

# Kocaeli İli İçin Hava Durumu Tahmin Modellerinin Oluşturulması ve Verimlilik Analizi

Gülsüm Demir  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
İstanbul, Türkiye  
demirr.gulsumm@gmail.com

Utku Genç  
Bilişim Sistemleri Mühendisliği  
Kocaeli Üniversitesi  
İstanbul, Türkiye  
utkugenc2003@gmail.com

**Özet—** Bu çalışmada, Kocaeli ili için geçmiş hava durumu verileri kullanılarak sıcaklık, nem, rüzgar hızı gibi temel iklim parametrelerinin zaman serisi tahmini gerçekleştirilmiştir. Günlük hava durumu verileri üzerine yoğunlaşan bu çalışmada, Reformer, Informer gibi modeller kullanılarak gelecekteki 7 gün için değerlerin tahmini yapılmıştır. Modellerin performansı, MSE, MAPE, MAE, RMSE ve R-Squared gibi değerlendirme metrikleri ile analiz edilmiştir. Ayrıca, tahminlerin doğruluğunu artırmak için veri temizleme, trend ve mevsimsellik analizi gibi ön işleme adımları uygulanmıştır. Bu çalışma, Kocaeli ili özelinde zaman serisi tahmininde modern derin öğrenme yaklaşımlarının etkinliğini göstermeyi amaçlamaktadır.

**Anahtar Kelimeler—** Zaman serisi tahmini, hava durumu tahmini, derin öğrenme modelleri, Reformer, Informer, performans metrikleri, MSE, MAPE, MAE, RMSE, R-Squared

## I. Giriş

Bu çalışma, Kocaeli ilinin günlük hava durumu verileri üzerinde zaman serisi modellerini kullanarak geleceğe yönelik tahminler yapmayı amaçlamaktadır. Sıcaklık, nem, rüzgar hızı gibi iklim parametreleri, Reformer, Informer, TFT ve diğer modern zaman serisi modelleriyle analiz edilmiştir. Bu tahminler, çevresel planlama, kısa vadeli tarımsal stratejiler ve iklim değişikliği çalışmalarına ışık tutacak şekilde modellenmiştir.

Veri toplama ve düzenleme süreci, günlük hava durumu verilerinin doğru ve tutarlı bir şekilde işlenmesini sağlamıştır. Bu süreçte, çevrim içi kaynaklardan Kocaeli bölgesine ait sıcaklık, nem, rüzgar hızı, basınç gibi parametreler toplanmış ve temizlenmiştir. Toplanan veriler, modellerin performansını artırmak için eksik değerlerin doldurulması, tutarsızlıkların giderilmesi ve uygun formatlara dönüştürülmesi gibi işlemlerden geçirilmiştir.

Veri toplama düzenleme hakkında daha fazla bilgiye ulaşmak için lütfen bağlantıyı ziyaret ediniz. Bağlantıda daha detaylı bilgiler bulunmaktadır:  
<https://github.com/Utku-Genc/WeatherWebScraping>

## II. PROJE GEREKSİNİMLERİ

### A. Gerekli Kütüphaneler ve Kurulum

#### 1) Gerekli Python Kütüphaneleri

##### a) Veri İşleme ve Analiz:

- **numpy:** Çok boyutlu dizilerle matematiksel işlemler.
- **pandas:** Veri manipülasyonu ve analizi.

##### b) Veri Görselleştirme:

- **matplotlib:** 2D grafikler ve görselleştirme.

- **seaborn:** İstatistiksel veri görselleştirme.

