the most effective prediction approach, and the influential factors affecting Istanbul's air quality are analyzed. This project aims to enhance the accuracy and reliability of air quality forecasting and contribute to environmental analyses.

## 1. GİRİŞ

Hava kirliliği, günümüzde özellikle büyük şehirlerde yaşayan insan sağlığını tehdit eden önemli çevresel problemlerden biri haline gelmiştir. Dünya Sağlık Örgütü (WHO) verilerine göre, hava kirliliği nedeniyle her yıl milyonlarca insan erken yaşta hayatını kaybetmektedir. Gelişen teknoloji ve artan sanayileşme ile birlikte hava kalitesinin takibi ve geleceğe yönelik tahmin edilmesi, çevre politikalarının geliştirilmesi ve halk sağlığının korunması açısından büyük önem arz etmektedir. Özellikle İstanbul gibi yoğun nüfuslu, trafik ve sanayi açısından hareketli şehirlerde hava kalitesinin izlenmesi ve tahmin edilmesi, sağlık uyarı sistemlerinin geliştirilmesine ve çevresel önlemlerin zamanında alınmasına olanak sağlar.

Hava kalitesi, çeşitli kirletici parametrelerle ölçülmektedir. Bu çalışmada odaklanılan başlıca kirleticiler; PM<sub>2.5</sub> (2.5 mikrondan küçük partikül maddeler), PM<sub>10</sub> (10 mikrona kadar olan partikül maddeler), SO<sub>2</sub> (kükürt dioksit) ve NO<sub>x</sub> (azot oksitler - NO ve NO<sub>2</sub>) gibi gazlardır. PM<sub>2.5</sub> ve PM<sub>10</sub>, solunum yolu hastalıklarına neden olabilecek düzeyde küçük partiküller olup, uzun süreli maruz kalınması durumunda ciddi sağlık problemleri yaratabilmektedir. SO<sub>2</sub> genellikle fosil yakıtların yanması sonucu ortaya çıkar ve solunum yollarını tahriş edici özelliktedir. NO<sub>x</sub> ise motorlu taşıtlardan ve endüstriyel faaliyetlerden salınan azot oksitlerini temsil eder ve hem hava kalitesi hem de iklim değişikliği açısından kritik öneme sahiptir.

Geleneksel istatistiksel yöntemler ve klasik makine öğrenmesi modelleri, hava kalitesi tahmini konusunda çeşitli çalışmalar yürütmüş olsa da, bu yöntemler genellikle verilerin mekansal ve zamansal karmaşıklığını yeterince modelleyememektedir. Oysa ki hava kalitesi; zamanla değişen, farklı bölgeler arasında etkileşim gösteren, çok boyutlu ve dinamik bir yapıya sahiptir. Bu nedenle, daha derin ve bağlamsal ilişkileri modelleyebilen yapay zeka temelli yöntemlerin kullanılması kaçınılmaz hale gelmiştir.

Bu projede, İstanbul'un hava kalitesinin daha doğru ve güvenilir şekilde tahmin edilmesi amacıyla Uzun Kısa Süreli Bellek (LSTM), Geçitli Tekrarlayan Birimler (GRU) ve Transformer tabanlı derin öğrenme yöntemleri kullanılmaktadır. Hava kalitesi parametreleri ile meteorolojik veriler arasındaki zamansal ve mekansal ilişkiler, grafik yapılar üzerinden modellenerek tahminleme süreci optimize edilmektedir. Bu sayede hem farklı bölgeler arasında bilgi transferi sağlanmakta hem de zaman serisi dinamikleri etkili şekilde öğrenilmektedir. Proje, hava kalitesi tahmininde doğruluğu artırmayı ve çevresel analizlerin kalitesine katkıda bulunmayı hedeflemektedir.

### 1.1. Bitirme Projesinin Amacı

Bu projenin temel amacı, İstanbul'daki hava kalitesini daha doğru ve güvenilir bir şekilde tahmin edebilmek için gelişmiş derin öğrenme yöntemlerini kullanarak etkili bir tahmin sistemi geliştirmektir. Bu doğrultuda, Uzun Kısa Süreli Bellek (LSTM), Geçitli Tekrarlayan Birimler (GRU) ve Transformer tabanlı yapılar gibi güncel derin öğrenme modelleri kullanılacaktır. Projede, hava kirliliği parametreleri (PM10, SO<sub>2</sub>, NO<sub>x</sub> vb.) ile meteorolojik değişkenler (sıcaklık, nem, rüzgar hızı vb.) arasında zamansal ve mekansal ilişkilerin modellenmesi hedeflenmektedir.

Geleneksel tahmin yöntemleri, çoğunlukla verilerin doğrusal yapılarına ve lokal özelliklerine odaklanmakta, bu da hava kalitesi gibi çok boyutlu, etkileşimli sistemlerin doğru şekilde modellenmesini zorlaştırmaktadır. LSTM ve GRU gibi sıralı derin öğrenme yöntemleri, zamansal bağımlılıkları modellemede etkilidir. Bu modeller, uygun veri ön işleme ve istasyonlar arası etkileşimi yansıtan veri temsilleriyle birlikte kullanıldığında, şehir genelindeki hava ölçüm istasyonlarının çıktılarındaki örüntüleri anlamada katkı sağlayabilir. Bunun yanında Transformer tabanlı modeller, zaman serisi verilerindeki uzun vadeli bağımlılıkları öğrenme konusunda üstün performans göstermektedir. Bu üç farklı yaklaşımın bir arada incelenmesi, tahmin doğruluğunu artırmakla kalmayıp, hangi yöntemlerin hangi koşullarda daha etkili çalıştığına dair de önemli bilgiler sağlayacaktır.

Hava kalitesinin doğru bir şekilde tahmin edilmesi; bireylerin günlük yaşam planlamasından, belediyelerin çevresel politikalarına kadar geniş bir etki alanına sahiptir. Bu doğrultuda geliştirilecek olan model, İstanbul'daki hava kirliliği ile ilgili risklerin önceden tespit edilmesini sağlayarak, kamu sağlığının korunmasına katkı sunabilir. Ayrıca, bu çalışma ile geliştirilen tahmin modelleri, farklı şehirlerde ya da ülkelerde benzer uygulamalar için de örnek teşkil edebilir.

### 1.2. Literatür Taraması

Ece Yağmur tarafından 2022 yılında gerçekleştirilen çalışmada [1], İstanbul Beşiktaş ilçesine ait 2020 yılına yönelik saatlik hava kalitesi, meteorolojik ve trafik verileri kullanılarak PM2.5 konsantrasyonlarının tahmini amaçlanmıştır. İstanbul Büyükşehir Belediyesi'nin açık veri portalından alınan veriler; PM10, NO<sub>x</sub>, SO<sub>2</sub> gibi kirleticiler, sıcaklık, nem, rüzgar hızı gibi meteorolojik değişkenler ve ortalama trafik hızı ile araç sayısı gibi parametreleri içermektedir. Verilerin işlenmesi Python programlama dili kullanılarak Jupyter Notebook ortamında gerçekleştirilmiş, Pandas, Scikit-learn, Seaborn, Matplotlib ve TensorFlow (Keras)

kütüphanelerinden yararlanılmıştır. Veri temizleme, normalizasyon ve eksik veri doldurma işlemleri (SARIMA ve K-En Yakın Komşu) sonrasında yedi farklı model kurgulanmış; Çoklu Doğrusal Regresyon (ÇDR), Rassal Orman (RO), Destek Vektör Makineleri (DVM) ve Yapay Sinir Ağları (YSA) algoritmaları karşılaştırılmıştır. Tüm modeller  $R^2$ , MAE, MSE ve RMSE performans metrikleri ile değerlendirilmiş, en iyi sonuç RO algoritması ve PM10'un dahil edildiği Model 6 ile elde edilmiştir ( $R^2 \approx 0.76$ ). Bu model, PM2.5 seviyelerinin tahmini için en başarılı performansı göstermiştir ve zaman, meteoroloji ve trafik faktörlerinin entegre edilmesinin önemini vurgulamıştır.

Nesibe Manav Mutlu ve Eyüp Burak Atahanlı tarafından 2024 yılında gerçekleştirilen çalışmada [2], İstanbul'un farklı ilçelerine ait 2010–2023 yıllarını kapsayan hava kalitesi verileri kullanılarak geçmiş eğilimler analiz edilmiş ve 2023–2025 yıllarına yönelik tahminler üretilmiştir. İstanbul Büyükşehir Belediyesi'nin açık API servislerinden alınan yaklaşık 430 MB büyüklüğündeki veri, Python programlama dili ve Google Colab ortamında işlenmiş; veri hazırlama, analiz ve modelleme adımlarında Pandas, JSON, Prophet ve görselleştirme kütüphaneleri kullanılmıştır. AQI (Air Quality Index) değerlerine odaklanan çalışmada CRISP-DM metodolojisi izlenmiş, veriler istasyon bazlı ayrılarak her biri için ayrı Prophet modelleri kurulmuştur. Tahminler mevsimsel, yıllık ve saatlik düzeyde analiz edilmiştir. Modelin başarı değerlendirmesi MAPE (Ortalama Mutlak Yüzde Hata) metriğiyle yapılmış, genel ortalama MAPE değeri 0.316 olarak bulunmuştur. Bazı istasyonlarda %20 altı hata oranı görülürken, Kandilli örneğinde olduğu gibi %100'ün üzerinde hatalar da gözlemlenmiştir. 2023-2025 dönemine yönelik tahminlerde kirlilikte genel bir artış trendi görülmemiş; ancak bazı ilçeler (ör. Sultangazi, Sarıyer, Üsküdar) için dönemsel yükseliş öngörülmüştür. Çalışma, karar vericiler için bölgesel politikaların oluşturulmasına katkı sağlamayı ve sürdürülebilir çevre planlamasına veri desteği sunmayı amaçlamaktadır.

Sibel Ünalı ve Nesibe Yalçın tarafından 2022 yılında gerçekleştirilen çalışmada [3], İstanbul'un Başakşehir ilçesinde 2016–2021 yılları arasında ölçülen hava kirleticilerine (PM10, CO, SO<sub>2</sub>, NO<sub>2</sub>, O<sub>3</sub>) ait günlük ortalama konsantrasyonlar ile meteorolojik parametreler (rüzgar yönü, hızı, basınç) kullanılarak hava kirliliği tahmini yapılmıştır. Geliştirilen tahmin modelleri Python programlama diliyle, NumPy, SciPy, Scikit-Learn ve Matplotlib kütüphaneleri yardımıyla oluşturulmuştur. Kullanılan makine öğrenmesi algoritmaları arasında Çoklu Doğrusal Regresyon (MLR), Destek Vektör Makineleri (SVM), K-En Yakın Komşu (KNN), Karar Ağaçları, Rastgele Orman (RF) ve Çok Katmanlı Algılayıcı Sinir Ağı (MLPNN) yer almaktadır. Veri seti 2.031 örnekten oluşmakta, %75 eğitim ve %25 test olarak ayrılmıştır. Performans değerlendirmesi  $R^2$ , MAE, RMSE, MSE ve

MAPE metrikleriyle yapılmış, genel olarak PM10, CO, SO2 ve O3 tahmininde en iyi sonuçlar RF ile elde edilirken, NO2 tahmininde en başarılı model MLR olmuştur (NO2 için  $R^2=0.999$ ). En düşük performans ise KNN yöntemiyle NO2 tahmininde gözlenmiştir. Araştırma, hava kirliliğinin şehir bazlı modellenmesinde makine öğrenmesinin güçlü bir araç olduğunu ortaya koymakta ve ileride meteorolojik ya da trafik verileri ile birleştirilerek daha yüksek doğruluklu modellerin geliştirilebileceğini önermektedir.

Mustafa Akgün ve Necla Barlık tarafından 2023 yılında gerçekleştirilen çalışmada [4], Türkiye'nin Ardahan, Kars, Erzurum ve Iğdır illerine ait 2019–2021 yılları arasındaki ısınma periyotlarında (15 Eylül – 15 Mayıs) ölçülen Hava Kalitesi İndeksi (HKİ) ile meteorolojik parametreler (sıcaklık, yağış, bulutluluk, basınç, nem, rüzgâr hızı) kullanılarak HKİ tahmini yapılmıştır. Veriler Ulusal Hava Kalitesi İzleme Ağı ve Meteoroloji Genel Müdürlüğü'nden sağlanmış, analizler RapidMiner 9.10 platformunda gerçekleştirilmiştir. İlk olarak çoklu doğrusal regresyon analizi uygulanmış ve HKİ'nin %41,8 oranında tahmin edilebildiği görülmüştür. Ardından makine öğrenmesi algoritmaları (Naive Bayes, Karar Ağacı, Derin Öğrenme, k-NN, Rastgele Orman) ile sınıflandırma ve tahminleme yapılmış; en yüksek doğruluk oranı %74,89 ile Rastgele Orman algoritmasına ait olmuştur. Derin öğrenme ve k-NN algoritmaları sırasıyla %73,05 ve %73,35 doğruluk sağlamıştır. Çalışma, klasik istatistiksel yöntemlere kıyasla makine öğrenmesinin HKİ tahmininde daha etkili olduğunu ortaya koymakta ve meteorolojik verilerle desteklenmiş yapay zeka temelli yaklaşımların uygulanabilirliğini vurgulamaktadır.

