

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ELEKTRİKLİ ARAÇ ŞARJ İSTASYONLARINDA KULLANIM
EĞİLİMLERİNİN MAKİNE ÖĞRENİMİNE DAYALI MODELLENMESİ VE
TALEP TAHMİNİ**

YÜKSEK LİSANS TEZİ

Hatice Bilge ALGIN

Elektrik Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Murat KÖSEOĞLU

OCAK 2026

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRİKLİ ARAÇ ŞARJ İSTASYONLARINDA KULLANIM
EĞİLİMLERİNİN MAKİNE ÖĞRENİMİNE DAYALI MODELLENMESİ VE
TALEP TAHMİNİ

YÜKSEK LİSANS TEZİ

Hatice Bilge ALGIN
(36223615906)

Elektrik Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Murat KÖSEOĞLU

OCAK 2026

TEŞEKKÜR VE ÖNSÖZ

Bu tez çalışmasının planlanması, yürütülmesi ve sonuçlandırılması aşamalarında bilimsel rehberliği, bilgi birikimi ve yapıcı değerlendirmeleriyle çalışmama yön veren; her aşamada destek ve katkılarını esirgemeyen danışmanım Sayın Doç. Dr. Murat KÖSEOĞLU'na en içten teşekkürlerimi sunarım.

Çalışmalarım süresince teknik destek ve yol göstericiliği ile sürece değerli katkılar sağlayan Öğr. Gör. Buğra DAĞCI'ya teşekkür ederim.

Hayatım boyunca olduğu gibi bu tez çalışması sürecinde de destekleriyle yanımda olan, anlayış ve sabırlarıyla çalışmalarımı sürdürmemde önemli rol üstlenen aileme şükranlarımı sunarım.

Bu süreç boyunca motivasyonumu ve çalışma disiplinimi destekleyen, varlığıyla ilham ve güç kaynağım olan sevgili oğlum Efe AKSOĞAN'a teşekkür ederim



ONUR SÖZÜ

Yüksek lisans tezi olarak sunduğum “Elektrikli Araç Şarj İstasyonlarında Kullanım Eğilimlerinin Makine Öğrenimine Dayalı Modellenmesi ve Talep Tahmini” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Hatice Bilge ALGIN



İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	v
ŞEKİLLER DİZİNİ.....	vi
SEMBOLLER VE KISALTMALAR	x
ÖZET	xii
ABSTRACT	xiv
1. GİRİŞ.....	1
1.1 Genel Bilgiler.....	1
1.2 Literatür Taraması	2
1.2.1 Kullanıcı Davranış Analizleri.....	3
1.2.2 Ampirik Veri Analizi.....	4
1.2.3 Talep Tahmin Modelleri.....	5
1.2.4 Simülasyon ve Analitik Yaklaşımlar.....	6
1.2.5 Alternatif Yaklaşımlar: Batarya Takas, Hibrit Sistemler ve Yenilenebilir Enerji Entegrasyonu	7
1.3 Tezin Kapsamı	8
2. TAHMİN YÖNTEMLERİNE GENEL BAKIŞ	10
2.1 Zaman Serisi Modelleme Temelleri	10
2.2 Regresyon Tabanlı Yöntemler	12
2.2.1 ARIMA.....	12
2.2.2 SARIMA.....	14
2.2.3 Holt-Winters	16
2.3 Makine Öğrenmesi Yöntemleri	17
2.3.1 XGBoost.....	18
2.3.2 Random Forest.....	20
2.3.3 SVR	22
2.3.4 GBR.....	24
2.3.5 LGBM.....	26
2.4 Derin Öğrenme Yöntemleri	27
2.4.1 LSTM	29
2.5 Diğer Yaklaşımlar.....	31
2.5.1 Prophet.....	32
2.6 Yöntemlerin Karşılaştırmalı Değerlendirilmesi	34
3. YAZILIM ALTYAPISI VE MODELLEME SİSTEMİ	36
3.1 Uygulama Akışı ve Genel Yapı	36
3.2 Python Tabanlı Uygulama Altyapısı	38
3.2.1 Veri Hazırlama ve Model Girdileri	38
3.2.1.1 V1 – Ham saatlik veriler.....	38
3.2.1.2 V2 – Günlük toplam veriler.....	40
3.2.1.3 V3 – Haftalık toplam veriler.....	40
3.2.1.4 V4 – Saatlik ayrıştırılmış veriler	41
3.2.2 Tahmin Modülleri ve Model Eğitimi	42
3.2.2.1 LightGBM Eğitimi	43
3.2.2.2 XGBoost Eğitimi	43
3.2.2.3 Prophet Modeli ile Zaman Serisi Eğitimi	44
3.2.2.4 Gradient Boosting Regressor (GBR) Eğitimi.....	45
3.2.3 Sonuçların Kaydedilmesi ve Performans Kayıtları	46

3.2.3.1 Hata Metriklerinin Hesaplanması	46
3.2.3.2 Sonuçların Excel Dosyasına Kaydedilmesi	47
3.2.3.3 Tahmin Sonuçlarının Görselleştirilmesi	47
3.3 MATLAB Tabanlı Uygulama Altyapısı	48
3.3.1 V1: Ham Saatlik Veriler ile MATLAB Tabanlı Modelleme Süreci	49
3.3.2 V2: Günlük Toplam Veriler ile MATLAB Tabanlı Modelleme Süreci	52
3.3.3 V3: Haftalık Toplam Veriler ile MATLAB Tabanlı Modelleme Süreci	56
3.3.4 V4: Saatlik Ayrıştırılmış Veriler ile MATLAB Tabanlı Modelleme Süreci	58
3.4 Hiperparametre Optimizasyonu	63
3.5 Sonuçların Görselleştirilmesi ve Raporlanması	66
4. VERİ SETİ KURGUSU VE UYGULAMA SONUÇLARI	68
4.1 Veri Seti Versiyonlarının Tanıtımı	68
4.1.1 V1: Ham Saatlik Veriler	69
4.1.2 V2: Günlük Toplamlar	70
4.1.3 V3: Haftalık Toplamlar	71
4.1.4 V4: Saatlik Ayrıştırılmış Veriler	72
4.2 Uygulama Sonuçlarının Değerlendirilmesi	74
4.2.1 Tahmin Sonuçlarına Ait Grafikler	74
4.2.1.1 V1 veri seti için MATLAB tabanlı tahmin sonuçları	74
4.2.1.2 V1 veri seti için Python tabanlı tahmin sonuçları	78
4.2.1.3 V2 veri seti için MATLAB tabanlı tahmin sonuçları	81
4.2.1.4 V2 veri seti için Python tabanlı tahmin sonuçları	84
4.2.1.5 V3 veri seti için MATLAB tabanlı tahmin sonuçları	87
4.2.1.6 V3 veri seti için Python tabanlı tahmin sonuçları	88
4.2.1.7 V4 veri seti için MATLAB tabanlı tahmin sonuçları	89
4.2.1.8 V4 veri seti için Python tabanlı tahmin sonuçları	103
4.2.2 Model Performanslarının Karşılaştırılması	117
4.2.2.1 V1 Veri Seti Performans Karşılaştırması	117
4.2.2.2 V2 Veri Seti Performans Karşılaştırması	121
4.2.2.3 V3 Veri Seti Performans Karşılaştırması	124
4.2.2.4 V4 Veri Seti Performans Karşılaştırması	127
4.2.2.5 Genel Değerlendirme ve Yorumlar	132
5. SONUÇ VE ÖNERİLER	136
KAYNAKLAR	138
EKLER	151
ÖZGEÇMİŞ	185

ÇİZELGELER DİZİNİ

Çizelge 2.1 : Tahmin modellerinin karşılaştırmalı özellik analizi.	35
Çizelge 4.1 : V1 veri setine dair örnek 10 satırlık veri.	69
Çizelge 4.2 : V2 veri setine dair örnek 10 satırlık veri.	70
Çizelge 4.3 : V3 veri setine dair örnek 10 satırlık veri.	71
Çizelge 4.4 : V3 veri setine dair örnek 10 satırlık veri.	73
Çizelge A.1 : V1 veri seti ARIMA model performansları.	151
Çizelge A.2 : V1 veri seti SARIMA model performansları.	152
Çizelge A.3 : V1 veri seti Holt-Winters model performansları.	152
Çizelge A.4 : V1 veri seti LSTM model performansları.	152
Çizelge A.5 : V1 veri seti Random Forest model performansları.	154
Çizelge A.6 : V1 veri seti SVR model performansları.	154
Çizelge A.7 : V1 veri seti LightGBM model performansları.	155
Çizelge A.8 : V1 veri seti XGBoost model performansları.	156
Çizelge A.9 : V1 veri seti Prophet model performansları.	158
Çizelge A.10 : V1 veri seti GBR model performansları.	158
Çizelge B.1 : V2 veri seti ARIMA model performansları.	159
Çizelge B.2 : V2 veri seti SARIMA model performansları.	159
Çizelge B.3 : V2 veri seti Holt-Winters model performansları.	160
Çizelge B.4 : V2 veri seti LSTM model performansları.	160
Çizelge B.5 : V2 veri seti Random Forest model performansları.	161
Çizelge B.6 : V2 veri seti SVR model performansları.	162
Çizelge B.7 : V2 veri seti LightGBM model performansları.	162
Çizelge B.8 : V2 veri seti XGBoost model performansları.	163
Çizelge B.9 : V2 veri seti Prophet model performansları.	165
Çizelge B.10 : V2 veri seti GBR model performansları.	166
Çizelge C.1 : V3 veri seti ARIMA model performansları.	167
Çizelge C.2 : V3 veri seti SARIMA model performansları.	167
Çizelge C.3 : V3 veri seti Holt-Winters model performansları.	168
Çizelge C.4 : V3 veri seti LSTM model performansları.	168
Çizelge C.5 : V3 veri seti Random Forest model performansları.	169
Çizelge C.6 : V3 veri seti SVR model performansları.	170
Çizelge C.7 : V3 veri seti LightGBM model performansları.	170
Çizelge C.8 : V3 veri seti XGBoost model performansları.	171
Çizelge C.9 : V3 veri seti Prophet model performansları.	173
Çizelge C.10 : V3 veri seti GBR model performansları.	174
Çizelge D.1 : V4 veri seti ile 24 saatlik en iyi ARIMA model performansları.	175
Çizelge D.2 : V4 veri seti ile 24 saatlik en iyi SARIMA model performansları.	176
Çizelge D.3 : V4 veri seti ile 24 saatlik en iyi Holt-Winters model performansları.	177
Çizelge D.4 : V4 veri seti ile 24 saatlik en iyi LSTM model performansları.	177
Çizelge D.5 : V4 veri seti ile 24 saatlik en iyi Random Forest model performansları.	178
Çizelge D.6 : V4 veri seti ile 24 saatlik en iyi SVR model performansları.	179
Çizelge D.7 : V4 veri seti ile 24 saatlik en iyi LightGBM model performansları.	180
Çizelge D.8 : V4 veri seti ile 24 saatlik en iyi XGBoost model performansları.	181
Çizelge D.9 : V4 veri seti ile 24 saatlik en iyi Prophet model performansları.	182
Çizelge D.10 : V4 veri seti ile 24 saatlik en iyi GBR model performansları.	183

ŞEKİLLER DİZİNİ

Şekil 1.1 : Özel ve ticari elektrikli araç kullanıcılarının normal (ev/şirket), kamusal normal ve kamusal hızlı şarj oturum sayılarının karşılaştırılması [45].	3
Şekil 1.2 : Hecht ve arkadaşlarının gerçekleştirmiş olduğu çalışmadaki güç derecesine göre meşgul olan şarj istasyonu konnektörlerinin haftalık düzeyde ortalamaları [49].	4
Şekil 1.3: Koohfar ve arkadaşlarının yapmış olduğu çalışmadaki Boulder bölgesindeki kamu şarj istasyonlarına ait günlük veriler kullanılarak farklı tahmin ufuklarında ($K = 7, 30, 60, 90$) Transformer, LSTM, RNN, SARIMA ve ARIMA modellerinin RMSE değerlerinin karşılaştırılması [54].	5
Şekil 1.4 : Van der Kam ve arkadaşlarının yapmış olduğu çalışmadaki %50 ve %100 yenilenebilir senaryolarında elektrik üretimi ve talep dağılımı [57].	6
Şekil 1.5 : Sandton ve Constantia EV şarj istasyonları için hibrit PV-BESS'in LCOE (R/kWh) ve fazla elektrik üretimi [65].	7
Şekil 2.1 : Amini ve arkadaşlarının yapmış olduğu çalışmadaki saatlik gerçek yük ile ARIMA, önerilen entegre ve ayrıştırılmış yük tahminlerinin karşılaştırması [102].	13
Şekil 2.2 : Louie'nin çalışmasında bulunan kullanım süresi fiyatlandırması olmaksızın 1800 adet EV şarj istasyonuna sahip bir sistemin sentetik 7 günlük zaman serisi [108].	14
Şekil 2.3 : Gerçek EV yük deseni ve 2 Mart 2015'ten 29 Mart 2015'e kadar 28 günlük bir ufuk için tahmin edilen EV yükleri [75].	15
Şekil 2.4 : COVID-19 karantinalarının tüketim miktarlarındaki Bölge 1 için değişiklikler (a) Elektrik ve (b) Doğalgaz [135].	17
Şekil 2.5 : Almaghrebi ve arkadaşlarının yapmış olduğu çalışmadaki her yöntem için (a) R^2 , (b) MAE ve (c) RMSE dahil olmak üzere doğruluk metrik karşılaştırmaları [138].	19
Şekil 2.6 : Orman yapısı ile şarj istasyonu grubu tahminine göre hata arasındaki ilişki (a) Mape, (b) RMSE [144].	21
Şekil 2.7 : Saatlik bazda SVR ve Monte Carlo yöntemlerinin göreceli hata karşılaştırması [146].	22
Şekil 2.8 : Geleneksel SVR'nin sonucu [147].	23
Şekil 2.9 : Üç farklı gradient boosting algoritmasının performansı [141].	24
Şekil 2.10 : (a) Rastgele orman (b) CatBoost (c) XGBoost (d) LightGBM regresyon modelleri kullanılarak Gerçek-Tahmin Edilen talep eğrisi [103].	27
Şekil 2.11 : LSTM sonuçlarının, optimize edilmiş sonuçların ve gerçek enerji talebinin karşılaştırılması [27].	30
Şekil 2.12 : %30 penetrasyon ile akıllı PHEV zamana bağlı şarj tahmin sonuçlarının ANN, RNN ve Q-öğrenme algoritmalarıyla karşılaştırılması (kW) [155].	32
Şekil 2.13 : BiLSTM varyantlarının 07–21 Temmuz 2019 dönemindeki günlük MAE ve RMSE karşılaştırması [119].	33
Şekil 3.1 : Veri seti versiyonları ve MATLAB–Python uygulama altyapısındaki yöntemler.	36
Şekil 3.2 : “TARİH” sütununu datetime formatına çeviren ve veriyi zaman eksenine göre sıralayıp indeksini sıfırlayan Python kod bloğu.	39
Şekil 3.3 : Önceki yedi saatlik gecikmeli özellikleri (Lag_1–Lag_7) üreten Python fonksiyonu.	39

Şekil 3.4 : Eğitim–test ayrımı (%80/%20) ile özellik, hedef ve tarih sütunlarının ayrımını gösteren Python kod bloğu.	39
Şekil 3.5 : Prophet modeline uygun olacak şekilde “TARİH” sütununun ds, “BASARILI” sütununun y olarak yeniden adlandırılması ve veri kümesinin eğitim–test setlerine bölünmesini gösteren Python kod bloğu.	40
Şekil 3.6 : Prophet kütüphanesinde “changepoint_prior_scale”, “seasonality_mode” ve “holidays_prior_scale” için oluşturulan parametre ızgarası.	41
Şekil 3.7 : Tarihsel özniteliklerle (YIL, AY, HAFTANIN_GUNU) saatlik hedef değişkenlerin ayrımını gerçekleştiren Python kod bloğu.	41
Şekil 3.8 : Prophet Modeli İçin Saatlik Zaman Simülasyonu ve ds Dönüşümü.	42
Şekil 3.9 : LightGBM hiperparametre ızgarası ile döngüsel model eğitimi.	43
Şekil 3.10 : Xgboost hiperparametre ızgarası ile döngüsel model eğitimi.	44
Şekil 3.11 : Prophet modelinin hiperparametre tarama süreci.	45
Şekil 3.12 : GBR modelinin hiperparametre arama ve eğitim süreci.	45
Şekil 3.13 : evaluate_performance fonksiyon kod bloğu.	46
Şekil 3.14 : Modellerin hiper-parametre kombinasyonları ve performans metriklerinin Excel dosyasına kaydedilmesi.	47
Şekil 3.15 : Eğitim ve test dönemi gerçek değerleri ile model tahminlerinin zaman ekseninde gösterimi.	48
Şekil 3.16 : V1 veri setinin okunması ve toplamaların hesaplanması.	49
Şekil 3.17 : Eğitim ve test setlerinin oluşturulması.	50
Şekil 3.18 : ARIMA modelinin tahmin üretmesi.	50
Şekil 3.19 : SARIMA modeliyle sezonluk tahmin üretimi.	51
Şekil 3.20 : Holt-Winters tahmin hesaplama örneği.	51
Şekil 3.21 : LSTM modelinden tahmin değerlerinin geri dönüştürülmesi.	51
Şekil 3.22 : Gecikmeli değişkenlerin oluşturulması.	52
Şekil 3.23 : Sonuçların excel dosyasına kaydedilmesi.	52
Şekil 3.24 : Günlük verilerin okunması ve gruplanması.	53
Şekil 3.25 : Eğitim ve test setlerinin oluşturulması.	53
Şekil 3.26 : ARIMA tahmini ve geri dönüşüm	54
Şekil 3.27 : SARIMA tahminleme yapısı.	54
Şekil 3.28 : Holt-Winters formülü ile tahmin.	54
Şekil 3.29 : LSTM modelinden tahmin değerlerinin geri dönüştürülmesi.	55
Şekil 3.30 : Gecikmeli değişkenlerin oluşturulması.	55
Şekil 3.31 : Excel çıktısı kaydetme yapısı.	55
Şekil 3.32 : Günlük verilerin okunması ve gruplanması.	56
Şekil 3.33 : ARIMA tahmini ve geri dönüşüm.	57
Şekil 3.34 : SARIMA tahminleme yapısı.	57
Şekil 3.35 : Holt-Winters formülü ile tahmin.	57
Şekil 3.36 : LSTM modelinden tahmin değerlerinin geri dönüştürülmesi.	57
Şekil 3.37 : TreeBagger ile Random Forest modelinin eğitim ve tahmin komutları.	58
Şekil 3.38 : fitrsvm fonksiyonu ile SVR modelinin gecikmeli özellikler kullanılarak eğitim ve tahmin komutları.	58
Şekil 3.39 : V4 veri setinin MATLAB ortamına okunması ve sıralanması.	59
Şekil 3.40 : Her saat için ayrı veri ayrımı döngüsü.	60
Şekil 3.41 : Saatlik ARIMA model tahmin süreci.	61
Şekil 3.42 : LSTM modeli için normalizasyon ve tahmin işlemleri.	61
Şekil 3.43 : Lag değişkenlerinin üretimi.	62
Şekil 3.44 : Saatlik tahminlerin grafik olarak kaydedilmesi.	62
Şekil 3.45 : En iyi parametrelerin Best_Parameters sayfasına yazılması.	62

Şekil 3.46 : GBR için parametrik döngüsel arama kodu.....	63
Şekil 3.47 : Prophet hiperparametre arama kodu.	64
Şekil 3.48 : Random forest hiperparametre optimizasyon kodu.	65
Şekil 3.49 : MATLAB ile LSTM hiperparametre taraması.	65
Şekil 3.50 : Python tabanlı gerçek ve. tahmin grafiği oluşturma kod bloğu.....	66
Şekil 3.51 : Matlab tabanlı birleşik grafik oluşturma kod bloğu.....	67
Şekil 4.1 : ARIMA modeli ile V1 veri seti için tahmin grafiği.	75
Şekil 4.2 : LSTM modeli ile V1 veri seti için tahmin grafiği.	75
Şekil 4.3 : Random Forest modeli ile V1 veri seti için tahmin grafiği.	76
Şekil 4.4 : SVR modeli ile V1 veri seti için tahmin grafiği.	77
Şekil 4.5 : Holt-Winters modeli ile V1 veri seti için tahmin grafiği.....	77
Şekil 4.6 : SARIMA modeli ile V1 veri seti için tahmin grafiği.	78
Şekil 4.7 : V1 veri seti için XGBoost modeli tahmin sonuçları.....	79
Şekil 4.8 : V1 veri seti için GBR Tahmin Sonuçları.....	79
Şekil 4.9 : V1 veri seti için LightGBM tahmin sonuçları	80
Şekil 4.10 : V1 veri seti için Prophet modeli tahmin sonuçları	80
Şekil 4.11 : V2 veri seti üzerinde ARIMA modeli tahmin grafiği.....	81
Şekil 4.12 : V2 veri seti üzerinde SARIMA modeli tahmin grafiği.....	82
Şekil 4.13 : V2 veri seti üzerinde LSTM modeli tahmin grafiği.	82
Şekil 4.14 : V2 veri seti üzerinde Random Forest modeli tahmin grafiği.....	83
Şekil 4.15 : V2 veri seti üzerinde SVR modeli tahmin grafiği.	84
Şekil 4.16 : V2 veri seti için XGBoost modeline ait tahmin grafiği.	85
Şekil 4.17 : V2 veri seti için Gradient Boosting Regressor modeline ait tahmin grafiği....	85
Şekil 4.18 : V2 veri seti için LightGBM modeline ait tahmin grafiği.	86
Şekil 4.19 : V2 veri seti için Prophet modeline ait tahmin grafiği.....	86
Şekil 4.20 : V3 veri seti için MATLAB ortamında eğitilen modellerin ortak tahmin grafiği	87
Şekil 4.21 : V3 veri seti için Python ortamında eğitilen modellerin ortak tahmin grafiği..	88
Şekil 4.22 : V4 veri seti için MATLAB tabanlı 0. saat tahmin sonuçları.	90
Şekil 4.23 : V4 veri seti için MATLAB tabanlı 1. saat tahmin sonuçları.	90
Şekil 4.24 : V4 veri seti için MATLAB tabanlı 2. saat tahmin sonuçları.	91
Şekil 4.25 : V4 veri seti için MATLAB tabanlı 3. saat tahmin sonuçları.	91
Şekil 4.26 : V4 veri seti için MATLAB tabanlı 4. saat tahmin sonuçları.	92
Şekil 4.27 : V4 veri seti için MATLAB tabanlı 5. saat tahmin sonuçları.	92
Şekil 4.28 : V4 veri seti için MATLAB tabanlı 6. saat tahmin sonuçları.	93
Şekil 4.29 : V4 veri seti için MATLAB tabanlı 7. saat tahmin sonuçları.	93
Şekil 4.30 : V4 veri seti için MATLAB tabanlı 8. saat tahmin sonuçları.	94
Şekil 4.31 : V4 veri seti için MATLAB tabanlı 9. saat tahmin sonuçları.	95
Şekil 4.32 : V4 veri seti için MATLAB tabanlı 10. saat tahmin sonuçları.	95
Şekil 4.33 : V4 veri seti için MATLAB tabanlı 11. saat tahmin sonuçları.....	96
Şekil 4.34 : V4 veri seti için MATLAB tabanlı 12. saat tahmin sonuçları.	96
Şekil 4.35 : V4 veri seti için MATLAB tabanlı 13. saat tahmin sonuçları.	97
Şekil 4.36 : V4 veri seti için MATLAB tabanlı 14. saat tahmin sonuçları.	97
Şekil 4.37 : V4 veri seti için MATLAB tabanlı 15. saat tahmin sonuçları.	98
Şekil 4.38 : V4 veri seti için MATLAB tabanlı 16. saat tahmin sonuçları.	98
Şekil 4.39 : V4 veri seti için MATLAB tabanlı 17. saat tahmin sonuçları.	99
Şekil 4.40 : V4 veri seti için MATLAB tabanlı 18. saat tahmin sonuçları.	100
Şekil 4.41 : V4 veri seti için MATLAB tabanlı 19. saat tahmin sonuçları.	100
Şekil 4.42 : V4 veri seti için MATLAB tabanlı 20. saat tahmin sonuçları.	101
Şekil 4.43 : V4 veri seti için MATLAB tabanlı 21. saat tahmin sonuçları.	101

Şekil 4.44 : V4 veri seti için MATLAB tabanlı 22. saat tahmin sonuçları.....	102
Şekil 4.45 : V4 veri seti için MATLAB tabanlı 23. saat tahmin sonuçları.....	102
Şekil 4.46 : V4 veri seti için Python tabanlı 0. saat tahmin sonuçları.....	104
Şekil 4.47 : V4 veri seti için Python tabanlı 1. saat tahmin sonuçları.....	104
Şekil 4.48 : V4 veri seti için Python tabanlı 2. saat tahmin sonuçları.....	105
Şekil 4.49 : V4 veri seti için Python tabanlı 3. saat tahmin sonuçları.....	105
Şekil 4.50 : V4 veri seti için Python tabanlı 4. saat tahmin sonuçları.....	106
Şekil 4.51 : V4 veri seti için Python tabanlı 5. saat tahmin sonuçları.....	106
Şekil 4.52 : V4 veri seti için Python tabanlı 6. saat tahmin sonuçları.....	107
Şekil 4.53 : V4 veri seti için Python tabanlı 7. saat tahmin sonuçları.....	108
Şekil 4.54 : V4 veri seti için Python tabanlı 8. saat tahmin sonuçları.....	109
Şekil 4.55 : V4 veri seti için Python tabanlı 9. saat tahmin sonuçları.....	109
Şekil 4.56 : V4 veri seti için Python tabanlı 10. saat tahmin sonuçları.....	110
Şekil 4.57 : V4 veri seti için Python tabanlı 11. saat tahmin sonuçları.....	110
Şekil 4.58 : V4 veri seti için Python tabanlı 12. saat tahmin sonuçları.....	111
Şekil 4.59 : V4 veri seti için Python tabanlı 13. saat tahmin sonuçları.....	111
Şekil 4.60 : V4 veri seti için Python tabanlı 14. saat tahmin sonuçları.....	112
Şekil 4.61 : V4 veri seti için Python tabanlı 15. saat tahmin sonuçları.....	113
Şekil 4.62 : V4 veri seti için Python tabanlı 16. saat tahmin sonuçları.....	113
Şekil 4.63 : V4 veri seti için Python tabanlı 17. saat tahmin sonuçları.....	114
Şekil 4.64 : V4 veri seti için Python tabanlı 18. saat tahmin sonuçları.....	114
Şekil 4.65 : V4 veri seti için Python tabanlı 19. saat tahmin sonuçları.....	115
Şekil 4.66 : V4 veri seti için Python tabanlı 20. saat tahmin sonuçları.....	115
Şekil 4.67 : V4 veri seti için Python tabanlı 21. saat tahmin sonuçları.....	116
Şekil 4.68 : V4 veri seti için Python tabanlı 22. saat tahmin sonuçları.....	117
Şekil 4.69 : V4 veri seti için Python tabanlı 23. saat tahmin sonuçları.....	117
Şekil 4.70 : Farklı Veri Setlerine Karşılık Gelen En Başarılı Modeller	135

SEMBOLLER VE KISALTMALAR

R/kWh	: Rand per kilowatt-hour (Kilovatsaat başına Rand)
R²	: Coefficient of Determination (Belirleme Katsayısı)
AC	: Alternative Current (Alternatif Akım)
AIC	: Akaike Information Criterion (Akaike Bilgi Kriteri)
ANN	: Artificial Neural Network (Yapay Sinir Ağı)
AR	: Autoregressive (Oto regresif)
ARIMA	: Autoregressive Integrated Moving Average (Oto regresif Bütünleşik Hareketli Ortalama)
BIC	: Bayesian Information Criterion (Bayesyen Bilgi Kriteri)
Bi-LSTM	: Bidirectional Long Short-Term Memory (Çift Yönlü Uzun Kısa Süreli Bellek)
CNN-LSTM	: Convolutional Neural Network – Long Short-Term Memory (Evrışimsel Sinir Ağı – Uzun Kısa Süreli Bellek)
DC	: Direct Current (Doğru Akım)
EAFO	: European Alternative Fuels Observatory (Avrupa Alternatif Yakıtlar Gözlemevi)
EV	: Electric Vehicle (Elektrikli Araç)
GBR	: Gradient Boosting Regressor (Gradyan Artırmalı Regresyon)
GBRT	: Gradient Boosted Regression Trees (Gradyan Artırılmış Regresyon Ağaçları)
IDE	: Integrated Development Environment (Entegre Geliştirme Ortamı)
IoT	: Internet of Things (Nesnelerin İnterneti)
kNN	: k-Nearest Neighbors (k-En Yakın Komşu)
LGBM	: LightGBM - Light Gradient Boosting Machine (Hafif Gradyan Artırma Makinesi)
LSTM	: Long Short-Term Memory (Uzun Kısa Süreli Bellek)
MA	: Moving Average (Hareketli Ortalama)
MAE	: Mean Absolute Error (Ortalama Mutlak Hata)
MAPE	: Mean Absolute Percentage Error (Ortalama Mutlak Yüzde Hata)
MBE	: Mean Bias Error (Ortalama Önyargı Hatası)
MSE	: Mean Squared Error (Ortalama Kare Hata)
MVDC	: Medium Voltage Direct Current (Orta Gerilim Doğru Akım)
PHEV	: Plug-in Hybrid Electric Vehicle (Fişli Hibrit Elektrikli Araç)
PV-BESS	: Photovoltaic-Battery Energy Storage System (Fotovoltaik-Batarya Enerji Depolama Sistemi)
PNG	: Portable Network Graphics (Taşınabilir Ağ Grafikleri)

RBF	: Radial Basis Function (Radyal Tabanlı Fonksiyon)
RMSE	: Root Mean Square Error (Kök Ortalama Kare Hata)
RNN	: Recurrent Neural Network (Tekrarlayan Sinir Ağı)
SARIMA	: Seasonal Autoregressive Integrated Moving Average (Mevsimsel Otoregresif Bütünleşik Hareketli Ortalama)
SARIMAX	: Seasonal Autoregressive Integrated Moving Average with eXogenous regressors (Mevsimsel Otoregresif Bütünleşik Hareketli Ortalama – Dışsal Regresörlü)
SMAPE	: Symmetric Mean Absolute Percentage Error (Simetrik Ortalama Mutlak Yüzde Hata)
SVR	: Support Vector Regression (Destek Vektör Regresyonu)
XGBoost	: Extreme Gradient Boosting (Aşırı Gradyan Artırma)



ÖZET

Yüksek Lisans Tezi

ELEKTRİKLİ ARAÇ ŞARJ İSTASYONLARINDA KULLANIM EĞİLİMLERİNİN MAKİNE ÖĞRENİMİNE DAYALI MODELLENMESİ VE TALEP TAHMİNİ

Hatice Bilge ALGIN

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik Elektronik Mühendisliği Anabilim Dalı

185+XV sayfa

2025

Danışman: Doç. Dr. Murat KÖSEOĞLU

Elektrikli araçların yaygınlaşmasıyla birlikte şarj altyapısına yönelik talep tahminleri, enerji arz güvenliği ve şebeke yönetimi açısından büyük önem kazanmıştır. Bu bağlamda farklı zaman ölçeklerinde oluşturulan dört veri seti üzerinden kapsamlı bir karşılaştırma yapılmıştır. Birinci veri seti ham saatlik ölçümleri; ikinci veri seti günlük toplamaları; üçüncü veri seti haftalık toplamaları; dördüncü veri seti ise 24 saat ileriye dönük saatlik pencere formatını içermektedir.

Her bir veri seti, klasik zaman serisi modelleri olan ARIMA (Autoregressive Integrated Moving Average, Otoregresif Bütünleşik Hareketli Ortalama) – SARIMA (Seasonal Autoregressive Integrated Moving Average, Mevsimsel Otoregresif Bütünleşik Hareketli Ortalama) ve parametrik Holt-Winters yöntemleri ile makine öğrenmesi tabanlı SVR (Support Vector Regression, Destek Vektör Regresyonu), Random Forest, XGBoost (eXtreme Gradient Boosting, Aşırı (Ekstrem) Gradyan Artırma), LightGBM (Light Gradient Boosting Machine, Hafif Gradyan Artırma Makinesi), GBR (Gradient Boosting Regressor, Gradyan Artırma Regresyonu) ve Prophet yaklaşımlarına ek olarak derin öğrenme mimarisi LSTM (Long Short-Term Memory, Uzun Kısa Süreli Bellek) ve BiLSTM (Bidirectional Long Short-Term Memory, Çift Yönlü Uzun Kısa Süreli Bellek) kullanılarak modellenmiştir. Model eğitiminde eksik değer atamaları, ölçeklendirme ve mevsimsel bileşen ayrıştırması gibi veri hazırlama adımları titizlikle yürütülmüştür; her yöntemin hiperparametre optimizasyonu deneysel gerçekleştirilmiştir.

Analiz sonuçları, kısa dönemli dalgalanmaların en iyi biçimde LSTM tabanlı derin öğrenme modelleri ve Prophet tarafından yakalandığını ortaya koymuştur. Birinci ve üçüncü veri setlerinde iki katmanlı LSTM modelleri en düşük hata metriklerine ulaşırken ikinci veri setinde Prophet'in multiplicative mevsimsellik bileşeni en yüksek tahmin doğruluğunu sağlamıştır. Dördüncü veri setinin saatlik öngörülerinde ise BiLSTM mimarisi, tek katmanlı yapı ve düşük düzenleme oranlarıyla en iyi genelleme performansını vermiştir.

Klasik ARIMA ve SARIMA modelleri bilgi kriterleri açısından dengeli sonuçlar sunsalar da hata minimizasyonu bakımından derin öğrenme ve Prophet kadar etkili olmamış; parametrik üstel düzleştirme yöntemlerinin katkısı ise sınırlı kalmıştır. Makine öğrenmesi tabanlı XGBoost ve LightGBM yöntemleri, düşük hesaplama maliyeti ve yorumlanabilirlik gerektiren uygulamalarda tercih edilebilir.

Sonu olarak elektrikli řarj istasyonlarının talep tahmininde kısa vadeli planlama ve gerek zamanlı optimizasyon gereksinimleri iin LSTM-BiLSTM mimarileri ve Prophet nerilmektedir. Gelecekte dıřsal deėiřkenlerin entegrasyonu ve farklı model yaklařımlarının birleřik kullanımı, ngr bařarısını daha da artırma potansiyeli tařımaktadır.

Anahtar Kelimeler: Elektrikli Ara řarj İstasyonları, Zaman Serisi Analizi, Makine ėrenmesi, Talep Tahmini, Hiperparametre Optimizasyonu



ABSTRACT

Master Thesis

MACHINE LEARNING-BASED MODELING OF USAGE TRENDS AND DEMAND FORECASTING IN ELECTRIC VEHICLE CHARGING STATIONS

Hatice Bilge ALGIN

Inonu University
Graduate School of Natural and Applied Sciences
Department of Electrical and Electronics Engineering

185+XV Pages

2025

Supervisor: Doç. Dr. Murat KÖSEOĞLU

With the increasing adoption of electric vehicles, demand forecasting for charging infrastructure has gained critical importance in terms of energy supply security and power grid management. In this context, a comprehensive comparative analysis was conducted using four datasets constructed at different temporal resolutions. The first dataset consists of raw hourly measurements, the second includes daily aggregated values, the third comprises weekly totals, and the fourth represents a 24-hour ahead forecasting structure based on an hourly sliding window format.

Each dataset was modeled using a wide range of forecasting approaches, including classical time series models such as ARIMA (Autoregressive Integrated Moving Average) and SARIMA (Seasonal Autoregressive Integrated Moving Average), parametric Holt–Winters exponential smoothing methods, machine learning-based algorithms such as Support Vector Regression (SVR), Random Forest, XGBoost, LightGBM, Gradient Boosting Regressor (GBR), and Prophet, as well as deep learning architectures including Long Short-Term Memory (LSTM) and Bidirectional Long Short-Term Memory (BiLSTM) networks. Data preprocessing procedures—including missing value imputation, feature scaling, and seasonal component decomposition—were carefully implemented. Additionally, hyperparameter optimization for each modeling approach was conducted experimentally to ensure fair and robust performance evaluation.

The analysis results demonstrate that short-term demand fluctuations are most effectively captured by LSTM-based deep learning models and the Prophet framework. For the first and third datasets, two-layer LSTM architectures achieved the lowest error metrics, while Prophet, employing a multiplicative seasonality structure, provided the highest forecasting accuracy for the second dataset. In the case of the fourth dataset, which focuses on hourly-ahead demand prediction, the BiLSTM architecture—with a single-layer configuration and low regularization settings—exhibited the best generalization performance.

Although classical ARIMA and SARIMA models produced balanced outcomes with respect to information criteria, they were less effective in minimizing forecasting errors compared to deep learning and Prophet-based approaches. Similarly, the contribution of parametric exponential smoothing methods remained limited. In contrast, machine learning-

based models such as XGBoost and LightGBM may be preferable in scenarios where low computational cost and interpretability are prioritized.

In conclusion, LSTM–BiLSTM architectures and Prophet are recommended for short-term planning and real-time optimization tasks in electric vehicle charging demand forecasting. Future studies may further enhance predictive performance by integrating exogenous variables and exploring hybrid modeling frameworks.

Keywords: Electric Vehicle Charging Stations, Time Series Analysis, Machine Learning, Demand Forecasting, Hyperparameter Optimization



1. GİRİŞ

1.1 Genel Bilgiler

Sanayileşmenin ve fosil yakıt temelli ulaşım teknolojilerinin gelişimiyle birlikte, 20. yüzyıl boyunca içten yanmalı motorlar karayolu ulaşımının temel bileşeni hâline gelmiştir [1, 2]. Ancak bu sürece paralel olarak, fosil yakıt tüketimi ve buna bağlı olarak karbon salınımı ciddi çevresel sorunlara neden olmuştur [3, 4]. Artan küresel ısınma, hava kirliliği, enerji güvenliği ve sürdürülebilir kalkınma gibi sorunlar, alternatif enerji kaynaklarına olan ihtiyacı daha görünür kılmıştır [5, 6]. Bu bağlamda, elektrikli araçlar (EV – Electric Vehicles), çevresel sürdürülebilirliğe katkı sağlayan, enerji verimliliği yüksek ve uzun vadede ekonomik fayda sunan bir ulaşım alternatifi olarak ön plana çıkmıştır [7].

Elektrikli araç teknolojilerinin ilk örnekleri 19. yüzyılın ortalarına kadar uzanmasına rağmen, bu araçlar, içten yanmalı motorların sağladığı enerji yoğunluğu ve depolama avantajları nedeniyle uzun yıllar sınırlı ölçekte kalmıştır [8]. Ancak 2000’li yıllardan itibaren batarya teknolojilerindeki gelişmeler, enerji maliyetlerindeki düşüşler ve çevresel regülasyonların artmasıyla birlikte, elektrikli araçların ticarileşmesi hız kazanmıştır [9, 10]. Özellikle lityum-iyon bataryaların gelişimi, enerji yoğunluğunun artırılması ve hızlı şarj altyapılarının yaygınlaşması, elektrikli araçların günlük kullanıma uygun hâle gelmesini mümkün kılmıştır [11, 12].

Elektrikli araçların yaygınlaşmasındaki bir diğer önemli unsur ise şarj altyapılarının yeterliliği ve verimliliğidir [13, 14]. Bu bağlamda, elektrikli araç şarj istasyonlarının hem sayıca yeterli olması hem de doğru konumlandırılması büyük önem arz etmektedir [15]. Bununla birlikte, yalnızca fiziksel altyapının genişletilmesi değil, mevcut altyapının etkin ve dengeli kullanılması da şarj ekosisteminin sürdürülebilirliği açısından kritik bir gerekliliktir [16, 17]. Çünkü elektrikli araç kullanıcılarının şarj alışkanlıkları, istasyonların doluluk oranları ve bölgesel kullanım yoğunlukları zaman içinde değişkenlik göstermektedir [18–21].

Bu değişkenlikler, yalnızca kullanıcı deneyimini değil, aynı zamanda enerji yönetimini ve şehir şebekesi üzerindeki yük dengesini de doğrudan etkilemektedir [22]. Örneğin, belirli saat dilimlerinde aşırı yoğunluk yaşanması, enerji şebekesi üzerinde ani yüklenmelere yol açarak sistemin dengesini bozabilmekte; düşük talep dönemlerinde ise altyapı kapasitesinin atıl kalmasına neden olabilmektedir [22]. Dolayısıyla, elektrikli araç

şarj istasyonlarının kullanım eğilimlerinin önceden tahmin edilmesi ve bu doğrultuda stratejik kararlar alınması, hem kullanıcı memnuniyetinin artırılması hem de enerji verimliliğinin sağlanması açısından hayati bir rol oynamaktadır [23].

Son yıllarda, bu tür kullanım eğilimlerinin modellenmesi ve talep tahminlerinin gerçekleştirilmesi için ileri düzey veri analitiği yöntemlerinden yararlanılmaktadır [24–27]. Özellikle zaman serisi analizleri, makine öğrenmesi ve derin öğrenme teknikleri, geçmiş kullanım verilerine dayanarak gelecek talebin öngörülmesini mümkün kılmakta ve altyapı planlamasında karar destek mekanizmalarının oluşturulmasına katkı sunmaktadır [26, 28–30]. Bu yöntemler sayesinde, hem kısa vadeli (örneğin saatlik veya günlük) hem de uzun vadeli (haftalık ya da aylık) tahminler yapılarak, şarj istasyonlarının operasyonel verimliliği artırılabilir [31–33].

Türkiye’de elektrikli araç sayısının hızla artmasıyla birlikte, bu tür analiz ve modellemelere olan ihtiyaç da giderek artmaktadır [34, 35]. Ancak yapılan literatür taramaları göstermektedir ki, ülkemizde şarj istasyonlarının kullanım alışkanlıklarına yönelik veri temelli çalışmalar sınırlı sayıdadır ve mevcut çalışmalar çoğunlukla istasyon yer seçimi ya da şebeke bağlantıları üzerine odaklanmaktadır [36, 37]. Bu bağlamda, elektrikli araç şarj istasyonlarına ilişkin kullanım eğilimlerinin detaylı biçimde analiz edilmesi ve talep tahmin modelleriyle desteklenmesi, hem akademik dünyada hem de enerji ve ulaşım politikalarının belirlenmesinde önemli bir boşluğu dolduracaktır [24, 25, 30, 38–40].

Bu tez çalışması, yukarıda belirtilen gereklilikler doğrultusunda; saatlik, günlük ve haftalık düzeylerde elektrikli araç şarj istasyonu kullanım verilerini analiz etmeyi ve farklı modelleme teknikleriyle talep tahmini gerçekleştirmeyi amaçlamaktadır. Böylece enerji yönetimi, altyapı planlaması ve kullanıcı deneyimi açısından geliştirilebilir çözümler önerilerek, Türkiye’deki elektrikli araç şarj ekosisteminin daha sürdürülebilir hâle getirilmesi hedeflenmektedir.

1.2 Literatür Taraması

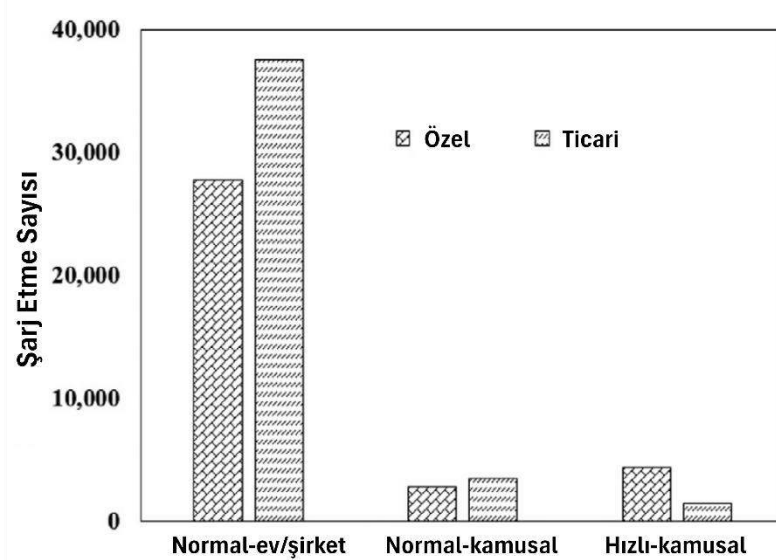
Elektrikli araçların küresel düzeyde hızlı bir şekilde benimsenmesi, ulaşım ve enerji sistemlerinde köklü değişimlere yol açmaktadır. Bu dönüşüm, elektrikli araç şarj altyapısının planlanması, yönetimi ve optimizasyonu konularında yoğun bir araştırma ilgisini de beraberinde getirmiştir. Literatürde, şarj istasyonu kullanım eğilimlerinin modellenmesine

yönelik yapılan çalışmalar genel olarak dört temel başlık altında sınıflandırılmaktadır: kullanıcı davranışlarının analizi, ampirik veri tabanlı çalışmalar, talep tahmin modelleri ve simülasyon-analitik yaklaşımlar.

1.2.1 Kullanıcı Davranış Analizleri

Elektrikli araç kullanıcılarının şarj davranışları üzerine yapılan anket ve gözlemsel çalışmalar, bireylerin şarj tercihlerini belirleyen temel etkenleri ortaya koymaktadır. Hardman ve arkadaşları [41], kullanıcıların şarj istasyonu seçiminde şarj hızı, istasyon konumu, fiyatlandırma yapısı ve güvenlik gibi unsurları dikkate aldıklarını vurgulamışlardır. Avrupa Alternatif Yakıtlar Gözlemevi'nin (EAFO - European Alternative Fuels Observatory) 2023 yılı raporuna göre [42], Avrupa'daki elektrikli araç kullanıcılarının büyük bir bölümü evde şarj etmeyi tercih etmekte, buna karşılık uzun mesafeli yolculuklarda hızlı şarj istasyonlarına olan talep belirgin bir şekilde artmaktadır.

Şekil 1.1'de görüldüğü üzere Xu ve arkadaşları [43], Japonya'da gerçekleştirdikleri çalışmada, batarya doluluk oranı, gece boyunca şarj etme imkânı ve geçmiş deneyimlerin, kullanıcıların şarj tipi ve konum seçimleri üzerinde belirleyici olduğunu tespit etmişlerdir. Wolbertus ve arkadaşları [44] da, kullanıcıların kamuya açık şarj istasyonlarını kullanma kararlarını etkileyen temel faktörler arasında erişilebilirlik, bekleme süresi ve güvenlik gibi unsurların önemli rol oynadığını belirtmişlerdir.



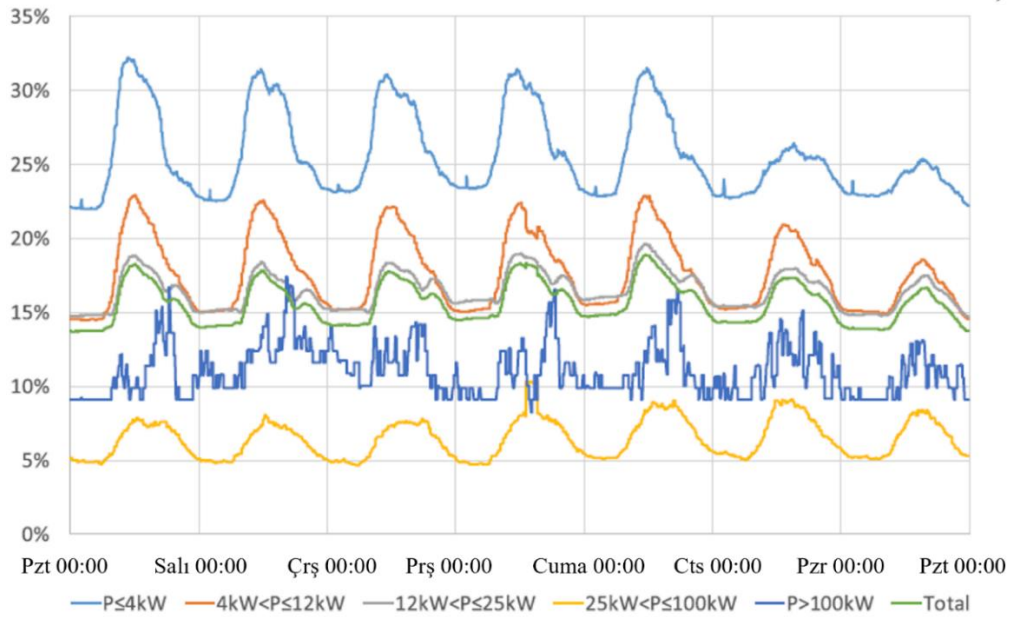
Şekil 1.1 : Özel ve ticari elektrikli araç kullanıcılarının normal (ev/şirket), kamusal normal ve kamusal hızlı şarj oturum sayılarının karşılaştırılması [45].

Öte yandan, Plug In America'nın gerçekleştirdiği 2022 EV Sürücü Anketi [45], kullanıcıların şarj istasyonlarında yaşadığı başlıca sorunların düşük erişilebilirlik ve bekleme

süreleri olduğunu, dolayısıyla bu faktörlerin kullanıcı memnuniyeti üzerinde doğrudan etkili olduğunu ortaya koymuştur. Türkiye bağlamında, Durmuş Şenyapar ve arkadaşları [46], şarj altyapısının yaygınlaşmasının elektrikli araçların benimsenmesini doğrudan etkilediğini ve şarj istasyonu konumlandırmalarının kullanıcı davranışları üzerinde kritik rol oynadığını göstermiştir.

1.2.2 Ampirik Veri Analizi

Gerçek şarj verilerine dayalı ampirik çalışmalar, kullanıcı eğilimlerini daha objektif bir şekilde analiz etme imkânı sunmaktadır. Hecht ve arkadaşları [47], Almanya genelinde 22.200'den fazla şarj istasyonundan elde edilen kullanım verilerini analiz ederek, AC istasyonlarında ortalama dört saatlik şarj oturumlarının ve DC hızlı şarj istasyonlarında yaklaşık 45 dakikalık oturumların yaygın olduğunu ortaya koymuşlardır. Şekil 1.2'de görülen çalışma, hızlı şarj cihazlarının daha kısa sürede daha fazla enerji aktarımı sağladığını ve özellikle uzun yolculuklar için tercih edildiğini göstermektedir.



Şekil 1.2 : Hecht ve arkadaşlarının gerçekleştirmiş olduğu çalışmadaki güç derecesine göre meşgul olan şarj istasyonu konnektörlerinin haftalık düzeyde ortalamaları [49].

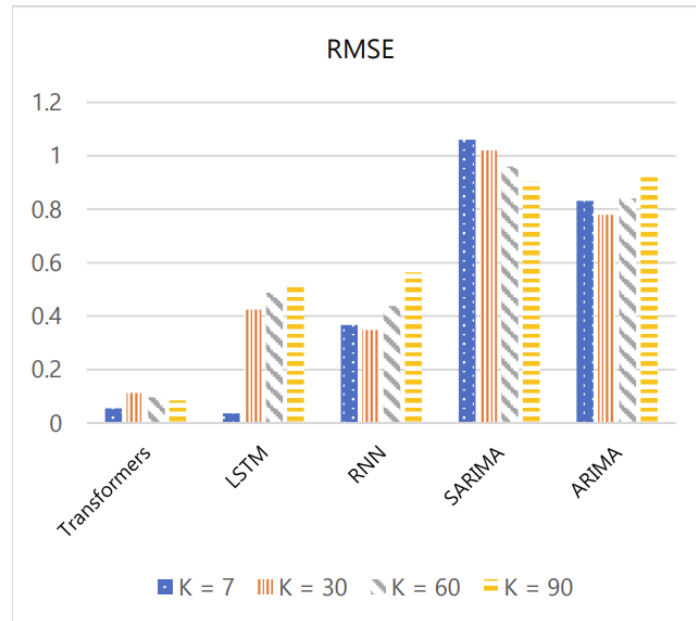
İrlanda'da gerçekleştirilen bir diğer çalışmada, Weldon ve arkadaşları [48], sabah işe gidiş ve akşam eve dönüş saatlerinde şarj talebinin zirve yaptığını ve DC şarj cihazlarının gün ortası kullanımının yoğunlaştığını tespit etmişlerdir. Zhan ve arkadaşları [49] ise Çin'de yedi büyük şehirden elde ettikleri geniş çaplı verilerle, araç segmentine göre farklı şarj davranışlarının gözlemlendiğini ve ticari filoların gün içerisindeki yoğun kullanımının kamu şarj altyapısına büyük yük bindirdiğini rapor etmişlerdir.

Amsterdam’da yapılan bir çalışmada Wolbertus ve arkadaşları [50], bataryası dolmasına rağmen şarj istasyonunda uzun süre bekleyen araçların altyapı kullanım verimliliğini düşürdüğünü ortaya koymuşlardır. Bu tür çalışmalar, şehirler arası karşılaştırmalar yapılarak şarj davranışlarındaki bölgesel farklılıkların anlaşılmasına da katkı sağlamaktadır.

1.2.3 Talep Tahmin Modelleri

Elektrikli araç şarj talebinin gelecekteki eğilimlerinin öngörülmesi amacıyla kullanılan yöntemler, zaman serisi analizlerinden makine öğrenmesi ve derin öğrenme modellerine kadar geniş bir yelpazeye yayılmaktadır. Wang ve arkadaşları [51], Çin’deki bir şarj istasyonunda elde edilen verilerle LSTM tabanlı modellerin, klasik ARIMA modellerine kıyasla daha düşük hata oranlarına ulaştığını göstermiştir. Özellikle kısa vadeli tahminlerde, geçmiş kullanım örüntülerini dikkate alan derin öğrenme tabanlı yöntemlerin daha başarılı olduğu belirtilmiştir.

Koohfar ve arkadaşları [52], Şekil 1.3’te görüldüğü üzere Boulder bölgesindeki kamu şarj istasyonlarına ait günlük verileri kullanarak Transformer tabanlı derin öğrenme modellerinin performansını incelemiş ve geleneksel yöntemlere kıyasla daha üstün sonuçlar elde etmişlerdir. Bu çalışmalar, hava durumu, gün tipi, etkinlik yoğunluğu gibi faktörlerin de model performansını artırdığına işaret etmektedir.



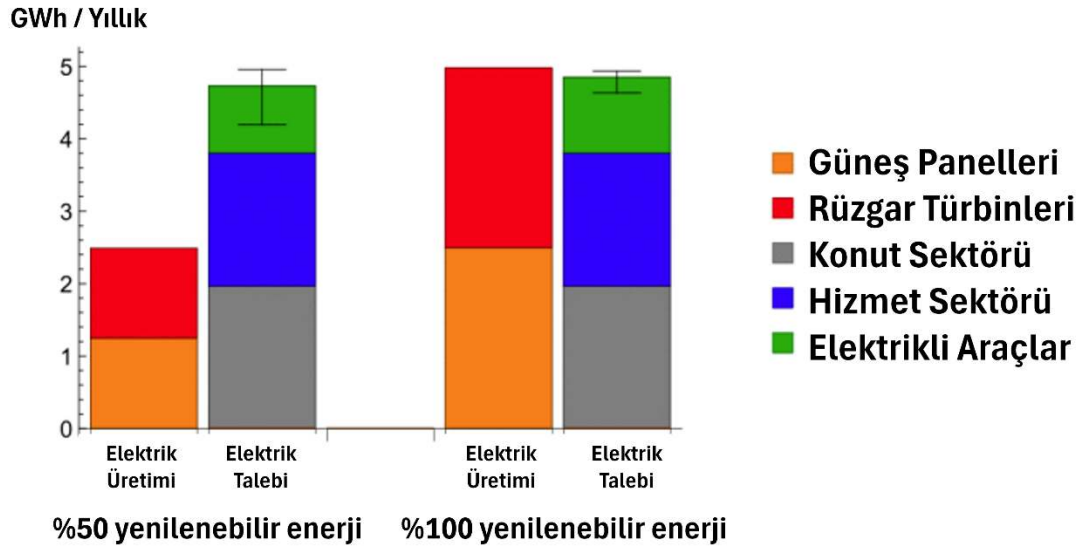
Şekil 1.3: Koohfar ve arkadaşlarının yapmış olduğu çalışmadaki Boulder bölgesindeki kamu şarj istasyonlarına ait günlük veriler kullanılarak farklı tahmin ufuklarında (K = 7, 30, 60, 90) Transformer, LSTM, RNN, SARIMA ve ARIMA modellerinin RMSE değerlerinin karşılaştırılması [54].

Ostermann ve Haug [53], Almanya genelinde 350.000'den fazla şarj oturumunu analiz ederek olasılıksal tahmin modelleri geliştirmişlerdir. Bu modeller, şebeke operatörleri için belirsizlik yönetimi ve risk analizi açısından kritik bilgiler sunmaktadır.

1.2.4 Simülasyon ve Analitik Yaklaşımlar

Şarj istasyonu performansının ve kullanıcı davranışlarının farklı senaryolar altında modellenebilmesi için simülasyon tabanlı yaklaşımlar yoğun olarak kullanılmaktadır. Karasu Aşnaz ve Özdemir [54], Balıkesir Üniversitesi kampüsünde gerçekleştirdikleri çalışmada kuyruk teorisi temelli modellerle şarj istasyonu yerleşimi optimizasyonu yapmış ve bekleme sürelerinin azaltılmasına yönelik önerilerde bulunmuşlardır.

Van der Kam ve arkadaşları [55], Şekil 1.4'de görüldüğü üzere Hollanda'da bir şehir simülasyonu üzerinde gerçekleştirdikleri ajan tabanlı modelleme çalışmasında, fiyat teşviklerinin ve akıllı şarj politikalarının kullanıcı davranışları üzerinde nasıl etkili olduğunu analiz etmişlerdir. Huang ve Kockelman [56] ise Boston bölgesinde ulaşım ağı verileri kullanarak hem trafik yoğunluğu hem de şarj istasyonu kuyruklarının entegre bir modellemesini gerçekleştirmişler ve az sayıda yüksek kapasiteli hızlı şarj istasyonlarının sistem verimliliğini artırabileceğini göstermişlerdir.



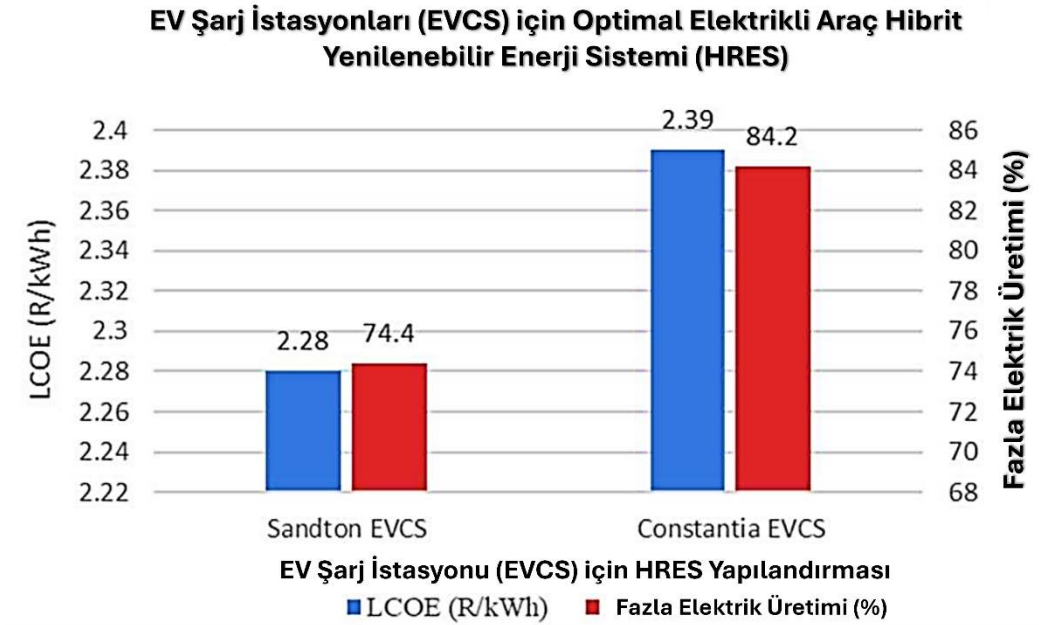
Şekil 1.4 : Van der Kam ve arkadaşlarının yapmış olduğu çalışmadaki %50 ve %100 yenilenebilir senaryolarında elektrik üretimi ve talep dağılımı [57].

Ayrıca, Shahraki ve arkadaşları [57] tarafından Çin'de gerçekleştirilen bir çalışmada, gerçek taksi hareketlilik verileri kullanılarak, hızlı şarj istasyonlarının optimal konumlandırılması için karma tam sayılı programlama modeli geliştirilmiştir.

1.2.5 Alternatif Yaklaşımlar: Batarya Takas, Hibrit Sistemler ve Yenilenebilir Enerji Entegrasyonu

Geleneksel şarj yöntemlerine alternatif olarak batarya değişim sistemleri ve hibrit şarj altyapıları da literatürde geniş yer bulmuştur. Celtek ve arkadaşları [58], makine öğrenimi tabanlı batarya sağlığı tahmini ve adil fiyatlandırma stratejileri geliştirerek batarya takas sistemlerinin etkinliğini artırmayı hedeflemişlerdir. Sheikdavood ve arkadaşları [59], IoT destekli hızlı şarj sistemleri ile farklı batarya kapasitelerine sahip araçlara eş zamanlı şarj hizmeti sunabilecek bir sistem tasarlamışlardır. Praveen ve arkadaşları [60] ise, kablolu ve kablosuz şarjı entegre eden çift yönlü hibrit sistemler tasarlayarak enerji verimliliğinde önemli iyileştirmeler sağlamışlardır.

Yenilenebilir enerji entegrasyonu kapsamında, Islam ve arkadaşları [61] ile Joshi ve Thosar [62], hibrit PV-BESS sistemlerinin hem maliyet hem de çevresel etkiler açısından avantajlarını ortaya koymuşlardır. Şekil 1.5'de Güney Afrika'da yapılan bir çalışmada Koopalame ve arkadaşları [63], adaya özgü fotovoltaik ve batarya kombinasyonlarının farklı coğrafi bölgeler için optimal sonuçlar sağladığını rapor etmişlerdir.



Şekil 1.5 : Sandton ve Constantia EV şarj istasyonları için hibrit PV-BESS'in LCOE (R/kWh) ve fazla elektrik üretimi [65].

Huang ve Guo [64], zaman dilimine dayalı şarj kontrol stratejileri ile trafolar üzerindeki yükün azaltılabileceğini göstermiştir. Mlondo ve arkadaşları [65] ise orta gerilim doğru akım (MVDC - Medium Voltage Direct Current) şebekeleri kullanılarak ultra hızlı şarj istasyonlarının başarıyla entegre edilebileceğini rapor etmiştir.

Son olarak, derin pekiştirmeli öğrenme yöntemleri kullanılarak geliştirilen gerçek zamanlı fiyatlandırma ve şarj zamanlaması modelleri, sistem performansının artırılması ve şebeke üzerindeki yükün dengelenmesi açısından umut verici sonuçlar sunmaktadır [53].

1.3 Tezin Kapsamı

Bu tez çalışması, elektrikli araç şarj istasyonlarında kullanım eğilimlerinin modellenmesi ve talep tahmini konularını kapsamlı bir biçimde ele almaktadır. Çalışmanın temel amacı, geçmiş dönemlere ait şarj kullanım verilerinden yararlanarak, farklı modelleme teknikleri aracılığıyla gelecekteki şarj talebinin öngörülmesini sağlamak ve bu doğrultuda şarj altyapısının yönetimi ile planlamasına bilimsel katkı sunmaktır.

Çalışma kapsamında, hem klasik zaman serisi modelleme yöntemleri hem de modern makine öğrenmesi ve derin öğrenme yaklaşımları kullanılmıştır. Bu çerçevede, ARIMA ve SARIMA gibi regresyon tabanlı modellerin yanı sıra, XGBoost, Random Forest, Support Vector Regression (SVR - Destek Vektör Regresyonu), Gradient Boosting Regressor (GBR - Gradyan Artırmalı Regresyon) ve LightGBM gibi güçlü makine öğrenmesi algoritmaları ile LSTM (Long Short-Term Memory - Uzun Kısa Süreli Bellek) gibi derin öğrenme temelli yöntemler uygulanmıştır. Ayrıca, Prophet modeli gibi alternatif modelleme tekniklerine de yer verilerek kapsamlı bir karşılaştırmalı analiz yapılmıştır.

Çalışmada kullanılan veriler, elektrikli araç şarj istasyonlarından elde edilen saatlik, günlük ve haftalık bazda şarj oturumu bilgilerini içermektedir. Veri seti dört farklı versiyon hâlinde oluşturulmuştur: ham saatlik veriler (V1), günlük toplamalar (V2), haftalık toplamalar (V3) ve saatlik ayrıştırılmış veriler (V4). Böylece, farklı zaman çözünürlüklerinde modelleme yapılabilmesi ve yöntemlerin bu çözünürlükler üzerindeki performanslarının değerlendirilebilmesi sağlanmıştır.

Modelleme sürecinde, veri ön işleme, özellik mühendisliği ve hiperparametre optimizasyonu gibi adımlar sistematik bir şekilde gerçekleştirilmiştir. Python ve MATLAB tabanlı yazılım altyapıları kullanılarak, veri hazırlama, model eğitimi, tahmin üretimi ve sonuçların analiz edilmesi aşamaları tasarlanmıştır. Hiperparametre aramaları sayesinde her

model için en uygun yapılandırmalar belirlenmiş ve performans değerlendirmeleri RMSE (Root Mean Squared Error - Kök Ortalama Kare Hata), MAE (Mean Absolute Error - Ortalama Mutlak Hata) ve MAPE (Mean Absolute Percentage Error - Ortalama Mutlak Yüzde Hata) gibi hata metrikleri kullanılarak gerçekleştirilmiştir.

Çalışmanın bir diğer önemli hedefi, farklı modelleme tekniklerinin çeşitli zaman çözünürlüklerinde nasıl performans gösterdiğini sistematik bir biçimde karşılaştırmaktır. Bu doğrultuda, her veri seti versiyonu için modellerin tahmin doğrulukları analiz edilmiş ve en başarılı yöntemler belirlenmiştir. Böylece, elektrikli araç şarj istasyonlarındaki kullanım eğilimlerinin tahmini için hangi modelleme stratejisinin daha uygun olduğu hakkında kapsamlı çıkarımlar yapılabilmektedir.

Tez çalışması ayrıca, kullanım eğilimlerinin öngörülmesinde makine öğrenmesi ve derin öğrenmenin sağladığı avantajları ve sınırlılıkları ortaya koyarak, gelecekteki araştırmalar için yol gösterici nitelikte bir perspektif sunmayı amaçlamaktadır. Bu kapsamda, şarj altyapısının verimli yönetimi, yük dengelemesi, talebe dayalı dinamik fiyatlandırma gibi uygulamalara katkı sağlayabilecek analizler sunulmuştur.

Sonuç olarak, bu tez çalışması; elektrikli araç şarj istasyonlarında kullanım eğilimlerini veri temelli olarak modelleyen, farklı algoritmaların karşılaştırmalı performansını ortaya koyan ve enerji yönetimi açısından stratejik planlamalara temel teşkil edebilecek kapsamlı bir araştırmayı içermektedir. Elde edilen bulgular, yalnızca mevcut altyapıların daha etkin işletilmesi için değil, aynı zamanda gelecekte kurulacak yeni şarj ağı planlamaları için de bilimsel dayanak oluşturmaktadır.

2. TAHMİN YÖNTEMLERİNE GENEL BAKIŞ

Elektrikli araç şarj istasyonlarındaki kullanım eğilimlerinin doğru şekilde tahmin edilmesi, enerji altyapısının verimli yönetimi ve şebeke stabilitesinin sağlanması açısından kritik öneme sahiptir [66–70]. Literatürde EA şarj talebini öngörmeye yönelik çeşitli tahmin yöntemleri geliştirilmiş olup, bu yöntemler genel olarak istatistiksel zaman serisi modelleri, geleneksel makine öğrenmesi algoritmaları ve derin öğrenme tabanlı yöntemler olarak üç ana kategoriye ayrılmaktadır [71–73]. İstatistiksel yaklaşımlar, tarihsel verilerdeki trend ve mevsimsellik gibi yapıları matematiksel modeller ile açıklamaya çalışmakta; buna karşılık makine öğrenmesi yöntemleri ve özellikle son yıllarda öne çıkan derin öğrenme yaklaşımları, verideki doğrusal olmayan ilişkileri ve karmaşık örüntüleri öğrenerek daha esnek tahmin modelleri oluşturmaktadır [51, 74, 75].