#### c) Zaman Serisi ve Modelleme:

- **scikit-learn:** Veri ön işleme ve performans metrikleri hesaplama.
  - **statsmodels:** Zaman serisi analizi ve decomposition işlemleri.
  - **torch** ve **torchvision:** PyTorch tabanlı derin öğrenme modellerinin geliştirilmesi.
  - **tensorflow:** Modellerin alternatif derin öğrenme platformu olarak eğitimi.
- #### d) Ek Kütüphaneler:
- **time:** Eğitim süresini ölçmek ve zaman takibi yapmak için kullanılır.
  - **pyro:** Bayesian yöntemlerini uygulamak için bir kütüphanedir. Probabilistik modelleme sağlar.
  - **lightning.pytorch:** Model eğitimini hızlandırmak ve düzenlemek için bir framework sağlar.
  - **pytorch\_forecasting:** Temporal Fusion Transformer gibi zaman serisi modellerini kolayca uygulamak için kullanılan bir PyTorch tabanlı kütüphanedir.

### 2) Kütüphanelerin Kurulumu

#### a) Kurulum Adımları

Gerekli tüm kütüphaneleri yüklemek için aşağıdaki adımları izleyebilirsiniz:

1. **Python** ve **pip:** Python 3.8 veya üstü bir sürüm ve pip paket yöneticisi sistemde yüklü olmalıdır.
2. **Kütüphanelerin Toplu Kurulumu:** Gerekli kütüphaneleri toplu olarak kurmak için şu komutu çalıştırabilirsiniz:  
`!pip install pandas numpy torch matplotlib scikit-learn time pyro reformer-pytorch pytorch-forecasting lightning pytorch`
3. **Informer Modeli için GitHub Üzerinden Kurulum:**

Informer modeli için, aşağıdaki adımları takip ederek GitHub üzerinden kurulumu gerçekleştirin:  
`!git clone https://github.com/zhouhaoyi/Informer2020.git`

`!cd Informer2020`

`!pip install -r requirements.txt`

`!python setup.py install`

### III. MODELLERİN ÇALIŞTIRILMASI VE AÇIKLANMASI

Bu bölümde, zaman serisi tahmini için kullanılan modellerin (Reformer, Informer, TFT, LSTNet, TST, BTSM, Vanilla) çalışma prensipleri ve uygulama adımları açıklanmaktadır. Açıklamalar, Reformer modeli üzerinden yapılmış olup, diğer modeller içinde benzer yapı ve yöntemler geçerlidir.

#### A. Zaman Serisi Modelleri

- **Reformer:** Uzun dönem bağımlılıklarını öğrenmek için Transformer tabanlı bir model. reformer\_pytorch kütüphanesi ile uygulanır.
- **Informer:** Zaman serisi tahmini için optimizasyon sağlanmış bir Transformer modelidir. Projeye özel models.model modülünden ithal edilmiştir.
- **TFT (Temporal Fusion Transformer):** pytorch-forecasting kütüphanesi ile zaman serisi tahmininde güçlü performans gösteren bir modeldir.
- **TST (Time Series Transformer):** Zaman serisi tahmini için bir başka Transformer tabanlı modeldir.
- **Vanilla Model:** Basit bir Encoder-Decoder yapısı ile zaman serisi tahmini yapan klasik bir derin öğrenme modelidir.

#### B. Veri Hazırlama

##### 1) Veri Yükleme ve İşleme

Çalışmada kullanılan hava durumu verileri CSV formatında yüklenmiştir. Veriler, pandas kütüphanesi ile işlenerek özellikler (ör. sıcaklık, nem) seçilmiştir. Zaman serisi verilerinin modellemeye uygun hale getirilmesi için StandardScaler ile ölçeklendirme işlemi gerçekleştirilmiştir.

##### 2) Zaman Serisi Sekansları Oluşturma

Veriler, belirli bir pencere boyutunda (ör. 30 günlük dilimler) sekanslara ayrılmıştır. Bu sekanslar, giriş verisi ve hedef çıktı olarak düzenlenmiştir. Eğitim ve test veri setleri %80 eğitim ve %20 test oranında ayrılmıştır.

##### 3) Veri Yükleme (DataLoader Kullanımı)

Eğitim ve test veri setleri, DataLoader kullanılarak mini-batch formatında modele uygun hale getirilmiştir. Bu süreç, eğitim sırasında hafıza verimliliğini artırmayı ve işlem hızını optimize etmeyi sağlar.