Hüseyin Özel, Pınar Cihan, H. Kurtuluş Özcan, Serdar Aydın ve Asude Hanedar tarafından 2021 yılında gerçekleştirilen çalışmada [5], Tekirdağ Çerkezköy Organize Sanayi Bölgesi'nde yer alan yoğun sanayi faaliyetlerinin hava kalitesi üzerindeki uzun vadeli etkileri incelenmiştir. Marmara Temiz Hava Merkezi'ne bağlı Çerkezköy hava kalitesi izleme istasyonundan elde edilen 2016–2020 yıllarına ait günlük Hava Kalitesi İndeksi (HKİ) verileri ile SO<sub>2</sub>, NO<sub>2</sub>, O<sub>3</sub>, CO, PM10 ve PM2.5 gibi kirleticilerin konsantrasyonları değerlendirilmiştir. Ölçümler sonucunda HKİ'nin çoğunlukla “orta” ve “iyi” kategorilerde olduğu, 2019–2020 yıllarında 2016–2017'ye göre iyileşme gösterdiği tespit edilmiştir. Ayrıca COVID-19 pandemisi döneminde (Nisan-Mayıs 2020), sanayi faaliyetlerindeki azalmaya paralel olarak elektrik ve doğalgaz tüketiminde düşüş yaşanmış; bu da hava kirliliği parametrelerinde ciddi oranda iyileşmeye neden olmuştur (örneğin SO<sub>2</sub>'de %352, PM10'da %33 düşüş). Analizler, hava kirliliğinin özellikle PM10 ve PM2.5 konsantrasyonlarından kaynaklandığını göstermiştir. Araştırma, HKİ'nin şehir ölçeğinde çevresel durumun izlenmesi ve politika geliştirme süreçlerinde etkin bir gösterge olduğunu vurgulamaktadır.

Ahmet Sel tarafından 2022 yılında gerçekleştirilen çalışmada [6], Türkiye’deki iller hava kalitesi bakımından sınıflandırılmış ve PM10 ile SO<sub>2</sub> değerleri kullanılarak 2015–2020 yıllarına ait günlük ortalama veriler analiz edilmiştir. Çevre ve Şehircilik Bakanlığı’nın izleme istasyonlarından alınan veriler SPSS yazılımı ile doğrusal enterpolasyon yöntemiyle temizlenmiş; ideal küme sayısı Silhouette İndeksi ile beş olarak belirlenmiştir. Ardından K-Ortalama (K-Means) kümeleme yöntemi WEKA yazılımı üzerinde uygulanmıştır. Sonuçta Türkiye, hava kalitesi açısından beş kümeye ayrılmış ve bu kümeler “çok iyi-iyi”, “az iyi-iyi”, “çok orta-iyi”, “orta orta-iyi” ve “az orta-iyi” şeklinde etiketlenmiştir. İstanbul, Antalya, Konya ve Sakarya gibi iller “çok orta-iyi” kümesinde yer alırken; hava kalitesinin en düşük olduğu “az orta-iyi” kümesinde Mersin, Kahramanmaraş, Malatya, Şanlıurfa ve Diyarbakır gibi şehirler yer almıştır. Hava Kalitesi İndeksi hesaplamasında PM10 değerine 2 kat ağırlık verilmiş, böylece partikül madde etkisinin sağlık üzerindeki önemi vurgulanmıştır. Çalışma, bölgesel çevre politikalarının geliştirilmesi ve hava kalitesinin iyileştirilmesine yönelik öneriler sunması açısından önemli bir kaynak niteliği taşımaktadır.

Burak Kotan ve Arzu Erener tarafından 2023 yılında gerçekleştirilen çalışmada [7], Kocaeli ili için 2008–2019 yılları arasındaki PM10 ve SO<sub>2</sub> kirletici konsantrasyonlarının 1 yıl önceden sezonsal ortalamalar şeklinde tahmin edilmesi amaçlanmış ve bu amaçla hem çoklu doğrusal regresyon (ÇDR) hem de ileri beslemeli yapay sinir ağı (YSA) modelleri geliştirilmiştir. Tahmin modelleri oluşturulurken sadece meteorolojik ve hava kirliliği verileri değil, coğrafi bilgi sistemleri (CBS) aracılığıyla elde edilen mekânsal (bina izi, sanayi alanları, yol ağı), demografik (nüfus) ve topografik (sayısal yükseklik modeli - SYM) veriler de kullanılmıştır. Veriler SPSS ve MATLAB R2021a ortamında analiz edilmiştir. ÇDR modellerinde “stepwise” metodu kullanılırken, YSA modelleri 42 değişkenle kurulmuş ve çok sayıda transfer fonksiyonu ile farklı yapılandırmalar denenmiştir. Model performansları, ortalama mutlak yüzde hata (OMYH) gibi çeşitli metriklerle değerlendirilmiştir. PM10 tahminlerinde YSA modelleri, özellikle ilkbahar mevsiminde %4.91 OMYH ile “çok iyi” sınıfına girmiştir. SO<sub>2</sub> için YSA, ÇDR’ye göre genel olarak daha iyi sonuçlar üretmiş; yaz mevsimi hariç diğer mevsimlerde “kabul edilebilir” doğruluk seviyelerine ulaşmıştır. Bu çalışma, mekânsal verilerin ve CBS destekli modellemelerin hava kirliliği tahmininde etkin biçimde kullanılabileceğini gösterirken, uzun vadeli (1 yıl sonrası) sezonsal tahminlerin çevresel planlama açısından kritik bir destek sağlayabileceğini ortaya koymuştur.

Tanisha Madan, Shrddha Sagar ve Deepali Virmani tarafından 2020 yılında gerçekleştirilen çalışmada [8], hava kalitesi tahmini üzerine yapılmış çok sayıda çalışmanın karşılaştırmalı analizi sunulmuştur. Makine öğrenmesi algoritmalarının hava kirleticilerini (PM2.5, PM10,

SO<sub>2</sub>, NO<sub>2</sub>, CO, O<sub>3</sub> vb.) ve meteorolojik parametreleri (sıcaklık, rüzgâr hızı, nem, basınç) dikkate alarak Hava Kalitesi İndeksi (AQI) tahmininde nasıl kullanıldıkları detaylı şekilde incelenmiştir. Literatürde kullanılan modeller arasında Doğrusal Regresyon, Karar Ağaçları, Rastgele Orman, Destek Vektör Makineleri (SVM), Yapay Sinir Ağları (ANN), Derin İnanç Ağı (DBN), Wavelet Sinir Ağı, Gated RNN, XGBoost, Extreme Learning Machine (ELM), Multilayer Perceptron (MLP) ve Hibrit Boosting yaklaşımları yer almaktadır. Farklı veri kümeleri (örneğin: Kaggle, sensör verisi, ulusal gözlem sistemleri) ve bölgeler (Hindistan, Çin, Ürdün, ABD, Kolombiya vb.) üzerinde test edilen bu algoritmaların başarı oranları %70 ile %99 arasında değişmektedir. Genel sonuçlara göre, Boosting tabanlı yöntemler ve sinir ağı modelleri (özellikle MLP ve Hibrit modeller) yüksek doğruluk oranlarıyla öne çıkmıştır. Öte yandan, sensör kalitesi, verinin eksikliği veya mevsimsel değişkenlikler gibi faktörlerin tahmin performansını etkilediği vurgulanmıştır. Bu çalışma, algoritma seçiminin bölge, kirletici tipi ve veri kalitesine göre değişebileceğini ve hava kalitesi tahmininde evrensel tek bir modelden çok, probleme özgü yaklaşım gereksinimini ortaya koymaktadır.

Yeşim Dokuz, Aslı Bozdağ ve Öznur Begüm Gökçek tarafından 2020 yılında gerçekleştirilen çalışmada [9], hava kirliliğinin tahmini ve mekânsal dağılımının analizinde kullanılan makine öğrenmesi yöntemleri detaylı şekilde ele alınmıştır. Öncelikle PM<sub>10</sub>, PM<sub>2.5</sub>, NO<sub>x</sub>, SO<sub>2</sub>, CO, O<sub>3</sub> gibi hava kirletici parametrelerin çevresel etkileri ve bu parametrelerin düzenli izlenmesinin önemi vurgulanmıştır. Ardından, hem geleneksel makine öğrenmesi algoritmaları (LASSO, SVM, Random Forest, k-NN) hem de derin öğrenme modelleri (DNN, CNN, RNN, LSTM) tanıtılmış; bu yöntemlerin veri hacmi, uygulama alanı, meteorolojik değişkenler ve zaman serisi uzunluğu gibi faktörlerle nasıl ilişkilendirileceği anlatılmıştır. Literatürdeki örnek çalışmalarda, özellikle derin öğrenme tabanlı CNN-LSTM, Boosted Trees, ve GW-GBM gibi modellerin farklı coğrafi bölgelerde yüksek doğruluk oranları ile hava kirliliği tahmini gerçekleştirdiği aktarılmıştır. Ayrıca çalışmada, tahmin sonuçlarının Coğrafi Bilgi Sistemleri (CBS) ile mekânsal olarak haritalandırılmasının, şehir planlaması ve çevresel strateji geliştirme açısından büyük önem taşıdığı vurgulanmıştır. Bu kapsamda enterpolasyon yöntemleri (Kriging, IDW) ile tahmin değerlerinin mekânsal görselleştirilmesine yönelik örnekler sunulmuştur. Çalışma, hava kalitesi modelleme sürecinde değişken seçimi, yöntem uyumu ve bölgesel bağlamın dikkate alınması gerektiğini belirterek, sürdürülebilir çevresel yönetim için yol gösterici bir kaynak niteliği taşımaktadır.

Büşra Duygu Çelik ve Nursal Arıcı tarafından 2021 yılında gerçekleştirilen çalışmada [10], Covid-19 salgını öncesi (2019) ve salgın sürecinde (2020) Zonguldak ili Çatalağzı bölgesine ait hava kirliliği verileri kullanılarak, makine öğrenmesi algoritmaları ile Hava Kalitesi

İndeksi (HKİ) tahmin edilmiştir. Veriler Çevre ve Şehircilik Bakanlığı'na bağlı hava izleme istasyonundan alınmış; PM10, SO2, NO2, CO ve O3 kirleticileri analiz edilmiştir. Veri ön işleme sürecinde eksik ve aykırı değerler temizlenmiş, veriler min-max normalizasyon ile ölçeklendirilmiştir. Python ortamında Scikit-learn kütüphanesi ile geliştirilen modelde, Karar Ağacı (DT), Rastgele Orman (RF), K-En Yakın Komşu (KNN), Destek Vektör Makineleri (SVM) ve Naive Bayes (NB) algoritmaları karşılaştırılmıştır. Veriler farklı oranlarda eğitim-test ayrımıyla denenmiş ve nihai olarak %67 eğitim - %33 test bölümüyle en yüksek doğruluk elde edilmiştir. 2019 ve 2020 verileriyle yapılan tahminlerde en iyi performans, %99,9 doğruluk ve RMSE=0.016–0.021 ile Karar Ağacı algoritmasında elde edilmiştir. En düşük başarı ise Naive Bayes ile gözlenmiştir. Pandemi önlemleri ile birlikte 2020 yılında kirleticiler seviyelerinde genel bir azalma görülmüş, normalleşme süreciyle birlikte yeniden artış tespit edilmiştir. Çalışma, hava kalitesinin hem sağlık hem de salgın yönetimi açısından kritik olduğunu, makine öğrenmesi modellerinin bu alanda yüksek doğrulukla kullanılabileceğini göstermektedir.

Manuel Méndez, Mercedes G. Merayo ve Manuel Núñez tarafından 2023 yılında gerçekleştirilen çalışmada [11], 2011–2021 yılları arasında yayımlanmış 155 akademik makale incelenerek hava kalitesi tahmini için kullanılan makine öğrenmesi (ML) ve derin öğrenme (DL) algoritmaları karşılaştırmalı olarak analiz edilmiştir. Araştırma kapsamında öne çıkan algoritmalar; SVR, Random Forest, Doğrusal Regresyon, ARIMA gibi klasik ML yöntemleri ile MLP, LSTM, CNN, GRU, Encoder-Decoder gibi DL yaklaşımlarıdır. Çalışma; kullanılan algoritmalar, tahmin edilen parametreler (AQI, PM2.5, NO2 vb.), bağımsız değişkenler (hava durumu, kirlilik değerleri, zaman faktörleri), coğrafi dağılım ve performans metrikleri ( $R^2$ , MAE, RMSE, MAPE) açısından detaylı bir sınıflandırma sunmaktadır. Bulgulara göre, özellikle LSTM tabanlı modellerin zaman serisi verileri üzerinde yüksek doğruluk sağladığı, ancak Random Forest ve MLP'nin de güçlü rakipler olduğu gözlemlenmiştir. Hibrit modellerin (örneğin: CNN-LSTM, GRU-LR, SVR-LSTM) çoğunlukla tekil modellere göre daha iyi performans verdiği rapor edilmiştir. Coğrafi olarak ise çalışmaların büyük çoğunluğunun Çin, Hindistan ve bazı Avrupa ülkelerinde yoğunlaştığı, en sık tahmin edilen parametrenin PM2.5 olduğu tespit edilmiştir. Bu çalışma, hava kalitesi tahmin modelleri geliştirilirken hangi algoritma türlerinin, hangi veri türleri ve tahmin hedefleri için daha uygun olduğunu sistematik biçimde özetlemekte ve derin öğrenme temelli modellerin artan etkinliğini vurgulamaktadır.

