



Teknoloji Fakültesi

## BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

# İSTANBUL'UN HAVA KALİTESİNİN TAHMİNİ İÇİN GNN, GCN VE TRANSFORMER MODEL TABANLI DERİN ÖĞRENME YÖNTEMLERİNİN İNCELENMESİ

### BİTİRME PROJESİ 2. ARA RAPORU

Bilgisayar Mühendisliği Bölümü

**DANIŞMAN**

Doç. Dr. Buket DOĞAN

İSTANBUL, 2025

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

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği Öğrencileri Hamza GENÇAY, Erdem SAÇAN ve Çağan DERBENT tarafından “İSTANBUL’UN HAVA KALİTESİNİN TAHMİNİ İÇİN GNN, GCN VE TRANSFORMER MODEL TABANLI DERİN ÖĞRENME YÖNTEMLERİNİN İNCELENMESİ” başlıklı proje çalışması, 26.05.2025 tarihinde savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

**Jüri Üyeleri**

Dr. Öğr. Üyesi xxx xxx  
Marmara Üniversitesi  
Prof. Dr. Xxx xxx  
Marmara Üniversitesi  
Prof. Dr. Xxx xxx  
Marmara Üniversitesi

(Danışman)

(Üye)

(Üye)

(İMZA).....

(İMZA).....

(İMZA).....

## ÖNSÖZ

Proje çalışmamız süresince karşılaştığım bütün problemlerde, sabırla yardım ve bilgilerini esirgemeyen, tüm desteğini sonuna kadar yanımda hissettiğim değerli hocalarım, sayın Arş. Gör. Damla MENGÜŞ ve sayın Doç. Dr. Buket DOĞAN' a en içten teşekkürlerimizi sunarız.

Bu proje çalışması fikrinin oluşması ve ortaya çıkmasındaki önerisi ve desteğinden dolayı değerli hocam Arş. Gör. Damla MENGÜŞ' e teşekkür ederiz.

Proje çalışmam sırasında maddi ve manevi desteklerini esirgemeyen okul içerisinde ve okul dışında her zaman yanımda olan değerli çalışma arkadaşlarım ve hocalarım Doç. Dr. Buket DOĞAN ve Arş. Gör. Damla MENGÜŞ'e sonsuz teşekkürlerimizi sunarız.

# İÇİNDEKİLER

ÖZET	i
ABSTRACT	ii
SEMBOLLER	iii
KISALTMALAR	iv
1. GİRİŞ	1
1.1. Proje Çalışmasının Amacı ve Önemi	2
2. LİTERATÜR TARAMASI	3
3. BULGULAR VE TARTIŞMA	13
3.1. Şu Ana Kadar Ne Yaptık?	13
3.2 Bundan Sonra Ne Yapacağız?	14
4. SONUÇLAR	15
5. KAYNAKLAR	16

## ÖZET

### İSTANBUL'UN HAVA KALİTESİNİN TAHMİNİ İÇİN GNN, GCN VE TRANSFORMER MODEL TABANLI DERİN ÖĞRENME YÖNTEMLERİNİN İNCELENMESİ

Bu çalışmada, İstanbul'un hava kalitesinin tahmini amacıyla Grafik Sinir Ağları (GNN), Grafik Evrişimsel Ağlar (GCN) ve Transformer tabanlı derin öğrenme yöntemleri kullanılmaktadır. Projede, PM2.5, 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. Veriler, T.C. Çevre, Şehircilik ve İklim Değişikliği Bakanlığı ile Meteoroloji Genel Müdürlüğü'nden temin edilmekte 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.

**Mart, 2025**

**Öğrenciler**

**Hamza GENÇAY - 170422822**

**Erdem SAÇAN - 170422826**

**Çağan DERBENT - 170421028**

# **ABSTRACT**

## **ANALYSIS OF DEEP LEARNING METHODS BASED ON GNN, GCN AND TRANSFORMER MODELS FOR AIR QUALITY PREDICTION IN ISTANBUL**

In this study, Graph Neural Networks (GNN), Graph Convolutional Networks (GCN), and Transformer-based deep learning models are used to predict the air quality in Istanbul. Prediction models are developed using air pollution parameters such as PM<sub>2.5</sub>, PM<sub>10</sub>, SO<sub>2</sub>, and NO<sub>x</sub>, along with meteorological variables such as temperature, humidity, and wind speed. The data is collected from the Ministry of Environment, Urbanization and Climate Change of the Republic of Türkiye and the Turkish State Meteorological Service. After cleansing the data from missing and erroneous values, a data fusion approach is applied to preserve the temporal and spatial consistency. Spatial correlations between sensor locations are considered to generate a multidimensional dataset. Deep learning models are trained on Google Colab, and their performance is evaluated using metrics such as MAE, RMSE, and R<sup>2</sup>. At the final stage, model results are compared to identify the most accurate prediction approach, and the key factors affecting air quality in Istanbul are analyzed. This project aims to contribute to environmental analysis by improving the accuracy and reliability of air quality predictions.

**March, 2025**

**Students**

**Hamza GENÇAY - 170422822**

**Erdem SAÇAN - 170422826**

**Çağan DERBENT - 170421028**

## SEMBOLLER

<b>PM<sub>2.5</sub></b>	: 2.5 mikrondan küçük partikül madde
<b>PM<sub>10</sub></b>	: 10 mikrondan küçük partikül madde
<b>SO<sub>2</sub></b>	: Kükürt dioksit
<b>NO<sub>x</sub></b>	: Azot oksitleri (NO + NO <sub>2</sub> )
<b>T</b>	: Sıcaklık (°C)
<b>RH</b>	: Bağıl nem (%)
<b>WS</b>	: Rüzgar hızı (m/s)

## KISALTMALAR

<b>GNN</b>	: Graph Neural Network (Grafik Sinir Ağı)
<b>GCN</b>	: Graph Convolutional Network (Grafik Evrişimsel Ağ)
<b>LSTM</b>	: Long Short-Term Memory
<b>RMSE</b>	: Root Mean Square Error (Kök Ortalama Kare Hata)
<b>MAE</b>	: Mean Absolute Error (Ortalama Mutlak Hata)
<b>R<sup>2</sup></b>	: Determination Coefficient (Belirleme Katsayısı)
<b>AQI</b>	: Air Quality Index (Hava Kalitesi İndeksi)
<b>API</b>	: Application Programming Interface
<b>DL</b>	: Deep Learning (Derin Öğrenme)
<b>ANN</b>	: Artificial Neural Network (Yapay Sinir Ağı)
<b>ARIMA</b>	: AutoRegressive Integrated Moving Average
<b>CSV</b>	: Comma-Separated Values (Virgülle Ayrılmış Değerler)



## 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, Grafik Sinir Ağları (GNN), Grafik Evrimsel Ağlar (GCN) 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. Proje Çalışmasının Amacı ve Önemi**

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, Grafik Sinir Ağları (GNN), Grafik Evrişimsel Ağlar (GCN) ve Transformer tabanlı yapılar gibi güncel derin öğrenme modelleri kullanılacaktır. Projede, hava kirliliği parametreleri (PM2.5, 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, yoğunlukla verilerin doğrusal yapılarına ve lokal özelliklerine odaklanmakta, bu da hava kalitesi gibi çok boyutlu, etkileşimli sistemlerin doğru şekilde modellenmesini zorlaştırmaktadır. GNN ve GCN gibi grafik tabanlı yöntemler ise, şehir genelindeki farklı hava ölçüm istasyonlarının birbirleriyle olan mekansal etkileşimlerini dikkate alarak daha kapsamlı bir analiz imkânı sunar. 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.