#### C. Model Yapısı

Zaman serisi tahmini için kullanılan modeller, her biri farklı mimari yaklaşımlar sunmaktadır. Reformer modeline ilişkin detaylar aşağıda verilmiştir:

- **Giriş Verisi Projeksiyonu:** Model, giriş verisini gizli katman boyutuna uygun bir biçimde yeniden projekte eder.
- **Reformer Katmanı:** Zaman serisi verilerinde uzun dönemli bağımlılıkları yakalamak için dikkat mekanizmaları ve segment tabanlı işlemler kullanılmıştır.

- **Çıkış Katmanı:** Model, son zaman adımındaki tahmini döndürmek üzere bir tam bağlantılı (fully connected) katman kullanır.

#### D. Eğitim Süreci

##### 1) Kayıp Fonksiyonu ve Optimizasyon

Modellerin eğitimi sırasında kayıp fonksiyonu olarak Ortalama Kare Hata (MSE) kullanılmıştır. Optimizasyon algoritması olarak Adam tercih edilmiştir. Ayrıca öğrenme oranını zamanla azaltmak için bir planlayıcı (StepLR) uygulanmıştır.

##### 2) Erken Durdurma Mekanizması

Eğitim sırasında doğrulama kayıpları izlenmiş ve belirli bir iyileşme gözlenmediğinde erken durdurma mekanizması devreye alınmıştır. Bu yöntem, aşırı öğrenmenin (overfitting) önlenmesine katkı sağlar.

##### 3) Eğitim ve Doğrulama Kayıpları

Her epoch sonunda eğitim ve doğrulama kayıpları kaydedilmiştir. Bu değerler grafiksel olarak sunularak modellerin öğrenme performansları değerlendirilmiştir.

#### E. Tahmin ve Performans Değerlendirmesi

##### 1) Test Performansı

Modellerin test seti üzerindeki performansı, aşağıdaki metriklerle değerlendirilmiştir:

- Ortalama Kare Hata (MSE)
- Ortalama Mutlak Hata (MAE)
- Kök Ortalama Kare Hata (RMSE)
- Ortalama Mutlak Yüzde Hatası (MAPE)
- $R^2$  Skoru

##### 2) Gelecek Tahminleri

Eğitilen modeller, mevcut verilerden yola çıkarak geleceğe yönelik 7 günlük tahminler yapmıştır. Bu tahminler, veri ölçeklendirme ters çevrilerek orijinal birimlere dönüştürülmüştür.

##### 3) Görselleştirme

Orijinal veri ve tahmin sonuçları karşılaştırmalı olarak grafiklerde sunulmuştur. Özellikle son 10 günlük veri ile tahminlerin doğruluğu analiz edilmiştir.

#### F. Performans ve Çıkarım Süreleri

Her model için eğitim ve çıkarım süreleri ölçülmüştür. Bu süreler, modellerin verimlilik ve gerçek zamanlı uygulamalara uygunluğunu değerlendirmede kullanılmıştır.

#### G. Kod ve Model Genel Çerçevesi

Bu süreç, tüm modeller için benzer bir yapıda uygulanmıştır. Her bir model, veri hazırlama, eğitim, değerlendirme ve tahmin adımları açısından uyarlanabilir bir çerçeveye sahiptir. Özellikle Informer modeli gibi harici kütüphanelerle gelen mimarilerde, kod yollarının doğru şekilde tanımlanması ve uygun kurulumların yapılması gerekmektedir.