Luo Zhang, Peng Liu, Lei Zhao, Guizhou Wang, Wangfeng Zhang ve Jianbo Liu tarafından 2022 yılında gerçekleştirilen çalışmada [12], PM2.5 tahmini amacıyla geliştirilmiş yarı



denetimli bir model olan EMD-BiLSTM (Empirical Mode Decomposition - Bidirectional Long Short-Term Memory) sunulmuştur. Model, sadece PM2.5 zaman serisi verilerini giriş olarak kullanmakta ve EMD ile frekans ve genlik bileşenlerini ayrıştırarak bu sinyal verilerinden öznitelik çıkarmaktadır. Ardından BiLSTM yapısı, zamansal ilişkileri öğrenmek için denetimli öğrenme aşamasında kullanılmıştır. Model, Çin'in Pekin kentindeki 13 çevresel izleme istasyonuna ait saatlik ve günlük PM2.5 verileriyle test edilmiştir. Saatlik tahminlerde  $RMSE=6.86 \mu g/m^3$ ,  $R^2=0.989$ , günlük tahminlerde ise  $RMSE=22.58 \mu g/m^3$ ,  $R^2=0.742$  değerleri elde edilerek standart LSTM ve BiLSTM modellerinden daha yüksek doğruluk sergilenmiştir. Ayrıca model, çok saatlik (3h, 6h, 12h) yeniden yapılandırılmış veri setleri ile de test edilmiş ve tüm zaman ölçeklerinde başarılı sonuçlar vermiştir. Yardımcı veri (örneğin meteorolojik) kullanmadan yüksek doğruluk sağlaması ve ani değişimlerde (trend kırılması) düşük hata oranıyla sonuç üretmesi, modelin öne çıkan yönleridir. Geliştirilen yöntem, az sayıda girdiyle sade bir yapıya sahip olmasına karşın yüksek kararlılık ve doğruluk sunmakta ve zaman serisi temelli hava kirliliği tahmininde güçlü bir yaklaşım olarak değerlendirilmektedir.

Tuğçe Pekdoğan tarafından 2024 yılında gerçekleştirilen çalışmada [13], yapay zekâ ve makine öğrenmesi tekniklerinin hava kalitesinin tahmini ve iyileştirilmesinde nasıl kullanılabileceği kapsamlı bir şekilde değerlendirilmiştir. Çeşitli yapay zekâ modelleri (Random Forest, SVM, ANN, CNN, RNN, GAN) üzerinden yapılan analizlerde; bu modellerin tahmin doğruluğu, uygulama alanları, avantajları ve dezavantajları karşılaştırılmıştır. Literatürde yer alan örnek çalışmalar üzerinden, farklı coğrafi bölgelerde (Türkiye, Hindistan, Kuveyt, Çin vb.) çeşitli hava kirliletiçi parametrelerin (PM2.5, SO2, NO2, CO vb.) tahmini yapılmış, özellikle CNN'in mekânsal analizlerde, RNN'in ise zaman serisi verilerinde yüksek doğruluk sunduğu ortaya konmuştur. Hibrit model kullanımı, eksik verilerin GAN ile üretilmesi ve gerçek zamanlı hava kalitesi izleme sistemlerinin entegrasyonu gibi yenilikçi yaklaşımlar vurgulanmıştır. Ayrıca çalışmada, şehir planlaması, trafik yönetimi, sanayi ve akıllı tarım alanlarında yapay zekâ tabanlı çözümlerin nasıl stratejik avantajlar sağladığı örneklenmiş ve gelecekteki araştırmalarda hibrit modellerin, uyarlanabilir algoritmaların ve veri kalitesinin rolünün daha da önemli hale geleceği ifade edilmiştir. Bu yönüyle çalışma, yapay zekânın hava kalitesi tahmini ve sürdürülebilir çevresel planlamadaki potansiyelini kapsamlı şekilde ortaya koymaktadır.

Ceren Ünal ve Gamze Özel tarafından 2023 yılında gerçekleştirilen çalışmada [14], 2018–2021 yılları arasında Ankara iline ait hava kirliliği parametreleri (PM10, SO2, NO, NO2, NOX, CO) ile meteorolojik değişkenler (sıcaklık, bağıl nem, rüzgâr hızı, rüzgâr yönü, hava

basıncı) arasındaki ilişkiler çoklu doğrusal regresyon analizi ile incelenmiştir. Veriler, Türkiye Cumhuriyeti Çevre ve Şehircilik Bakanlığı ile Meteoroloji Genel Müdürlüğü'nden temin edilmiştir. Analizler SPSS yazılımında gerçekleştirilmiş; Ankara'nın sekiz istasyonuna ait günlük veriler kullanılmıştır. Bulgulara göre, en yüksek açıklayıcılık CO için elde edilmiş (%32.5), bunu sırasıyla SO<sub>2</sub> (%16.2), NO (%14.0) ve PM<sub>10</sub> (%10.2) takip etmiştir. NO<sub>2</sub> ve NOX için açıklayıcılık oranları sırasıyla %3.6 ve %7.7 ile sınırlı kalmıştır. CO düzeyleri üzerinde en etkili değişkenler sıcaklık, rüzgâr yönü, rüzgâr hızı ve hava basıncı olurken; PM<sub>10</sub> ve SO<sub>2</sub> gibi diğer kirleticiler için özellikle sıcaklık, bağıl nem ve rüzgâr yönü anlamlı bulunmuştur. Genel olarak, meteorolojik faktörlerin hava kirleticileriyle çoğunlukla ters yönlü ve zayıf düzeyde ilişkili olduğu tespit edilmiştir. Sonuçlar, hava kirliliği tahmininde istatistiksel modellerin etkili olabileceğini gösterirken, aynı zamanda CO gibi kirleticilerin meteorolojik koşullardan daha yüksek oranda etkilendiğini ortaya koymuştur.

K. Kumar ve B. P. Pande tarafından 2023 yılında gerçekleştirilen çalışmada [15], Hindistan'ın 23 farklı kentinden 2015–2020 yılları arasında toplanan hava kirliliği verileri kullanılarak Air Quality Index (AQI) tahmini gerçekleştirilmiştir. 29.000'den fazla gözlem içeren veri setinde PM<sub>2.5</sub>, PM<sub>10</sub>, NO<sub>2</sub>, CO, SO<sub>2</sub>, O<sub>3</sub> gibi 12 kirletici parametre yer almaktadır. Veriler, eksik ve aykırı değerler temizlenerek normalize edilmiş; ardından korelasyon analizine dayalı öznitelik seçimi yapılmıştır. En yüksek korelasyon değerine sahip kirleticiler (PM<sub>10</sub>, PM<sub>2.5</sub>, CO vb.) tahmin modeline dahil edilmiştir. Dengesiz veri dağılımı SMOTE (Synthetic Minority Oversampling Technique) yöntemiyle çözülmüştür. Beş makine öğrenmesi modeli (KNN, Gaussian Naive Bayes, SVM, Random Forest, XGBoost) hem SMOTE öncesi hem sonrası senaryolarda test edilmiş; modeller Google Colab Pro ortamında Python (Scikit-learn, NumPy, Pandas) kütüphaneleriyle eğitilmiştir. Test verisi üzerinde en iyi sonuç XGBoost ile elde edilmiş (Doğruluk: %90, R<sup>2</sup>=0.834), en düşük başarı ise SVM modeliyle kaydedilmiştir. Özellikle 2020 yılında pandemi kaynaklı insan etkinliklerinin durması sonucu tüm kirletici seviyelerinde ciddi düşüş gözlenmiştir. Bu çalışma, eksiksiz veri ön işleme, istatistiksel analiz, model karşılaştırması ve sonuçların yorumlanması yönünden kapsamlı olup, XGBoost'un AQI tahminindeki yüksek başarısını güçlü biçimde ortaya koymaktadır.

Alka Pant, Sanjay Sharma ve Kamal Pant tarafından 2023 yılında gerçekleştirilen çalışmada [16], Hindistan'ın Uttarakhand eyaletindeki çeşitli şehirlerden (Dehradun, Rishikesh, Haridwar, Haldwani, Rudrapur, Kashipur) elde edilen beş yıllık (aylık) hava kirliliği verileri kullanılarak Air Quality Index (AQI) tahmini gerçekleştirilmiş ve altı farklı makine öğrenmesi algoritması (Logistic Regression, Random Forest, Support Vector Machine, Naive Bayes, K-Nearest Neighbors, Decision Tree) karşılaştırılmıştır. Veriler PM<sub>10</sub>, PM<sub>2.5</sub>, SO<sub>2</sub> ve NO<sub>2</sub>

kirleticilerini içermekte olup eksik veriler ortalama deęerle doldurulmuř, ardından %65 eęitim ve %35 test olarak ayrılmıřtır. Tm modeller Python (Jupyter Notebook) ortamında eęitilmiř; bařarıları doęruluk, hassasiyet, geri aęırma ve F1 skoru gibi metriklerle deęerlendirilmiřtir. Elde edilen sonulara gre, Random Forest ve Decision Tree algoritmaları %99 doęruluk ile en yksek bařarıyı saęlamıřtır. K-Nearest Neighbors algoritması da %98 doęruluk ile ne ıkarken, en dřk performans %84 doęrulukla Naive Bayes algoritmasında gzlemlenmiřtir. Random Forest, kararlılıęı nedeniyle AQI tahmini iin en uygun algoritma olarak seilmiřtir. Ayrıca PM10 ve SO<sub>2</sub> arasında gl bir pozitif iliřki tespit edilmiř; festival dnemlerinde kirlilikte artıř, yaęıřlı mevsimlerde ise dřř gzlemlenmiřtir. Bu alıřma, hem yntem karřılařtırması hem de AQI tahmininin pratik uygulaması aısından nemli bulgular sunmakta ve karar vericilere politika geliřtirme konusunda katkı saęlamaktadır.

Ling Chen, Jiahui Xu, Binqing Wu, Yuntao Qian, Zhenhong Du, Yansheng Li ve Yongjun Zhang tarafından 2021 yılında gerekleřtirilen alıřmada [17], lke genelindeki řehirler iin hava kalitesi tahmini amacıyla geliřtirilen Group-Aware Graph Neural Network (GAGNN) modeli sunulmuřtur. Model, hem coęrafi olarak yakın řehirler arasındaki meknsal baęımlılıkları hem de coęrafi olarak uzak fakat iliřkili řehirler arasındaki gizli (latent) baęımlılıkları ğrenebilen hiyerarřik bir GNN mimarisi zerine kuruludur. GAGNN, řehirler arası iliřkileri modellemek iin bir řehir grafięi ve řehir grubu grafięi oluřturur; řehir gruplarını dinamik olarak ğrenmek iin farklılařtırılabilir bir gruplama aęı kullanır. Modelin encoder kısmı, Transformer tabanlı oklu bařlı self-attention katmanları ile zaman serisi zelliklerini ıkartırken; decoder, hem řehir hem de grup iliřkilerini kullanarak AQI deęerlerini tahmin eder. in’e ait 209 řehirden 850 gnlk AQI ve meteorolojik veriyle eęitilen model, gemiř 24 saati kullanarak gelecek 6 saatin AQI deęerlerini tahmin etmektedir. Yapılan karřılařtırmalı deneylerde, GAGNN; LSTM, XGBoost, GC-LSTM, ST-UNet, DeeperGCN ve SHARE gibi SOTA (state-of-the-art) modellere kıyasla daha dřk MAE ve RMSE deęerleri ile stn performans gstermiřtir (rneęin, 1 saatlik tahminde MAE: 5.56). Modelin PyTorch ve PyTorch Geometric ile geliřtirildięi ve tek bir GPU (RTX 2080Ti) zerinde alıřtırıldıęı belirtilmiřtir. Sonular, hiyerarřik GNN mimarilerinin geniř alansal tahminlerde gl avantajlar sunduęunu ve ğrenilebilir gruplama ile řehirler arası gizli etkileřimlerin bařarıyla yakalanabileceęini gstermektedir.