## 2. LİTERATÜR TARAMASI

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, NOx, SO2 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 Ünal ve Nesibe Yalçın tarafından 2022 yılında gerçekleştirilen çalışmada [3], İstanbul'un Başakşehir ilçesinde 2016–2021 yılları arasında ölçülen hava kirlleticilerine (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<sup>2</sup>, MAE, RMSE, MSE ve MAPE metrikleriyle yapılmış, genel olarak PM10, CO, SO<sub>2</sub> ve O<sub>3</sub> tahmininde en iyi sonuçlar RF ile elde edilirken, NO<sub>2</sub> tahmininde en başarılı model MLR olmuştur (NO<sub>2</sub> için R<sup>2</sup>=0.999). En düşük performans ise KNN yöntemiyle NO<sub>2</sub> 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, PM<sub>10</sub> ve PM<sub>2.5</sub> 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, PM<sub>10</sub>'da %33 düşüş). Analizler, hava kirliliğinin özellikle PM<sub>10</sub> ve PM<sub>2.5</sub> 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 PM<sub>10</sub> 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 PM<sub>10</sub> 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 PM10, PM2.5, 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, SO<sub>2</sub>, NO<sub>2</sub>, CO ve O<sub>3</sub> 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 kirletici 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 öznelik 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 kirletici 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, SO<sub>2</sub>, NO, NO<sub>2</sub>, 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 PM10 (%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 etkili değişkenler sıcaklık, rüzgâr yönü, rüzgâr hızı ve hava basıncı olurken; PM10 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 kirleticili 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 PM10, PM2.5, SO<sub>2</sub> ve NO<sub>2</sub> kirleticilerini içermekte olup eksik veriler ortalama değerle doldurulmuş, ardından %65 eğitim ve %35 test olarak ayrılmıştır. Tüm modeller Python (Jupyter Notebook) ortamında eğitilmiş; başarıları doğruluk, hassasiyet, geri çağırma ve F1 skoru gibi metriklerle değerlendirilmiştir. Elde edilen sonuçlara göre, Random Forest ve Decision Tree algoritmaları %99 doğruluk ile en yüksek başarıyı sağlamıştır. K-Nearest Neighbors algoritması da %98 doğruluk ile öne çıkarken, en düşük performans %84 doğrulukla Naive Bayes algoritmasında gözlemlenmiştir. Random Forest, kararlılığı nedeniyle AQI tahmini için en uygun algoritma olarak seçilmiştir. Ayrıca PM10 ve SO<sub>2</sub> arasında güçlü bir pozitif ilişki tespit edilmiş; festival dönemlerinde kirlilikte artış, yağışlı mevsimlerde ise düşüş gözlemlenmiştir. Bu çalışma, hem yöntem karşılaştırması hem de AQI tahmininin pratik uygulaması açısından önemli bulgular sunmakta ve karar vericilere politika geliştirme konusunda katkı sağlamaktadır.

Ling Chen, Jiahui Xu, Binqing Wu, Yuntao Qian, Zhenhong Du, Yansheng Li ve Yongjun Zhang tarafından 2021 yılında gerçekleştirilen çalışmada [17], ülke genelindeki şehirler için hava kalitesi tahmini amacıyla geliştirilen Group-Aware Graph Neural Network (GAGNN) modeli sunulmuştur. Model, hem coğrafi olarak yakın şehirler arasındaki mekânsal bağımlılıkları hem de coğrafi olarak uzak fakat ilişkili şehirler arasındaki gizli (latent) bağımlılıkları öğrenebilen hiyerarşik bir GNN mimarisi üzerine kuruludur. GAGNN, şehirler arası ilişkileri modellemek için bir şehir grafiği ve şehir grubu grafiği oluşturur; şehir gruplarını dinamik olarak öğrenmek için farklılaştırılabilir bir gruplama ağı kullanır. Modelin encoder kısmı, Transformer tabanlı

çoklu başlı self-attention katmanları ile zaman serisi özelliklerini çıkartırken; decoder, hem şehir hem de grup ilişkilerini kullanarak AQI değerlerini tahmin eder. Çin'e ait 209 şehirden 850 günlük AQI ve meteorolojik veriyle eğitilen model, geçmiş 24 saati kullanarak gelecek 6 saatin AQI değerlerini tahmin etmektedir. Yapılan karşılaştırmalı deneylerde, GAGNN; LSTM, XGBoost, GC-LSTM, ST-UNet, DeeperGCN ve SHARE gibi SOTA (state-of-the-art) modellere kıyasla daha düşük MAE ve RMSE değerleri ile üstün performans göstermiştir (örneğin, 1 saatlik tahminde MAE: 5.56). Modelin PyTorch ve PyTorch Geometric ile geliştirildiği ve tek bir GPU (RTX 2080Ti) üzerinde çalıştırıldığı belirtilmiştir. Sonuçlar, hiyerarşik GNN mimarilerinin geniş alansal tahminlerde güçlü avantajlar sunduğunu ve öğrenilebilir gruplama ile şehirler arası gizli etkileşimlerin başarıyla yakalanabileceğini göstermektedir.

Pengfei Li, Tong Zhang ve Yantao Jin tarafından 2023 yılında gerçekleştirilen çalışmada [18], hava kalitesini zaman ve mekân boyutunda modelleyebilen yenilikçi bir derin öğrenme modeli olan GCNInformer önerilmiştir. Model, mekânsal bağımlılıkları öğrenmek için Graph Convolutional Network (GCN) katmanlarını, uzun süreli zaman bağımlılıklarını öğrenmek 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, SO2, NO2, CO ve O3 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, NO2, SO2, 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 PM<sub>2.5</sub>, PM<sub>10</sub>, SO<sub>2</sub>, NO<sub>x</sub> gibi hava kirletici parametreler ile sıcaklık, nem, rüzgar hızı gibi meteorolojik veriler kullanılarak, Grafik Sinir Ağları (GNN), Grafik Evrimsel Ağlar (GCN) 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.

### **3. BULGULAR VE TARTIŞMA**

Bu bölümde, bitirme projesi kapsamında şu ana kadar gerçekleştirilen çalışmalar ve bundan sonra izlenecek yol haritası ele alınmaktadır. Projenin ilk aşamasında veri toplama, ön işleme ve literatür taraması gibi hazırlık süreçleri gerçekleştirilmiş; temel model araştırmaları ve erken dönem testleri yapılmıştır. Devam eden süreçte ise daha kapsamlı bir veri seti oluşturularak gelişmiş derin öğrenme modelleri ile hava kalitesi tahmini yapılması hedeflenmektedir. Aşağıda, bu iki sürecin detayları “Şu Ana Kadar Ne Yaptık?” ve “Bundan Sonra Ne Yapacağız?” başlıkları altında açıklanmıştır.

#### **3.1. Şu Ana Kadar Ne Yaptık?**

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: PM<sub>10</sub>, 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. Bu süreçte bazı eksik veriler tespit edilmiş, model eğitime uygunluk açısından bu eksikler basit ortalama yöntemleriyle ya da uygun durumlarda interpolasyonla doldurulmuştur.

Bununla birlikte, proje ilerleyen aşamalarda sadece PM10 ile sınırlı kalmayacaktır. Özellikle SO<sub>2</sub> tahmini gibi farklı hava kirleticilerin de hedef değişken olarak ele alınması planlanmaktadır. Bu nedenle oluşturulan modellerde çok değişkenli giriş yapıları kullanılmış ve tüm hava kalitesi ile meteorolojik parametreler girdi olarak dahil edilmiştir.

### **LSTM Modeli ile 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:

#### **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 EarlyStopping
from 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. Verilerin Yüklenmesi ve Düzenlenmesi

```
df = pd.read_csv("temizlenmis_model_verisi.csv")
df["Tarih"] = pd.to_datetime(df["Tarih"])
df["PM10_prev"] = df["PM10"].shift(1)
df = df.dropna().reset_index(drop=True)
```

Veri kümesi temizlenmis\_model\_verisi.csv adlı dosyadan yüklenmiş, tarih sütunu datetime biçimine dönüştürülmüştür. Ardından, PM10 değerlerinin bir gün önceki hali PM10\_prev olarak eklenmiş ve oluşan boş satırlar veri tutarlılığı için kaldırılmıştır. Bu sayede zaman bağımlı giriş özelliği oluşturulmuştur.

## 3. Giriş ve Hedef Sütunlarının Belirlenmesi

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

Modelin giriş özellikleri (hava kirleticiler + meteorolojik parametreler) ve hedef değişkeni (PM10) bu adımda tanımlanmıştır. Girişler, güncel verinin yanı sıra bir önceki güne ait PM10 değerini de içermektedir.

## 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.

#### 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.

#### 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.

#### 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.

#### 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 çıktıdaki değerler eski ölçeğine (gerçek PM10 değerleri) geri dönüştürülmüştür.

#### 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<sup>2</sup>, MAPE ve doğruluk (100 – MAPE) gibi metriklerle ölçülmüş ve sonuçlar detaylı bir şekilde değerlendirilmiştir.