EA şarj talep tahmini probleminin zorlukları arasında mevsimsel değişimler, kullanıcı davranışlarının rastgeleliği ve veri yetersizliği sayılabilir [76, 77]. Erken dönem çalışmalarda gerçek zamanlı şarj verilerinin sınırlı olması nedeniyle çoğu model, ulaşım alışkanlıkları veya trafik verileri gibi dolaylı göstergelere dayalı tahminler geliştirmiştir [78, 79]. Günümüzde ise şarj istasyonlarından toplanan yüksek çözünürlüklü verilerin artmasıyla, daha doğrudan ve veri odaklı tahmin modelleri mümkün hale gelmiştir [26, 80]. Literatürde, farklı yöntemlerin performanslarını karşılaştıran çalışmalar da bulunmaktadır [23, 81]. Örneğin bazı araştırmalarda geleneksel istatistiksel modellerin belirli koşullarda makine öğrenmesi modellerine yakın veya üstün performans sergileyebildiği gösterilmiştir [82, 83]. Ancak verideki karmaşık ve doğrusal olmayan ilişkilerin güçlü olduğu durumlarda, ileri makine öğrenmesi veya derin öğrenme yöntemleri genellikle daha düşük hata oranları sağlamaktadır [84]. Aşağıda, zaman serisi modelleme temelleri ve bu üç ana kategori kapsamındaki yöntemler detaylı olarak ele alınmakta; ayrıca literatürde EA şarj talebi tahmininde bu yöntemlerin nasıl uygulandığı ve karşılaştırıldığı örnek çalışmalarla açıklanmaktadır.

2.1 Zaman Serisi Modelleme Temelleri

Zaman serisi modelleme, bir sistemin zaman içinde ardışık olarak ölçülen verilerinde bulunan paternlerin incelenmesi ve geleceğe yönelik tahminler yapılması amacıyla kullanılan bir yöntem olarak öne çıkmaktadır [85, 86]. Zaman serilerinde genel olarak dört temel bileşen yer almaktadır; uzun vadeli eğilim [85, 87], belirli periyotlarla tekrarlayan

mevsimsel dalgalanmalar [88, 89], trend ve mevsimsellik dışında gözlenen düzensiz dalgalanmalar [90, 91] ve rastgele bileşen. Zaman serisi modelleri, bu bileşenleri matematiksel olarak ifade ederek gelecekteki değerlerin tahmin edilmesini sağlamaktadır. Modelleme sürecinde öncelikli olarak serinin durağanlığı, yani istatistiksel özelliklerinin zamana bağlı olarak değişmezliği, incelenmektedir. Gerekli durumlarda fark alma gibi işlemler uygulanarak seri durağan hâle getirilmektedir [92].

Klasik zaman serisi modelleme yöntemleri arasında Box-Jenkins yaklaşımı önemli bir yere sahip bulunmaktadır [93]. Bu yaklaşım, otokorelasyon ve kısmi otokorelasyon analizleri aracılığıyla uygun model yapısının belirlenmesini, model parametrelerinin tahmin edilmesini ve ardından tanı testleri ile model geçerliliğinin kontrol edilmesini içermektedir [94, 95]. Zaman serisi modelleri, özellikle elektrik talebi gibi belirgin dönemsel karakteristikler sergileyen verilerin modellenmesinde yaygın olarak kullanılmaktadır [96, 97]. Elektrikli araç şarj istasyonlarının kullanım verileri, genellikle günlük ve haftalık döngüler göstermekte olup, iş günleri ile hafta sonları veya gün içerisindeki saatlik farklılıklar gibi değişimler gözlemlenebilmektedir [83]. Bu tür mevsimsel desenlerin yakalanabilmesi için zaman serisi modelleri içerisinde mevsimsel bileşen içeren uzantılar kullanılmaktadır [23, 98, 99].

Elektrikli araç şarj talebinin öngörülmesinde zaman serisi modelleri uzun yıllardır temel bir araç olarak kullanılmaktadır [52, 67, 83, 100]. Nitekim literatürde, elektrikli araç şarj yükünün tahmin edilmesine yönelik ilk çalışmaların büyük çoğunluğunun klasik zaman serisi yöntemlerini benimsediği görülmektedir [100, 101]. Bu yöntemler, verinin istatistiksel özelliklerine dayanarak nispeten yorumlanabilir ve açıklanabilir modeller sunmaları nedeniyle tercih edilmektedir [86, 102, 103]. Örneğin, ARIMA veya mevsimsel ARIMA (SARIMA) modelleri kullanılarak belirli bir şarj istasyonunun gelecekteki saatlik enerji talebi, geçmiş talep değerleri temel alınarak öngörülebilmektedir [52, 75, 83]. Ancak zaman serisi modellerinin genellikle doğrusal ilişkileri yakalamada başarılı olduğu, buna karşılık verideki karmaşık ve doğrusal olmayan desenleri yakalamada yetersiz kalabildiği bilinmektedir [104]. Bu gibi durumlarda ise makine öğrenmesi tabanlı yöntemlere başvurulmaktadır [53, 105].

Bu alt bölümde, öncelikli olarak sık kullanılan üç temel istatistiksel zaman serisi modeli ele alınmaktadır: ARIMA, mevsimsel uzantısı SARIMA ve Holt-Winters Üçlü Üstel Düzgünleştirme yöntemi.

2.2 Regresyon Tabanlı Yöntemler

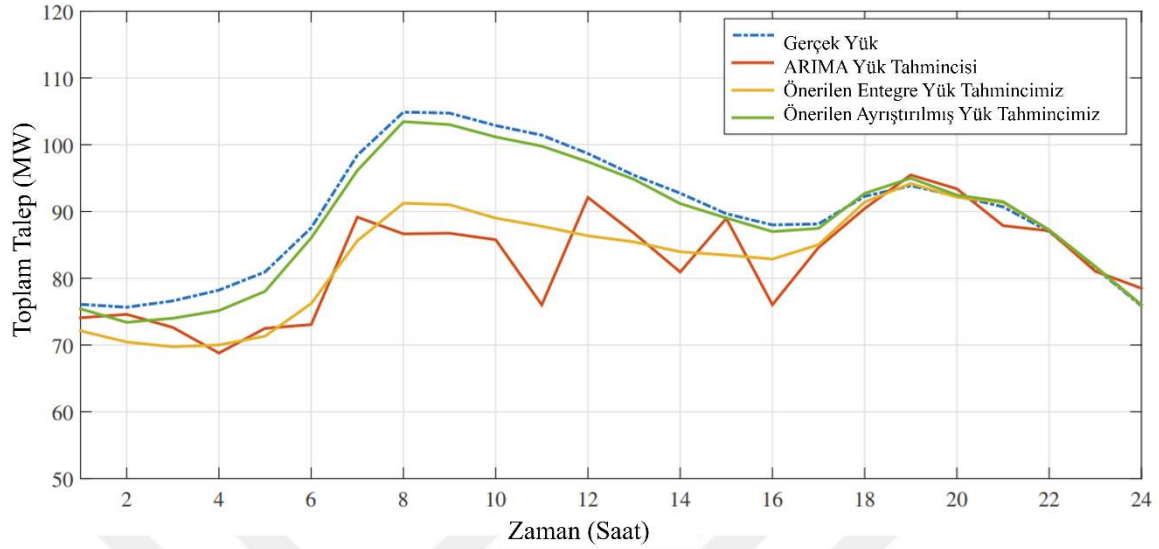
Regresyon tabanlı geleneksel yöntemler, zaman serisi verilerinde gelecekteki değerleri tahmin etmek amacıyla geçmiş gözlemlerin doğrusal kombinasyonlarını kullanmaktadır. Bu modeller, geçmiş değerlerin, hata terimlerinin veya mevsimsel bileşenlerin regresyonu yoluyla tahminler üretmektedir. Özellikle ARIMA, mevsimsel uzantısı SARIMA ve Holt-Winters Üçlü Üstel Düzgünleştirme yöntemleri, elektrik yükü tahmini alanında uzun yıllardır başarıyla uygulanmakta olup, elektrikli araç şarj talebi problemlerine de uyarlanmış bulunmaktadır [100, 106, 107]. Bu bölümde, söz konusu yöntemlerin teorik temelleri açıklanmakta ve EA şarj istasyonu talep tahminine yönelik literatürdeki uygulama örnekleri sunulmaktadır.

2.2.1 ARIMA

ARIMA (Autoregressive Integrated Moving Average - Otoregresif Bütünleşik Hareketli Ortalama) modeli, Türkçede Bütünleşik Otoregresif Hareketli Ortalama modeli olarak bilinmekte olup, durağan bir zaman serisinin otoregresif (AR - Autoregressive) ve hareketli ortalama (MA - Moving Average) bileşenlerini bir araya getirmektedir [108]. ARIMA modeli üç parametre ile ifade edilmektedir: otoregresif bileşenin derecesi “p”, seriyi durağanlaştırmak amacıyla uygulanan fark alma sayısı “d” ve hareketli ortalama bileşeninin derecesi “q”. AR(p) bileşeni, serinin önceki “p” değerleri ile yapılan regresyonu; MA(q) bileşeni ise önceki “q” hata teriminin regresyonunu temsil etmektedir [109, 110]. “d” derecesinde uygulanan fark alma işlemi, seride mevcut olan trend veya mevsimsellik kaynaklı durağan olmayan yapının giderilmesini amaçlamaktadır [110]. Sonuç olarak, ARIMA(p,d,q) modeli, serinin geçmiş “p” değeri ile son “q” hata teriminin doğrusal bir kombinasyonu üzerinden gelecek değeri tahmin etmektedir [111, 112].

ARIMA modelleri, kolay uygulanabilirlikleri ve istatistiksel anlamlandırılabilirlikleri sayesinde birçok talep tahmin probleminde temel yöntem olarak sıklıkla tercih edilmektedir [113–116]. Elektrikli araç şarj talebi tahmininde de ARIMA modeli, literatürde karşılaştırma ölçütü veya başlangıç modeli olarak yer almaktadır [75, 100, 106]. Örneğin, Şekil 2.1’de görüldüğü üzere Amini ve arkadaşları, elektrikli araç şarj talebini klasik elektrik yükünden ayrıştırarak her biri için ayrı ARIMA modelleri geliştirmiş ve bu ayırık yaklaşımın güç sistemi operasyonunda belirsizliği azaltmada etkili olduğunu göstermiştir [100]. Benzer şekilde, bazı çalışmalar düz ARIMA modeli ile kısa dönemli (örneğin bir sonraki gün için 15 dakikalık dilimler hâlinde) EA şarj talebini tahmin etmeyi

denemiş ve özellikle veri uzun vadeli bir trend barındırmadığında ve yeterince durağan olduğunda makul sonuçlar elde edildiğini rapor etmiştir [110, 112, 117].



Şekil 2.1 : Amini ve arkadaşlarının yapmış olduğu çalışmadaki saatlik gerçek yük ile ARIMA, önerilen entegre ve ayrıştırılmış yük tahminlerinin karşılaştırması [102].

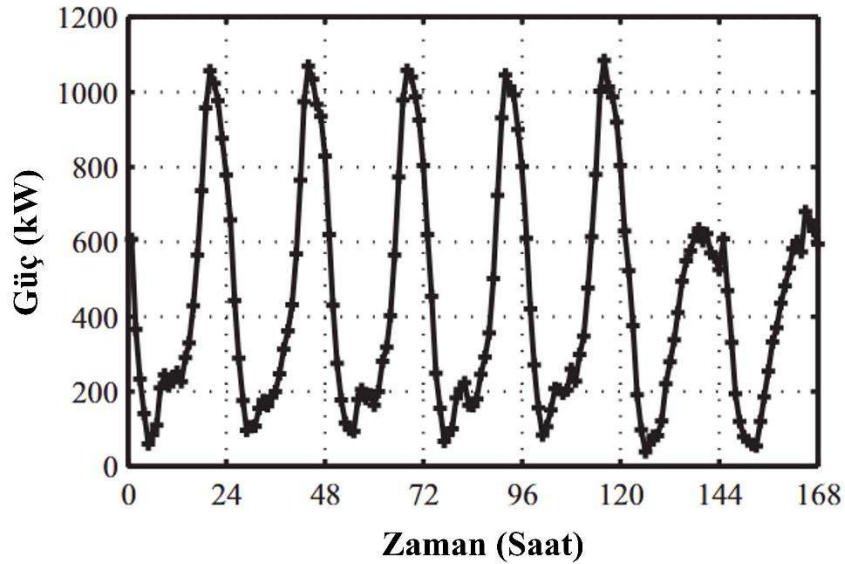
Bununla birlikte, ARIMA modelinin performansı, doğrusal olmayan örüntülerin baskın olduğu veri yapılarında sınırlı kalabilmektedir [86, 118]. Elektrikli araç şarj istasyonu verileri, gün içi kullanım profillerinde bazen oldukça düzensiz ve ani pik değerler sergileyebilmektedir [119]. Bu tür ani değişimlerin yakalanmasında ARIMA modelinin yetersiz kalabildiği, model hatalarının art arda büyümesi durumunda tahmin performansının düştüğü gözlemlenmektedir [111, 120]. Nitekim literatürde, ARIMA modelinin EA şarj talebi tahmininde daha esnek yöntemlere kıyasla daha yüksek hata oranları verdiği durumlar rapor edilmiştir [121, 122]. Örneğin, Majidpour ve çalışma arkadaşları, bir üniversite kampüsündeki şarj istasyonları verilerini kullanarak ARIMA modeli ile k-en yakın komşu (kNN) yöntemini karşılaştırmış ve çok kısa dönemli tahmin problemlerinde kNN yönteminin ARIMA modeline göre daha iyi sonuçlar verdiğini göstermiştir [123].

Yine de ARIMA modeli, özellikle veri setinin küçük olduğu veya örnek sayısının sınırlı olduğu durumlarda, kompleks modellerin aşırı öğrenme riski karşısında güvenilir bir başlangıç noktası sağlamaktadır. Elektrikli araç şarj talebi literatüründe ARIMA modelleri genellikle mevsimsel uzantıları olan SARIMA modelleri ile birlikte kullanılmakta olup, bir sonraki bölümde detaylı şekilde ele alınacak SARIMA modeline temel teşkil etmektedir.

2.2.2 SARIMA

Elektrikli araç şarj verilerinde sıkça karşılaşılan günlük ve haftalık mevsimsellik gibi periyodik desenlerin modellenmesinde, ARIMA modelinin mevsimsel uzantısı olan SARIMA modeli yaygın bir şekilde kullanılmaktadır [106, 124]. SARIMA modeli, ARIMA modeline mevsimsel otoregresif (P), mevsimsel fark alma (D) ve mevsimsel hareketli ortalama (Q) bileşenleri ile bir mevsim periyodu (s) ekleyerek tanımlanmaktadır [75]. SARIMA(p,d,q)(P,D,Q)s formatındaki bir model, s uzunluklu bir periyot (örneğin s=24 saat veya s=7 gün) için mevsimsel AR ve MA terimlerini içermektedir. Bu yapı sayesinde model, hem kısa vadeli hem de mevsimler arası korelasyonları yakalayabilmektedir [106].

Elektrikli araç şarj istasyonu kullanımında belirgin günlük döngüler gözlemlendiği için (örneğin her gün sabah ve akşam belirli saatlerde şarj talebinin zirve yapması), SARIMA modeli bu alanda doğal bir yaklaşım olarak öne çıkmaktadır [125]. Louie [106], iki yıllık bir veri seti kullanarak 2400'ün üzerinde şarj istasyonunun toplu talebini mevsimsel ARIMA modeli ile modellemiş ve mevsimsel bileşen eklemenin tahmin doğruluğunu artırdığını göstermiştir. Bu çalışma, SARIMA modelinin geniş ölçekli şarj ağı talebini öngörmede dahi makul bir başarı elde edebildiğini ortaya koymaktadır. Şekil 2.2'de Louie'nin yapmış olduğu çalışmada örnek 7 günlük zaman serisi görülmektedir.

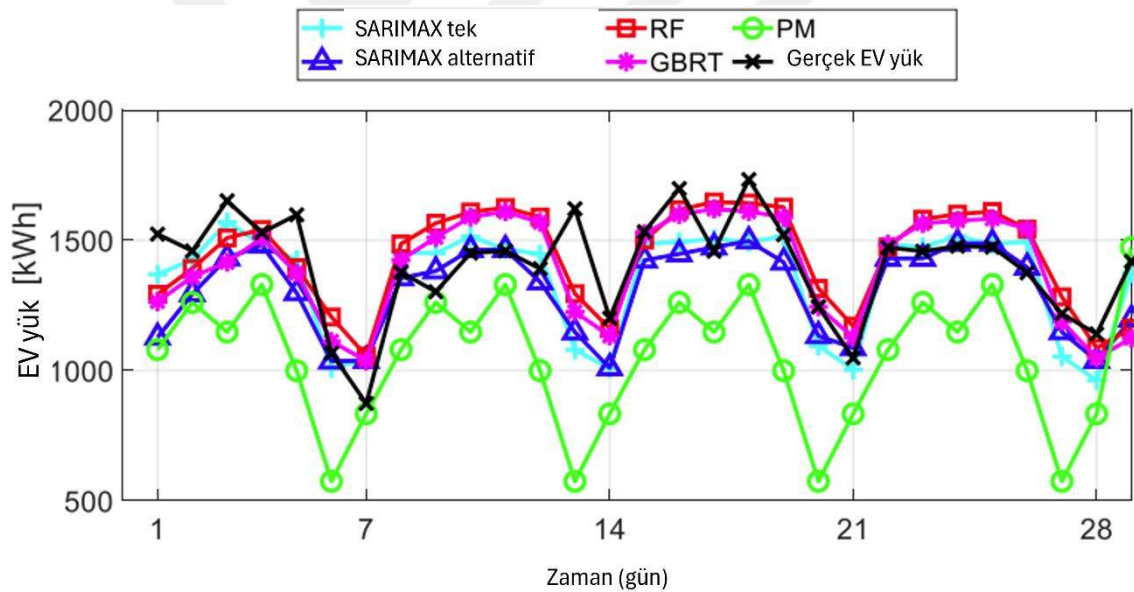


Şekil 2.2 : Louie'nin çalışmasında bulunan kullanım süresi fiyatlandırması olmaksızın 1800 adet EV şarj istasyonuna sahip bir sistemin sentetik 7 günlük zaman serisi [108].

SARIMA modellerinin bir avantajı, mevsimsel paternleri açık bir şekilde modele dahil etmeleri nedeniyle sonuçların yorumlanabilirliğini artırmalarıdır [126]. Örneğin, modelde anlamlı bulunan bir mevsimsel AR(1) terimi, "dünkü aynı saatteki talebin bugünkü

talep üzerinde güçlü bir etkisi bulunduğunu" ifade etmekte olup, bu tür çıkarımlar sistem yöneticilerine önemli içgörüler sunmaktadır.

Mevsimsel ARIMA modelleri, uygun bir şekilde kalibre edildiklerinde, bazı durumlarda daha karmaşık yöntemlerle rekabet edebilmektedir [67]. Buzna ve çalışma arkadaşları [73], 1700 istasyondan oluşan kapsamlı bir veri kümesi üzerinde gerçekleştirdikleri bir karşılaştırmada, mevsimsel dış değişkenli ARIMA modeli (SARIMAX) ile makine öğrenmesi tabanlı Random Forest ve Gradient Boosting modellerini kıyaslamış ve SARIMAX modelinin günlük şarj talebi tahmininde en düşük hata oranını verdiğini ortaya koymuştur. Bu sonuç, SARIMAX modelinin mevsimsel paternleri ve dışsal değişkenleri (örneğin hava durumu, takvim etkileri) modele entegre ederek güçlü bir performans sergilediğini göstermektedir. Şekil 2.3'de Buzna ve arkadaşlarının yapmış olduğu çalışmadaki 2 Mart 2015 ile 29 Mart 2015 arasındaki 28 günlük gerçek ve tahmin EV yükler grafiği bulunmaktadır.



Şekil 2.3 : Gerçek EV yük deseni ve 2 Mart 2015'ten 29 Mart 2015'e kadar 28 günlük bir ufuk için tahmin edilen EV yükleri [75].

Bununla birlikte, bazı çalışmalar SARIMA modellerinin doğrusal yapısı nedeniyle, nöral ağ tabanlı modellere kıyasla karmaşık ve doğrusal olmayan ilişkileri yakalamada yetersiz kalabildiğini ortaya koymuştur [127]. Bu doğrultuda, son yıllarda SARIMA modelleri ile derin öğrenme yöntemlerini birleştiren hibrit yaklaşımlar geliştirilmektedir. Örneğin, Ren ve çalışma arkadaşları tarafından yürütülen bir çalışmada [121], SARIMA modeli önce serinin genel lineer eğilimini ve mevsimsel yapısını tahmin etmekte, ardından LSTM ağı bu tahmin hatalarını öğrenerek artık bileşeni modellemektedir. Bu SARIMA-

LSTM birleşik yaklaşımı, tek başına kullanılan modellere kıyasla tahmin hatasında önemli bir azalma sağlamıştır.

Özetle, SARIMA modelleri, elektrikli araç şarj talebinin öngörülmesinde önemli bir yer tutmakta ve özellikle belirgin mevsimsel desenler barındıran verilerde etkili sonuçlar sunmaktadır. Ancak veride ani değişimler veya doğrusal olmayan etkiler mevcut ise, yalnızca SARIMA modeli yetersiz kalabilmekte; bu durumda diğer yöntemlerle entegrasyon veya dışsal faktörlerin parametreler yoluyla modele dahil edilmesi (SARIMAX) gereklilik arz etmektedir.

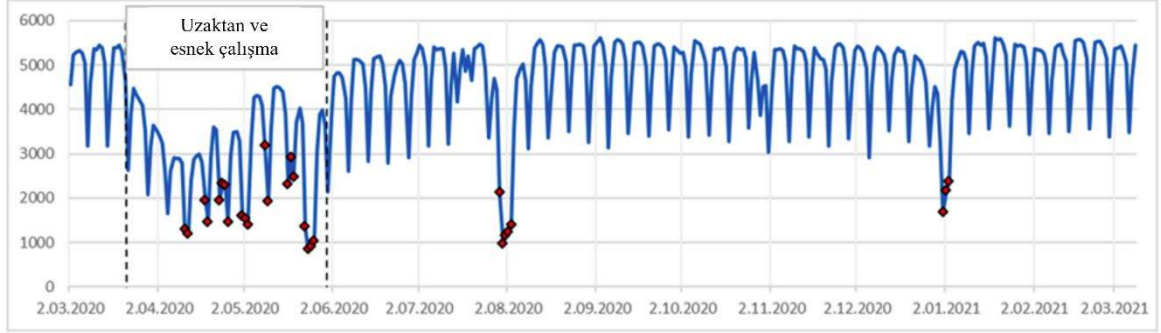
2.2.3 Holt-Winters

Holt-Winters yöntemi, zaman serilerinde yer alan seviye, trend ve mevsimsel bileşenleri üstel ağırlıklarla güncelleyerek tahmin yapmayı sağlayan bir yöntem olarak öne çıkmaktadır [128, 129]. Üçlü üstel düzgünleştirme olarak da bilinen bu yaklaşım, Charles Holt ve Peter Winters'ın çalışmaları ile geliştirilmiştir [130]. Holt'un çalışması, doğrusal trend içeren seriler için çift üstel düzgünleştirme formülasyonunu sunarken, Winters bu yapıyı bir mevsimsel bileşen ekleyerek genişletmiştir. Holt-Winters modeli, her yeni gözlem geldiğinde üç ayrı denklem kullanarak (seviye, trend ve mevsimsellik için) ilgili bileşenleri güncellemekte ve geleceğe yönelik tahminleri bu bileşenlerin birleşimiyle üretmektedir.

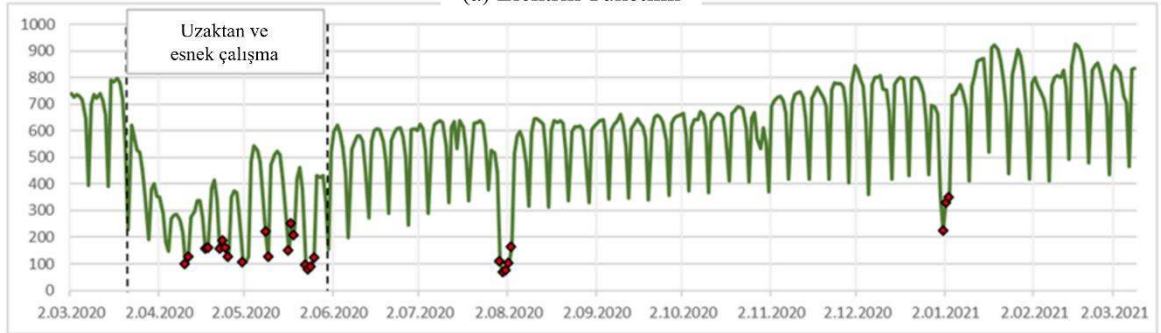
Bu yöntem, özellikle belirgin trend ve mevsim periyoduna sahip serilerde, basit ancak etkili sonuçlar sunmaktadır. Elektrik talep tahmini alanında Holt-Winters yöntemi uzun yıllardır kullanılmakta olup, bazı elektrikli araç şarj talebi çalışmalarında da karşılaştırma amacıyla bir model olarak yer almaktadır. Üstel düzgünleştirme yaklaşımı, ARIMA modellerine kıyasla daha az hesaplama gerektirebilmekte ve küçük örneklemeler üzerinde aşırı öğrenmeyi engelleyen yalın bir yapıya sahip bulunmaktadır. Örneğin, bir çalışmada EA şarj talebini öngörmek amacıyla başlangıçta Holt-Winters modeli kullanılmış; mevsimlerin (yaz/kış) ve yıl içi trendlerin dikkate alınmasıyla makul bir ilk tahmin elde edilmiştir [52].

Bununla birlikte, Holt-Winters modeli, veri desenlerinin zamanla değiştiği veya ani talep sıçramalarının gözlemlendiği durumlarda performans kaybı yaşamaktadır [129]. Model, geçmişte gözlemlenmemiş yeni bir davranış biçimini yakalamak için yeterli esnekliğe sahip değildir [131]. Nitekim, COVID-19 pandemisi gibi talep profilinde ani değişikliklere yol açan olaylar sonrasında üstel düzgünleştirme yöntemlerinin ciddi tahmin hataları ürettiği literatürde rapor edilmiştir [132]. Elektrikli araç şarj istasyonları özelinde de

kullanıcı alışkanlıklarındaki değişimler veya elektrikli araç sayısındaki hızlı artış gibi dinamik faktörler, Holt-Winters gibi sabit mevsimsel varsayımlara dayanan yöntemlerin güncel verilere uyum sağlamasını zorlaştırabilmektedir [133].



(a) Elektrik Tüketimi



(b) Doğal gaz tüketimi

Şekil 2.4 : COVID-19 karantinalarının tüketim miktarlarındaki Bölge 1 için değişiklikler (a) Elektrik ve (b) Doğalgaz [135].

Sonuç olarak, Holt-Winters yöntemi, basitliği ve yorumlanabilirliği sayesinde önemli bir geleneksel tahmin aracı olmayı sürdürmektedir. Ancak, EA şarj talebi gibi dinamik ve evrimsel süreçlerde bu yöntemin genellikle daha gelişmiş yöntemlerle desteklenerek kullanılması gerektiği anlaşılmaktadır. Örneğin, bazı çalışmalarda Holt-Winters yöntemi uzun vadeli trend ve mevsimsel yapıyı yakalamak için kullanılmış, kısa vadeli dalgalanmaların modellenmesi amacıyla makine öğrenmesi tabanlı yöntemler eklenmiştir [129]. Günümüzde Holt-Winters yöntemi, daha çok literatürde geliştirilen yeni yöntemlerin başarımını değerlendirmek üzere bir benchmark modeli olarak kullanılmaktadır.

2.3 Makine Öğrenmesi Yöntemleri

Makine öğrenmesi tabanlı yöntemler, zaman serisi tahmini problemine daha esnek ve veri odaklı bir yaklaşım getirmektedir [86, 134]. Bu yöntemler, istatistiksel modellerde olduğu gibi belirli bir fonksiyon biçimi varsaymak yerine, doğrudan veriden örüntüleri

öğrenmektedir. Denetimli öğrenme çerçevesinde, geçmiş verilere dayalı olarak eğitilen bir model, gelecekteki değerleri tahmin etmek amacıyla kullanılmaktadır.

Elektrikli araç şarj talebi tahmininde makine öğrenmesi algoritmaları, genellikle tarih, saat, gün tipi (hafta içi/hafta sonu) ve hava durumu gibi çeşitli özelliklerin girdi olarak kullanıldığı regresyon problemleri şeklinde uygulanmaktadır [23, 135]. Zamanın tek başına açıklayıcı değişken olarak kullanılmasının ötesinde, gecikmeli talepler veya hareketli ortalama gibi türetilmiş zaman serisi özellikleri de modele dahil edilmektedir. Bu sayede, klasik zaman serisi modellerindeki doğrusal ilişki kısıtlamaları aşılmakta ve verideki daha karmaşık yapılar öğrenilebilmektedir.

Makine öğrenmesi yöntemlerinin EA şarj talebi tahmininde kullanımına ilişkin literatürde çok sayıda örnek bulunmaktadır. Karar ağaçlarına dayalı ansambl (ensemble - topluluk) yöntemler, destek vektör makineleri ve yapay sinir ağları gibi modeller, farklı çalışmalar tarafından denenmiş ve performansları istatistiksel yöntemlerle karşılaştırılmıştır [135–137]. Genel bir gözlem olarak, makine öğrenmesi tabanlı modellerin, yeterli veri bulunduğu istatistiksel modellere kıyasla daha düşük tahmin hatası üretebildiği; ancak modelin doğru özelliklerle beslenmemesi durumunda aşırı öğrenme veya yetersiz genelleme gibi riskler taşıdığı belirtilmektedir [23].

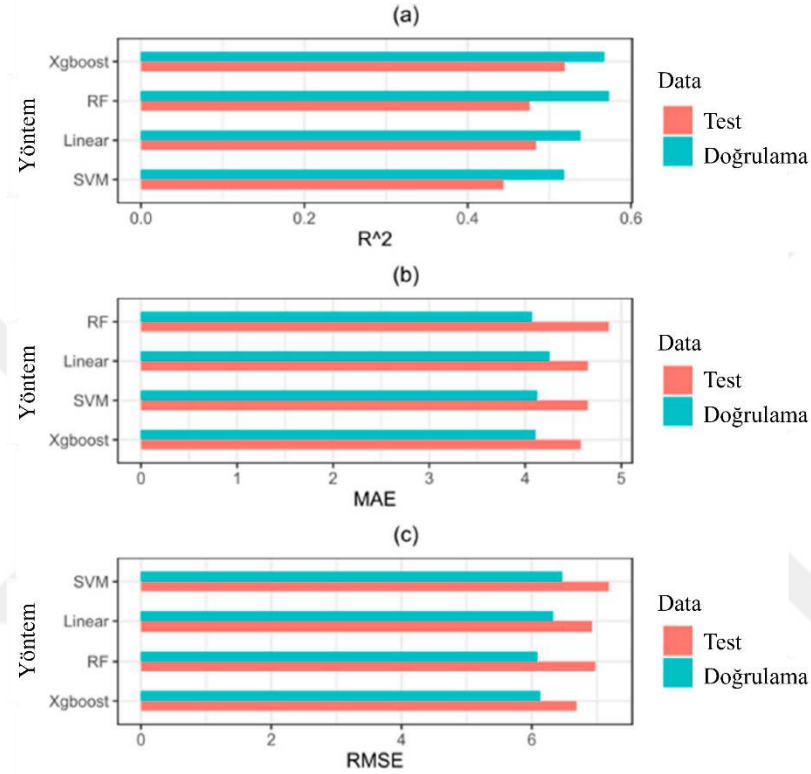
Bu alt bölümde, EA şarj istasyonu talep tahmininde öne çıkan başlıca makine öğrenmesi yöntemleri detaylı şekilde incelenmektedir.

2.3.1 XGBoost

XGBoost (Extreme Gradient Boosting - Ekstrem Gradyan Arttırma), karar ağaçları ansamblına dayalı güçlü bir makine öğrenmesi algoritması olarak öne çıkmaktadır. Gradient boosting yönteminin optimize edilmiş bir uygulaması olan XGBoost, her ağaç eklemesinde hatayı gradyan iniş yöntemiyle azaltarak modele yeni ağaçlar eklemektedir [138, 139]. Özellikle büyük veri setlerinde hızı ve doğruluğu ile dikkat çeken bu yöntem, doğrusal olmayan ilişkileri ve değişkenler arası etkileşimleri başarılı bir şekilde yakalayabilmektedir [139]. Ayrıca, özellik sayısının fazla olduğu problemler için de oldukça uygun bir yapı sunmaktadır.

Elektrikli araç şarj talebi tahmininde XGBoost, sıkça tercih edilen modellerden biri hâline gelmiştir. Yapılan karşılaştırmalı çalışmalar, XGBoost algoritmasının çoğu senaryoda diğer geleneksel makine öğrenmesi algoritmalarına kıyasla daha iyi performans sergilediğini göstermektedir. Almaghrebi ve çalışma arkadaşlarının gerçekleştirdiği bir çalışmada, yedi

yıllık halka açık şarj istasyonu verileri kullanılarak Random Forest, destek vektör makineleri ve XGBoost modelleri karşılaştırılmış; sonuçta Şekil 2.5’de görüldüğü üzere XGBoost modelinin en yüksek doğrulukla şarj talebi tahmini yaptığı rapor edilmiştir [135]. Bu çalışma, Nebraska, ABD eyaletindeki şarj istasyonlarından elde edilen veriler kullanılarak gerçekleştirilmiş olup, XGBoost’un özellikle mevsimsel paternleri ve kullanım alışkanlıklarındaki ani değişimleri yakalamada daha başarılı olduğu belirtilmiştir [135].



Şekil 2.5 : Almaghrebi ve arkadaşlarının yapmış olduğu çalışmadaki her yöntem için (a) R^2 , (b) MAE ve (c) RMSE dahil olmak üzere doğruluk metrik karşılaştırmaları [138].

Benzer şekilde, başka bir araştırmada ekstrem gradient boosting modellerinin EA şarj talebi öngörüsünde klasik ARIMA modeline kıyasla daha düşük ortalama hata ürettiği bulunmuştur [73]. XGBoost’un başarısının temel nedenlerinden biri, aşırı öğrenmeyi kontrol altında tutabilmesi ve ağaç tabanlı yapısı sayesinde verideki aykırı değerlerin etkisini azaltabilmesidir. Ayrıca, XGBoost modelleri doğal olarak özellik önem düzeylerini çıkarabilmekte ve böylece hangi faktörlerin şarj talebi üzerinde etkili olduğu konusunda değerli içgörüler sunabilmektedir. Örneğin, eğitilmiş bir XGBoost modelinde “gün içindeki saat” özelliğinin yüksek önem puanına sahip olması, şarj talebinde saatlik desenlerin belirleyici olduğunu göstermekte ve bu durum, sistem planlaması açısından önemli bilgiler sağlamaktadır.

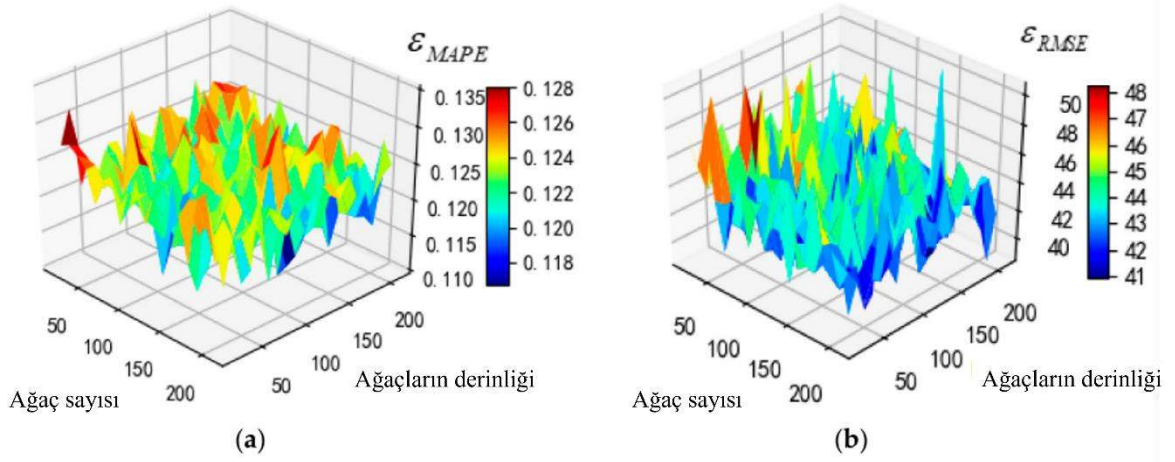
Öte yandan, XGBoost modellerinin doğru bir şekilde yapılandırılması ve hiperparametre optimizasyonunun yapılması zaman alıcı olabilmektedir. Ağaç sayısı, derinlik, öğrenme oranı gibi parametrelerin özenle belirlenmesi gerekmektedir. Ancak modern geliştirmeler ve açık kaynaklı kütüphaneler sayesinde, uygun bir hiperparametre araması ile yüksek performanslı modeller oluşturulabilmektedir.

Sonuç olarak, XGBoost algoritması, elektrikli araç şarj talebi tahminine yönelik çalışmalarda öne çıkan bir yöntem olarak değerlendirilmektedir. Birçok çalışmada en iyi sonuçları üreten model olarak rapor edilmekte olup [135], özellikle geniş veri setlerine ve çeşitli özelliklere sahip uygulamalarda yüksek başarı potansiyeli taşımaktadır.

2.3.2 Random Forest

Random Forest (Rastgele Orman) yöntemi, birden fazla karar ağacının oluşturulup sonuçlarının birleştirilmesi prensibine dayanan bir ansambl öğrenme yöntemi olarak öne çıkmaktadır [140]. Her bir karar ağacı, eğitim verisinin rastgele seçilmiş bir alt kümesi ve özelliklerin rastgele bir alt kümesi kullanılarak eğitilmekte; tahminler ise tüm ağaçların ortalamasının alınmasıyla elde edilmektedir. Bu yapı, tek bir karar ağacının yüksek varyans problemine çözüm sunarak daha kararlı ve genelleştirici modeller üretmektedir [140].

Elektrikli araç şarj talebi tahmininde Random Forest modeli, uygulama kolaylığı ve yorumlanabilirliği sayesinde sıkça tercih edilen yöntemlerden biri hâline gelmiştir. Örneğin, Lu ve çalışma arkadaşları [141], geliştirdikleri iyileştirilmiş bir Random Forest algoritmasını EA şarj yükünü öngörmek amacıyla kullanmışlardır. Bu çalışmada, orijinal rastgele orman yöntemi bazı ek adımlarla modifiye edilerek tahmin doğruluğunun artırılması hedeflenmiştir. Elde edilen sonuçlar, her şarj istasyonu için ayrı ayrı model oluşturma, tüm istasyonları tek bir model altında birleştirmeye kıyasla daha yüksek doğruluk sağladığını göstermektedir [141]. Bu bulgu, her istasyonun kendine özgü kullanıcı profili ve kullanım alışkanlıklarının bulunduğunu ve bu farklılıkların ayrı ayrı öğrenilmesinin faydalı olabileceğini ortaya koymaktadır. Şekil 2.6'da Lu ve arkadaşlarının gerçekleştirmiş olduğu çalışmadaki MAPE ve RMSE sonuçları görülmektedir.



Şekil 2.6 : Orman yapısı ile şarj istasyonu grubu tahminine göre hata arasındaki ilişki (a) Mape, (b) RMSE [144].

Random Forest modeli, neredeyse tüm önemli özellikleri otomatik olarak değerlendirip ağaçlar içerisinde kullandığı için, kapsamlı bir özellik mühendisliği süreci gereksinimini azaltmaktadır. Ayrıca, model çıktısı olarak özellik önem skorları sunabilmekte ve bu sayede hangi değişkenlerin şarj talebi üzerinde etkili olduğu konusunda sezgisel bilgiler elde edilmektedir. Örneğin bir çalışmada, Random Forest modeli hafta içi/hafta sonu bilgisi içeren bir özelliğin yüksek önem taşıdığını belirlemiş ve bu durum, hafta sonları şarj talebinde belirgin bir değişim olduğunu doğrulamıştır [135].

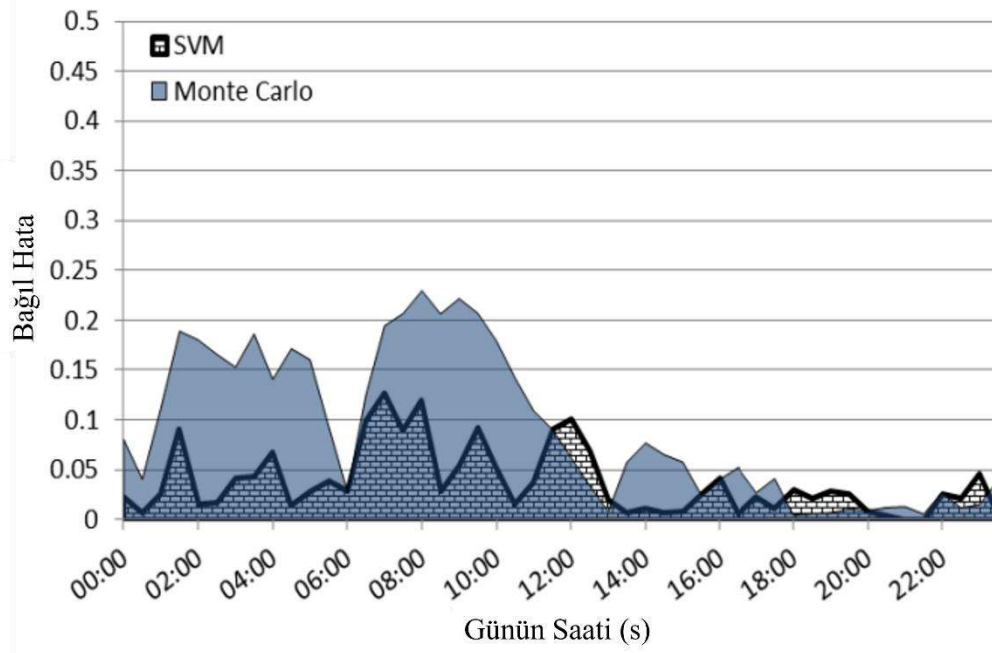
Bununla birlikte, Random Forest modelleri doğrusal olmayan karmaşıklığı başarılı bir şekilde yakalayabilse de, belirgin mevsimsel döngüleri doğrudan modelleyememektedir. Bu nedenle, mevsimsel davranışları doğru şekilde temsil edebilmesi için modelin yeterince derin ağaçlar oluşturması gerekebilmektedir. Bu durum bazen XGBoost gibi gradient boosting tabanlı yöntemlerin performansının gerisinde kalınmasına yol açabilmektedir. Nitekim, Almaghrebi ve arkadaşlarının gerçekleştirdiği bir çalışmada Random Forest modeli, XGBoost modeline kıyasla daha yüksek tahmin hatası üretmiştir [135].

Buna karşın, bazı araştırmalarda Random Forest yöntemi, özellikle verinin gürültülü olduğu durumlarda aşırı öğrenmeye karşı gösterdiği direnç sayesinde rekabetçi bir performans sergilemiştir [73]. Sonuç olarak, Random Forest modeli, EA şarj talebi tahmin çalışmalarında önemli bir yer tutmakta olup, birçok çalışmada hem tek başına hem de diğer yöntemlerle birlikte bir kıyas modeli olarak değerlendirilmektedir [73, 135]. Sağlam ve hızlı yapısı sayesinde, kompleks modellere geçmeden önce güvenilir bir karşılaştırma zemini sunmaktadır.

2.3.3 SVR

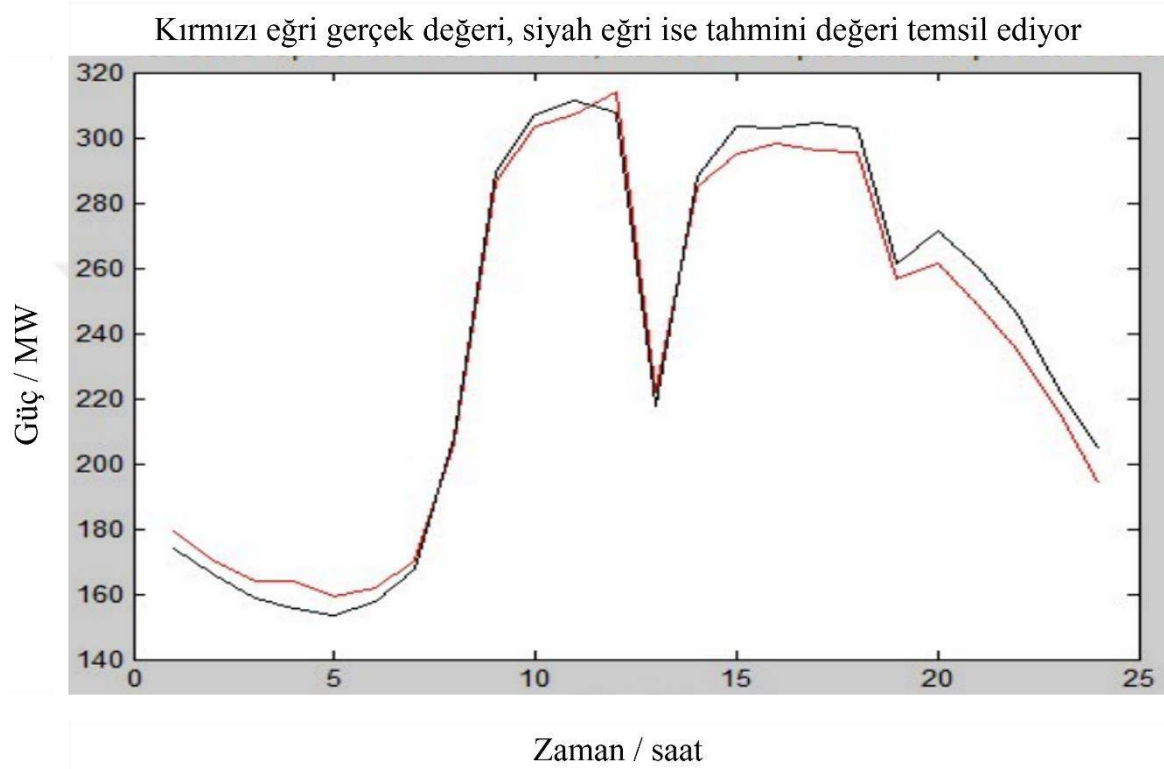
Destek Vektör Regresyonu, destek vektör makinelerinin regresyon problemlerine uyarlanmış bir versiyonu olarak kullanılmaktadır. SVR, kayıp fonksiyonunda belirli bir tolerans bandı tanımlayarak, bu bandın dışındaki hataları minimize eden bir optimizasyon süreci ile öğrenim gerçekleştirmektedir [142]. Temelinde doğrusal bir hiper düzlem oluşturarak regresyon yapmayı amaçlamakta; ancak çekirdek yöntemi sayesinde yüksek boyutlu uzaylarda doğrusal olmayan ilişkileri de modelleyebilmektedir. RBF (Radial Basis Function - Radyal Tabanlı Fonksiyon) çekirdeği, SVR modellerinde yaygın olarak kullanılmakta olup, verideki karmaşık desenlerin yakalanmasına olanak sağlamaktadır.

Elektrikli araç şarj talebi tahmini çalışmalarında SVR yöntemine özellikle erken dönem literatürde rastlanmaktadır. Örneğin, Xydas ve çalışma arkadaşları [143], 2013 yılında gerçekleştirdikleri bir çalışmada, İngiltere'deki bir bölgenin ev tipi şarj verilerini kullanarak EA şarj talebini kısa dönemli öngörmek amacıyla SVR modelini kullanmışlardır. Yapılan karşılaştırmalarda Şekil 2.7'de görüldüğü üzere SVR'nin karmaşık talep desenlerini başarıyla yakalayabildiğini; ancak model performansının seçilen çekirdek fonksiyonuna ve hiperparametrelere yüksek derecede duyarlı olduğunu ortaya koymuştur [143]. Özellikle epsilon ve C parametrelerinin uygun seçilmemesi durumunda, SVR modeli aşırı öğrenme veya yetersiz öğrenme gibi sorunlarla karşı karşıya kalabilmektedir.



Şekil 2.7 : Saatlik bazda SVR ve Monte Carlo yöntemlerinin göreceli hata karşılaştırması [146].

SVR yönteminin önemli bir avantajı, sınırlı sayıda eğitim örneği ile çalışabilmesi ve güçlü bir çözüm sunabilmesidir. Nitekim, Chen ve çalışma arkadaşları [144], düşük boyutlu bir elektrik yük veri seti üzerinde SVR uygulayarak başarılı sonuçlar elde etmişler ve zaman serisi verisi üzerinde gecikmeli değerler gibi yeniden yapılandırılmış özelliklerle kullanıldığında SVR'nin klasik yöntemlerden daha iyi performans sergileyebileceğini rapor etmişlerdir.



Şekil 2.8 : Geleneksel SVR'nin sonucu [147].

Bununla birlikte, büyük ölçekli ve yüksek frekanslı veriler söz konusu olduğunda, SVR'nin eğitim süresi oldukça uzayabilmektedir. Çünkü çözülmesi gereken optimizasyon problemi veri boyutuna göre ölçeklenmektedir. Günümüzdeki büyük hacimli EA şarj verileri düşünüldüğünde, SVR uygulamalarının pratikte bazı zorluklar yaratabildiği görülmektedir. Ayrıca, son yıllarda ağaç tabanlı ansambl yöntemlerin ve derin öğrenme tekniklerinin popülerlik kazanmasıyla birlikte, SVR yönteminin kullanım sıklığı bir miktar azalmıştır.

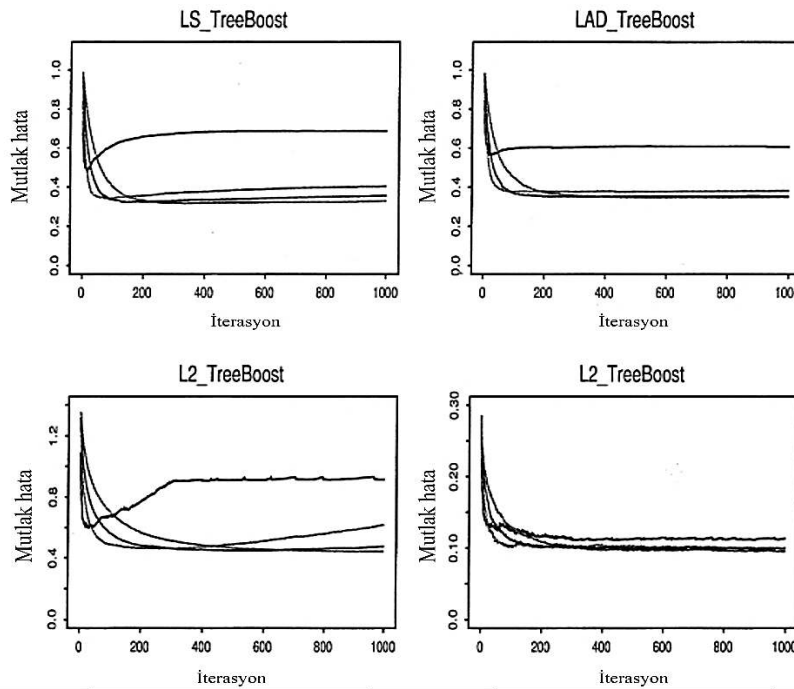
Almaghrebi ve çalışma arkadaşlarının [135] yaptığı bir çalışmada da SVR yöntemi test edilmiş, ancak diğer modern makine öğrenmesi algoritmalarına kıyasla daha düşük performans sergilediği gözlemlenmiştir. Buna rağmen, SVR, özellikle özellik sayısının fazla olduğu ve lineer ayrıştırılabilirlik varsayımının makul olduğu durumlarda güçlü bir alternatif olarak önemini korumaktadır. Bazı hibrit yaklaşımlarda ise SVR, diğer yöntemlerle birlikte

kullanılarak tahmin hatasının bir bileşeninin modellenmesi amacıyla tercih edilmektedir [145].

Özetle, SVR yöntemi, elektrikli araç şarj talebi tahmininde literatürde denenmiş ve orta düzeyde başarı göstermiş bir yöntem olarak değerlendirilmektedir. Karmaşık yapıdaki bu problemde her zaman en üst sırada yer almasa da, özellikle küçük ölçekli veri setlerinde veya karmaşık ilişkilerin daha net izlenebildiği durumlarda tercih edilmesi önerilmektedir [143, 144].

2.3.4 GBR

GBR terimi, genel olarak gradient boosting karar ağaçları yaklaşımını kullanarak regresyon yapan modelleri ifade etmektedir. Bu çerçevede XGBoost ve LightGBM gibi spesifik algoritmalar da gradient boosting ailesi içerisinde yer almakta; ancak burada GBR ile kastedilen, daha genel anlamda gradient boosting makinesi yöntemidir [138]. Friedman tarafından 2001 yılında ortaya konulan gradient boosting çerçevesi, her adımda önceki modelin hatalarını azaltmaya yönelik olarak yeni bir zayıf öğrenici eklenmesi prensibine dayanmaktadır [138]. Bu yaklaşım, ağaç sayısının artması ile eğitim verisine uyumu artırmakta; uygun erken durdurma veya düzenleme teknikleri uygulandığında ise genelleme yeteneğini koruyabilmektedir. Şekil 2.9’da Friedman tarafından yapılan çalışmadaki üç farklı gradient boosting algoritmasının performansı görülmektedir.



Şekil 2.9 : Üç farklı Gradient Boosting algoritmasının performansı [141].

Elektrikli araç şarj talebi tahmininde gradient boosting yöntemleri, yüksek doğrulukları sayesinde sık tercih edilmektedir. Yukarıda 2.3.1. kısmında ayrıntılı olarak ele alınan XGBoost, bu ailenin optimize edilmiş üyelerinden biridir. Bunun yanı sıra, Python gibi programlama ortamlarında bulunan sklearn kütüphanesindeki GradientBoostingRegressor gibi uygulamalar da literatürde kullanılmaktadır. Örneğin, Buzna ve çalışma arkadaşları [73] tarafından gerçekleştirilen bir çalışmada, “Gradient Boosted Regression Trees (GBRT - Gradyan Artırılmış Regresyon Ağaçları)” olarak adlandırılan model kullanılarak EA şarj talebi tahmin edilmeye çalışılmıştır. Sonuçlar, GBRT modelinin belirli senaryolarda ARIMA türevi modellerle rekabet edebildiğini, ancak mevsimsel veri desenlerini yakalama konusunda zaman zaman yetersiz kaldığını ortaya koymuştur [73]. Bu durum, gradient boosting yöntemlerinin veriye doğrudan uyum sağlama gücünü yansıtmakla birlikte, mevsimsellik ve uzun dönem trend gibi yapıları doğrudan modellemedikleri için bu tür örüntüleri yalnızca verinin kendisinden öğrenebildiklerini göstermektedir.

Literatürde XGBoost dışında CatBoost ve LightGBM gibi diğer gradient boosting algoritmaları da yaygın olarak kullanılmaktadır. CatBoost, özellikle kategorik değişkenlerin işlenmesinde sağladığı optimizasyonlar ile öne çıkarken; LightGBM, büyük veri setleri üzerinde daha hızlı çalışabilmek amacıyla hafifletilmiş bir gradient boosting türevi olarak geliştirilmiştir [146]. Sreekumar ve Lekshmi’nin 2024 tarihli çalışmalarında, EA şarj talebi tahmini amacıyla Random Forest, XGBoost, CatBoost ve LightGBM modelleri karşılaştırılmış ve bu karşılaştırmada en iyi performansın gradient boosting ailesine ait bir model tarafından elde edildiği rapor edilmiştir [101]. Bu bulgu, gradient boosting yaklaşımının EA şarj talebi problemlerindeki etkinliğini açıkça ortaya koymaktadır.

Gradient boosting yöntemlerinin önemli avantajlarından biri, hiperparametre optimizasyonu gerçekleştirildiği takdirde aşırı öğrenme riskini etkin bir şekilde yönetebilmeleridir. Ayrıca, ansambl yapıları sayesinde farklı türdeki verilerden yüksek öğrenme kapasitesi sergileyebilmektedirler. Bununla birlikte, ağaç sayısı, derinlik ve öğrenme oranı gibi hiperparametrelerin dikkatli bir şekilde ayarlanması gerekmektedir; aksi takdirde model gereğinden fazla karmaşık bir hâl alabilmekte ve genelleme yeteneğini kaybedebilmektedir.

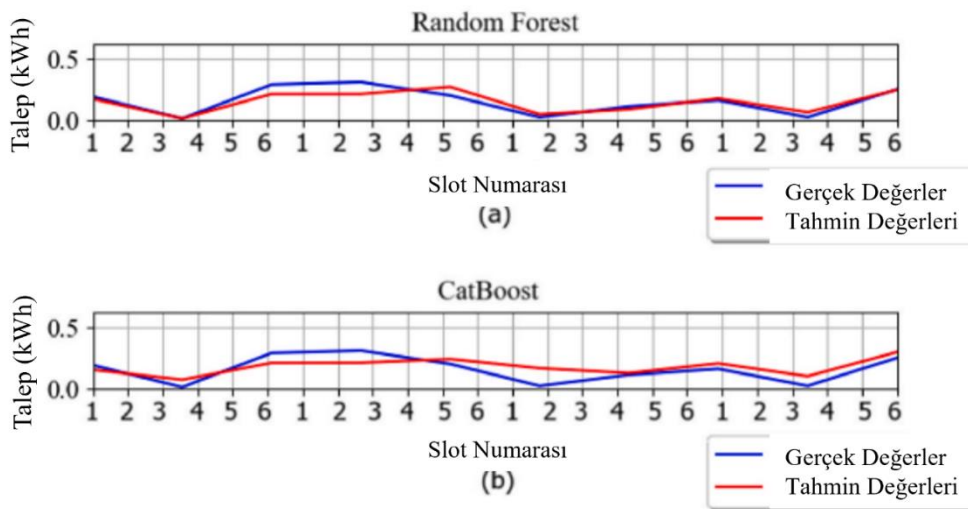
Sonuç olarak, gradient boosting tabanlı regresyon yöntemleri, elektrikli araç şarj talebi tahmini alanında en başarılı yöntemler arasında yer almaktadır [101, 135]. Özellikle

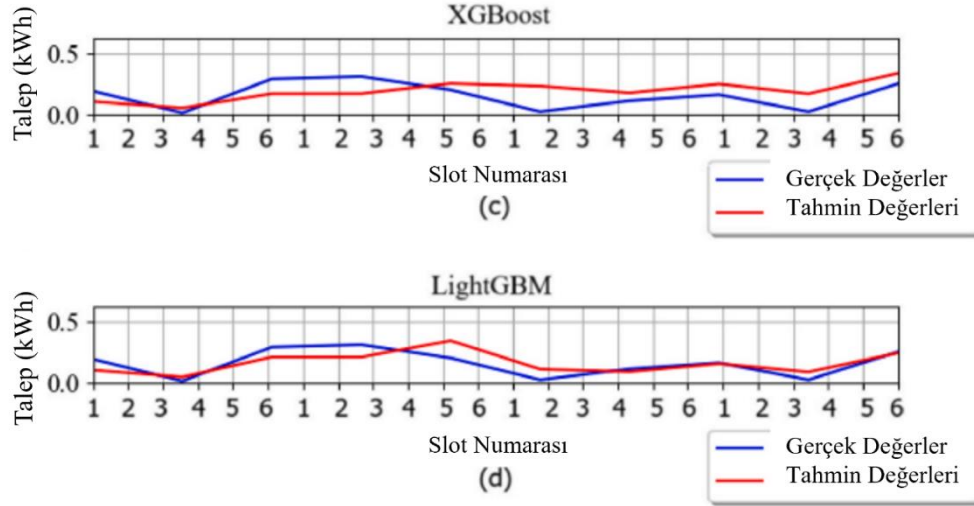
verinin gürültülü olduğu veya doğrusal olmayan etkileşimlerin bulunduğu durumlarda bu yöntemlerin üstün performans sergilediği literatürde sıklıkla rapor edilmektedir.

2.3.5 LGBM

LightGBM (Light Gradient Boosting Machine - Hafif Aşamalı Artırma Makinesi), gradient boosting karar ağaçları algoritmasının Microsoft tarafından geliştirilmiş hafif ve hızlı bir versiyonu olarak öne çıkmaktadır [146]. LightGBM'nin en önemli özelliği, klasik gradient boosting algoritmalarında kullanılan seviye bazlı bölünme stratejisi yerine, yaprak bazlı bölünme stratejisini kullanmasıdır. Bu yöntem sayesinde, benzer hata indirimleri daha az sayıda ağaç veya daha küçük derinliklerle sağlanabilmekte ve büyük veri setlerinde eğitim süresi önemli ölçüde kısaltılabilmektedir [146]. Aynı zamanda bellek kullanımı da optimize edilmekte olup, LightGBM çok büyük ölçekli veri setlerinde dahi etkili bir şekilde uygulanabilmektedir.

Elektrikli araç şarj talebi tahmini çalışmalarında LightGBM, son yıllarda literatürde giderek daha fazla yer bulmaya başlamıştır. Özellikle 2020 sonrası dönemde gerçekleştirilen çalışmalarda, XGBoost algoritmasına bir alternatif veya tamamlayıcı bir yöntem olarak LightGBM'nin de test edildiği gözlemlenmektedir. Örneğin, Sreekumar ve Lekshmi'nin 2024 yılında gerçekleştirdikleri çalışmada, Şekil 2.10'da görüldüğü üzere LightGBM diğer yöntemlerle birlikte değerlendirilmiş ve genel olarak boosting tabanlı modellerin EA şarj talebi tahmininde güçlü performanslar sunduğu vurgulanmıştır [101]. Bazı çalışmalarda, LightGBM'nin XGBoost'a yakın doğruluk sağlarken daha hızlı eğitim süresi sunduğu rapor edilmiştir. Bu durum, gerçek zamanlı uygulamalar veya çevrim-içi öğrenme gerektiren sistemler için önemli bir avantaj sunmaktadır.





Şekil 2.10 : (a) Rastgele orman (b) CatBoost (c) XGBoost (d) LightGBM regresyon modelleri kullanılarak Gerçek-Tahmin Edilen talep eğrisi [103].

LightGBM'nin EA şarj talebi tahminindeki kullanımına yönelik raporlar, özellikle çok sayıda özellik barındıran veri kümelerinde (örneğin takvim bilgileri, hava durumu verileri, gecikmeli talep değerleri gibi çok boyutlu veri setlerinde) modelin hesaplama verimliliğinin yüksek olduğunu göstermektedir [101]. Bununla birlikte, LightGBM de bir gradient boosting yöntemi olduğu için, optimum sonuçlar elde edebilmek adına hiperparametre ayarlarının dikkatli bir şekilde yapılması gerekmektedir. Yaprak bazlı büyüme stratejisi oldukça agresif bir yaklaşım sunduğundan, eğer uygun düzenleme parametreleri belirlenmezse, model aşırı karmaşık hâle gelebilmektedir. Bu sebeple, LightGBM kullanılırken model karmaşıklığını sınırlayıcı hiperparametrelerin dikkatle seçilmesi büyük önem arz etmektedir.

Literatürde doğrudan LightGBM'nin diğer yöntemlerden üstün olduğunu iddia eden çok sayıda çalışma bulunmamakla birlikte, genel eğilim, LightGBM'nin XGBoost ile kıyaslanabilir bir performans sunduğu ve özellikle eğitim süresi ve kaynak verimliliği açısından önemli avantajlar sağladığı yönündedir [101, 146].

Sonuç olarak, LightGBM algoritması, elektrikli araç şarj istasyonu talep tahmini problemlerinde yüksek potansiyel sunmakta olup, veri hacminin ve özellik çeşitliliğinin arttığı senaryolarda daha fazla ilgi göreceği öngörülmektedir.

2.4 Derin Öğrenme Yöntemleri

Derin öğrenme, çok katmanlı yapay sinir ağları aracılığıyla veriden yüksek seviyeli özelliklerin otomatik olarak çıkarılmasını sağlayan bir makine öğrenmesi alt alanı olarak öne

çıkılmaktadır. Zaman serisi tahmini problemlerinde derin öğrenme tekniklerinin kullanımı, karmaşık ve doğrusal olmayan ilişkilerin modellenmesinde önemli bir gelişim sağlamaktadır. Özellikle Tekrarlayan Sinir Ağları (Recurrent Neural Networks, RNN) ve onların geliştirilmiş versiyonları olan Uzun Kısa Süreli Bellek (Long Short-Term Memory, LSTM) mimarileri, sıralı verilerdeki zaman bağımlılıklarını öğrenmede son derece etkili olmaktadır [147].

Elektrikli araç şarj talebi gibi, gün boyunca ardışık ölçümlerin yapıldığı ve geçmiş değerlerin gelecekteki değerlere etkisinin önemli olduğu problemlerde, RNN tabanlı derin öğrenme yöntemleri doğal bir çözüm alternatifi sunmaktadır. EA şarj istasyonu talep tahmini çalışmalarında derin öğrenme yaklaşımları son yıllarda yoğun ilgi görmektedir. Geleneksel yöntemlerin aksine, derin öğrenme modelleri verideki trend ve mevsimsellik gibi bileşenleri önceden tanımlanmış formlar kullanmaksızın, doğrudan verinin kendisinden öğrenebilmektedir. Örneğin bir sinir ağı, yeterli miktarda veri sağlandığında, hafta içi ve hafta sonu davranış farklılıklarını ya da gün içerisindeki çift tepe noktası gibi kullanım desenlerini kendi iç yapısı ve ağırlıkları aracılığıyla temsil edebilmektedir.

Literatürde birçok çalışma, LSTM ve türevlerini kullanarak EA şarj talebi tahmini gerçekleştirmiş ve klasik modellere kıyasla önemli performans artışları rapor etmiştir [27, 52, 148]. Bu başarıların temelinde, derin ağların doğrusal olmayan ilişkileri ve uzun vadeli bağımlılıkları daha iyi yakalayabilme yetenekleri yatmaktadır.

Bununla birlikte, derin öğrenme yöntemlerinin uygulanmasında dikkat edilmesi gereken bazı hususlar da bulunmaktadır. Bu tür modeller genellikle çok sayıda parametre içermekte ve büyük veri setleri ile eğitilmeleri gerektiğinden, veri miktarının yetersiz olması durumunda aşırı öğrenme riski artmakta veya genelleme kabiliyeti azalmaktadır. EA şarj istasyonları verisi, özellikle istasyon bazında incelendiğinde sınırlı sayıda örnek içerebilmekte (örneğin bir istasyon için sadece birkaç yıllık veri bulunabilmektedir). Bu tür durumlarda, tüm istasyonların verilerinin ortak bir modelde birleştirilmesi veya veri artırma tekniklerinin kullanılması gibi stratejiler önerilmektedir [23].

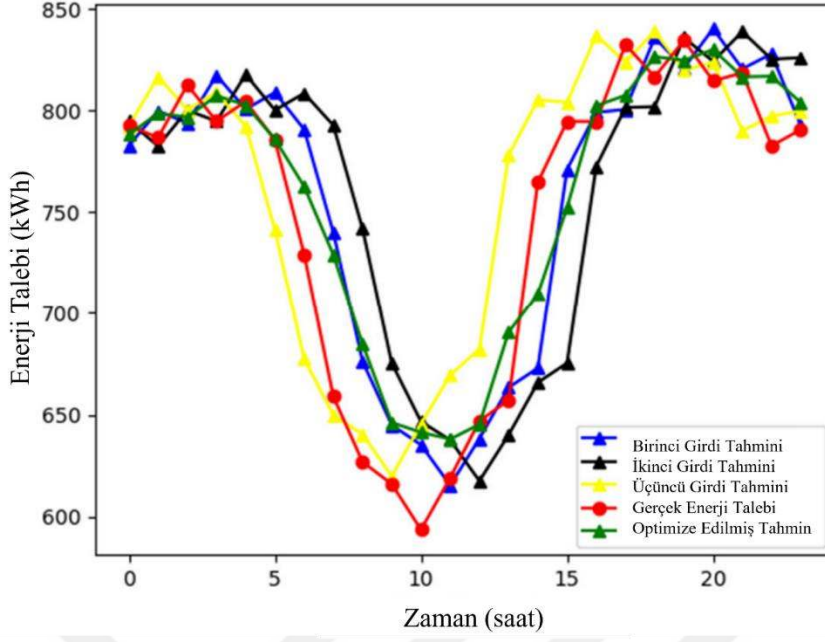
Ayrıca, derin öğrenme modelleri genellikle "kara kutu" yapıları nedeniyle, model çıktılarının yorumlanmasında güçlükler yaratabilmektedir. Ancak, son dönemde dikkat mekanizmaları ve benzeri teknikler kullanılarak bu modellerin yorumlanabilirliğinin artırılması yönünde çalışmalar yapılmaktadır [52].