#### IV. ÇIKTI SONUÇLARI VE GENEL DEĞERLENDİRME

##### A. Model Analizleri

Model	MSE	MAE	RMSE	R <sup>2</sup>	MAPE	Training Time (s)	Inference Time (s)	Overall Score
LSTNet	0,3283	0,4007	0,573	0,6745	2,0265	120,14	0,02	-0,9946
Bayesian Reformer	0,3368	0,4032	0,5803	0,6657	2,1397	1206,08	0,13	-0,8237
Autoformer	0,3404	0,4098	0,5834	0,6617	2,0048	487,6	1,21	-0,9177
Reformer	0,334	0,4007	0,5779	0,6683	2,091	834,09	0,12	-0,8817
TFT	59,747	5,1653	6,5174	-0,3063	1,0293	1036,72	21,17	0,8081
Vanilla	0,333	0,4025	0,577	0,6694	2,0644	83,81	0,02	-0,9949
TST	0,3531	0,415	0,5943	0,6502	2,0166	285,38	0,05	-0,9453
Informer	0,4121	0,4569	0,642	0,5869	2,2587	212,57	0,04	-0,8924

Fig. 1. Eğitim verileri tablosu

##### 1) Model Performans Metriği Analizi

Metrikler üzerinden modellerin performansı şu şekilde sıralanabilir:

##### a) En Düşük Hata Değerleri (MSE, MAE, RMSE):

- Vanilla, LSTNet ve Reformer modelleri en düşük hata değerlerini sunuyor:
- MSE: LSTNet (0.3283), Vanilla (0.333)
- MAE: LSTNet ve Vanilla (0.4007)
- RMSE: LSTNet (0.573), Vanilla (0.577)

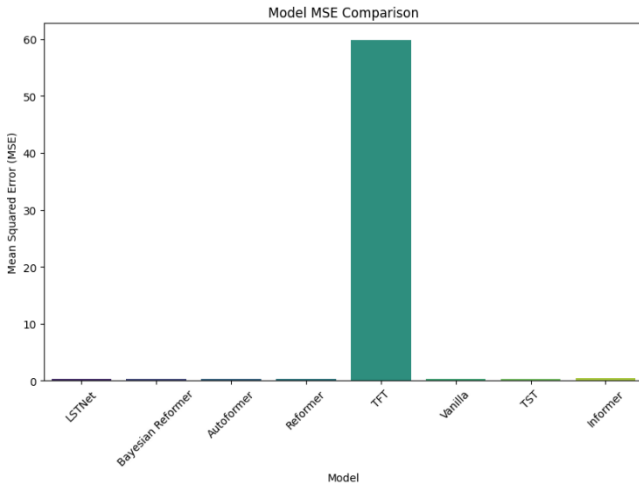


Fig. 2. Modellerin MSE grafiği

##### b) En Yüksek R<sup>2</sup> (Determinasyon Katsayısı) Değerleri:

- LSTNet (0.6745), Vanilla (0.6694), Reformer (0.6683)

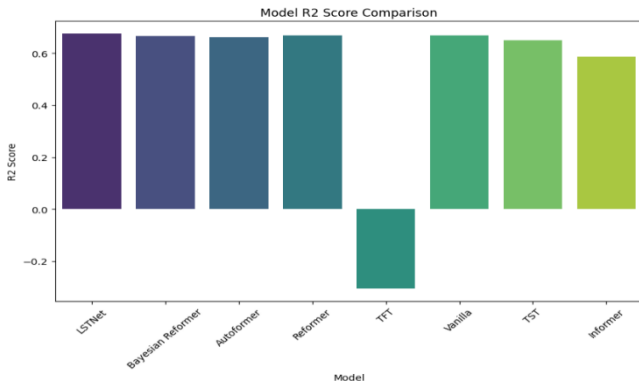


Fig. 3. Modellerin R<sup>2</sup> (Determinasyon Katsayısı) Değerleri

##### c) En Düşük MAPE (Oransal Hata) Değerleri:

- TFT (1.0293) ve LSTNet (2.0265) değerleri ile en düşük hatalara sahip.