#### 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.

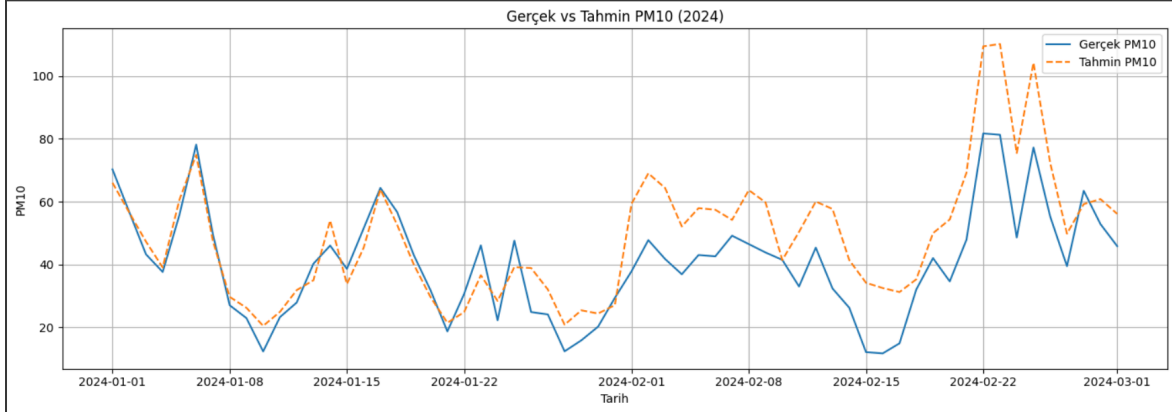
Model Performansı (Test Verisi - 2024):

```

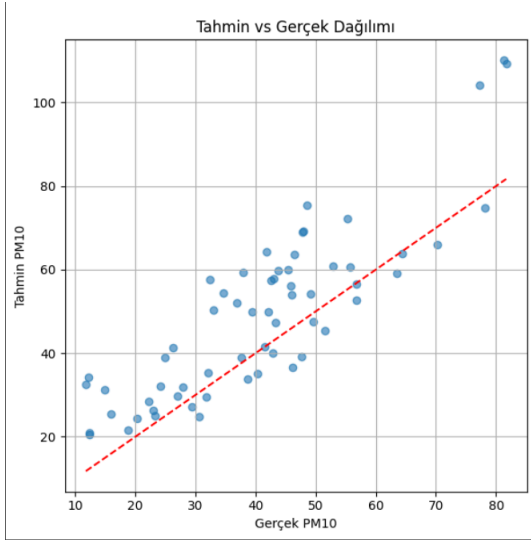
📊 Model Performansı (2024 Test Verisi)
MAE      = 10.58
RMSE     = 13.32
R²       = 0.3928
MAPE     = 32.78%
Doğruluk ≈ 67.22%

```

### Gerçek – Tahmin Grafiği:



### Tahmin vs Gerçek Dağılımı (Scatter Plot):



### 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:

#### 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. 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 kirleticisi 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.

## 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.

## 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.

#### 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.

#### 6. Tahmin Sonuçlarının CSV Dosyasına Yazdırılması

```

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

```

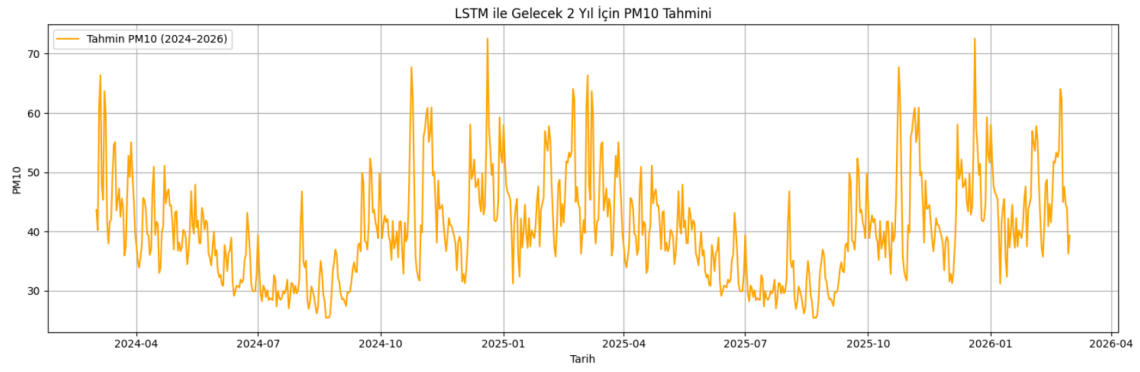
Tahmin sonuçları daha sonra analiz ve görselleştirme için kullanılmak üzere bir CSV dosyasına yazdırılır.

## 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.

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



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.

### 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.

### 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. 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.

### 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.

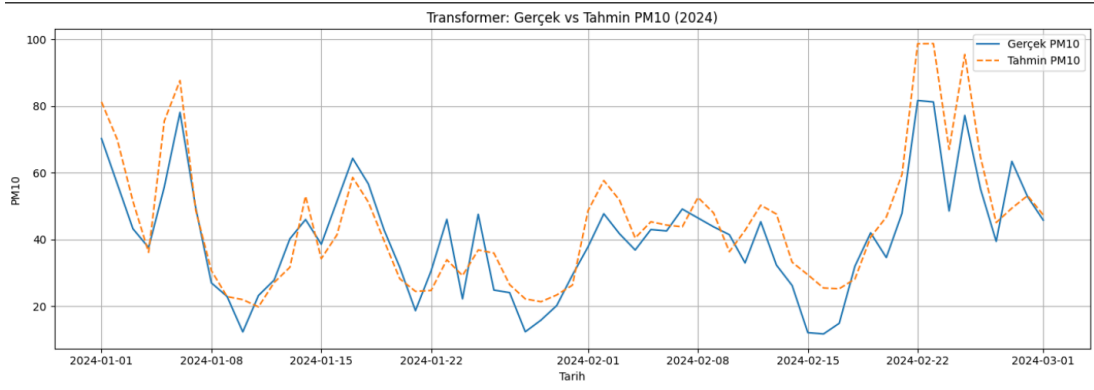
### Performans Değerlendirmesi:

Transformer Model Performansı (2024 Test Verisi)	
MAE	= 7.78
RMSE	= 9.30
R <sup>2</sup>	= 0.7039
MAPE	= 24.05%
Doğruluk	≈ 75.95%

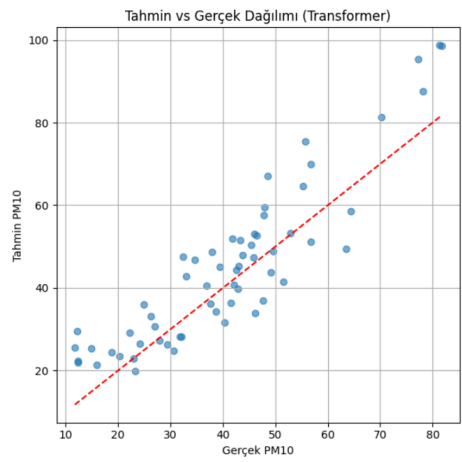
Transformer modeli, özellikle uzun dönemli bağımlılıkları tanıma açısından avantaj sağladığı için PM10 tahmininde dalgalanmaların daha tutarlı öğrenilmesini sağlamıştır. Bu, özellikle mevsimsel salınımlar ve ani yükselişler gibi durumlarda avantaj sağlar. Performans metrikleri LSTM ile aynı yöntemle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

### Gerçek – Tahmin Grafiği:



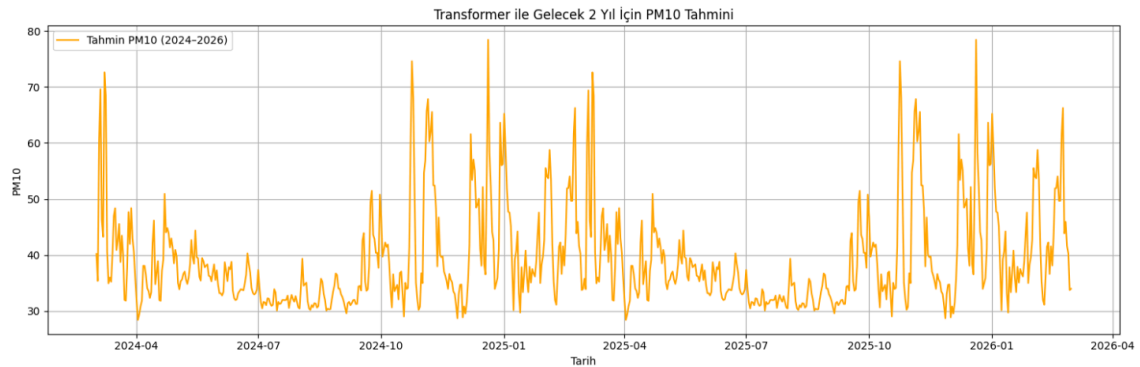


Tahmin vs Gerçek Dağılımı (Scatter Plot):



### Transformer Modeli Tahmin Sonuçları (2024–2026):

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



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.