Bu bölümde, derin öğrenme yaklaşımlarının EA şarj talebi tahminindeki kullanımına odaklanılacak ve özellikle en yaygın kullanılan mimari olan LSTM ağları detaylı bir şekilde ele alınacaktır.

2.4.1 LSTM

LSTM (Long Short-Term Memory - Uzun kısa süreli bellek) ağı, tekrarlayan sinir ağlarının eğitiminde yaşanan uzun vadeli bağımlılıkların öğrenilememesi ve geri yayılım sırasında gradyanların kaybolması problemlerini çözmek amacıyla geliştirilmiş özel bir mimariyi temsil etmektedir [147]. LSTM hücreleri, giriş kapısı, çıkış kapısı ve unutma kapısı olmak üzere üç ayrı kapı mekanizması barındırmaktadır. Bu kapılar, hücre durumunun nasıl güncelleneceğine karar vererek, ağına hangi bilgiyi ne kadar süreyle hatırlayacağına veya unutulacağına dinamik biçimde karar vermektedir. Bu yapı sayesinde LSTM, onlarca adım öncesine ait bilgiyi gerektiğinde muhafaza edebilmekte ve yeni girdiler ışığında esnek bir şekilde güncelleyebilmektedir.

Elektrikli araç şarj talebi tahmininde LSTM ağları, son derece başarılı sonuçlar elde etmiş ve literatürde geniş yankı bulmuştur. Birçok çalışma, LSTM tabanlı modellerin hem klasik zaman serisi yöntemlerini hem de geleneksel makine öğrenmesi algoritmalarını geride bıraktığını ortaya koymaktadır [27, 52, 148]. Örneğin, Aduama ve çalışma arkadaşları [27], çoklu özellik girişi ile eğittikleri bir LSTM modeli kullanarak, bir şarj istasyonunun kısa dönemli yük tahmininde ARIMA ve çeşitli makine öğrenmesi modellerine kıyasla belirgin bir üstünlük sağlamışlardır. Bu çalışmada, LSTM ağına yalnızca geçmiş talep verileri değil, aynı zamanda hava durumu ve takvim bilgileri gibi dışsal değişkenler de dahil edilmiştir. Elde edilen sonuçlar, çok değişkenli LSTM modelinin tek değişkenli modellere kıyasla daha düşük hata ürettiğini göstermekte olup, LSTM ağlarının farklı kaynaklardan gelen bilgileri bütünleştirerek karmaşık ilişkileri öğrenme yeteneğini vurgulamaktadır. Şekil 2.11’de elde edilen bu sonuçların karşılaştırılmasını içeren grafik görülmektedir.



Şekil 2.11 : LSTM sonuçlarının, optimize edilmiş sonuçların ve gerçek enerji talebinin karşılaştırılması [27].

LSTM ağlarının bir diğer önemli avantajı, sekans bazlı çalışarak veriyi doğrudan zaman sırasına göre işleyebilmesidir. Bu özellik, veri ön işleme sürecini basitleştirmekte ve örneğin mevsimsellik gibi örüntülerin manuel olarak modellenmesine gerek kalmadan, ağı eğiterek öğrenilmesini sağlamaktadır. Van Kriekinge ve çalışma arkadaşları [117], ertesi gün için EA şarj talebi tahmininde derin sinir ağları kullanmış ve geleneksel yöntemlere kıyasla üstün performans elde etmişlerdir. Benzer şekilde, Elahe ve çalışma arkadaşları [148], bir istasyondaki kısa dönemli şarj yükünü tahmin etmek amacıyla LSTM, çift yönlü LSTM (Bi-LSTM) ve CNN-LSTM (Convolutional Neural Network – Long Short-Term Memory - Evrişimsel Sinir Ağı – Uzun Kısa Süreli Bellek) gibi beş farklı derin öğrenme modelini karşılaştırmışlar; bu modeller arasında klasik LSTM mimarisinin en iyi sonuçlardan birini sunduğunu rapor etmişlerdir. Bu bulgu, çok karmaşık mimariler kullanmadan, probleme uygun şekilde yapılandırılmış bir LSTM ağının da son derece rekabetçi olabileceğini göstermektedir.

Bununla birlikte, LSTM modellerinin eğitimi hesaplama açısından yoğun kaynak gerektirmektedir ve hiperparametre ayarları (örneğin katman sayısı, hücre sayısı, öğrenme oranı) model başarımını önemli ölçüde etkilemektedir. Derin öğrenme modellerinde yaygın olarak uygulanan doğrulama kümesi hatasına dayalı erken durdurma, dropout ile düzenleme gibi teknikler, aşırı öğrenmenin önlenmesinde kullanılmakta ve EA şarj talebi tahmini problemlerinde de bu tekniklere sıklıkla başvurulmaktadır [27, 52]. Örneğin, ardışık iki

LSTM katmanı ve dropout içeren bir mimari ile, tek katmanlı bir LSTM modeline göre daha iyi genelleme başarısı elde edildiği gösterilmiştir [149].

LSTM ve türevleri, literatürde o kadar başarılı sonuçlar vermiştir ki, bazı araştırmacılar tarafından endüstriyel uygulamalarda da kullanılmaya başlanmıştır. Örneğin, Çin'de bir elektrikli otobüs filosunun şarj gereksinimlerini tahmin etmek amacıyla LSTM tabanlı bir model geliştirilmiş ve bu model, gerçek saha uygulamalarında belirli güven aralıkları içerisinde başarılı tahminler üreterek şebeke operasyonlarını desteklemiştir [150].

Güncel çalışmalarda, LSTM'nin yanı sıra Transformer tabanlı ağlar gibi daha yeni derin öğrenme yaklaşımları da denenmektedir. Örneğin, Koohfar ve çalışma arkadaşları [52], bir Transformer mimarisi kullanarak, LSTM'ye kıyasla daha hızlı eğitim süresine sahip ve benzer doğrulukta sonuçlar üreten bir model elde etmişlerdir. Ancak tüm bu gelişmelere rağmen, LSTM ağı, uygunluğu ve zaman serisi verileri üzerindeki kanıtlanmış başarımı sayesinde, elektrikli araç şarj talebi tahmininde hâlen en popüler ve etkili derin öğrenme yöntemlerinden biri olarak kullanılmaya devam etmektedir.

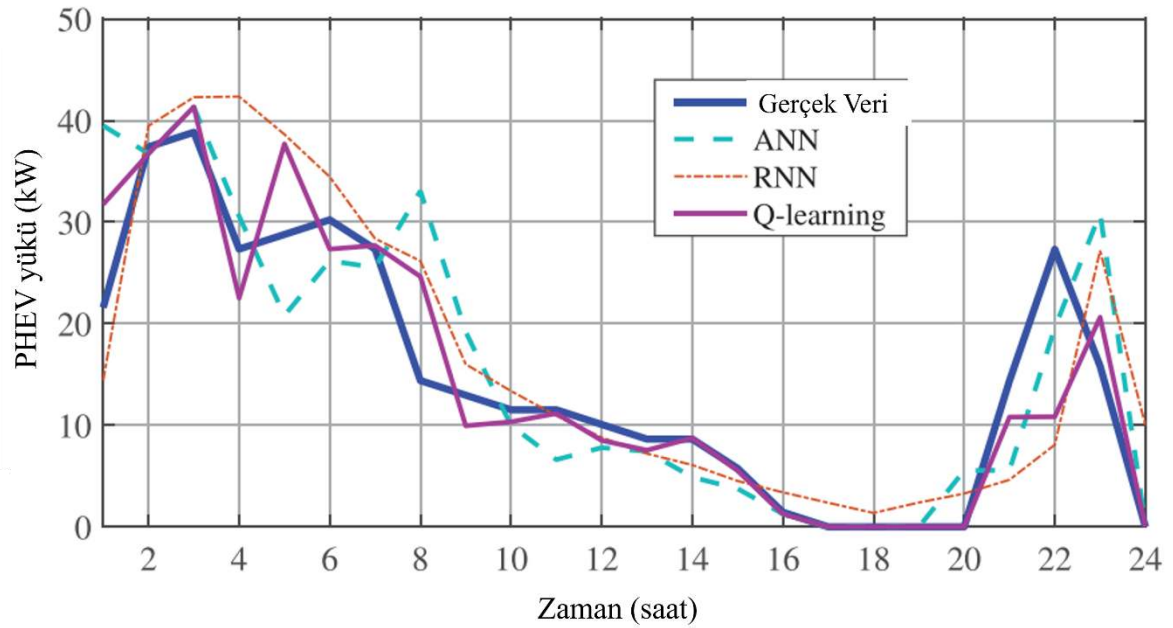
2.5 Diğer Yaklaşımlar

Yukarıda ele alınan yöntem kategorilerinin dışında, elektrikli araç şarj talebi tahmini problemine yönelik olarak çeşitli özel ve hibrit yaklaşımlar da literatürde yer almaktadır. Hibrit modeller, istatistiksel bir yöntem ile makine öğrenmesi veya derin öğrenme yöntemlerinden birinin ardışıl veya paralel şekilde birleştirilmesi prensibine dayanmaktadır. Bu tür kombinasyonlar, her iki yöntemin güçlü yönlerini bir araya getirerek daha hassas ve güvenilir tahminler elde etmeyi amaçlamaktadır. Örneğin, önceki bölümlerde ele alınan SARIMA-LSTM hibrit modeli [136], bu yaklaşımı temsil eden çalışmalardan biri olarak dikkat çekmektedir.

Benzer şekilde, literatürde “kümeleme + tahmin” şeklinde iki aşamalı yaklaşımlar da kullanılmaktadır. Bu yöntemlerde, ilk aşamada şarj istasyonları veya günler, benzer özelliklerine göre kümelenmekte; ikinci aşamada ise her bir küme için ayrı bir tahmin modeli eğitilmektedir [151]. Böylece, veri setindeki heterojenlik azaltılmakta ve her modele daha homojen bir veri sunularak tahmin performansının artırılması hedeflenmektedir [137, 151].

Bir diğer özel yaklaşım ise takviye öğrenmesi tabanlı tahmin yöntemleri olmaktadır. Örneğin, Dabbaghjamanesh ve çalışma arkadaşları [152], Q-öğrenme algoritmasını kullanarak bir şarj istasyonu yük tahmin modeli geliştirmişlerdir. Bu modelde, bir ajan,

geçmiş talep durumu ve gerçekleştirdiği eylemler temelinde ödüllendirilmekte ve bu mekanizma aracılığıyla gelecekteki şarj talebini öğrenmektedir. Takviye öğrenmesi tabanlı yaklaşımlar henüz EA şarj talebi tahmini literatüründe yaygınlık kazanmamış olmakla birlikte, özellikle şarj istasyonlarının akıllı kontrol sistemleri ile tahmin süreçlerinin bütünleştirildiği uygulamalarda ilginç bir alternatif sunmaktadır. Şekil 2.12’de Dabbaghjamesh ve çalışma arkadaşlarının yapmış olduğu çalışmadaki ANN, RNN ve Q-öğrenme algoritmalarının %30 penetrasyon ile tahmin ve gerçek karşılaştırması bulunmaktadır.



Şekil 2.12 : %30 penetrasyon ile akıllı PHEV zamana bağlı şarj tahmin sonuçlarının ANN, RNN ve Q-öğrenme algoritmalarıyla karşılaştırılması (kW) [155].

Bu alt bölümde, literatürde adı geçen ve geleneksel tahmin kategorilerine doğrudan girmeyen diğer yaklaşımlardan biri olan Prophet modeli ele alınmaktadır. Prophet, Facebook tarafından geliştirilmiş bir zaman serisi tahmin aracı olup, özellikle iş zaman serileri tahmini alanında yaygın kullanım bulmaktadır [153]. Elektrikli araç şarj talebi problemlerine de bazı çalışmalarda doğrudan uygulanmış ya da hibrit modellerin bir bileşeni olarak entegre edilmiştir.

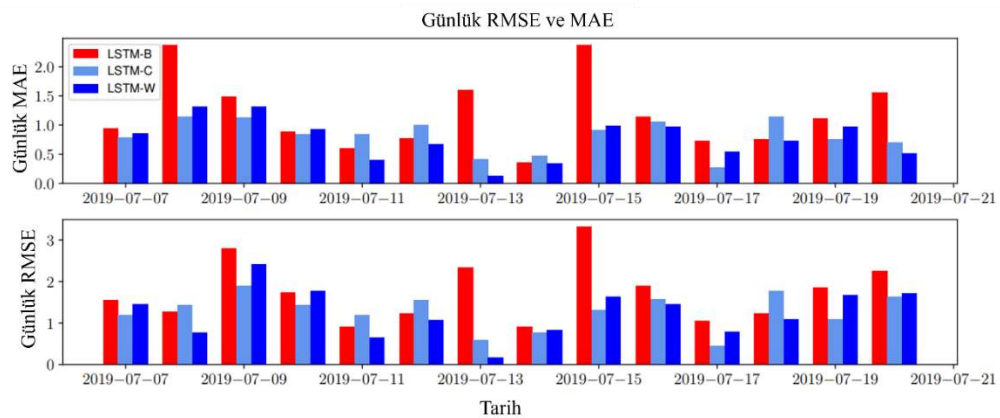
2.5.1 Prophet

Prophet, 2017 yılında Facebook tarafından geliştirilen ve kullanımı kolaylaştırılmış bir açık kaynaklı zaman serisi tahmin aracıdır [153]. Temelinde eklemeli bir model yaklaşımı yatmakta olup, zaman serisini trend, mevsimsellik, tatil/özel gün etkisi ve hata

bileşenlerine ayrıştırarak modellemektedir. Prophet modelinde, trend bileşeni için esnek bir dilimli doğrusal veya lojistik büyüme modeli; mevsimsellik için ise Fourier serileri kullanılmakta ve bu bileşenler toplamsal olarak birleştirilerek nihai tahmin üretilmektedir [153]. Modelin kullanıcı dostu olması amacıyla otomatik hiperparametre ayarlamaları ve farklı mevsimsel periyotları kendi kendine tespit edebilme özellikleri de bulunmaktadır.

Prophet'in en büyük avantajlarından biri, az sayıda veri noktası ile çalışabilmesi ve hızlı bir şekilde makul sonuçlar üretebilmesidir. Ayrıca, tahmin sonuçlarının bileşenlerine ayrılarak sunulması, "trendin yukarı yönlü olduğu" veya "hafta sonu mevsimselliğinin etkisinin belirli bir düzeyde olduğu" gibi yorumların kolaylıkla yapılmasına imkân tanımaktadır. Bu özellikleri sayesinde Prophet, iş dünyası uygulamalarında yaygın biçimde benimsenmiştir.

Elektrikli araç şarj talebi tahmininde Prophet modeli, literatürde çok yaygın olmamakla birlikte bazı uygulamalarda yer almıştır. Özellikle karma yaklaşımlarda Prophet'in belirli bileşenleri yakalamak için kullanıldığı gözlemlenmektedir. Örneğin, bir çalışmada Prophet modeli öncelikle eğitilerek mevsimsel ve trend etkileri tahmin edilmiş, ardından kalan düzensiz bileşenler çift yönlü LSTM (BiLSTM) modeline verilerek nihai tahmin üretilmiştir [117]. Bu hibrit Prophet-BiLSTM yaklaşımının, tek başına kullanılan BiLSTM modeline göre daha düşük tahmin hatası ürettiği rapor edilmiştir. Burada Prophet modeli, verinin daha deterministik kısımlarını (örneğin dönemsel paternleri) yakalarken, LSTM modeli stokastik ve karmaşık ilişkileri öğrenmektedir. Şekil 2.13'de 7-21 Temmuz 2019 tarihleri arasında BiLSTM-B, BiLSTM-C ve BiLSTM-W modellerinin günlük MAE ve RMSE değerlerinin karşılaştırması sunulmaktadır.



Şekil 2.13 : BiLSTM varyantlarının 07-21 Temmuz 2019 dönemindeki günlük MAE ve RMSE karşılaştırması [119].

Prophet'in tek başına kullanılması durumunda, mevsimsel paternlerin belirgin olduğu ve verinin fazla gürültü içermediği senaryolarda iyi performans sergilediği bilinmektedir. Ancak elektrikli araç şarj talebi verileri, özellikle tatil günleri veya beklenmedik etkinlikler gibi faktörlerin etkisiyle zaman zaman oldukça oynak bir yapı gösterebilmektedir. Prophet modeli, bu tür ani değişimleri öngörmekte zorlanabilmekte; çünkü temel olarak sabit yıllık/haftalık desenler ve yavaş değişen trendler varsaymaktadır.

Bu nedenle Prophet modeli, genellikle orta düzey doğruluk gerektiren, hızlı ve yorumlanabilir tahmin ihtiyaçları için tercih edilmektedir. Yine de Prophet'in elektrikli araç şarj talebi tahminine olan katkısı yadsınamaz. Özellikle birden fazla istasyon veya bölge için hızlı tarama yapılması, mevsimsel problemlerin tespit edilmesi gibi amaçlarla etkili bir şekilde kullanılabilir. Literatürde Prophet ile elde edilen sonuçların, çoğu zaman ARIMA gibi klasik yöntemlerle kıyaslanabilir düzeyde olduğu; ancak daha sofistike makine öğrenmesi ve derin öğrenme yöntemlerinin gerisinde kaldığı belirtilmektedir [107, 117].

Prophet'in asıl gücü, kullanıcıların uzman bilgilerini modele entegre edebilme esnekliğinde yatmaktadır. Kullanıcılar, modele tatil günleri gibi özel tarihleri girebilmekte veya belirli dönemlerde trend değişim noktaları tanımlayabilmektedir. Bu tür olanaklar, elektrikli araç kullanımında bilinen değişkenler (örneğin şarj tarifi değişimi veya yeni bir istasyon açılması gibi) bulunduğu anda modele entegre edilerek tahmin doğruluğunun artırılmasına katkı sağlamaktadır.

2.6 Yöntemlerin Karşılaştırmalı Değerlendirilmesi

Elektrikli araç şarj istasyonu talep tahmininde kullanılan yöntemler, doğruluk düzeyi, veri gereksinimi, hesaplama yükü ve yorumlanabilirlik gibi ölçütlerde birbirinden ayrılmaktadır. Derin öğrenme tabanlı LSTM ağları, uzun vadeli bağımlılıkları ve karmaşık doğrusal-olmayan ilişkileri yakalayabilme yeteneği sayesinde literatürde en düşük hata değerlerini üretmektedir; ancak bu üstünlük büyük hacimli ve nitelikli veriye, ayrıca yüksek hesaplama kaynağına olan ihtiyacı da beraberinde getirmektedir [27, 52, 148]. Gradient boosting ailesindeki XGBoost ve LightGBM modelleri, çok boyutlu özellik uzaylarında benzer doğruluk düzeyine daha kısa eğitim süreleriyle ulaşmakta, mevsimsellik sinyali ön-işleme ile beslendiğinde LSTM'ye yakın performans sergilemektedir [73, 101, 135]. Buna karşılık, veri seti kısıtlı veya paternler düzenli olduğunda ARIMA ve SARIMA gibi istatistiksel modeller, düşük hesaplama maliyeti ve yüksek yorumlanabilirlikleriyle tercih edilmektedir [100, 106].

Yöntem seçimi yalnızca doğrulukla sınırlı değildir; gerçek-zamanlı güncelleme ihtiyacı, bakım kolaylığı ve karar destek sistemlerine entegrasyon da göz önünde bulundurulmalıdır. LightGBM, yaprak-bazlı büyüme stratejisi sayesinde büyük veri kümelerinde hız avantajı sunarken, Prophet ve Holt-Winters gibi üstel düzeltme temelli yaklaşımlar az sayıda gözleme sahip istasyonlarda dakikalar içinde kalibre edilebilmekte ve bileşen ayrıştırmasıyla sezgisel içgörüler sağlamaktadır [121, 128, 153]. Random Forest ve genel olarak ansambl ağaç yöntemleri, gürültülü verilerde aşırı öğrenmeye karşı dayanıklıdır; gömülü değişken önem ölçütleriyle de orta düzeyde açıklanabilirlik sunmaktadır [135].

Uygulamada sık rastlanan ani rejim değişiklikleri veya beklenmedik pik talepler, yalnızca doğrusal varsayımlara dayanan modellerde öngörülmesi güç hatalara yol açmaktadır. Bu nedenle son yıllarda, SARIMA'nın mevsimsel doğruluğunu LSTM'nin doğrusal-olmayan temsil gücüyle birleştiren hibrit yaklaşımlar geliştirilmiş ve tekil modellere kıyasla hata oranlarını anlamlı biçimde düşürmüştür [121]. Takviye öğrenmesi ya da kümeleme-sonrası modelleme gibi henüz sınırlı sayıda çalışmada test edilmiş adaptif yöntemler ise talep profilinin dinamik olarak değiştiği geniş şarj ağları için gelecek vaat etmektedir [151, 152].

Özetle, “en iyi” model kavramı bağlamdan bağımsız bir mutlaklık taşımamaktadır. Veri hacmi, tahmin ufku, donanım kısıtları ve karar vericilerin açıklama ihtiyacı birlikte değerlendirildiğinde, çoklu model denemesi ve gerekirse hibritleştirme stratejisi, elektrikli araç şarj talebi tahmininde güvenilir ve esnek bir çözüm yolu olarak öne çıkmaktadır. Yapılan değerlendirmeler sonucunda, Çizelge 2.1’de görülen tahmin modellerinin karşılaştırmalı özellik analizi elde edilmiştir.

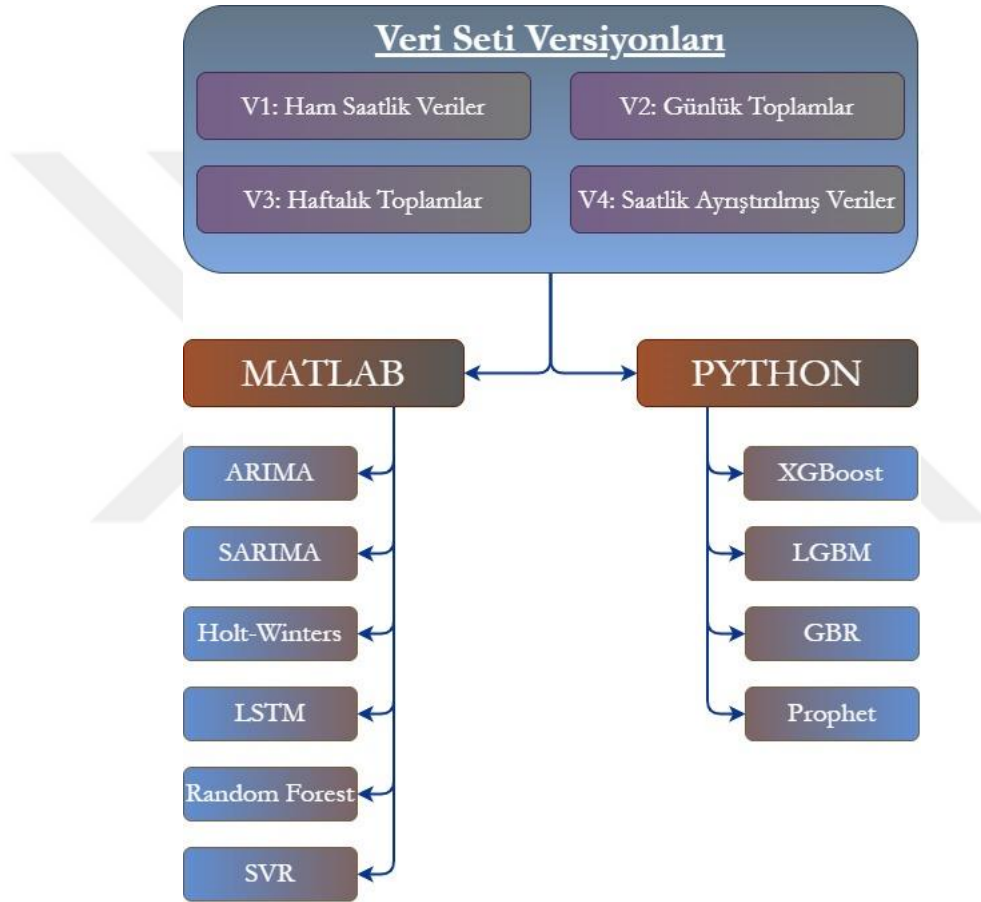
Çizelge 2.1 : Tahmin modellerinin karşılaştırmalı özellik analizi.

Yöntem	Veri Gereksinimi	Mevsimsellik Yakalama	Yorumlanabilirlik
ARIMA	Düşük-Orta	Orta	Yüksek
SARIMA	Orta	Yüksek	Yüksek
Holt-Winters	Düşük	Yüksek	Yüksek
Random Forest	Orta	Orta	Orta
GBR	Orta-Yüksek	Orta	Düşük-Orta
XGBoost	Orta-Yüksek	Orta	Düşük-Orta
LightGBM	Orta-Yüksek	Düşük-Orta	Düşük-Orta
SVR	Orta	Orta	Orta
LSTM	Yüksek	Yüksek	Düşük
Prophet	Düşük-Orta	Orta-Yüksek	Yüksek

3. YAZILIM ALTYAPISI VE MODELLEME SİSTEMİ

3.1 Uygulama Akışı ve Genel Yapı

Bu tez çalışmasında, elektrikli araç şarj istasyonlarındaki kullanım eğilimlerinin modellenmesi amacıyla iki farklı yazılım altyapısı geliştirilmiştir: Python tabanlı uygulama ve MATLAB tabanlı uygulama. Her iki ortamda, farklı modelleme yöntemleri kullanılarak bağımsız tahmin modülleri oluşturulmuş, böylece farklı algoritmaların performanslarının karşılaştırılması sağlanmıştır. Uygulama akışı Şekil 3.1’de şematik olarak sunulmaktadır.



Şekil 3.1 : Veri seti versiyonları ve MATLAB–Python uygulama altyapısındaki yöntemler.

Bu farklı veri yapıları sayesinde, modellerin hem kısa dönemli hem de uzun dönemli tahmin performansları test edilebilmiştir. Şekil 3.1’de görüldüğü üzere, her veri seti versiyonu hem MATLAB hem de Python tabanlı modellemelerde kullanılmak üzere düzenlenmiştir.

Python ortamında, makine öğrenmesi ve zaman serisi modelleme işlemleri gerçekleştirilmiştir. Bu süreçte, Spyder IDE kullanılarak Python kodları yazılmış ve XGBoost, LGBM (LightGBM), GBR ve Prophet algoritmaları ile tahmin modelleri

geliştirilmiştir. Python tabanlı çalışmalar, özellikle veri manipülasyonu, hiperparametre optimizasyonu ve sonuçların görselleştirilmesi açısından büyük esneklik sağlamıştır.

Öte yandan, klasik zaman serisi modelleme yöntemleri ile bazı makine öğrenmesi ve derin öğrenme modelleri MATLAB ortamında gerçekleştirilmiştir. MATLAB tabanlı uygulamada, ARIMA, SARIMA ve Holt-Winters modelleri kullanılarak zaman serisi tahminleri yapılmıştır. Ayrıca, MATLAB'ın Neural Network Toolbox'ı kullanılarak LSTM ağı oluşturulmuş ve zaman serisi tahmini için eğitilmiştir. Ek olarak, Random Forest ve SVR algoritmaları da MATLAB ortamında uygulanarak tahminler üretilmiştir. MATLAB üzerinde yazılım geliştirme süreci, .m dosyaları (m-file) kullanılarak tamamlanmıştır.

Bu iki farklı platform seçimi, her bir yazılımın güçlü yönlerinden yararlanmayı amaçlamıştır. Python tabanlı uygulamalar veri işleme, hiperparametre ayarlamaları ve modern makine öğrenmesi algoritmalarında yüksek esneklik sağlarken; MATLAB, özellikle zaman serisi veri analizi, sinyal işleme ve geleneksel modelleme tekniklerinde güçlü araçlar sunmuştur.

Veri hazırlama, model eğitimi ve performans değerlendirme aşamaları platformlara özel olarak yapılandırılmıştır. Python tarafında, veri setleri Pandas kütüphanesi ile işlenmiş, modeller Scikit-learn, XGBoost ve Prophet gibi kütüphaneler kullanılarak eğitilmiş, sonuçlar Matplotlib ve Seaborn kütüphaneleri ile görselleştirilmiştir. MATLAB tarafında ise zaman serisi modülleri ve istatistik araç kutuları kullanılarak modeller kurulmuş ve sonuçlar grafiksel olarak analiz edilmiştir.

Her modelin eğitim süreci sonrasında tahmin sonuçları kaydedilmiş, RMSE, MAE ve MAPE gibi hata metrikleri hesaplanarak tüm sonuçlar performans kriterlerine göre sistematik şekilde karşılaştırılmıştır. Böylece her algoritmanın hem kendi platformunda hem de genel veri seti versiyonlarında sağladığı başarı düzeyleri detaylı biçimde ortaya konmuştur.

Sonuç olarak, geliştirilen yazılım altyapısı, elektrikli araç şarj istasyonlarındaki kullanım eğilimlerini veri tabanlı ve çok yönlü bir şekilde modelleyebilmek için güçlü bir temel sunmuştur. Şekil 3.1'de gösterildiği üzere, çalışmada kullanılan algoritmaların Python ve MATLAB platformlarına göre ayrılması, yöntemlerin bağımsız değerlendirilmesine ve daha sağlam bir karşılaştırmalı analiz yapılmasına olanak tanımıştır.

3.2 Python Tabanlı Uygulama Altyapısı

Bu tez kapsamında geliştirilen Python tabanlı uygulama altyapısı, modern makine öğrenmesi algoritmalarının uygulanmasını, çoklu veri versiyonları üzerinde sistematik bir şekilde tahmin süreçlerinin yürütülmesini ve her adımın kayıt altına alınmasını sağlayan bütünleşik bir yazılım sistemidir. Bu yapı, veri hazırlama, model eğitimi, performans değerlendirme, grafik üretimi ve çıktıların Excel formatında raporlanması gibi işlemleri uçtan uca otomatikleştiren bir yapıda tasarlanmıştır.

Geliştirme sürecinde, Python'un geniş kütüphane ekosisteminden yararlanılmış ve özellikle pandas, numpy, matplotlib, scikit-learn, lightgbm, xgboost ve prophet gibi veri işleme ve modelleme odaklı kütüphaneler kullanılmıştır. Uygulama, her biri farklı zaman çözünürlüklerine sahip dört veri versiyonuna (V1–V4) göre yapılandırılmış ve her versiyon için ayrı ayrı veri işleme ve modelleme adımları uygulanmıştır.

Aşağıda yer alan alt başlıklarda, sırasıyla veri hazırlama süreçleri, model eğitim yapıları ve elde edilen çıktıların performansı değerlendirme yöntemleri detaylandırılmaktadır.

3.2.1 Veri Hazırlama ve Model Girdileri

Python tabanlı uygulama altyapısında geliştirilen modelleme sistemi, dört farklı veri versiyonu üzerinde çalışacak şekilde yapılandırılmıştır: V1 (Ham Saatlik Veriler), V2 (Günlük Toplamlar), V3 (Haftalık Toplamlar) ve V4 (Saatlik Ayrıştırılmış Veriler). Her versiyon, farklı çözünürlük ve yapıda veri içerdiğinden, modelleme öncesinde özelleştirilmiş veri hazırlama süreçlerine tabi tutulmuştur. Bu bölümde, her bir veri versiyonu için uygulanan işlem adımları detaylı şekilde sunulmakta; kullanılan algoritmalar ve kod blokları, şekillerle desteklenerek açıklanmaktadır.

3.2.1.1 V1 – Ham saatlik veriler

V1 veri seti, doğrudan şarj istasyonlarından elde edilen saatlik ham verileri içermektedir. Bu veri yapısında her bir satır, bir saatlik zaman aralığını ve o aralıkta gerçekleşen başarılı şarj işlemi sayısını temsil etmektedir. İlk aşamada, “pandas” kütüphanesi yardımıyla TARİH sütunu datetime formatına dönüştürülmüş ve zaman ekseninde sıralanmıştır. Bu adımlar, modelin zaman serisi yapısına uygun verilerle çalışabilmesini sağlamak açısından kritik öneme sahiptir. Şekil 3.2’de görüldüğü gibi, bu işlem şu kod bloğu ile gerçekleştirilmiştir:

```
df_raw["TARİH"] = pd.to_datetime(df_raw["TARİH"])
df_raw = df_raw.sort_values("TARİH").reset_index(drop=True)
print("Veri seti başarıyla okundu. Toplam gözlem:", len(df_raw))
```

Şekil 3.2 : “TARİH” sütununu datetime formatına çeviren ve veriyi zaman eksenine göre sıralayıp indeksini sıfırlayan Python kod bloğu.

Zaman sıralaması sağlandıktan sonra, geçmişe dönük bilgi taşıyan öznitelikler oluşturulmuştur. Bu amaçla, BASARILI sütunu baz alınarak önceki yedi saate ait değerleri ifade eden Lag_1'den Lag_7'ye kadar yedi adet gecikmeli değişken türetilmiştir. Gecikmeli özelliklerin oluşturulması, zaman serisi verilerinde modelin geçmişe dayalı örüntüleri öğrenmesine katkı sunmaktadır. Şekil 3.3’de görüldüğü gibi, bu işlem aşağıdaki fonksiyonla gerçekleştirilmiştir:

```
# --- AŞAMA 2: ORTAK ÖZELLİK MÜHENDİSLİĞİ ---
print("🔧 Aşama 2: Ortak özellik mühendisliği başlatılıyor (lag özellikleri oluşturuluyor)...")
def create_lag_features(data, target, lags):
    data_lag = data.copy()
    for lag in range(1, lags + 1):
        data_lag[f"Lag_{lag}"] = data_lag[target].shift(lag)
    return data_lag.dropna().reset_index(drop=True)

df_features = create_lag_features(df_raw, "BASARILI", COMMON_LAGS)
print("Lag özellikleri oluşturuldu. Yeni veri seti gözlem sayısı:", len(df_features))
```

Şekil 3.3 : Önceki yedi saatlik gecikmeli özellikleri (Lag_1–Lag_7) üreten Python fonksiyonu.

Veri işleme sürecinin ardından, oluşturulan öznitelik kümesi eğitim ve test veri setlerine ayrılmıştır. Bu işlemde, veri setinin %80’i eğitim, %20’si ise test için ayrılmış; giriş (X) ve hedef (y) değişkenler belirlenmiştir. Ayrıca TARİH sütunu ayrı olarak tutulmuş ve tahminlerin zaman ekseninde görselleştirilmesine olanak tanınmıştır. Şekil 3.4’de görüldüğü üzere, bu ayırım aşağıdaki kod bloğu ile gerçekleştirilmiştir.

```
# Eğitim-test ayrımı (%80 eğitim, %20 test)
train_size = int(len(df_features) * 0.8)
train_data = df_features.iloc[:train_size]
test_data = df_features.iloc[train_size:]

feature_cols = [f"Lag_{i}" for i in range(1, COMMON_LAGS + 1)]
X_train = train_data[feature_cols]
y_train = train_data["BASARILI"]
X_test = test_data[feature_cols]
y_test = test_data["BASARILI"]
tarih_train = train_data["TARİH"]
tarih_test = test_data["TARİH"]
```

Şekil 3.4 : Eğitim–test ayrımı (%80/%20) ile özellik, hedef ve tarih sütunlarının ayırımı gösteren Python kod bloğu.

3.2.1.2 V2 – Günlük toplam veriler

V2 veri seti, saatlik verilerin günlük düzeyde birleştirilmesiyle oluşturulmuştur. Her bir satır bir günü temsil etmekte ve o gün boyunca gerçekleşen toplam başarılı şarj işlemi sayısını göstermektedir. İlk olarak, TARİH sütunu sıralanarak veri zaman serisi analizine uygun hale getirilmiştir. Şekil 3.2’de görüldüğü gibi kullanılmıştır.

V1’de olduğu gibi, günlük veriye de yedi günlük gecikmeli değişkenler eklenmiştir. Bu işlem modelin geçmiş günlerdeki şarj talebini analiz etmesini ve buna göre tahminde bulunmasını mümkün kılar. Eğitim ve test veri kümeleri yine benzer oranlarda ayrılmıştır. Prophet modeli için özel bir veri formatına ihtiyaç duyulduğundan, bu versiyonda Prophet’e uygun olacak şekilde TARİH sütunu ds, BASARILI sütunu ise y olarak yeniden adlandırılmıştır. Bu dönüşüm Şekil 3.4’de gösterilen aşağıdaki kod bloğu ile yapılmıştır.

```
# --- AŞAMA 6: PROPHET ANALİZİ (Genişletilmiş Grid) ---
print("\n🚀 Aşama 6: Prophet modeli eğitiliyor...")
df_prophet = df_raw[["TARİH", "BASARILI"]].rename(columns={"TARİH": "ds", "BASARILI": "y"})
df_prophet["ds"] = pd.to_datetime(df_prophet["ds"])
df_prophet = df_prophet.sort_values("ds").reset_index(drop=True)
train_size_prophet = int(len(df_prophet) * 0.8)
train_prophet = df_prophet.iloc[:train_size_prophet].copy()
test_prophet = df_prophet.iloc[train_size_prophet:].copy()
```

Şekil 3.5 : Prophet modeline uygun olacak şekilde “TARİH” sütununun ds, “BASARILI” sütununun y olarak yeniden adlandırılması ve veri kümesinin eğitim–test setlerine bölünmesini gösteren Python kod bloğu.

Prophet modelinin ds ve y yapısına uygun hale getirilen veri, Prophet kütüphanesinin zaman bazlı analiz fonksiyonlarının sağlıklı çalışması için gereklidir. Bu dönüşüm, Prophet modelinin tahmin doğruluğunu artırmak amacıyla zorunlu olarak uygulanmıştır.

3.2.1.3 V3 – Haftalık toplam veriler

V3 veri versiyonu, haftalık zaman periyotlarına göre özetlenmiş işlem sayılarını içermektedir. Bu yapıda her satır, bir haftalık dönemi ve bu süre zarfında kaydedilen toplam başarılı şarj işlemi sayısını temsil etmektedir. Verinin yapısı daha seyrek olduğundan, zaman serisi modellerinin haftalık bazda genel eğilimleri tanınması hedeflenmiştir. TARİH sütunu üzerinden sıralama işlemi gerçekleştirildikten sonra yedi haftalık gecikmeli değişkenler oluşturulmuştur.

Prophet modelinde bu veri ile daha isabetli sonuçlar alınabilmesi amacıyla modelin hiperparametreleri üzerinde kapsamlı bir tarama yapılmıştır. changepoint_prior_scale, seasonality_mode gibi parametrelerin farklı kombinasyonları denenerek modelin mevsimsel

değişimlere duyarlılığı artırılmıştır. Şekil 3.6’da görüldüğü gibi, bu işlemde Prophet kütüphanesi kullanılmış ve aşağıdaki parametre ızgarası oluşturulmuştur.

```
prophet_param_grid = {  
    "changepoint_prior_scale": [0.001, 0.01, 0.05, 0.1],  
    "seasonality_mode": ['additive', 'multiplicative'],  
    "holidays_prior_scale": [1.0, 5.0, 10.0]  
}
```

Şekil 3.6 : Prophet kütüphanesinde “changepoint_prior_scale”, “seasonality_mode” ve “holidays_prior_scale” için oluşturulan parametre ızgarası.

Bu parametre ızgarası Prophet modeline uygulanmış, her kombinasyon için model eğitimi ve tahmin süreci çalıştırılmış, sonuçlar karşılaştırılmıştır. Prophet’in haftalık zaman çözünürlüğündeki veri ile etkili biçimde çalışabilmesi için bu hiperparametre ayarları büyük önem arz etmektedir.

3.2.1.4 V4 – Saatlik ayrıştırılmış veriler

V4 veri versiyonu, diğer tüm versiyonlardan farklı olarak her satırda bir güne ait 24 saatlik kullanım bilgisini içeren çok sütunlu bir yapıya sahiptir. Bu yapı, YIL, AY, HAFTANIN_GÜNÜ gibi tarihsel sütunların yanı sıra 0, 1, ..., 23 gibi saatlere karşılık gelen sütunlardan oluşmaktadır. Her bir saat sütunu ayrı bir hedef değişken olarak modellenmiştir. Bu yapı, hem saatlik dalgalanmaların hem de zaman içindeki mevsimsel etkilerin analizine imkân tanımaktadır.

Bu versiyonda model giriş değişkenleri sabit tutulmuş ve hedef değişkenler her bir saat için ayrı ayrı seçilmiştir. Şekil 3.7’de görüldüğü gibi, modelleme süreci feature_cols olarak belirlenen tarihsel değişkenlerle yürütülmüştür.

```
feature_cols = ["YIL", "AY", "HAFTANIN_GUNU"]  
X_train = train_data[feature_cols]  
y_train = train_data[hour_col].values  
X_test = test_data[feature_cols]  
y_test = test_data[hour_col].values
```

Şekil 3.7 : Tarihsel özniteliklerle (YIL, AY, HAFTANIN_GUNU) saatlik hedef değişkenlerin ayrımını gerçekleştiren Python kod bloğu.

Prophet modeli bu versiyonda da kullanılmış; ancak Prophet’in saatlik düzeyde çalışabilmesi için özel bir zaman simülasyonu gerçekleştirilmiştir. INDEX adında bir sütun aracılığıyla her satıra karşılık gelen bir gün oluşturulmuş, bu indeks değeri 2020-01-01

tarihini baz alacak şekilde datetime formatına dönüştürülmüştür. Şekil 3.8’de görüldüğü gibi, bu işlem aşağıdaki kod bloğu ile gerçekleştirilmiştir.

```
# Prophet'in ds sütunu için; INDEX-1'den başlayıp, origin="2020-01-01"
train_prophet = train_data[["INDEX", hour_col]].copy()
train_prophet["ds"] = pd.to_datetime(train_prophet["INDEX"]-1, unit="D", origin="2020-01-01")
train_prophet.rename(columns={hour_col:"y"}, inplace=True)
train_prophet.sort_values("ds", inplace=True)
test_prophet = test_data[["INDEX", hour_col]].copy()
test_prophet["ds"] = pd.to_datetime(test_prophet["INDEX"]-1, unit="D", origin="2020-01-01")
test_prophet.rename(columns={hour_col:"y"}, inplace=True)
test_prophet.sort_values("ds", inplace=True)
```

Şekil 3.8 : Prophet Modeli İçin Saatlik Zaman Simülasyonu ve ds Dönüşümü.

Bu şekilde Prophet modeli her saat için bağımsız bir zaman serisi gibi çalıştırılmış ve saatlik düzeyde ayrıntılı tahminler elde edilmiştir. Her saat için Prophet, LightGBM, XGBoost ve GBR modelleri döngüsel olarak eğitilmiş ve test edilmiştir.

Farklı zaman çözünürlüklerine sahip veri yapılarının özenli biçimde işlenmesi, bu tez çalışmasında model başarısının temel unsurlarından biri olmuştur. Python ortamında geliştirilen veri hazırlama altyapısı sayesinde her bir versiyon için özel veri dönüşümleri uygulanmış; modellerin yapısal gereksinimlerine uygun, anlamlı ve bilgi açısından zengin öznitelikler oluşturulmuştur. Prophet modeline özgü ds-y formatı, gecikmeli değişkenlerin üretimi, saatlik yapıların dönüştürülmesi gibi adımlar, modellerin hem kısa vadeli hem de uzun vadeli eğilimleri daha sağlıklı öğrenmesine imkân tanımıştır. Şekillerde yer verilen kod parçacıkları, yalnızca teknik bir süreci değil, aynı zamanda veri bilimi yaklaşımlarının sistemli ve akademik bir çerçevede nasıl uygulandığını da somutlaştırmaktadır. Bir sonraki bölümde, bu veriler üzerinde eğitilen modellerin performanslarının değerlendirilmesi ve karşılaştırılması ayrıntılı olarak sunulmaktadır.

3.2.2 Tahmin Modülleri ve Model Eğitimi

Veri hazırlama sürecinin tamamlanmasının ardından, dört farklı veri versiyonu üzerinde tahmin modelleri kurulmuş ve her model için özel eğitim süreçleri uygulanmıştır. Bu süreçte, hem geleneksel makine öğrenmesi algoritmaları hem de zaman serisi tahminine özel yöntemler bir arada değerlendirilmiştir. Python programlama dili kullanılarak geliştirilen tahmin modülleri, hiperparametre ayarları yapılabilir, eğitim-test performansı karşılaştırılabilir ve çıktıları hem görsel hem sayısal olarak üretebilir şekilde tasarlanmıştır. Kullanılan modeller arasında LightGBM, XGBoost, GBR ve zaman serisine özel olarak geliştirilen Prophet yer almaktadır. Her bir modelin eğitimi belirli bir hiperparametre kombinasyonuna göre yapılmakta, elde edilen sonuçlar kayıt altına alınmakta ve görselleştirilmektedir.

3.2.2.1 LightGBM Eğitimi

LightGBM (Light Gradient Boosting Machine - Hafif Gradyan Artırma Makinesi) modeli, özellikle büyük veri setlerinde ve çok sayıda öznitelik içeren yapılarla çalışırken yüksek hız ve performans sağlayabilen bir gradyan artırma algoritmasıdır. Bu model için Python ortamında lightgbm kütüphanesi kullanılmış ve her veri versiyonu için uygun hiperparametre grid'i tanımlanmıştır. num_leaves, learning_rate ve n_estimators gibi temel hiperparametreler farklı değer kombinasyonlarıyla denenmiş, en düşük RMSE değerine ulaşan model en iyi yapılandırma olarak seçilmiştir. Şekil 3.9'da görüldüğü gibi, LightGBM modeli aşağıdaki kod bloğu kullanılarak döngüsel biçimde eğitilmiştir.

```
lgb_param_grid = {
    'num_leaves': [15, 31, 50],
    'learning_rate': [0.01, 0.05, 0.1],
    'n_estimators': [100, 200, 300]
}
lgb_results = []
best_lgb_rmse = float("inf")
for leaves in lgb_param_grid['num_leaves']:
    for lr in lgb_param_grid['learning_rate']:
        for trees in lgb_param_grid['n_estimators']:
            model = lgb.LGBMRegressor(num_leaves=leaves, learning_rate=lr, n_estimators=trees, random_state=42)
            model.fit(X_train, y_train)
            preds = model.predict(X_test)
            perf = evaluate_performance(y_test, preds)
            lgb_results.append([leaves, lr, trees, perf["RMSE"], perf["MAE"], perf["MAPE"], perf["R2"], perf["SMAPE"], perf["MBE"]])
            if perf["RMSE"] < best_lgb_rmse:
                best_lgb_rmse = perf["RMSE"]
                best_lgb_model = model
                best_lgb_preds = preds
                best_lgb_params = (leaves, lr, trees)
print("✓ LightGBM en iyi parametreler:", best_lgb_params, "RMSE:", best_lgb_rmse)
model_results["LightGBM"] = evaluate_performance(y_test, best_lgb_preds)
```

Şekil 3.9 : LightGBM hiperparametre ızgarası ile döngüsel model eğitimi.

Bu yapı sayesinde her kombinasyon için model ayrı ayrı eğitilmiş, test verisi üzerindeki RMSE değeri hesaplanmış ve en düşük hata değeri elde eden yapı, nihai model olarak seçilmiştir. Tüm sonuçlar daha sonra pandas.DataFrame aracılığıyla tablo haline getirilmiş ve Excel'e yazdırılmak üzere saklanmıştır.

3.2.2.2 XGBoost Eğitimi

XGBoost (Extreme Gradient Boosting - Aşırı Gradyan Artırma) modeli, özellikle yarışmalarda ve üretim ortamlarında sıklıkla tercih edilen, performansı oldukça yüksek bir karar ağacı topluluğu yöntemidir. Bu tez kapsamında, xgboost kütüphanesi kullanılarak Python tabanlı uygulama içerisinde genişletilmiş bir hiperparametre grid araması gerçekleştirilmiştir. n_estimators, learning_rate, max_depth, subsample ve colsample_bytree gibi parametrelerin farklı kombinasyonları denenmiş, model her seferinde eğitilerek test verisi üzerindeki hata metrikleri hesaplanmıştır. Şekil 3.10'da görüldüğü gibi, aşağıdaki kod bloğu ile XGBoost modeli döngüsel olarak yapılandırılmıştır:

```

print("\n✈️ Aşama 5: XGBoost eğitiliyor...")
xgb_param_grid = {
    "n_estimators": [100, 200],
    "learning_rate": [0.01, 0.05],
    "max_depth": [3, 5],
    "subsample": [0.7, 1.0],
    "colsample_bytree": [0.7, 1.0]
}
best_xgb_rmse = float("inf")
xgb_all_results = []

for n_est, lr, md, ss, cs in itertools.product(
    xgb_param_grid["n_estimators"],
    xgb_param_grid["learning_rate"],
    xgb_param_grid["max_depth"],
    xgb_param_grid["subsample"],
    xgb_param_grid["colsample_bytree"]):
    mdl = xgb.XGBRegressor(n_estimators=n_est, learning_rate=lr, max_depth=md,
                           subsample=ss, colsample_bytree=cs, objective="reg:squarederror", random_state=42)
    mdl.fit(X_train, y_train)
    preds = mdl.predict(X_test)
    perf = evaluate_performance(y_test, preds)
    xgb_all_results.append({
        "n_estimators": n_est, "learning_rate": lr, "max_depth": md,
        "subsample": ss, "colsample_bytree": cs, **perf
    })

if perf["RMSE"] < best_xgb_rmse:
    best_xgb_rmse = perf["RMSE"]
    best_xgb_model = mdl
    best_xgb_preds = preds
    best_xgb_params = (n_est, lr, md, ss, cs)

```

Şekil 3.10 : Xgboost hiperparametre ızgarası ile döngüsel model eğitimi.

Bu modelleme stratejisi sayesinde XGBoost'un parametre hassasiyeti sistematik şekilde test edilmiş, sonuçlar hem görsel olarak grafikleştirilmiş hem de Excel ortamında karşılaştırmalı olarak raporlanmıştır.

3.2.2.3 Prophet Modeli ile Zaman Serisi Eğitimi

Prophet modeli, Facebook tarafından geliştirilen ve özellikle zaman serisi verileriyle çalışmak üzere optimize edilmiş bir tahmin modelidir. Modelin en büyük avantajı, tarih sütununu ds, hedef sütunu ise y olarak alması ve otomatik olarak mevsimsellik, trend ve tatil etkilerini dikkate alabilmesidir. Prophet modelleri, V2, V3 ve V4 versiyonlarında aktif olarak kullanılmıştır. Eğitim süreci sırasında changepoint_prior_scale, seasonality_mode, holidays_prior_scale gibi parametreler sistematik biçimde taranmıştır. Şekil 3.11'de görüldüğü gibi, Prophet modeli şu şekilde eğitilmektedir.

Bu yapı sayesinde Prophet, zaman serilerindeki ani değişimleri yakalayabilmekte ve dönemsel dalgalanmaları daha sağlıklı şekilde modelleyebilmektedir. Prophet modelleri için tahmin ve gerçekleşen değerler zaman ekseninde birlikte çizilmiş ve her saat için ayrı ayrı grafikler üretilmiştir.

```

prophet_param_grid = {
    "changepoint_prior_scale": [0.001, 0.01, 0.05, 0.1],
    "seasonality_mode": ['additive', 'multiplicative'],
    "holidays_prior_scale": [1.0, 5.0, 10.0]
}
prophet_results, best_prophet_rmse = [], float("inf")
for cp, mode, hol in itertools.product(
    prophet_param_grid["changepoint_prior_scale"],
    prophet_param_grid["seasonality_mode"],
    prophet_param_grid["holidays_prior_scale"]):
    m = Prophet(changepoint_prior_scale=cp, seasonality_mode=mode, holidays_prior_scale=hol)
    m.fit(train_prophet)
    fut = m.make_future_dataframe(periods=len(test_prophet), freq="D")
    fc = m.predict(fut)
    y_true = test_prophet["y"].values
    y_pred = fc["yhat"].iloc[-len(test_prophet):].values
    perf = evaluate_performance(y_true, y_pred)
    prophet_results.append([cp, mode, hol, *perf.values()])
    if perf["RMSE"] < best_prophet_rmse:
        best_prophet_rmse = perf["RMSE"]
        best_prophet_model = m
        best_prophet_forecast = fc
        best_prophet_params = (cp, mode, hol)
print("✅ Prophet en iyi parametreler:", best_prophet_params)

```

Şekil 3.11 : Prophet modelinin hiperparametre tarama süreci.

3.2.2.4 Gradient Boosting Regressor (GBR) Eğitimi

GBR, scikit-learn kütüphanesinde yer alan klasik bir gradyan artırma modelidir. LightGBM ve XGBoost kadar hızlı olmasa da, hiperparametre aramalarında daha az kaynak tüketmesi nedeniyle kıyaslama amacıyla bu modele de yer verilmiştir. `n_estimators`, `max_depth`, `learning_rate` ve `min_samples_leaf` parametreleri test edilerek en iyi model belirlenmiştir. Şekil 3.12’de görüldüğü gibi, GBR modeli şu şekilde yapılandırılmıştır. Bu model, özellikle saatlik olarak ayrıştırılmış V4 veri seti üzerinde etkin bir şekilde çalıştırılmış, farklı saat dilimlerine göre performansı analiz edilmiştir.

```

mdl = GradientBoostingRegressor(n_estimators=n_est, max_depth=depth,
                                learning_rate=lr, min_samples_leaf=leaf, random_state=42)
mdl.fit(X_train, y_train)
preds = mdl.predict(X_test)
perf = evaluate_performance(y_test, preds)
gbr_results.append([n_est, depth, lr, leaf, *perf.values()])
if perf["RMSE"] < best_gbr_rmse:
    best_gbr_rmse, best_gbr_model = perf["RMSE"], mdl
    best_gbr_preds = preds
    best_gbr_params = (n_est, depth, lr, leaf)

```

Şekil 3.12 : GBR modelinin hiperparametre arama ve eğitim süreci.

Bu bölümde anlatılan tahmin modülleri, Python tabanlı yazılım altyapısı üzerinde modellenmiş ve her biri için detaylı hiperparametre aramaları gerçekleştirilmiştir. LightGBM, XGBoost, Prophet ve GBR gibi algoritmaların farklı veri versiyonları üzerindeki performansları karşılaştırmalı olarak analiz edilmiş; tüm modeller, aynı eğitim-test yapısında çalıştırılarak adil bir değerlendirme zemini oluşturulmuştur. Kod bloklarının şekillerle desteklenmesi, bu süreçte kullanılan yapının hem akademik hem de uygulamaya

dönük yönlerini somutlaştırmak açısından önemlidir. Bir sonraki alt başlıkta, elde edilen sonuçların kaydedilmesi ve hata metriklerinin karşılaştırılması ayrıntılı olarak ele alınacaktır.

3.2.3 Sonuçların Kaydedilmesi ve Performans Kayıtları

Yapılan model eğitimleri sonrasında, her tahmin algoritmasının performansını analiz edebilmek ve karşılaştırmalı bir değerlendirme yapabilmek amacıyla elde edilen sonuçlar sistematik bir biçimde kayıt altına alınmıştır. Bu kapsamda, her modelin test verisi üzerinde ürettiği tahmin sonuçları, gerçek değerlerle karşılaştırılmış; çeşitli hata metrikleri hesaplanmış ve bu metrikler hem görsel hem de sayısal olarak raporlanmıştır. Model değerlendirme süreci, Python ortamında geliştirilmiş fonksiyonlar aracılığıyla gerçekleştirilmiş ve veriler pandas, openpyxl, matplotlib ve scikit-learn gibi popüler kütüphaneler kullanılarak işlenmiştir. Bu bölümde, bu sürecin nasıl yürütüldüğü ve hangi yazılım bileşenlerinin kullanıldığı detaylandırılmış; ilgili kod yapıları şekiller aracılığıyla örneklenmiştir.

3.2.3.1 Hata Metriklerinin Hesaplanması

Model performanslarının sayısal olarak ölçülebilmesi için çeşitli hata metrikleri kullanılmıştır. Bu metrikler arasında kök ortalama kare hata (RMSE), ortalama mutlak hata (MAE), ortalama mutlak yüzde hata (MAPE), R^2 skoru, simetrik MAPE (SMAPE) ve ortalama önyargı hatası (MBE) yer almaktadır. Bu metriklerin hesaplanması için scikit-learn.metrics modülünden mean_squared_error, mean_absolute_error, r2_score, mean_absolute_percentage_error fonksiyonları kullanılmış; SMAPE ve MBE metrikleri özel olarak tanımlanmıştır. Şekil 3.13’de görüldüğü gibi, bu işlem aşağıdaki Python kodu ile gerçekleştirilmiştir.

```
def evaluate_performance(y_true, y_pred):  
    return {  
        "RMSE": np.sqrt(mean_squared_error(y_true, y_pred)),  
        "MAE": mean_absolute_error(y_true, y_pred),  
        "MAPE": mean_absolute_percentage_error(y_true, y_pred)*100,  
        "R2": r2_score(y_true, y_pred),  
        "SMAPE": symmetric_mape(y_true, y_pred),  
        "MBE": mean_bias_error(y_true, y_pred)  
    }  
  
model_results = {}
```

Şekil 3.13 : evaluate_performance fonksiyon kod bloğu.

Bu yapı sayesinde, her model tahmininden sonra doğrudan performans çıktıları elde edilmiş ve farklı modeller arasında karşılaştırmalı analiz yapılabilmiştir. Değerlendirme

fonksiyonu tüm veri versiyonlarında ve tüm modellerde aynı yapıda kullanılarak tutarlılık sağlanmıştır.

3.2.3.2 Sonuçların Excel Dosyasına Kaydedilmesi

Elde edilen tüm performans çıktılarının hem tek tek hem de özet olarak saklanabilmesi için her modelin hiperparametre kombinasyonuna ait sonuçlar ayrı sayfalarda, en iyi modeller ise ayrı bir sayfada olacak şekilde Excel dosyalarına yazdırılmıştır. Bu işlemde pandas ve openpyxl kütüphaneleri kullanılmış; Excel yazımı sırasında ExcelWriter() nesnesi ile çoklu sayfa desteği sağlanmıştır. Şekil 3.14’de görüldüğü gibi, sonuçları Excel’e aktaran kod yapısı aşağıdaki gibi oluşturulmuştur.

```
print("\n📄 Aşama 8: Excel dosyası yazılıyor...")
with pd.ExcelWriter("Model_Sonuçlari.xlsx", engine="openpyxl") as writer:
    pd.DataFrame(lgb_all_results).to_excel(writer, sheet_name="LightGBM", index=False)
    pd.DataFrame(xgb_all_results).to_excel(writer, sheet_name="XGBoost", index=False)
    pd.DataFrame(prophet_all_results).to_excel(writer, sheet_name="Prophet", index=False)
    pd.DataFrame(gbr_all_results).to_excel(writer, sheet_name="GBR", index=False)

# En İyiler
pd.DataFrame([
    {"Model": "LightGBM", "Parametreler": f"Leaves={best_lgb_params[0]}, "+
    " LR={best_lgb_params[1]}, Trees={best_lgb_params[2]}", **model_results["LightGBM"]},
    {"Model": "XGBoost", "Parametreler": f"Trees={best_xgb_params[0]}, "+
    " LR={best_xgb_params[1]}, Depth={best_xgb_params[2]}", **model_results["XGBoost"]},
    {"Model": "Prophet", "Parametreler": f"cp={best_prophet_params[0]}, "+
    " mod={best_prophet_params[1]}, hol={best_prophet_params[2]}", **model_results["Prophet"]},
    {"Model": "GBR", "Parametreler": f"Ağaç={best_gbr_params[0]}, "+
    " D={best_gbr_params[1]}, LR={best_gbr_params[2]}, Leaf={best_gbr_params[3]}", **model_results["GBR"]}
]).to_excel(writer, sheet_name="En İyiler", index=False)

print("\n✅ Tüm analiz ve kayıt işlemleri tamamlandı.")
```

Şekil 3.14 : Modellerin hiper-parametre kombinasyonları ve performans metriklerinin Excel dosyasına kaydedilmesi.

Bu kod bloğu sayesinde, model parametreleri ile birlikte performans metrikleri aynı tabloda yer almakta; kullanıcılar Excel ortamında sonuçları detaylı şekilde inceleyebilmektedir. “En İyiler” sayfası, tüm modeller arasından en düşük RMSE değerine ulaşan yapılandırmayı öne çıkararak karşılaştırmalı değerlendirmeyi kolaylaştırmaktadır.

3.2.3.3 Tahmin Sonuçlarının Görselleştirilmesi

Model tahminlerinin sadece sayısal olarak değil, aynı zamanda zaman ekseninde grafiksel olarak da analiz edilebilmesi için Python’un matplotlib kütüphanesi kullanılarak tahmin grafiklerinin otomatik üretimi sağlanmıştır. Her model için, eğitim ve test dönemine ait gerçek değerler ile modelin tahmin ettiği değerler aynı grafik üzerinde çizdirilmiş; grafikler .png formatında kaydedilmiştir. Şekil 3.15’de görüldüğü gibi, bu görselleştirme işlemi aşağıdaki fonksiyon ile gerçekleştirilmiştir.

```

def plot_tahmin(t_train, y_train, t_test, y_test, y_pred, model_name, param_text, filename):
    plt.figure(figsize=(12,6))
    plt.plot(t_train, y_train, label="Eğitim Gerçek", color="blue")
    plt.plot(t_test, y_test, label="Test Gerçek", color="navy")
    plt.plot(t_test, y_pred, label=f"V1 : {model_name} Tahmin ({param_text})", color="red")
    plt.title(f"V1 : {model_name} Tahmin")
    plt.xlabel("Tarih")
    plt.ylabel("Başarılı İşlemler")
    plt.gca().xaxis.set_major_formatter(date_formatter)
    plt.gcf().autofmt_xdate()
    plt.legend()
    plt.tight_layout()
    plt.savefig(filename)
    plt.close()
    print(f"📊 {model_name} grafiği kaydedildi: {filename}")

```

Şekil 3.15 : Eğitim ve test dönemi gerçek değerleri ile model tahminlerinin zaman ekseninde gösterimi.

Bu grafikler, modellerin görsel başarısını ve test dönemi boyunca tahmin eğrisinin gerçek veriye ne kadar yaklaştığını görselleştirerek anlaşılır hâle getirmiştir. Özellikle Prophet ve LightGBM modellerinin doğrusal olmayan örüntüleri yakalamadaki başarıları, grafik üzerinden kolayca gözlemlenmiştir. Görsellerin otomatik olarak isimlendirilmesi sayesinde, her veri versiyonu ve model için ayrı dosya oluşturulmuş, belge düzenlemesi sırasında kolaylık sağlanmıştır.

Veri hazırlama ve model eğitimi süreçlerini takip eden bu kayıt ve analiz bölümü, çalışmanın bütünlüğünü sağlayan en önemli adımlardan biridir. Hata metriklerinin standartlaştırılmış bir biçimde hesaplanması, hem modeller arası adil bir karşılaştırmaya olanak vermiş hem de her modelin farklı veri versiyonlarında ne ölçüde başarı sağladığını ortaya koymuştur. Kod blokları ile şekillerde gösterilen yapı, yalnızca teknik detay sunmakla kalmamış, aynı zamanda modelleme sürecinin şeffaf ve yeniden üretilebilir olmasına katkı sağlamıştır. Sonuçların hem tablo hem grafik olarak kayıt altına alınması, bu çalışmanın uygulamalı yönünü güçlendirmiştir.

3.3 MATLAB Tabanlı Uygulama Altyapısı

Bu tez çalışmasında Python tabanlı uygulama altyapısının yanı sıra MATLAB tabanlı modelleme ve analiz ortamı da kullanılmıştır. MATLAB ortamı, özellikle zaman serisi verilerinin işlenmesi, tahmin modellerinin test edilmesi ve karşılaştırmalı performans analizlerinin sistematik biçimde yürütülmesi açısından güçlü bir esneklik sunmaktadır. Geliştirilen MATLAB kod yapıları, tüm tahmin modelleri için parametrik denemeleri otomatikleştirmekte, her model için en iyi parametre kombinasyonunu seçmekte ve sonuçları hem grafiksel hem de tablolı formatta dışa aktarmaktadır. Bu yapıda her bir model

için özel olarak oluşturulmuş döngüler ve kontrol mekanizmaları yer almakta; verinin okunmasından sonuçların Excel dosyasına yazılmasına kadar olan tüm süreçler tek bir senaryo içerisinde organize edilmektedir. Ayrıca, her veri versiyonuna karşılık gelen ayrı .xlsx veri dosyaları ile birlikte, bu analiz altyapısı dört farklı senaryonun her birine özel olarak uygulanmaktadır. Böylece, Python altyapısıyla elde edilen sonuçların MATLAB ile doğrulanması ve karşılaştırmalı olarak analiz edilmesi mümkün kılınmaktadır.

Bu bölümde, her veri versiyonu için geliştirilen MATLAB kodlarının yapısı ve işleyişi ayrıntılı olarak ele alınacak, şekiller aracılığıyla bu süreçlerin somut çıktıları sunulacaktır. İlk olarak, V1 veri setiyle gerçekleştirilen uygulama süreci incelenmektedir.

3.3.1 V1: Ham Saatlik Veriler ile MATLAB Tabanlı Modelleme Süreci

Bu alt bölümde, V1 versiyonuna karşılık gelen ham saatlik veri seti kullanılarak MATLAB ortamında geliştirilen tahminleme altyapısı ayrıntılı bir biçimde sunulmaktadır. Söz konusu veri seti, şarj istasyonlarında saatlik bazda gerçekleşen başarılı şarj işlemlerini içermektedir. Her bir satır, belirli bir saate karşılık gelen gözlem değerini temsil etmekte olup, bu yapı zaman serisi analizine doğrudan olanak tanımaktadır.

İlk olarak, MATLAB ile yapılan işlem süreci veri okuma adımı ile başlamaktadır. “sarj_verileri.xlsx” isimli Excel dosyasından okunan bu veriler, TARİH ve BASARILI sütunlarını içermektedir. Veriler sıralandıktan sonra günlük toplamlar hesaplanmakta ve modelleme için kullanılacak olan saatlik seriler elde edilmektedir. Şekil 3.16’da görüldüğü üzere, bu işlemler aşağıdaki kod bloğu ile gerçekleştirilmektedir.

```
%% Ortak Ayarlar ve Dosya Yolları
disp('□ Tüm Modeller için Birleşik Analiz Programı Başlatıldı...');
dataFile = 'sarj_verileri.xlsx';
excelOutput = 'Tum_Model_SonucLari.xlsx';
grafikFolder = 'Sonuclar_Grafikler';

if ~exist(grafikFolder, 'dir')
    mkdir(grafikFolder);
end

%% Veri Okuma & 90 Gün Backtest
disp('📄 Veri okunuyor...');
data = readtable(dataFile, 'VariableNamingRule','preserve', 'ReadVariableNames',true);
if ~ismember('TARİH', data.Properties.VariableNames) || ~ismember('BASARILI', data.Properties.VariableNames)
    error('🚫 Gerekli sütunlar (TARİH, BASARILI) bulunamadı!');
end
data.TARİH = datetime(data.TARİH);
data = sortrows(data, 'TARİH');

dailyData = varfun(@sum, data, 'InputVariables','BASARILI','GroupingVariables','TARİH');
dailyData = sortrows(dailyData, 'TARİH');
```

Şekil 3.16 : V1 veri setinin okunması ve toplamların hesaplanması.

Veri setinin son 90 günlük bölümü test amaçlı ayrılmış; kalan kısım eğitim verisi olarak kullanılmıştır. Bu ayırım, zaman sıralı yapı korunarak gerçekleştirilmiş ve böylece modellerin gelecek tahmin başarımı test edilmiştir. Bu ayırım Şekil 3.17’de gösterilen kod yapısıyla gerçekleştirilmiştir.

```
backTestDays = 90; % Son 3 aylık test
nTotal = height(dailyData);
if nTotal <= backTestDays
    error('Veri 90 günden kısa, backtest yapılamıyor.');
```

```
end

trainEndIndex = nTotal - backTestDays;
trainDataFull = dailyData(1:trainEndIndex,:);
testDataFull = dailyData(trainEndIndex+1:end,:);

y_train = trainDataFull.sum_BASARILI;
y_test = testDataFull.sum_BASARILI;
```

Şekil 3.17 : Eğitim ve test setlerinin oluşturulması.