Pengfei Li, Tong Zhang ve Yantao Jin tarafından 2023 yılında gerekleřtirilen alıřmada [18], hava kalitesini zaman ve mekn boyutunda modelleyebilen yeniliki bir derin ğrenme modeli olan GCNInformer nerilmiřtir. Model, meknsal baęımlılıkları ğrenmek iin Graph Convolutional Network (GCN) katmanlarını, uzun sreli zaman baęımlılıklarını ğrenmek

için ise düşük hesaplama maliyetine sahip Informer katmanlarını birleştirir. Ayrıca meteorolojik veriler ve diğer hava kalitesi parametrelerinden düşük boyutlu özellik çıkarımı için çok katmanlı algılayıcı (MLP) katmanları da içermektedir. Model, Çin'in Pekin kentindeki 12 izleme istasyonundan alınan saatlik PM2.5, PM10, SO<sub>2</sub>, NO<sub>2</sub>, CO ve O<sub>3</sub> verileriyle eğitilmiş ve farklı zaman adımlarında (1, 8, 16, 24 saat) tahmin performansı ARIMA, SVR, LSTM, Transformer ve Informer gibi yöntemlerle karşılaştırılmıştır. Elde edilen sonuçlara göre, GCNInformer tüm görevlerde en düşük hata oranlarını (örneğin, 1 saatlik tahminde RMSE: 0.3367, MAE: 0.1784) vererek en başarılı model olmuştur. Ablasyon deneyleri, MLP ve GCN katmanlarının çıkarılmasıyla model performansında belirgin düşüş olduğunu göstermiştir. Modelin farklı kirleticiler (PM10, NO<sub>2</sub>, SO<sub>2</sub>, vb.) için de etkili şekilde genellenebildiği test edilmiştir. GCNInformer, hava kalitesi tahminine yönelik zamansal-mekânsal ilişkileri başarılı şekilde modelleyerek hem kamu sağlığı yönetimi hem de çevresel karar destek sistemleri için güçlü bir araç sunmaktadır.

Tüm bu çalışmalar, hava kalitesi tahmini alanında makine öğrenmesi ve derin öğrenme temelli yöntemlerin yaygın biçimde kullanıldığını ve bu yöntemlerin çeşitli çevresel faktörlerle entegre edilerek güçlü öngörüler sağlayabildiğini ortaya koymaktadır. Literatürde farklı bölgelerde yapılan çalışmalar, hava kirletici parametrelerin ve meteorolojik değişkenlerin birlikte değerlendirilmesinin model başarısını artırdığını göstermektedir. Bu kapsamda, projemiz İstanbul özelinde PM2.5, PM10, SO<sub>2</sub>, NO<sub>x</sub> gibi hava kirletici parametreler ile sıcaklık, nem, rüzgar hızı gibi meteorolojik veriler kullanılarak, LSTM, GRU ve Transformer tabanlı derin öğrenme modelleri ile yüksek doğrulukta hava kalitesi tahmini yapmayı amaçlamaktadır. Literatürde sıkça vurgulanan mekânsal ve zamansal ilişkilere odaklanan bu çalışma, veri birleştirme (data fusion) yöntemleriyle desteklenmiş yapısıyla hem model başarısını artırmayı hem de çevresel planlamalar için güvenilir bir karar destek mekanizması oluşturmayı hedeflemektedir. Proje sonuçlarının, özellikle büyük şehirlerdeki hava kalitesinin izlenmesi ve gelecekteki kirlilik eğilimlerinin öngörülmesi açısından önemli katkılar sağlayacağı düşünülmektedir.

## **2. MATERYAL VE YÖNTEM**

### **2.1. Veri Setinin Temin Edilmesi**

Proje kapsamında İstanbul'un hava kalitesinin tahminine yönelik veri toplama ve ön işleme çalışmaları gerçekleştirilmiştir. Bu doğrultuda hava kalitesi verileri ARŞ. GÖR. Damla Mengüş aracılığıyla sağlanmış olup, Başakşehir ilçesindeki ölçüm istasyonlarına ait günlük

değerleri içermektedir. Verilerde yer alan temel kirletici parametreler şunlardır: PM10, SO<sub>2</sub>, CO, NO<sub>2</sub>, NO<sub>x</sub>, NO, O<sub>3</sub>. Meteorolojik veriler ise aynı bölgeye ait olmak üzere T.C. Çevre, Şehircilik ve İklim Değişikliği Bakanlığı Meteoroloji Genel Müdürlüğü tarafından sağlanmış ve günlük ortalama sıcaklık, nem ve rüzgar hızı bilgilerini içermektedir.

Veri birleştirme aşamasında, farklı kaynaklardan alınan hava kalitesi ve meteorolojik veriler tarih bazında birleştirilmiş ve yalnızca ortak günlere ait kayıtlar korunmuştur. 2019 yılı Ocak ayından 2024 yılı Mart ayına kadar olan tüm günler için eksiksiz ve uyumlu bir zaman serisi oluşturulmuştur

## 2.2. Veri ön İşleme Adımları

Proje kapsamında elde edilen ham veri dosyasında, sayıların yazım biçimi ve karakter kullanımıyla ilgili bazı tutarsızlıklar tespit edilmiştir. Özellikle binlik ayırıcı olarak nokta, ondalık ayırıcı olarak virgöl kullanılması ve sayıların tırnak işaretleri içinde yer alması gibi durumlar verinin kullanılmasını zorlaştırmıştır. Bu nedenle, modelleme aşamasına geçmeden önce veriler üzerinde gerekli temizlik işlemleri yapılmıştır.

İlk olarak, veri dosyası satır satır okunmuş ve her satıra temizlik fonksiyonu uygulanmıştır. “1.076,82” gibi sayılarda önce binlik ayırıcı olan nokta kaldırılmış, ardından virgöl yerine nokta konularak sayı, olması gereken biçime getirilmiştir. Böylece bu değerler, yazılım ortamında sayısal olarak sorunsuz şekilde işlenebilecek hale gelmiştir. Aynı işlem virgüllü ondalık sayılar için de yapılmış, virgöl karakteri nokta ile değiştirilmiştir. Bu dönüşüm sayesinde sayılar tutarlı hale getirilmiş ve tüm sütunlar rahatlıkla sayısal veri tipine çevrilebilmiştir. Son olarak verilerde kalan tırnak işaretleri de temizlenmiş ve tüm satırlar işlenerek yeni bir dosyaya aktarılmıştır. Temizlik işlemleri tamamlandıktan sonra, düzenli ve hatasız veriler içeren yeni bir dosya (temizlenmis\_veri\_final.csv) oluşturulmuştur. Bu dosya bir sonraki adımda tekrar okunmuş ve ilk sütunda yer alan gereksiz indeks bilgileri çıkarılmıştır. Böylece analizde yalnızca anlamlı veri sütunları kalmıştır. Elde edilen yeni dosya (veri\_ilk\_sutunsuz.csv) sonraki işlemler için hazırlanmıştır. Devamında veride eksik olan bazı alanlar üzerinde çalışılmıştır. "-" sembolleri eksik değer olarak kabul edilerek bunlar NaN ile değiştirilmiştir. Sayısal olmayan “Tarih” sütunu veri kümesinden ayrılmış, sadece sayısal değerlere yönelik işlem yapılmıştır. Ardından bu sayısal alanlar uygun veri tipine çevrilmiş ve eksik değerleri tahmin etmek için komşuluk temelli bir yöntem olan KNN (K-Nearest Neighbors) algoritması kullanılmıştır. Burada her bir eksik değer, en yakın 3 günün verilerine göre tahmin edilerek doldurulmuştur. Eksik değerler tamamlandıktan sonra, sayısal

veriler daha düzenli görünsün diye üç ondalık basamakla yuvarlanmıştır. Son olarak tarih sütunu tekrar eklenmiş ve tüm sütunlar bir araya getirilerek son dosya oluşturulmuştur. Bu veri seti veri\_knn\_tamamlanmis\_yuvarlanmis.csv adıyla kaydedilmiştir.

```
# 1. Veriyi oku
df = pd.read_csv('veri_ilk_sutunsuz.csv', sep=',', encoding='utf-8')

# 2. "-" olan yerleri NaN yap
df.replace('-', np.nan, inplace=True)

# 3. Tarih sütununu ayır
tarih = df['Tarih']
df_numeric = df.drop(columns=['Tarih'])

# 4. Sayısal verilere çevir
df_numeric = df_numeric.astype(float)

# 5. KNN ile doldur (örneğin k=3)
imputer = KNNImputer(n_neighbors=3)
df_imputed = pd.DataFrame(imputer.fit_transform(df_numeric),
                           columns=df_numeric.columns)

# 6. Sayısal değerleri 3 ondalık basamağa yuvarla
df_imputed = df_imputed.round(3)

# 7. Tarihi geri ekle
df_final = pd.concat([tarih.reset_index(drop=True), df_imputed], axis=1)

# 8. Sonucu gör
print(df_final.head())

# 9. (İsteğe bağlı) Kaydet
df_final.to_csv('veri_knn_tamamlanmis_yuvarlanmis.csv', index=False)
```

Uygulanan bu işlemler sayesinde eksik veriler tamamlanmış, sayılar standart hale getirilmiş ve zaman bazlı tutarlılığı koruyan temiz bir veri kümesi elde edilmiştir. Bu haliyle veri, doğrudan model eğitimi ve analizlerde kullanılmaya hazır duruma gelmiştir.

### 2.3. LSTM İle PM10 Tahmini

Projede ilk olarak derin öğrenmeye dayalı LSTM (Long Short-Term Memory) modeli uygulanmıştır. LSTM'ler, zaman serisi verilerinde geçmiş bilgileri uzun süre hatırlayabilme kabiliyetiyle öne çıkar. Bu özellik, PM10 gibi zamana bağlı kirleticilerin tahmini için oldukça uygundur.

Modelin giriş parametreleri olarak PM10'un bir gün önceki değeri (PM10\_prev) ile birlikte

diğer kirletici gazlar (SO<sub>2</sub>, CO, NO<sub>2</sub>, NO<sub>x</sub>, NO, O<sub>3</sub>) ve meteorolojik değişkenler (sıcaklık, nem, rüzgar hızı) kullanılmıştır. Çıkış parametresi olarak yalnızca PM10 hedeflenmiştir.

Eğitim sürecinde 2019 – 2023 tarihleri arasında kalan veriler kullanılmış, 2024 yılına ait veriler ise test için ayrılmıştır. Veriler MinMaxScaler ile normalize edilmiş ve modelin zaman serisi formatına uygun şekilde yeniden şekillendirilmiştir. Model, erken durdurma (early stopping) stratejisiyle aşırı öğrenmenin önüne geçecek biçimde eğitilmiştir.

Model eğitiminde kullanılan kod blokları ve açıklamaları aşağıdaki gibidir:

### 2.3.1. Gerekli Kütüphanelerin Yüklenmesi

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_absolute_error, mean_squared_error,
r2_score
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.callbacks
import EarlyStoppingfrom tensorflow.keras.optimizers import Adam
```

Bu bölümde veri analizi (Pandas, NumPy), görselleştirme (Matplotlib), veri ölçekleme ve başarı metrikleri (scikit-learn), LSTM modeli oluşturma (Keras) için gerekli tüm kütüphaneler projeye dahil edilmiştir.

### 2.3.2. Verilerin Yüklenmesi ve Düzenlenmesi

```
df = pd.read_csv("veri_knn_tamamlanmis_yuvarlanmis.csv")
df["Tarih"] = pd.to_datetime(df["Tarih"])
df["PM10_prev"] = df["PM10"].shift(1)
df["PM10_next"] = df["PM10"].shift(-1)
df["Ay"] = df["Tarih"].dt.month
df["Gun"] = df["Tarih"].dt.day
df = df.dropna().reset_index(drop=True)
```

Modelleme sürecinde kullanılacak veri kümesi, "veri\_knn\_tamamlanmis\_yuvarlanmis.csv" adlı dosyadan Pandas kütüphanesi yardımıyla içe aktarılmıştır. Ardından, zaman bilgisi içeren "Tarih" sütunu datetime formatına dönüştürülerek veri setinin zamana bağlı işlemler için uygun hale getirilmesi sağlanmıştır. PM10 değişkeninin bir gün öncesine ait değeri PM10\_prev sütunu olarak eklenmiş ve böylece geçmiş veriye dayalı öngörüler yapılmasına imkân tanınmıştır. Ayrıca yalnızca eğitim sürecinde hedef olarak kullanılmak üzere, PM10'un bir gün sonrasını ifade eden PM10\_next sütunu oluşturulmuştur. Bu sütun, ileriye dönük

tahmin işlemlerinde kullanılmamakta, sadece modelin öğrenme aşamasında gerçek değeri temsil etmektedir. Mevsimsel etkileri modellemeye yardımcı olması amacıyla, “Tarih” sütunundan ay ve gün bilgileri çıkarılarak sırasıyla Ay ve Gun sütunları oluşturulmuştur. Son olarak, shift işlemleri sonucu oluşabilecek eksik veriler veri kümesinden dropna() komutu ile temizlenmiş ve indeksler sıfırlanarak analiz için eksiksiz, tutarlı bir zaman serisi verisi elde edilmiştir.

### 2.3.3. Giriş ve Hedef Sütunlarının Belirlenmesi