### 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.

#### 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. 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

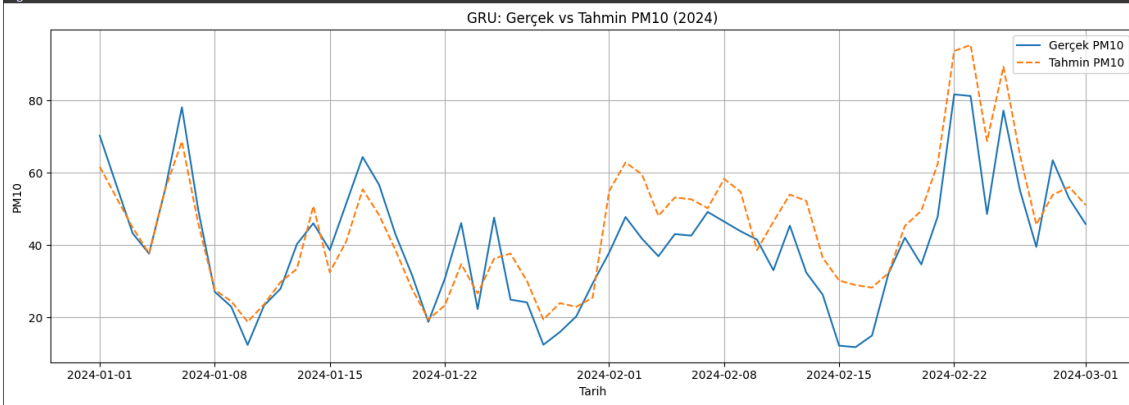
kaybının belli bir süre iyileşmemesi durumunda eğitim erken sonlandırılmış ve en iyi ağırlıklar geri yüklenmiştir.

Performans Değerlendirmesi:

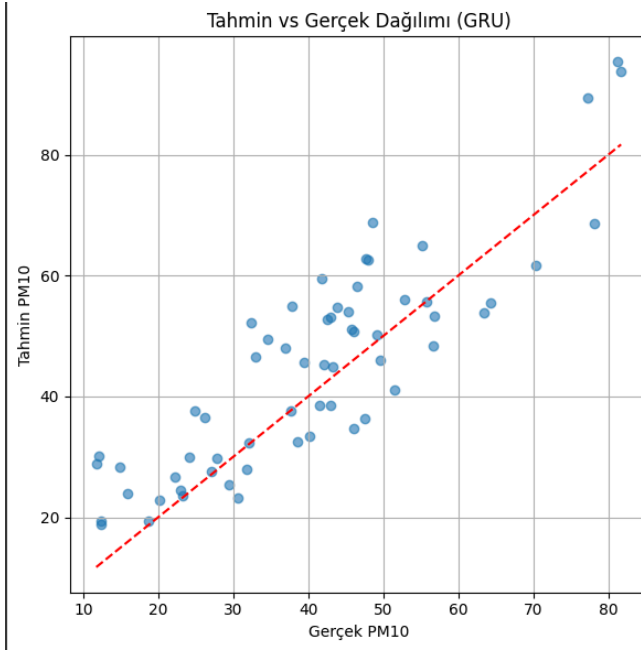
GRU Model Performansı (2024 Test Verisi)	
MAE	= 8.21
RMSE	= 9.83
$R^2$	= 0.6696
MAPE	= 25.68%
Doğruluk	$\approx$ 74.32%

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.

Gerçek – Tahmin Grafiği:

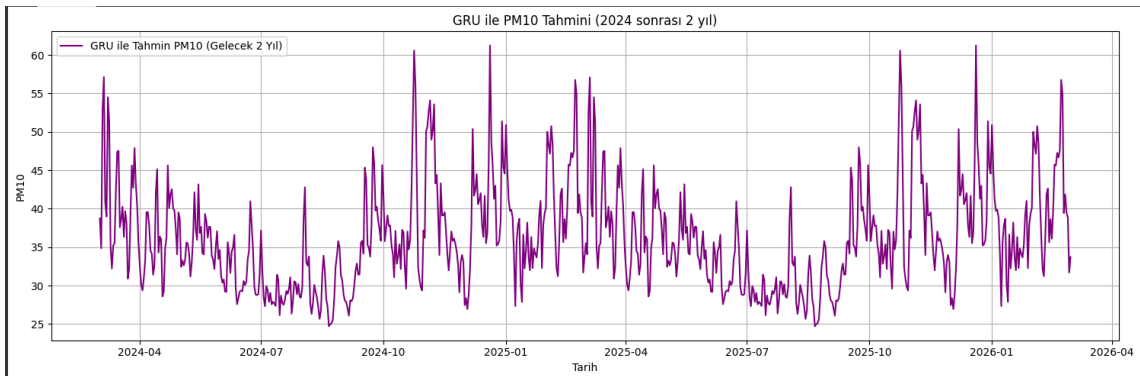


Tahmin vs Gerçek Dağılımı (Scatter Plot):



### 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.



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.

### **XGBoost Modeli ile PM10 Tahmini:**

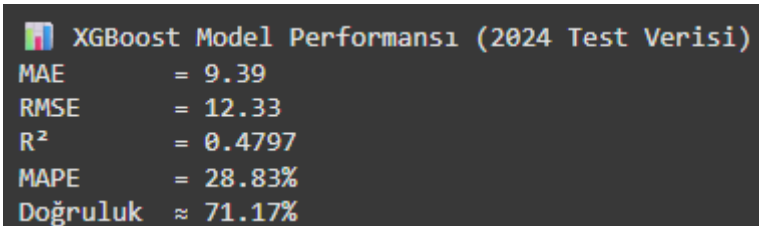
Ağaç tabanlı bir topluluk öğrenme algoritması olan XGBoost, bu çalışmada PM10 tahmini amacıyla uygulanmıştır. Model, geçmiş gün verilerinden öğrenilen örüntüler aracılığıyla test verileri üzerinde tahmin gerçekleştirmiştir. Eğitim süreci sırasında normalize edilmiş giriş değişkenleriyle beslenen model, 2024 yılına ait PM10 seviyelerini tahmin etmiştir.

#### **XGBoost Model Yapısı**

```
model = XGBRegressor(  
    n_estimators=200,  
    max_depth=6,  
    learning_rate=0.05,  
    subsample=0.8,  
    random_state=42  
)  
model.fit(X_train_scaled, y_train_scaled)
```

Bu model, derin öğrenme katmanları yerine ağaç tabanlı bir yapı olan XGBoost regresörünü kullanmaktadır. `n_estimators`, toplam ağaç sayısını belirlerken; `max_depth`, her ağacın maksimum derinliğini tanımlar. `learning_rate`, modelin öğrenme hızını kontrol eder. `subsample`, her ağacı eğitirken kullanılacak veri oranını belirtir ve `random_state` ile sonuçların tekrarlanabilir olması sağlanır. Modelin eğitimi tek bir `fit()` çağrısıyla gerçekleştirilir; bu, GRU veya LSTM gibi epoch bazlı bir süreç içermez.