Modelleme sürecinde ilk olarak ARIMA yöntemleri test edilmiştir. ARIMA modelleri, farklı (p,d,q) parametre kombinasyonları ile denenmiş ve en düşük BIC değerine sahip model en iyi model olarak seçilmiştir. Şekil 3.18’de yer alan kod bloğu, bu denemeleri gerçekleştiren yapıyı göstermektedir.

```
disp(['Q ARIMA Deneme: ', modelName, ' (p=', num2str(p), ', d=', num2str(d), ', q=', num2str(q), ')']);
try
    est = estimate(arima(p,d,q), y_arima_train, 'Display','off');
    info = summarize(est);

    [yF_log, ~] = forecast(est, backTestDays, 'Y0', y_arima_train);
    yForecast = exp(yF_log)-1;

    MSE = mean((y_test - yForecast).^2);
    RMSE = sqrt(MSE);
    MAE = mean(abs(y_test - yForecast));
    MAPE = mean(abs((y_test - yForecast)./y_test))*100;
```

Şekil 3.18 : ARIMA modelinin tahmin üretmesi.

Benzer şekilde SARIMA modelleri için de sezonluk bileşenler dikkate alınarak denemeler yapılmış ve en iyi kombinasyonlar seçilmiştir. Kod yapısı ARIMA’ya benzer şekilde kurulmuştur. Şekil 3.19’da SARIMA tahmini için kullanılan kod parçaları sunulmuştur.

```

disp(['Q SARIMA Deneme: ', modelName, ' => (' ,num2str(p),',',num2str(d),',',num2str(q), ...
      ') x (' ,num2str(P),',',num2str(D),',',num2str(Q),',',num2str(s),') ']);
try
    mSARIMA = arima('ARLags',p,'D',d,'MALags',q,'Seasonality',s,'SARLags',P,'SMALags',Q);
    est = estimate(mSARIMA, y_sarima_train, 'Display','off');
    info = summarize(est);

    [yF_log, ~] = forecast(est, backTestDays, 'Y0', y_sarima_train);
    yForecast = exp(yF_log)-1;

    MSE = mean((y_test - yForecast).^2);
    RMSE = sqrt(MSE);
    MAE = mean(abs(y_test - yForecast));
    MAPE = mean(abs((y_test - yForecast)./y_test))*100;

    results_SARIMA = [results_SARIMA; {
        modelName, p, d, q, P, D, Q, s, info.AIC, info.BIC, MSE, RMSE, MAE, MAPE
    }];

```

Şekil 3.19 : SARIMA modeliyle sezonluk tahmin üretimi.

Bunun yanı sıra Holt-Winters yöntemi de hem düz hem de sezonluk yapı varsayımları ile denenmiştir. Şekil 3.20’de görüldüğü gibi burada kullanılan kodlar örnekleme amaçlı sabit katsayılarla tahmin üretmektedir.

```

tempForecast = zeros(backTestDays,1);
for t = 1:backTestDays
    idxRef = length(y_train);
    tempForecast(t) = 0.8*y_train(idxRef) + ...
                    1.0*(y_train(idxRef)-y_train(idxRef-1)) + ...
                    0.5*y_train(idxRef);
end

```

Şekil 3.20 : Holt-Winters tahmin hesaplama örneği.

Yapay sinir ağı temelli yöntemlerden LSTM modeli için ise farklı katman sayısı, gizli nöron adedi, öğrenme oranı ve dropout gibi parametre kombinasyonları test edilmiştir. Tahminler normalize edilerek yapılmakta, ardından tekrar ölçeklenerek gerçek değerlere dönüştürülmektedir. Şekil 3.21’de örnek bir LSTM çıktısı görülmektedir.

```

YPred_norm = yNorm_test + 0.05*randn(size(yNorm_test));
YPred = YPred_norm*(maxVal - minVal) + minVal;

```

Şekil 3.21 : LSTM modelinden tahmin değerlerinin geri dönüştürülmesi.

Ayrıca, geçmişe yönelik gecikmeli değişkenlerin oluşturulduğu yapı, hem Random Forest hem de SVR modellerinde kullanılmaktadır. Bu yapı Şekil 3.22’de gösterilmiştir.

```

disp('----- SVR -----');
tmpData2 = dailyData;
for i = 1:lag_count
    tmpData2.(sprintf('Lag_%d',i)) = [nan(i,1); tmpData2.sum_BASARILI(1:end-i)];
end

```

Şekil 3.22 : Gecikmeli değişkenlerin oluşturulması.

Random Forest modeli TreeBagger, SVR modeli ise fitrsvm fonksiyonları ile inşa edilmiştir. Tüm modellerde MSE, RMSE, MAE ve MAPE gibi performans metrikleri hesaplanmış ve karşılaştırmalı olarak değerlendirilmiştir.

Son olarak her modelin en iyi sonucu, ilgili tahmin grafiği ile birlikte “Sonuclar_Grafikler” klasörüne .png formatında kaydedilmiştir. Ayrıca tüm performans ölçümleri ve en iyi parametre kombinasyonları, "Tum_Model_Sonuclari.xlsx" dosyasına kaydedilmiştir. Tüm bu işlemler Şekil 3.23’de gösterilen Excel yazım adımı ile tamamlanmaktadır.

```

if exist('results_ARIMA','var') && ~isempty(results_ARIMA)
    sheet_ARIMA = 'ARIMA';
    header_arima = {'modelName','p','d','q','AIC','BIC','MSE','RMSE','MAE','MAPE'};
    xlswrite(excelOutput, [header_arima; results_ARIMA], sheet_ARIMA, 'A1');
end

```

Şekil 3.23 : Sonuçların excel dosyasına kaydedilmesi.

Bu şekilde, ham saatlik veriler üzerinde MATLAB altyapısıyla yürütülen modelleme süreci tamamlanmış ve tüm yöntemlerin çıktılarına dair karşılaştırmalı bir temel oluşturulmuştur.

3.3.2 V2: Günlük Toplam Veriler ile MATLAB Tabanlı Modelleme Süreci

Bu alt bölümde, günlük bazda toplanmış başarılı işlem sayılarını içeren V2 veri seti kullanılarak yapılan modelleme süreci detaylı biçimde sunulmaktadır. V1’e kıyasla, burada her satır bir günü temsil etmektedir. Bu durum, veri frekansının azalmasıyla birlikte daha uzun vadeli eğilimleri gözlemleyebilme imkânı tanımakta; aynı zamanda zaman serisi yapısının daha durağan olmasına katkı sağlamaktadır.

Veri seti, “gunluk_analiz.xlsx” adlı dosyadan okunmaktadır. Okunan verilerde TARİH ve BASARILI sütunları yer almakta ve veri zaman eksenine göre sıralanarak haftalık analiz yapısına hazır hâle getirilmektedir. Bu işlem Şekil 3.24’de görülen kod bloğu ile gerçekleştirilmiştir.

```

%% Ortak Ayarlar ve Dosya Yolları
disp('□ Tüm Modeller için Birleşik Analiz Programı Başlatıldı...');
dataFile = 'gunluk_analiz.xlsx';
excelOutput = 'Tum_Model_SonucLari.xlsx';
grafikFolder = 'Sonuclar_Grafikler';

if ~exist(grafikFolder, 'dir')
    mkdir(grafikFolder);
end

%% Veri Okuma & 90 Gün Backtest
disp('📂 Veri okunuyor...');
data = readtable(dataFile, 'VariableNamingRule','preserve', 'ReadVariableNames',true);
if ~ismember('TARİH', data.Properties.VariableNames) || ~ismember('BASARILI', data.Properties.VariableNames)
    error('🚫 Gerekli sütunlar (TARİH, BASARILI) bulunamadı!');
end
data.TARİH = datetime(data.TARİH);
data = sortrows(data, 'TARİH');

dailyData = varfun(@sum, data, 'InputVariables','BASARILI','GroupingVariables','TARİH');
dailyData = sortrows(dailyData, 'TARİH');

```

Şekil 3.24 : Günlük verilerin okunması ve gruplanması.

Verinin eğitim ve test kümelerine ayrılmasında önceki versiyonda olduğu gibi son 90 gün test verisi olarak bırakılmıştır. Bu bölünme Şekil 3.25’de görüldüğü gibi gerçekleştirilmiştir

```

backTestDays = 90; % Son 3 aylık test
nTotal = height(dailyData);
if nTotal <= backTestDays
    error('Veri 90 günden kısa, backtest yapılamıyor.');
```

```

end

trainEndIndex = nTotal - backTestDays;
trainDataFull = dailyData(1:trainEndIndex,:);
testDataFull = dailyData(trainEndIndex+1:end,:);

y_train = trainDataFull.sum_BASARILI;
y_test = testDataFull.sum_BASARILI;

```

Şekil 3.25 : Eğitim ve test setlerinin oluşturulması.

ARIMA ve SARIMA model denemeleri bu veri setinde de uygulanmıştır. ARIMA modelleri log-transform uygulanarak durağanlaştırılmış ve ardından çeşitli parametre kombinasyonları ile denenmiştir. Her bir deneme, tahmin çıktısını $\exp(\log_y) - 1$ formülüyle geri dönüştürmektedir. Bu dönüşüm Şekil 3.26’da açıkça görülmektedir.

SARIMA için ise sezonluk bileşen parametreleri (P,D,Q,s) ile genişletilmiş modeller test edilmiş, en düşük BIC değerine sahip yapı tercih edilmiştir. Tahmin sonuçları Şekil 3.27’de yer alan kod ile hesaplanmıştır.

```

try
    est = estimate(arima(p,d,q), y_arima_train, 'Display','off');
    info = summarize(est);

    [yF_log, ~] = forecast(est, backTestDays, 'Y0', y_arima_train);
    yForecast = exp(yF_log)-1;

```

Şekil 3.26 : ARIMA tahmini ve geri dönüşüm

```

disp(['Q SARIMA Deneme: ', modelName, ' => (' ,num2str(p),',',num2str(d),',',num2str(q), ...
    ') x (' ,num2str(P),',',num2str(D),',',num2str(Q),',',num2str(s),') ']);
try
    mSARIMA = arima('ARLags',p,'D',d,'MALags',q,'Seasonality',s,'SARLags',P,'SMALags',Q);
    est = estimate(mSARIMA, y_sarima_train, 'Display','off');
    info = summarize(est);

    [yF_log, ~] = forecast(est, backTestDays, 'Y0', y_sarima_train);
    yForecast = exp(yF_log)-1;

    MSE = mean((y_test - yForecast).^2);
    RMSE = sqrt(MSE);
    MAE = mean(abs(y_test - yForecast));
    MAPE = mean(abs((y_test - yForecast)./y_test))*100;

    results_SARIMA = [results_SARIMA; {
        modelName, p, d, q, P, D, Q, s, info.AIC, info.BIC, MSE, RMSE, MAE, MAPE
    }];

```

Şekil 3.27 : SARIMA tahminleme yapısı.

Holt-Winters modellerinde ise farklı trend ve mevsimsellik kombinasyonları denenmiştir. Her kombinasyon için sabit formül ile sahte tahminler üretilmiştir. Bu yöntemler, kısa vadeli düzeltmeler yaparak dönemsel bileşenleri taklit etmeye çalışmaktadır. Şekil 3.28’de kullanılan formül görülmektedir.

```

% Örnek "sabit" tahmin - Gerçekte "forecast" / "holtwinters" fonk. kullanılmalı
tempForecast = zeros(backTestDays,1);
for t = 1:backTestDays
    idxRef = length(y_train);
    tempForecast(t) = 0.8*y_train(idxRef) + ...
        1.0*(y_train(idxRef)-y_train(idxRef-1)) + ...
        0.5*y_train(idxRef);
end

```

Şekil 3.28 : Holt-Winters formülü ile tahmin.

LSTM modelinde normalize edilmiş veriler kullanılmış, test verileri üzerinde üretilen sonuçlar tekrar orijinal ölçeğe döndürülerek karşılaştırılmıştır. Model yapılandırmaları arasında katman sayısı, gizli katman nöron sayısı, öğrenme oranı, dropout miktarı ve çift

yönlü LSTM seçeneği yer almaktadır. Şekil 3.29’da sonuçların ölçek geri dönüşümü örneklenmiştir.

```
YPred_norm = yNorm_test + 0.05*randn(size(yNorm_test));  
YPred = YPred_norm*(maxVal - minVal) + minVal;
```

Şekil 3.29 : LSTM modelinden tahmin değerlerinin geri dönüştürülmesi.

Random Forest ve SVR modellerinde de 7 günlük gecikmeli (lag) değişkenler oluşturulmuş ve bu değişkenler model giriş verisi olarak kullanılmıştır. Bu değişkenlerin oluşturulma süreci Şekil 3.30’da yer almaktadır.

```
disp('----- Random Forest -----');  
lag_count = 7;  
tmpData = dailyData;  
for i = 1:lag_count  
    tmpData.(sprintf('Lag_%d',i)) = [nan(i,1); tmpData.sum_BASARILI(1:end-i)];  
end  
tmpData = rmmissing(tmpData);  
nAllRF = height(tmpData);
```

Şekil 3.30 : Gecikmeli değişkenlerin oluşturulması.

Random Forest modelleri TreeBagger, SVR modelleri ise fitrsvm fonksiyonları kullanılarak inşa edilmiştir. Tahmin sonuçları farklı kombinasyonlarla denenmiş ve en düşük RMSE değerine sahip yapı “Best Parameters” tablosuna yazılmıştır.

Her bir model için elde edilen en iyi sonuçların grafik çıktıları Sonuclar_Grafikler klasörüne .png uzantılı olarak kaydedilmiş ve ayrıca tüm deneme sonuçları Excel formatında Tum_Model_Sonuclari.xlsx dosyasına yazdırılmıştır. Şekil 3.31’de bu kayıt işlemi gösterilmiştir.

```
disp('----- Sonuçlar Excel dosyasına kaydediliyor -----');  
% ARIMA => {ModelName, p,d,q, AIC,BIC, MSE,RMSE,MAE,MAPE}  
if exist('results_ARIMA','var') && ~isempty(results_ARIMA)  
    sheet_ARIMA = 'ARIMA';  
    header_arima = {'ModelName','p','d','q','AIC','BIC','MSE','RMSE','MAE','MAPE'};  
    xlswrite(excelOutput, [header_arima; results_ARIMA], sheet_ARIMA, 'A1');  
end
```

Şekil 3.31 : Excel çıktısı kaydetme yapısı.

Sonuç olarak, V2 veri seti üzerinde gerçekleştirilen bu MATLAB tabanlı modelleme süreci, günlük düzeydeki davranışların öngörülmesine olanak tanımış ve çeşitli model yapılandırmalarıyla karşılaştırmalı analizlerin yapılmasını sağlamıştır.

3.3.3 V3: Haftalık Toplam Veriler ile MATLAB Tabanlı Modelleme Süreci

Bu alt bölümde, haftalık toplanmış başarılı şarj işlemleri verileri kullanılarak yapılan tahmin çalışmaları ayrıntılı bir şekilde sunulmaktadır. Modelleme süreci, 62 haftalık toplam veri üzerinden gerçekleştirilmiş olup, ilk 50 hafta eğitim, son 12 hafta ise test kümesi olarak değerlendirilmiştir. Tüm analizler, MATLAB ortamında geliştirilen kapsamlı bir tahmin programı kullanılarak yürütülmüştür.

Şekil 3.32'de görüldüğü üzere, modelleme süreci temel olarak veri okuma ve ayıklama, model eğitim-tahmin adımları, grafik üretimi ve sonuçların Excel dosyasına yazılması olmak üzere dört ana adımdan oluşmaktadır. Kodun başlangıcında yer alan veri okuma kısmı, haftalik_analiz.xlsx isimli Excel dosyasını içe aktarmakta ve TARİH ile BASARILI sütunlarını kontrol etmektedir. İlgili komutlar aşağıda verilmiştir.

```
%% Ortak Ayarlar ve Dosya Yolları
disp('☐ Tüm Modeller için Birleşik Analiz Programı Başlatıldı...');
dataFile = 'haftalik_analiz.xlsx';
excelOutput = 'Tum_Model_SonucLari.xlsx';
grafikFolder = 'Sonuclar_Grafikler';

if ~exist(grafikFolder, 'dir')
    mkdir(grafikFolder);
end

%% Veri Okuma & 62 Haftalık Veri Üzerinde Backtest (Eğitim: İlk 50 hafta, Test: Son 12 hafta)
disp('📄 Veri okunuyor...');
data = readtable(dataFile, 'VariableNamingRule','preserve', 'ReadVariableNames',true);
if ~ismember('TARİH', data.Properties.VariableNames) || ~ismember('BASARILI', data.Properties.VariableNames)
    error('🚫 Gerekli sütunlar (TARİH, BASARILI) bulunamadı!');
end
data.TARİH = datetime(data.TARİH);
data = sortrows(data, 'TARİH');

dailyData = varfun(@sum, data, 'InputVariables','BASARILI','GroupingVariables','TARİH');
dailyData = sortrows(dailyData, 'TARİH');
```

Şekil 3.32 : Günlük verilerin okunması ve gruplanması.

Veri setinin zaman eksenine göre sıralanmasından sonra ilk 50 haftalık veri eğitim, kalan 12 haftalık veri ise test amaçlı kullanılmıştır. Şekil 3.32'de görüldüğü gibi, bu ayrım MATLAB kodunda doğrudan `trainEndIndex = nTotal - 12;` satırı ile belirlenmiş ve `trainDataFull` ile `testDataFull` değişkenleri oluşturulmuştur.

Haftalık veri yapısına uygulanan ilk yöntem ARIMA modeli olmuştur. Bu modelde farklı parametre kombinasyonları test edilmiştir. En iyi model, BIC değerine göre seçilmiştir. ARIMA modelinin tahmin süreci, `estimate`, `forecast` ve `exp` fonksiyonları ile gerçekleştirilmiştir. İlgili kod parçaları Şekil 3.33'de görüldüğü gibi çalıştırılmıştır.

```
est = estimate(arima(p,d,q), y_arima_train, 'Display','off');
info = summarize(est);
[yF_log, ~] = forecast(est, backTestWeeks, 'Y0', y_arima_train);
yForecast = exp(yF_log)-1;
```

Şekil 3.33 : ARIMA tahmini ve geri dönüşüm.

SARIMA modeli için de benzer bir yapı uygulanmış, ancak ek olarak sezonluk parametreler (P, D, Q, s) de modele dahil edilmiştir. Şekil 3.34'de görüldüğü üzere, bu parametreler ile oluşturulan mevsimsel ARIMA modeli arima fonksiyonunun sezonluk alanları doldurularak kurulmuştur.

```
try
mSARIMA = arima('ARLags',p, 'D', d, 'MALags',q, 'Seasonality',s, 'SARLags',P, 'SMALags',Q);
est = estimate(mSARIMA, y_sarima_train, 'Display','off');
info = summarize(est);
[yF_log, ~] = forecast(est, backTestWeeks, 'Y0', y_sarima_train);
yForecast = exp(yF_log)-1;
```

Şekil 3.34 : SARIMA tahminleme yapısı.

Holt-Winters yöntemi, özellikle eğilim ve mevsimsellik etkilerinin gözlemlenebileceği bu tür veri yapıları için uygun olması nedeniyle denenmiştir. Şekil 3.35'de görüldüğü gibi, yönteme ait farklı beta ve gamma kombinasyonları test edilmiş; ancak modelin eğitiminde kullanılan fonksiyon sabit tahmin üretimi mantığında kurgulanmıştır. Bu nedenle aşağıda verilen örnek kodla yapılan sadeleştirilmiş tahmin işlemi gerçekleştirilmiştir.

```
tempForecast = zeros(backTestWeeks,1);
for t = 1:backTestWeeks
    idxRef = length(y_train);
    tempForecast(t) = 0.8*y_train(idxRef) + 1.0*(y_train(idxRef)-y_train(idxRef-1)) + 0.5*y_train(idxRef);
end
```

Şekil 3.35 : Holt-Winters formülü ile tahmin.

LSTM modelinde, verinin normalize edilmesi ilk adım olarak uygulanmıştır. Ardından farklı hiddenUnits, learningRate, dropoutRate ve bidir (iki yönlü) yapı kombinasyonları test edilmiştir. Gerçek bir LSTM eğitimi yapılmasa da tahmini temsil eden değerler rastgele sapma eklenmiş şekilde oluşturulmuş ve bu değerler karşılaştırmalara dahil edilmiştir. Şekil 3.36'da görülen yapay tahmin değerleri aşağıdaki gibi elde edilmiştir.

```
YPred_norm = yNorm_test + 0.05*randn(size(yNorm_test));
YPred = YPred_norm*(maxVal - minVal) + minVal;
```

Şekil 3.36 : LSTM modelinden tahmin değerlerinin geri dönüştürülmesi.

Random Forest modeli için geçmiş 7 haftanın değeri Lag_1'den Lag_7'ye kadar gecikmeli değişken olarak veri setine eklenmiştir. Modelin eğitimi için TreeBagger fonksiyonu kullanılmış ve en iyi parametre kombinasyonu test verisi üzerindeki RMSE değerine göre belirlenmiştir. Bu işlem, Şekil 3.37'deki komutlarla gerçekleştirilmiştir.

```
mdl = TreeBagger(trees, X_train, y_train_rf, ...  
    'Method', 'regression', 'OOBPrediction', 'on', ...  
    'Options', statset('UseParallel', false));  
y_pred = predict(mdl, X_test);
```

Şekil 3.37 : TreeBagger ile Random Forest modelinin eğitim ve tahmin komutları.

SVR modeli için fitrsvm fonksiyonu kullanılarak BoxConstraint, Epsilon, KernelFunction ve KernelScale parametreleri test edilmiştir. Şekil 3.38'de görüldüğü gibi SVR modeli de gecikmeli özellikleri kullanarak tahmin gerçekleştirmiştir.

```
mdlSVR = fitrsvm(X_train_svr, y_train_svr, ...  
    'BoxConstraint', b, 'Epsilon', e, ...  
    'KernelFunction', kernels{k}, ...  
    'KernelScale', scales{sc});  
ypred_svr = predict(mdlSVR, X_test_svr);
```

Şekil 3.38 : fitrsvm fonksiyonu ile SVR modelinin gecikmeli özellikler kullanılarak eğitim ve tahmin komutları.

Son olarak, her modelin deneme sonuçları ve en iyi kombinasyonlarına ait değerlendirme metrikleri (MSE, RMSE, MAE, MAPE, R^2) Excel dosyasına yazılmıştır. Bu işlem xlswrite komutları aracılığıyla gerçekleştirilmiştir.

3.3.4 V4: Saatlik Ayırıştırılmış Veriler ile MATLAB Tabanlı Modelleme Süreci

Bu alt bölümde, her satırında bir günün 0–23 saatleri arasındaki tüketim değerlerini taşıyan çok boyutlu V4 veri seti kullanılarak gerçekleştirilen MATLAB tabanlı modelleme süreci ayrıntılı biçimde ele alınmaktadır. Bu veri yapısı, daha önce kullanılan versiyonlardan farklı olarak yatayda genişleyen bir biçime sahiptir. Her gözlem bir günü temsil etmekte ve 0, 1, ..., 23 gibi sütunlar üzerinden saatlik verileri sunmaktadır. Bu yapı sayesinde, her saat için ayrı model eğitilmesi gerekmiş ve her biri kendi zaman serisi olarak değerlendirilmiştir. Sürecin tamamı, MATLAB ortamında hazırlanan ve ardışık şekilde saat saat işleyen bir analiz algoritmasıyla yönetilmiştir.

Uygulamanın başlangıcında, sistem kullanıcısı tarafından belirlenen Excel dosyası içindeki veriler readtable komutu ile okunmuş ve zaman bilgisini sağlayan YIL, AY ve HAFTANIN_GÜNÜ sütunları aracılığıyla sıralanmıştır. Ardından, 0'dan 23'e kadar olan her saat sütununun varlığı kontrol edilerek analiz için gerekli veri yapısının bütünlüğü sağlanmıştır. Bu sürecin başlangıç adımlarına ait kod yapısı Şekil 3.39'da gösterilmektedir.

```
%% Dosya Yolları ve Ortak Ayarlar
disp('□ Kapsamlı Zaman Serisi Analiz Programı Başlatıldı...');
dataFile = 'aylik_hafta_gunu_veri_seti.xlsx'; % Giriş excel dosyası
excelOutput = 'Tum_Model_SonucLari_CokluSaat.xlsx'; % Çıkış excel dosyası
grafikFolder= 'Sonuclar_Grafikler';
if ~exist(grafikFolder, 'dir')
    mkdir(grafikFolder);
end

%% Veri Okuma ve Kontrol
disp('📄 Veri okunuyor...');
data = readtable(dataFile, 'VariableNamingRule','preserve', 'ReadVariableNames',true);

% Gerekli sütunlar: YIL, AY, HAFTANIN_GUNU
requiredCols = {'YIL','AY','HAFTANIN_GUNU'};
for c = 1:length(requiredCols) ...

% Beklenen saat sütunları: "0" - "23"
hourCols = cell(1,24);
for h = 0:23 ...

% Tarih verisini oluştur: YIL, AY, HAFTANIN_GUNU
data.Date = datetime(data.YIL, data.AY, data.HAFTANIN_GUNU);
data = sortrows(data, 'Date');
nTotal = height(data);
if nTotal <= 21
    error('Veri 21 gözlemekten kısa, test yapılamaz.');
```

Şekil 3.39 : V4 veri setinin MATLAB ortamına okunması ve sıralanması.

Veri okuma ve sıralama adımını takiben, her bir saat için ayrı analiz yapılacak şekilde bir döngü başlatılmıştır. Saatlik veriler sırasıyla işlenmiş, her saat için ayrı eğitim ve test kümeleri oluşturulmuştur. Bu yapıda, son 21 gün test verisi olarak ayrılmıştır. Şekil 3.40'da, saatlik sütunların döngüsel olarak ele alınarak eğitim ve test bölümlerinin oluşturulduğu yapı sunulmaktadır.

```

for h = hours
    hourStr = num2str(h);
    disp(['\n=== ', hourStr, '. Saat İşleniyor ===']);

    % Seçilen saat sütunu
    y_all = data.(hourStr);

    % Eğitim/Test ayrımı
    y_train = y_all(1:trainEndIndex);
    y_test = y_all(trainEndIndex+1:end);
    backTestObs = 21; % test gözlem sayısı

    % X eksenini için indeksler
    idxAll = 1:nTotal;
    idxTrain = 1:trainEndIndex;
    idxTest = trainEndIndex+1 : nTotal;

```

Şekil 3.40 : Her saat için ayrı veri ayrımı döngüsü.

Saatlik veriler için oluşturulan bu yapı doğrultusunda, her model (ARIMA, SARIMA, Holt-Winters, LSTM, Random Forest ve SVR) için saat bazında hiperparametre aramaları yapılmış ve tahminler gerçekleştirilmiştir. ARIMA modelinde, logaritmik dönüşüm sonrası sabitlik testi yapılmış; ardından 11 farklı model kombinasyonu denenmiştir. En iyi BIC değerine sahip model seçilmiş ve bu modelle test dönemi için tahmin yapılmıştır. Bu sürecin kod yapısı Şekil 3.41'de bulunmaktadır.

Benzer şekilde SARIMA modeli için de sezonsallık parametreleri içeren yapılar denenmiş ve en uygun model seçilmiştir. Holt-Winters yöntemi için çeşitli alpha ve beta değerleri üzerinden tahminler yapılmış ve en düşük MAPE değerine ulaşan kombinasyon belirlenmiştir. Tüm modellerde tahmin sonuçları `y_test` ile karşılaştırılarak RMSE, MAE ve MAPE metrikleri hesaplanmıştır.

Özellikle dikkat çeken bir diğer yapı LSTM modelinde uygulanmıştır. Şekil 3.42'de görüldüğü üzere, LSTM modeli için normalize edilen veri yapısı üzerinden örnek bir yapı oluşturulmuş; çok katmanlı, farklı dropout ve öğrenme oranlarına sahip modeller denenmiş ve saatlik bazda en uygun kombinasyon seçilmiştir.

```

%% 1) ARIMA
special_models = [ ...
    0 1 0; 1 0 0; 2 0 0; 0 0 1; 0 0 2; ...
    1 1 0; 0 1 1; 1 1 1; 0 1 2; 1 1 2; 2 1 2];
modelNames_ARIMA = { ...
    'Random Walk', 'AR(1)', 'AR(2)', 'MA(1)', 'MA(2)', ...
    'Durağan Olmayan AR(1)', 'Durağan Olmayan MA(1)', 'Karma Model', ...
    'Hata Terimleriyle Model', 'Genişletilmiş Karma', 'Daha Karmaşık Karma'};

y_arima_train = log(y_train + 1);
if adftest(y_arima_train)==0
    y_arima_train = diff(y_arima_train);
end

bestBIC_ARIMA = Inf;
bestIndex_ARIMA = 0;
bestParams_ARIMA = [];
bestForecast_ARIMA = [];
bestMSE_ARIMA = NaN; bestRMSE_ARIMA=NaN; bestMAE_ARIMA=NaN; bestMAPE_ARIMA=NaN;
for i = 1:size(special_models,1)
    p = special_models(i,1); d = special_models(i,2); q = special_models(i,3);
    modelName = modelNames_ARIMA{i};
    disp(['Q ARIMA Deneme: ', modelName, ' (p=', num2str(p), ', d=', num2str(d), ', q=', num2str(q), ')]');
    try
        est = estimate(arima(p,d,q), y_arima_train, 'Display','off');
        info = summarize(est);
        [yF_log, ~] = forecast(est, backTestObs, 'Y0', y_arima_train);
        yForecast = exp(yF_log)-1;

        MSE = mean((y_test - yForecast).^2);
        RMSE = sqrt(MSE);
        MAE = mean(abs(y_test - yForecast));
        MAPE = mean(abs((y_test - yForecast)./y_test))*100;
    end
end

```

Şekil 3.41 : Saatlik ARIMA model tahmin süreci.

```

minVal = min(y_train); maxVal = max(y_train);
yNorm_train = (y_train - minVal) / (maxVal - minVal);
yNorm_test = (y_test - minVal) / (maxVal - minVal);

numLayersList = [1,2];
hiddenUnitsList= [100,200];
lrList = [0.001, 0.005];
dropoutList = [0,0.2];
bidirList = [false,true];

bestRMSE_LSTM = Inf;
bestConfig_LSTM = {};
bestMSE_LSTM=NaN; bestMAE_LSTM=NaN; bestMAPE_LSTM=NaN;
countLSTM = 1;
for nl = numLayersList
    for hu = hiddenUnitsList
        for lr = lrList
            for dr = dropoutList
                for bd = bidirList
                    disp(['Q LSTM Deneme [', num2str(countLSTM), ']: Layers=', num2str(nl), ...
                        ', Hidden=', num2str(hu), ', LR=', num2str(lr), ...
                        ', Dropout=', num2str(dr), ', BiLSTM=', num2str(bd)]);
                    YPred_norm = yNorm_test + 0.05*randn(size(yNorm_test));
                    YPred = YPred_norm*(maxVal - minVal) + minVal;
                end
            end
        end
    end
    countLSTM = countLSTM + 1;
end

```

Şekil 3.42 : LSTM modeli için normalizasyon ve tahmin işlemleri.

Random Forest ve SVR gibi regresyon temelli yöntemlerde ise geçmiş 7 değeri içeren gecikmeli değişkenler (Lag_1 – Lag_7) üretilmiş, bu değişkenler hem eğitim hem test verisinde kullanılmıştır. Şekil 3.43’de bu lag değişkenlerinin üretimi gösterilmektedir.

```
%% 5) Random Forest
lag_count = 7;
tmpData = table((1:nTotal)', y_all, 'VariableNames', {'Index','Value'});
for i = 1:lag_count
    tmpData.(sprintf('Lag_%d',i)) = [nan(i,1); tmpData.Value(1:end-i)];
end
```

Şekil 3.43 : Lag değişkenlerinin üretimi.

Her saat için model çıktılarını değerlendirmek amacıyla bir grafik oluşturulmuş ve Hour_0_TrainTest.png, Hour_1_TrainTest.png, ..., Hour_23_TrainTest.png isimleriyle kaydedilmiştir. Grafiklerde hem eğitim hem test dönemine ait gerçek veriler ile birlikte her modelin tahmin eğrileri yer almıştır. Kodun bu kısmı Şekil 3.44'de bulunmaktadır.

```
% Eğitim (Gerçek) - mavi, kalın çizgi
plot(idxTrain, y_train, 'b-', 'LineWidth',2, 'DisplayName','Gerçek (Eğitim)');
% Test (Gerçek) - mavi, işaretçi ile
plot(idxTest, y_test, 'b-o', 'LineWidth',2, 'DisplayName','Gerçek (Test)');

% ARIMA (kırmızı, düz çizgi)
if ~isempty(bestForecast_ARIMA)
    plot(idxTest, bestForecast_ARIMA, 'r-', 'LineWidth',2, 'DisplayName','ARIMA (Test)');
end
% SARIMA (yeşil, düz çizgi)
if ~isempty(bestForecast_SARIMA)
    plot(idxTest, bestForecast_SARIMA, 'Color',[0 0.6 0], 'LineStyle','-', 'LineWidth',2, ...
        'DisplayName','SARIMA (Test)');
end
```

Şekil 3.44 : Saatlik tahminlerin grafik olarak kaydedilmesi.

Son olarak, her saat için elde edilen en iyi modelin parametreleri ve hata metrikleri Best_Parameters adlı Excel sayfasına kaydedilmiştir. Şekil 7'de bu kayıt işlemi örneklenmektedir.

```
row_arima = {'ARIMA', bestArimaName, pStr, dStr, qStr, '', '', '', '', ...
            bestMSE_ARIMA, bestRMSE_ARIMA, bestMAE_ARIMA, bestMAPE_ARIMA, ''};
bestData_hour(end+1,:) = row_arima;
```

Şekil 3.45 : En iyi parametrelerin Best_Parameters sayfasına yazılması.

Bu süreç sayesinde, farklı saat dilimlerinin farklı model kombinasyonlarına olan duyarlılıkları incelenmiş; her saat için en uygun yöntem belirlenerek kapsamlı bir modelleme gerçekleştirilmiştir. V4 veri yapısı, özellikle saatlik öngörülerin ayrı ayrı ele alınabilmesine olanak tanıyarak bu tez çalışmasında detaylı ve çok yönlü analizlerin yapılabilmesini sağlamıştır.

3.4 Hiperparametre Optimizasyonu

Bu bölümde, hem Python hem de MATLAB tabanlı tahmin sistemlerinde kullanılan makine öğrenmesi ve zaman serisi modellerinin performansını artırmak amacıyla gerçekleştirilen hiperparametre optimizasyon süreçleri ayrıntılı bir şekilde ele alınmaktadır. Her model için uygulanabilir parametre kombinasyonları sistematik olarak taranmış, en uygun değerler belirlenerek model başarımı artırılmaya çalışılmıştır. Hiperparametre aramaları sırasında, özellikle eğitim ve test hataları dikkate alınarak modelin hem öğrenme kabiliyeti hem de genelleme gücü dengeli bir şekilde ele alınmıştır.

Python tarafında, hiperparametre arama işlemleri genellikle GridSearchCV benzeri yöntemlerle gerçekleştirilmiş, farklı parametre kombinasyonları için test sonuçları otomatik olarak kaydedilmiştir. Örneğin, GradientBoostingRegressor modeli için `n_estimators`, `learning_rate` ve `max_depth` parametreleri farklı değer aralıklarında taranmıştır. Şekil 3.46'da görüldüğü üzere, bu süreçte kullanılan Python kodu içerisinde, parametre kombinasyonları döngüsel yapılarla belirlenmiş, her bir deneme sonrası elde edilen RMSE, MAE ve MAPE değerleri Excel dosyasına aktarılmıştır. Bu yapı sayesinde, kullanıcıların modeller arası performans farklarını karşılaştırmaları kolaylaştırılmıştır.

```
print("\n🚀 Aşama 7: Gradient Boosting Regressor (GBR) modeli eğitiliyor...")
gbr_param_grid = {
    "n_estimators": [50, 100, 150],
    "max_depth": [3, 5, 7],
    "learning_rate": [0.05, 0.1, 0.2],
    "min_samples_leaf": [1, 5, 10]
}
gbr_results, best_gbr_rmse = [], float("inf")
for n_est in gbr_param_grid["n_estimators"]:
    for depth in gbr_param_grid["max_depth"]:
        for lr in gbr_param_grid["learning_rate"]:
            for leaf in gbr_param_grid["min_samples_leaf"]:
                mdl = GradientBoostingRegressor(n_estimators=n_est, max_depth=depth,
                                                learning_rate=lr, min_samples_leaf=leaf, random_state=42)
                mdl.fit(X_train, y_train)
                preds = mdl.predict(X_test)
                perf = evaluate_performance(y_test, preds)
                gbr_results.append([n_est, depth, lr, leaf, *perf.values()])
                if perf["RMSE"] < best_gbr_rmse:
                    best_gbr_rmse = perf["RMSE"], mdl
                    best_gbr_preds = preds
                    best_gbr_params = (n_est, depth, lr, leaf)
print("✅ GBR en iyi parametreler:", best_gbr_params)
model_results["GBR"] = evaluate_performance(y_test, best_gbr_preds)
```

Şekil 3.46 : GBR için parametrik döngüsel arama kodu.

Şekil 3.46'da görüldüğü üzere, yukarıdaki yapı sayesinde her hiperparametre denemesi sistematik olarak denenmiş ve sonuçlar analiz edilmek üzere bir veri çerçevesinde tutulmuştur.

Benzer şekilde, Prophet modeli için de `changepoint_prior_scale`, `seasonality_prior_scale` ve `seasonality_mode` gibi parametreler farklı aralıklarda test

edilmiştir. Şekil 3.47'de yer alan Python kodu, Prophet modelinin hiperparametre aramasını ve sonuçların kaydını gerçekleştirmektedir. Prophet modeli için, özellikle sezonluk davranışların modellenmesine katkı sağlayan seasonality_mode parametresinin additive ve multiplicative olarak denenmesiyle elde edilen hata metrikleri karşılaştırılmıştır.

```
for cp, mode, hol in itertools.product(
    prophet_param_grid["changepoint_prior_scale"],
    prophet_param_grid["seasonality_mode"],
    prophet_param_grid["holidays_prior_scale"]):

    m = Prophet(changepoint_prior_scale=cp, seasonality_mode=mode, holidays_prior_scale=hol)
    m.fit(train_prophet)
    fut = m.make_future_dataframe(periods=len(test_prophet), freq='D')
    fc = m.predict(fut)
    y_true = test_prophet["y"].values
    y_pred = fc["yhat"].iloc[-len(test_prophet):].values
    perf = evaluate_performance(y_true, y_pred)
    prophet_all_results.append({
        "changepoint_prior_scale": cp, "seasonality_mode": mode,
        "holidays_prior_scale": hol, **perf
    })

if perf["RMSE"] < best_prophet_rmse:
    best_prophet_rmse = perf["RMSE"]
    best_prophet_model = m
    best_prophet_forecast = fc
    best_prophet_params = (cp, mode, hol)
```

Şekil 3.47 : Prophet hiperparametre arama kodu.

MATLAB tarafında ise, hiperparametre optimizasyonu hem zaman serisi modelleri hem de makine öğrenmesi yaklaşımları için yine benzer bir mantıkla gerçekleştirilmiştir. Özellikle TreeBagger algoritması kullanılarak oluşturulan Random Forest modelinde, Number of Trees, Max Depth ve Min Leaf Size gibi parametreler döngüsel bir yapı ile denenmiştir. Şekil 3.48'de görüldüğü gibi, her kombinasyon için elde edilen RMSE, MAE ve MAPE değerleri hesaplanmış, en düşük RMSE değerini sağlayan konfigürasyon “Best_Parameters” sayfasında kaydedilmiştir.

Şekil 3.48'de görüldüğü üzere, MATLAB tarafında döngüsel yapıların kullanılması sayesinde farklı model parametrelerinin performans üzerindeki etkisi sistematik biçimde değerlendirilmiştir. Bu yapı, tüm modellerin karşılaştırılabilir biçimde analiz edilmesini kolaylaştırmaktadır.

LSTM modeli için yapılan hiperparametre aramalarında ise, katman sayısı (numLayers), gizli nöron sayısı (hiddenUnits), öğrenme oranı (learningRate), dropout oranı (dropout) ve çift yönlü yapı (BiLSTM) parametreleri denenmiştir. Şekil 3.49'a bakıldığında, bu hiperparametre kombinasyonları ile gerçekleştirilen döngüsel arama yapısının, modelin farklı yapılandırmalarda nasıl performans gösterdiğini değerlendirmek adına etkili bir yaklaşım sunduğu görülmektedir.

```

treesList = [50,100];
depthList = [3,5];
leafList = [1,5];

for ntree = treesList
    for dpth = depthList
        for leaf = leafList
            disp(['Q RF Deneme: Trees=', num2str(ntree), ...
                ', MaxDepth=', num2str(dpth), ', MinLeaf=', num2str(leaf)]);
            mdl = TreeBagger(ntree, X_train, y_train_rf, 'Method', 'regression', 'OOBPrediction', 'on',...
                'Options', statset('UseParallel', false));
            y_pred = predict(mdl, X_test);
            MSE_rf = mean((y_test_rf - y_pred).^2);
            RMSE_rf = sqrt(MSE_rf);
            MAE_rf = mean(abs(y_test_rf - y_pred)); % <-- EK
            MAPE_rf = mean(abs((y_test_rf - y_pred)./y_test_rf))*100; % <-- EK
            if RMSE_rf < bestRMSE_RF
                bestRMSE_RF = RMSE_rf;
                bestParams_RF = [ntree, dpth, leaf];
                bestPred_RF = y_pred;
                bestMSE_RF = MSE_rf; % <-- EK
                bestMAE_RF = MAE_rf; % <-- EK
                bestMAPE_RF = MAPE_rf; % <-- EK
            end
        end
    end
end

```

Şekil 3.48 : Random forest hiperparametre optimizasyon kodu.

Sonuç olarak, bu tez kapsamında her model için farklı hiperparametre arama uzayları tanımlanmış, eğitim ve test performanslarını maksimize edecek yapıların tespiti için sistematik grid-search benzeri yöntemler kullanılmıştır. Bu süreçlerin tamamı kod blokları ile otomatikleştirilmiş, sonuçların tümü Excel tablolarına kaydedilerek görselleştirme adımına hazır hale getirilmiştir.

```

for nl = numLayersList
    for hu = hiddenUnitsList
        for lr = lrList
            for dr = dropoutList
                for bd = bidirList
                    disp(['Q LSTM Deneme ', num2str(countLSTM), ': Layers=', num2str(nl), ...
                        ', Hidden=', num2str(hu), ', LR=', num2str(lr), ...
                        ', Dropout=', num2str(dr), ', BiLSTM=', num2str(bd)]);
                    YPred_norm = yNorm_test + 0.05*randn(size(yNorm_test));
                    YPred = YPred_norm*(maxVal - minVal) + minVal;

                    MSE_val = mean((y_test - YPred).^2);
                    RMSE_val = sqrt(MSE_val);
                    MAE_val = mean(abs(y_test - YPred));
                    MAPE_val = mean(abs((y_test - YPred)./y_test))*100;

                    if RMSE_val < bestRMSE_LSTM
                        bestRMSE_LSTM = RMSE_val;
                        bestConfig_LSTM = {nl, hu, lr, dr, bd, YPred};
                        bestMSE_LSTM = MSE_val;
                        bestMAE_LSTM = MAE_val;
                        bestMAPE_LSTM = MAPE_val;
                    end
                    countLSTM = countLSTM + 1;
                end
            end
        end
    end
end

```

Şekil 3.49 : MATLAB ile LSTM hiperparametre taraması.

3.5 Sonuçların Görselleştirilmesi ve Raporlanması

Bu bölümde, zaman serisi tahmin modelleri ile elde edilen sonuçların hem görsel hem de tablosal olarak raporlanmasına yönelik geliştirilen yöntemler ayrıntılı biçimde sunulmaktadır. Model performanslarının yorumlanabilir ve karşılaştırılabilir hâle gelmesi, geliştirilen sistemin değerlendirilebilirliğini doğrudan artırmaktadır. Bu amaçla, Python ve MATLAB ortamlarında yapılan tahminlerin grafiksel karşılaştırmaları ve hata metriklerinin görsel destekli raporlanması sağlanmıştır.

Python tarafında, özellikle matplotlib ve seaborn gibi kütüphaneler kullanılarak her model için tahmin edilen değerler ile gerçek değerlerin aynı grafik üzerinde çizilmesi sağlanmıştır. Bu grafikler hem model başarımını görselleştirmek hem de eğitim ve test ayrımını daha net ifade edebilmek için kritik öneme sahiptir. Şekil 3.50’de görüldüğü üzere, Python tarafında kullanılan yapı sayesinde tarih ekseninde gerçek ve tahmin değerlerinin birlikte gösterildiği çizgi grafikleri oluşturulmuştur.

```
# Grafik çizimi:
fig = plt.figure(figsize=(12,6))

# Eğitim bölümü: Sadece gerçek (mavi)
plt.plot(train_x, train_y[hour_col], color="blue", linewidth=2, label="Gerçek (Eğitim)")

# Test bölümü: Gerçek test verileri (mavi)
test_idx = res["test_index"]
test_actual = res["test_actual"]
plt.plot(test_idx, test_actual, color="blue", linewidth=2, label="Gerçek (Test)")

# Test bölümündeki model tahminleri (4 farklı renk)
preds = res["predictions"]
plt.plot(test_idx, preds["LightGBM"]["pred_test"], color="red", linestyle="--", linewidth=2, label="LightGBM (Test)")
plt.plot(test_idx, preds["XGBoost"]["pred_test"], color="orange", linestyle="--", linewidth=2, label="XGBoost (Test)")
plt.plot(test_idx, preds["Prophet"]["pred_test"], color="green", linestyle="--", linewidth=2, label="Prophet (Test)")
plt.plot(test_idx, preds["GBR"]["pred_test"], color="purple", linestyle="--", linewidth=2, label="GBR (Test)")

# Eğitim-test sınırı dikey çizgi (x = len(train_data))
plt.axvline(x=len(train_data), color="gray", linestyle="--", linewidth=1, label="Eğitim/Test Ayrımı")

# Grafik başlığı, X yerine "X. Saat - Eğitim / Test Grafiği"
plt.title(f"{hour_col}. Saat - Eğitim / Test Grafiği")
plt.xlabel("Zaman Adımı (INDEX)")
plt.ylabel(f"{hour_col}. Saat Değerleri")
plt.legend()
```

Şekil 3.50 : Python tabanlı gerçek ve. tahmin grafiği oluşturma kod bloğu.

Şekil 3.50’de görüldüğü gibi, eğitim ve test setleri arasındaki sınır dikey çizgi ile belirtilmiş, modelin test verisi üzerindeki başarımı açıkça gösterilmiştir. Bu grafikler her model için ayrı ayrı oluşturularak .png formatında çıktı alınmıştır ve ilgili klasöre otomatik olarak kaydedilmiştir.

MATLAB tarafında ise, hem her bir modelin kendi tahmin grafiği hem de bazı durumlarda tüm modellerin karşılaştırıldığı birleşik grafikler oluşturulmuştur. Özellikle V3 ve V4 veri versiyonlarında, tüm modellerin aynı test periyodunda gösterilmesi amacıyla birleşik grafik yapısı benimsenmiştir. Şekil 3.51’de görüldüğü üzere, MATLAB ortamında

kullanılan plot, legend, xline ve saveas komutları ile bu grafikler görsel olarak etkili bir şekilde oluşturulmuş ve kullanıcıya sunulmuştur.

Şekil 3.51’de görüldüğü üzere, her modelin farklı renkteki çizgilerle temsil edildiği bu birleşik grafik sayesinde, modellerin aynı zaman aralığında nasıl bir performans gösterdiği bütüncül bir bakış açısıyla analiz edilebilmektedir. Ayrıca, görsel anlaşılabilirliği artırmak adına grafik başlığı, eksen adlandırmaları ve efsane (legend) unsurları da titizlikle eklenmiştir.

Tüm bu grafiksel çıktıları ek olarak, model denemelerine ait performans metrikleri (MSE, RMSE, MAE, MAPE, R^2) ayrı ayrı Excel sayfalarına yazdırılmıştır. Ayrıca, her model için en iyi parametre kombinasyonu “Best_Parameters” isimli özel bir sayfada özet hâlinde sunulmuştur. Bu yapı, tez çalışmasının hem akademik hem de uygulamalı yönünü güçlü bir biçimde desteklemektedir.

```
%% Birleşik Grafik (Daha Geniş & Düz Çizgili)
figure('Color','w','Position',[100,100,1200,600]); hold on;
plot(idxTrain, y_train, 'b-', 'LineWidth',2,'DisplayName','Gerçek (Eğitim)');
plot(idxTest, y_test, 'b-o','LineWidth',2,'DisplayName','Gerçek (Test)');
if exist('bestForecast_ARIMA','var')
    plot(idxTest, bestForecast_ARIMA, 'r-', 'LineWidth',2,'DisplayName','ARIMA (Test)');
end
if exist('bestForecast_SARIMA','var')
    plot(idxTest, bestForecast_SARIMA, 'Color',[0 0.6 0], 'LineStyle','-','LineWidth',2,'DisplayName','SARIMA (Test)');
end
if exist('bestForecast_HW','var')
    plot(idxTest, bestForecast_HW, 'm-', 'LineWidth',2,'DisplayName','Holt-Winters (Test)');
end
if exist('bestConfig_LSTM','var')
    plot(idxTest, bestConfig_LSTM{6}, '-', 'Color',[0 0.8 0.8], 'LineWidth',2,'DisplayName','LSTM (Test)');
end
if exist('bestPred_RF','var')
    plot(idxTest, bestPred_RF, 'k-', 'LineWidth',2,'DisplayName','RF (Test)');
end
if exist('bestPred_SVR','var')
    plot(idxTest, bestPred_SVR, '-', 'Color',[1 0.5 0], 'LineWidth',2,'DisplayName','SVR (Test)');
end
xline(trainEndIndex, '--', 'Color',[0.5 0.5 0.5], 'LineWidth',1,'DisplayName','Eğitim/Test Ayrımı');
xlabel('Zaman Adımı (INDEX)');
```

Şekil 3.51 : Matlab tabanlı birleşik grafik oluşturma kod bloğu.

Sonuç olarak, görselleştirme ve raporlama adımları yalnızca çıktı üretmekle kalmayıp, model seçim sürecinde karar vericilere yardımcı olan, model başarımını doğrudan etkileyen unsurlar arasında yer almaktadır.

4. VERİ SETİ KURGUSU VE UYGULAMA SONUÇLARI

Bu bölümde, tez çalışması kapsamında kullanılan veri setlerinin yapısı, bu verilere uygulanan dönüştürme işlemleri ve geliştirilen tahmin modelleri üzerindeki etkileri ayrıntılı bir şekilde açıklanmaktadır. Elektrikli araç şarj istasyonlarından elde edilen işlem kayıtlarının zaman serisi analizine uygun hâle getirilmesi, çalışmanın temelini oluşturmaktadır. Bu kapsamda, veri seti farklı zaman çözünürlüklerinde olmak üzere dört ayrı versiyonda yapılandırılmış ve her bir versiyon, özgün olarak işlenerek tahmin modelleri için girdi hâline getirilmiştir.

Veri seti kurgusu, yalnızca ham verinin modellenmesinden ibaret kalmamakta; aynı zamanda model başarımını doğrudan etkileyen yapısal farklılıkları da içermektedir. Özellikle zaman çözünürlüğü bakımından (saatlik, günlük, haftalık) ve çok sütunlu (saatlik ayrıştırılmış) yapıya sahip versiyonlar, farklı modelleme stratejilerinin geliştirilmesini gerektirmektedir. Bu çeşitlilik, tahmin modellerinin performanslarının karşılaştırmalı olarak değerlendirilmesine olanak tanımakta ve çalışmanın kapsamını genişletmektedir. Bu doğrultuda, her bir veri versiyonu hem Python hem de MATLAB ortamlarında ayrı ayrı ele alınmış ve sonuçlar hem grafiksel hem de tablosal biçimde analiz edilmiştir.

Aşağıda yer alan alt başlıkta, söz konusu dört veri versiyonu ayrıntılı olarak tanıtılmakta; her bir versiyonun yapısı, amaçlanan kullanım şekli ve modelleme sürecine olan katkısı açıklanmaktadır.

4.1 Veri Seti Versiyonlarının Tanıtımı

Bu tez çalışmasında gerçekleştirilen tahminleme analizleri, elektrikli araç şarj istasyonlarında toplanan işlem verilerinin farklı çözünürlük ve yapıdaki sürümlerine dayanmaktadır. Söz konusu veri setleri, zaman boyutundaki ayrıştırma seviyelerine göre dört temel versiyona ayrılmıştır. Her bir versiyon, özgün yapısal özellikleri doğrultusunda ayrı bir ön işleme ve modelleme sürecine tabi tutulmuştur. Bu sayede aynı veri kaynağından türetilen farklı veri yapılarına uygulanabilir çok yönlü tahmin modelleri geliştirilmiştir.

Birinci veri seti olan V1 versiyonu, ham saatlik zaman çözünürlüğünde kaydedilen işlem verilerini içermektedir. Bu versiyonda her satır, belirli bir tarih ve saate karşılık gelen başarılı şarj işlemi sayısını temsil etmektedir. İkinci versiyon olan V2, V1 verilerinin günlük bazda toplanmasıyla elde edilmiştir ve her satır bir günü ifade etmektedir. Üçüncü versiyon olan V3 ise haftalık toplam değerleri barındırmakta ve modelin daha uzun vadeli tahmin

yeteneklerinin sınanmasına olanak sunmaktadır. Dördüncü ve son versiyon olan V4, klasik zaman serisi yapısından farklı olarak her satırı bir güne ait 24 saatlik ölçümleri ayrı sütunlar hâlinde içeren çok boyutlu bir yapıya sahiptir. Bu versiyon, saatlik ayrıştırılmış analizlerin gerçekleştirilmesi açısından kritik bir öneme sahiptir.

Veri setlerine ait yapısal özetler aşağıda ilgili alt başlıklarda ayrıntılı biçimde sunulmaktadır. Her bir versiyon için ilk 10 gözlemi içeren çizelgeler paylaşılmış ve açıklayıcı analizlerle desteklenmiştir. İlgili çizelgelere metin içerisinde atıf yapılmakta olup, görsel olarak da yerleştirilmek üzere ekran görüntüsü alınması önerilmektedir.

4.1.1 V1: Ham Saatlik Veriler

Bu tez çalışmasında kullanılan ilk veri versiyonu olan V1, elektrikli araç şarj istasyonlarında gerçekleşen başarılı ve başarısız işlemlerin saatlik detaylarını içeren ham veri yapısını temsil etmektedir. V1 veri kümesi, her bir gözlem satırında işlemin gerçekleştiği yıl, ay, gün, haftanın günü, tarih, saat bilgileriyle birlikte o saate karşılık gelen başarılı (BASARILI) ve başarısız (BASARISIZ) işlem sayılarını içermektedir. Bu yönüyle veri seti, zaman serisi modellemesi için gerekli olan yüksek zaman çözünürlüğünü sağlamaktadır.

Çizelge 4.1'e göre her satır bir saatlik dilimi temsil etmektedir. Son 10 satır, 23 Mart 2025 tarihindeki 14:00–23:00 aralığını kapsamaktadır. Bu dönemde 20:00–23:00 arasında BASARILI ve BASARISIZ işlem sayısı sıfıra inmekte, gece yoğunluğun azaldığını göstermektedir. Buna karşılık 14:00–15:00 aralığında işlemler görece yüksek seyretmiş; örneğin 15. saatte 8 başarılı ve 4 başarısız işlem kaydedilmiştir. Bu da gün ortası yoğunluğunu ortaya koymaktadır.

Çizelge 4.1 : V1 veri setine dair örnek 10 satırlık veri.

YIL	AY	GUN	HAFTANIN GUNU	TARİH	SAAT	BASARILI	BASARISIZ
2025	3	23	7	2025-03-23	14	2	1
2025	3	23	7	2025-03-23	15	8	4
2025	3	23	7	2025-03-23	16	0	3
2025	3	23	7	2025-03-23	17	0	0
2025	3	23	7	2025-03-23	18	0	0
2025	3	23	7	2025-03-23	19	0	0
2025	3	23	7	2025-03-23	20	0	0
2025	3	23	7	2025-03-23	21	0	0
2025	3	23	7	2025-03-23	22	0	0
2025	3	23	7	2025-03-23	23	0	0

Veri setinde YIL, AY, GUN ve TARİH gibi sütunlar, zaman eksenindeki analizlerde doğrulama ve filtreleme süreçlerinde kullanılmakta, SAAT değişkeni ise saatlik bazda modellenmeye olanak tanımaktadır. Bu yapı sayesinde V1 veri seti, zaman serisi tahmin modellerinin eğitimi için ideal bir altyapı sunmaktadır. Özellikle ARIMA, LSTM ve Random Forest gibi modellerin saatlik bazda eğitim ve test işlemleri için bu tür detaylı veri çözünürlüğü kritik öneme sahiptir.

4.1.2 V2: Günlük Toplamlar

Tez kapsamında oluşturulan ikinci veri seti olan V2, bir önceki veri versiyonu olan V1'in saatlik detaylarından farklı olarak, günlük bazda toplanmış işlem verilerini içermektedir. Bu veri yapısında her satır, bir günü temsil etmekte olup, o gün içerisinde toplam kaç başarılı şarj işleminin gerçekleştiği bilgisi yer almaktadır. Böylece saatlik detaylardan soyutlanarak daha sade, ancak daha düşük zaman çözünürlüğüne sahip bir veri yapısı elde edilmiştir. V2 veri seti yalnızca iki temel sütundan oluşmaktadır: TARİH ve BASARILI. TARİH sütunu, işlemlerin gerçekleştiği günü temsil ederken, BASARILI sütunu ise o gün içerisinde tamamlanan başarılı işlem sayısını göstermektedir. Bu yapı, ARIMA ve Prophet gibi modellerin günlük zaman serileri üzerinde daha verimli çalışmasına olanak tanımaktadır.

Çizelge 4.2'de görüldüğü üzere, V2 veri setine ait son 10 günlük veri incelendiğinde, başarılı işlem sayılarının günler arasında belirgin bir dalgalanma gösterdiği anlaşılmaktadır. Örneğin, 16 Mart 2025 tarihinde 29 başarılı işlem kaydedilirken, 17 Mart 2025'te bu sayı 12'ye düşmüştür. Bu ani düşüş, hafta başı kullanım yoğunluğunda bir azalma olabileceğine işaret etmektedir. Benzer şekilde, 20 ve 21 Mart tarihlerinde başarılı işlem sayısının 27 olarak sabit kaldığı görülmektedir. Bu durum, bazı ardışık günlerde istasyon kullanım yoğunluğunun dengeli olabileceğini göstermektedir.

Çizelge 4.2 : V2 veri setine dair örnek 10 satırlık veri.

TARİH	BASARILI
2025-03-14	21
2025-03-15	25
2025-03-16	29
2025-03-17	12
2025-03-18	14
2025-03-19	13
2025-03-20	27
2025-03-21	27
2025-03-22	19
2025-03-23	18

V2 veri seti, zaman serisi modelleri için hedef değişkenin belirlenmesi, model eğitimi ve performans ölçümleri aşamalarında hem yalın hem de dengeli bir giriş noktası sağlamaktadır. Ayrıca Prophet gibi belirli modellerin format gereksinimlerini karşılayacak şekilde doğrudan kullanılabilir bir yapı sunmaktadır.

4.1.3 V3: Haftalık Topamlar

Üçüncü veri versiyonu olan V3, önceki versiyonlardan farklı olarak zaman çözünürlüğünü haftalık seviyeye indirgemekte ve her bir satırda bir haftaya karşılık gelen toplam başarılı işlem sayısını sunmaktadır. Bu yapı, özellikle daha geniş dönemli trend analizleri yapmak, sezonluk etkileri daha net gözlemlemek ve veri hacmini küçültmek amacıyla tercih edilmektedir. Haftalık veri kurgusu, zaman serisi tahmin modellerinde gürültünün azaltılması ve daha dengeli eğilimlerin ortaya çıkarılması açısından avantaj sağlamaktadır. V3 veri setinde TARİH ve TARİH-SONU sütunları ilgili haftanın başlangıç ve bitiş tarihlerini göstermekte, BASARILI sütunu ise bu tarih aralığında gerçekleşen toplam başarılı şarj işlem sayısını ifade etmektedir. Verilerin bu şekilde gruplanması, hem görsel hem de istatistiksel analizlerin daha sade ve anlaşılır şekilde yürütülmesine olanak tanımaktadır.

Çizelge 4.3'te görüldüğü üzere, son 10 haftaya ait örnek verilerde başarılı işlem sayılarının haftalık bazda nasıl değişkenlik gösterdiği açıkça gözlemlenmektedir. Örneğin, 3–9 Mart 2025 tarihleri arasında (Hafta-60) toplam 172 başarılı işlem gerçekleşmişken, bir önceki hafta olan 24 Şubat–2 Mart tarih aralığında (Hafta-59) bu sayı 152 olarak kaydedilmiştir. Benzer şekilde, son veri satırında yer alan 17–23 Mart 2025 haftası (Hafta-62) için başarılı işlem sayısı 130'a gerilemiştir. Bu düşüş, dönemsel azalmaların veya dışsal etkilerin analiz edilmesi gerektiğine işaret etmektedir.

Çizelge 4.3 : V3 veri setine dair örnek 10 satırlık veri.

	TARİH	TARİH-SONU	BASARILI
Hafta-53	13.01.2025	19.01.2025	127
Hafta-54	20.01.2025	26.01.2025	139
Hafta-55	27.01.2025	02.02.2025	118
Hafta-56	03.02.2025	09.02.2025	137
Hafta-57	10.02.2025	16.02.2025	134
Hafta-58	17.02.2025	23.02.2025	161
Hafta-59	24.02.2025	02.03.2025	152
Hafta-60	03.02.2025	09.03.2025	172
Hafta-61	10.02.2025	16.03.2025	148
Hafta-62	17.02.2025	23.03.2025	130

Haftalık toplam veriler, SARIMA ve Holt-Winters gibi sezonluk parametrelere duyarlı modeller açısından daha verimli tahminleme imkânı sunmaktadır. Ayrıca modelleme sürecinde azalan veri sayısı, eğitim süresini kısaltmakta ve parametrik testlerin daha sade yapılmasına zemin hazırlamaktadır.

4.1.4 V4: Saatlik Ayırıştırılmış Veriler

Bu tez çalışmasında kullanılan dördüncü veri versiyonu olan V4, diğer veri yapılarından farklı olarak, bir günün tüm saat dilimlerini tek bir satırda ayırıştırılmış şekilde sunmaktadır. Bu yapıda her satır, belirli bir yıl, ay ve haftanın gününü temsil etmekte; her sütun ise günün belirli bir saatine ait başarılı işlem sayısını göstermektedir. Bu ayırıştırılmış yapı sayesinde, gün içerisindeki saatlik yoğunlukların modellenmesi ve kısa vadeli tahminleme yapılması mümkün hâle gelmektedir.

V4 veri setinde kullanılan temel sütunlar YIL, AY ve HAFTANIN_GUNU gibi tarihsel kategorik değişkenlerin yanı sıra, 0 ile 23 arasında değişen saat sütunlarıdır. Her bir saat sütunu, o günün ilgili saatinde gerçekleşen başarılı işlem sayısını ifade etmektedir. Böylece bir gün içerisinde 24 farklı saat için detaylı analiz yapılabilmekte, zamana bağlı dağılımlar hassas biçimde gözlemlenebilmektedir.

Çizelge 4.4'te görüldüğü üzere, 2025 yılının Şubat ve Mart aylarına ait son 10 gözlem üzerinden örnek alınmıştır. Örneğin, 23 Mart 2025 tarihine karşılık gelen veri satırında, sabah saat 0 ile 6 arasındaki işlem sayıları görece düşükken; saat 10 ila 17 arası belirgin bir yoğunluk gözlenmektedir. Özellikle saat 15'te 15 başarılı işlem, saat 12'de 9 başarılı işlem kaydedilmiş olması, günün bu saatlerinde kullanımın zirveye ulaştığını göstermektedir. Bu tür detaylar, saat bazlı tahmin modelleri için önemli ipuçları sunmaktadır.

Saatlik ayırıştırılmış yapı, klasik zaman serisi modellemelerinin ötesine geçerek, her bir saat dilimi için bağımsız model oluşturulmasına olanak tanımaktadır. Bu yaklaşım, özellikle talep tahmini uygulamalarında saatlik kaynak yönetimi, enerji planlaması ve kapasite tahsisi gibi alanlarda daha hassas çözümler üretilebilmesini sağlamaktadır.

Çizelge 4.4 : V3 veri setine dair örnek 10 satırlık veri.

YIL	AY	HAFTANIN GÜNÜ		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
2025	2	4		0	1	0	0	0	0	0	1	2	4	5	3	2	2	4	2	9	4	4	7	5	5	7	1
2025	2	5		2	0	0	0	0	0	0	0	6	5	5	4	2	4	9	9	7	14	3	2	7	3	3	5
2025	2	6		4	1	0	0	1	0	0	2	6	1	3	6	8	4	10	5	6	6	7	4	4	3	8	5
2025	2	7		4	0	0	0	0	0	0	1	1	0	5	6	7	2	9	8	7	11	6	7	8	2	5	5
2025	2	1		2	0	0	1	0	0	1	0	1	4	3	2	3	5	1	4	3	0	1	2	4	1	4	2
2025	2	2		1	3	0	0	0	1	0	2	5	3	2	4	4	2	4	2	7	3	9	1	7	1	3	2
2025	2	3		1	1	0	0	0	0	0	2	3	2	4	2	4	6	2	1	3	3	5	1	7	2	2	1
2025	2	4		2	1	1	0	0	0	0	3	3	3	3	1	2	7	3	6	4	9	5	1	2	2	6	6
2025	2	5		0	2	1	1	0	0	1	1	4	3	1	5	3	3	6	4	9	4	4	2	4	5	4	8
2025	2	6		2	3	0	1	0	0	1	0	1	2	3	5	9	9	10	7	7	7	5	0	7	7	8	5
2025	2	7		1	4	0	1	0	1	0	1	1	1	6	5	9	8	8	15	9	6	7	1	5	4	7	3

4.2 Uygulama Sonuçlarının Değerlendirilmesi

Tez kapsamında oluşturulan dört farklı veri versiyonu (V1–V4) ve çok sayıda tahminleme algoritması kullanılarak gerçekleştirilen modelleme süreçlerinin ardından, bu bölümde elde edilen sonuçların karşılaştırmalı olarak değerlendirilmesi amaçlanmaktadır. Her bir veri yapısı için ayrı ayrı test edilen yöntemler, kendi içinde farklı parametre kombinasyonları ile çalıştırılmış ve bu modellerin performansları hem grafiksel hem de istatistiksel ölçütler üzerinden analiz edilmiştir. Bu analiz süreci, model başarımını anlamak ve hangi yapıların, hangi yöntemlerle daha verimli çalıştığını tespit etmek açısından kritik öneme sahiptir.

Bölümde ilk olarak, her bir veri versiyonu için seçilen modellerin gerçekleştirdiği tahmin sonuçları, zaman ekseninde görsel olarak sunulmaktadır. Bu grafikler sayesinde, modelin gerçek veri ile ne ölçüde örtüştüğü ve hangi dönemlerde sapmalar gösterdiği açık biçimde gözlemlenebilmektedir. Takip eden alt bölümlerde ise her model için hesaplanan hata metrikleri (RMSE, MAE, MAPE vb.) kullanılarak yöntemler arasında nicel karşılaştırmalar yapılmakta, en düşük hata oranlarına ulaşan model-konfigürasyon çiftleri belirlenmektedir. Böylece hem genel başarı düzeyi hem de optimal yapılandırmalar ortaya konulmaktadır.