```
feature_columns = ["PM10_prev", "SO2", "CO", "NO2", "NOX", "NO", "O3",  
"Sıcaklık", "Nem", "Rüzgar Hızı"]  
target_column = "PM10"
```

Bu aşamada modelin giriş (bağımsız) ve hedef (bağımlı) değişkenleri tanımlanmıştır. feature\_columns adlı liste içerisinde, modele girdi olarak verilecek tüm özellikler belirlenmiştir. Bu özellikler arasında PM10’un bir gün önceki değeri olan PM10\_prev ve yalnızca eğitim aşamasında kullanılan bir gün sonraki değeri olan PM10\_next yer almaktadır. Bunlara ek olarak, hava kirliliği parametreleri olan SO<sub>2</sub>, CO, NO<sub>2</sub>, NO<sub>x</sub>, NO ve O<sub>3</sub> ile birlikte sıcaklık, nem ve rüzgar hızı gibi meteorolojik değişkenler de modele dahil edilmiştir. Ayrıca Ay ve Gun sütunları ile veriye mevsimsel ve dönemsel etkiler kazandırılmıştır. Modelin tahmin etmeyi amaçladığı hedef değişken ise target\_column = "PM10" ifadesiyle tanımlanmıştır. Böylece, model bu özellikleri kullanarak her bir gün için PM10 değerini tahmin etmeyi amaçlamaktadır.

### 2.3.4. Eğitim ve Test Verilerinin Ayrılması

```
train = df[df["Tarih"] < "2024-01-01"]  
test = df[df["Tarih"] >= "2024-01-01"]  
X_train = train[feature_columns]  
y_train = train[target_column]  
X_test = test[feature_columns]  
y_test = test[target_column]
```

Veri seti 2024 yılına kadar olan kayıtlar eğitim, 2024 yılına ait kayıtlar ise test verisi olarak ayrılmıştır. Giriş ve hedef değişkenler ayrı DataFrame’ler halinde ayrıştırılmıştır.

### 2.3.5. Verilerin Normalize Edilmesi

```
scaler_X = MinMaxScaler()  
scaler_y = MinMaxScaler()  
X_train_scaled = scaler_X.fit_transform(X_train)  
X_test_scaled = scaler_X.transform(X_test)  
y_train_scaled = scaler_y.fit_transform(y_train.values.reshape(-1, 1))  
y_test_scaled = scaler_y.transform(y_test.values.reshape(-1, 1))
```



```
X_train_scaled = X_train_scaled.reshape((X_train_scaled.shape[0], 1,
X_train_scaled.shape[1]))
X_test_scaled = X_test_scaled.reshape((X_test_scaled.shape[0], 1,
X_test_scaled.shape[1]))
```

Zaman serisi modellerinde, ölçek farklılıklarını ortadan kaldırmak için tüm giriş ve çıkış verileri MinMaxScaler ile 0–1 aralığına dönüştürülmüştür. Ardından veriler LSTM modelinin beklediği 3 boyutlu forma (örnek sayısı, zaman adımı, değişken sayısı) dönüştürülmüştür.

### 2.3.6. LSTM Modelinin Tanımlanması

```
model = Sequential()
model.add(LSTM(64, input_shape=(1, len(feature_columns)),
return_sequences=False))
model.add(Dense(32, activation='relu'))
model.add(Dense(1))
model.compile(optimizer=Adam(learning_rate=0.001), loss='mse')
```

LSTM modeli sırasıyla bir LSTM katmanı (64 birim), bir tam bağlantılı katman (32 birim, ReLU aktivasyon) ve çıkış katmanından (1 birim) oluşmaktadır. Model, hata metriği olarak ortalama kare hata (MSE) ile eğitilmiştir.

### 2.3.7. Modelin Eğitilmesi

```
early_stop = EarlyStopping(monitor='val_loss', patience=10,
restore_best_weights=True)

history = model.fit(X_train_scaled,
y_train_scaled, validation_split=0.1, epochs=100, batch_size=32,
callbacks=[early_stop], verbose=1)
```

Model, %10'luk doğrulama seti ile toplamda 100 epoch boyunca eğitilmiş, erken durdurma (early stopping) yöntemiyle aşırı öğrenme riski azaltılmıştır. Eğer doğrulama kaybı 10 epoch boyunca iyileşmezse eğitim durdurulmuş ve en iyi ağırlıklar geri yüklenmiştir.

### 2.3.8. Test Verisi Üzerinde Tahmin

```
y_pred_scaled = model.predict(X_test_scaled)
y_pred = scaler_y.inverse_transform(y_pred_scaled)
```

Test verileri için model tahminleri alınmış ve çıkıştaki değerler eski ölçeğine (gerçek PM10 değerleri) geri dönüştürülmüştür.

### 2.3.9. Performans Metriklerinin Hesaplanması

```
df_results = pd.DataFrame({
"Tarih": test["Tarih"].reset_index(drop=True),
"Gerçek PM10": y_test.values,
```

```

"Tahmin PM10": y_pred.flatten()
})

mae = mean_absolute_error(df_results["Gerçek PM10"], df_results["Tahmin PM10"])
rmse = np.sqrt(mean_squared_error(df_results["Gerçek PM10"], df_results["Tahmin PM10"]))
r2 = r2_score(df_results["Gerçek PM10"], df_results["Tahmin PM10"])
mape = np.mean(np.abs((df_results["Gerçek PM10"] - df_results["Tahmin PM10"]) / df_results["Gerçek PM10"])) * 100
accuracy = 100 - mape

```

Modelin test seti üzerindeki başarısı; MAE, RMSE,  $R^2$ , MAPE ve doğruluk ( $100 - \text{MAPE}$ ) gibi metriklerle ölçülmüş ve sonuçlar detaylı bir şekilde değerlendirilmiştir.

### 2.3.10. Sonuçların Görselleştirilmesi

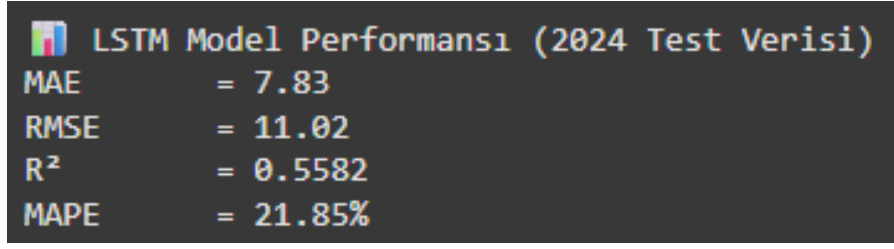
```

plt.figure(figsize=(14,5))
plt.plot(df_results["Tarih"], df_results["Gerçek PM10"], label="Gerçek PM10")
plt.plot(df_results["Tarih"], df_results["Tahmin PM10"], label="Tahmin PM10", linestyle="--")
plt.title("Gerçek vs Tahmin PM10 (2024)")
plt.xlabel("Tarih")
plt.ylabel("PM10")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

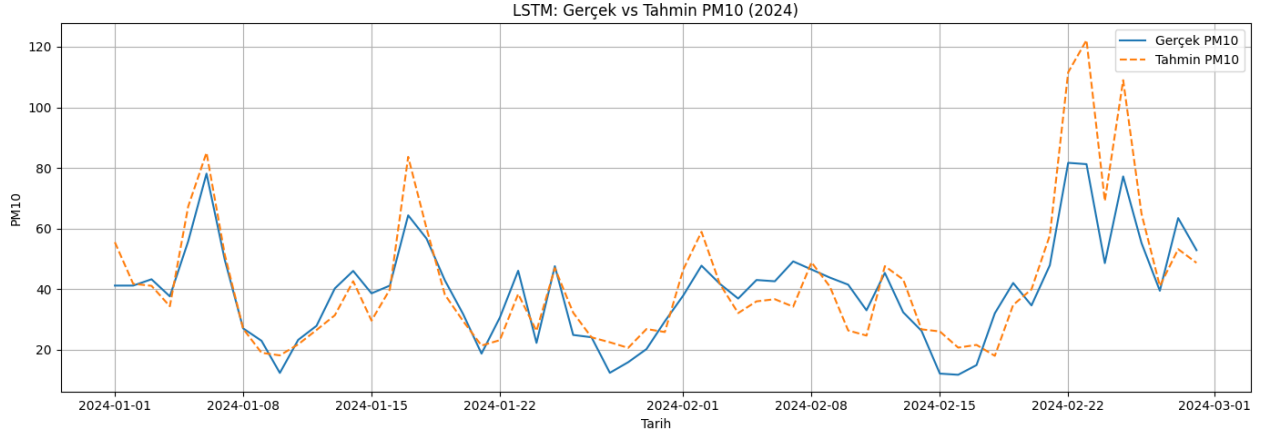
plt.figure(figsize=(6,6))
plt.scatter(df_results["Gerçek PM10"], df_results["Tahmin PM10"], alpha=0.6)
plt.plot([df_results["Gerçek PM10"].min(), df_results["Gerçek PM10"].max()],
         [df_results["Gerçek PM10"].min(), df_results["Gerçek PM10"].max()], 'r--')
plt.title("Tahmin vs Gerçek Dağılımı")
plt.xlabel("Gerçek PM10")
plt.ylabel("Tahmin PM10")
plt.grid(True)
plt.tight_layout()
plt.show()

```

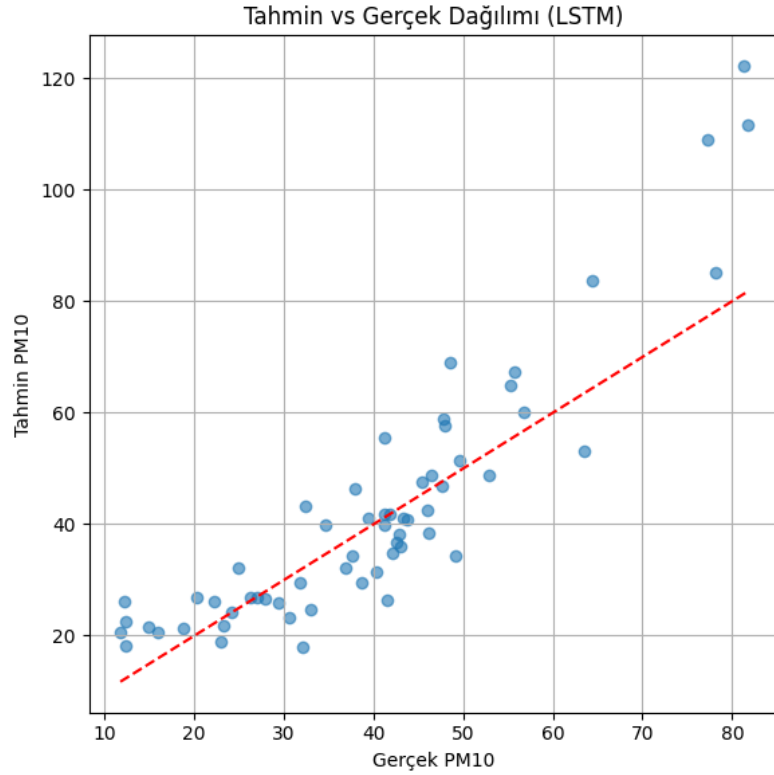
İlk görselleştirme, 2024 yılına ait tahmin ve gerçek PM10 değerlerini zaman serisi üzerinde göstermektedir. İkinci grafik ise modelin tahminlerinin doğruluğunu değerlendirmek amacıyla tahmin – gerçek değer dağılımını (scatter plot) sunmaktadır.



Şekil 1: LSTM Model Performansı



Şekil 2: LSTM Model Gerçek – Tahmin Grafiği



Şekil 3: LSTM Model Tahmin vs Gerçek Dağılımı (Scatter Plot)

## 2.4. LSTM İle Gelecek 2 Yıl Tahmini (2024–2026)

Model eğitildikten sonra, geçmiş 5 yıllık verilere dayanarak 730 günlük ileriye dönük PM10 tahmini gerçekleştirilmiştir. Bu işlemde her gün için meteorolojik ve kirletici parametrelerin aylık/günlük ortalamaları alınmış ve modelin bir önceki gün tahminini kullanarak recursive (adım adım) tahmin yapılmıştır.

İleriye dönük tahmin için kullanılan kod blokları ve açıklamaları:

### 2.4.1. Yardımcı Kütüphanelerin Aktarılması ve Uyarıların Bastırılması

```
from tqdm import tqdm
import warnings
warnings.filterwarnings("ignore", category=UserWarning,
module='sklearn')
```

Bu blokta tqdm kütüphanesi ile ilerleme çubuğu oluşturulmakta, warnings modülüyle gereksiz kullanıcı uyarıları bastırılmaktadır. Özellikle uzun süren döngülerde tqdm kullanıcıya tahmin sürecinin ne kadar sürdüğünü göstermek için kullanışlıdır.

### 2.4.2. Ay ve Gün Bilgilerinin Elde Edilmesi