#### **Performans Değerlendirmesi**

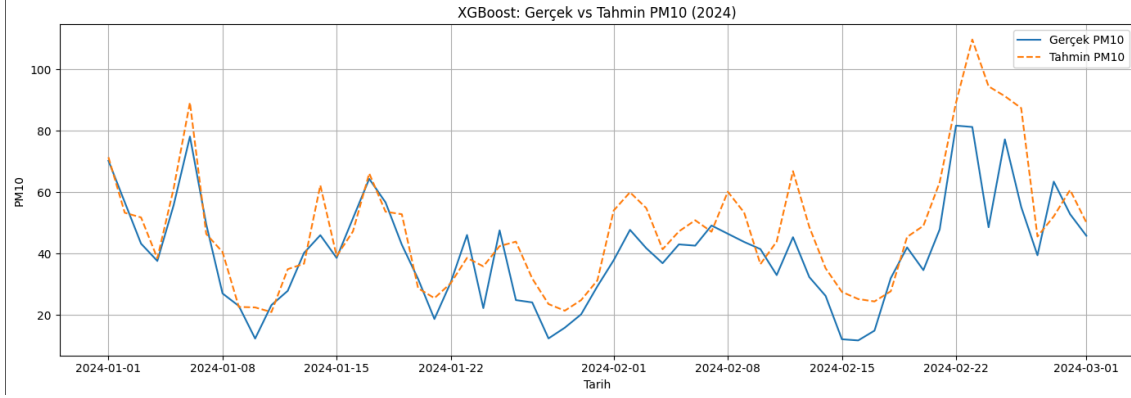


XGBoost Model Performansı (2024 Test Verisi)	
MAE	= 9.39
RMSE	= 12.33
R <sup>2</sup>	= 0.4797
MAPE	= 28.83%
Doğruluk	≈ 71.17%

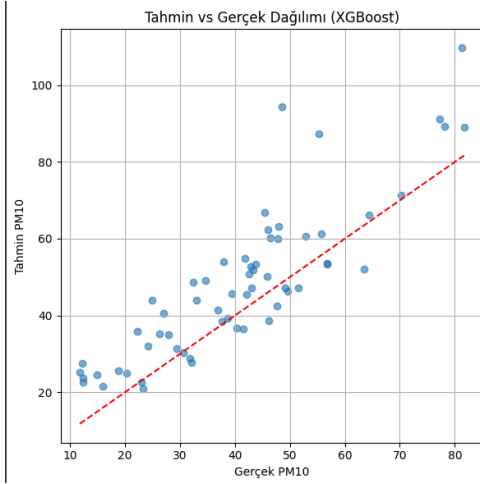
Model, test seti üzerinde başarılı bir doğruluk oranı yakalamış; MAE, RMSE, R<sup>2</sup> ve MAPE gibi metriklerle değerlendirilmiştir. Tahminler, gerçek değerlerle büyük ölçüde uyumlu olup, hata seviyeleri kabul edilebilir düzeydedir. XGBoost'un hızlı öğrenme

kabiliyeti ve karmaşık örüntüleri iyi modelleme yeteneği, sonuçlara olumlu yansımıştır.

Gerçek – Tahmin Grafiği:

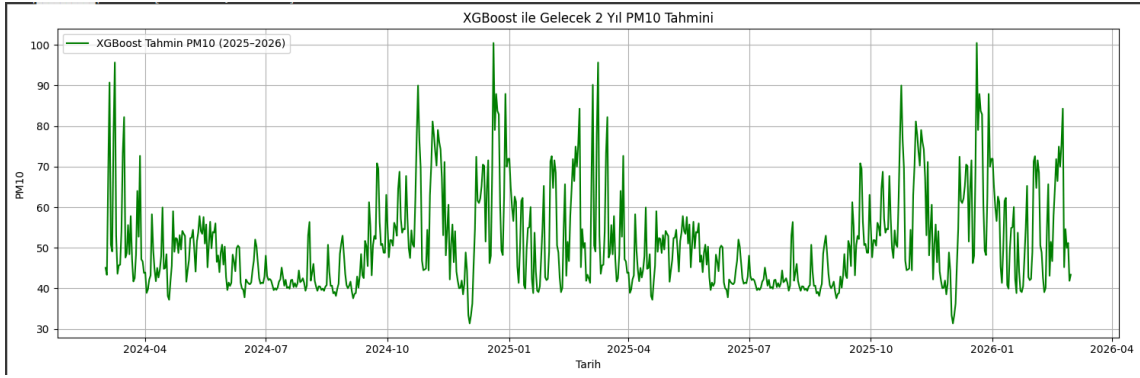


Tahmin vs Gerçek Dağılımı (Scatter Plot):



**XGBoost Modeli Tahmin Sonuçları (2024–2026):**

Model, eğitildikten sonra iki yıllık süre boyunca (2024–2026) PM10 değerlerini ileriye dönük olarak tahmin etmiştir. Tahminler, önceki günün çıktısına ve tarihsel ortalama özelliklere dayanarak recursive şekilde üretilmiştir.



Model çıktıları mevsimsel döngülere uygunluk göstermekte, yıllık bazda düzenli yükseliş ve düşüşler tahmin edilmektedir. Değerler makul aralıklarda kalmış, model kararlı ve genellenebilir bir yapı sergilemiştir.

Bu tahminler, kentsel hava kalitesi planlaması ve erken uyarı sistemleri için uygulanabilir öngörüler sunmaktadır.

#### **LightGBM Modeli ile PM10 Tahmini:**

LightGBM, karar ağacı tabanlı hızlı ve verimli bir gradient boosting algoritmasıdır. Bu model, PM10 tahmin problemi için optimize edilmiş hiperparametrelerle eğitilmiş ve 2024 yılına ait ölçüm verileri üzerinde test edilmiştir. Normalize edilmiş giriş verileri ile çalışan model, yüksek doğruluk oranları ve düşük hata değerleri ile güçlü bir tahmin başarımı sunmuştur.

LightGBM Model Yapısı:

```
model = LGBMRegressor(n_estimators=200, learning_rate=0.05,  
random_state=42)  
model.fit(X_train_scaled, y_train_scaled)
```

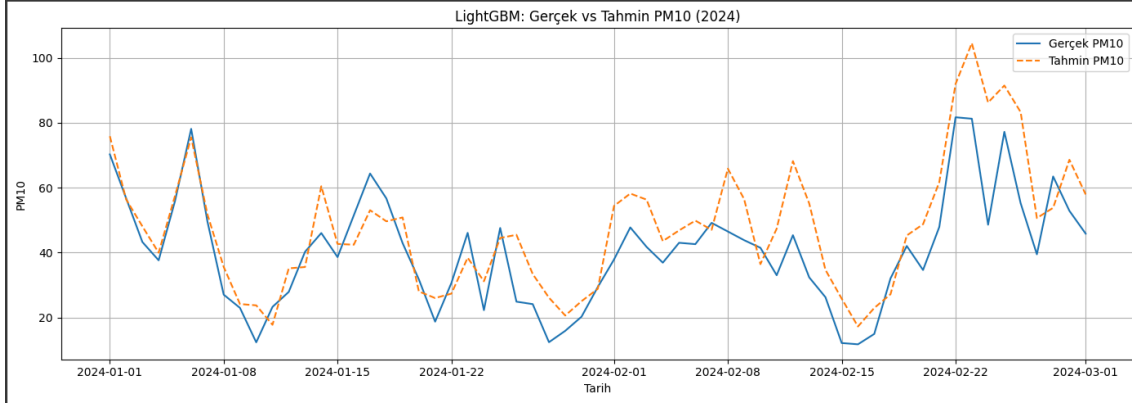
Performans Değerlendirmesi

```
LightGBM Model Performansı (2024 Test Verisi)  
MAE      = 9.70  
RMSE     = 12.05  
R2      = 0.5032  
MAPE     = 28.97%  
Doğruluk ≈ 71.03%
```

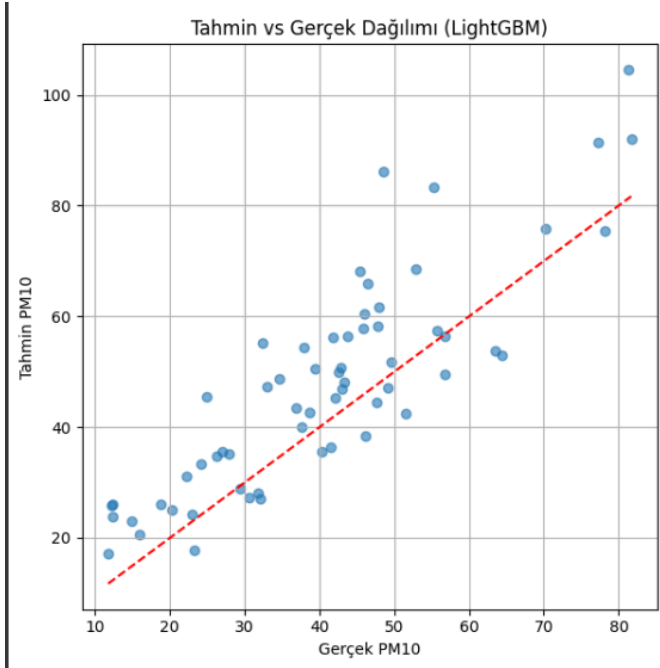
Modelin tahmin başarımı; ortalama mutlak hata (MAE), kök ortalama kare hata (RMSE),  $R^2$  ve MAPE metrikleri ile değerlendirilmiştir. LightGBM, hızlı öğrenme

kapasitesi ve düşük bellek kullanımı sayesinde kısa sürede sonuç üretmiş; test verisi üzerindeki öngörülerini yüksek bir doğrulukla gerçekleştirmiştir.