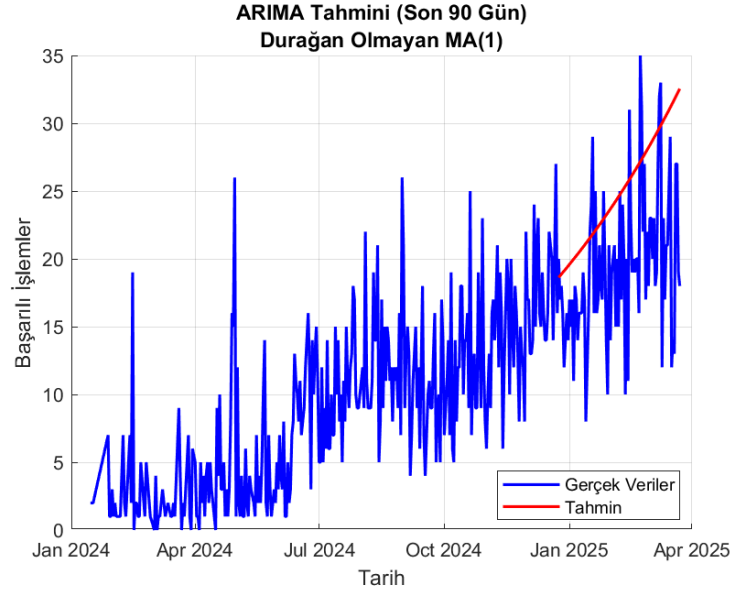
4.2.1 Tahmin Sonuçlarına Ait Grafikler

4.2.1.1 V1 veri seti için MATLAB tabanlı tahmin sonuçları

Bu alt bölümde, ham saatlik verileri içeren V1 veri setine ait MATLAB tabanlı modelleme süreci sonucunda elde edilen tahmin grafiklerinin görsel analizine yer verilmektedir. Her bir model, 90 günlük test dönemi boyunca gerçekleştirdiği tahmin performansına göre grafik üzerinde gerçek verilerle karşılaştırmalı biçimde sunulmuştur. Aşağıda sıralanan her grafik, ilgili modelin veri eğilimlerine ne derece uyum sağladığını ve tahmin kabiliyetini gözler önüne sermektedir.

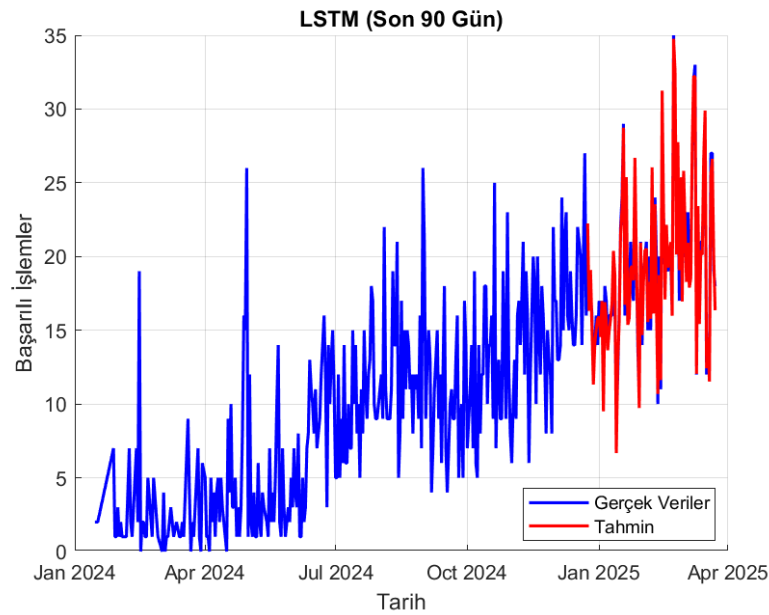
İlk olarak ARIMA modeli üzerinden elde edilen tahmin grafiği incelendiğinde, modelin özellikle verinin genel artış eğilimini takip etmeye çalıştığı görülmektedir. Şekil 4.1'de yer alan görselde, mavi çizgi gerçek saatlik başarılı işlem sayılarını, kırmızı çizgi ise ARIMA modelinin öngördüğü tahminleri temsil etmektedir. Modelin durağan olmayan MA(1) konfigürasyonu ile oluşturulduğu ve test dönemi boyunca genel eğilimi yukarı yönlü

izlediği anlaşılmaktadır. Ancak tahmin değerleri, özellikle dönem sonuna doğru artan sapma eğilimi göstermektedir.



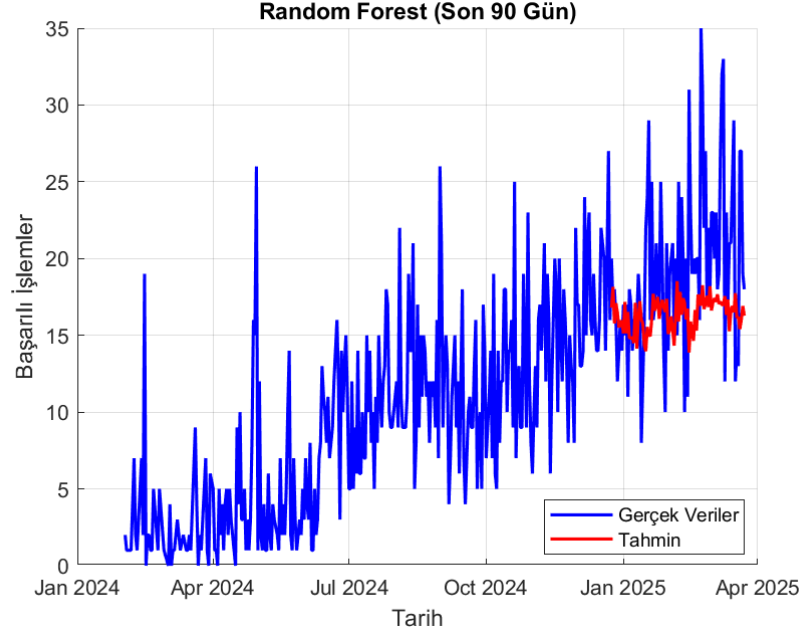
Şekil 4.1 : ARIMA modeli ile V1 veri seti için tahmin grafiği.

İkinci olarak uzun kısa vadeli bellek (LSTM) modeli ile yapılan tahmin sonuçları değerlendirildiğinde, Şekil 4.2’de görüldüğü üzere modelin genel desenleri daha iyi yakalayabildiği gözlemlenmektedir. Kırmızı çizgi, özellikle dalgalı ve ani artış gösteren zaman aralıklarında dahi mavi çizgiyi belirli aralıklarla takip edebilmiş ve bu yönüyle veri setinin kısa vadeli değişkenliğine duyarlılık göstermiştir. Ancak bazı bölümlerde fazlaca tepki verdiği ve aşırı tahmin eğiliminde bulunduğu da dikkati çekmektedir.



Şekil 4.2 : LSTM modeli ile V1 veri seti için tahmin grafiği.

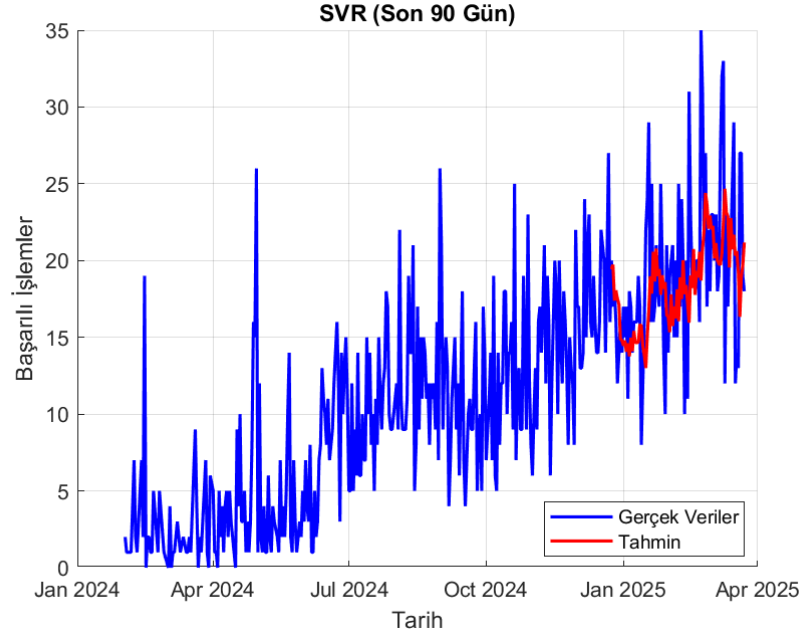
Üçüncü grafik, rastgele ormanlar (Random Forest) algoritmasıyla elde edilmiştir. Şekil 4.3’de sunulan bu grafik incelendiğinde, modelin tahmin çıktılarının oldukça düz bir seyir izlediği ve veri içindeki dalgalanmalara göre daha az tepki verdiği görülmektedir. Bu durum, modelin genel ortalamayı temsil eden bir tahmin stratejisi izlediğini ve kısa dönemli değişimlere karşı daha az duyarlı olduğunu göstermektedir.



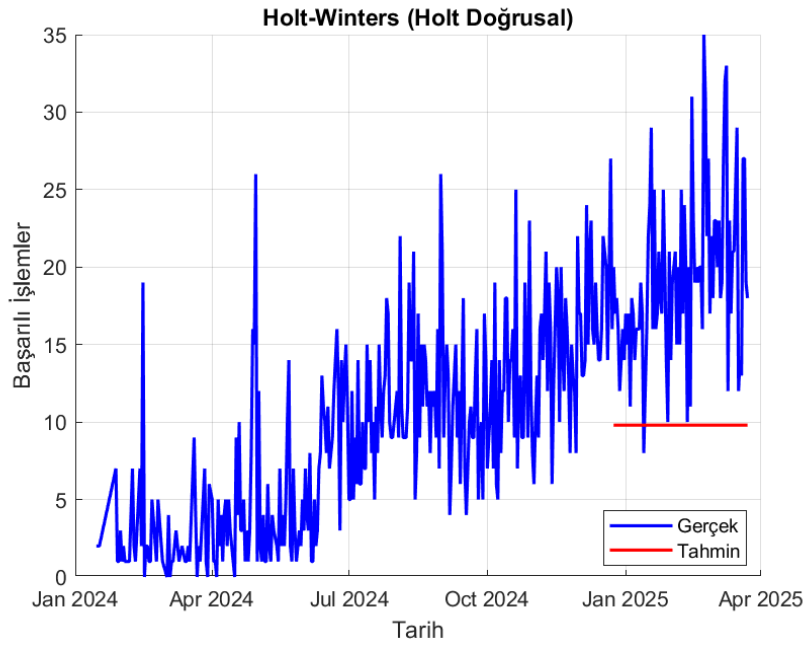
Şekil 4.3 : Random Forest modeli ile V1 veri seti için tahmin grafiği.

Destek vektör regresyonu (SVR) ile yapılan tahmin sonuçları ise Şekil 4.4’te gösterilmektedir. Bu grafikte, kırmızı çizgi gerçek verilerin genel eğilimini oldukça yakın biçimde takip etmekte; ani sıçramalar karşısında ise belli ölçüde gecikmeli tepkiler vermektedir. Özellikle test döneminin ikinci yarısında tahminler ile gerçek veriler arasındaki farkın azaldığı görülmektedir.

Bir diğer model olan Holt-Winters yaklaşımı ile elde edilen tahmin grafiği Şekil 4.5’te sunulmuştur. Bu modelin özellikle Holt doğrusal konfigürasyonunun kullanıldığı bu tahminde, kırmızı çizginin sabit bir düzeyde kaldığı gözlemlenmektedir. Bu durum, modelin eğilimleri yakalamakta zorlandığını ve daha basit yapıda bir yaklaşım sunduğunu göstermektedir. Test döneminde gerçekleşen yükselişleri takip edememesi, bu yöntemin karmaşık desenler içeren veriler için yeterli olmadığını ortaya koymaktadır.

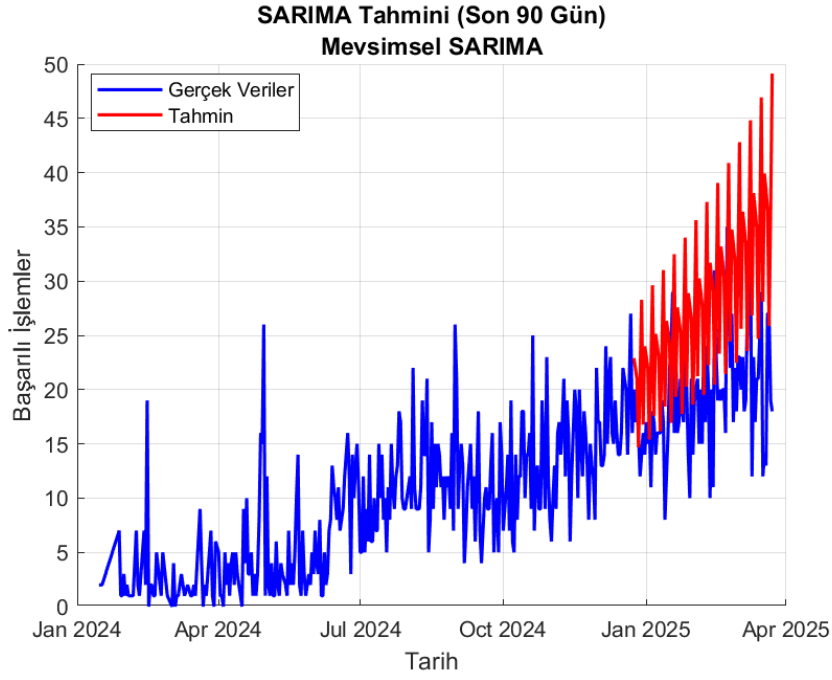


Şekil 4.4 : SVR modeli ile V1 veri seti için tahmin grafiği.



Şekil 4.5 : Holt-Winters modeli ile V1 veri seti için tahmin grafiği.

Son olarak, Şekil 4.6’da mevsimsel bileşenleri içeren SARIMA modeline ait tahmin sonuçları yer almaktadır. Grafik dikkatle incelendiğinde, kırmızı tahmin çizgisinin mevsimsel desenleri taklit etmeye çalıştığı ve genel olarak yükselen trendi yansıttığı görülmektedir. Ancak tahmin eğrisi bazı bölümlerde gerçek değerlerin çok üzerinde kalmakta, bu da modelin bazı senaryolarda fazla iyimser sonuçlar ürettiğini göstermektedir.



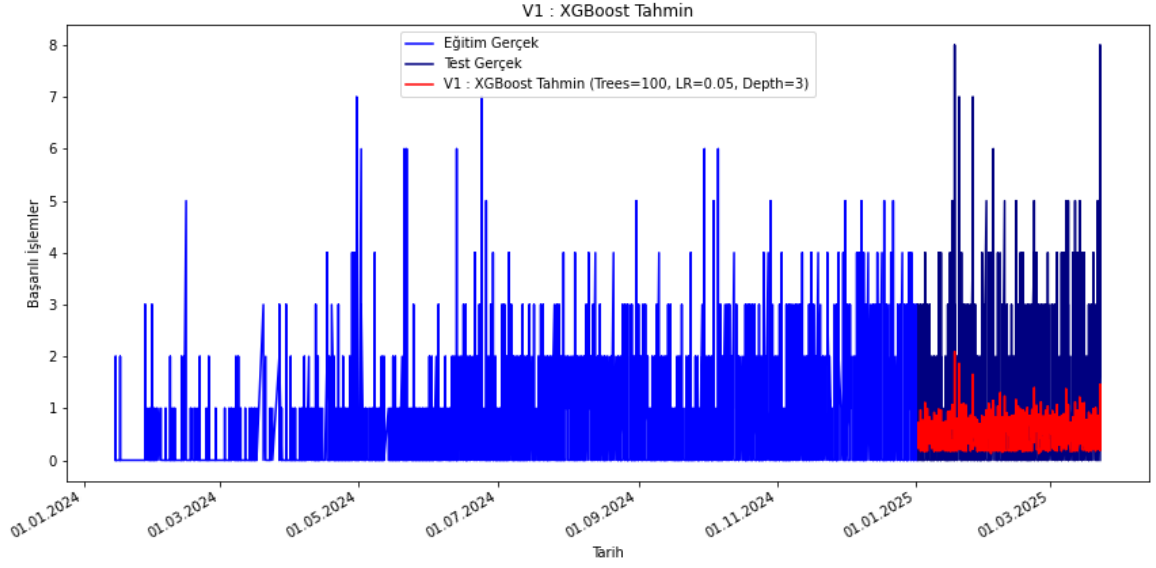
Şekil 4.6 : SARIMA modeli ile V1 veri seti için tahmin grafiği.

4.2.1.2 V1 veri seti için Python tabanlı tahmin sonuçları

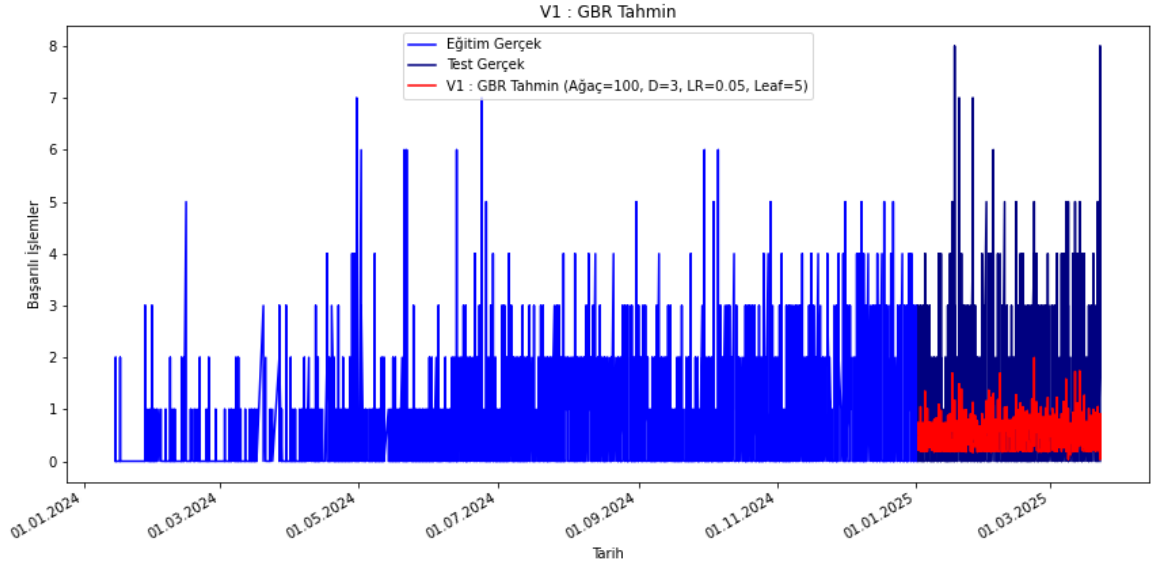
V1 veri setine ait Python tabanlı tahmin uygulamaları kapsamında XGBoost, Gradient Boosting Regressor (GBR), LightGBM ve Prophet modelleri değerlendirmeye alınmıştır. Her bir modelin eğitim ve test dönemlerine ait tahmin başarımı, grafiksel olarak görselleştirilmiş ve model tahminlerinin gerçek verilere olan yakınlığı üzerinden yorumlanmıştır.

İlk olarak XGBoost algoritmasının tahmin performansı ele alınmaktadır. Şekil 4.7’de görüldüğü üzere modelin eğitim verileri üzerinde yüksek doğrulukla sonuçlar üretebildiği, ancak test verileri üzerinde daha sınırlı bir uyum gösterdiği dikkat çekmektedir. Özellikle test dönemindeki düşük değerler, modelin minimum noktalarda yeterince esneklik sağlayamadığını göstermektedir. Kullanılan parametreler grafik başlığında belirtildiği gibi ağaç sayısı 100, öğrenme oranı 0.05 ve derinlik 3 olarak seçilmiştir.

İkinci olarak incelenen model olan Gradient Boosting Regressor (GBR) için Şekil 4.8’e bakıldığında, test verisi üzerindeki tahmin eğrisinin XGBoost’a benzer şekilde, dar bantta kaldığı ve gerçek veriyle sınırlı düzeyde örtüştüğü gözlemlenmektedir. Bu durum, parametrik ayarların (örneğin öğrenme oranı, derinlik, minimum yaprak boyutu gibi) özellikle test seti için yeniden optimizasyon gerektirebileceğini düşündürmektedir. Eğitim verileri ile güçlü uyum sağlanmış olsa da test setinde genellemeye yönelik sınırlamalar gözlenmiştir.

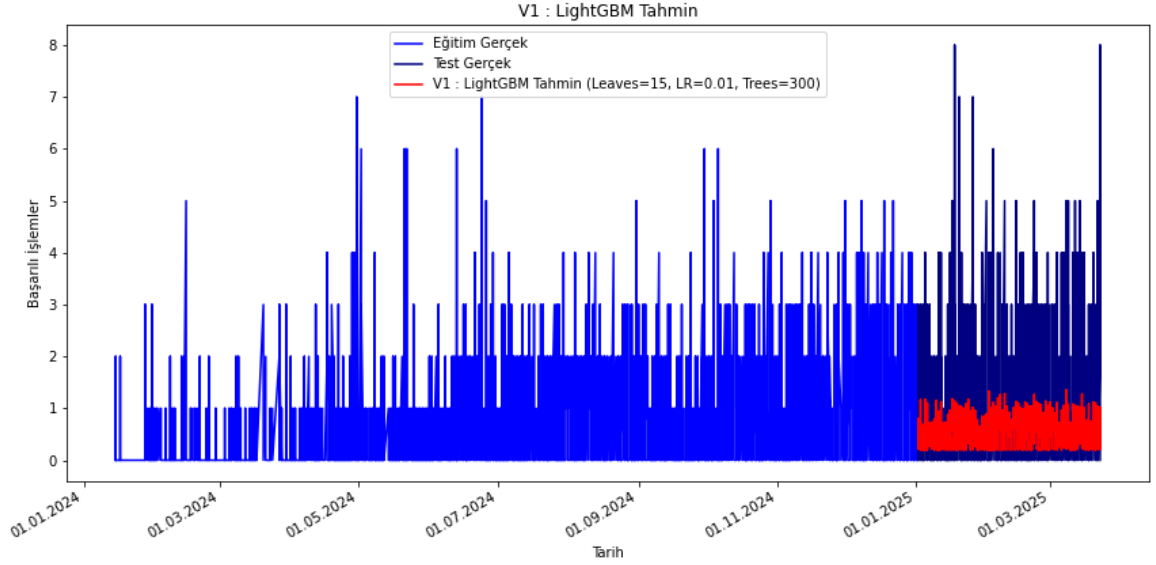


Şekil 4.7 : V1 veri seti için XGBoost modeli tahmin sonuçları.



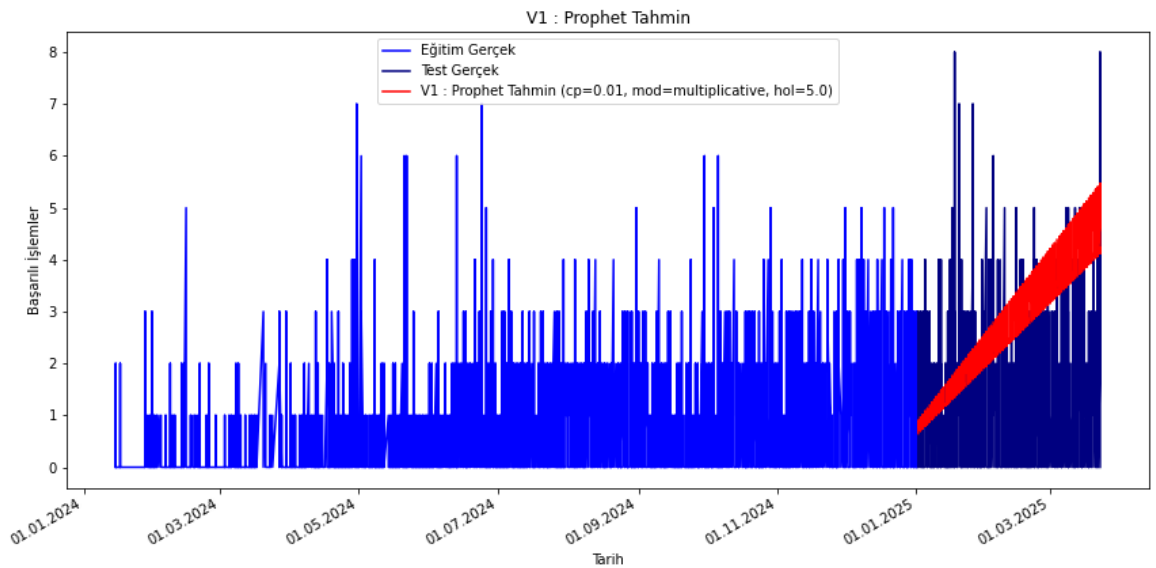
Şekil 4.8 : V1 veri seti için GBR Tahmin Sonuçları

LightGBM modelinin tahmin çıktıları ise Şekil 4.9’da sunulmaktadır. Bu görselde, modelin eğitim sürecinde başarılı bir öğrenme gerçekleştirdiği ve test döneminde ise benzer dar bantta hareket ettiği görülmektedir. Özellikle, tahmin değerlerinin zaman içerisindeki eğilimi yakalayabildiği ancak ani sıçramalar karşısında tepkisinin sınırlı kaldığı anlaşılmaktadır. Bu durum, düşük öğrenme oranı (0.01) ve yüksek ağaç sayısı (300) gibi hiperparametre seçimlerinin test esnekliğini düşürdüğünü göstermektedir.



Şekil 4.9 : V1 veri seti için LightGBM tahmin sonuçları

Son olarak Prophet modeline ait tahmin performansı Şekil 4.10’da gösterilmektedir. Prophet, özellikle trend ve sezonluk bileşenlerin ayrıştırılmasına odaklanan yapısıyla dikkat çekmektedir. Görsel incelendiğinde, test dönemi boyunca tahminlerin ciddi şekilde artış gösterdiği ve bu artışın gerçek verilere oranla daha dik bir ivme içerdiği gözlenmektedir. Bu durum, modelin mevsimsel bileşenleri daha yoğun algılaması ve tahmin eğrisini bu doğrultuda genişletmesinden kaynaklanmaktadır. Ayrıca, Prophet modelinde kullanılan parametrelerin (örneğin değişim noktası esnekliği, sezonsal mod, yıllık periyot) etkisi bu eğilimde doğrudan hissedilmektedir.



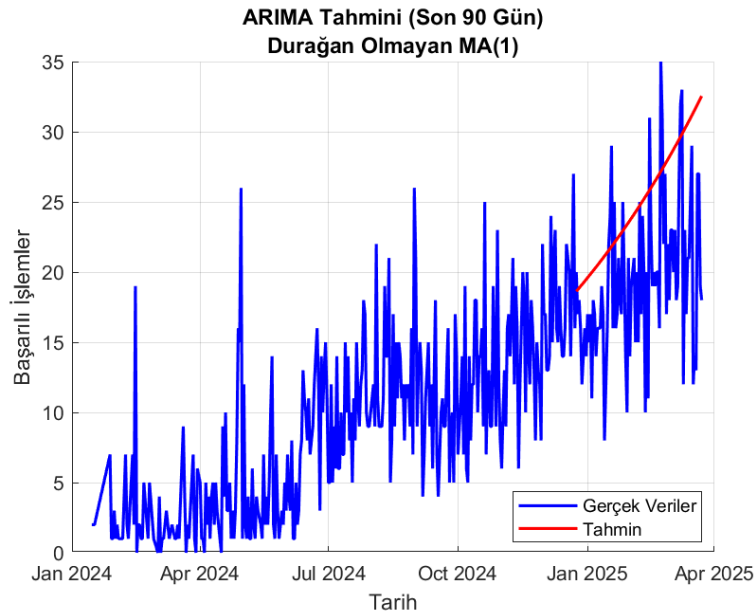
Şekil 4.10 : V1 veri seti için Prophet modeli tahmin sonuçları

4.2.1.3 V2 veri seti için MATLAB tabanlı tahmin sonuçları

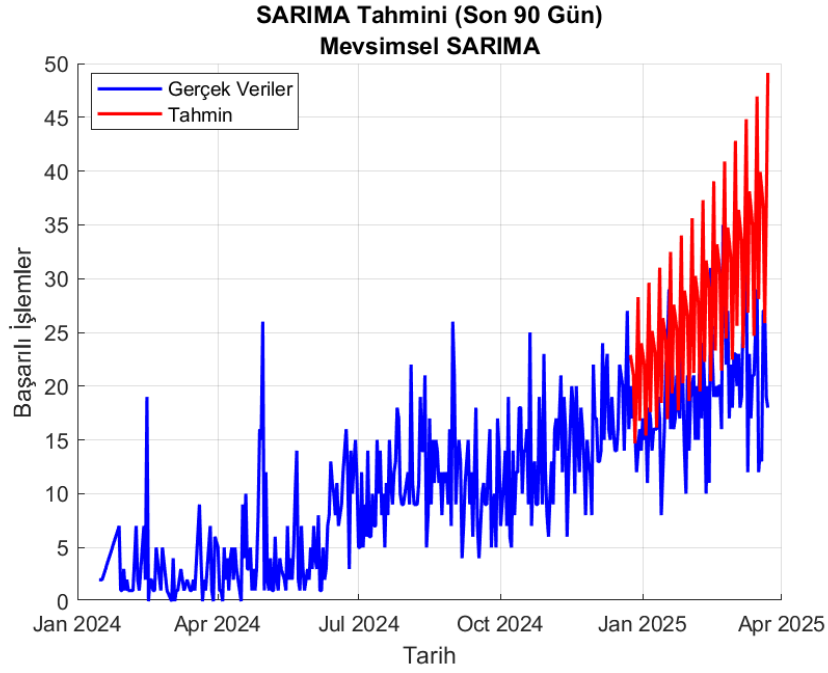
V2 veri seti, elektrikli araç şarj istasyonlarında günlük toplam başarılı işlemleri içeren zaman serilerinden oluşmaktadır. Bu versiyonun yapısı gereği, düşük frekanslı ve daha az örnekleme barındıran bir veri dağılımı söz konusudur. MATLAB ortamında gerçekleştirilen modelleme sürecinde, ARIMA, SARIMA, LSTM, Random Forest ve SVR algoritmaları değerlendirilmiş; her biri için tahmin-gözlem karşılaştırmaları grafiksel olarak raporlanmıştır.

İlk olarak, Şekil 4.11’de yer alan ARIMA modeli sonuçları incelendiğinde, modelin test bölgesine ilişkin tahmin eğrisinin yükselen bir doğrultuda seyrettiği gözlemlenmektedir. Şekil 4.11’de görüldüğü üzere, modelin özellikle durağan olmayan MA(1) konfigürasyonu ile oluşturulmuş tahminleri, gerçek değerlerin üzerindeki eğilimi yansıtmaktadır. Bu sapmanın, ARIMA modelinin trend bileşenine aşırı duyarlılığı sonucu ortaya çıktığı düşünülmektedir.

SARIMA modeline ait tahmin eğrisi ise Şekil 4.12’de sunulmaktadır. Şekil 4.12’de görüldüğü gibi, SARIMA modeli mevsimsel döngüleri modelleme konusunda başarılı olmakla birlikte, tahmin bölgesinde meydana gelen yukarı yönlü sapma dikkat çekmektedir. Modelin parametrelerinde yer alan çarpanlı mevsimsel yapı, geçmiş verilerdeki periyodik artışları agresif biçimde genellemiş ve bu nedenle gerçek değerlerden sapma göstermiştir.

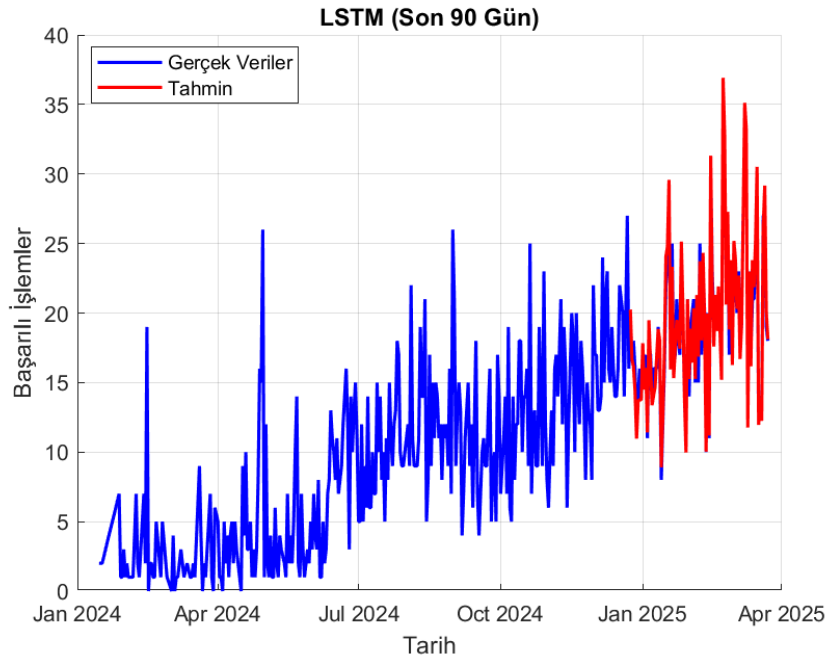


Şekil 4.11 : V2 veri seti üzerinde ARIMA modeli tahmin grafiği.



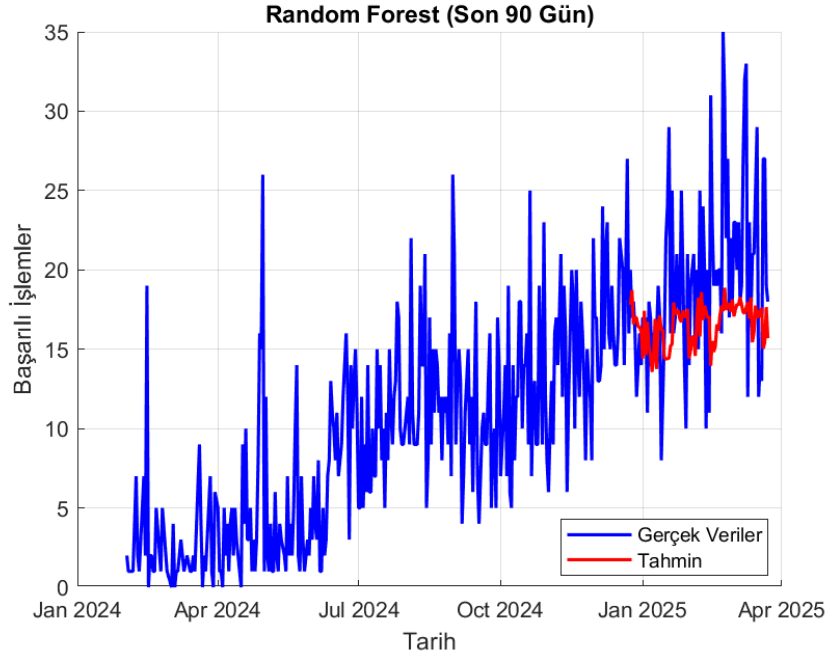
Şekil 4.12 : V2 veri seti üzerinde SARIMA modeli tahmin grafiği.

LSTM modeline ilişkin tahminler Şekil 4.13’de yer almaktadır. Şekil 4.13’de görüldüğü üzere, LSTM modeli veri setinin eğilimlerine duyarlı bir şekilde tahmin üretmiş ve belirli bölgelerdeki dalgalanmaları izleyebilmiştir. Bununla birlikte, bazı yüksek değerli uç noktalarda sapmalar meydana gelmiş, bu durum düşük örneklem sayısı ve LSTM mimarisinin veri içi öğrenmeye olan duyarlılığı ile açıklanabilmektedir.



Şekil 4.13 : V2 veri seti üzerinde LSTM modeli tahmin grafiği.

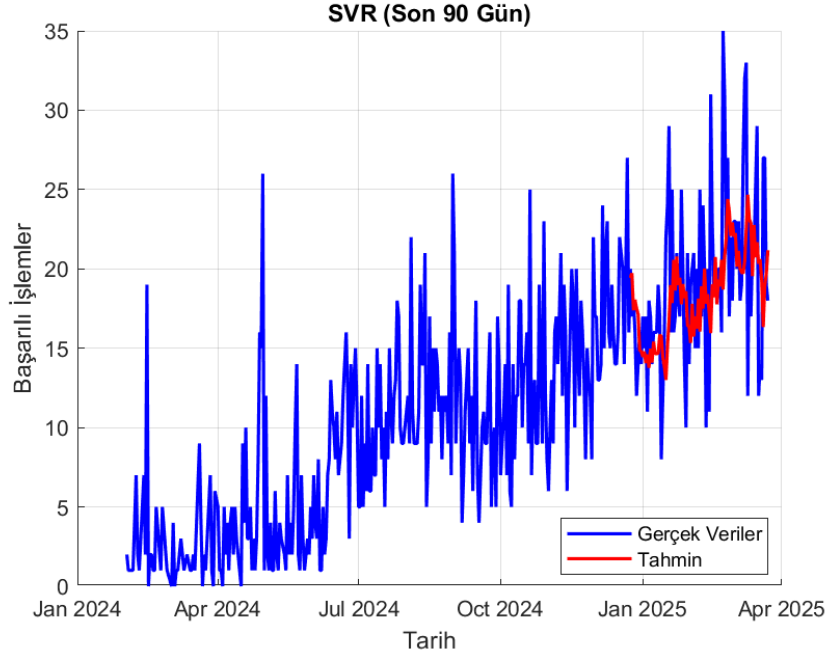
Random Forest algoritmasının tahmin performansı Şekil 4.14’de sunulmuştur. Şekil 4.14 incelendiğinde, modelin genel eğilimle uyumlu tahminler ürettiği fakat tahmin varyansının düşük seviyede kaldığı görülmektedir. Bu durum, Random Forest modelinin yapısal olarak eğilimi dengeleme yönlü çalışmasından kaynaklanmakta, özellikle ani değişimlerin olduğu zaman dilimlerinde başarımı sınırlı kalmaktadır.



Şekil 4.14 : V2 veri seti üzerinde Random Forest modeli tahmin grafiği.

Şekil 4.15’de sunulan SVR modeline ait tahmin eğrisi incelendiğinde, modelin test bölgesindeki değişkenliği makul ölçüde yansıtmayı başardığı gözlemlenmektedir. Şekil 4.15’de de görüldüğü üzere, SVR modeli özellikle negatif yönlü değişimleri izleyebilmiş, ancak pozitif yönlü artışlarda sabit kalma eğilimi göstermiştir. Kernel yapılandırmasının bu sonuç üzerinde etkili olduğu düşünülmektedir.

Bu bölümde yer almayan modellerden biri de Holt-Winters yöntemidir. Söz konusu yöntem her ne kadar zaman serisi tahmini için sıklıkla kullanılan bir yaklaşım olsa da, V2 veri seti üzerinde gerçekleştirilen analizlerde bu yöntemin, özellikle giriş verilerinin yapısı gereği, önceki V1 versiyonunda ürettiği tahmin deseninden anlamlı biçimde ayrılmadığı tespit edilmiştir. Yani, veri yapısındaki değişime rağmen Holt-Winters modelinin verdiği çıktıların önceki grafikte aynı yapıyı sürdürdüğü gözlemlenmiş, bu nedenle gereksiz tekrarın önüne geçmek amacıyla bu modelin grafiksel sunumuna bu bölümde yer verilmemiştir. Ancak bu model de genel performans değerlendirmelerinde sayısal karşılaştırma kapsamında ele alınmaktadır.



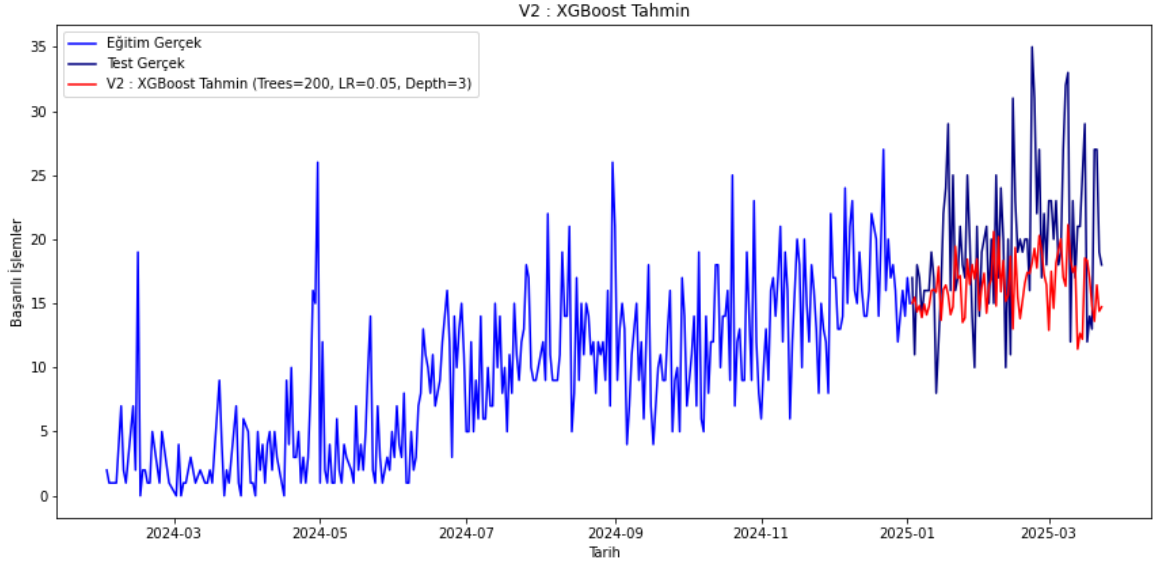
Şekil 4.15 : V2 veri seti üzerinde SVR modeli tahmin grafiği.

Sonuç olarak, V2 veri setine yönelik olarak uygulanan MATLAB tabanlı modelleme sürecinde, özellikle trendi abartan modeller ile varyansı sönümleyen modellerin farklı davranış paternleri sergilediği ortaya konulmuştur. LSTM ve SVR gibi esnek modellerin daha dengeli tahmin eğrileri sunduğu; SARIMA ve ARIMA modellerinin ise yüksek eğilimli tahminler ile sapma eğiliminde olduğu belirlenmiştir.

4.2.1.4 V2 veri seti için Python tabanlı tahmin sonuçları

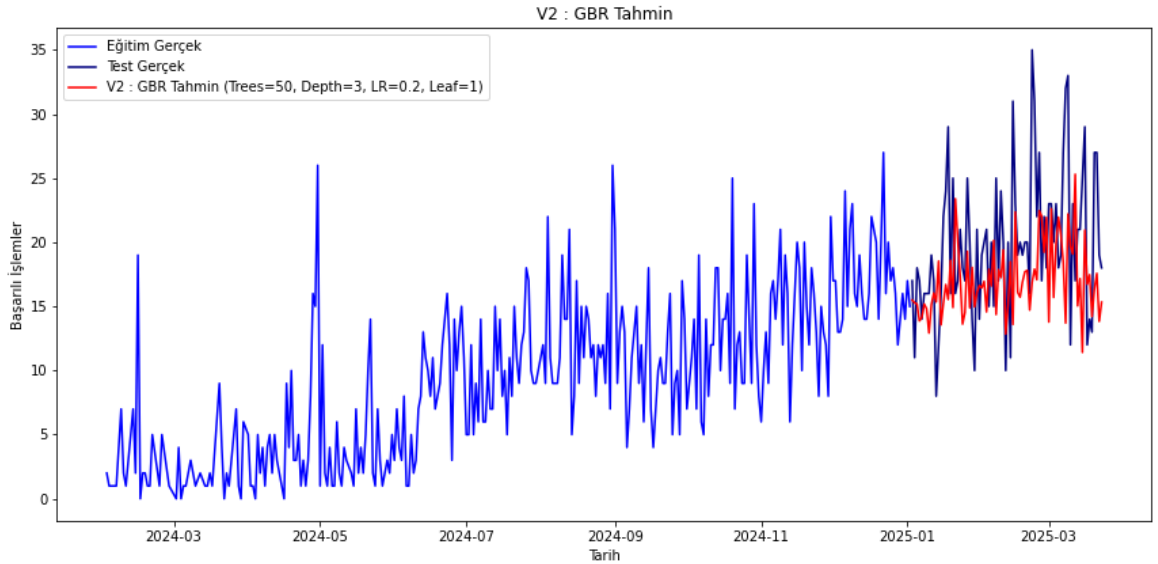
V2 veri seti kapsamında, günlük toplam işlem verileri kullanılarak Python tabanlı dört farklı makine öğrenmesi algoritması ile tahminleme gerçekleştirilmiştir. Bu kapsamda XGBoost, Gradient Boosting Regressor (GBR), LightGBM ve Prophet modelleri eğitilmiş ve elde edilen tahmin çıktıları görsel olarak değerlendirilmiştir.

Şekil 4.16’da görüldüğü üzere, XGBoost algoritması günlük toplam başarılı işlem sayısının tahmininde eğilimleri takip eden fakat kısa vadeli dalgalanmalarda belirgin sapmalar sergileyen bir yapı sunmaktadır. Eğitim dönemindeki veriler ile test dönemine geçiş çizgisi net şekilde ayrılmakta ve model test bölgesinde özellikle ani sıçramaları baskılama eğilimi göstermektedir. Kullanılan parametreler arasında 200 adet ağaç, 0.05 öğrenme oranı ve maksimum derinlik olarak 3 belirlenmiştir.



Şekil 4.16 : V2 veri seti için XGBoost modeline ait tahmin grafiği.

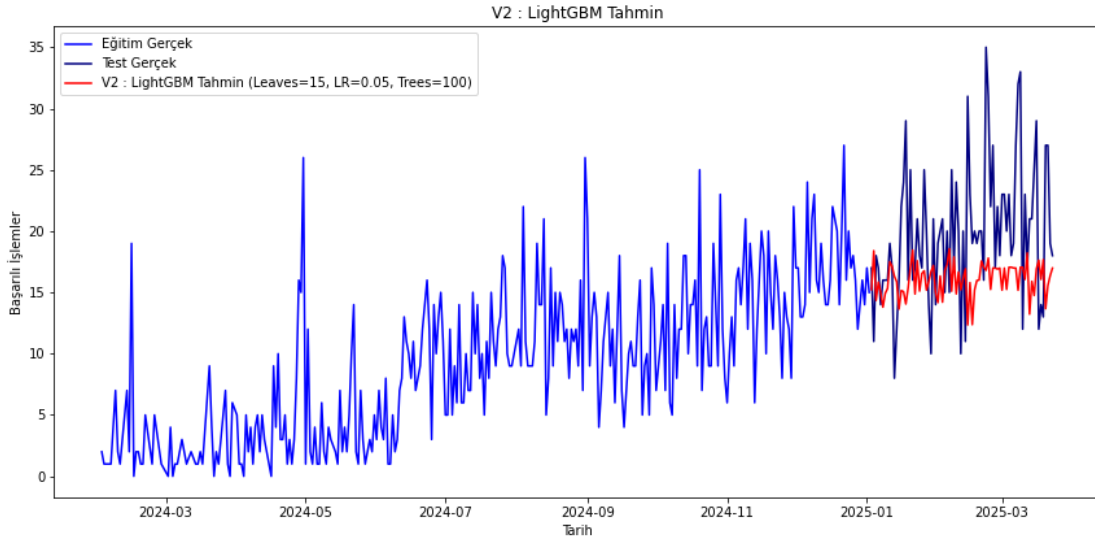
GBR algoritmasının çıktısı Şekil 4.17’de sunulmaktadır. Görsel incelendiğinde, test verilerindeki genel yönelimi başarıyla yakaladığı fakat zaman zaman artan değerleri düzleştirme eğilimi sergilediği gözlemlenmektedir. Bu durum, modelin fazla dalgalanmaları filtreleyerek genellenmiş bir tahmin üretmeye çalıştığını göstermektedir. En iyi sonuç 50 ağaç sayısı, 3 derinlik ve 0.2 öğrenme oranı ile elde edilmiştir.



Şekil 4.17 : V2 veri seti için Gradient Boosting Regressor modeline ait tahmin grafiği.

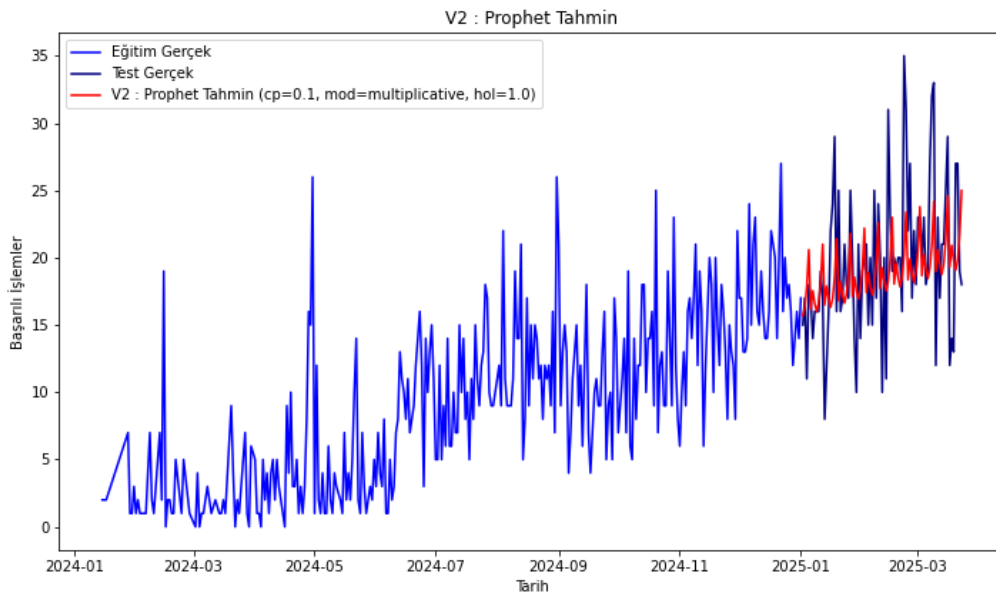
LightGBM algoritmasına ilişkin tahmin sonuçları Şekil 4.18’de yer almaktadır. Modelin eğilim takibi açısından GBR ile benzer bir performans sergilediği, ancak daha düşük sapmalarla daha stabil bir yapı sunduğu görülmektedir. Bu modelde en iyi parametre kombinasyonu olarak 15 yaprak sayısı, 0.05 öğrenme oranı ve 100 ağaç sayısı tespit

edilmiştir. Grafik üzerinde, tahminlerin gerçek verilere göre göreceli olarak daha az dalgalanma gösterdiği ve orta seviye değerlerde yoğunlaştığı dikkati çekmektedir.



Şekil 4.18 : V2 veri seti için LightGBM modeline ait tahmin grafiği.

Son olarak Şekil 4.19’da Prophet modeline ait tahmin görselleştirmesi yer almaktadır. Prophet algoritması, diğer modellere kıyasla test döneminin geneline yayılmış düzgün bir tahmin eğrisi sunmakta; mevsimsellik bileşenlerini içeren yapısı sayesinde dönemsel örüntüleri yakalayabilmektedir. Bu modelde kullanılan parametreler arasında değişim noktası esnekliği (changepoint prior scale) 0.1, çarpımsal mevsimsellik (multiplicative seasonality) ve tatil etkisi parametresi (holidays prior scale) olarak 1.0 belirlenmiştir.



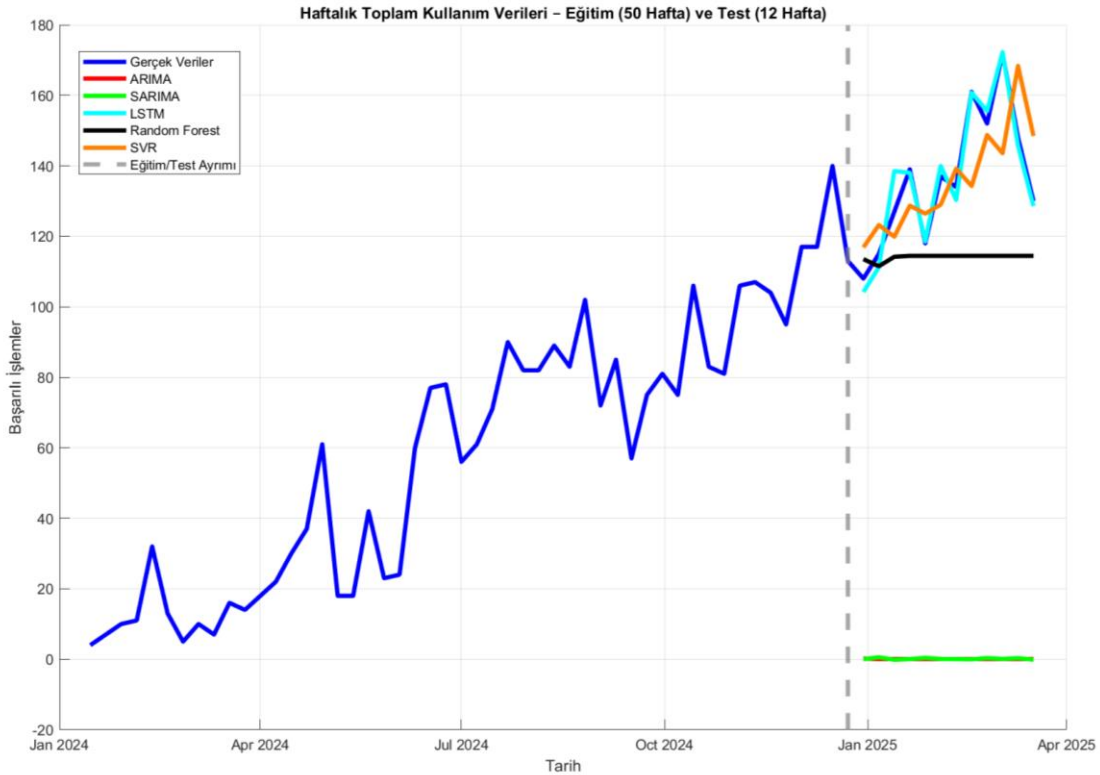
Şekil 4.19 : V2 veri seti için Prophet modeline ait tahmin grafiği.

4.2.1.5 V3 veri seti için MATLAB tabanlı tahmin sonuçları

V3 veri seti, haftalık bazda birleştirilmiş toplam işlem sayılarını içermektedir. Bu yapı doğrultusunda, tahmin süreci tüm modeller için aynı görsel üzerinde gerçekleştirilmiş ve Şekil 4.20’de sunulmuştur. Söz konusu grafik, 50 haftalık eğitim ve 12 haftalık test dönemini kapsamaktadır. Grafik üzerinde, her bir modelin tahmin eğrisi, ilgili algoritmaya özel renklerle ifade edilmiştir.

ARIMA ve SARIMA modelleri, tahmin eğrilerini sırasıyla kırmızı ve yeşil çizgilerle sunmaktadır. ARIMA modeli, genel eğilimi yukarı yönlü devam ettirirken, SARIMA’nın mevsimsel yapıyı yansıtma çabasıyla daha karmaşık bir örüntü sergilediği gözlemlenmektedir. Ancak her iki modelin de, haftalık toplanmış verilerin ani dalgalanma içermemesi nedeniyle, sınırlı bir öngörü kabiliyetine sahip olduğu değerlendirilmiştir.

Grafik üzerinde öne çıkan modellerden biri LSTM algoritmasıdır. LSTM, test döneminde gerçek verilere yakın seyir izleyerek haftalık dalgalanmaları yakalayabilmiş ve artan trende uyum göstermiştir. Maviyle gösterilen gerçek verilerle turkuaz renkli LSTM tahmini arasında test bölgesinde yüksek benzerlik elde edilmiştir. Benzer şekilde, turuncu çizgiyle temsil edilen SVR modeli de gerçek verilere paralel bir eğilim göstermiştir.



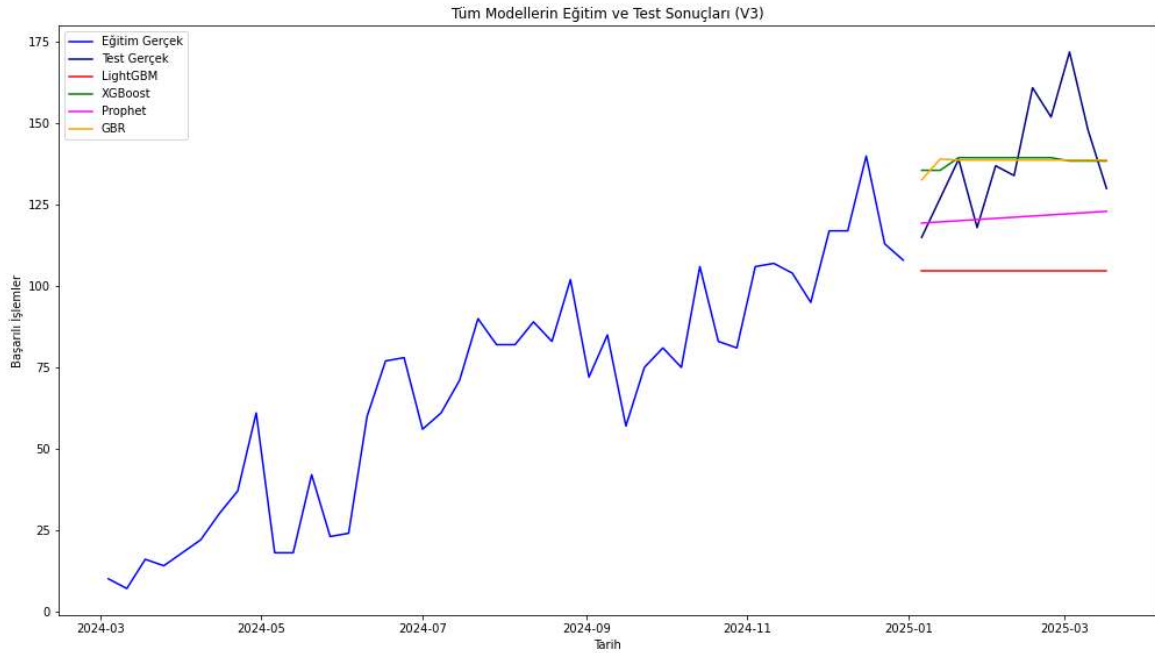
Şekil 4.20 : V3 veri seti için MATLAB ortamında eğitilen modellerin ortak tahmin grafiği

Random Forest yöntemi ise siyah çizgiyle gösterilmekte olup, test döneminde sabit bir tahmin değeri sunmuştur. Bu durum, modelin haftalık bazlı verilerdeki değişkenliği algılamada yetersiz kaldığını ve eğitim döneminde öğrenilen değere sıkı sıkıya bağlı kaldığını göstermektedir.

Bu görselde tüm modellerin bir arada sunulması, modellerin eğilim takibi, dalga yakalama kapasitesi ve genel tahmin performansı açısından karşılaştırmalı değerlendirme yapılmasına imkân tanımaktadır. Elde edilen sonuçlara göre haftalık toplam verilerde en iyi görsel uyum LSTM ve SVR algoritmaları tarafından sağlanmıştır.

4.2.1.6 V3 veri seti için Python tabanlı tahmin sonuçları

V3 veri setine ilişkin Python tabanlı uygulamada, tüm modellerin tahmin sonuçları tek bir grafik üzerinde toplanarak bütüncül bir karşılaştırma yapılması amaçlanmıştır. Şekil 4.21’de sunulan bu grafik, test dönemine ait tahmin performanslarının birbirine göre konumunu görsel olarak ortaya koymaktadır.



Şekil 4.21 : V3 veri seti için Python ortamında eğitilen modellerin ortak tahmin grafiği.

Grafikte mavi çizgi ile gösterilen değerler, eğitim ve test dönemlerine ait gerçek verileri yansıtmaktadır. Diğer renkler ise her bir makine öğrenmesi algoritmasına ait tahmin sonuçlarını temsil etmektedir. LightGBM, XGBoost, Prophet ve Gradient Boosting Regressor (GBR) algoritmaları bu karşılaştırmaya dahil edilmiştir.

Prophet modeli, test döneminin genel ortalamasına yakın bir tahmin üretmekle birlikte, düz çizgisel bir eğri sunmuş ve zamana bağlı değişkenlikleri yansıtmakta sınırlı

kalmıştır. Benzer şekilde, XGBoost algoritması da sabit bir düzeyde kalarak test döneminde veri dalgalanmalarına yeterince uyum sağlayamamıştır. Bu iki modelin tahmin çizgileri, Şekil 4.21’de sırasıyla kırmızı ve mor tonlarda yatay veya neredeyse yatay eğriler olarak görünmektedir.

Buna karşın, LightGBM ve GBR modelleri daha dinamik bir yapı sergileyerek test dönemindeki dalgalanmaları takip edebilmiş ve gerçek verilere daha yakın bir tahmin başarısı göstermiştir. Bu modellerin tahmin çizgileri, gerçek veri eğrisiyle paralellik göstererek görsel uyum açısından avantaj sağlamıştır. Bu durum, LightGBM ve GBR algoritmalarının haftalık toplam verilerdeki değişkenlikleri yakalamada daha etkili olduğunu göstermektedir.

Bu karşılaştırmalı analiz neticesinde, sabit yapılı tahmin çizgileriyle öne çıkan Prophet ve XGBoost modellerinin performanslarının sınırlı kaldığı, buna karşılık GBR ve LightGBM algoritmalarının daha başarılı sonuçlar ürettiği değerlendirilmektedir. İlgili grafik, model seçiminin veri yapısına duyarlılığını net bir biçimde ortaya koymaktadır.

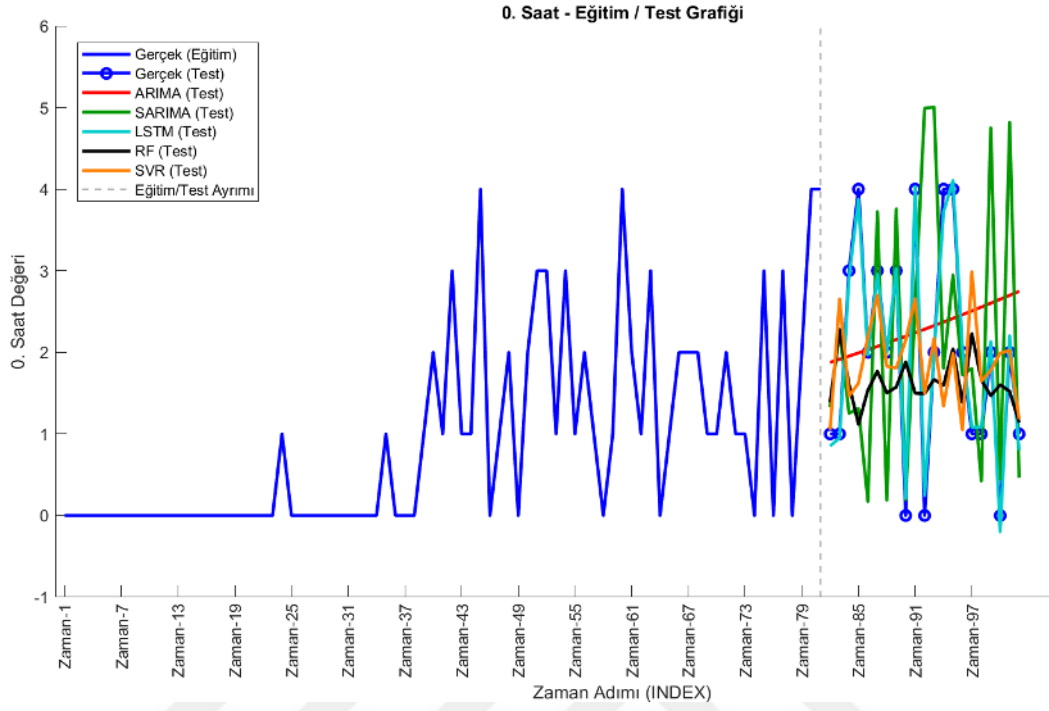
4.2.1.7 V4 veri seti için MATLAB tabanlı tahmin sonuçları

Bu alt başlık altında, V4 veri setine ait ilk altı saate ilişkin MATLAB tabanlı modelleme sonuçları değerlendirilmektedir. Her saat için farklı tahmin modellerinin performansı grafikler üzerinden incelenmiştir. Grafiklerde eğitim ve test ayrımı açıkça belirtilmiş, test dönemine ait tahmin eğrileri model bazlı renklendirilmiştir. Aşağıda her bir saate karşılık gelen analiz ve ilgili şekil bilgisi yer almaktadır.

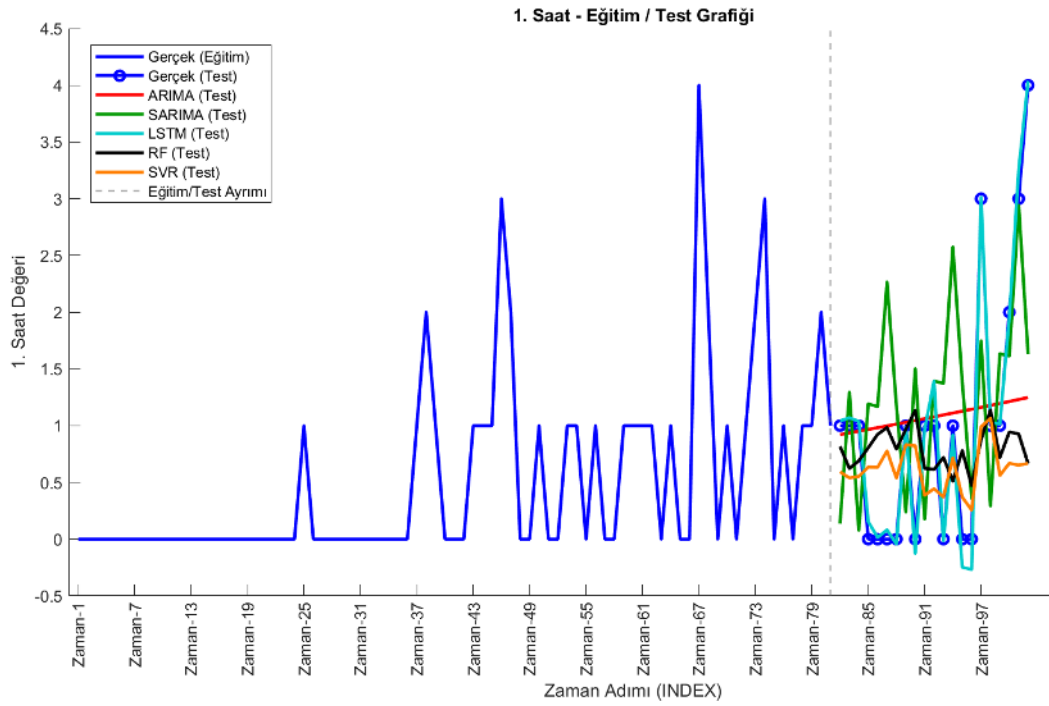
İlk olarak, 0. saat için elde edilen sonuçlar Şekil 4.22’de sunulmaktadır. Bu grafikte LSTM ve SARIMA modellerinin test bölgesinde yüksek varyanslı ancak gerçeğe daha yakın tahminler sunduğu gözlemlenmektedir. Diğer modeller ise genel eğilimleri düzeltirme eğilimindedir.

Ardından gelen 1. saat verilerine ait tahminler Şekil 4.23’de verilmiştir. Bu grafikte, LSTM ve SARIMA’nın özellikle ani artışları ve düşüşleri izleyebilme kabiliyeti dikkat çekmektedir. Ancak bu modeller arasında LSTM’nin daha tutarlı bir tahmin eğrisi sunduğu söylenebilir.

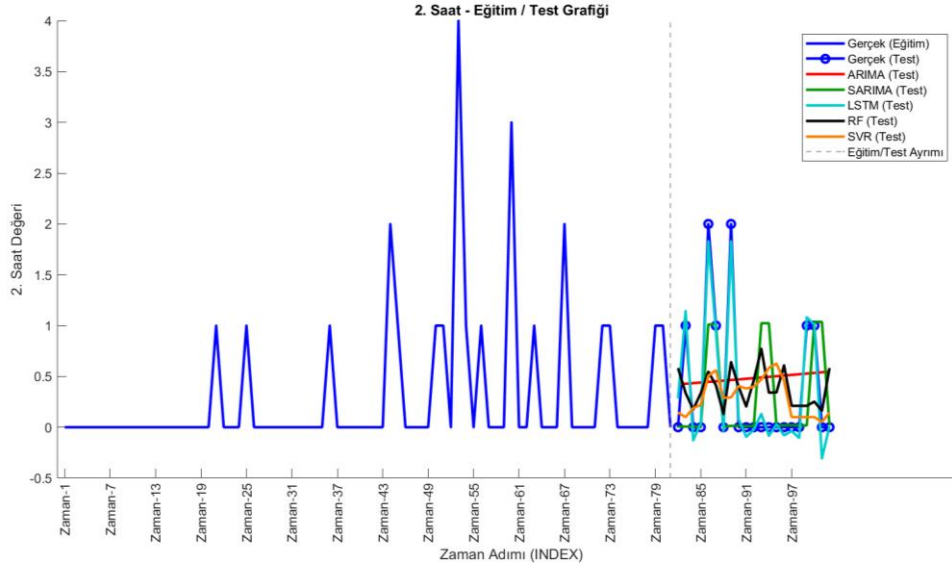
2. saat tahmin sonuçlarının yer aldığı Şekil 4.24 incelendiğinde, veri setinde oldukça düşük değerler bulunduğu, bu durumun da modellerin tepki aralığını daralttığı görülmektedir. SARIMA ve LSTM bu küçük sıçramalara duyarlı kalabilmişken, SVR, RF ve ARIMA’nın düz hatlara sahip olduğu dikkati çekmektedir.