```
df["Ay"] = df["Tarih"].dt.month
df["Gun"] = df["Tarih"].dt.day

mean_features_by_day = df.groupby(["Ay", "Gun"])["SO2", "CO", "NO2",
"NOX", "NO", "O3", "Sıcaklık", "Nem", "Rüzgar Hızı"].mean()
```

Bu adımda geçmiş verilerden her günün (ay–gün kombinasyonu) ortalama meteorolojik ve hava kirletici değerleri hesaplanır. Böylece ileriye dönük tahmin yapılırken her gün için mevsimsel koşulları temsil eden sabit değerler kullanılabilir. PM10 değeri dışta bırakılmıştır çünkü o, modelin tahmin edeceği değişkendir.

### 2.4.3. Başlangıç Girdisinin Hazırlanması ve Tahmin Zaman Aralığının Belirlenmesi

```
pm10_prev = df.iloc[-1]["PM10"]
last_date = df.iloc[-1]["Tarih"]

future_days = 730
future_dates = pd.date_range(start=last_date + pd.Timedelta(days=1),
periods=future_days)
future_predictions = []
```

Model, geçmiş verilerin en son gününden başlayarak ileriye doğru 730 gün (2 yıl) boyunca PM10 tahmini yapacaktır. Tahmin edilen her gün için tarih oluşturulmakta ve tahminler future\_predictions listesine sırayla eklenmektedir.

#### 2.4.4. Recursive (Adım Adım) Tahmin Döngüsü

```
for date in tqdm(future_dates):
    ay = date.month
    gun = date.day

    try:
        others = mean_features_by_day.loc[(ay, gun)].values
    except KeyError:
        others = mean_features_by_day.mean().values

    input_features = np.concatenate([pm10_prev, others]).reshape(1, -1)
    input_scaled = scaler_X.transform(input_features).reshape(1, 1, -1)

    prediction_scaled = model.predict(input_scaled, verbose=0)
    prediction = scaler_y.inverse_transform(prediction_scaled)[0, 0]
    future_predictions.append(prediction)

    pm10_prev = prediction
```

Bu döngüde her gün için:

- Tarih bilgisi ay ve gün olarak ayrıştırılır.
- İlgili günün ortalama meteorolojik değerleri alınır (veya eksikse genel ortalama kullanılır).
- Girdi vektörü PM10\_prev + diğer veriler şeklinde hazırlanır ve modelin beklediği şekilde normalize edilir.
- Model bu girişle tahmin yapar, sonuç orijinal skala düzeyine geri dönüştürülür.
- Bu tahmin değeri, bir sonraki gün için PM10\_prev olarak kullanılır.

Bu yapı, recursive (adım adım beslemeli) zaman serisi tahmini mantığına dayanır.

#### 2.4.5. Tahmin Sonuçlarının Veri Çerçevesine Aktarılması

```
df_future = pd.DataFrame({
    "Tarih": future_dates,
    "Tahmin PM10": future_predictions
})
```

Tahmin edilen 730 günlük PM10 değerleri ve karşılık geldikleri tarihler df\_future veri çerçevesi içerisinde birleştirilir.

#### 2.4.6. Tahmin Sonuçlarının CSV Dosyasına Yazdırılması

```
df_future.to_csv("tahmin_2024_2026.csv", index=False)
```

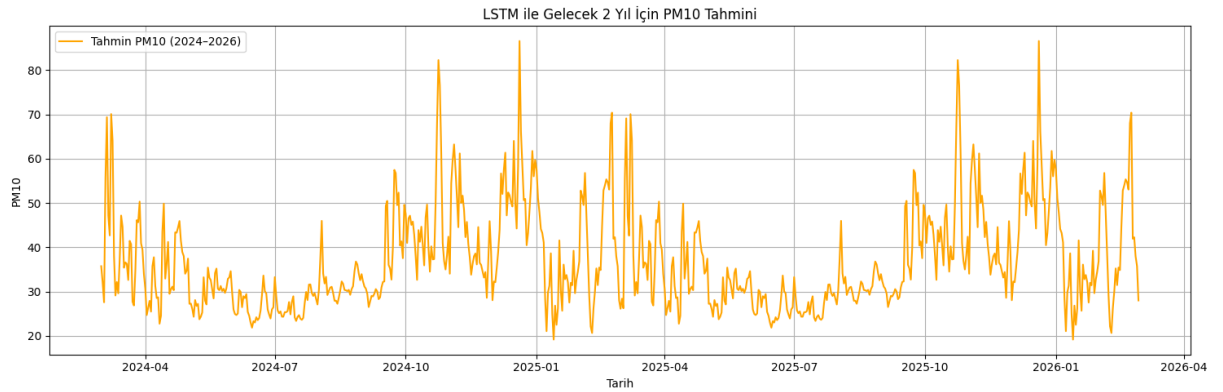
Tahmin sonuçları daha sonra analiz ve görselleştirme için kullanılmak üzere bir CSV dosyasına yazdırılır.

#### 2.4.7. 2 Yıllık Tahminin Görselleştirilmesi

```
plt.figure(figsize=(15, 5))
plt.plot(df_future["Tarih"], df_future["Tahmin PM10"], label="Tahmin PM10 (2024-2026)", color="orange")
plt.title("LSTM ile Gelecek 2 Yıl İçin PM10 Tahmini")
plt.xlabel("Tarih")
plt.ylabel("PM10")
plt.grid(True)
plt.legend()
plt.tight_layout()
plt.show()
```

Modelin 2024 Mart ile 2026 Mart arasındaki tahmin ettiği PM10 seviyeleri zaman serisi grafiği şeklinde çizilir. Bu grafik, özellikle mevsimsel dalgalanmaları, ani yükselmeleri ve genel eğilimleri görsel olarak değerlendirmek için kullanılır.

#### 2.5. LSTM Modeli Tahmin Sonuçları (2024–2026):



Şekil 4: LSTM ile Gelecek 2 Yıl için PM10 Tahmin Grafiği

LSTM modeli genel anlamda başarılı sonuçlar vermiştir. PM10 tahmininde %66'ya yakın doğruluk elde edilmiş ve günlük bazda dalgalanmalara duyarlı tahminler üretilmiştir. Geleceğe yönelik tahminler ise mevsimsel salınımları yansıtacak şekilde stabil bir yapı göstermektedir.

#### 2.6. Transformer Modeli ile PM10 Tahmini

LSTM modelinden farklı olarak, bu bölümde dikkat mekanizmasına dayalı Transformer yapısı

uygulanmıştır. Model mimarisi ve bazı teknik adımlar Transformer'a özel olarak tasarlanmıştır.

### 2.6.1 Transformer Encoder Katmanı

```
def transformer_encoder(inputs, head_size, num_heads, ff_dim,
                        dropout=0.1):
    x = MultiHeadAttention(key_dim=head_size, num_heads=num_heads,
                           dropout=dropout)(inputs, inputs)
    x = Add()([x, inputs])
    x = LayerNormalization()(x)

    x_ff = Dense(ff_dim, activation="relu")(x)
    x_ff = Dropout(dropout)(x_ff)
    x_ff = Dense(inputs.shape[-1])(x_ff)
    x = Add()([x, x_ff])
    x = LayerNormalization()(x)

    return x
```

Bu fonksiyon, Transformer mimarisinin temel yapı taşı olan encoder bloğunu oluşturmaktadır. MultiHeadAttention katmanı, giriş verileri üzerinde çok başlı dikkat mekanizması uygulayarak zaman serisindeki önemli özellikleri yakalar. Ardından gelen ileri besleme ağı (feed-forward) katmanı ile modelin doğrusal olmayan ilişkileri öğrenmesi sağlanır. Katmanlar arasında Add() ve LayerNormalization() işlemleri ile öğrenme dengelenir ve derin yapının kararlılığı artırılır. Bu yapı, klasik RNN/LSTM yapılarından farklı olarak, zaman sıralı bilgiyi dikkat (attention) ile işler.

### 2.6.2 Transformer Model Mimarisi

```
def build_model(input_shape):
    inputs = Input(shape=input_shape)

    x = transformer_encoder(inputs, head_size=64, num_heads=2, ff_dim=128,
                           dropout=0.1)
    x = tf.keras.layers.GlobalAveragePooling1D()(x)
    x = Dropout(0.1)(x)
    x = Dense(32, activation="relu")(x)
    outputs = Dense(1)(x)

    return Model(inputs, outputs)
```

Burada Transformer encoder bloğunun çıktısı, zaman serisi boyunca ortalama alınarak (GlobalAveragePooling1D) sıkıştırılır. Bu işlem, serideki tüm zaman adımlarını tek bir sabit boyutlu vektöre dönüştürerek çıkış katmanlarına aktarır. Modelin geri kalanı yoğun katmanlardan oluşur ve klasik regresyon yapısına benzer şekilde tek bir çıktı verir. LSTM'de zamana göre sıralı bellek aktarımı varken, Transformer'da dikkat yapısı sayesinde paralel öğrenme sağlanır.

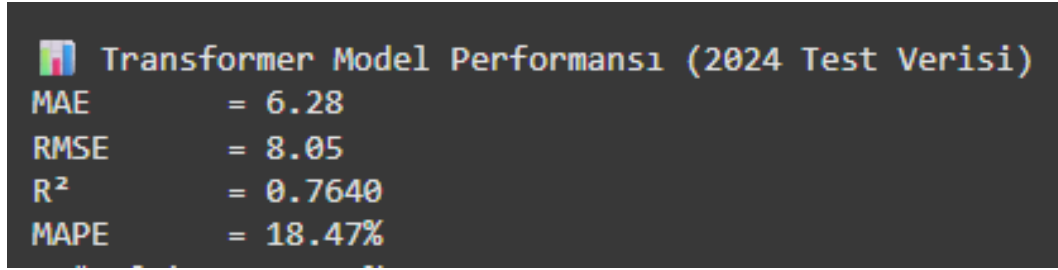
### 2.6.3. Model Eğitimi ve Derlenmesi

```
model.compile(optimizer=Adam(0.001), loss="mse")