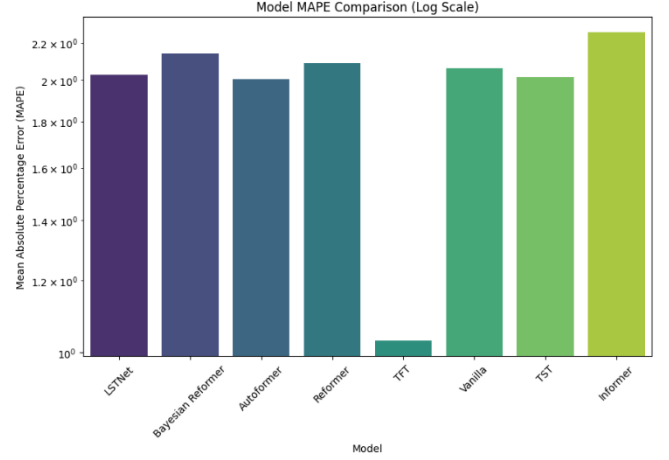


Fig. 4. Modellerin MAPE (Oransal Hata) Değerleri

##### 2) Zaman Metrikleri (Eğitim ve Çıkarım Süresi)

##### a) En Hızlı Eğitim Süresi:

- Vanilla modeli (83.81 saniye), LSTNet (120.14 saniye).

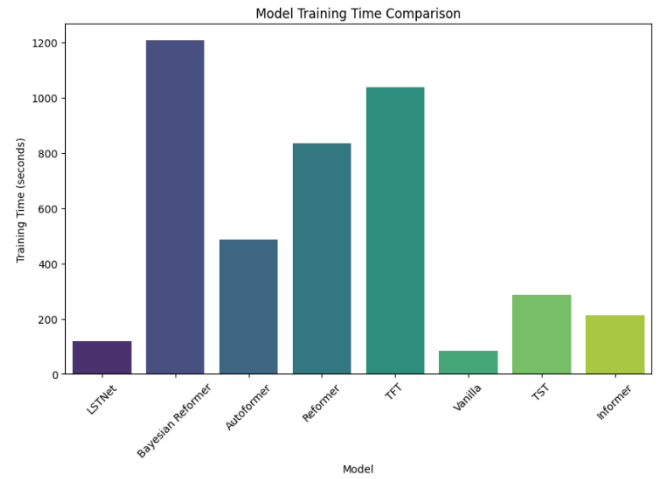


Fig. 5. Modellerin eğitim süreleri

##### b) En Hızlı Çıkarım Süresi:

- LSTNet ve Vanilla modelleri (0.02 saniye) öne çıkıyor.

##### 3) Overall Score (Genel Skor)

Genel skor, modellerin performans, eğitim ve çıkarım sürelerini dikkate alarak bir değerlendirme sunuyor. Bu bağlamda en düşük skor Vanilla (-0.9949) ve LSTNet (-0.9946) modellerinde görülmektedir. Düşük skor daha iyi sonuçlar veren modelleri göstermektedir.

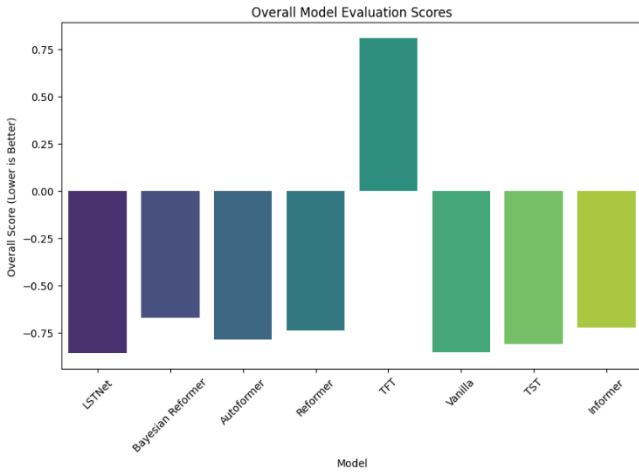


Fig. 6. Modellerin genel öğrenme skoru

### B. Model Performans Karşılaştırmaları

#### 1) TFT Modeli Performansı

**TFT modeli**, diğer modellere kıyasla farklı bir performans sergiliyor:

- **MSE (59.7474)** ve **RMSE (6.5174)** oldukça yüksek.
- **R<sup>2</sup> (-0.3063)** negatif bir değer olarak modelin veriyi genelleştiremediğini gösteriyor.
- **MAPE (1.0293)** düşük gibi görünse de diğer hata metrikleri bu modelin performans açısından yetersiz olduğunu ortaya koyuyor.
- **Eğitim Süresi:** 1036.72 saniye (en uzun sürelerden biri).
- **Çıkarım Süresi:** 21.17 saniye (en uzun süre).