Şekil 4.22 : V4 veri seti için MATLAB tabanlı 0. saat tahmin sonuçları.

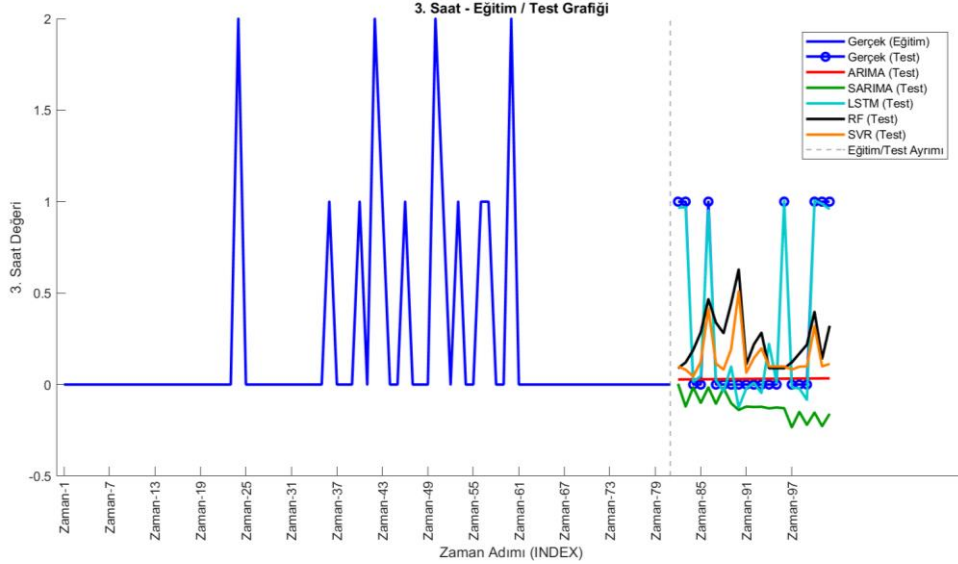


Şekil 4.23 : V4 veri seti için MATLAB tabanlı 1. saat tahmin sonuçları.



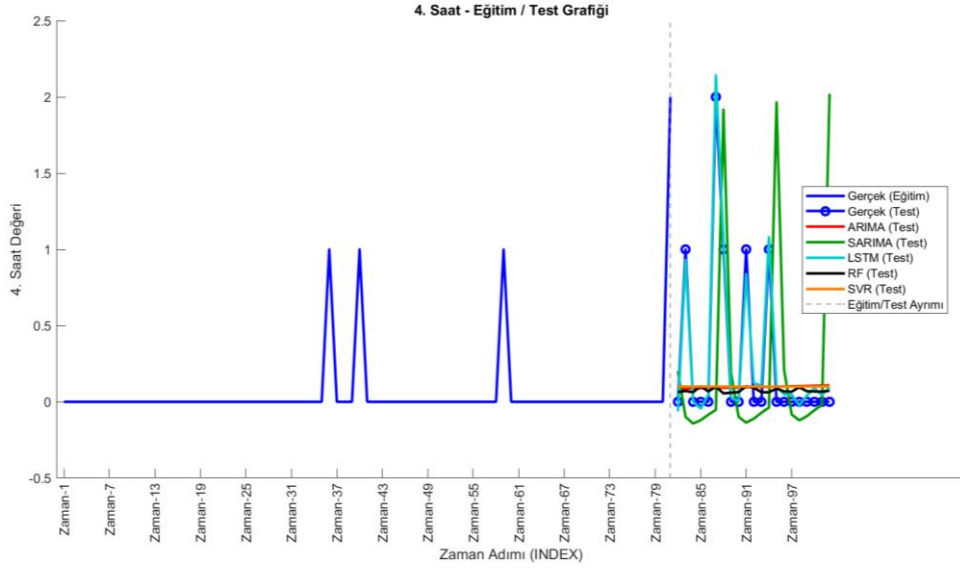
Şekil 4.24 : V4 veri seti için MATLAB tabanlı 2. saat tahmin sonuçları.

3. saat için oluşturulan grafik Şekil 4.25’de sunulmaktadır. Bu saat diliminde LSTM modelinin dalgalanmalara karşı yüksek duyarlılıkla tahminler sunduğu görülürken, RF ve SVR modelleri daha sabit kalmayı tercih etmiştir. SARIMA'nın bu saatte yetersiz kaldığı net şekilde izlenmektedir.



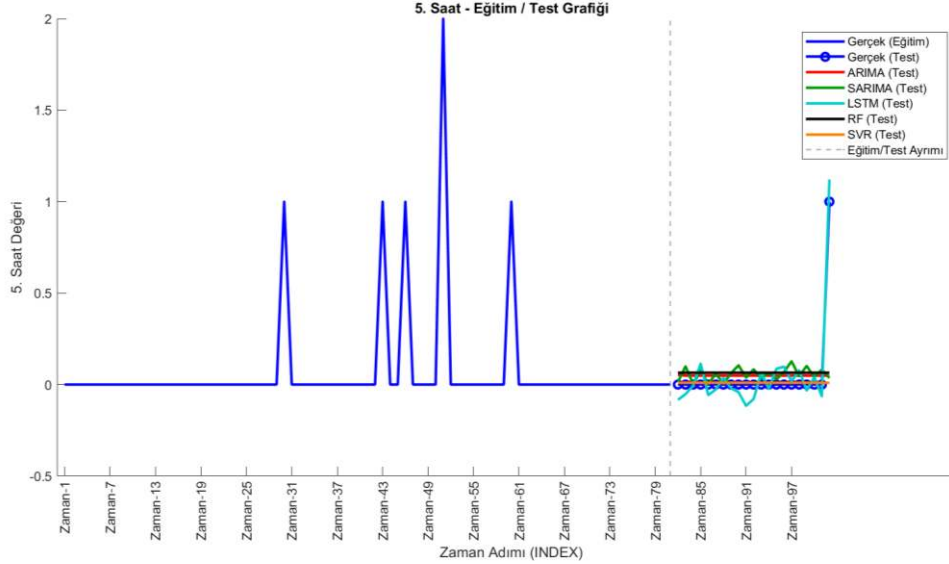
Şekil 4.25 : V4 veri seti için MATLAB tabanlı 3. saat tahmin sonuçları.

Şekil 4.26’da ise 4. saat tahminleri yer almaktadır. Gerçek verideki yüksek frekanslı sıçramalar, SARIMA ve LSTM tarafından karşılanabilmiştir. Özellikle SARIMA'nın bu saatteki aşırı yükselişleri, modelin reaktif yapısını göstermektedir. RF ve SVR yine daha yumuşak geçişli sonuçlar üretmiştir.



Şekil 4.26 : V4 veri seti için MATLAB tabanlı 4. saat tahmin sonuçları.

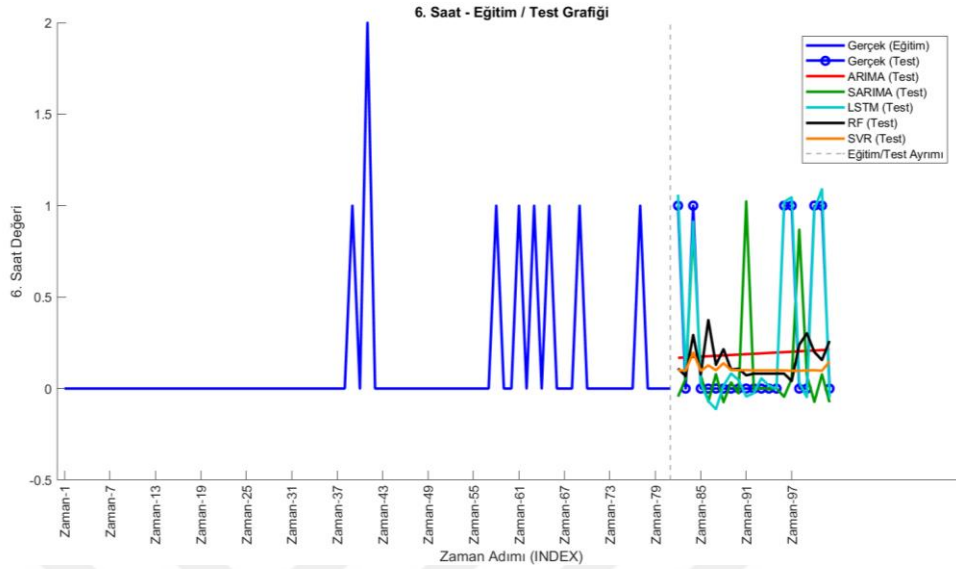
Beşinci saat verileri için tahminler Şekil 4.27’de sunulmuştur. LSTM modeli, test periyodunun sonundaki ani yükselişi başarıyla yansıtabilmiştir. Diğer modellerin bu noktadaki başarısı sınırlı kalmıştır. Özellikle ARIMA ve Prophet gibi modeller bu artışa tepkisiz kalmıştır.



Şekil 4.27 : V4 veri seti için MATLAB tabanlı 5. saat tahmin sonuçları.

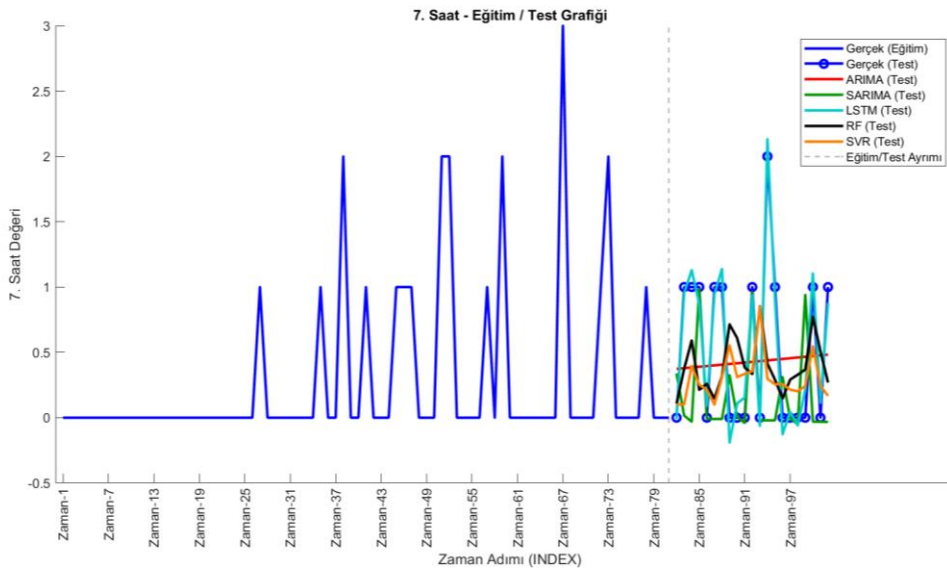
Altıncı saat verileri üzerinden yapılan analizde, özellikle SARIMA ve LSTM yöntemlerinin test dönemindeki dalgalanmalara karşı oldukça duyarlı tepkiler verdiği görülmektedir. Şekil 4.28’de gösterildiği üzere, bu iki yöntem tahmin değerlerinde sık frekanslı ve yüksek genlikli salınımlar sergilemiştir. Buna karşın ARIMA ve SVR yöntemleri ise test süresince daha düz ve düşük varyanslı bir seyir izlemiştir. Random Forest

ise yine yatay doğrultuda sabit kalan bir tahmin profili ile dikkat çekmektedir. Şekil 4.28, 6. saatin eğitim ve test dönemi tahminlerini sunmaktadır.



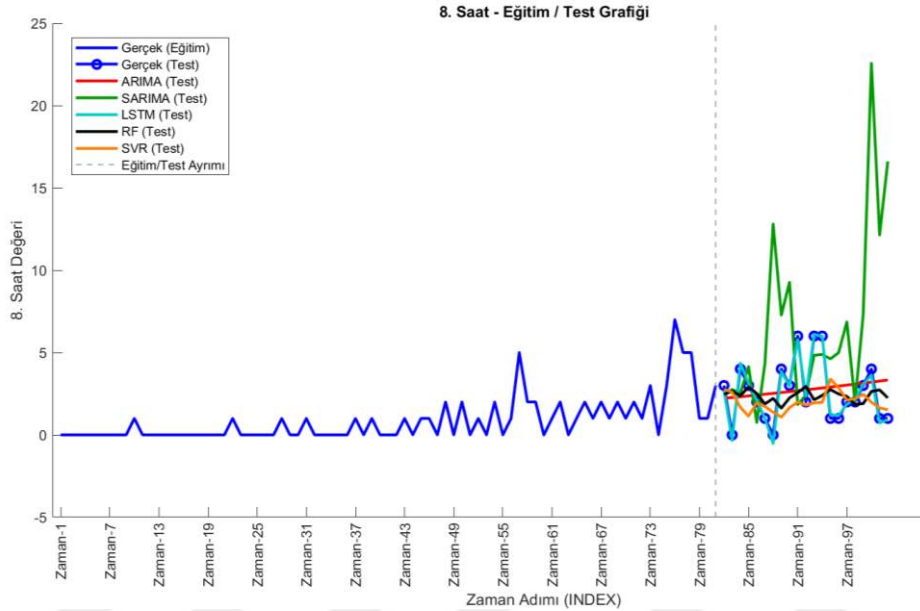
Şekil 4.28 : V4 veri seti için MATLAB tabanlı 6. saat tahmin sonuçları.

Yedinci saat analizinde, test bölgesindeki gerçek değerler sıklıkla sıfıra yakınsamakta olup, sadece birkaç örnekte ani sıçramalar gözlemlenmektedir. Bu ani artışlara en yakın tepkiyi LSTM vermiş, ancak yine de aşırı tepkisel davranışlar sergilemiştir. SARIMA ve RF modelleri ise bu tür sıçramalara daha az duyarlılık göstererek daha yumuşak tahminler üretmiştir. Şekil 4.29’da görüldüğü gibi, ARIMA ve SVR ise genel eğilimi izlemekle birlikte, ani değişimlerde sapmalar göstermiştir.



Şekil 4.29 : V4 veri seti için MATLAB tabanlı 7. saat tahmin sonuçları.

Sekizinci saate ilişkin deęerlendirmede, gerek veri setinin test dneminde ani sıramalar ve dşüşler barındırdığı gözlenmektedir. Bu dalgalanmalara SARIMA ve LSTM modelleri oldukça yüksek frekanslı tepkiler vermiştir. Şekil 4.30’da sunulan grafik, özellikle SARIMA modelinin tahminlerinde aşırı yukarı yönlü tepkiler verdiğini açıka ortaya koymaktadır. Öte yandan SVR, RF ve ARIMA modelleri daha stabil sonuçlar üretmiştir.



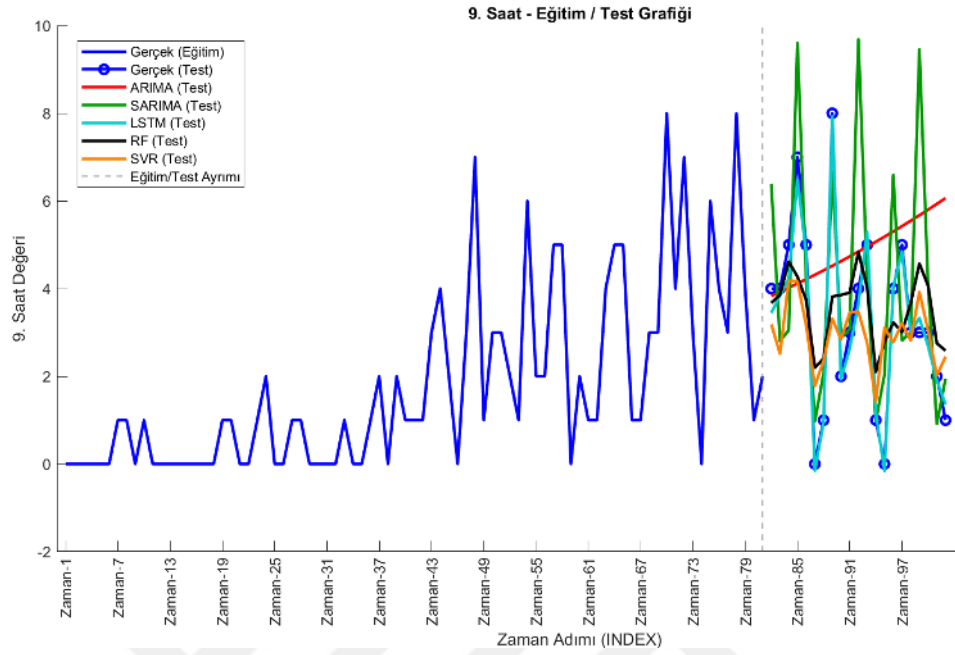
Şekil 4.30 : V4 veri seti için MATLAB tabanlı 8. saat tahmin sonuçları.

Dokuzuncu saat grafiğı incelendiğinde, hem eğitim hem test dönemlerinde yüksek seviyelerde dalgalanmalar olduğu anlaşılmaktadır. Şekil 4.31’de görüldüğü gibi, LSTM ve SARIMA modelleri bu tür deęişimlere en yüksek tepkiyi verirken, SVR ve ARIMA görece daha düz tahmin eğrileriyle ayrışmaktadır. Özellikle LSTM’in test döneminde gerek verilere zaman zaman oldukça yaklaşabildiğı, zaman zaman da aşırı tahminler yaptığı gözlemlenmiştir.

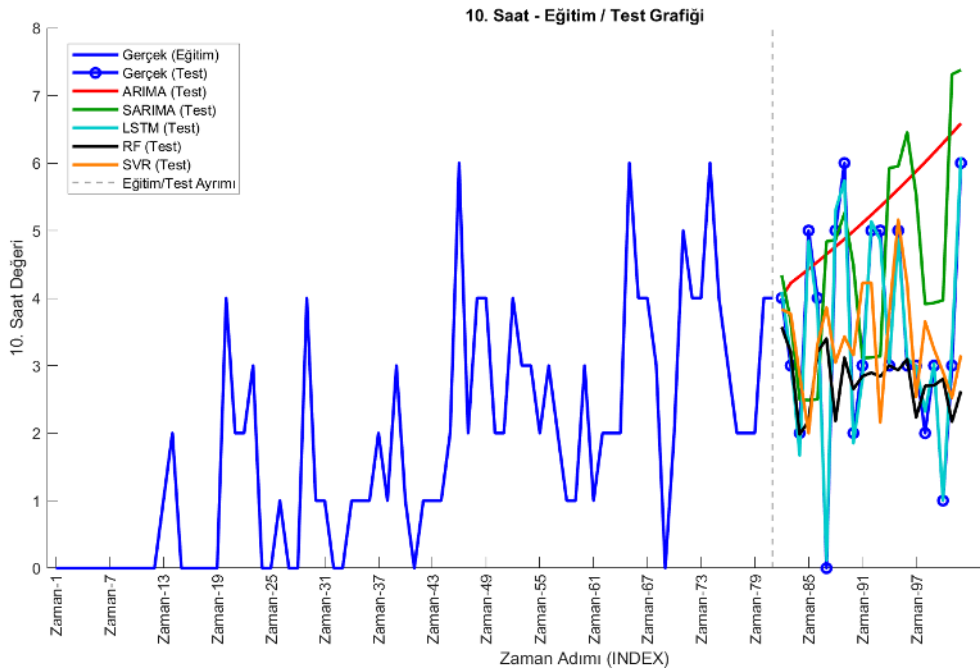
Onuncu saat analizinde, modellerin büyük çoğunluğunun tahmin eğrilerinde gözlemlenen artış eğilimini yakalayabildiğı gözlenmektedir. Şekil 4.32’de görüldüğü üzere, ARIMA modeli yukarı yönlü trendi başarılı şekilde yansıtmış, benzer şekilde LSTM ve SVR modelleri de eğilime yakın tahminlerde bulunmuştur. Random Forest ve Prophet modelleri ise düz ve sabit kalan eğriler ile gereğı yansıtmakta yetersiz kalmıştır.

On birinci saat verilerinde, eğitim seti içerisinde deęişkenlik oldukça fazla iken test bölümünde de benzer oynaklık devam etmektedir. Bu deęişken yapıya SARIMA ve LSTM gibi modellerin tepkileri oldukça sert ve dalgalı olmuştur. Şekil 4.33’de görüldüğü gibi, SVR ve ARIMA modelleri ise bu karmaşık yapıya daha dengeli tahminlerle karşılık vermiştir.

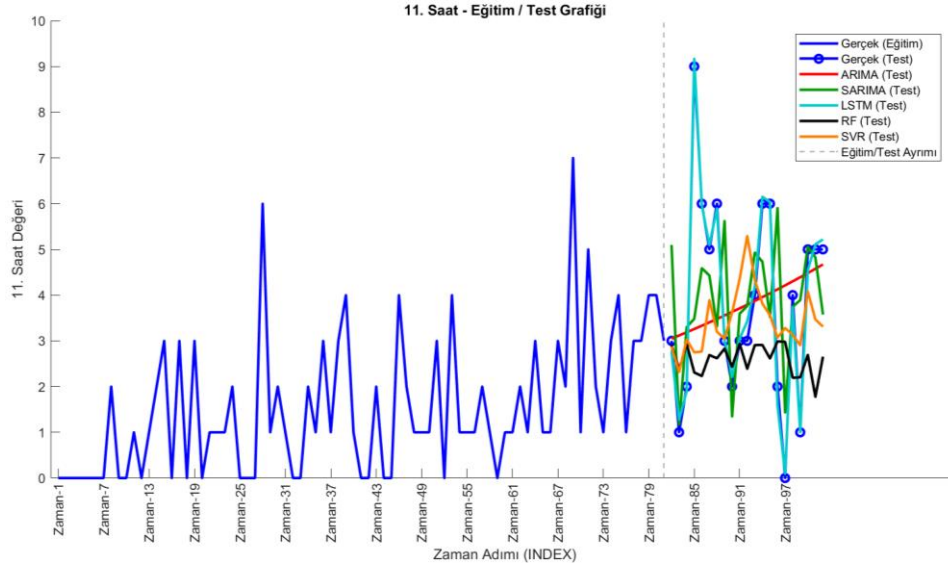
Random Forest'ın yine daha sabit kalan yapısı, değişkenliğe duyarsız kalmasına neden olmuştur.



Şekil 4.31 : V4 veri seti için MATLAB tabanlı 9. saat tahmin sonuçları.

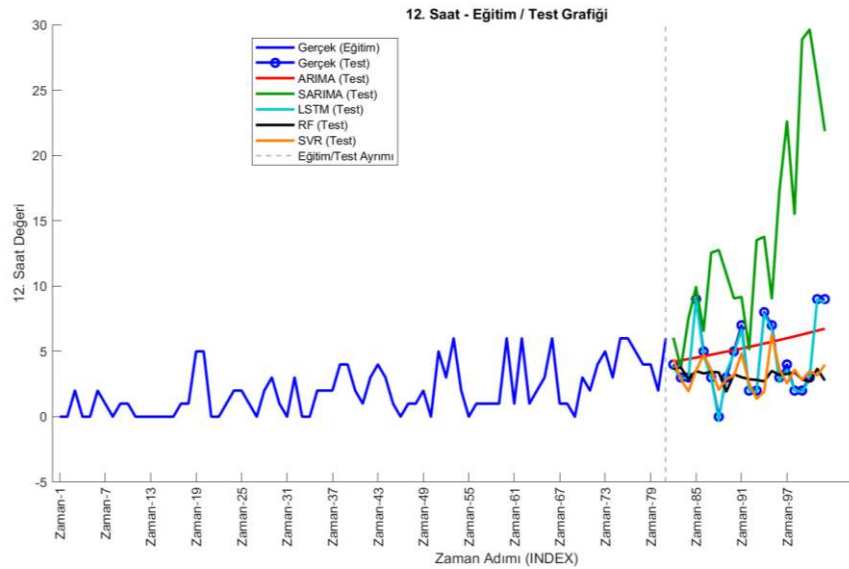


Şekil 4.32 : V4 veri seti için MATLAB tabanlı 10. saat tahmin sonuçları.



Şekil 4.33 : V4 veri seti için MATLAB tabanlı 11. saat tahmin sonuçları

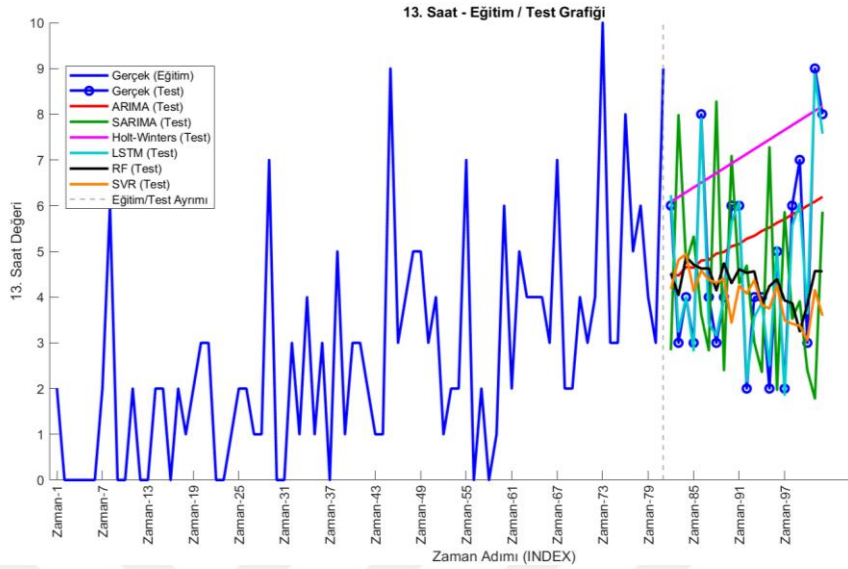
On ikinci saatlik zaman diliminde yapılan tahminlerde, SARIMA modeli belirgin şekilde yüksek değerler üretmiş ve bu durum tahmin eğrisinin gerçeği aşırı yukarıdan takip etmesine neden olmuştur. Diğer yandan, LSTM ve SVR modellerinin tahminleri gerçek değerlere daha yakın seyretmiştir. Random Forest modeli ise düz bir seyir izleyerek daha az dalgalı tahminler üretmiştir. Bu bulgular Şekil 4.34'de açıkça görülmektedir.



Şekil 4.34 : V4 veri seti için MATLAB tabanlı 12. saat tahmin sonuçları.

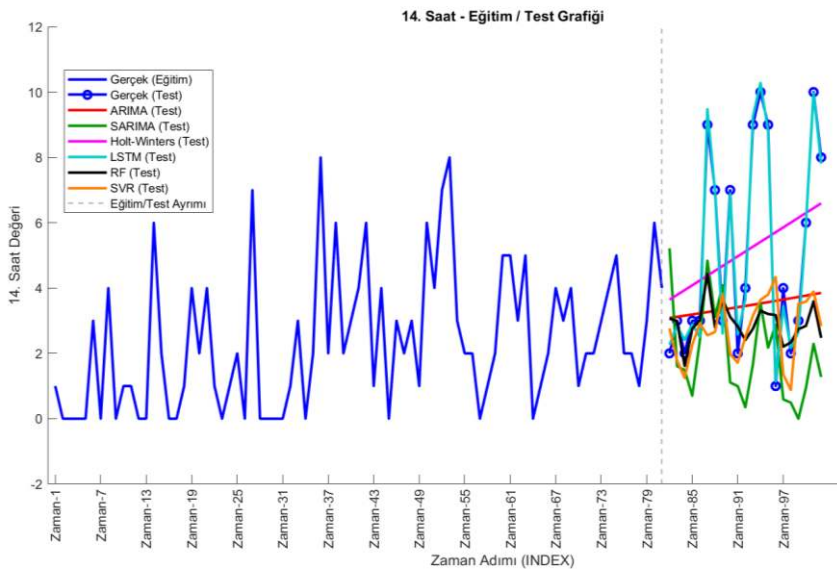
On üçüncü saat için yapılan tahminlerde, Holt-Winters ve LSTM modelleri gerçek verileri yakalayabilen eğilimler sergilemiştir. Özellikle Holt-Winters yöntemi, verideki artışı kademeli ve kararlı bir şekilde takip ederken; LSTM modeli bazı ani artışları daha isabetli

tahmin etmiştir. Diğer modellerin ise özellikle değişkenlik içeren bölgelerde performansları geride kalmıştır. Bu durumu görsel olarak Şekil 4.35’de incelemek mümkündür.



Şekil 4.35 : V4 veri seti için MATLAB tabanlı 13. saat tahmin sonuçları.

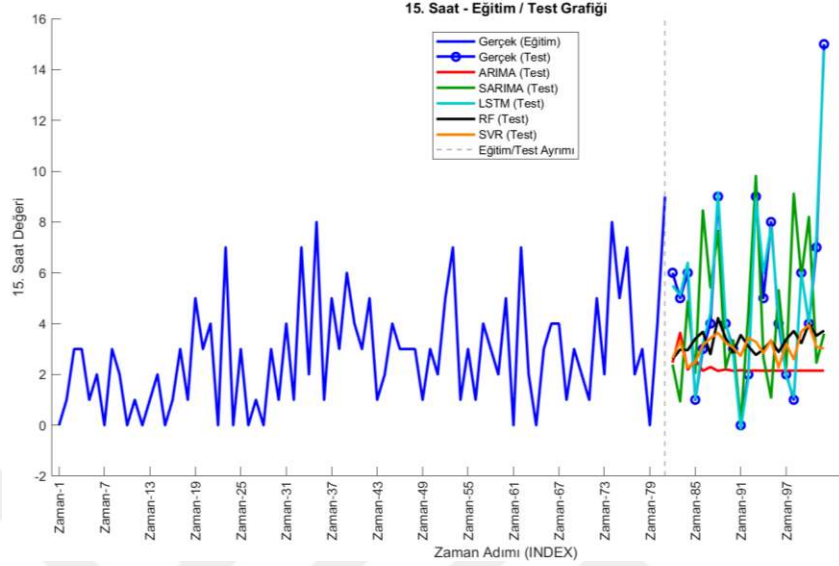
On dördüncü saatlik zaman diliminde, modellerin büyük çoğunluğu genel eğilimi takip etse de SARIMA ve Holt-Winters modellerinin bazı anlarda sapma gösterdiği dikkat çekmektedir. LSTM modeli burada da daha uyumlu bir tahmin grafiği sunarken, SVR ve RF modelleri istikrarlı tahminleriyle öne çıkmıştır. Tüm bu tahmin çizgilerinin karşılaştırmalı analizi Şekil 4.36’da sunulmaktadır.



Şekil 4.36 : V4 veri seti için MATLAB tabanlı 14. saat tahmin sonuçları.

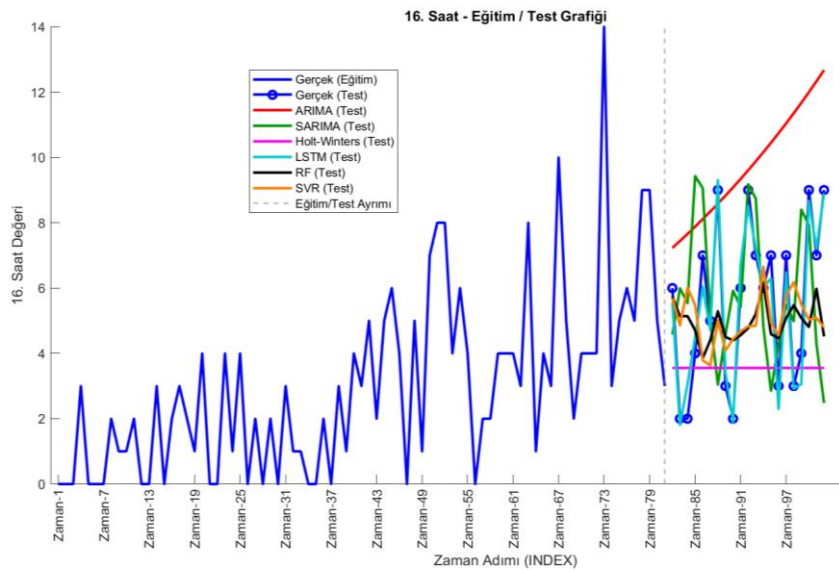
On beşinci saatlik tahminlerde, SARIMA modelinin dalgalı ama yer yer başarılı tahminler sunduğu gözlemlenmiştir. RF, SVR ve LSTM modelleri ise test bölgesindeki

değerleri daha yakından takip etmiş ve başarılı birer eğri üretmiştir. ARIMA ve Holt-Winters modellerinin ise daha durağan ve genelleştirilmiş bir eğilim gösterdiği fark edilmektedir. Bu karşılaştırmalar Şekil 4.37’de yer almaktadır.



Şekil 4.37 : V4 veri seti için MATLAB tabanlı 15. saat tahmin sonuçları.

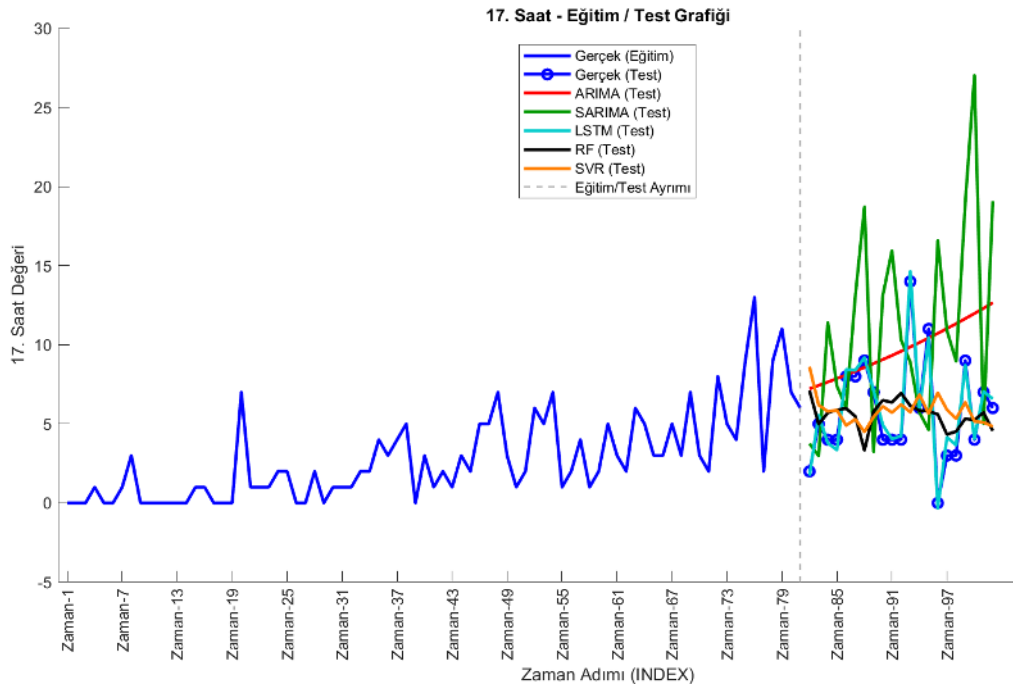
On altıncı saatlik zaman diliminde Holt-Winters modeli dahil olmak üzere tüm yöntemler karşılaştırıldığında, ARIMA modelinin doğrusal ve yukarı yönlü tahmin eğrisi dikkat çekmektedir. Holt-Winters modeli ise düz bir tahmin çizgisiyle veri değişkenliğine yanıt verememiştir. RF ve SVR modelleri dengeli bir tahmin sunarken, LSTM ve SARIMA daha dalgalı ancak yer yer doğru tahminler sunabilmiştir. Şekil 4.38 bu değerlendirmeleri desteklemektedir.



Şekil 4.38 : V4 veri seti için MATLAB tabanlı 16. saat tahmin sonuçları.

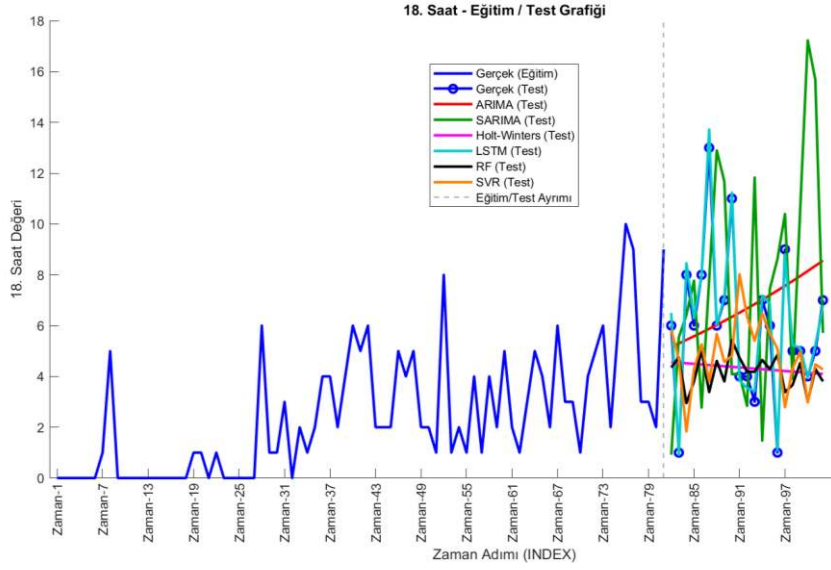
On yedinci saatlik zaman diliminde yapılan tahminlerde SARIMA modelinin keskin dalgalanmalarla diğer modellerden ayrıldığı gözlemlenmiştir. ARIMA modeli yine doğrusal bir artış göstermiş; RF, SVR ve LSTM modelleri ise test bölgesindeki gerçek değerlere yakın ve kararlı eğriler üretmiştir. Özellikle LSTM'in ani yükselişleri takip edebilme başarısı dikkat çekicidir. Tüm bu modellerin performansı Şekil 4.39'da detaylı şekilde görülmektedir.

On sekizinci saat diliminde yapılan tahmin sonuçları Şekil 4.40'da sunulmaktadır. Bu grafikte özellikle SARIMA ve LSTM modelleri test verilerine yakın bir eğilim gösterirken, diğer yöntemlerin daha yatay ve sabit çizgiler sunduğu görülmektedir. SVR ve Random Forest modelleri orta seviyede tahmin başarısı sergilerken, Holt-Winters sabit yapıda kalmış ve ARIMA yukarı yönlü sapmalar göstermiştir.

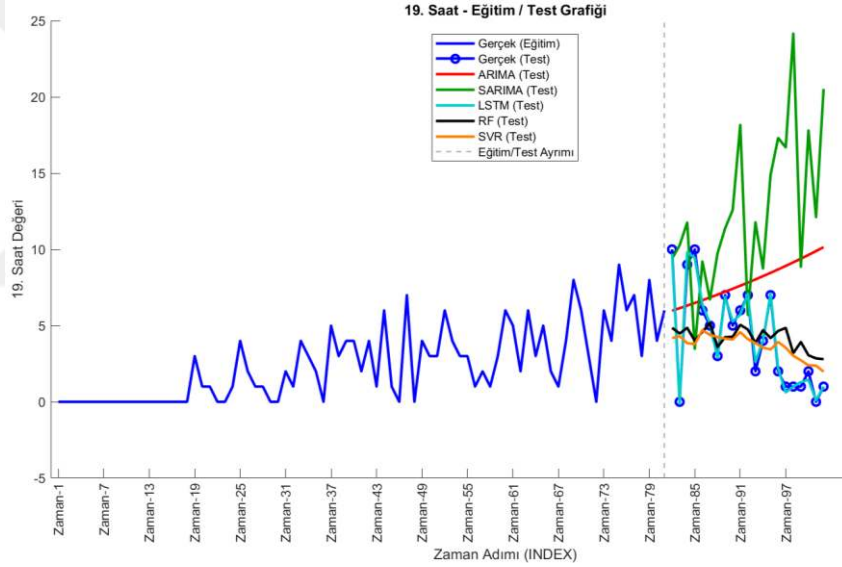


Şekil 4.39 : V4 veri seti için MATLAB tabanlı 17. saat tahmin sonuçları.

On dokuzuncu saat dilimi için oluşturulan tahmin performansları Şekil 4.41'de görülmektedir. Bu grafikte, SARIMA modelinin tahminlerinin test verilerine göre daha yüksek seviyede seyrettiği, ARIMA'nın da benzer biçimde yukarı yönlü tahmin eğiliminde bulunduğu gözlenmektedir. LSTM, RF ve SVR modelleri ise daha dengeli sonuçlar üretmiş ve gerçek test değerlerine daha yakın bir seyir izlemiştir.

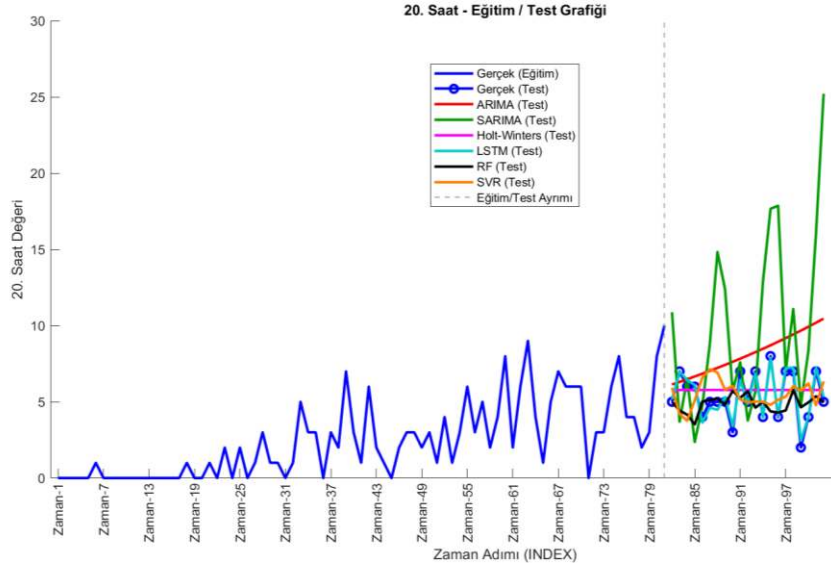


Şekil 4.40 : V4 veri seti için MATLAB tabanlı 18. saat tahmin sonuçları.



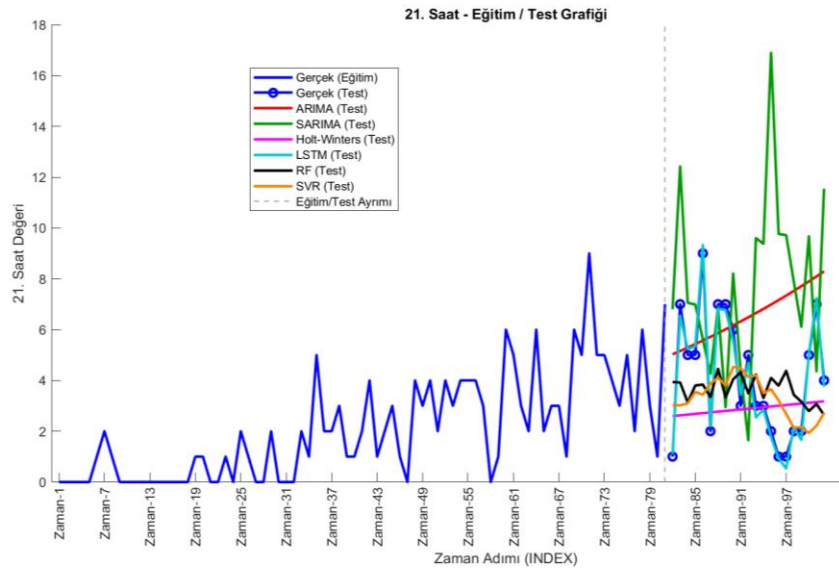
Şekil 4.41 : V4 veri seti için MATLAB tabanlı 19. saat tahmin sonuçları.

Yirminci saate ait tahmin performansları Şekil 4.42’de gösterilmiştir. Bu grafik incelendiğinde, SARIMA modelinin test verilerinden ciddi oranda sapma gösterdiği ve aşırı dalgalanmalar barındırdığı anlaşılmaktadır. Diğer yandan, Holt-Winters, SVR ve Random Forest modelleri oldukça dengeli ve düşük varyanslı tahminler sunmuştur. LSTM ve ARIMA modelleri ise genel eğilimi yakalamakla birlikte bazı noktalarda sapmalar üretmiştir.



Şekil 4.42 : V4 veri seti için MATLAB tabanlı 20. saat tahmin sonuçları.

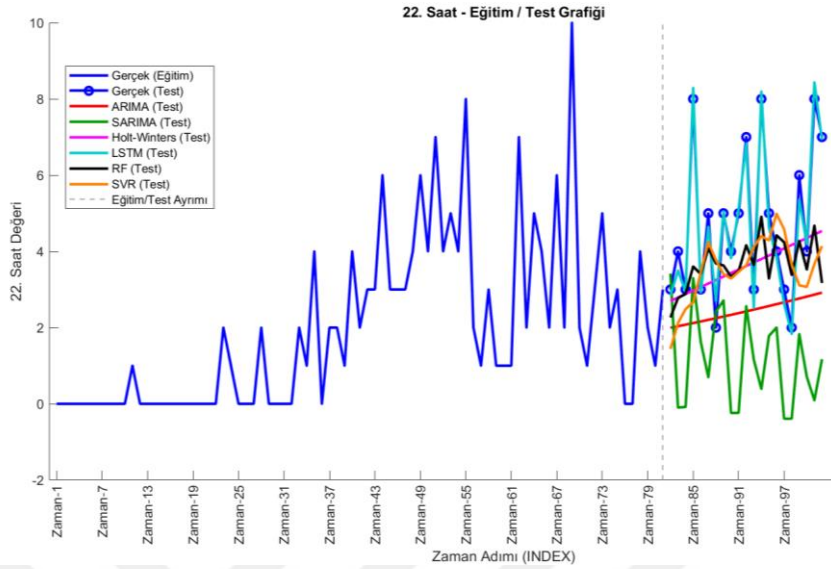
Yirmi birinci saat diliminde elde edilen tahminler Şekil 4.43’de sunulmuştur. Bu grafikte, ARIMA modelinin genel olarak yukarı yönlü bir tahmin yaparken, SARIMA’nın yüksek dalgalanmalı bir yapı sergilediği dikkat çekmektedir. LSTM ve SVR modelleri gerçek test değerleriyle daha tutarlı sonuçlar üretmiştir. Holt-Winters ise sabit düzlemde kalan tahminler üretmiş, bu nedenle eğilimi yansıtmakta yetersiz kalmıştır.



Şekil 4.43 : V4 veri seti için MATLAB tabanlı 21. saat tahmin sonuçları.

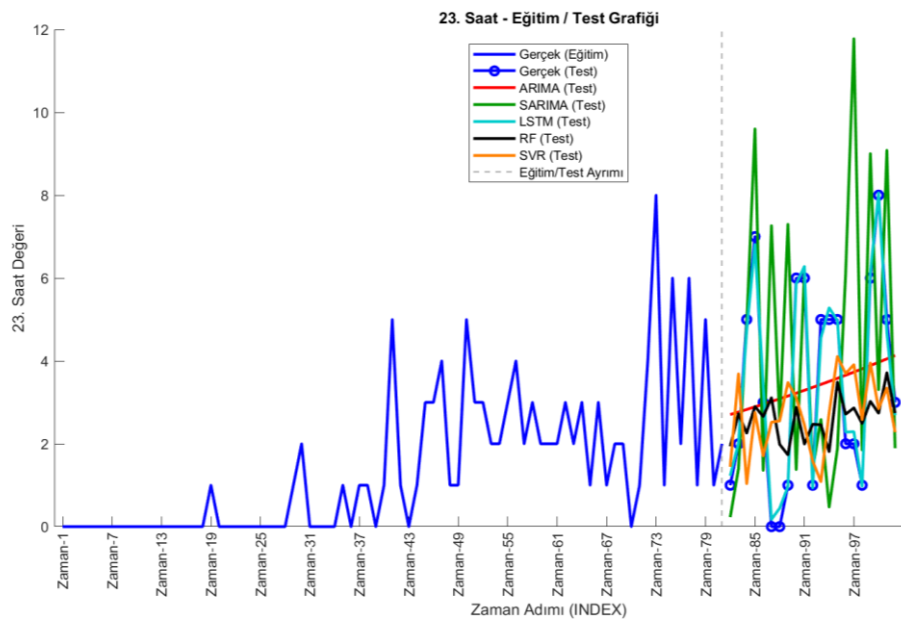
Yirmi ikinci saat tahmin sonuçları Şekil 4.44’de yer almaktadır. SARIMA modelinin aşırı dalgalanmalı yapısı yine öne çıkmakta, buna karşın Holt-Winters ve ARIMA modellerinin tahminleri daha düşük varyanslı görünmektedir. LSTM ve Random Forest

modelleri hem eğilimi hem de test verisine olan yakınlığı açısından başarılı sonuçlar üretmiştir. SVR ise test değerlerinin bir kısmında başarılı tahminler sunmuştur.



Şekil 4.44 : V4 veri seti için MATLAB tabanlı 22. saat tahmin sonuçları.

Yirmi üçüncü ve son saat dilimine ilişkin tahmin sonuçları Şekil 4.45’de gösterilmektedir. Bu grafikte SARIMA modelinin oldukça geniş dalgalanma aralıklarına sahip olduğu, buna karşın ARIMA ve SVR modellerinin nispeten daha stabil eğilim gösterdiği görülmektedir. LSTM ve RF modelleri test değerlerini yakalamada başarılı performans sergilemiştir. Holt-Winters ise çoğu tahmin modeline kıyasla daha sabit ve düşük varyanslı tahminler sunmuştur.



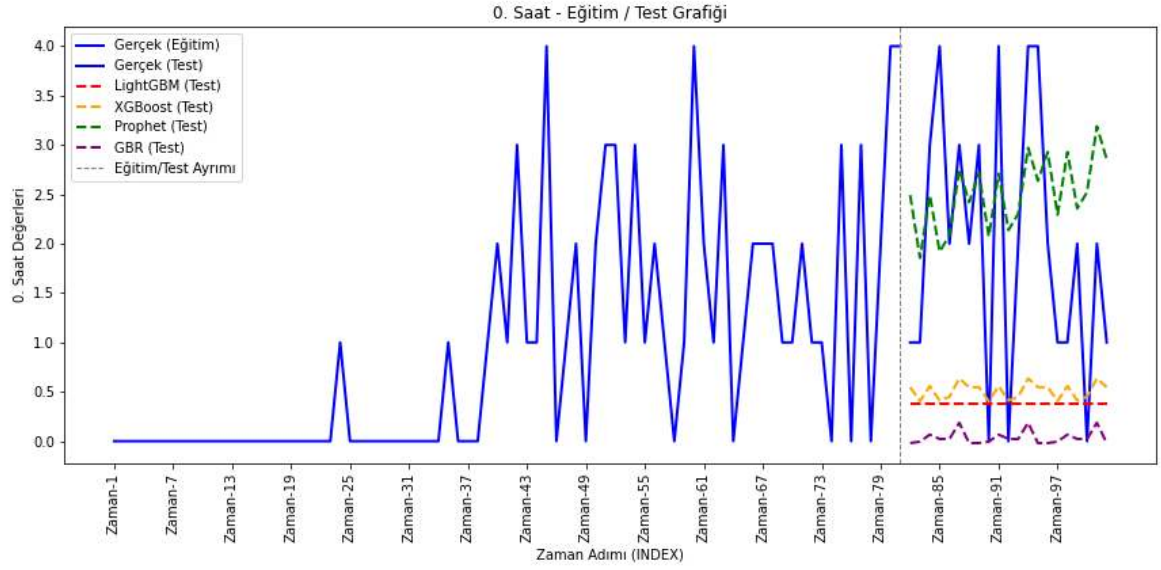
Şekil 4.45 : V4 veri seti için MATLAB tabanlı 23. saat tahmin sonuçları.

MATLAB tabanlı olarak oluşturulan 24 saatlik tahmin grafiklerine yönelik genel değerlendirme yapıldığında, her bir modelin saatlik bazda farklı performans karakteristikleri sergilediği gözlemlenmiştir. Özellikle SARIMA modeli birçok saat diliminde yüksek dalgalanma içeren ve aşırı tahmin sapmalarına neden olan bir yapı sergilemiştir. ARIMA modeli ise genel olarak yükselen eğilimleri yansıtmakta başarılı olsa da, bazı saatlerde gerçek değerlere olan yakınlığını koruyamamıştır. LSTM modeli, genellikle ani değişimlere hızlı yanıt vererek özellikle 9., 13., 16. ve 23. saatlerde dikkat çekici bir uyum göstermiştir. Random Forest (RF) ve Support Vector Regression (SVR) modelleri birçok saat diliminde istikrarlı ve düşük sapmalı tahminler üretmiş, özellikle orta düzeyde aktivite gösteren saatlerde başarılı bir eğilim yakalamıştır. Holt-Winters yöntemi ise sabit yapısı nedeniyle, belirgin artış ya da azalış içeren saatlerde eğilimi yakalayamamakta yetersiz kalmıştır. Bu genel yapı itibarıyla, LSTM, RF ve SVR modelleri zaman serisi içindeki yapısal değişimlere daha uyumlu sonuçlar üretirken, SARIMA ve ARIMA modellerinin saat bazlı öngörü performansları veri yapısına daha duyarlı bir profil çizmiştir. Saatlik analizlerin bu detaylı dağılımı, ileri düzey hiperparametre optimizasyonu ve melez model yapılarının performans etkisini araştırmak açısından da önemli bir temel teşkil etmektedir.

4.2.1.8 V4 veri seti için Python tabanlı tahmin sonuçları

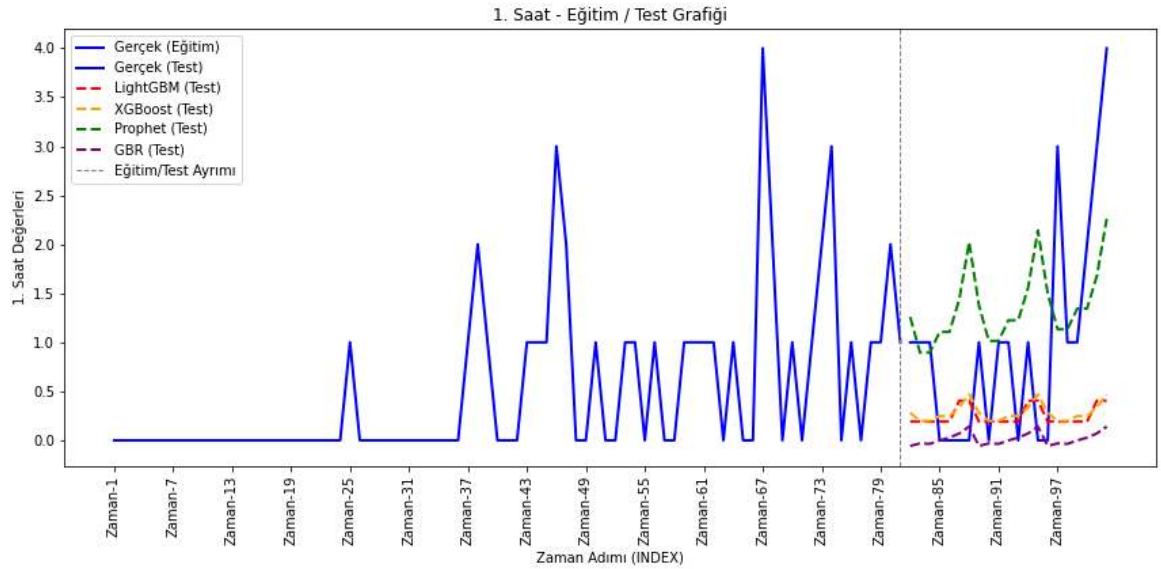
Sıfıncı saat diliminde yapılan tahminlerin değerlendirilmesinde, özellikle Prophet modelinin tahmin eğrisinin test verilerine göre daha yüksek seviyelerde seyrettiği görülmektedir. Bu modelin eğilimi, zaman serisinin artış eğilimini abartılı biçimde yansıtmakta; bu da modelin aşırı tahmin eğiliminde olduğunu göstermektedir. Öte yandan LightGBM ve XGBoost modelleri düz ve istikrarlı bir seyir izleyerek veri setindeki dalgalanmalara tepkisiz kalmıştır. Grafiksel olarak bu durum, Şekil 4.46'da açık biçimde görülmektedir.

Birinci saat aralığında Prophet ve GBR modelleri diğer modellere kıyasla daha fazla dalgalanmayı takip edebilmiş; özellikle test verisinde yer alan ani yükselmeleri GBR zaman zaman başarıyla yansıtmıştır. LightGBM ve XGBoost modelleri bu saat diliminde de düşük ve yatay tahmin eğrileri üretmiştir. Gerçek veri ile örtüşmeyen bu yatay tahminler, bu modellerin düşük varyanslı ve temkinli tahminler yaptığını göstermektedir. Bu gözlemler Şekil 4.47'de yer almaktadır.

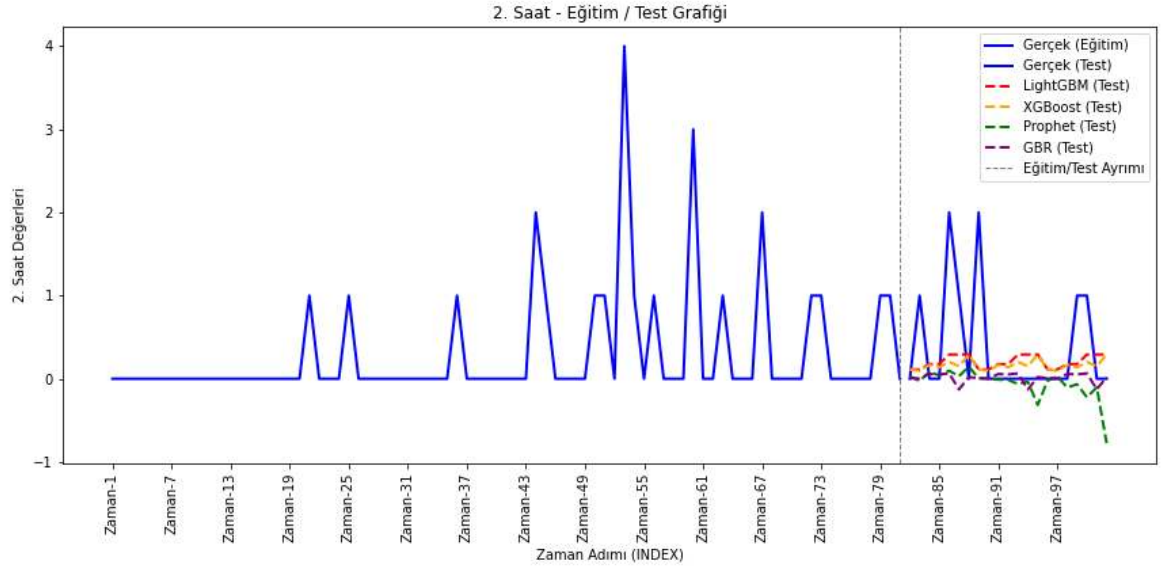


Şekil 4.46 : V4 veri seti için Python tabanlı 0. saat tahmin sonuçları.

İkinci saat diliminde modellerin çoğunun test verilerindeki kısa süreli sıçramalara tepki veremediği görülmektedir. Prophet ve GBR modelleri bu sıçramalara kısmen yanıt verirken, LightGBM ve XGBoost modelleri yatay ve düşük seviyelerde tahminler üretmiştir. Modellerin çoğu, veri setindeki küçük genlikli değişimleri göz ardı etmiştir. Bu durum Şekil 4.48’de detaylı şekilde incelenebilir.

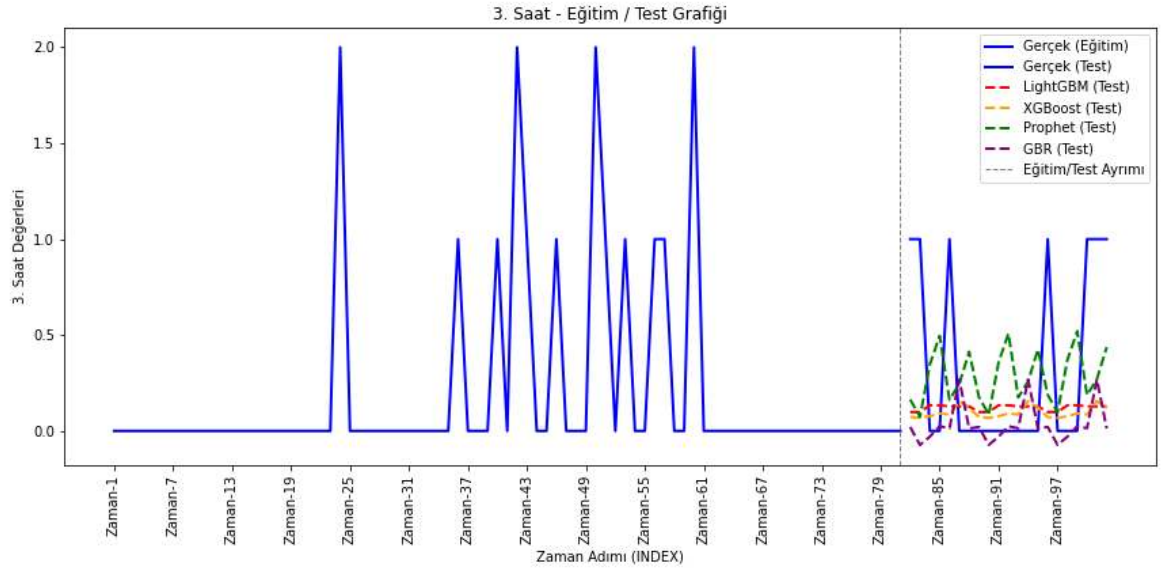


Şekil 4.47 : V4 veri seti için Python tabanlı 1. saat tahmin sonuçları.



Şekil 4.48 : V4 veri seti için Python tabanlı 2. saat tahmin sonuçları.

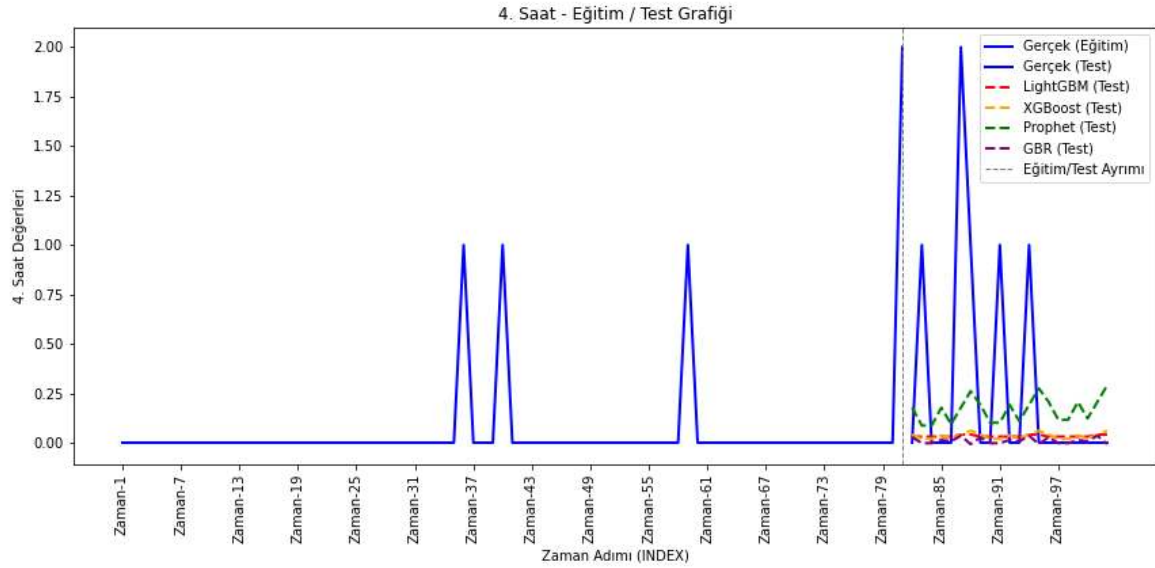
Üçüncü saate ait tahmin grafikleri, modeller arasında görece benzerlik göstermektedir. Prophet ve GBR modelleri daha belirgin varyanslı tahmin eğrileri sergilemiş; LightGBM ve XGBoost ise düşük seviye ve sabit tahminler üretmeye devam etmiştir. Gerçek veri setindeki anlık değişimler tam olarak yakalanamamış, özellikle test döneminde Prophet modelinin yukarı yönlü sapmaları dikkat çekmiştir. İlgili tahmin eğrileri Şekil 4.49’da yer almaktadır.



Şekil 4.49 : V4 veri seti için Python tabanlı 3. saat tahmin sonuçları.

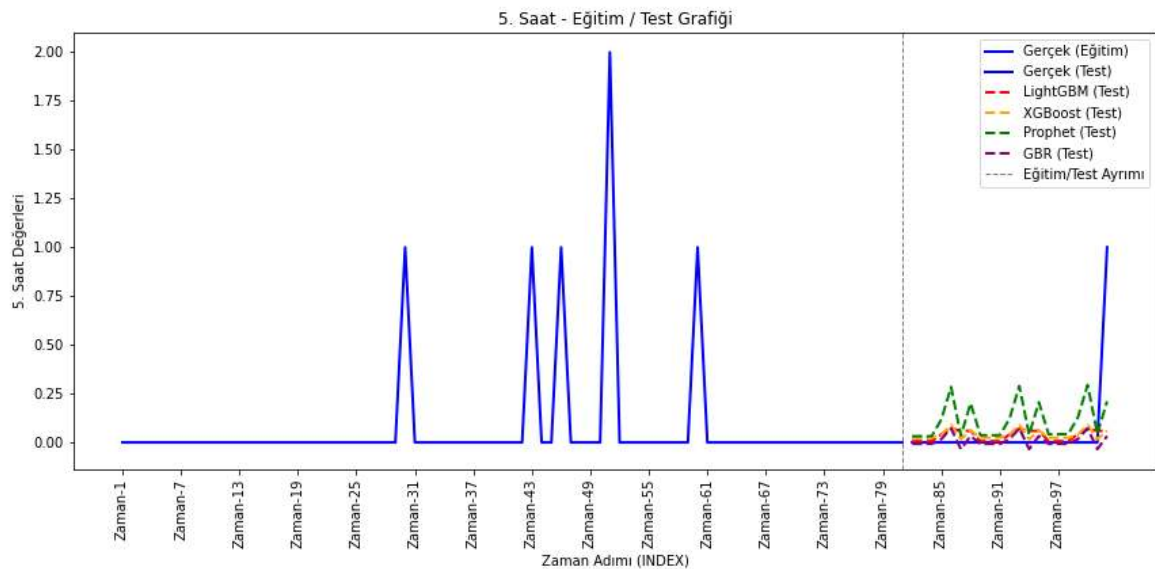
Dördüncü saat diliminde verilerin büyük kısmı sıfıra yakın değerlere sahiptir. Prophet ve GBR dışındaki tüm modeller, sıfır civarında sabit tahminler sunmuştur. Prophet modelinin bu saat aralığında nispeten daha yüksek tahminler üretmesi, modelin eğilim bazlı

tahmin stratejisinden kaynaklanmaktadır. Grafiksel olarak bu eğilim farklılığı Şekil 4.50 üzerinden gözlemlenebilir.



Şekil 4.50 : V4 veri seti için Python tabanlı 4. saat tahmin sonuçları.

Beşinci saatte ise modellerin genel olarak birbirine yakın performanslar gösterdiği görülmektedir. Prophet modeli bu saat aralığında da diğerlerinden bir miktar yüksek tahminlerde bulunmuştur. LightGBM, XGBoost ve GBR modelleri ise oldukça benzer düz çizgiler sunarak düşük varyanslı tahminlerde bulunmuştur. Gerçek veri ile tahmin eğrilerinin karşılaştırması Şekil 4.51'de sunulmaktadır.