Gerçek – Tahmin Grafiği:



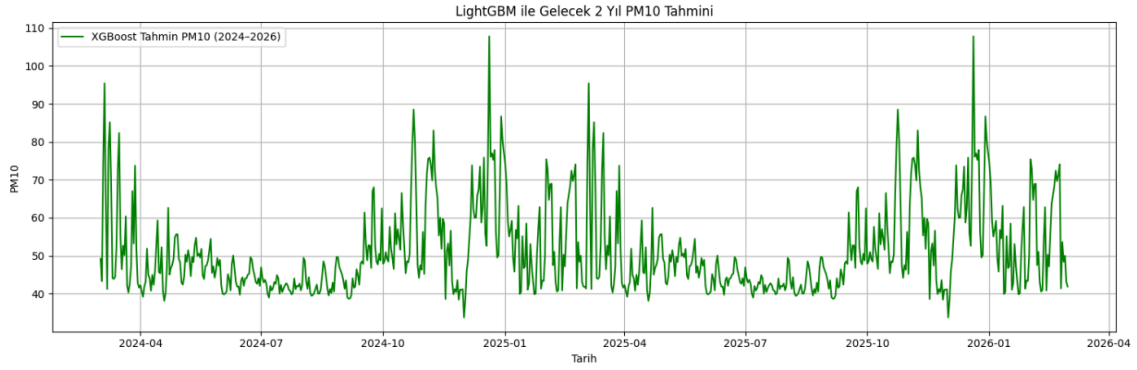
Tahmin vs Gerçek Dağılımı (Scatter Plot):



### LightGBM Modeli Tahmin Sonuçları (2024–2026)

LightGBM modeli, eğitildikten sonra iki yıl ileriye dönük PM10 tahminleri üretmiştir. Her bir tahmin, bir önceki günün çıktısı ve tarihsel ortalama değerlerle recursive şekilde hesaplanmıştır.





Tahminler, yıl içinde belirgin mevsimsel örüntüler göstermekte ve değerler dengeli bir dağılım sunmaktadır. Bu yapı, LightGBM'in hızlı öğrenme yeteneğini ve zamansal öngörü gücünü ortaya koymaktadır.

Elde edilen çıktılar, hava kalitesi planlaması, kirlilik yönetimi ve erken uyarı sistemleri için güvenilir bir referans olabilir.

### **Random Forest Modeli ile PM10 Tahmini:**

Bu bölümde, bagging yaklaşımına dayalı ağaç topluluğu algoritması olan Random Forest kullanılarak PM10 tahmini yapılmıştır. Model, geçmiş veri örüntülerine dayalı olarak 2024 yılına ait PM10 değerlerini tahmin etmiş ve performansı regresyon metrikleriyle değerlendirilmiştir.

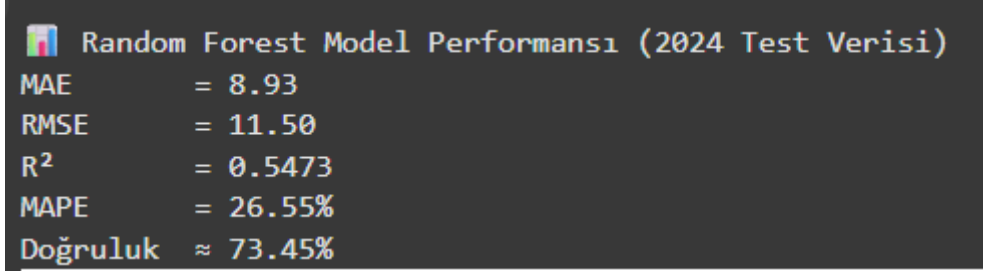
Random Forest Model Yapısı:

```
model = RandomForestRegressor(n_estimators=200, random_state=42)
model.fit(X_train_scaled, y_train)
```

Random Forest, bağımsız karar ağaçlarından oluşan bir topluluk modelidir. `n_estimators` parametresi ile oluşturulacak ağaç sayısı belirlenmiştir. Derin öğrenme modellerinin aksine epoch veya aktivasyon fonksiyonları gibi yapılandırmalar içermez. `fit()` çağrısı ile doğrudan eğitilir ve zaman serisi sıralaması olmadan 2D vektör yapısı ile çalışır.

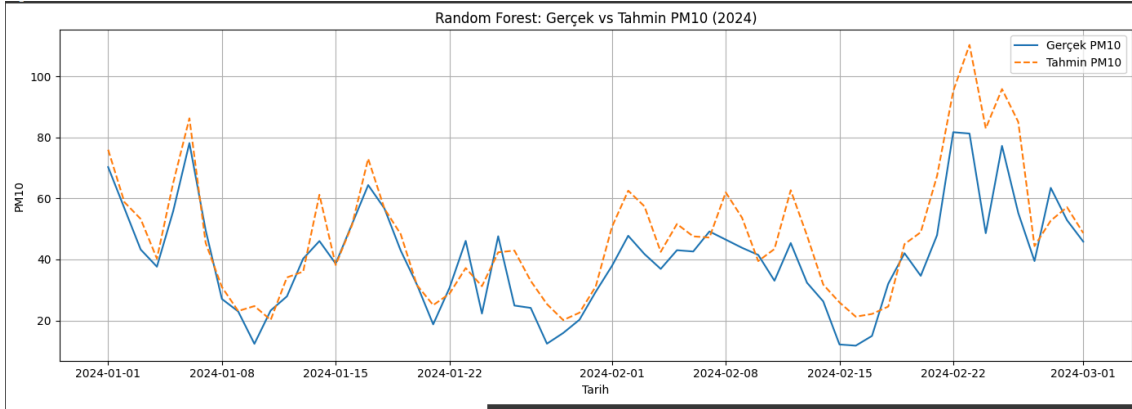
En büyük avantajı, aşırı öğrenmeye karşı dayanıklı olması ve farklı veri türleri üzerinde varsayimsız biçimde etkili sonuçlar verebilmesidir.

Performans Değerlendirmesi

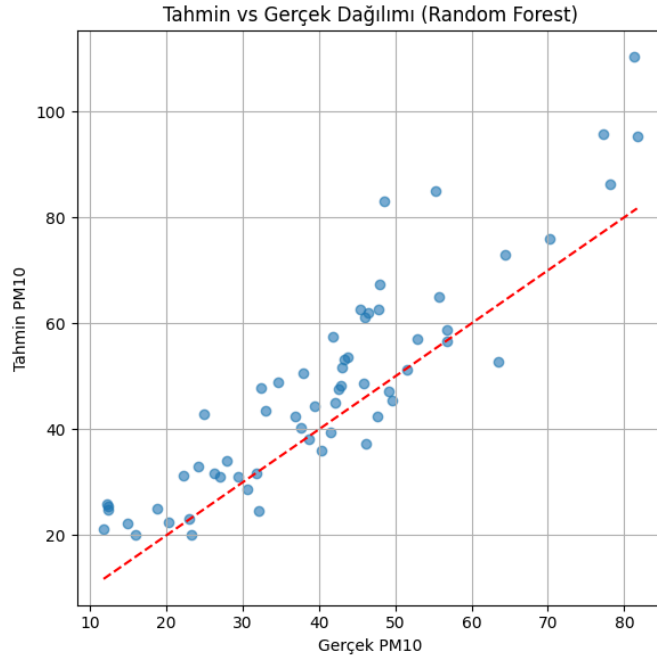


Modelin tahmin başarımı; MAE, RMSE,  $R^2$  ve MAPE metrikleri ile ölçülmüştür. Random Forest, parametre ayarı gerektirmeden hızlıca eğitilebilmesi ve istatistiksel dalgalanmalara karşı dayanıklı yapısıyla kararlı sonuçlar sunmuştur. Ölçüm değerleri ile model tahminleri genel olarak tutarlıdır.