#### 2) Bayesian Reformer ve Reformer Karşılaştırması

Bayesian Reformer daha düşük bir MSE (0.3368) ve MAE (0.4032) sunmasına rağmen, eğitim süresi (1206.08 saniye) ve çıkarım süresi (0.13 saniye) standart Reformer modeline kıyasla daha maliyetli. Standart Reformer daha dengeli bir performans sergiliyor.

### C. Genel Değerlendirme

#### 1) Performans Öncelikli Senaryolar:

Vanilla ve LSTNet modelleri, hızlı eğitim/çıkartım süreleri ve düşük hata metrikleriyle öne çıkıyor.

#### 2) Dengeli Performans:

Reformer, hata metriklerinde düşük ve zaman açısından dengeli sonuçlar sunuyor.

#### 3) TFT Modeli Kullanımı:

TFT modeli, bu veri seti ve problem için uygun görünmemektedir. Eğitim ve çıkarım süreleri uzun olup, hata metrikleri beklentinin çok altındadır.

## V. ÖRNEK MODEL TAHMİN ÇIKTISI

Bu bölümde, modelin tahmin performansını analiz etmek ve yorumlamak amacıyla 7 günlük tahmin sonuçları, eğitim/doğrulama kayıpları ve tahmin edilen değişkenlerin trend grafikleri değerlendirilmiştir.

### A. Eğitim ve Doğrulama Kayıp Grafiği

Eğitim ve doğrulama kayıplarını gösteren ilk grafik, modelin eğitim sürecindeki performansını görselleştirmektedir. Grafik incelendiğinde:

#### 1) Eğitim kaybı (Train Loss):

İlk epoch'ta oldukça yüksek bir değerle başlamakta (~0.41) ancak daha sonra hızlı bir şekilde azalarak stabil hale gelmektedir (~0.33). Bu durum, modelin eğitim verilerine başarıyla uyum sağladığını göstermektedir.

#### 2) Doğrulama kaybı (Validation Loss):

Eğitim kaybıyla benzer bir azalma trendi göstermektedir. Ancak doğrulama kaybı, 8. epoch'tan sonra hafif bir artış göstermiştir. Bu durum, modelin daha fazla eğitildiğinde doğrulama verilerinde aşırı öğrenmeye (overfitting) yatkın olabileceğini işaret edebilir.

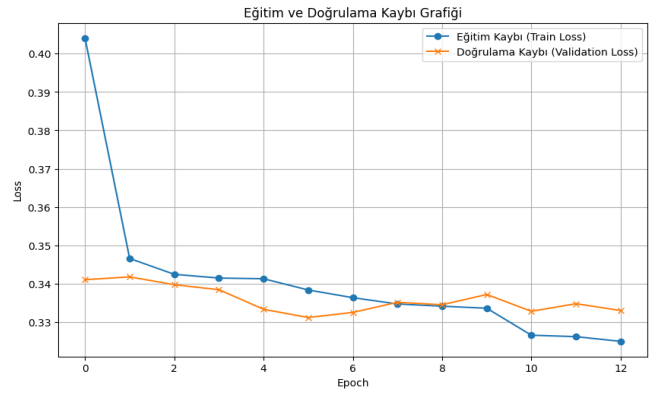


Fig. 7. Reformer modelinin eğitim ve doğrulama kaybı grafiği

### B. 7 Günlük Tahminler

Modelin ürettiği tahmin sonuçları, farklı meteorolojik değişkenler için analiz edilmiştir. Tahmin edilen her bir değişkenin 7 günlük trend grafiği ikinci görselde yer almaktadır:

#### 1) Temperature (Sıcaklık):

Sıcaklık tahminleri hafif bir artış göstermekte ve 15.70 ile 16.32 arasında değişmektedir. Tahminlerin yumuşak bir eğilim göstermesi, modelin stabil ve tutarlı tahminler ürettiğini göstermektedir.