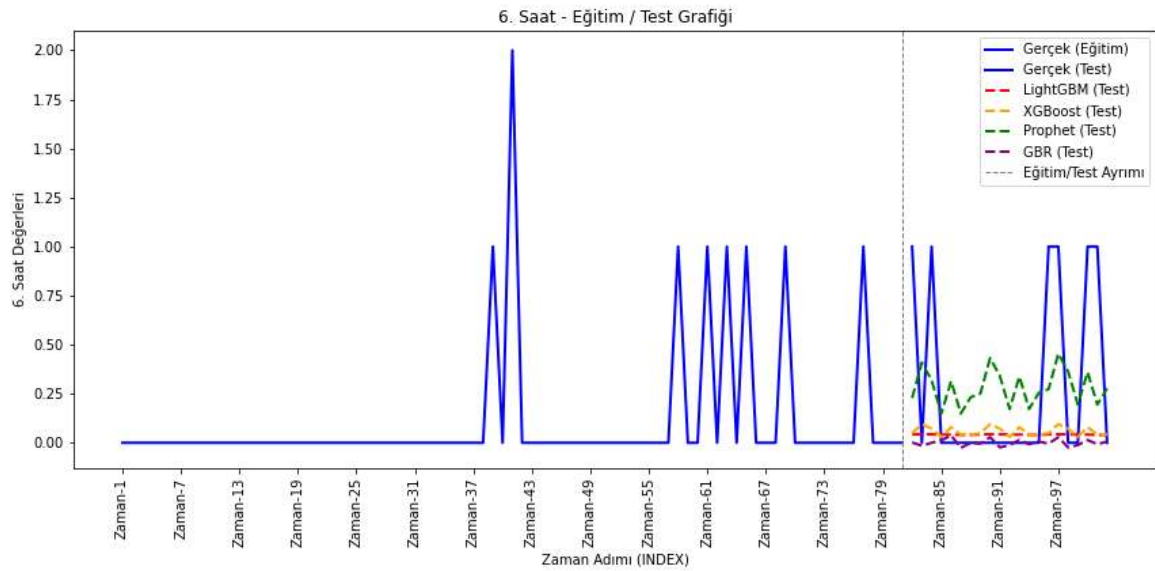


Şekil 4.51 : V4 veri seti için Python tabanlı 5. saat tahmin sonuçları.

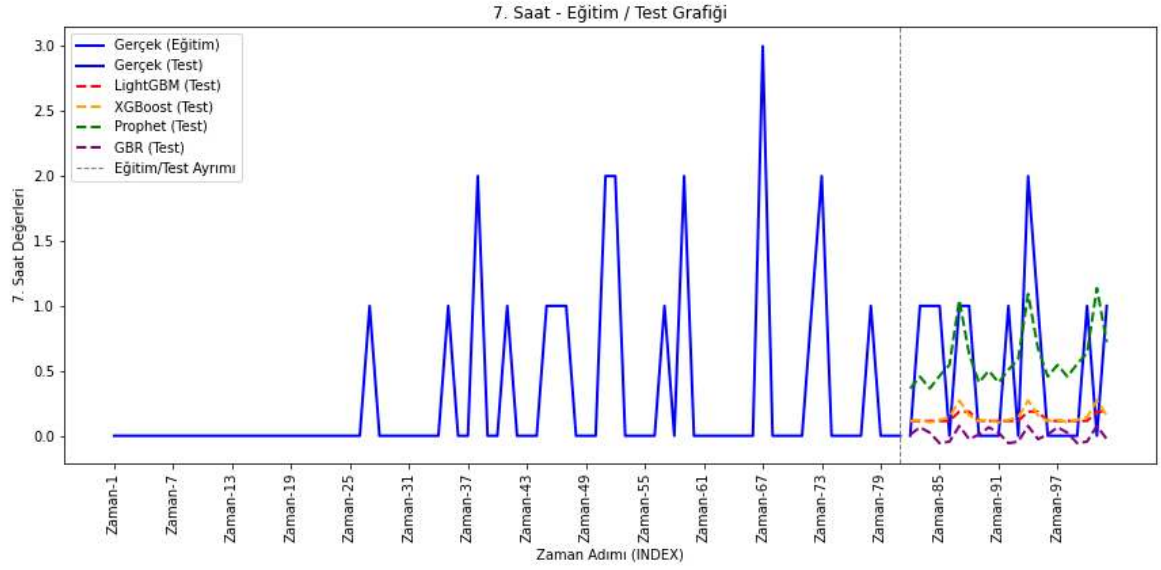
Altıncı saat dilimine ait tahmin performansları Şekil 4.52'de sunulmaktadır. Bu saat aralığında gözlemlenen gerçek veriler, önceki saat dilimlerine kıyasla oldukça düşük

düzeyde seyretmektedir. Gerçek değerlerde ani sıçramaların ve sıfır düzeyine sıkça dönüşlerin görülmesi, modellerin öğrenmesini zorlaştıran bir yapı sunmuştur. Bu doğrultuda Prophet modeli, diğer modellere kıyasla daha yüksek düzeyli ve değişken bir tahmin eğrisi sunmuştur. LightGBM ve XGBoost modelleri ise oldukça yatay ve düşük seviyeli çizgiler üretmiştir. Özellikle Prophet ve GBR modelleri, test verisindeki yukarı yönlü artışları kısmen takip edebilmiş; diğer modeller ise düşük varyanslı tahminlerde bulunarak bu dalgalanmaları yakalayamamıştır. Şekil 4.52’de de görüldüğü üzere, bu saat diliminde model başarımı oldukça sınırlıdır ve verinin seyrekliği tahmin doğruluğunu doğrudan etkilemiştir.

Yedinci saat için elde edilen tahmin sonuçları incelendiğinde, LightGBM, XGBoost ve Prophet modellerinin tahminlerinin oldukça benzer bir düzlemde seyrettiği, ancak gerçek test verilerini genel olarak düşükten tahmin ettikleri görülmektedir. Özellikle Prophet modelinin düz ve durağan yapıda kalması, ani değişimleri yakalayamadığını ortaya koymaktadır. Öte yandan GBR modelinin tahminlerinin daha geniş aralıkta dalgalandığı ve bazı noktalarda gerçek veriye yaklaşıp da genel doğruluğunun diğer modeller kadar başarılı olmadığı gözlemlenmektedir. Bu gözlem, Şekil 4.53’de açık biçimde görülmektedir.



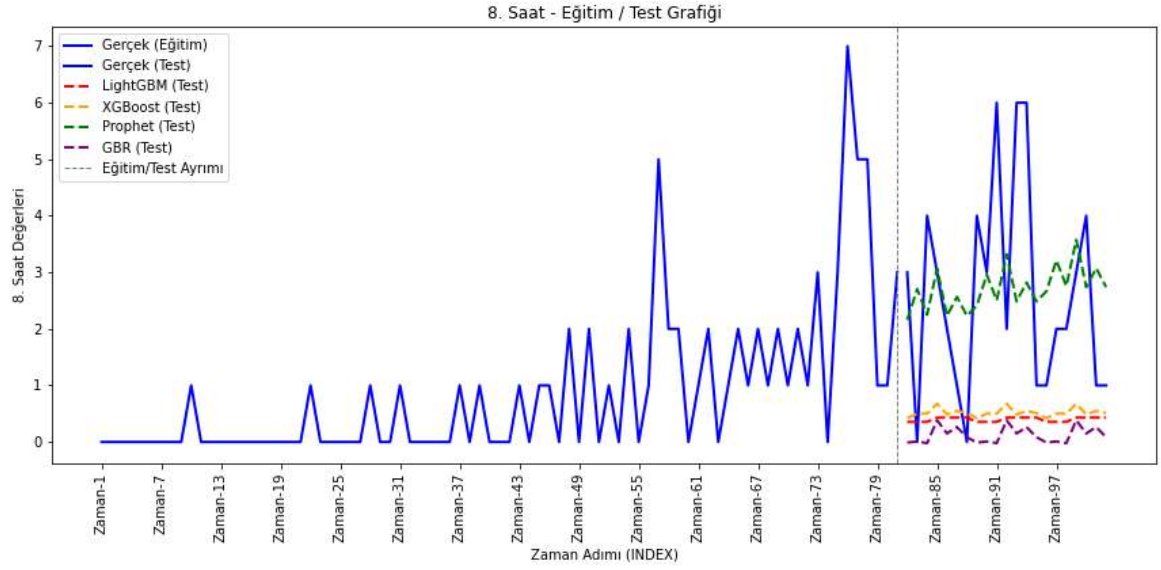
Şekil 4.52 : V4 veri seti için Python tabanlı 6. saat tahmin sonuçları.



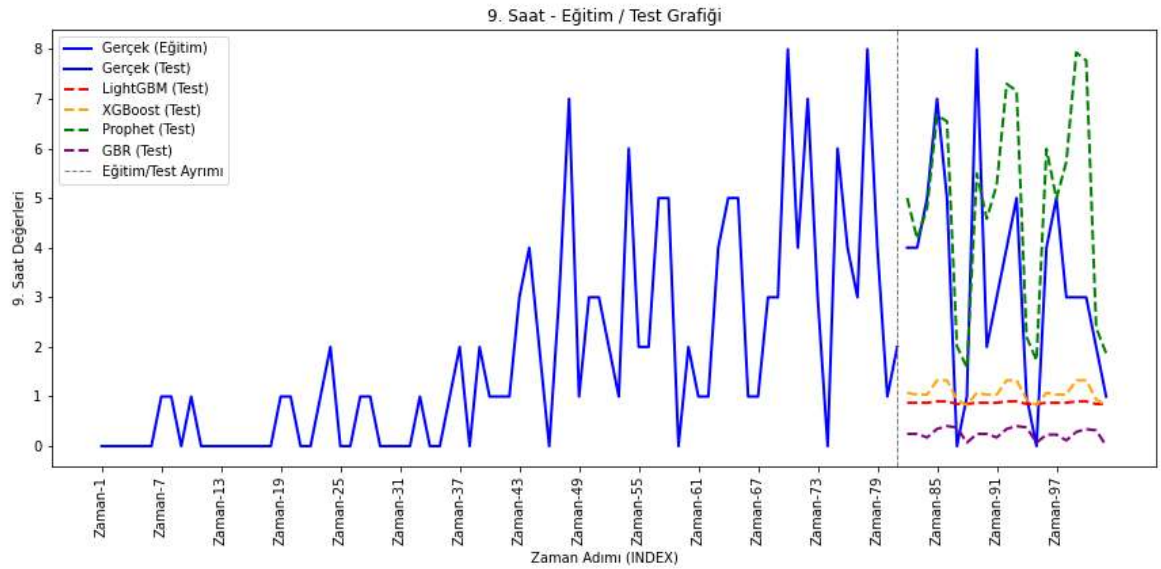
Şekil 4.53 : V4 veri seti için Python tabanlı 7. saat tahmin sonuçları.

Sekizinci saate ait tahminlerde ise Prophet ve GBR modellerinin görece daha yüksek değerlere ulaşabildiği ve bu yönüyle gerçek veriye daha yakın seyrettiği fark edilmektedir. LightGBM ve XGBoost tahminleri benzer bir bantta yer almakla birlikte, genel olarak daha düşük düzeyde kalmaktadır. Gerçek test verisi bu saat aralığında daha fazla dalgalanma göstermesine rağmen modellerin düz seyri, sınırlı bir yaklaşıklık sağlamıştır. Özellikle Prophet modelinin bu saat diliminde görece olarak daha başarılı olduğu söylenebilir. İlgili değerlendirme Şekil 4.54’de sunulmaktadır.

Dokuzuncu saat analizine bakıldığında, gerçek test verisinin yüksek frekansta dalgalandığı bir saat dilimi olduğu göze çarpmaktadır. Prophet modeli bu dalgalanmaları kısmen yakalamakta başarılı olurken, LightGBM ve XGBoost modellerinin sabit ve düşük tahminleriyle sınırlı kaldığı görülmektedir. Bu durum, modellerin genel düzlemdeki performansını aşağı çekmektedir. GBR modelinin ise geniş varyanslı yapısı bu saat özelinde öne çıkmakta, ancak zaman zaman aşırı sapma gösterdiği de anlaşılmaktadır. Tüm bu bulgular Şekil 4.55’de görselleştirilmiştir.

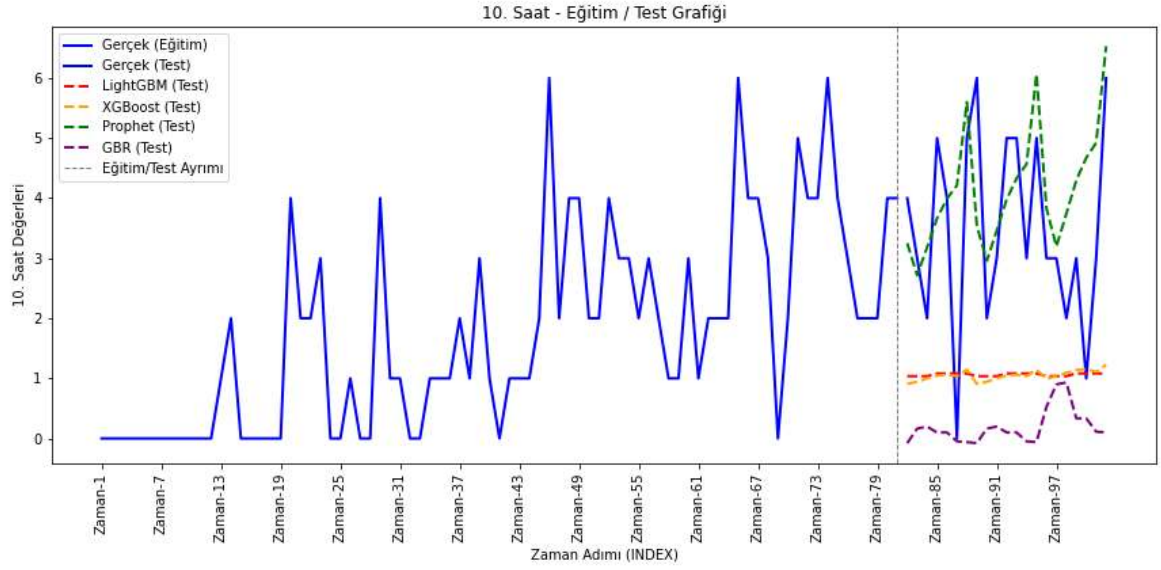


Şekil 4.54 : V4 veri seti için Python tabanlı 8. saat tahmin sonuçları.



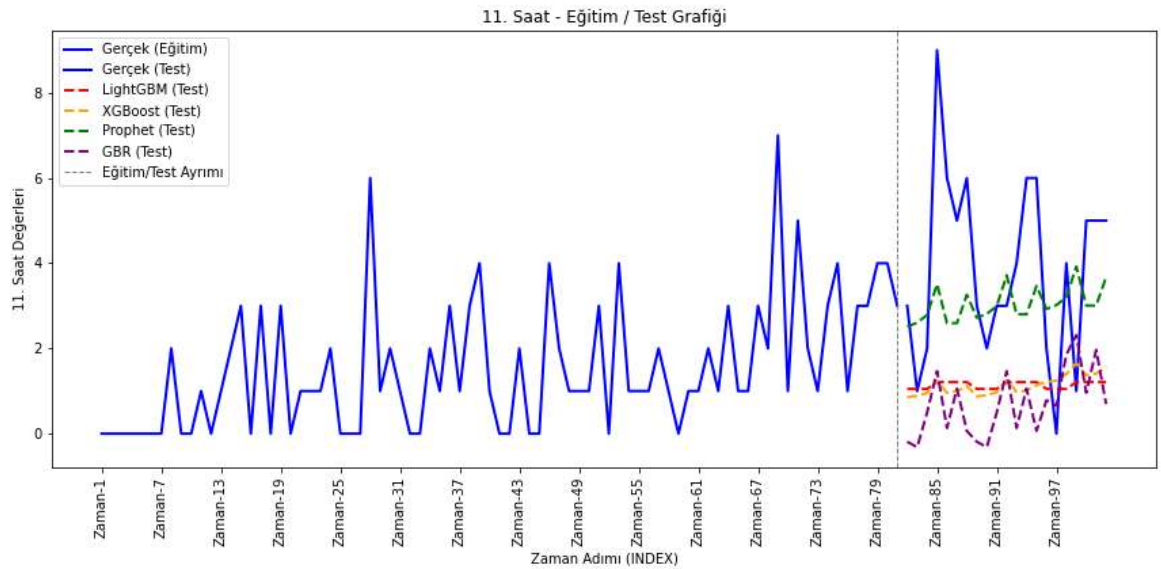
Şekil 4.55 : V4 veri seti için Python tabanlı 9. saat tahmin sonuçları.

Onuncu saate ilişkin sonuçlar incelendiğinde, Prophet modelinin test döneminde belirgin bir yükseliş eğiliminde olduğu, bu nedenle gerçek verinin genel trendine en yakın tahmini sunduğu ifade edilebilir. LightGBM ve XGBoost modelleri bu eğilimi daha yatay ve durağan şekilde yansıtırken, GBR modelinin daha serbest ve değişken bir çizgide ilerlediği gözlemlenmiştir. Prophet modelinin özellikle bu saatlik aralıkta istikrarlı performans sergilemesi, Şekil 4.56’da dikkat çekmektedir.



Şekil 4.56 : V4 veri seti için Python tabanlı 10. saat tahmin sonuçları.

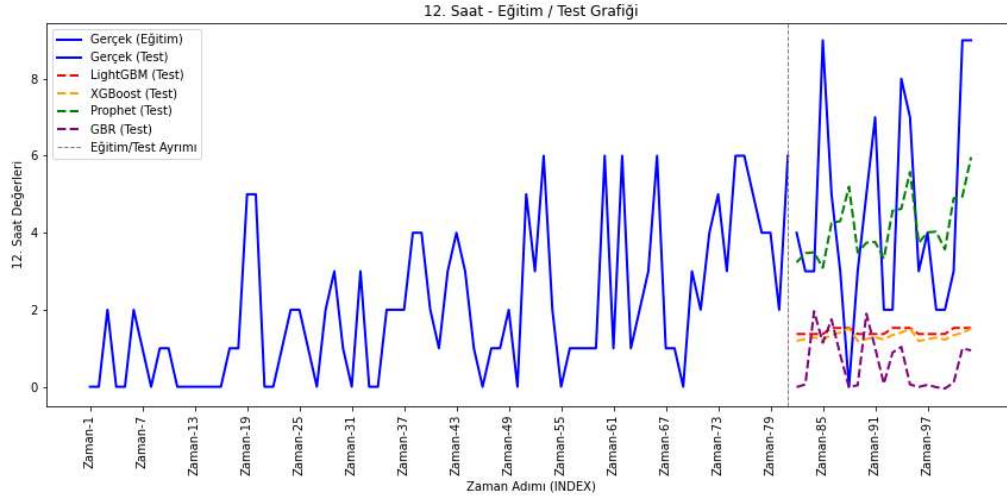
On birinci saat diliminde, modeller arasında genel benzerlikler görülmekle birlikte Prophet ve GBR tahminleri gerçek test verisine daha yakın seyretmiştir. Özellikle Prophet modeli, test verisindeki dalgalanmaları daha iyi takip edebilmiş ve ortalamaya yakın tahminlerde bulunmuştur. Diğer modellerin (LightGBM, XGBoost) tahminleri ise benzer eğriler izlese de değer bazında daha düşük kalmıştır. Gerçek test verisinin yüksekte açılıp düşüş gösterdiği bu saat özelinde Prophet'ın avantajlı olduğu Şekil 4.57'de açıkça görülebilmektedir.



Şekil 4.57 : V4 veri seti için Python tabanlı 11. saat tahmin sonuçları.

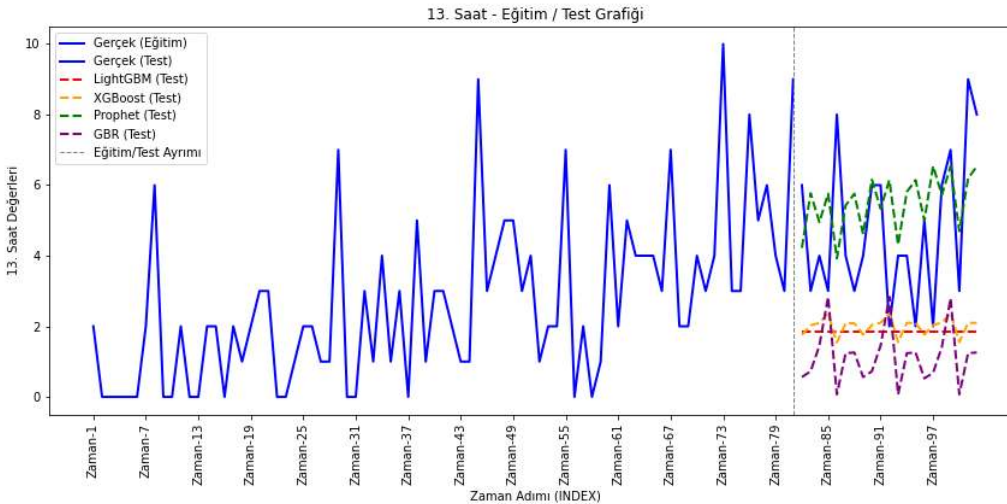
On ikinci saat diliminde modellerin performansları incelendiğinde, Prophet yönteminin genel eğilimi en iyi yansıttığı görülmektedir. Ancak gözlenen ani sıçramaları

yakalama konusunda hafif sapmalar gözlemlenmiştir. LightGBM, XGBoost ve GBR modelleri ise düşük varyanslı ve yatay seyirli tahminler üretmiş, bu nedenle özellikle dalgalı yapıdaki veri ile uyumlu sonuçlar verememiştir. Şekil 4.58’de bu değerlendirmeler açıkça görülmektedir.



Şekil 4.58 : V4 veri seti için Python tabanlı 12. saat tahmin sonuçları.

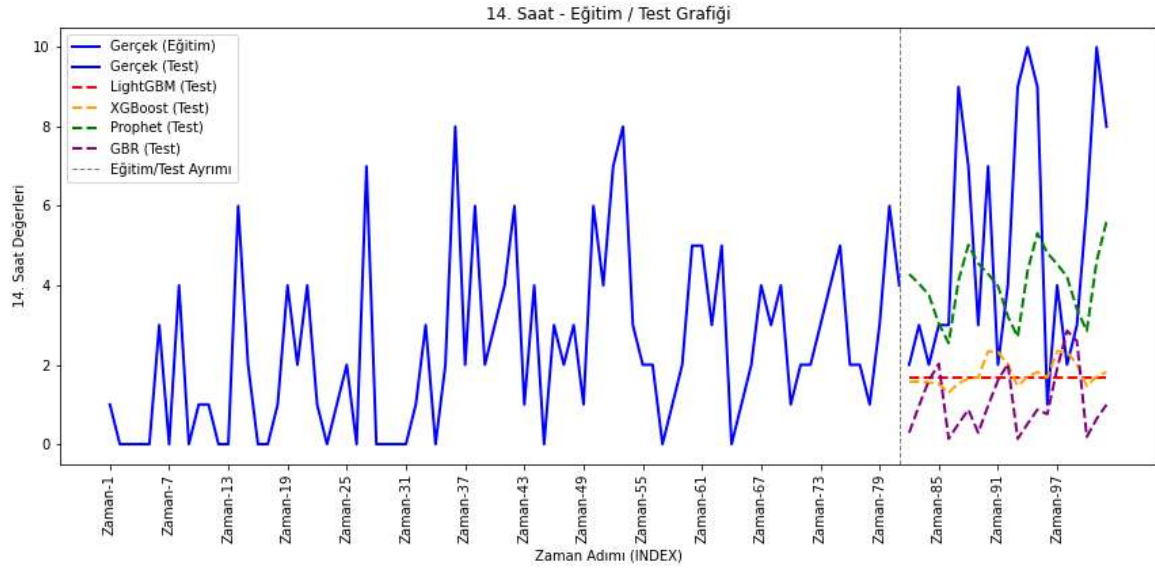
On üçüncü saat verileri incelendiğinde Prophet ve GBR yöntemlerinin eğilimleri daha dengeli bir biçimde yakaladığı gözlemlenmektedir. XGBoost ve LightGBM gibi boosting temelli modeller ise önceki saate benzer şekilde düşük varyanslı çizgiler üretmiş, bu da ani artışları açıklamakta yetersiz kalmıştır. Prophet çizgisinin yukarı yönlü eğilimi, özellikle test dönemindeki artışla uyum içindedir. Şekil 4.59 bu durumu desteklemektedir.



Şekil 4.59 : V4 veri seti için Python tabanlı 13. saat tahmin sonuçları.

On dördüncü saatlik tahminler incelendiğinde, Prophet modeli yine diğerlerine kıyasla daha esnek bir tahmin eğrisi sunmuştur. Ancak test döneminde gözlenen yüksek

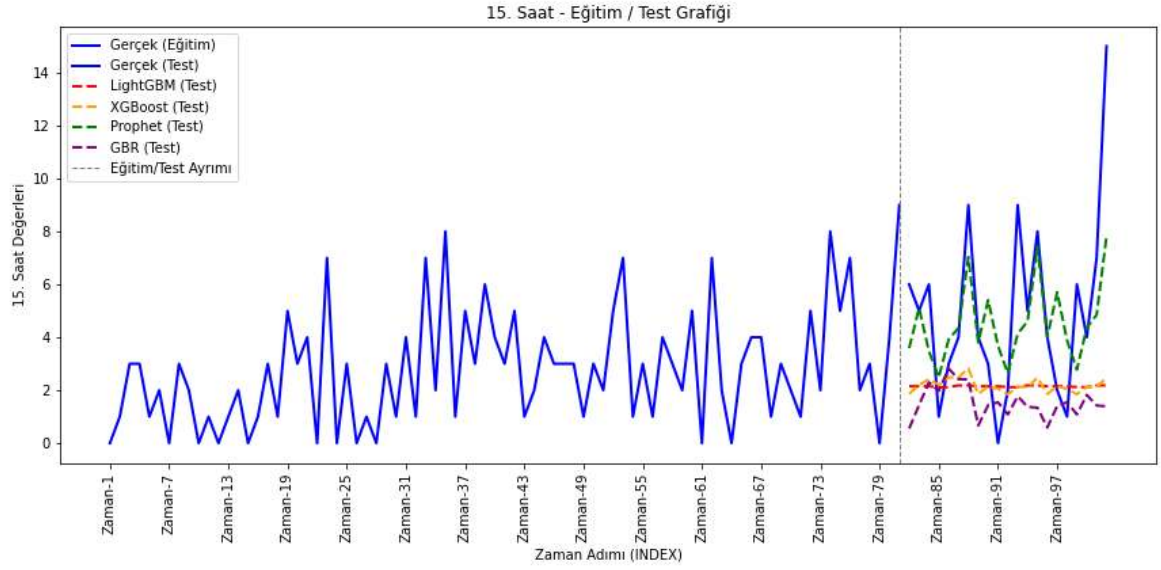
varyans Prophet için de zorluk yaratmıştır. LightGBM, XGBoost ve GBR yöntemleri bu saatte benzer çizgiler üretmiş, özellikle test verisinin düzeyine yakın tahmin değerleri elde edilmiştir. Şekil 4.60’da bu saat dilimine ait model sonuçları karşılaştırmalı olarak verilmektedir.



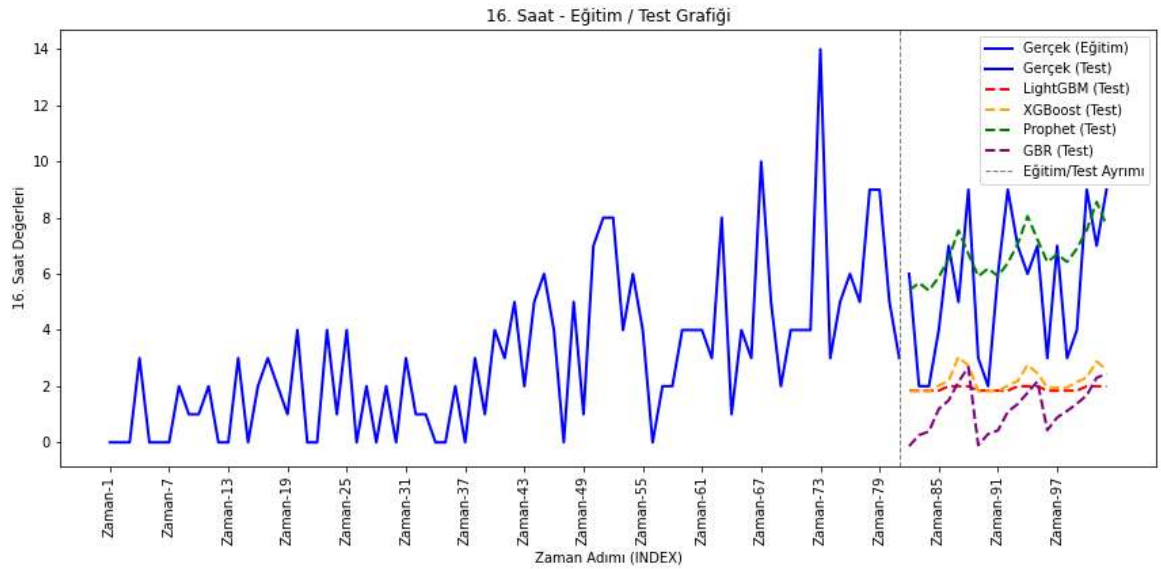
Şekil 4.60 : V4 veri seti için Python tabanlı 14. saat tahmin sonuçları.

On beşinci saat için yapılan analizde Prophet, verinin yukarı yönlü seyrini genel hatlarıyla takip edebilmiştir. Bununla birlikte, özellikle en yüksek değerlerin bulunduğu test kısımlarında Prophet modelinin bir miktar aşağıda kaldığı gözlemlenmiştir. Boosting modelleri bu saat için de istikrarlı ve düşük sapmalı tahminler üretmiştir. Ancak bu durum yüksek varyansa sahip gerçek değerlere kıyasla eksik kalmalarına neden olmuştur. Şekil 4.61’de bu duruma ait grafiksel değerlendirme yer almaktadır.

On altıncı saatlik tahmin performansları incelendiğinde Prophet yönteminin doğrusal olmayan artışı yakalama konusundaki başarısı dikkat çekmiştir. Boosting modelleri ise bu eğilimi daha az izleyebilmiş, tahmin değerleri sabit kalmıştır. Özellikle GBR modelinin bu saat diliminde Prophet’e göre daha düşük eğilim gösterdiği gözlenmektedir. Şekil 4.62 incelendiğinde Prophet’in test dönemindeki hareketliliğe daha iyi uyum sağladığı anlaşılmaktadır.

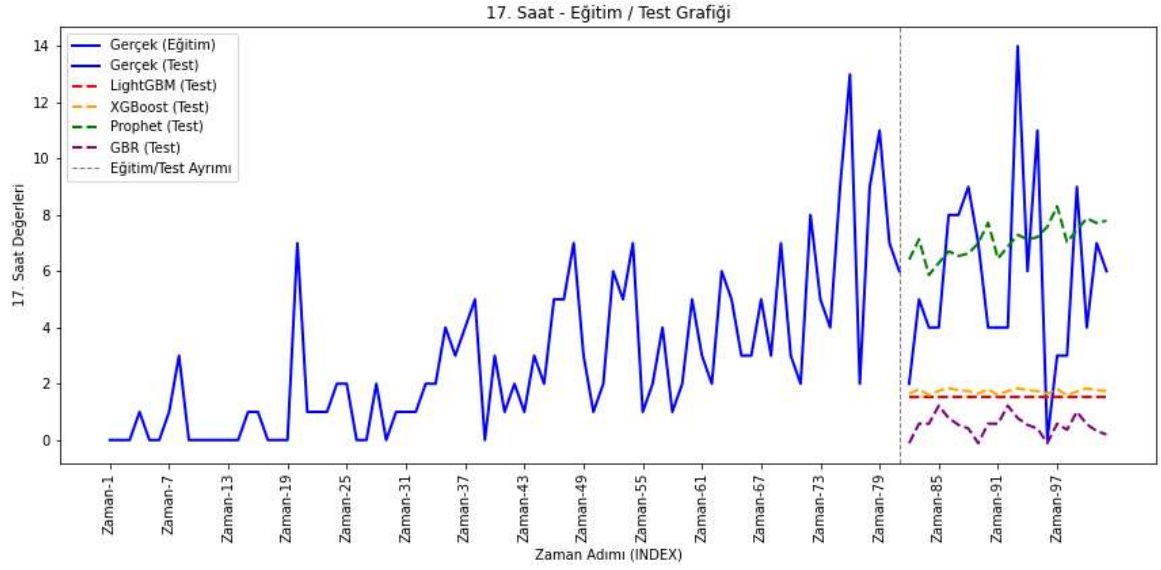


Şekil 4.61 : V4 veri seti için Python tabanlı 15. saat tahmin sonuçları.



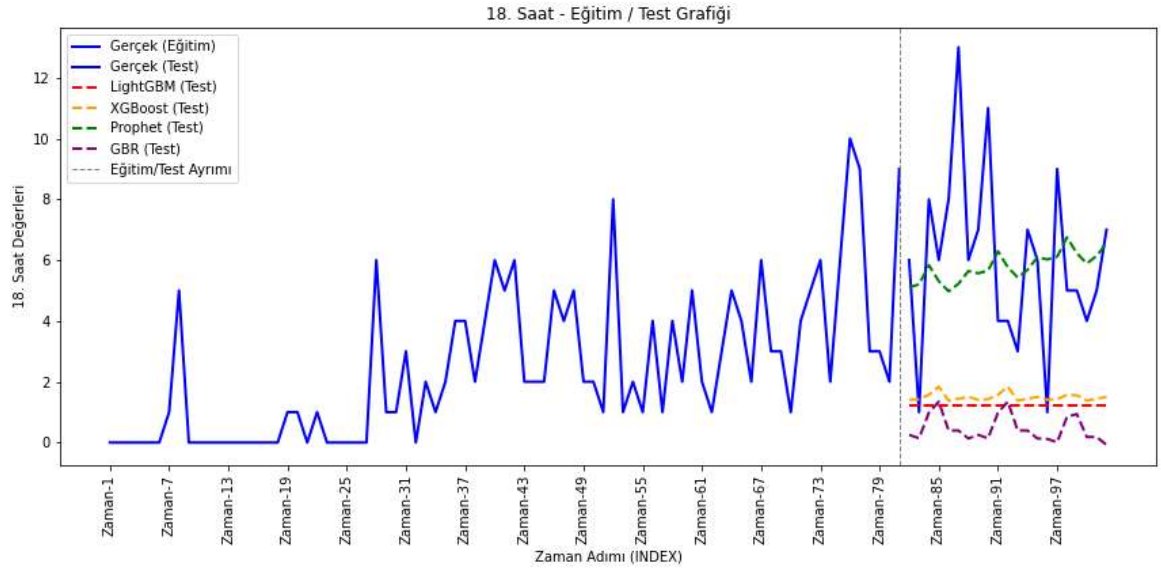
Şekil 4.62 : V4 veri seti için Python tabanlı 16. saat tahmin sonuçları.

On yedinci saat için elde edilen sonuçlarda Prophet yöntemi yüksek varyansa sahip veriyi takip ederken, diğer yöntemler bu değişkenliği yansıtmakta zorlanmıştır. Özellikle LightGBM ve XGBoost modellerinin sınırlı tahmin aralıkları, artış yönünde seyreden test verisini temsil etmekte yetersiz kalmıştır. Prophet, bu saat diliminde gerek eğilim gerekse düzey açısından en başarılı grafik çizimini sunmuştur. Şekil 4.63’de bu saatlik analiz görselleştirilmektedir.



Şekil 4.63 : V4 veri seti için Python tabanlı 17. saat tahmin sonuçları.

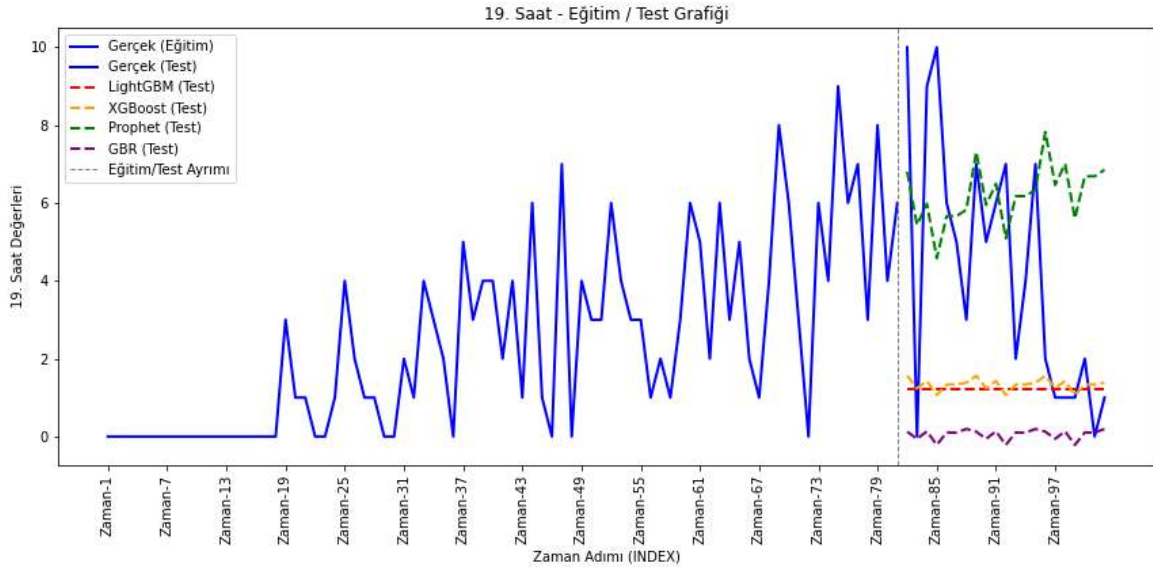
On sekizinci saat tahmin sonuçlarına ait grafik Şekil 4.64’de sunulmaktadır. Gerçek değerlerin oldukça dalgalı bir seyir izlediği bu saat diliminde, modellerin tahmin çizgileri birbirine oldukça yakın ilerlemiş ancak genel düzeyde ortalamaya yakın değerler vermiştir. Prophet ve LightGBM modelleri sabit düzeyde kalırken, XGBoost tahmin çizgisi görece olarak yukarı yönlü eğilim sergilemiştir. Bu saat diliminde en yüksek varyans GBR modelinde görülmektedir.



Şekil 4.64 : V4 veri seti için Python tabanlı 18. saat tahmin sonuçları.

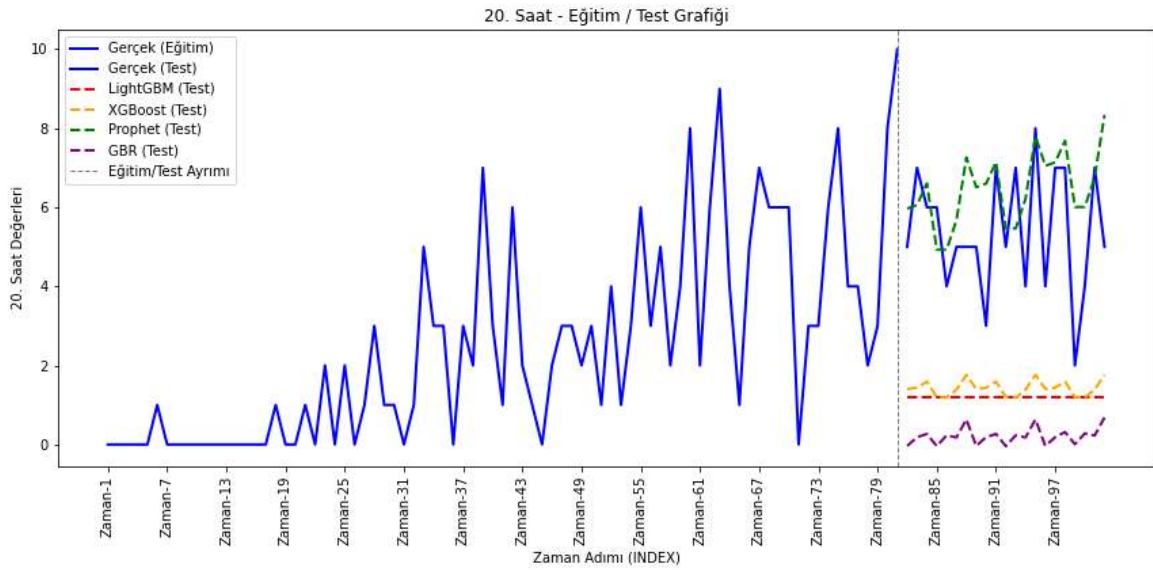
Şekil 4.65’de gösterilen on dokuzuncu saat grafiklerinde, Prophet modeli diğer modellerden belirgin biçimde ayrılarak test periyodu boyunca görece daha yüksek tahmin değerleri üretmiştir. LightGBM ve XGBoost modellerinin yatay düzleme yakın ilerlemesi,

bu saat diliminde stabil bir tahmin performansı hedeflediklerini göstermektedir. GBR modelinin tahmin eğrisi ise Prophet'e göre daha düşük ancak değişkenliği daha az bir seyir izlemiştir.



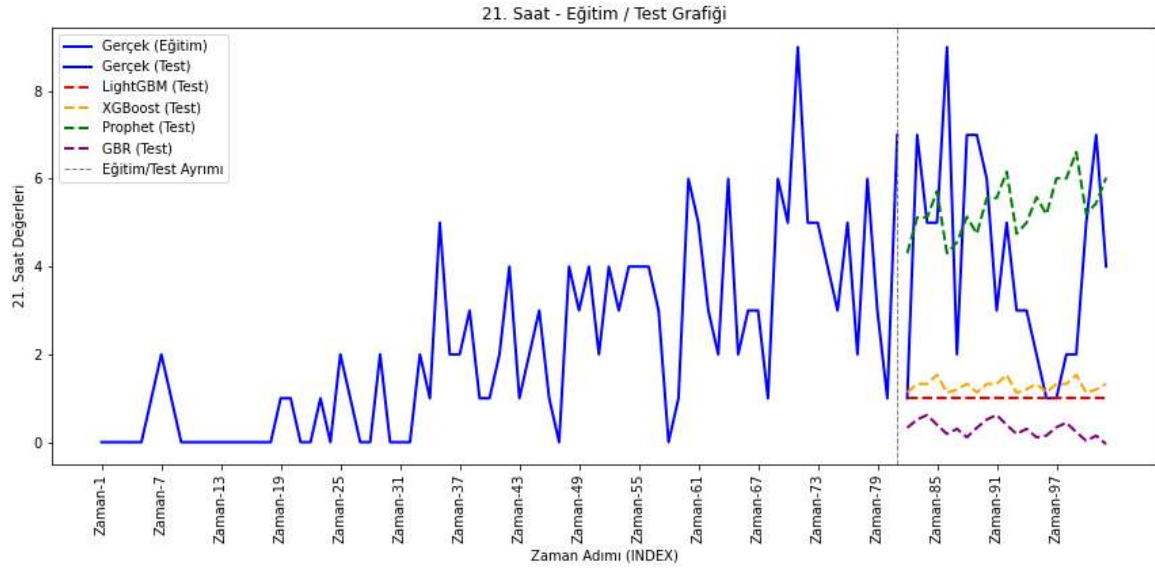
Şekil 4.65 : V4 veri seti için Python tabanlı 19. saat tahmin sonuçları.

Yirminci saate ilişkin tahmin performansları Şekil 4.66'da verilmektedir. Gerçek verilerdeki artış eğilimi, Prophet dışındaki modellerin çoğu tarafından sınırlı düzeyde temsil edilebilmiştir. Özellikle LightGBM ve XGBoost modelleri düz çizgilerle sabitlenmiş bir tahmin eğilimi sunarken, GBR modelinin tahmin çizgisi gerçek değerlere en yakın seyreden yapıda gözlemlenmiştir.



Şekil 4.66 : V4 veri seti için Python tabanlı 20. saat tahmin sonuçları.

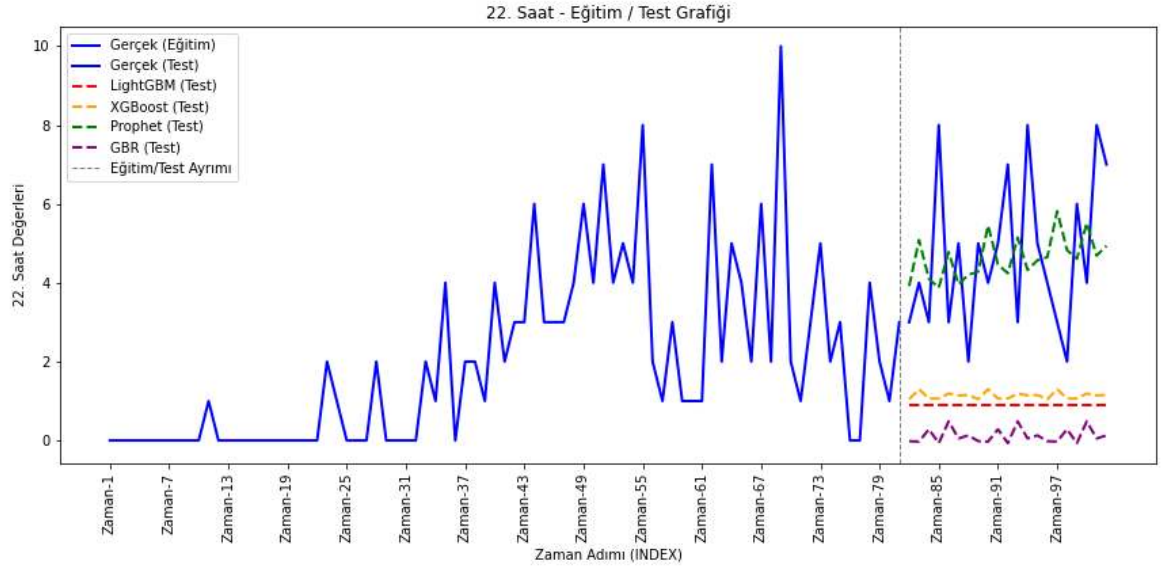
Yirmi birinci saat dilimine ilişkin model performansları Şekil 4.67’de sunulmaktadır. Bu zaman aralığında da Prophet modeli test bölümünde yükselme eğilimi göstermiştir. LightGBM ve XGBoost sabit tahmin çizgileriyle ayrışırken, GBR eğrisi gerçek verilerle örtüşen orta seviyeli tahminler üretmiştir. Model eğrilerinin birbirine çok yakın seyretmemesi, bu saatlik tahminin model seçimi açısından kritik olabileceğini göstermektedir.



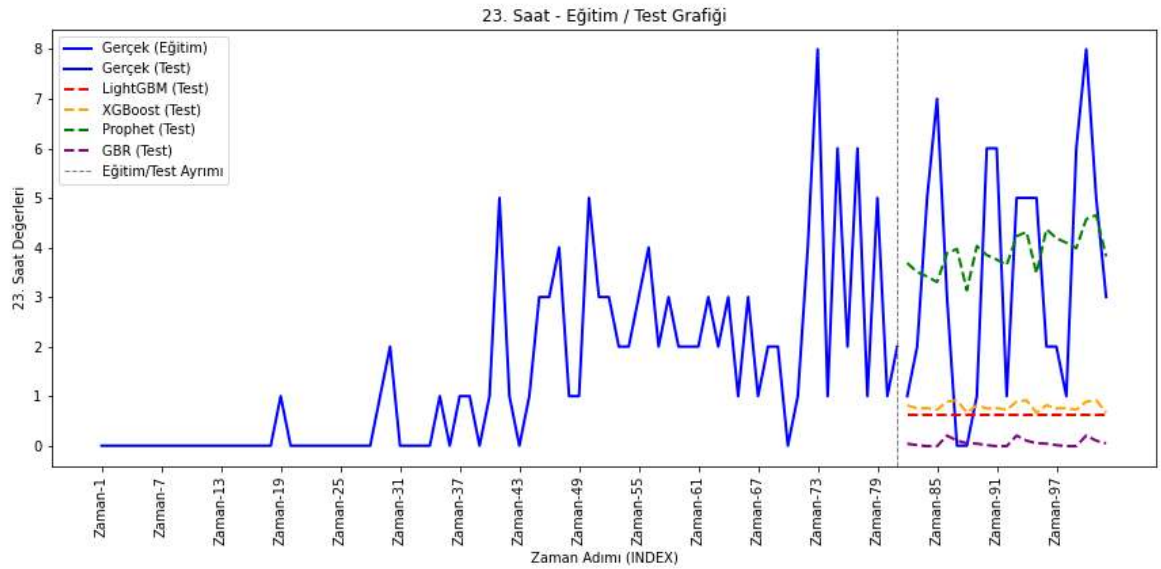
Şekil 4.67 : V4 veri seti için Python tabanlı 21. saat tahmin sonuçları.

Şekil 4.68’de gösterilen yirmi ikinci saat grafiklerine göre tüm modeller gerçek verinin alt bandında kalmıştır. Prophet ve GBR modelleri, gerçek eğrinin genel eğilimini en iyi izleyen modeller olmuştur. Diğer modeller sabit ve daha düşük düzeylerde seyrederek verinin dinamiklerini temsil etme açısından sınırlı kalmıştır.

Son olarak yirmi üçüncü saate ilişkin tahmin sonuçları Şekil 4.69’da verilmektedir. Bu saat diliminde Prophet modelinin yukarı yönlü tahmin çizgisi, gerçek verilere yaklaşıırken; LightGBM ve XGBoost çizgileri yine sabit düzeyde kalmıştır. GBR modeli ise orta seviyeli, fakat Prophet’e göre daha dar bantlı bir tahmin eğrisi sunmuştur. Bu da GBR’nin kontrollü ama esnek bir tahmin sunduğunu göstermektedir.



Şekil 4.68 : V4 veri seti için Python tabanlı 22. saat tahmin sonuçları.



Şekil 4.69 : V4 veri seti için Python tabanlı 23. saat tahmin sonuçları.

4.2.2 Model Performanslarının Karşılaştırılması

4.2.2.1 V1 Veri Seti Performans Karşılaştırması

ARIMA modelleri kapsamında farklı p, d ve q parametre kombinasyonlarıyla oluşturulan yapılar detaylı bir biçimde test edilmiştir. Bu analiz sürecinde, klasik AR(1), AR(2), MA(1), MA(2) gibi temel modellerin yanı sıra, durağan olmayan ARIMA türleri ile çeşitli karma modeller de değerlendirilmiştir. Çizelge A.1 incelendiğinde, hata metrikleri açısından en düşük RMSE (5.4233), MSE (29.4122) ve MAE (4.1047) değerlerinin Random Walk modeline ait olduğu görülmektedir. Bu model, herhangi bir parametre içermeyen ve

yalnızca bir önceki değerin korunacağı varsayımıyla çalışan oldukça basit bir yaklaşımdır. Bu yönüyle, tahmin gücü referans kabul edilse de, modelin açıklayıcılık derinliği sınırlıdır.

Bu bağlamda, istatistiksel olarak daha anlamlı modeller arasında değerlendirilen AR(2), sahip olduğu düşük AIC (545.3918) ve BIC (560.4021) değerleri ile öne çıkmaktadır. AIC (Akaike Information Criterion - Akaike Bilgi Kriteri) ve BIC (Bayesian Information Criterion - Bayesyen Bilgi Kriteri), modellerin karmaşıklığını ve veri üzerindeki uyumunu birlikte değerlendirerek daha iyi modellerin seçilmesini sağlayan kriterlerdir. Bu değerler, modelin veriye ne kadar iyi uyduğunu ifade eden hata terimlerini ve modelin karmaşıklığını birlikte ele alır. Her iki ölçüt için de daha düşük değerler, modelin hem daha iyi uyum sağladığını hem de aşırı karmaşık olmadığını, yani "aşırı öğrenmeden" kaçınıldığını göstermektedir. Bu nedenle AR(2) modeli, yalnızca hata metrikleri açısından değil, model seçimi kriterleri bakımından da tutarlı ve dengeli bir yapı sergilemektedir.

Aynı şekilde, Karma ve Daha Karmaşık Karma modelleri de benzer biçimde hem düşük hata değerleri hem de uygun AIC/BIC sonuçları ile dikkat çekmekte ve V1 veri seti üzerinde daha güvenilir sonuçlar verebilecek yapılar olarak öne çıkmaktadır. Öte yandan, durağan olmayan ARIMA modelleri, RMSE ve MAPE gibi hata metriklerinde anlamlı artışlara sebep olmuş, bu da veri setinin yapısal olarak durağan modellere daha uygun olduğunu göstermiştir. Sonuç olarak, her ne kadar hata metrikleri bağlamında Random Walk modeli en düşük değerleri sunsa da, açıklayıcılığı yüksek, parametrelili ve optimize edilmiş yapılar açısından AR(2) ve Karma modellerin tercih edilmesi, hem istatistiksel hem de pratik anlamda daha anlamlı ve güvenilir bir modelleme yaklaşımı sunmaktadır.

Çizelge A.2 incelendiğinde, SARIMA modelleri arasında en düşük RMSE (6.9970) ve MSE (48.9578) değerlerine sahip olan Basit SARIMA modeli, aynı zamanda düşük AIC ve BIC skorlarıyla da öne çıkmaktadır. Buna karşın, Mevsimsel SARIMA ve Hibrit SARIMA modelleri, benzer AIC/BIC değerlerine rağmen yüksek hata oranları ile zayıf performans sergilemiştir. Yüksek Dereceli SARIMA ise karmaşık yapısına rağmen hata metriklerinde belirgin bir üstünlük sağlayamamıştır. Bu sonuçlar, V1 veri seti üzerinde daha yalın ve optimize edilmiş SARIMA modellerinin daha etkili olduğunu göstermektedir.

Çizelge A.3 incelendiğinde, Holt-Winters yöntemi kapsamında test edilen dört farklı yapı; trendli, katı mevsimsel, adaptif mevsimsel ve basit üstel düzgünleştirme; tüm hata metriklerinde birebir aynı sonuçları vermiştir (RMSE: 10.9528, MAE: 9.5511, MAPE:

45.4845). Bu durum, ilgili yöntemlerin V1 veri seti üzerinde düşük parametrik duyarlılığa sahip olduğunu ve ayırt edici bir modelleme katkısı sunmadığını ortaya koymaktadır.

Çizelge A.4 (devam-2) V1 seri setiyle yapılan çalışmada LSTM mimarisinin en iyi performansı iki katmanlı ve her katmanda 200 gizli birim ile, öğrenme oranı 0.005 ve dropout oranı 0.2 olarak ayarlandığında elde edildi. Bu konfigürasyonda MSE 1.406, RMSE 1.1858, MAE 0.9594, MAPE 5.2824, SMAPE 5.3098 olarak ölçüldü ve MBE 0.0456 ile düşük kaldı. Aynı yapıda dropout uygulanmadığında RMSE 1.2945, MAE 1.0324; öğrenme oranı 0.001'e düşürüldüğünde ise hata metriklerinde belirgin bozulma gözlemlendi. Dolayısıyla V1 seri setinde daha yüksek gizli birim sayısı ve uygun düzeyde dropout ile doğru öğrenme oranı seçiminin LSTM'in genelleme kapasitesini güçlendirerek hem noktasal hem de oransal hata metriklerinde anlamlı iyileşme sağladığı söylenebilir.

Çizelge A.5'e göre en iyi sonuçlar 100 ağaçlı, derinlik 3 ve minimum yaprak sayısı 5 olarak ayarlandığında elde edilmiştir. Bu konfigürasyonda MSE 39.1399, RMSE 6.2562, MAE 4.7335, MAPE 23.4930, SMAPE 25.0660 ve MBE 2.9453 olarak ölçülmüş; tüm hata metrikleri diğer kombinasyonlara kıyasla en düşük veya en dengeli değerleri sergilemiştir. 50 ağaçlı ve derinlik 5, yaprak sayısı 5 ayarında ikinci en düşük MSE (39.2162) görülse de ağaç sayısının artırılması modelin önyargısını daha da azaltarak hataların daha da düşmesini sağlamıştır. Bununla birlikte daha yüksek derinlik veya daha küçük yaprak değeri içeren ayarlarda RMSE ve MAE değerlerinde artış gözlemlenmiş, bu da aşırı uyuma işaret etmektedir. Dolayısıyla 100 ağaç, derinlik 3 ve yaprak sayısı 5 kombinasyonu hem genelleme hem de hata minimizasyonu açısından Random Forest modelinin V1 seri seti üzerinde en uygun hiperparametre seçimini göstermektedir.

Çizelge A.6 V1 seri setine ait SVR sonuçlarına göre en düşük hata metrikleri $C = 0.1$, $\epsilon = 0.1$, linear çekirdek ve ölçek = 1 kombinasyonunda elde edildi. Bu ayarda MSE 30.095, RMSE 5.4859, MAE 4.0742, MAPE 22.1639, SMAPE 21.4202 ve MBE 0.9092 olarak ölçüldü. Karşılaştırmalı olarak aynı C değeri ve çekirdek için $\epsilon = 0.01$ olduğunda hem MSE hem de RMSE ve MAE değerleri artış gösterdi; ölçek = auto seçildiğinde ise tüm hata metriklerinde belirgin bozulma yaşandı. $C = 1$ değerinde linear çekirdeğin performansı kıyasla daha yüksek hatalar üretti; gaussian çekirdek ise tüm kombinasyonlarda ciddi ölçüde kötü sonuçlar verdi. Dolayısıyla V1 seri setinde düşük C ve ϵ değerleriyle linear çekirdek seçimi ve ölçek = 1 ayarı, SVR modelinin genelleme ve hata minimizasyonu açısından en uygun hiperparametre kombinasyonunu göstermektedir.

Çizelge A.7'ye göre LightGBM modelinin en düşük RMSE ve MAE değerleri 15 yaprak, öğrenme oranı 0.01 ve 300 tahminci sayısı kombinasyonunda elde edilmiştir. Bu ayarda RMSE 1.1028, MAE 0.7949 ve R^2 -0.0191 ile model diğer tüm konfigürasyonlara kıyasla hem noktasal hem de görelî hata metriklerinde en iyi performansı sergilemiştir. Daha yüksek öğrenme oranı kullanıldığında RMSE ve MAE değerlerinde artış gözlenirken, yaprak sayısının 31 veya 50'ye çıkarılması da genelleme yeteneğini zayıflatarak hata metriklerini kötüleştirmiştir. Tahminci sayısının 200'den 300'e çıkması modelin stabilitesini artırmış, ancak 300'den fazla ağaç eklemenin aşırı hesap maliyetine rağmen anlamlı bir iyileşme sağlamadığı anlaşılmıştır. Dolayısıyla V1 seri seti üzerinde düşük öğrenme oranı, sınırlı yaprak sayısı ve yeterli sayıda tahminci kombinasyonu LightGBM'in genelleme kapasitesini en üst düzeye çıkaran en uygun hiperparametre ayarlarını temsil etmektedir.

Çizelge A.8'e göre XGBoost modelinin en iyi performansı 100 tahminci, öğrenme oranı 0.05, maksimum derinlik 3, alt örnek oranı 0.7 ve sütun oranı 0.7 kombinasyonunda elde edilmiştir. Bu ayarda RMSE 1.0961, MAE 0.7921, R^2 -0.0067 ve MBE -0.2548 ile diğer yapılandırmalara kıyasla hata metriklerinde belirgin iyileşme görülmüştür. Öğrenme oranının 0.01'den 0.05'e çıkarılması RMSE ve MAE değerlerini anlamlı düzeyde düşürürken, ağaç sayısının 200'e yükseltilmesinin sağladığı ek iyileşme çok sınırlı kalmıştır. Maksimum derinlik 5'e çıkarıldığında veya alt örnek ve sütun oranları tam seçildiğinde hata metriklerinde artış tespit edilmiştir; bu da daha karmaşık ağaç yapılarının genelleme kapasitesini olumsuz etkilediğini göstermektedir. Dolayısıyla V1 seri seti üzerinde dengeli ağaç sayısı, ılımlı derinlik ve kısmi örnekleme oranları XGBoost'un genelleme performansını en üst düzeye çıkaran ideal hiperparametre ayarlarını temsil etmektedir.

Çizelge A.9 V1 seri setiyle Prophet modeli sonuçlarına bakıldığında en düşük RMSE 2.3675 ve MAE 2.0249 ile değişim noktası ölçek 0.01, mevsimsellik multiplicative ve tatil ölçeği 5 veya 10 ayarlarında elde edilmiştir. Additive mevsimsellikte RMSE 2.3711 iken multiplicative seçildiğinde hata metriklerinde hafif bir iyileşme görülmüştür. Tatil ölçeği 5 ile 10 arasında performans farkı gözlenmemiştir ve her iki durumda da SMAPE 134.6907, MBE 1.8116 değerleri tekrarlanmıştır. Daha yüksek değişim noktası ölçek değerleri (0.05) modelin hata metriklerini belirgin şekilde artırarak genelleme kapasitesini zayıflatmıştır. Dolayısıyla V1 seri setinde Prophet için düşük değişim noktası ölçek, multiplicative mevsimsellik ve orta düzey tatil ölçeği kombinasyonu modelin en iyi tahmin performansını sağlamıştır.

Çizelge A.10 V1 seri setiyle yapılan çalışmada GBR modelinin en düşük RMSE 1.1014 ile 100 tahminci, maksimum derinlik 3, öğrenme oranı 0.05 ve min yaprak sayısı 5 konfigürasyonunda elde edildi. Bu ayarda MAE 0.7961, MAPE 1.1901E+17, R2 -0.0163, SMAPE 145.4411 ve MBE -0.2623 olarak ölçüldü. Öğrenme oranı aynı kalmak kaydıyla tahminci sayısı 50'ye indirildiğinde veya yaprak sayısı 10'a çıkarıldığında RMSE ve MAE değerlerinde hafif artış gözlenirken, öğrenme oranının 0.1'e yükseltilmesi de hata metriklerini kötüleştirdi. Maksimum derinliğin 5'e çıkarılması modelin değişkenliği azaltmak yerine artan hata değerlerine yol açtı. Dolayısıyla V1 veri seti üzerinde yeterli sayıda ağaç, sınırlı derinlik ve orta düzey öğrenme oranı ile min yaprak sayısı kombinasyonu GBR'nin genelleme yeteneğini en üst düzeye çıkaran en uygun hiperparametre ayarlarını temsil etmektedir.

4.2.2.2 V2 Veri Seti Performans Karşılaştırması

Çizelge B.1'e göre V2 veri setinde en düşük RMSE değeri 5.4233 ile Random Walk modeline aittir; aynı modelin MSE'si 29.4122, MAE'si 4.1047 ve MAPE'si 25.1524 olarak ölçülmüştür. Diğer tüm ARIMA varyantları bu hata metriklerinin gerisinde kalmış; örneğin genişletilmiş karma modelde RMSE 7.5977, MSE 57.7251 ve MAE 6.2871 olarak yükselmiştir. AIC ve BIC kriterlerine göre en iyi uyum 472.7207 ve 491.4836 ile genişletilmiş karma modelde gözlense de tahmin hatası bakımından Random Walk'un temel bir referans modeli olarak daha başarılı olduğu açıktır. Dolayısıyla V2 serisinde zaman serisi dinamiklerini modelleme çabaları karmaşık ARIMA yapılarıyla artan bilgi kriteri iyileşmesine yol açsa da öngörü doğruluğu açısından en yalın yaklaşım olan Random Walk en iyi performansı sunmuştur.

Çizelge B.2'de görüldüğü üzere V2 veri seti için en düşük hata metrikleri basit SARIMA modelinde elde edilmiştir. Bu modelde MSE 48.9578, RMSE 6.9970 ve MAE 5.5560 ile diğer yapılara kıyasla belirgin üstünlük sergilenmiştir. Mevsimsel ve hibrit SARIMA modelleri AIC ve BIC açısından aynı derecede iyi uyum gösterse de hata metriklerinde sırasıyla RMSE 11.7142 ve 11.7141 değerlerine ulaşarak öngörü doğruluğu bakımından basit yapının gerisinde kalmıştır. Yüksek dereceli SARIMA'da da RMSE 8.2046 ile daha yüksek bir hata söz konusudur. Aylık SARIMA ise ölçeklendirme sorunları nedeniyle anlamsız sonuçlar üretmiştir. Dolayısıyla V2 serisinde hem hata minimizasyonu hem de model yalınlığı açısından basit SARIMA en uygun hiperparametre ayarını temsil etmektedir.

Çizelge B.3 V2 veri seti için incelendiğinde tüm yöntemlerin MSE 119.9644, RMSE 10.9528, MAE 9.5511 ve MAPE 45.4845 değerleri ile tamamen özdeş performans sergilediği görülmektedir. Hem Holt doğrusal trend hem de Holt-Winters katı mevsimsel, adaptif mevsimsel ve basit üstel düzgünleştirme modelleri hata metrikleri açısından birbirinden ayrılmamıştır. Bu durumda model seçimi, ek bilgi veya bileşen (trend ya da mevsimsellik) eklemenin getireceği karmaşıklık yerine yalınlık ve hesaplama verimliliği göz önünde bulundurularak yapılabilir. Dolayısıyla V2 serisinde en basit yapı olan basit üstel düzgünleştirmenin yeterli olduğu ve gereksiz model karmaşıklığından kaçınılmasının öngörü performansını etkilemeyeceği sonucuna varılabilir.

Çizelge B.4 V1 seri setiyle yapılan LSTM sonuçlarına göre en düşük hata iki katmanlı, her katmanda 200 gizli birim, öğrenme oranı 0.005 ve dropout oranı 0.2 konfigürasyonunda elde edildi. Bu ayarda MSE 1.4053, RMSE 1.1855, MAE 0.9099 ve MAPE 4.7693 olarak ölçüldü. Karşılaştırmalı olarak tek katmanlı yapıda 100 birim, aynı öğrenme oranı ve dropout 0.2 ile RMSE 1.1915 iken iki katmanlı yapı ve 100 birim konfigürasyonunda RMSE 1.2536'ya yükselmiş; öğrenme oranı 0.001'e düşürüldüğünde veya dropout uygulanmadığında hata metriklerinde belirgin bozulma gözlenmiştir. Dolayısıyla V1 serisinde derinliği artırmak yerine gizli birim sayısını yükseltmek ve uygun düzeyde düzenleme uygulamak, LSTM'in genelleme kapasitesini güçlendirerek hem noktasal hem de oransal hata metriklerinde anlamlı iyileşme sağlamıştır.

Çizelge B.5'e göre Random Forest modelinin en iyi performansı 50 ağaçlı, maksimum derinlik 5 ve minimum yaprak sayısı 5 konfigürasyonunda elde edilmiştir. Bu ayarda MSE 38.6505, RMSE 6.2170 ve MAE 4.7051 ile diğer tüm kombinasyonların altında kalmış; MAPE 23.6158 ve SMAPE 38.6505 ile hata oranlarını en aza indirmiştir. Derinlik 5 ve yaprak sayısı 1 ayarında RMSE 6.3341 olarak ölçülürken yaprak sayısının 5'e çıkarılması RMSE'yi 0.1171 puan iyileştirmiştir. Ağaç sayısının 100'e yükseltilmesi ise genelleme kapasitesine kayda değer bir katkı sağlamayarak RMSE değerini artırmıştır. Dolayısıyla B2 veri seti üzerinde Random Forest için sınırlı ağaç sayısı, makul derinlik ve yeterli yaprak sayısı kombinasyonu en uygun hiperparametre seçimini temsil etmektedir.

Çizelge B.6'ya göre V2 veri setinde SVR modelinin en düşük hata metrikleri C 0.1, hata payı 0.1, linear çekirdek ve ölçek 1 kombinasyonunda elde edilmiştir. Bu ayarda RMSE 5.4859, MAE 4.0742 ve MAPE 22.1639 ile diğer tüm yapılandırmalardan daha başarılı sonuçlar söz konusudur. Aynı C değeri ve çekirdek için ölçek auto seçildiğinde RMSE 6.2794'e yükselerek hata metriklerinde belirgin bozulma gözlenmiştir. C 0.1 ve hata payı

0.01 ayarında ise ölçek 1 için RMSE 5.5325, MAE 4.1024 olarak ölçülmüştür. C 1 değerine çıktığında ise RMSE 5.5144 ile performans bir miktar gerilemiş; gaussian çekirdek tüm kombinasyonlarda anlamlı şekilde daha kötü sonuçlar üretmiştir. Dolayısıyla V2 setinde düşük C ve hata payı değerleriyle linear çekirdek ve ölçek 1 seçimi SVR'nin genelleme ve hata minimizasyonu açısından en uygun hiperparametre konfigürasyonunu temsil etmektedir.