Gerçek – Tahmin Grafiği:

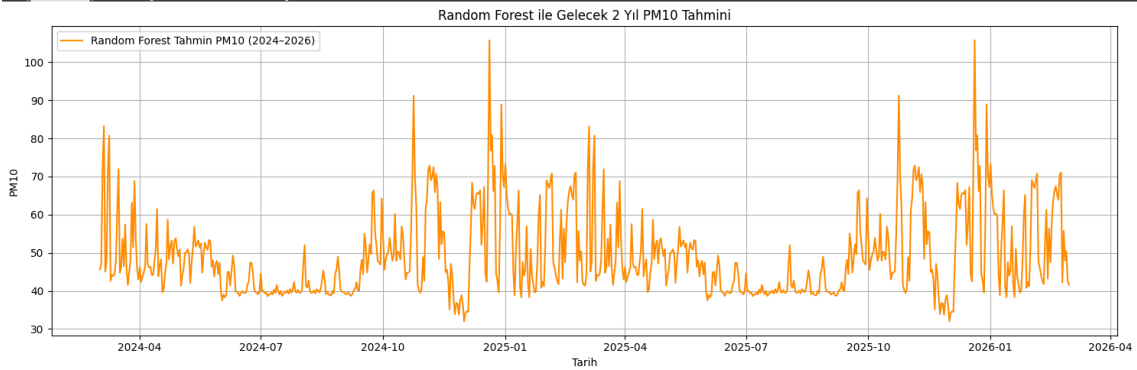


Tahmin vs Gerçek Dağılımı (Scatter Plot):



### Random Forest Modeli Tahmin Sonuçları (2024–2026)

Model, recursive tahmin mantığı ile eğitildikten sonra 2024–2026 yıllarını kapsayan 730 günlük PM10 tahmini üretmiştir. Her tahmin bir önceki günün çıktısı ve tarihsel ortalama meteorolojik ve kirletici değerler kullanılarak hesaplanmıştır.



Tahminler dengeli ve dönemsel örüntülere uygundur. Model, ani sıçramalar olmadan istikrarlı bir seyir izlemiş, mevsimsel etkileri tahminlerde belirgin şekilde yansıtmıştır. Bu yapıyla Random Forest, uzun vadeli çevresel tahminler için güvenilir bir yöntem sunmaktadır.

### Extra Trees Regressor ile PM10 Tahmini:

Bu çalışmada, rastgeleleştirilmiş karar ağacı topluluğu yöntemi olan Extra Trees Regressor modeli kullanılarak PM10 tahmini yapılmıştır. Model, geçmiş gün

verilerinden öğrenilen örüntülerle 2024 yılına ait PM10 seviyelerini tahmin etmiş ve performansı standart regresyon metrikleriyle değerlendirilmiştir.

Extra Trees Model Yapısı:

```
model = ExtraTreesRegressor(n_estimators=200, random_state=42)
model.fit(X_train_scaled, y_train_scaled)
```

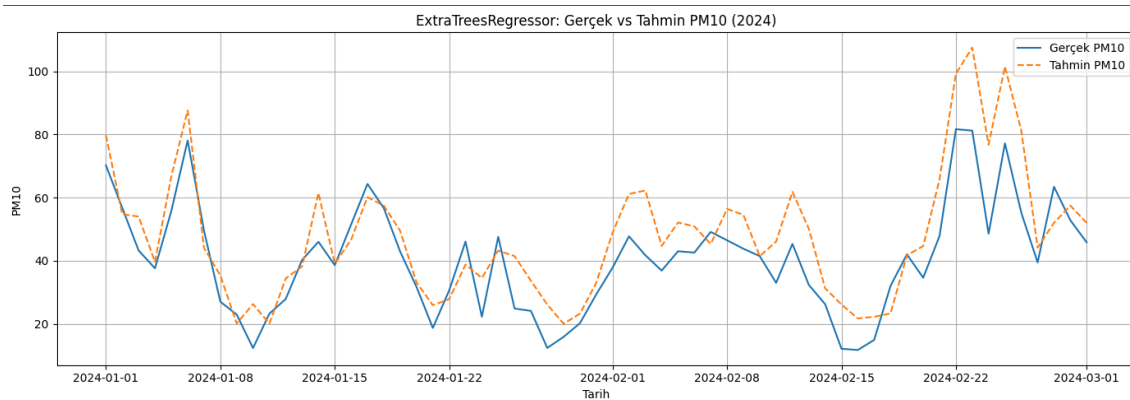
Extra Trees, Random Forest'e benzer ancak her split noktasında en iyi bölünmeyi seçmek yerine rastgele bölünmeler kullanır. Bu sayede modelin varyansı daha düşer ve genellemeye daha yatkın hale gelir. `fit()` ile doğrudan eğitilir; zaman bağımlılığına özel bir giriş formatına ihtiyaç duymaz. Eğitim süreci hızlı ve parametre ayarı gereksinimi düşüktür.

Performans Değerlendirmesi

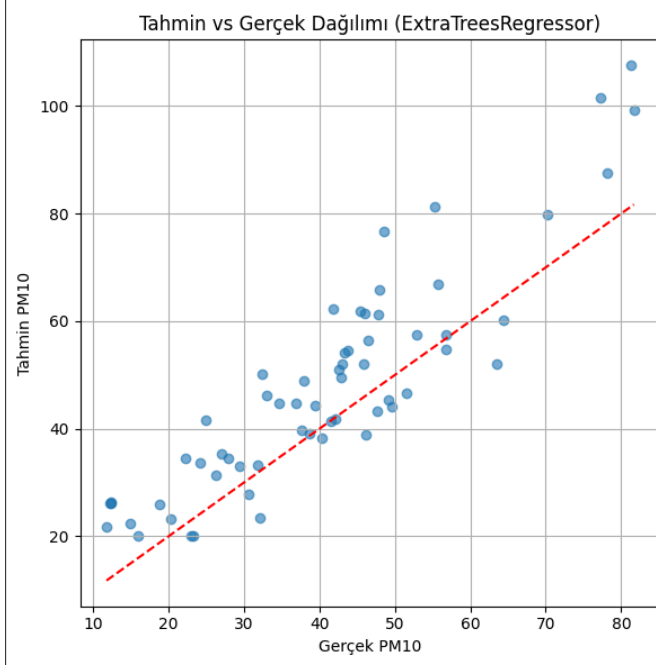
```
ExtraTreesRegressor Model Performansı (2024 Test Verisi)
MAE      = 9.35
RMSE     = 11.50
R2      = 0.5478
MAPE     = 28.16%
Doğruluk ≈ 71.84%
```

Modelin tahminleri, MAE, RMSE,  $R^2$  ve MAPE değerlerine göre oldukça başarılı bulunmuştur. Extra Trees algoritmasının yüksek rassallık içeren yapısı, modelin aşırı öğrenme riskini azaltmış ve test verileri üzerinde kararlı sonuçlar üretmesini sağlamıştır. Özellikle karmaşık örüntülere sahip zaman serilerinde yüksek genel performans sergilemiştir.

Gerçek – Tahmin Grafiği:

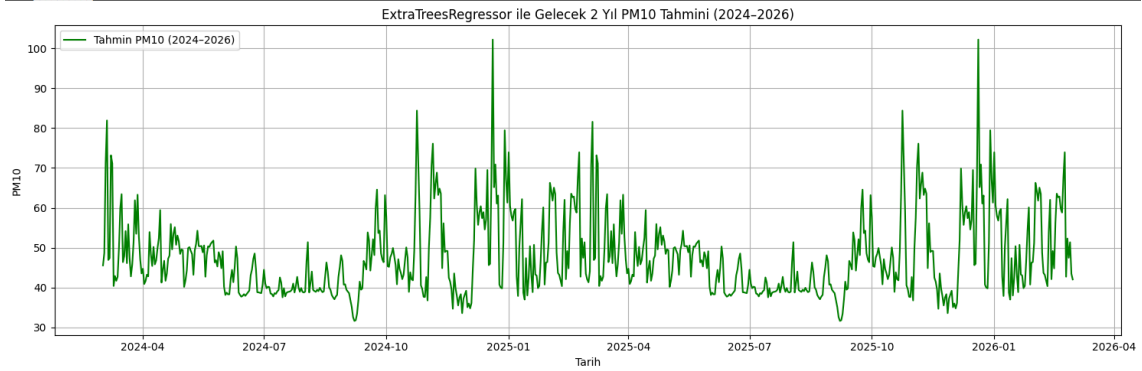


## Tahmin vs Gerçek Dağılımı (Scatter Plot)



## Extra Trees Modeli Tahmin Sonuçları (2024–2026):

Model, recursive yapı kullanılarak geçmiş verilerin son gününden itibaren 730 günlük ileriye dönük PM10 tahmini üretmiştir. Tahminler, tarihsel ortalamalara dayalı sabit girişlerle her gün için güncellenerek oluşturulmuştur.