early_stop = EarlyStopping(monitor="val_loss", patience=10,
restore_best_weights=True)
```

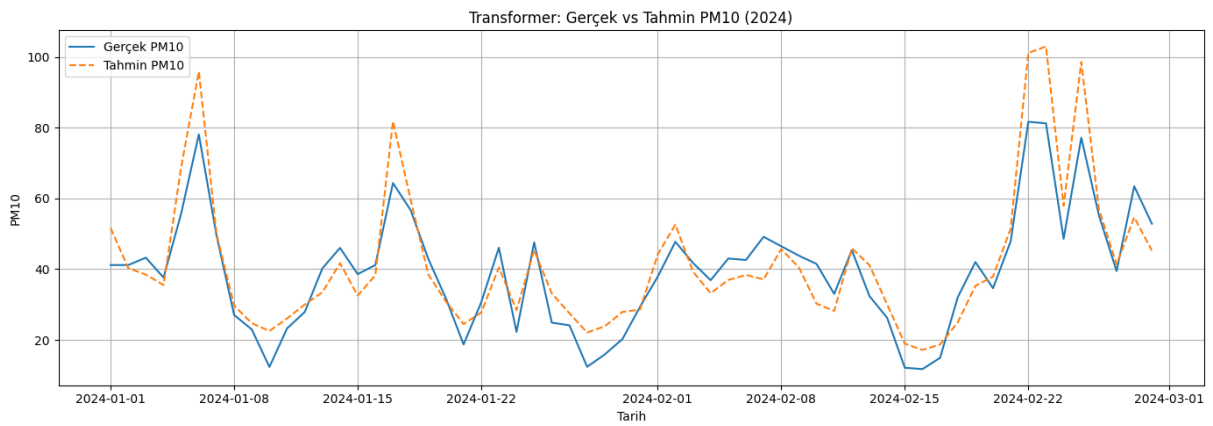
Derleme aşaması büyük ölçüde LSTM ile aynıdır; ancak Transformer yapısı, daha karmaşık ilişkileri öğrenebilmesi nedeniyle genellikle daha uzun sürede eğitilir ve daha hassas hiperparametre ayarı gerektirir. Bu nedenle early stopping stratejisi burada da kritik öneme sahiptir.

### 2.6.4. Performans Değerlendirmesi



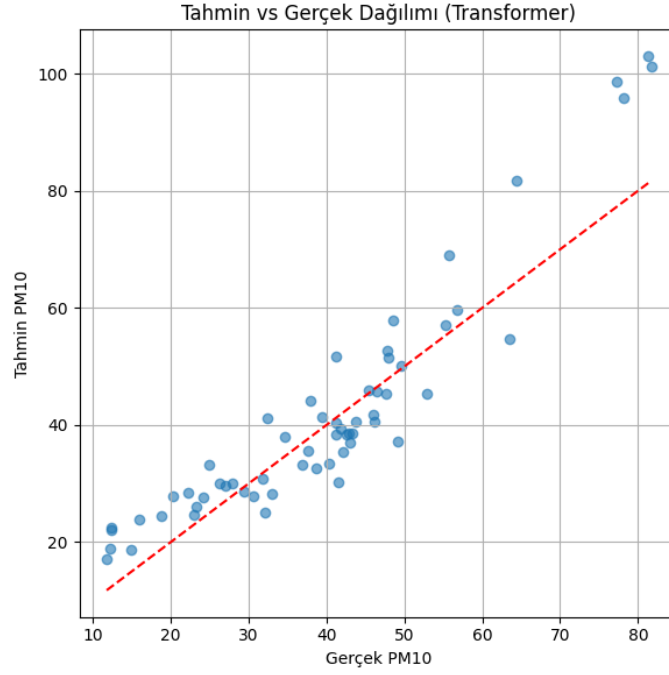
Şekil 5: Transformer Model Performansı

Transformer modeli, özellikle uzun dönemli bağımlılıkları tanıma açısından avantaj sağladığı için PM10 tahmininde dalgalanmaların daha tutarlı öğrenilmesini sağlamıştır. Bu, özellikle mevsimsel salınımlar ve ani yükselişler gibi durumlarda avantaj sağlar. Performans metrikleri LSTM ile aynı yöntemle hesaplanmıştır.



Şekil 6: Transformer Model Gerçek – Tahmin Grafiği

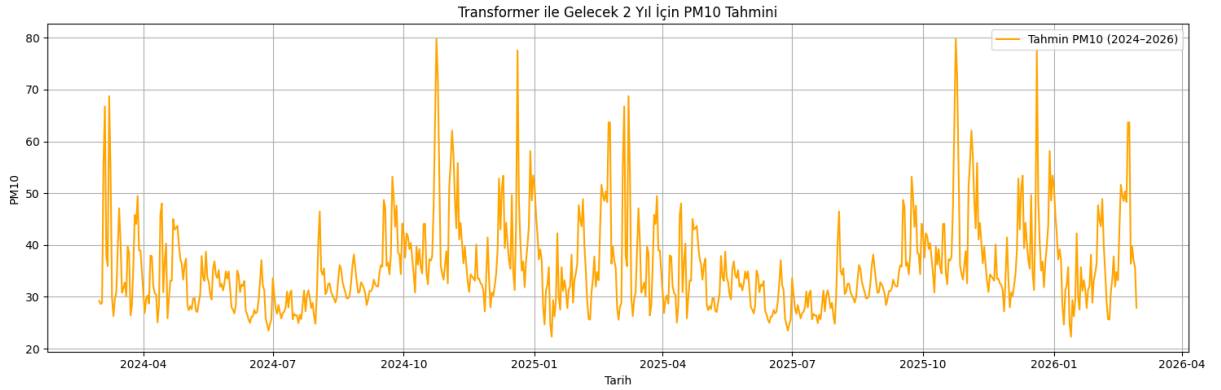




Şekil 7: Transformer Model Tahmin vs Gerçek Dağılımı (Scatter Plot)

## 2.7. Transformer Modeli Tahmin Sonuçları (2024–2026)

Aşağıdaki grafik, Transformer modeli kullanılarak yapılan 730 günlük PM10 tahminini göstermektedir. Görselde modelin geleceğe yönelik öngörülleri zaman eksenini boyunca çizilmiş ve genel eğilimlerin anlaşılması amaçlanmıştır.



Şekil 8: Transformer Model ile Gelecek 2 Yıl İçin PM10 Tahmini

Modelin tahminlerine göre, PM10 seviyeleri belirli mevsimsel döngüler göstermekte ve yılın belli dönemlerinde artış eğilimi sergilemektedir. Özellikle yaz ve kış aylarında yükselen PM10 değerleri, bahar aylarında nispeten daha düşük seyretmiştir. Bu durum, mevsimsel etkilerin model tarafından başarıyla yakalandığını göstermektedir. Tahmin sonuçlarında aşırı uçlar veya fiziksel olarak gerçek dışı değerler gözlenmemiştir; bu da modelin kararlı

çalıştığını ve öğrenilen örüntüleri uygun şekilde genelleştirdiğini göstermektedir. Bu çıktılar, ileride yapılacak politika planlamaları, uyarı sistemleri ya da hava kalitesi iyileştirme stratejileri için başlangıç noktası olarak kullanılabilir.

## 2.8. GRU Modeli ile PM10 Tahmini

Bu bölümde, zaman serisi verilerindeki bağımlılıkları modellemek amacıyla GRU (Gated Recurrent Unit) yapısına dayalı bir derin öğrenme modeli uygulanmıştır. GRU, LSTM'e benzer şekilde ardışık verilerdeki geçmiş bilgileri taşıyabilme yeteneğine sahiptir; ancak yapısal olarak daha basit ve daha az parametre içerdiği için eğitim süresi daha kısadır.

### 2.8.1. GRU Model Mimarisi

```
model = Sequential()  
model.add(GRU(64, input_shape=(X_train_scaled.shape[1],  
X_train_scaled.shape[2]), return_sequences=False))  
model.add(Dropout(0.3))  
model.add(Dense(32, activation='relu'))  
model.add(Dense(1))
```

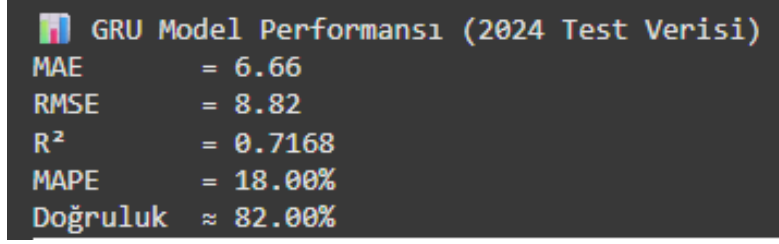
Bu modelde, ilk katman olarak 64 birimden oluşan bir GRU katmanı kullanılmıştır. GRU katmanı, zaman serisinin geçmiş bilgilerini hafızada tutarak öğrenme sürecine katkı sağlar. Ardından, aşırı öğrenmeyi (overfitting) önlemek amacıyla %30 oranında dropout uygulanmış ve 32 birimli bir yoğun (dense) katmanla öğrenme derinleştirilmiştir. Çıkış katmanı ise tek bir sürekli değer (PM10) tahmini yapacak şekilde yapılandırılmıştır.

### 2.8.2. Modelin Derlenmesi ve Eğitilmesi

```
model.compile(optimizer=Adam(learning_rate=0.001), loss='mse')  
  