#### 2) Dew Point (Çiy Noktası):

Çiy noktası tahminleri, 11.54 ile 13.03 arasında artış göstermiştir. Bu durum, havadaki nemin artabileceğine işaret etmektedir.

#### 3) Humidity (Nem):

Nem tahminleri sabit bir artış trendine sahiptir ve 76.71'den 83.28'e yükselmiştir. Bu, ilerleyen günlerde nem oranının artış göstereceğini öngörmektedir.

#### 4) Wind Speed (Rüzgar Hızı):

Rüzgar hızı tahminleri oldukça sabit bir seyir izlemekte ve 9.00 ile 9.65 arasında değişmektedir. Rüzgar hızında büyük dalgalanmaların olmaması, modelin bu değişkene yönelik başarılı bir tahmin yaptığını göstermektedir.

#### 5) Pressure (Basınç):

Basınç tahminleri, hafif bir düşüş eğilimindedir (29.81'den 29.71'e). Bu, meteorolojik koşullardaki olası bir değişime işaret edebilir.

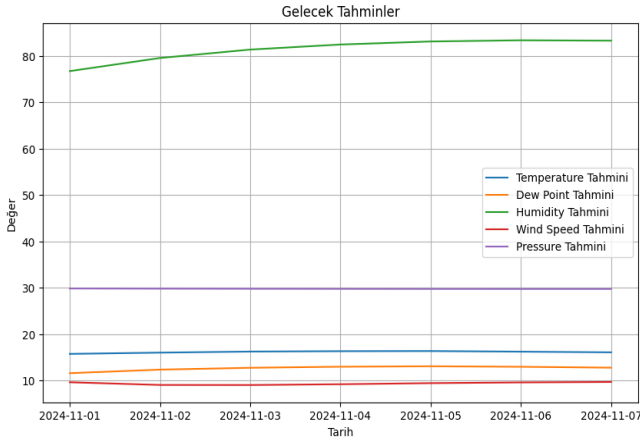


Fig. 8. Gelecek 7 günün tahmini değerlerinin grafiği

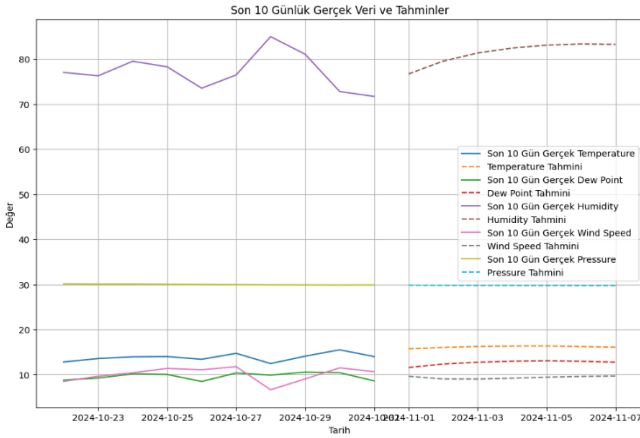


Fig. 9. Son 10 gün ile gelecek 7 günün tahmini değerlerinin grafiği

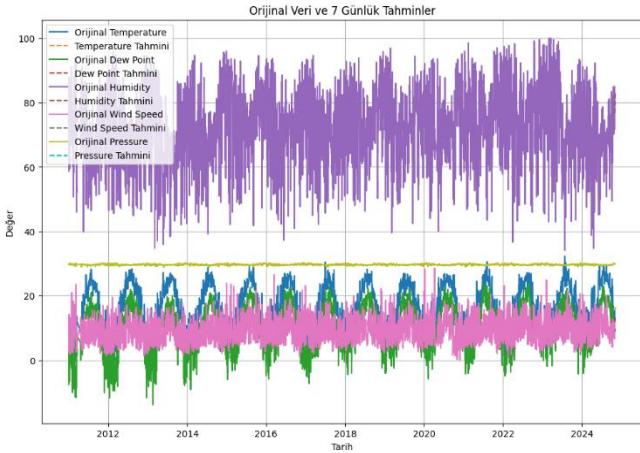


Fig. 10. Tüm veri ve gelecek 7 günün tahmini değerlerinin grafiği

## VI. SONUÇ VE DEĞERLENDİRME

Modellerin ürettiği tahminler genel olarak tutarlı ve gerçekçidir. Eğitim ve doğrulama kayıplarındaki düşüş, modelin eğitimi sırasında veriyi iyi bir şekilde öğrendiğini göstermektedir. Ancak öğrenme kaybında veya doğrulama kaybında gözlemlenen ufak artışlar modelin aşırı öğrenme yapabileceğine işaret eder. Bu durum kodumuzda kontrol edildiği için normalde 35 döngülük eğitimimiz 13. döngüde öğrenme kaybı bizim belirlediğimiz orandan daha az olduğu için durduruluyor.

Ayrıca, tahminler doğrusal bir şekilde ilerlediğinden modellerin kısa vadeli tahminler için uygundur. Uzun vadeli tahminler için ise daha kompleks modeller kullanılması daha mantıklı olacaktır.