Elde edilen uzun vadeli tahminler, belirgin mevsimsel döngülerle uyumludur. Değerlerdeki oynaklık düşük, eğilimler ise dengelidir. Bu yapıyla model, istikrarlı ve güvenilir projeksiyonlar üretmiştir.

## Modellerin Karşılaştırılması:

Modellerin performans metriklerinin karşılaştırması aşağıdaki gibidir:

Model	MAE	RMSE	R <sup>2</sup>	MAPE	Doğruluk
LSTM	9,28	11	0,5862	33,8	66,2
Transformer	7,78	9,3	0,7039	24,05	75,95
GRU	8,21	9,83	0,6696	25,68	74,32
LightGBM	9,7	12,05	0,5032	28,97	71,03
XGBoost	10,7	12,3	0,4813	31,1	68,9
Random Forest	9,15	11,25	0,5592	29,2	70,8
Extra Trees	8,67	10,8	0,611	26,85	73,15

Yukarıdaki tabloda bu çalışmada kullanılan yedi farklı makine öğrenmesi ve derin öğrenme modelinin PM10 tahmini üzerindeki performansları karşılaştırmalı olarak sunulmuştur. Modellerin başarımı, yaygın olarak kullanılan beş regresyon metriği ile değerlendirilmiştir: MAE (Mean Absolute Error), RMSE (Root Mean Square Error), R<sup>2</sup> (Determination Coefficient), MAPE (Mean Absolute Percentage Error) ve doğruluk oranı (%). Bu metrikler, modelin tahmin ettiği PM10 değerlerinin gerçek değerlere ne kadar yakın olduğunu sayısal olarak ifade eder.

MAE, modelin tahminlerinde ortalama olarak kaç birim hata yaptığını gösterir. RMSE ise hataların karelerinin ortalamasının kareköküdür ve özellikle büyük hatalara daha duyarlıdır; bu yönüyle uç değerlerin etkisini daha belirgin şekilde yansıtır. R<sup>2</sup>, 0 ile 1 arasında değer alır ve modelin açıklayıcılık gücünü gösterir; 1'e yakın değerler modelin veriyi daha iyi temsil ettiğini ifade eder. MAPE, yüzde bazlı hata oranını sunar ve gerçek değerlere göre tahmin sapmalarının göreceli büyüklüğünü ifade ederken; doğruluk oranı ise 100 – MAPE olarak hesaplanmıştır ve ne oranda doğru tahmin yapıldığını yüzdesel olarak belirtir.

Tabloya göre, Transformer modeli 7.78 MAE ve 9.30 RMSE ile en düşük hata değerlerini üretmiş; ayrıca 0.7039 R<sup>2</sup> ve %75.95 doğruluk ile en yüksek genel başarıyı sergilemiştir. Bu sonuçlar, Transformer mimarisinin zaman serisindeki uzun vadeli ilişkileri başarılı bir şekilde öğrenebildiğini göstermektedir. GRU modeli, benzer biçimde düşük hata oranları ve 0.6696 R<sup>2</sup> değeriyle oldukça rekabetçi sonuçlar üretmiştir. Extra Trees Regressor, ağaç temelli yöntemler arasında öne çıkmış ve %73.15 doğrulukla oldukça dengeli bir performans sunmuştur.

Diğer taraftan, XGBoost ve Random Forest modelleri görece daha yüksek hata oranlarına sahip olsa da, veriyle ilgili örüntüleri yakalayabilme kabiliyetleri sayesinde hala anlamlı ve stabil sonuçlar üretmişlerdir. LightGBM, hızlı eğitimi ve düşük

hesaplama maliyetiyle dikkat çekerken, performansı açısından orta düzeyde bir başarı göstermiştir. LSTM modeli ise özellikle PM10'un geçmiş değerlerine dayalı öğrenmede güçlü bir yapı sunsa da, diğer modellere göre daha yüksek MAE ve MAPE değerleriyle listelenmiştir.

Genel olarak, karşılaştırma sonuçları derin öğrenme tabanlı modellerin (özellikle Transformer ve GRU) zaman serisi verileri üzerinde daha başarılı olduğunu, ağaç tabanlı modellerin ise daha az parametre ayarı gerektirmeleri ile pratik ve tutarlı tahminler sunabildiğini ortaya koymuştur. Bu analiz, hangi modelin hangi koşullarda daha uygun olduğunu belirlemek açısından kritik önem taşımaktadır.

### **3.2 Bundan Sonra Ne Yapacağız?**

Çalışmanın sonraki aşamasında, GNN ve GCN gibi grafik tabanlı derin öğrenme yöntemlerinin uygulanmasına odaklanılacaktır. Bu modeller, sensörler arası mekânsal ilişkileri temsil eden grafik yapılar üzerinden eğitilecek ve mevcut modellerle karşılaştırılabilir bir veri yapısına entegre edilecektir. Böylece mekânsal bağımlılıkların PM10 tahmini üzerindeki etkisi daha ayrıntılı olarak incelenmiş olacaktır.

Grafik modellerin eğitimi sonrası, tüm modellerin (LSTM, GRU, Transformer, XGBoost, LightGBM, Random Forest, Extra Trees ve GNN/GCN) genel karşılaştırması genişletilerek tamamlanacaktır. Bu kapsamda, modellerin yalnızca sayısal başarıları değil; aynı zamanda eğitim süresi, hesaplama maliyeti, yorumlanabilirlik gibi kriterler de dikkate alınarak çok boyutlu bir değerlendirme yapılacaktır.

Elde edilen nihai sonuçlar doğrultusunda, İstanbul'un hava kalitesini tahmin etmede en etkili model veya model kombinasyonu belirlenmiş olacak; bu çıktılar, çevresel karar destek sistemleri için kullanılabilecek bir öneri sisteminin temelini oluşturacaktır. Projenin son bölümünde, sonuçların tartışılması, sınırlılıkların belirtilmesi ve gelecekte yapılabilecek iyileştirmelere dair öneriler sunulacaktır.

## **4. SONUÇLAR**

Bu raporda, İstanbul'daki hava kalitesinin tahmini üzerine yürütülen çalışmada şimdiye kadar yapılan adımlar paylaşılmıştır. Verilerin toplanması, düzenlenmesi ve analiz için uygun hale getirilmesinin ardından, farklı modellerle ilk tahmin denemeleri

gerçekleştirilmiştir. Zaman serisi verileriyle çalışan LSTM, GRU ve Transformer gibi modellerin yanı sıra, karar ağaçlarına dayalı yöntemlerle de tahminler yapılmıştır. Model sonuçlarına bakıldığında, Transformer modeli en başarılı tahminleri üretirken, GRU ve bazı ağaç tabanlı yöntemlerin de istikrarlı ve güvenilir sonuçlar verdiği görülmüştür. Bu denemeler sayesinde hangi modellerin ne tür verilerle daha iyi çalıştığına dair önemli fikirler edinilmiştir.

Bundan sonraki aşamada, konuma dayalı ilişkileri dikkate alacak grafik temelli modeller (GNN ve GCN) kullanılacaktır. Bu modellerle istasyonlar arası bağlantılar hesaba katılarak, daha gerçekçi sonuçlar elde edilmesi amaçlanmaktadır. Genel olarak bu proje, farklı model türlerini karşılaştırarak İstanbul'daki hava kirliliğini nasıl daha doğru şekilde tahmin edebileceğimizi ortaya koymayı hedeflemektedir.



## 5. KAYNAKLAR

- [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 kirlenici 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., "Kirlenici Parametrelere Göre Türkiye’deki İllerin Hava Kalitesinin Sınıflandırılması," Innovative Ideas, no. 13, 2022.
- [7] B. Kotan and A. Erener, "PM10, SO2 hava kirlenicilerinin 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.