early_stop = EarlyStopping(monitor='val_loss', patience=10,  
restore_best_weights=True)
```

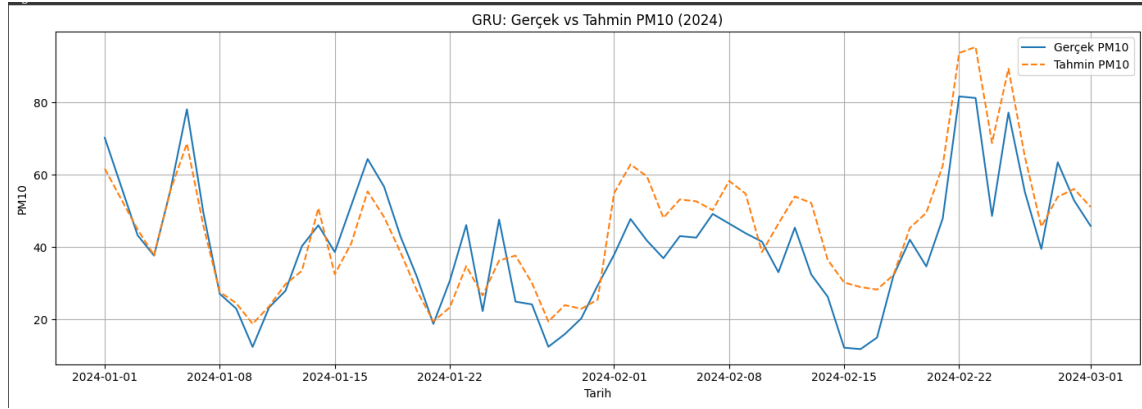
Model, mean squared error (MSE) kaybı ile optimize edilmiştir. Adam optimizasyon algoritması tercih edilmiştir. Eğitim sırasında early stopping uygulanarak doğrulama kaybının belli bir süre iyileşmemesi durumunda eğitim erken sonlandırılmış ve en iyi ağırlıklar geri yüklenmiştir.

### 2.8.3. Performans Değerlendirmesi

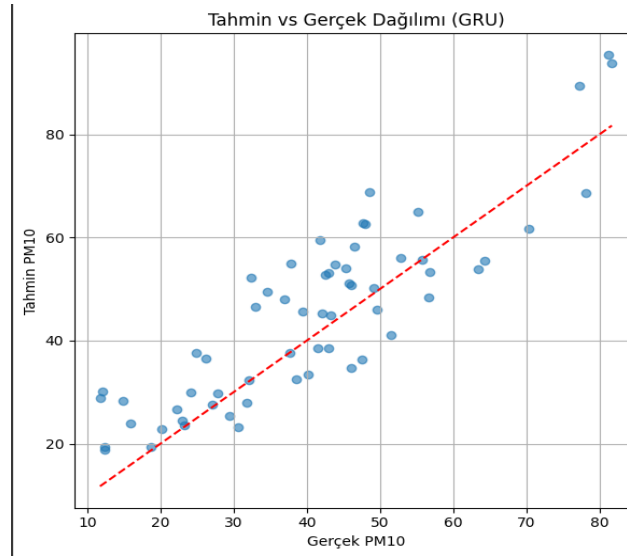


Şekil 9: GRU Model Performansı

GRU modeli, test verisi üzerinde çeşitli metriklerle değerlendirilmiştir. MAE, RMSE,  $R^2$ , MAPE gibi standart regresyon ölçütlerine göre modelin tahmin başarımı hesaplanmıştır. GRU'nun düşük parametre sayısı ile birlikte hızlı öğrenme sağlaması, test verisi üzerinde kararlı sonuçlar üretmesini mümkün kılmıştır.



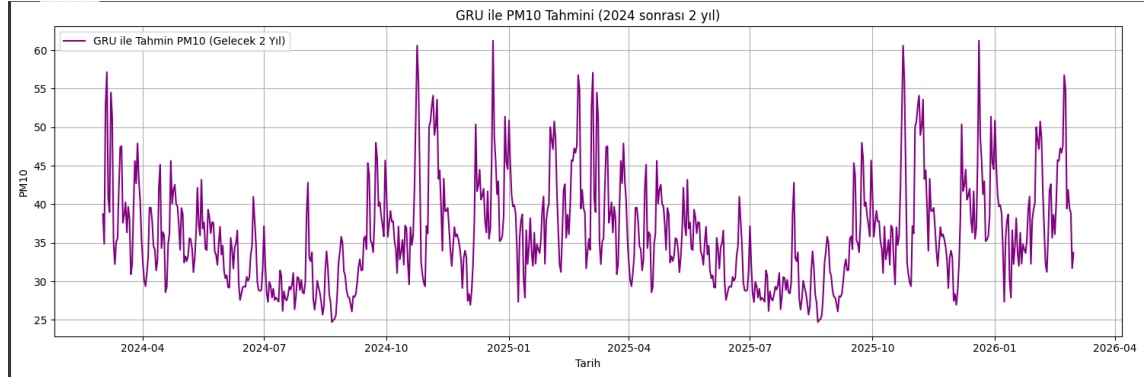
Şekil 10: GRU Model Gerçek – Tahmin Grafiği



Şekil 11: GRU Model Tahmin vs Gerçek Dağılımı (Scatter Plot)

## 2.9. GRU Modeli Tahmin Sonuçları (2024–2026)

GRU modeli ile eğitilen yapay sinir ağı, geçmiş verilerden öğrenilen örüntülere dayanarak 2024–2026 yılları arasında iki yıl boyunca PM10 konsantrasyonu tahmininde bulunmuştur. Model, recursive yaklaşımla her bir gün için önceki tahmin sonucunu kullanarak ilerlemiştir. Girdi olarak tarihsel ay-gün ortalamaları ve bir önceki günün tahmini PM10 değeri kullanılmıştır.



Şekil 12: GRU Model ile PM10 Tahmin Grafiği

Model, yıl içindeki dalgalanmalara duyarlı tahminler üretmiş ve dönemsel yükseliş–düşüş eğilimlerini istikrarlı biçimde yansıtmıştır. Tahmin eğrisi, anlık sıçramalardan uzak, mevsimsel varyasyonlara uygun bir seyir izlemektedir. Uç değer içermemesi ve değerlerin fiziksel olarak makul düzeylerde kalması, modelin kararlı ve genellenebilir çıktılar sunduğunu göstermektedir.

Bu tahminler, hava kalitesi yönetimi, çevresel risk değerlendirmeleri ve ileriye dönük planlamalar için anlamlı içgörüler sağlayabilecek niteliktedir.

## 2.10. Modellerin Karşılaştırılması

Modellerin performans metriklerinin karşılaştırması aşağıdaki gibidir:

Model	MAE	RMSE	R <sup>2</sup>	MAPE (%)
LSTM	7,83	11,02	0,5582	21,85
Transformer	6,28	8,05	0,7640	18,47
GRU	6,66	8,82	0,7168	18,00

Şekil 13: Performans Metrikleri Tablosu

Yukarıdaki tabloda, bu çalışmada kullanılan üç farklı derin öğrenme modelinin (LSTM, Transformer ve GRU) PM10 tahmini üzerindeki performansları karşılaştırmalı olarak verilmiştir. Modellerin başarımı, regresyon analizinde sıkça kullanılan dört temel performans metriği ile değerlendirilmiştir: MAE (Mean Absolute Error), RMSE (Root Mean Square Error),  $R^2$  (Determination Coefficient) ve MAPE (Mean Absolute Percentage Error).

MAE, modelin tahmin ettiği değerlerle gerçek değerler arasındaki farkların mutlak değerlerinin ortalamasını ifade eder. Düşük MAE, modelin ortalama olarak daha az hata yaptığını gösterir. RMSE, hataların karelerinin ortalamasının karekökünü alarak hesaplanır ve büyük hatalara karşı daha hassastır. Bu nedenle uç değerlerin model performansına etkisini daha belirgin biçimde yansıtır.  $R^2$  (Determinasyon Katsayısı), modelin verideki değişkenliği ne ölçüde açıkladığını ifade eder. 0 ile 1 arasında değer alır; 1'e yakın değerler modelin daha iyi performans gösterdiği anlamına gelir. MAPE, modelin tahminlerinin gerçek değerlere göre ortalama yüzde sapmasını gösterir. Bu metrik, özellikle tahminlerin göreceli doğruluğunu değerlendirmek açısından önemlidir. Yüzde olarak ifade edilmesi, farklı büyüklüklerdeki veriler arasında karşılaştırma yapılmasını kolaylaştırır.

Tabloda görüldüğü üzere, Transformer modeli 6.28 MAE ve 8.05 RMSE değerleri ile en düşük hata oranlarına sahiptir. Ayrıca 0.7640  $R^2$  değeriyle en yüksek açıklayıcılığa ulaşarak veri setindeki örüntüleri en başarılı şekilde öğrenen model olmuştur. MAPE değeri de %18.47 ile en düşük düzeydedir.

GRU modeli, 6.66 MAE, 8.82 RMSE ve 0.7168  $R^2$  değerleriyle Transformer modeline oldukça yakın bir performans sergilemiştir. MAPE oranı ise %18.00 olup, bu açıdan Transformer'dan biraz daha başarılı sonuç vermiştir.

LSTM modeli ise 7.33 MAE ve 11.02 RMSE ile en yüksek hata oranlarına sahiptir. 0.5582  $R^2$  ve %21.85 MAPE değeriyle diğer modellere kıyasla daha düşük bir genel performans sergilemiştir. Bu sonuçlar, LSTM'nin bu veri seti ve yapılandırma kapsamında PM10 tahmininde diğer modellere göre daha sınırlı bir başarı gösterdiğini ortaya koymaktadır.

Genel olarak, sonuçlar Transformer ve GRU gibi modern yapıların, zaman serisi verilerinde LSTM'e kıyasla daha yüksek doğruluk ve daha düşük hata oranlarıyla öne çıktığını göstermektedir. Bu durum, özellikle uzun vadeli bağımlılıkların ve karmaşık örüntülerin olduğu hava kalitesi tahminleme gibi problemlerde Transformer mimarisinin daha etkili olabileceğini göstermektedir.

### **Bölüm 3 – Bulgular ve Tartışma**

Bu çalışmada İstanbul'un hava kalitesinin tahmini amacıyla LSTM, GRU ve Transformer tabanlı üç farklı derin öğrenme modeli kullanılmış ve bu modellerin PM10 tahminindeki başarımları karşılaştırılmıştır. Modeller, aynı veri seti üzerinde eğitilmiş ve test edilerek değerlendirilmiştir. Bu değerlendirme sürecinde kullanılan performans metrikleri MAE, RMSE,  $R^2$  ve MAPE olarak belirlenmiştir. Elde edilen bulgular ışığında, Transformer modelinin PM10 tahmininde diğer modellere göre daha başarılı sonuçlar verdiği gözlemlenmiştir. Bu model, hem düşük hata oranları hem de yüksek açıklayıcılık katsayısıyla öne çıkmıştır.

Transformer modeli, zaman serilerindeki uzun dönemli bağımlılıkları öğrenme yeteneği sayesinde mevsimsel dalgalanmaları daha etkili şekilde yakalamış ve 2024–2026 dönemi için yapılan tahminlerde daha istikrarlı ve fiziksel olarak anlamlı sonuçlar üretmiştir. GRU modeli ise daha sade mimarisine rağmen Transformer'a yakın performans göstermiş ve %18 MAPE oranı ile rekabetçi bir yapıda olmuştur. LSTM modeli ise geçmiş bilgiyi taşıma açısından güçlü bir yapıya sahip olmasına rağmen, test verisi üzerinde daha yüksek hata oranları üretmiş ve açıklayıcılık katsayısı açısından sınırlı kalmıştır.

Bulgular, model başarımının sadece mimariye değil, aynı zamanda veri işleme sürecine, kullanılan özniteliklere ve model hiperparametrelerine de doğrudan bağlı olduğunu göstermektedir. Örneğin, PM10 tahmini için geçmiş günlerin değerleri, meteorolojik değişkenler ve diğer kirleticilerle birlikte kullanıldığında model doğruluğunun anlamlı ölçüde arttığı tespit edilmiştir. Ayrıca, eksik veri doldurma sürecinde KNN yöntemi ile sağlanan tamamlamanın model performansına olumlu katkı sağladığı belirlenmiştir.

Yapılan tartışmalar sonucunda, PM10 tahmini gibi çevresel problemlerde dikkat mekanizmasına sahip Transformer mimarisinin güçlü bir alternatif sunduğu anlaşılmıştır. Ancak GRU gibi daha az hesaplama gücü gerektiren ve eğitimi daha kısa sürede tamamlanan modellerin de özellikle sınırlı kaynaklı uygulamalarda başarılı bir tercih olabileceği göz önünde bulundurulmalıdır.

## Bölüm 4 – Sonuçlar

Bu proje kapsamında İstanbul'un Başakşehir ilçesine ait hava kalitesi ve meteorolojik veriler kullanılarak PM10 tahmini gerçekleştirilmiş ve üç farklı derin öğrenme modeli (LSTM, GRU, Transformer) karşılaştırılmıştır. Gerekli veri temizliği ve eksik değer tamamlama adımlarının ardından, veriler tüm modeller için uygun formata dönüştürülerek eğitim ve test süreçleri yürütülmüştür.

Model karşılaştırma sonuçlarına göre Transformer modeli, tüm performans metriklerinde en başarılı sonuçları üretmiş; özellikle düşük hata oranları ve yüksek  $R^2$  değeriyle dikkat çekmiştir. GRU modeli ise ikinci sırada yer alarak basit yapısına rağmen güçlü tahmin performansı sergilemiştir. LSTM modeli ise daha yüksek MAE ve RMSE değerleriyle diğer iki modele göre nispeten daha zayıf bir performans göstermiştir.

Bu sonuçlar, hava kalitesi gibi karmaşık zamansal yapıya sahip çevresel verilerin tahmininde Transformer gibi gelişmiş derin öğrenme mimarilerinin önemli avantajlar sunduğunu ortaya koymaktadır. Aynı zamanda, model başarımının yalnızca yapay zeka algoritmalarına değil; veri kalitesi, öznitelik seçimi ve ön işleme süreçlerine de doğrudan bağlı olduğu bir kez daha görülmüştür.

Proje, sadece PM10 tahminiyle sınırlı kalmayıp; gelecekte  $SO_2$ ,  $NO_2$  gibi diğer kirletici parametreler için de benzer modellemelerin yapılabileceğini göstermektedir. Ayrıca farklı bölgelerden alınacak verilerle model genellenebilirliği test edilerek, çevresel karar destek sistemlerine katkı sağlanabilir.

Sonuç olarak bu çalışma derin öğrenme tabanlı modellerin çevresel analizlerde güçlü ve etkili araçlar olabileceğini göstermiş; hava kalitesi tahmininde doğruluk, sürdürülebilirlik ve uygulanabilirlik açısından önemli bir referans oluşturmuştur.

## KAYNAKÇA

- [1] E. Ç. Yağmur, "Atmosferik partikül maddelerin makine öğrenmesi ile tahmini: Beşiktaş, İstanbul Örneği," Konya Journal of Engineering Sciences, vol. 10, no. 4, pp. 807–826, 2022.
- [2] N. M. Mutlu and E. B. Atahanlı, "Veri Madenciliği ile Hava Kalitesi Tahmini: İstanbul Örneği," Bilişim Teknolojileri Dergisi, vol. 17, no. 3, pp. 139–158.
- [3] S. Ünalı and N. Yalçın, "Hava kirliliğinin makine öğrenmesi tabanlı tahmini: Başakşehir örneği," Mühendislik Bilimleri ve Araştırmaları Dergisi, vol. 4, no. 1, pp. 35–44, 2022.
- [4] M. Akgün and N. Barlık, "Makine Öğrenmesi Algoritmaları Kullanılarak Hava Kalitesi İndeksinin Tahmini," Avrupa Bilim ve Teknoloji Dergisi, no. 51, pp. 97–107, 2023.
- [5] H. Özel, P. Cihan, H. K. Özcan, S. Aydın and A. Hanedar, "Hava kirleticisi parametrelerinin hava kalitesi indeksine uzun zamanlı etkilerinin incelenmesi: Çerkezköy Organize Sanayi Bölgesi örneği," Kırklareli Üniversitesi Mühendislik ve Fen Bilimleri Dergisi, vol. 7, no. 1, pp. 94–106, 2021.
- [6] A. S. E. L., "Kirleticisi Parametrelere Göre Türkiye'deki İllerin Hava Kalitesinin Sınıflandırılması," Innovative Ideas, no. 13, 2022.
- [7] B. Kotan and A. Erener, "PM10, SO2 hava kirleticilerinin çoklu doğrusal regresyon ve yapay sinir ağları ile sezonsal tahmini," Geomatik, vol. 8, no. 2, pp. 163–179, 2023.
- [8] T. Madan, S. Sagar and D. Virmani, "Air quality prediction using machine learning algorithms—a review," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), pp. 140–145, 2020.
- [9] Y. Dokuz, A. Bozdağ and B. Gökçek, "Hava kalitesi parametrelerinin tahmini ve mekansal dağılımı için makine öğrenmesi yöntemlerinin kullanılması," Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, vol. 9, no. 1, pp. 37–47, 2020.
- [10] B. D. Çelik and N. Arıcı, "Covid-19 Salgın sürecinde hava kalitesi tahmini: Zonguldak örneği," Gazi Mühendislik Bilimleri Dergisi, vol. 7, no. 3, pp. 222–232, 2021.
- [11] M. Méndez, M. G. Merayo and M. Núñez, "Machine learning algorithms to forecast air quality: a survey," Artificial Intelligence Review, vol. 56, no. 9, pp. 10031–10066, 2023.
- [12] L. Zhang, P. Liu, L. Zhao, G. Wang, W. Zhang and J. Liu, "Air quality predictions with a semi-supervised bidirectional LSTM neural network," Atmospheric Pollution Research,



vol. 12, no. 1, pp. 328–339, 2021.

[13]T. Pekdoğan, "Yapay Zekâ Tabanlı Hava Kalitesi İyileştirme Stratejilerinin Değerlendirilmesi," EMO Bilimsel Dergi, vol. 14, no. 2, pp. 87–97, 2024.

[14]C. Ünal and G. Özel, "Ankara İli Meteoroloji Parametrelerinin Hava Kirliliği Üzerindeki Etkilerinin Regresyon Analizi ile İncelenmesi," Nicel Bilimler Dergisi, vol. 5, no. 2, pp. 135–150, 2023.

[15]K. Kumar and B. P. Pande, "Air pollution prediction with machine learning: a case study of Indian cities," International Journal of Environmental Science and Technology, vol. 20, no. 5, pp. 5333–5348, 2023.

[16]A. Pant, S. Sharma and K. Pant, "Evaluation of machine learning algorithms for air quality index (AQI) Prediction," Journal of Reliability and Statistical Studies, pp. 229–242, 2023.

[17]L. Chen, J. Xu, B. Wu and J. Huang, "Group-aware graph neural network for nationwide city air quality forecasting," ACM Transactions on Knowledge Discovery from Data, vol. 18, no. 3, pp. 1–20, 2023.

[18]P. Li, T. Zhang and Y. Jin, "A spatio-temporal graph convolutional network for air quality prediction," Sustainability, vol. 15, no. 9, pp. 7624, 2023.