Sonuç olarak, bu çalışmada elde edilen bulgular, kısa vadeli hava durumu tahminleri için kullanılan modellerin yeterliliğini ve başarı düzeyini ortaya koymuştur. Gelecekte, daha kapsamlı veri setleri ve ileri düzey modellerle yapılacak çalışmalar, uzun vadeli tahminlerde daha güçlü performans elde edilmesini sağlayabilir.

## VII. KAYNAKÇA

- [1] [Pandas documentation — pandas 2.2.3 documentation](#) [Erişim tarihi: 2 Ocak 2025].
- [2] [NumPy documentation — NumPy v2.2 Manual](#) [Erişim tarihi: 2 Ocak 2025].
- [3] [PyTorch documentation — PyTorch 2.5 documentation](#) [Erişim tarihi: 2 Ocak 2025].
- [4] [Using Matplotlib — Matplotlib 3.10.0 documentation](#) [Erişim tarihi: 2 Ocak 2025].
- [5] [scikit-learn: machine learning in Python — scikit-learn 1.6.0 documentation](#) [Erişim tarihi: 2 Ocak 2025].
- [6] [time — Time access and conversions — Python 3.10.16 documentation](#) [Erişim tarihi: 2 Ocak 2025].
- [7] [Getting Started With Pyro: Tutorials, How-to Guides and Examples — Pyro Tutorials 1.9.1 documentation](#) [Erişim tarihi: 2 Ocak 2025].
- [8] [lucidrains/reformer-pytorch: Reformer, the efficient Transformer, in Pytorch](#) [Erişim tarihi: 2 Ocak 2025].
- [9] [PyTorch Forecasting Documentation — pytorch-forecasting documentation](#) [Erişim tarihi: 2 Ocak 2025].
- [10] [zhouhaoyi/Informer2020: The GitHub repository for the paper "Informer" accepted by AAAI 2021.](#) [Erişim tarihi: 2 Ocak 2025].
- [11] [Getting started ⚡ Lightning AI - Docs](#) [Erişim tarihi: 2 Ocak 2025].
- [12] [The Complete Guide to Time Series Models](#) [Erişim tarihi: 2 Ocak 2025].
- [13] [Time-series modeling: DataRobot docs](#) [Erişim tarihi: 2 Ocak 2025].
- [14] [Reformers and Performers: A comprehensive summary of space and time optimizations on Transformers \(Part — 2\) | by Priya Shree | Walmart Global Tech Blog | Medium](#) [Erişim tarihi: 2 Ocak 2025].

## VIII. EKLER

Proje Kaynak Kodları : [Proje Github Linki](#)

Colab Bağlantıları: [GDrive](#)

Veri Seti : [GDrive](#)