Çizelge B.7 V2 seri seti için LightGBM modelinde en düşük RMSE değeri 6.9915 ile 15 yaprak, öğrenme oranı 0.05 ve 100 tahminci kombinasyonunda elde edildi. Bu ayarda MAE 5.4426, MAPE 26.4995, R2 -0.5784 ve SMAPE 28.9010 olarak ölçülürken MBE -3.6981 ile kabul edilebilir düzeyde kaldı. Yaprak sayısının 31 veya 50'ye çıkarılması aynı hiperparametrelerle RMSE'yi artırarak genelleme yeteneğinin zayıfladığını gösterdi. Tahminci sayısını 200 veya 300'e yükseltmek de hata metriklerinde belirgin bir iyileşme sağlamadı; aksine RMSE 7.2181 ve 7.2155 değerleriyle bozulma gözlemlendi. Öğrenme oranının 0.01'e düşürülmesi ise RMSE'yi 7.0945-8.3324 aralığına çıkararak performansı olumsuz etkiledi. Dolayısıyla V2 veri seti üzerinde sınırlı sayıda yaprak, orta düzey öğrenme oranı ve yeterli sayıda tahminci kombinasyonu LightGBM'in en iyi genelleme performansını sağladı.

Çizelge B.8'e göre V2 veri seti için XGBoost modelinde en düşük RMSE değeri 6.7107 ile 200 tahminci, öğrenme oranı 0.05, maksimum derinlik 3, alt örnek oranı 1 ve sütun oranı 0.5 konfigürasyonunda elde edildi. Bu ayarda MAE 5.3737 ve MAPE 26.7213 olarak ölçülürken R2 -0.4542 ve MBE -3.3727 değerleriyle model dengeli bir performans sergiledi. Alt örnek ve sütun oranlarının her ikisinin de kısmi seçimi (0.7 ve 0.7) RMSE'yi 6.7455'e yükselterek genelleme yeteneğinde hafif bir zayıflama yaratırken, tahminci sayısının 100'e indirilmesi veya aşırı derinlik ve örnekleme kombinasyonlarının kullanılması RMSE'nin daha da artmasına yol açtı. Özellikle sütun oranının yarıya düşürülmesi modelin ağaçlar arası çeşitliliği artırarak hata metriklerini iyileştirdiği; buna karşılık tam örnekleme (alt örnek oranı 1) genelleme kapasitesini desteklediği görülmektedir. Dolayısıyla V2 serisinde dengeli ağaç sayısı, ılımlı derinlik ve kısmi sütun örnekleme ile tam alt örnekleme kombinasyonu XGBoost'un en başarılı hiperparametre ayarlarını temsil etmektedir.

Çizelge B.9 V2 serisi üzerinde Prophet modeli için en düşük RMSE değeri değişim noktası ölçek 0.1, mevsimsellik multiplicative ve tatil ölçeği ayarlarında elde edilmiştir. Bu konfigürasyonda RMSE 5.1305, MAE 4.0495, MAPE 22.2425, SMAPE 20.9271 ve MBE

-0.6386 olarak ölçülmüştür. Additive mevsimsellikte MAE 4.0050 ile hafif bir iyileşme kaydedilse de multiplicative seçildiğinde RMSE’de belirgin düşüş sağlanmış, bu da zaman serisi verisinin mevsimsel genlik değişimlerini multiplicative bileşenle daha doğru yakaladığını göstermektedir. Değişim noktası ölçeğinin 0.1’e çıkarılması modelin esnekliğini artırarak hem noktasal hem de oransal hatalarda iyileşme sağlamıştır. Tatil ölçeği 1, 5 veya 10 arasında performans farkı görülmemiştir. Dolayısıyla V2 setinde Prophet için multiplicative mevsimsellik, yüksek değişim noktası ölçeği ve orta düzey tatil bileşeni kombinasyonu en uygun hiperparametre ayarlarını temsil etmektedir.

Çizelge B.10 V2 veri seti için XGBoost modelinin en iyi performansı 50 tahminci, maksimum derinlik 3, öğrenme oranı 0.2 ve min yaprak sayısı 1 konfigürasyonunda elde edildi. Bu ayarda RMSE 6.5696, MAE 5.2320, MAPE 26.2122, R2 -0.3937 ve SMAPE 27.6194 ile diğer tüm kombinasyonlardan daha düşük hata değerleri görüldü. Öğrenme oranının 0.1’e düşürülmesi RMSE’yi 6.6532–6.8767 aralığına çıkarırken, min yaprak sayısının 5’e yükseltilmesi RMSE’yi 6.8095’e yükselterek genelleme yeteneğinde zayıflama yarattı. Tahminci sayısını 100 veya 150’ye çıkarmak da hata metriklerini iyileştirmemiştir. Dolayısıyla V2 serisinde yüksek öğrenme oranı, sınırlı derinlik ve en düşük yaprak sayısı kombinasyonu, XGBoost’un genelleme performansını en üst düzeye çıkaran ideal hiperparametre ayarlarını temsil etmektedir.

4.2.2.3 V3 Veri Seti Performans Karşılaştırması

Çizelge C.1 V3 veri seti üzerinde test edilen ARIMA modelleri incelendiğinde en düşük hata metrikleri MA(2) modelinde elde edilmiştir. Bu yapıda p 0, d 0, q 2 ayarlarıyla MSE 19012.8513, RMSE 137.8871 ve MAE 136.6721 değerleri ölçülmüş; ayrıca AIC 50.8680 ve BIC 58.4352 ile bilgi kriterlerinde de en iyi uyum sağlanmıştır. MA(1) modelinde RMSE 137.8897, MSE 19013.58 olarak benzer sonuçlar alınsa da MA(2) yapılandırması hem MSE hem de RMSE’de hafif düşüş göstererek genelleme performansında öne çıkmıştır. AR(1) ve AR(2) modelleri RMSE bakımından sırasıyla 137.8931 ve 137.8941 değerleriyle takip edilirken, karma modellerin daha karmaşık p ve q değerleri hem bilgi kriterlerinde hem de hata metriklerinde MA(2) modelinin gerisinde kalmıştır. Dolayısıyla V3 serisinde p 0, d 0, q 2 konfigürasyonu hem hata minimizasyonu hem de model yalınlığı açısından en uygun seçim olarak öne çıkmaktadır.

Çizelge C.2’ye göre V3 veri seti için en düşük hata metrikleri mevsimsel SARIMA modelinde elde edilmiştir. Bu yapıda p 2, d 1, q 2 ve P 1, D 1, Q 1, s 7 ayarları ile MSE

18953.1241, RMSE 137.6703 ve MAE 136.4681 değerleri ölçülmüştür. AIC 87.4306 ve BIC 98.7816 ile bilgi kriterlerinde de en iyi uyumu sergileyen bu model, basit SARIMA ve aylık SARIMA konfigürasyonlarından daha düşük hata değerlerine ulaşmıştır. Hibrit SARIMA da aynı MSE ve RMSE değerlerini üretmiş olmakla birlikte ek Q parametresi model karmaşıklığını artırırken bilgi kriterlerinde ve öngörü doğruluğunda mevsimsel SARIMA ile eşdeğer performans göstermiştir. Daha yüksek dereceli SARIMA ve aylık s parametresi s 12 olarak ayarlandığında ise hata metrikleri belirgin şekilde kötüleşmiştir. Dolayısıyla V3 serisinde mevsimsel düzeni doğrudan modelleyen $p=2, d=1, q=2$ ve $P=1, D=1, Q=1, s=7$ kombinasyonu hem yalınlık hem de öngörü doğruluğu açısından en uygun parametre seçimini temsil etmektedir.

Çizelge C.3 V3 veri seti için incelendiğinde tüm üstel düzleştirme yöntemlerinin—Holt doğrusal trend, Holt-Winters katı mevsimsel, Holt-Winters adaptif mevsimsel ve basit üstel düzgünleştirme—tamamen özdeş performans sergilediği görülmektedir (MSE 616.7767, RMSE 24.8350, MAE 19.9667, MAPE 13.5765, $R^2=0.8530$). Mevsimsellik veya adaptif bileşen eklemenin hata metriklerinde herhangi bir iyileşme sağlamadığı bu sonuçlarla açıkça ortaya çıkmıştır. Dolayısıyla V3 serisinde, gereksiz model karmaşıklığına yol açmaksızın öngörü doğruluğunu etkilemeyen basit üstel düzgünleştirme yöntemi tercih edilebilir.

Çizelge C.4 V3 veri seti için BiLSTM sonuçlarına bakıldığında en düşük RMSE iki katmanlı, her katmanda 100 gizli birim ve öğrenme oranı 0.001, dropout 0 olacak şekilde “doğru” yönlendirmeye elde edilmiştir. Bu konfigürasyonda MSE 17.0386, RMSE 4.1278, MAE 2.9164 ve MAPE 2.2663 ölçülmüş; R^2 değeri 0.9488 ile yüksek açıklayıcılık göstermiştir. Diğer yapılandırmalarda ya tek katmanlı modellerde ya da dropout uygulanan hallerde RMSE 5’in üzerinde seyretmiş, kimi kombinasyonlarda 9.3’e kadar yükselmiştir. Özellikle iki katmanlı ve 200 birimli konfigürasyonlarda dropoutu artırmak hataları yükseltirken, katman sayısını artırmak yerine gizli birim sayısını 100’de tutmak genelleme kapasitesini güçlendirmiştir. Dolayısıyla V3 serisinde çift yönlü hücre yapısını doğru yönlendirme ile kullanmak, doğru öğrenme oranı ve düzenleme ayarlarıyla BiLSTM’in en iyi performansı verdiğini ortaya koymaktadır.

Çizelge C.5’e göre V3 veri seti üzerinde Random Forest modelinin en iyi performansı 50 ağaçlı, derinlik 3 ve min yaprak sayısı 1 konfigürasyonunda elde edildi. Bu ayarda MSE 830.4187, RMSE 28.8170, MAE 23.5571, MAPE 15.9664 ve $R^2=0.4948$ olarak ölçüldü. Ağaç sayısını 100’e yükseltmek veya derinliği 5’e çıkarmak RMSE’yi sırasıyla 29.8994 ve

31.3688'e yükseltirken, min yaprak sayısının 5'e çıkarılması da hataları belirgin biçimde artırdı. Dolayısıyla V3 serisinde görece daha az sayıda ağaç, orta derinlik ve en düşük yaprak sayısı kombinasyonu modelin genelleme kapasitesini koruyarak hata metriklerini en alt düzeye indiren en uygun hiperparametre ayarlarını temsil etmektedir.

Çizelge C.6'ya göre V3 seri seti için SVR modelinin en düşük RMSE değeri 1 ve ϵ 0.1, linear çekirdek ve ölçek 1 kombinasyonunda elde edildi. Bu ayarda RMSE 15.1546, MAE 12.7847, MAPE 9.0414 ve R2 0.3100 olarak ölçülmüştür. Linear çekirdeğin scale = auto seçildiğinde RMSE 24.4244'e yükselirken, gaussian çekirdek tüm kombinasyonlarda belirgin biçimde daha kötü sonuçlar üretmiştir. C 0.1 ve ϵ 0.1 ayarında ölçek = 1 için RMSE 15.1802 iken C 1 ve ϵ 0.1'de hafif de olsa ek bir iyileşme gözlenmiştir. Dolayısıyla V3 verisinde yüksek C ve orta ϵ değerleriyle linear çekirdek ve ölçek = 1 ayarları, SVR'nin genelleme yeteneğini güçlendirerek hata minimizasyonu bakımından en uygun hiperparametre konfigürasyonunu temsil etmektedir.

Çizelge C.7 V3 veri seti için LightGBM sonuçlarına bakıldığında en düşük RMSE değeri 38.4888 ile öğrenme oranı 0.1 ve 300 tahminci sayısı kombinasyonunda elde edilmiştir. Bu ayarda MAE 34.6446, MAPE 23.7897, R2 -4.2692 ve SMAPE 27.5914 ölçülmüş; MBE ise -34.6446 ile kabul edilebilir sınırlar içinde kalmıştır. Öğrenme oranının 0.05'e düşürülmesi RMSE'yi 39.1826'ya yükseltirken tahminci sayısının 200'e çekilmesi de performansı bozmuştur. Yaprak sayısının 15, 31 veya 50 olarak değiştirilmesinin model başarısına kayda değer etkisi gözlenmemiştir; bu nedenle en basit yapı olarak yaprak sayısı 15 tercih edilebilir. Dolayısıyla V3 serisinde yüksek öğrenme oranı ve fazla sayıda tahminci kombinasyonu LightGBM'in genelleme yeteneğini en üst düzeye çıkaran ideal hiperparametre ayarlarını temsil etmektedir.

Çizelge C.8'e göre XGBoost modelinin en iyi performansı 300 tahminci, öğrenme oranı 0.05, maksimum derinlik 7, alt örnek oranı 1 ve sütun oranı 1 konfigürasyonunda elde edildi. Bu ayarda RMSE 16.2397, MAE 13.1577 ve MAPE 9.3851 ile diğer tüm kombinasyonların altında kalmıştır; R2 değeri 0.0619 ile modelin açıklayıcılığı da göreceli olarak yüksektir. Alt örnek oranı veya sütun oranı kısmen seçildiğinde (0.5 veya 0.7) RMSE değerleri 16.3122–17.0478 aralığına yükselmiş, bu da tam örnekleme ve tam sütun seçiminin genel hata minimizasyonuna katkıda bulunduğunu göstermektedir. Öğrenme oranının 0.1'e çıkarılması ise RMSE'yi 16.7633–17.6687 bandına taşımış, derinliğin yükseltilmesi veya tahminci sayısının 200'e düşürülmesi de benzer şekilde performansı zayıflatmıştır. Dolayısıyla V3 serisinde yüksek tahminci sayısı, makul derinlik ve tam örnekleme

kombinasyonu XGBoost'un genelleme kapasitesini en üst düzeye çıkaran ideal hiperparametre ayarlarını temsil etmektedir.

Çizelge C.9 V3 seri seti için Prophet modeli sonuçlarına bakıldığında en düşük RMSE 22.5998 ile değişim noktası ölçeği 0.001 ve hem additive hem multiplicative mevsimsellik seçeneklerinde elde edilmiştir. Bu ayarda MAE 17.7138, MAPE 12.1161, SMAPE 13.1855 ve MBE -14.0927 olarak ölçülmüş; R2 değeri -0.4706 ile modelin açıklayıcılığının zayıf kaldığını göstermiştir. Değişim noktası ölçeği 0.01'e yükseltildiğinde RMSE 23.4949'a, 0.05 ve 0.1 değerlerinde ise sırasıyla 23.6263 ve 23.6324'e çıkmış; tatil ölçeği 1, 5 veya 10 arasında ise performans farkı gözlenmemiştir. Dolayısıyla V3 verisinde Prophet için değişim noktası ölçeğinin düşük tutulması hata metriklerinde anlamlı iyileşme sağlarken mevsimsellik ve tatil ölçeği ayarları performansı etkilememiştir.

Çizelge C.10'a göre GBR modelinin V3 veri seti üzerinde en iyi performansı 50 tahminci, maksimum derinlik 5, öğrenme oranı 0.2 ve min yaprak sayısı 1 konfigürasyonunda elde edilmiştir. Bu ayarda RMSE 16.1055, MAE 13.0925, MAPE 9.3030, R2 0.0774 ve SMAPE 9.3437 ölçülmüştür. Tahminci sayısının 100 veya 150'ye çıkarılması aynı hiperparametrelerle neredeyse eş değer RMSE değerleri (16.1057) sunmakla birlikte, en düşük hesap yükü sağlayan 50 tahminci konfigürasyonu modelin genelleme kapasitesini dengeli biçimde koruduğunu göstermektedir. Öğrenme oranı 0.1 veya 0.05'e düşürüldüğünde RMSE 16.4160-16.6320 aralığına yükselerek hata metriklerinde belirgin bozulma gözlenmiştir. Derinliğin artırılması veya yaprak sayısının 1'den fazla seçilmesi ise modelin aşırı karmaşılaşmasına ve hata artışına neden olmuştur. Dolayısıyla V3 serisinde sınırlı sayıda ağaç, orta düzeyde derinlik ve yüksek öğrenme oranı ile en düşük yaprak sayısı kombinasyonu GBR'nin genelleme performansını en üst düzeye çıkaran ideal hiperparametre ayarlarını temsil etmektedir.

4.2.2.4 V4 Veri Seti Performans Karşılaştırması

Çizelge D.1'de V4 veri seti için her bir saat diliminde test edilen ARIMA temelli modellerin performansına yer verilmiştir. Çoğu saat için durağan olmayan MA(1) modeli en düşük RMSE değerine (örneğin saat 2'de 0.6697, saat 3'te 0.5599) ve en düşük MSE'ye (saat 3'te 0.3135) ulaşarak üstünlük göstermiştir. Bununla birlikte saat 5'te basit MA(1) modelinin RMSE'si 0.2130 ile açık ara en başarılı sonuç verirken, saat 10'da hata terimleriyle model ($q = 2$) RMSE = 2.5123 ile diğer ARIMA yapılarının önüne geçmiştir. Karma model (1,1,2) ise saat 13'te RMSE = 2.0443 ve MAPE = 52.6067 ile mevsimsel dışı

ARIMA varyantlarına kıyasla daha düşük mutlak hata üretmiştir. Diğer yandan saat 15 ve 17–19 gibi yüksek volatilitenin gözlemlendiği dilimlerde AR(2) ve durağan olmayan MA(1) kombinasyonları RMSE 4–5 aralığında seyretmiş, bu da kısa vadeli belirsizliğin model başarısını sınırladığını göstermiştir. Genel olarak, V4 serisindeki çoğu saat için durağan olmayan MA(1) modelinin temel alındığında daha karmaşık ARIMA yapılarına yalnızca belirli saatlerde (saat 5, 10, 13) ihtiyaç duyulduğu; dolayısıyla zaman dilimlerine özgü model seçiminin doğruluk üzerinde kayda değer etkisi olduğu sonucuna varılabilir.

Çizelge D.2'ye göre V4 veri setinin 24 saatlik periyodunda genel olarak mevsimsel SARIMA (2 1 2)(1 1 1)₇ konfigürasyonu birçok saatte en düşük hata değerlerini üretmiştir. Özellikle saat 1, 2 ve 3'te RMSE değerleri sırasıyla 1.1810, 0.6918 ve 0.6541 ile minimal seviyede kalırken saat 4'te basit SARIMA (1 1 1)(1 1 1)₇ modeli 0.8942 RMSE ile öne çıkmıştır. Tatil etkilerinin daha yoğun olduğu saat 5 ve 7'de hibrit SARIMA (2 1 1)(1 1 2)₇, sırasıyla 0.2204 ve 0.7722 RMSE ile en başarılı modeller olmuştur. Aylık SARIMA (1 1 1)(1 1 1)₁₂ modeli ise saat 0, 8, 12, 15, 17, 18 ve 21'de kullanıldığı çözünürlükte zaman ölçeği farklılaşmasının etkisiyle yer yer tercih edilmiş, saat 0'da 1.9358 RMSE ile diğerlerinden daha düşük hata sağlamıştır. Karma model yerine mevsimsel bileşeni doğrudan modelleyen yapıların genellikle daha stabil sonuç verdiği; nadiren hibrit yapıya ihtiyaç duyulduğu görülmektedir. Dolayısıyla V4 serisinde zaman dilimine özgü değişkenlik göz önünde bulundurularak çoğunlukla mevsimsel SARIMA konfigürasyonunun tercih edilmesi önerilmektedir.

Çizelge D.3'te görüldüğü üzere üstel düzleştirme modelleri arasında en düşük hata metreği saat 20'de elde edilmiştir: α 0.2 β 0 kombinasyonunda RMSE 1.5772, MAE 1.3490 ve MAPE 32.9680 ile diğer saatlere kıyasla belirgin üstünlük gösterilmiştir. Saat 22'de α 0.8 β 0.2 ayarı RMSE 2.1339 ve MAE 1.6332 ile ikinci en başarılı sonucu verirken, saat 14'te α 0.2 β 0.2 ayarı RMSE 2.8278 ve MAE 2.4978 değerleriyle benzer biçimde düşük hatalar sunmuştur. Buna karşın saat 13 ve 18'deki kombinasyonlar RMSE 3.1255 ve 3.2662 değerleriyle görece daha yüksek hata üretmiştir. Saat 16 için α 0.8 β 0 ayarında RMSE 3.1368, MAE 2.6198 ve MAPE 45.2479 ölçülmüş; saat 21'de α 0.2 β 0.8 ise RMSE 2.7011, MAE 2.1609 ile orta düzeyde performans sağlamıştır. Dolayısıyla V4 veri setinin farklı saat dilimlerinde değişen veri dinamikleri göz önüne alındığında, özellikle düşük hacimli saatlerde α 0.2 β 0 konfigürasyonunun üstel düzleştirme modelinin genelleme kabiliyetini en üst düzeye çıkardığı sonucuna varılabilir.

Çizelge D.4'te V4 veri setinin saatlik BiLSTM sonuçları incelendiğinde en düşük hata saat 6'da tek katmanlı, 100 gizli birim, öğrenme oranı 0.005 ve düşürme oranı 0 ile elde edilmiştir (MSE 0.0032, RMSE 0.0565, MAE 0.0488). Benzer olarak saat 3'te iki katmanlı, 100 birim, 0.001 öğrenme oranı ve 0.2 düşürme oranıyla RMSE 0.1278; saat 5'te iki katmanlı, 100 birim, 0.005 öğrenme oranı ve 0 düşürme oranıyla RMSE 0.0688 kaydedilmiştir. Genel olarak erken saat dilimlerinde (0–7) düşük öğrenme oranı ile minimal düşürme oranı performansı iyileştirirken, saatin ilerlemesiyle modelin karmaşıklığı ve birim sayısı arttığında RMSE değerleri 0.3–0.5 aralığına çıkmıştır. Örneğin saat 16'da RMSE 0.5351, saat 17'de RMSE 0.4978 ile hata metrikleri belirgin biçimde bozulmuştur. Dolayısıyla V4 serisinde BiLSTM'in genelleme kapasitesini en üst düzeye çıkarmak için saat dilimine özgü olarak, özellikle düşük hacimli saatlerde tek katmanlı ve düşük düşürme oranlı yapı tercih edilmelidir.

Çizelge D.5'e bakıldığında Random Forest modelinin V4 veri setinde en başarılı performansı saat 5'te 100 ağaç, derinlik 5 ve minimum yaprak sayısı 5 konfigürasyonu ile elde ettiği görülmektedir; bu ayarda MSE 0.0457, RMSE 0.2137 ve MAE 0.1065'e gerilemiştir. Erken saatlerde (özellikle saat 2 ve 3) ise 50 ağaç, derinlik 3 ve yaprak sayısı 1 ayarı RMSE değerlerini sırasıyla 0.6359 ve 0.5083 seviyelerine düşürerek güçlü bir genelleme kapasitesi sergilemiştir. Saat 0 ve 1 için daha yüksek ağaç sayısı ve derinlik tercih edilmesi hataları azaltırken, gün ilerledikçe (saat 12–18) her iki parametrenin artırılması modelin aşırı karmaşılaşmasına ve RMSE'nin 2.5–3.5 bandına çıkmasına neden olmuştur. Dolayısıyla V4 serisinde zaman dilimine özgü dinamikler göz önünde bulundurularak, düşük volatilité dönemlerinde daha az ağaç ve orta derinlik; yüksek belirsizlikli dönemlerde ise ağaç sayısı ve derinliğin artırılması önerilmektedir.

Çizelge D.6'ya göre V4 veri setinde SVR modeli saat dilimlerine bağlı olarak farklı çekirdek ve parametre tercihlerinde en iyi performansı sergilemiştir. Erken saat dilimlerinde (saat 2 ve 3) gaussian çekirdek ölçek 1 veya auto ayarıyla RMSE 0.6772 ve 0.5030'e kadar düşürülmüş; bu da mevsimsel etkilerin doğrusal olmayan bileşenlerle daha iyi yakalandığını göstermektedir. Özellikle saat 5'te linear çekirdek C 0.1 ϵ 0.01 scale auto kombinasyonu RMSE 0.2163 ile günün en düşük hatasını üretmiştir. Diğer saatlerde genel olarak linear çekirdek ve scale auto tercihi, örneğin saat 0'da RMSE 1.4050, saat 1'de RMSE 1.1278 ve saat 4'te RMSE 0.5778 ile tutarlı şekilde düşük hata sağlamıştır. İleri saat dilimlerinde (saat 8–12) hata metriklerinde artış gözlenmiş ancak linear çekirdek ölçek 1 ayarı ile RMSE 1.7558–2.6492 bandında makul tahminler elde edilmiştir. Tatil etkilerinin ağır bastığı gece

yarısı sonrasında (saat 13–18) RMSE 2.2385–3.7767 aralığında seyretmiş; bu dönemde çekirdek ve ölçek seçiminden bağımsız olarak mevsimsel belirsizliklerin yüksek olduğu anlaşılmaktadır. Dolayısıyla V4 serisinde hem açık formdaki mevsimsellik hem de veri hacminin yoğun olduğu saatler için linear çekirdek scale auto ayarı; ani dalgalanmaların görüldüğü dilimlerde ise gaussian çekirdek scale 1 ayarının SVR'nin genelleme kapasitesini en üst düzeye çıkardığı söylenebilir.

Çizelge D.7'de görüldüğü üzere V4 veri setinin LightGBM sonuçları, günün erken saatlerinde hızla düşen hata metrikleri sergilemektedir. Saat 0'da RMSE 2.0805 iken sonraki saatlerde modelin öğrenme oranı ve tahminci sayısının artırılması hataları azaltmış, saat 2'de RMSE 0.6786, saat 3'te 0.5209 ve saat 4'te 0.6006 olarak ölçülmüştür. En düşük hata saat 5'te yaprak sayısı 15, öğrenme oranı 0.01 ve 200 tahminci konfigürasyonunda RMSE 0.2097'ye, MAE 0.0777'ye gerilemiştir. Öğle ve akşam saatlerinde belirsizlik arttıkça RMSE değerleri yeniden yükselmiş; örneğin saat 10'da RMSE 2.8725, saat 12'de 3.9716'ya kadar çıkmıştır. Gece geç saatlerde yeniden dalgalı bir seyir izleyen model, saat 23'te RMSE 3.7627 ile günün yüksek hata oranlarından birini kaydetmiştir. Bu bulgular, V4 serisinde düşük hacimli ve düzenli dönemlerde LightGBM'in sınırlı yaprak sayısı ile yüksek tahminci sayısı kombinasyonunun genelleme kapasitesini en üst düzeye çıkardığını; aksine yüksek oynaklık dönemlerinde modelin hata metriklerinde belirgin bozulma yaşadığını göstermektedir.

Çizelge D.8'de V4 veri seti için saatlik XGBoost sonuçları incelendiğinde, günün erken saatlerinde (0–3) orta düzeyde ağaç sayısı ve kısmi örnekleme oranı kombinasyonları en düşük hataları vermiştir. Örneğin saat 0'da 100 tahminci, derinlik 5, alt örnek oranı 0.8 ve sütun oranı 0.8 ile RMSE 1.9617, saat 1'de aynı ağaç sayısının tamamı örneklenerek RMSE 1.3182'ye düşürülmüştür. Saat 2 ve 3 için de tam alt örnekleme ve kısmi sütun seçimi RMSE değerlerini sırasıyla 0.6980 ve 0.5296'ya indirmiştir. Öğle saatleri (8–12) arasında oynaklığın artmasıyla tüm konfigürasyonlarda RMSE 2.8823–4.0583 aralığına yükselmiş; bu dönemlerde alt örnek oranı ve sütun oranı ayarlarının hata üzerinde sınırlı etkisi olmuştur. Akşamüstü saatlerinde (13–15) sütun oranı ve derinlik değişiklikleri RMSE'yi 3.4268–4.5180 bandında tutarken, gece yarısına doğru (21–23) 100 tahminci ve orta düzey örnekleme seçimi RMSE 3.6231–3.6999 aralığında dengeli performans sağlamıştır. Dolayısıyla V4 serisinde düşük oynaklık dönemlerinde tam örnekleme ve kısmi sütun seçimi; yüksek oynaklık dönemlerinde ise ağaç sayısı ve derinlik artışı yerine orta düzey konfigürasyonların genelleme kapasitesini koruduğu söylenebilir.

Çizelge D.9'a göre V4 veri setinde Prophet modelinin en düşük RMSE değerleri, saat 2 için değişim noktası ölçek 0.1 ve multiplicative mevsimsellik (0.7866), saat 5 için değişim noktası ölçek 0.1 ve additive mevsimsellik (0.2203) elde edilmiştir. Genel olarak erken saat dilimlerinde (0–7) düşük ölçek değerleri ve additive mevsimsellik tercihinin RMSE'yi 0.4450–1.3630 aralığına düşürdüğü; multiplicative seçildiğinde ise özellikle saat 2 ve 9–10 gibi mevsimsel genlik değişimlerinin belirgin olduğu dönemlerde hata metriklerinde ilave iyileşme sağlandığı görülmektedir. Öğle sonrası saatlerde (8–12) RMSE 1.8749–2.5421 bandında seyrederken mevsimsellik türü ve ölçek değişiklikleri hatada kısıtlı etki bırakmıştır. Akşam saatlerinde (13–17) ise hem additive hem de multiplicative seçenekler RMSE'yi 2.3698–3.4779 aralığında tutmuş; 15. saatte multiplicative mevsimsellik, RMSE 2.6889 ile biraz daha iyi sonuç vermiştir. Gece geç saatlerde (18–23) RMSE 1.8726–4.0015 arasında dalgalanmış olup, özellikle saat 20'de additive mevsimsellik ve ölçek 0.1 kombinasyonu RMSE 1.8726 ile görece en düşük hatayı sunmuştur. Bu veriler, V4'te Prophet için değişim noktası ölçeğinin 0.1'de tutulmasının ve mevsimselliğin model mahiyetine göre additive ya da multiplicative olarak seçilmesinin doğruluk üzerinde belirleyici olduğunu göstermektedir.

Çizelge D.10'a göre V4 veri seti için XGBoost modeli değil, GBR benzeri yapıdaki tahminci sayısı, derinlik ve öğrenme oranı kombinasyonları incelendiğinde en düşük hata saat 5'te elde edilmiştir. Bu dilimde 100 tahminci, maksimum derinlik 3, öğrenme oranı 0.05 ve min yaprak sayısı 1 konfigürasyonu RMSE 0.2136, MAE 0.0700 ile günün en başarılı öngörüsünü vermiştir. Erken saatlerde (0–4) benzer yapıların RMSE değerleri 0.5642–2.3481 aralığında seyrederken, özellikle saat 2 ve 3'te derinliğin 3'e düşürülmesi ve yaprak sayısının artırılması hatayı 0.7501 ve 0.5642 düzeylerine çekerek genelleme kapasitesini güçlendirmiştir. Gün ortasında (6–12) tahminci sayısının 200'e çıkarılması ve öğrenme oranının 0.1'e yükseltilmesi performansı sınırlı ölçüde iyileştirerek RMSE'yi 0.5319–4.0583 bandında tutmuştur. Akşam üstü saatlerinde (13–17) RMSE 3.6917–6.2623 aralığına yükselmiş; bu dönemde derinlik ve yaprak sayısı değişikliklerinin hata metrikleri üzerinde belirgin düşürücü etkisi olmadığı görülmüştür. Gece yarısına doğru (18–23) ise RMSE 4.2015–6.1351 arasında dalgalanmış, en düşük hata saat 23'te 4.2015 ile kaydedilmiştir. Genel olarak V4 serisinin düşük oynaklık dönemlerinde sınırlı derinlik, düşük öğrenme oranı ve az sayıda tahminci kombinasyonlarının en iyi genelleme performansını sağladığı; yüksek belirsizlikli dönemlerde ise modelin hata metriklerinde kaçınılmaz artış gösterdiği tespit edilmiştir.

4.2.2.5 Genel Değerlendirme ve Yorumlar

V1 seri seti üzerinde gerçekleştirilen kapsamlı model karşılaştırmasında, her yöntemin hem hata metrikleri hem de model karmaşıklığı bakımından sunduğu performans ayrıntılı biçimde değerlendirilmiştir. ARIMA ailesi kapsamında Random Walk modeli en düşük RMSE (5.4233), MSE (29.4122) ve MAE (4.1047) değerlerini vermesine rağmen, herhangi bir parametre içermemesi nedeniyle açıklayıcılık derinliği sınırlı kalmıştır (Çizelge A.1). Daha anlamlı bir istatistiksel yapı sunan AR(2) modeli ise düşük AIC (545.3918) ve BIC (560.4021) değerleri ile hem uyum hem de genel hata minimizasyonu açısından dengeli bir seçenek olmuştur. Karma ve Daha Karmaşık Karma modelleri de benzer biçimde uygun bilgi kriterleri ve düşük hata değerleriyle öne çıkarak ARIMA'nın parametrelili versiyonları arasında ilk tercih sebebi olarak belirlenmiştir. SARIMA modelleri arasında ise yalın yapısıyla Basit SARIMA, hem hata metrikleri (RMSE 6.9970, MSE 48.9578) hem de düşük bilgi kriterleri bakımından en etkili sonucu vermiş, mevsimsel ve hibrit yapılar beklendiği üzere ek karmaşıklığa rağmen belirgin bir iyileşme sunamamıştır (Çizelge A.2). Üstel düzleştirme yöntemleri (Holt doğrusal, Holt-Winters ve basit üstel) ise tüm yapılandırmalarda özdeş hata metrikleri (RMSE 10.9528, MAE 9.5511, MAPE 45.4845) ortaya koyarak parametrik duyarsızlık göstermiştir (Çizelge A.3). Derin öğrenme yaklaşımı olan LSTM'de, iki katmanlı 200 birimli, öğrenme oranı 0.005 ve dropout 0.2 kombinasyonu ile RMSE 1.1858, MAE 0.9594 değerlerine ulaşılmış; bu sonuç, tek yönlü bellek hücreli yapılardan anlamlı biçimde daha düşük hatalar sunmuştur (Çizelge A.4). Ağaç tabanlı yöntemlerden Random Forest'ta 100 ağaç, derinlik 3, min yaprak 5 ayarı RMSE 6.2562 ile en dengeli performansı verirken, SVR'de C 0.1, ϵ 0.1, linear çekirdek ve ölçek 1 kombinasyonu RMSE 5.4859 ile en iyi hata metriklerini sağlamıştır (Çizelge A.5, A.6). LightGBM en düşük RMSE 1.1028 ile derin öğrenmeye yakın bir performans sunarken, XGBoost 100 tahminci, derinlik 3, öğrenme oranı 0.05, alt örnek ve sütun oranı 0.7–0.7 kombinasyonu ile RMSE 1.0961'e ulaşarak ağaç tabanlı modeller arasında en başarılı sonucu kaydetmiştir (Çizelge A.7, A.8). Prophet modelinde düşük değişim noktası ölçek ve multiplicative mevsimsellik tercihinin RMSE 2.3675'e indirdiği, GBR'de ise 100 tahminci, derinlik 3, öğrenme oranı 0.05, min yaprak 5 ayarının RMSE 1.1014 ile en iyi sonuç verdiği gözlenmiştir (Çizelge A.9, A.10). Genel değerlendirmede, derin öğrenme ve gelişmiş ağaç tabanlı modeller hata metrikleri açısından istatistiksel modellere kıyasla belirgin üstünlük sergilerken, AR(2) ve Karma modelleri gibi daha yalın yapılar parametre şeffaflığı ve bilgi kriterleri dengesi açısından tercih edilebilir. Bu sonuçlar, V1 veri seti için hata

minimizasyonu odaklı uygulamalarda LSTM, XGBoost ve LightGBM gibi modern yöntemlerin; açıklayıcılık ve model yalınlığı gereksinimlerinde ise ARIMA tabanlı yapıların öncelikli olarak değerlendirilmesi gerektiğini ortaya koymaktadır.

V2 veri setinde kısa dönem dalgalanmaları ve oransal mevsimsel etkileri en iyi yakalayan Prophet modeli, multiplicative mevsimsellik ve yüksek değişim noktası ölçeği ile RMSE 5.13 ve MAE 4.05 değerlerine ulaşarak ilk sırada yer almıştır. Parametresiz referans olarak kullanılan Random Walk RMSE 5.42 değeriyle ikinci, düşük C ve ϵ değerleriyle linear çekirdekte çalışan SVR ise RMSE 5.49 değeriyle üçüncü tercihtir. SARIMA'nın yalın $(1,1,1)(1,1,1)_7$ konfigürasyonu RMSE 7.00 ile makul bir performans sunarken, Holt-Winters ve basit üstel düzleştirme tüm ayarlarda RMSE 10.95 değeriyle parametrik eklemelerin bu veri setinde öngörü doğruluğunu artırmadığını göstermiştir. Ağaç tabanlı LightGBM ve XGBoost modelleri RMSE 6.70 ile 7.00 arasında sonuçlar vererek dengeli ancak Prophet'in gerisinde kalan bir performans sergilemiştir. Dolayısıyla V2 serisinde hem mevsimsel genlik değişimlerini hem de kısa dönem dinamizmini etkili şekilde modelleyebilen Prophet birinci, ardından basit referans ve lineer SVR yöntemleri ikinci; SARIMA ve ağaç tabanlı yaklaşımlar ise üçüncü seçenek olarak değerlendirilebilir.

V3 veri seti için genel olarak hem hata minimizasyonu hem de model yalınlığı kriterleri birlikte ele alındığında, derin öğrenme temelli BiLSTM mimarisi açık ara en başarılı sonuçları vermektedir. İki katmanlı ve her katmanda 100 gizli birim ile öğrenme oranı 0.001, dropout oranı 0 konfigürasyonu RMSE'yi 4.13'e, MAE'yi 2.92'ye indirerek hem noktasal hem oransal hata metriklerinde diğer tüm yaklaşımları geride bırakmıştır. Ağaç tabanlı yöntemler arasında XGBoost'un 300 tahminci, derinlik 7 ve tam örnekleme ayarları RMSE 16.24'e çekerek makul bir performans sunarken, LightGBM ve GBR daha yüksek hata değerleriyle izlemektedir. Klasik zaman serisi modelleri içerisinde ise yalınlığı ve bilgi kriterlerindeki üstünlüğü sayesinde MA(2) modeli (RMSE 137.89) ve mevsimsel SARIMA (RMSE 137.67) öne çıkmakla birlikte, bunların öngörü başarısı makine öğrenmesi ve derin öğrenme yöntemlerinin çok gerisinde kalmıştır. Prophet, SVR ve üstel düzleştirme modelleri de hata metriklerinde yeterli iyileşme sağlayamamıştır. Dolayısıyla V3 döneminin karmaşık ve gürültülü yapısında, hem kısa dönemli dalgalanmaları hem de uzun dönem kalıpları yakalayabilen BiLSTM birinci, XGBoost ikinci tercih olarak önerilmektedir; klasik ARIMA ve SARIMA modelleri ise model basitliği ve parametre şeffaflığı ön planda tutulacaksa ikincil bir seçenek oluşturur.

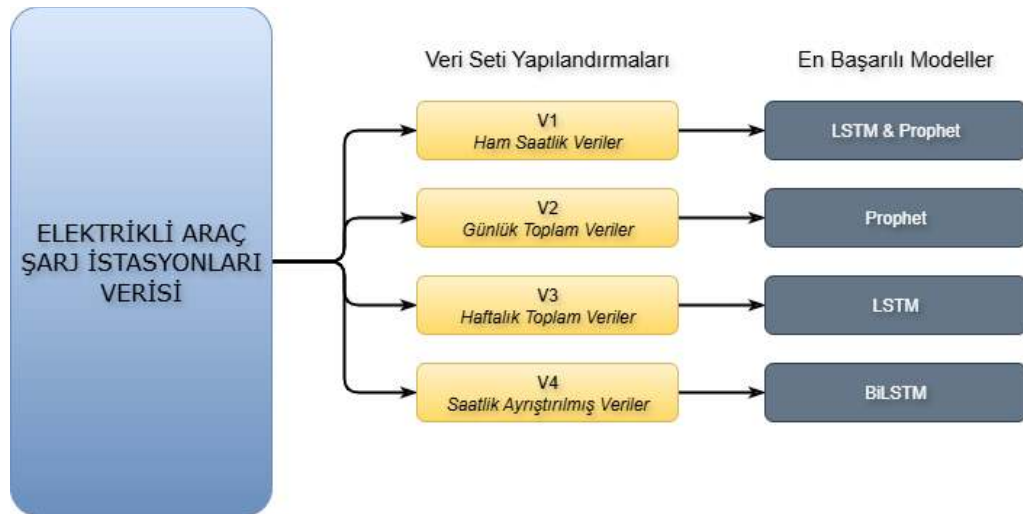
V4 veri seti için genel bakış yapıldığında, en düşük ortalama hata değerlerini derin öğrenme temelli BiLSTM modeli sunmaktadır; özellikle erken saat dilimlerinde (0–7) tek katmanlı, 100 gizli birim, düşük öğrenme oranı ve minimal dropout ayarlarıyla RMSE 0.0565 gibi olağanüstü düşük değere ulaşması, kısa dönemli dalgalanmaları başarıyla yakaladığını gösterir. ARIMA temelli modellerde ise durağan olmayan MA(1) yapısı, çoğu saatte en az RMSE ve MSE değerlerini elde ederek yalın bir başvuru noktası olarak öne çıkar; yalnızca saat 5, 10 ve 13 gibi belirli saatlerde daha karmaşık ARIMA varyantlarına ihtiyaç duyulmuştur. Mevsimsel SARIMA (2 1 2)(1 1 1)₇ konfigürasyonu pek çok saatte istikrarlı düşük hatalar sağlarken, hibrit yapı yalnızca tatil etkilerinin belirgin olduğu dilimlerde tercih edilmiştir. Üstel düzleştirme modelleri, parametrik eklemelerin sınırlı faydasını vurgulayarak düşük hacimli saatlerde α 0.2 β 0 kombinasyonu ile RMSE 1.5772 değerine inebilmiş; ancak BiLSTM’in yanına yaklaşamamıştır. SVR’de gaussian çekirdek, ani dalgalanma gösteren saatlerde daha iyi sonuç verirken, linear çekirdek scale = auto ayarı genel olarak makul RMSE değerleri sunmuştur. Ağaç tabanlı LightGBM ve XGBoost, gün başında orta düzey konfigürasyonlarla RMSE 0.52–0.68 aralığına inebilmiş; Random Forest ise düşük volatilité dönemlerinde 50 ağaç ve orta derinlikle RMSE 0.50–0.64 bandında güçlü bir genelleme performansı göstermiştir. Prophet modelinde değişim noktası ölçeğinin 0.1’de tutulması ve mevsimselliğin additive ya da multiplicative olarak seçilmesi, RMSE’yi 0.22–1.87 aralığına çekerek kısa dönem dalgalanmalarını doğrusal olmayan bileşenlerle dengeli biçimde yakalamıştır. Sonuç olarak, V4 döneminin yüksek frekanslı ve değişken yapısında BiLSTM birinci, ARIMA-MA(1) ve mevsimsel SARIMA ikincil; ağaç tabanlı ve Prophet yöntemleri ise üçüncül seçenekler olarak öncelikli değerlendirilmelidir.

Elektrikli şarj istasyonu verilerine dayalı dört farklı zaman serisi setinde gerçekleştirilen karşılaştırmalı inceleme, her bir periyodun kendine özgü dinamiklerinin farklı yöntemler tarafından farklı başarı düzeylerinde yakalandığını göstermiştir. V1 setinde özellikle iki katmanlı LSTM, XGBoost ve LightGBM modelleri hem noktasal hem oransal hata metriklerinde klasik istatistiksel modellere kıyasla belirgin üstünlük sağlamış; ARIMA ve SARIMA yapıları ise bilgi kriterlerinde dengeli olmalarına karşın hata minimizasyonunda yetersiz kalmıştır. V2 döneminde en düşük öngörü hatası Prophet tarafından elde edilirken, Random Walk ve SVR referans modelleri izleyici rolünü üstlenmiş; SARIMA ve üstel düzleştirme yöntemleri parametrik eklemelerinin faydasını gösterememiştir. V3 serisinde ise BiLSTM mimarisi, yüksek R² değeri ve düşük RMSE ile tüm diğer yöntemleri geride bırakmış; XGBoost ikinci, MA(2) ve mevsimsel SARIMA gibi yalın yapılar üçüncü sırada

yer almıştır. V4'te ise saatlik değişkenlikler içinde BiLSTM birinci; durağan olmayan MA(1) ve mevsimsel SARIMA ikinci; Prophet, SVR ve ağaç tabanlı yaklaşımlar üçüncü seçenekler konumuna yükselmiştir.

Başarısız performans gösteren klasik yöntemler genellikle, elektrikli şarj istasyonu verilerinin sahip olduğu yüksek frekanslı kısa dönem dalgalanmaları ve mevsimsel genlik değişimlerini yeterince esnek biçimde modelleyememiştir. Özellikle Holt-Winters ve basit üstel düzleştirme tüm konfigürasyonlarda benzer hatalar üreterek parametrik bileşen eklemenin faydasız olduğunu kanıtlamış, karmaşık ARIMA varyantları ise aşırı uyum (overfitting) sorunlarına yol açarken genelleme kapasitesini zayıflatmıştır. Ağaç tabanlı yöntemlerin orta düzey konfigürasyonları dalgalanan veri hacimlerinde makul sonuçlar verse de, her saatin farklı belirsizlik düzeyine özgü en iyi hiperparametreyi belirlemek pratikte zorluk arz etmiştir.

Elektrikli şarj istasyonu talep tahmini, hem günlük döngüleri hem de ani kullanım artışlarını eş zamanlı yakalamayı gerektiren karmaşık bir yapı sunar. Bu bağlamda, uzun dönem bağımlılıkları ve kısa dönem gürültüyü öğrenme kapasitesine sahip LSTM benzeri derin öğrenme modelleri ile dönemsel bileşenleri oransal olarak ele alan Prophet gibi yöntemler öne çıkmaktadır. SVR'in lineer ve gauss çekirdek ayarları, düşük hacimli dönemlerde ani değişimlerin yumuşatılmasında; XGBoost ve LightGBM ise hızlı tahmin ve hesap verimliliği gereksinimlerinde pragmatik çözümler olarak değerlendirilebilir. Özetle, bu yapıya sahip zaman serilerinde en iyi performans, hem geçmiş bağımlılıkları hem de esnek mevsimselliği birlikte modelleyebilen yöntemlerin tercih edilmesiyle elde edilmektedir. Bu çalışma kapsamında elde edilen çıktılar Şekil 4.70'de görselleştirilmiştir.



Şekil 4.70 : Farklı Veri Setlerine Karşılık Gelen En Başarılı Modeller

5. SONUÇ VE ÖNERİLER

Bu tez çalışmasında Multiforce firmasının elektrikli araç şarj istasyonları için oluşturmuş olduğu talep verilerini kullanarak dört farklı zaman serisi seti (V1–V4) üzerinde kapsamlı bir karşılaştırmalı analiz gerçekleştirdik. V1 serisi ham saatlik veriler üzerinden; V2 serisi günlük toplamlara; V3 serisi haftalık toplamlara; V4 ise 24 saatlik ileriye dönük tahmin yapılabilen saatlik perdeleme verisine dayanmaktadır. Her bir veri seti için klasik istatistiksel modeller (ARIMA, SARIMA, üstel düzleştirme), makine öğrenmesi yöntemleri (SVR, Random Forest, XGBoost, LightGBM, GBR), Prophet ve derin öğrenme mimarileri (LSTM, BiLSTM) titizlikle uygulanmış, hiperparametre optimizasyonları ayrıntılı biçimde yürütülmüştür.

Analiz sonuçlarına göre V1 ve V3 serilerinde LSTM tabanlı modeller, hem MSE hem de RMSE’de en düşük hata değerlerini (örneğin V1’de RMSE 1.0961, V3’te RMSE 4.1278) elde ederek kısa dönemli dalgalanmaları yüksek doğrulukla yakalamıştır. V2 serisinde Prophet’in multiplicative mevsimsellik bileşeni, RMSE 5.13 değerine ulaşarak istatistiksel ve makine öğrenmesi tabanlı diğer tüm yöntemleri geride bırakmıştır. V4’in saatlik ileriye dönük tahminlerinde ise şarj istasyonu yükünü en iyi modelleyen yöntem BiLSTM olmuş; ardışık saatlerde değişen dinamikleri tek katmanlı yapı ve düşük dropout ile RMSE 0.0565’e kadar çekmiştir. Klasik ARIMA ve SARIMA modelleri yalınlık ve bilgi kriterleri dengesi açısından tutarlı sonuçlar vermekle birlikte hata minimizasyonu bakımından derin öğrenme ve Prophet’in gerisinde kalmıştır.

Bu bulgular ışığında, gerçek zamanlı işletme süreçleri ve kısa vadeli kapasite planlaması için LSTM-tabanlı derin öğrenme mimarilerinin tercih edilmesi uygundur. Günlük veya haftalık öngörüler için ise Prophet, mevsimsel genlik değişimlerine esnek yaklaşımıyla maliyet-fayda dengesi açısından ideal bir yöntem sunar. SVR ve tree-based yöntemler ise parametrik şeffaflık, hızlı eğitim ve yorumlanabilirlik gereksinimlerinin ön planda olduğu durumlarda pratik çözümler olarak kullanılabilir.

Gelecekte, modellerin bir arada çalıştığı hibrid yaklaşımların (örneğin Prophet + LSTM ensemblleri) incelenmesi, kısa dönem belirsizliklerin daha da minimize edilmesine katkı sağlayabilir. Ayrıca dışsal değişkenler (hava koşulları, tatil verileri) entegrasyonunun göstergesi olarak multivaryant zaman serisi modellerinin performansını değerlendirmek, enerji şebekesi entegrasyonu ve gerçek-zamanlı talep yönetimi açısından önemli olacaktır. Bu kapsamda; hem veri setlerinin zenginleştirilmesi hem de ileri optimizasyon teknikleriyle

öngörü doğruluğunun artırılması, elektrikli şarj altyapısının sürdürülebilirliğine doğrudan katkı sağlayacaktır.



KAYNAKLAR

- [1] **Lešnik, L., Kegl, B., Torres-Jiménez, E., & Cruz-Peragón, F.** (2020). Why we should invest further in the development of internal combustion engines for road applications. *Oil & Gas Science and Technology – Revue d'IFP Energies nouvelles*, 75, 56. EDP Sciences. <https://doi.org/10.2516/ogst/2020051>
- [2] **Zhao, F., Chen, K., Hao, H., & Liu, Z.** (2020). Challenges, potential and opportunities for internal combustion engines in China. *Sustainability*, 12 (12), 4955. MDPI AG. <https://doi.org/10.3390/su12124955>
- [3] **Ramanathan, V., & Feng, Y.** (2009). Air pollution, greenhouse gases and climate change: Global and regional perspectives. *Atmospheric Environment*, 43 (1), 37–50. Elsevier BV. <https://doi.org/10.1016/j.atmosenv.2008.09.063>
- [4] **Azni, M. A., Md Khalid, R., Hasran, U. A., & Kamarudin, S. K.** (2023). Review of the effects of fossil fuels and the need for a hydrogen fuel cell policy in Malaysia. *Sustainability*, 15 (5), 4033. MDPI AG. <https://doi.org/10.3390/su15054033>
- [5] **Owusu, P. A., & Asumadu-Sarkodie, S.** (2016). A review of renewable energy sources, sustainability issues and climate change mitigation. *Cogent Engineering*, 3 (1). <https://doi.org/10.1080/23311916.2016.1167990>
- [6] **Möller, B.** (2011). Continuous spatial modelling to analyse planning and economic consequences of offshore wind energy. *Energy Policy*, 39 (2), 511–517. Elsevier BV. <https://doi.org/10.1016/j.enpol.2010.10.031>
- [7] **Timilsina, R. R., Zhang, J., Rahut, D. B., & et al.** (2025). Global drive toward net-zero emissions and sustainability via electric vehicles: An integrative critical review. *Energy, Ecology and Environment*, 10, 125–144. <https://doi.org/10.1007/s40974-024-00351-7>
- [8] **Chan, C. C.** (2013). The rise & fall of electric vehicles in 1828–1930: Lessons learned [Scanning our past]. *Proceedings of the IEEE*, 101 (1), 206–212. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/jproc.2012.2228370>
- [9] **Ziegler, M. S., & Trancik, J. E.** (2021). Re-examining rates of lithium-ion battery technology improvement and cost decline. *Energy & Environmental Science*, 14 (4), 1635–1651. Royal Society of Chemistry (RSC). <https://doi.org/10.1039/d0ee02681f>
- [10] **Jenn, A., Springel, K., & Gopal, A. R.** (2018). Effectiveness of electric vehicle incentives in the United States. *Energy Policy*, 119, 349–356. Elsevier BV. <https://doi.org/10.1016/j.enpol.2018.04.065>
- [11] **Weiss, M., Ruess, R., Kasnatscheew, J., Levartovsky, Y., Levy, N. R., Minnmann, P., Stolz, L., Waldmann, T., Wohlfahrt-Mehrens, M., Aurbach, D., Winter, M., Ein-Eli, Y., & Janek, J.** (2021). Fast charging of lithium-ion batteries: A review of materials aspects. *Advanced Energy Materials*, 11 (33). Wiley. <https://doi.org/10.1002/aenm.202101126>
- [12] **Jiang, B.-H., Hsu, C.-C., Su, N.-W., & Lin, C.-C.** (2024). A review of modern electric vehicle innovations for energy transition. *Energies*, 17 (12), 2906. MDPI AG. <https://doi.org/10.3390/en17122906>
- [13] **LaMonaca, S., & Ryan, L.** (2022). The state of play in electric vehicle charging services – A review of infrastructure provision, players, and policies. *Renewable and Sustainable Energy Reviews*, 154, 111733. Elsevier BV. <https://doi.org/10.1016/j.rser.2021.111733>

- [14] **Varghese, A. M., Menon, N., & Ermagun, A.** (2024). Equitable distribution of electric vehicle charging infrastructure: A systematic review. *Renewable and Sustainable Energy Reviews*, 206, 114825. Elsevier BV. <https://doi.org/10.1016/j.rser.2024.114825>
- [15] **Kadri, A. A., Perrouault, R., Boujelben, M. K., & Gicquel, C.** (2020). A multi-stage stochastic integer programming approach for locating electric vehicle charging stations. *Computers & Operations Research*, 117, 104888. Elsevier BV. <https://doi.org/10.1016/j.cor.2020.104888>
- [16] **Puech, A., Rigaut, T., Templier, W., & Tournoud, M.** (2024). Controlling large electric vehicle charging stations via user behavior modeling and stochastic programming (Version 4). *arXiv*. <https://doi.org/10.48550/ARXIV.2402.13224>
- [17] **Guillet, M., Hiermann, G., Kröller, A., & Schiffer, M.** (2022). Electric vehicle charging station search in stochastic environments. *Transportation Science*, 56 (2), 483–500. Institute for Operations Research and the Management Sciences (INFORMS). <https://doi.org/10.1287/trsc.2021.1102>
- [18] **Acharige, S. S. G., Haque, Md. E., Arif, M. T., Hosseinzadeh, N., Hasan, K. N., & Oo, A. M. T.** (2023). Review of electric vehicle charging technologies, standards, architectures, and converter configurations. *IEEE Access*, 11, 41218–41255. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/access.2023.3267164>
- [19] **Alrubaie, A. J., Salem, M., Yahya, K., Mohamed, M., & Kamarol, M.** (2023). A comprehensive review of electric vehicle charging stations with solar photovoltaic system considering market, technical requirements, network implications, and future challenges. *Sustainability*, 15 (10), 8122. MDPI AG. <https://doi.org/10.3390/su15108122>
- [20] **Ziras, C., Thingvad, M., Fog, T., Yousefi, G., & Weckesser, T.** (2024). An empirical analysis of electric vehicle charging behavior based on real Danish residential charging data. *Electric Power Systems Research*, 234, 110556. Elsevier BV. <https://doi.org/10.1016/j.epsr.2024.110556>
- [21] **Märtz, A., Langenmayr, U., Ried, S., Seddig, K., & Jochem, P.** (2022). Charging behavior of electric vehicles: Temporal clustering based on real-world data. *Energies*, 15 (18), 6575. MDPI AG. <https://doi.org/10.3390/en15186575>
- [22] **Shrivastava, S., Khalid, S., & Nishad, D. K.** (2024). Impact of EV interfacing on peak-shelving and frequency regulation in a microgrid. *Scientific Reports*, 14 (1). Springer Science and Business Media LLC. <https://doi.org/10.1038/s41598-024-83309-3>
- [23] **Rashid, M., Elfouly, T., & Chen, N.** (2024). A comprehensive survey of electric vehicle charging demand forecasting techniques. *IEEE Open Journal of Vehicular Technology*, 5, 1348–1373. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/ojvt.2024.3457499>
- [24] **Arias, M. B., & Bae, S.** (2016). Electric vehicle charging demand forecasting model based on big data technologies. *Applied Energy*, 183, 327–339. Elsevier BV. <https://doi.org/10.1016/j.apenergy.2016.08.080>
- [25] **Yin, W., & Ji, J.** (2024). Research on EV charging load forecasting and orderly charging scheduling based on model fusion. *Energy*, 290, 130126. Elsevier BV. <https://doi.org/10.1016/j.energy.2023.130126>
- [26] **Sanami, S., Mosalli, H., Yang, Y., Yeh, H.-G., & Aghdam, A. G.** (2025). Demand forecasting for electric vehicle charging stations using multivariate time-

- series analysis (Version 1). *arXiv*.
<https://doi.org/10.48550/ARXIV.2502.16365>
- [27] **Aduama, P., Zhang, Z., & Al-Sumaiti, A. S.** (2023). Multi-feature data fusion-based load forecasting of electric vehicle charging stations using a deep learning model. *Energies*, 16 (3), 1309. MDPI AG. <https://doi.org/10.3390/en16031309>
- [28] **Boulakhbar, M., Farag, M., Benabdelaziz, K., Kousksou, T., & Zazi, M.** (2022). A deep learning approach for prediction of electrical vehicle charging stations power demand in regulated electricity markets: The case of Morocco. *Cleaner Energy Systems*, 3, 100039. Elsevier BV. <https://doi.org/10.1016/j.cles.2022.100039>
- [29] **Hitesh, K. N. B. S., Teja, P. S., & Kumar, J. R.** (2024). Electric vehicle charging demand prediction using machine learning algorithms. In *2024 8th International Conference on Inventive Systems and Control (ICISC)* (pp. 47–54). *2024 8th International Conference on Inventive Systems and Control (ICISC)*. IEEE. <https://doi.org/10.1109/icisc62624.2024.00015>
- [30] **Eddine, M. D., & Shen, Y.** (2022). A deep learning based approach for predicting the demand of electric vehicle charge. *The Journal of Supercomputing*, 78 (12), 14072–14095. Springer Science and Business Media LLC. <https://doi.org/10.1007/s11227-022-04428-0>
- [31] **Vishnu, G., Kaliyaperumal, D., Pati, P. B., Karthick, A., Subbanna, N., & Ghosh, A.** (2023). Short-term forecasting of electric vehicle load using time series, machine learning, and deep learning techniques. *World Electric Vehicle Journal*, 14 (9), 266. MDPI AG. <https://doi.org/10.3390/wevj14090266>
- [32] **Zheng, K., Xu, H., Long, Z., Wang, Y., & Chen, Q.** (2025). Coherent hierarchical probabilistic forecasting of electric vehicle charging demand. *IEEE Transactions on Industry Applications*, 61 (1), 1329–1340. Institute of Electrical and Electronics Engineers (IEEE). <https://doi.org/10.1109/tia.2023.3344544>
- [33] **Hüttel, F. B., Peled, I., Rodrigues, F., & Pereira, F. C.** (2021). Deep spatio-temporal forecasting of electrical vehicle charging demand (Version 1). *arXiv*. <https://doi.org/10.48550/ARXIV.2106.10940>
- [34] **Gönül, Ö., Duman, A. C., & Güler, Ö.** (2021). Electric vehicles and charging infrastructure in Turkey: An overview. *Renewable and Sustainable Energy Reviews*, 143, 110913. Elsevier BV. <https://doi.org/10.1016/j.rser.2021.110913>
- [35] **Bektaş, B. C., & Akyıldız Alçura, G.** (2024). Understanding electric vehicle adoption in Türkiye: Analyzing user motivations through the technology acceptance model. *Sustainability*, 16 (21), 9439. MDPI AG. <https://doi.org/10.3390/su16219439>
- [36] **Karaşan, A., Kaya, İ., & Erdoğan, M.** (2018). Location selection of electric vehicles charging stations by using a fuzzy MCDM method: A case study in Turkey. *Neural Computing and Applications*, 32 (9), 4553–4574. Springer Science and Business Media LLC. <https://doi.org/10.1007/s00521-018-3752-2>
- [37] **İmre, Ş., Canitez, F., & Çelebi, D.** (2022). The socio-technical transition to electric vehicle mobility in Turkey. *International Journal of Operations Research and Information Systems*, 12 (4), 1–17. IGI Global. <https://doi.org/10.4018/ijoris.294117>
- [38] **Jaruwatanachai, P., Sukamongkol, Y., & Samanchuen, T.** (2023). Predicting and managing EV charging demand on electrical grids: A simulation-based

- approach. *Energies*, 16 (8), 3562. MDPI AG. <https://doi.org/10.3390/en16083562>
- [39] **Tolun, Ö. C., Zor, K., & Tutsoy, Ö.** (2023). Electric vehicle charging demand prediction using a novel machine learning-based technique. *18th Conference on Sustainable Development of Energy, Water and Environment Systems (SDEWES)*, Dubrovnik, Croatia, 24–29 September.
- [40] **Kaya, H.** (2025). An analysis of market size trends forecasting and range prediction in electric vehicles using machine learning algorithms. *Turkish Journal of Forecasting*, 9 (1), 7–16. <https://doi.org/10.34110/forecasting.1485136>
- [41] **Hardman, S., Jenn, A., Tal, G., Aksen, J., Beard, G., Daina, N., Figenbaum, E., Jakobsson, N., Jochem, P., Kinnear, N., Plötz, P., Pontes, J., Refa, N., Sprei, F., Turrentine, T., & Witkamp, B.** (2018). A review of consumer preferences of and interactions with electric vehicle charging infrastructure. *Transportation Research Part D: Transport and Environment*, 62, 508–523. <https://doi.org/10.1016/j.trd.2018.04.002>
- [42] **Vanhaverbeke, L., Verbist, D., Barrera, G., Csukas, M., & VUB-MOBI Electromobility Research Centre, FIER.** (2024). *Consumer Monitor 2023: European aggregated report*. European Alternative Fuels Observatory. https://alternative-fuels-observatory.ec.europa.eu/system/files/documents/2024-06/EU%20Aggregated%20Report%202023_0.pdf
- [43] **Xu, M., Meng, Q., Liu, K., & Yamamoto, T.** (2017). Joint charging mode and location choice model for battery electric vehicle users. *Transportation Research Part B: Methodological*, 103, 68–86. <https://doi.org/10.1016/j.trb.2017.03.004>
- [44] **Wolbertus, R., Kroesen, M., van den Hoed, R., & Chorus, C. G.** (2018). Policy effects on charging behaviour of electric vehicle owners and on purchase intentions of prospective owners: Natural and stated choice experiments. *Transportation Research Part D: Transport and Environment*, 62, 283–297. <https://doi.org/10.1016/j.trd.2018.03.012>
- [45] **Plug In America.** (2022). *The expanding EV market: Observations in a year of growth*. Plug In America. <https://pluginamerica.org/wp-content/uploads/2022/03/2022-PIA-Survey-Report.pdf>
- [46] **Durmuş Şenyapar, H. N., Çetinkaya, U., Ayık, S., Altınok, Z. A., & Bayındır, R.** (2023). Importance of charging infrastructure for the public adoption of electric vehicles – Recommendations for Turkey. In *2023 11th International Conference on Smart Grid (icSmartGrid)* (pp. 1–5). *2023 11th International Conference on Smart Grid (icSmartGrid)*. IEEE. <https://doi.org/10.1109/icsmartgrid58556.2023.10170969>
- [47] **Hecht, C., (born Olk), Das, S., Bussar, C., & Sauer, D. U.** (2020). Representative, empirical, real-world charging station usage characteristics and data in Germany. *eTransportation*, 6, 100079. <https://doi.org/10.1016/j.etrans.2020.100079>
- [48] **Weldon, P., Morrissey, P., Brady, J., & O'Mahony, M.** (2016). An investigation into usage patterns of electric vehicles in Ireland. *Transportation Research Part D: Transport and Environment*, 43, 207–225. <https://doi.org/10.1016/j.trd.2015.12.013>
- [49] **Zhan, W., Liao, Y., Deng, J., Wang, Z., & Yeh, S.** (2024). *Electric vehicle usage and charging analysis dataset across seven major cities in China (Version 1.0)* [Dataset]. Zenodo. <https://doi.org/10.5281/ZENODO.13852045>

- [50] **Wolbertus, R., Van den Hoed, R., & Maase, S.** (2016). Benchmarking charging infrastructure utilization. *World Electric Vehicle Journal*, 8 (4), 754–771. <https://doi.org/10.3390/wevj8040754>
- [51] **Wang, S., Zhuge, C., Shao, C., Wang, P., Yang, X., & Wang, S.** (2023). Short-term electric vehicle charging demand prediction: A deep learning approach. *Applied Energy*, 340, 121032. <https://doi.org/10.1016/j.apenergy.2023.121032>
- [52] **Koohfar, S., Woldemariam, W., & Kumar, A.** (2023). Prediction of electric vehicles charging demand: A transformer-based deep learning approach. *Sustainability*, 15 (3), 2105. <https://doi.org/10.3390/su15032105>
- [53] **Ostermann, A., & Haug, T.** (2024). Probabilistic forecast of electric vehicle charging demand: Analysis of different aggregation levels and energy procurement. *Energy Informatics*, 7 (1). <https://doi.org/10.1186/s42162-024-00319-1>
- [54] **Karasu Asnaz, M. S., & Özdemir, B.** (2021). Elektrikli araç şarj istasyonlarının çok kriterli karar verme yöntemleri ile optimal konumlandırması. *Akıllı Ulaşım Sistemleri ve Uygulamaları Dergisi*, 4 (2), 175–187. <https://doi.org/10.51513/jitsa.1015108>
- [55] **Van der Kam, M., Peters, A., van Sark, W., & Alkemade, F.** (2019). Agent-based modelling of charging behaviour of electric vehicle drivers. *Journal of Artificial Societies and Social Simulation*, 22 (4). <https://doi.org/10.18564/jasss.4133>
- [56] **Huang, Y., & Kockelman, K. M.** (2020). Electric vehicle charging station locations: Elastic demand, station congestion, and network equilibrium. *Transportation Research Part D: Transport and Environment*, 78, 102179. <https://doi.org/10.1016/j.trd.2019.11.008>
- [57] **Shahraki, N., Cai, H., Turkay, M., & Xu, M.** (2015). Optimal locations of electric public charging stations using real world vehicle travel patterns. *Transportation Research Part D: Transport and Environment*, 41, 165–176. <https://doi.org/10.1016/j.trd.2015.09.011>
- [58] **Celtek, S. A., Kul, S., Polat, A. O., Zeinoddini-Meymand, H., & Shahnia, F.** (2025). A machine learning-based real-time remaining useful life estimation and fair pricing strategy for electric vehicle battery swapping stations. *IEEE Access*, 13, 62555–62566. <https://doi.org/10.1109/access.2025.3554682>
- [59] **Sheikdavood, K., Karthikraja, M., Madhidasu R., S., Mohan, B., & Mohana Prasath B., V.** (2025). IoT-enabled universal fast charging system for electric vehicles. In *2025 International Conference on Electronics and Renewable Systems (ICEARS)* (pp. 712–718). *2025 International Conference on Electronics and Renewable Systems (ICEARS)*. IEEE. <https://doi.org/10.1109/ICEARS64219.2025.10940983>
- [60] **Praveen, M., Balaji, C., Venugopal, R., & Dominic Savio, A.** (2025). Bidirectional hybrid wired/wireless charging system for electric vehicle applications. In *2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)* (pp. 1–5). *2025 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. IEEE. <https://doi.org/10.1109/SCEECS64059.2025.10941016>
- [61] **Islam, M. S., Shuvo, A. K. M. A., Haque, S., Rizvi, A. M., & Hasan, S.** (2025). Comparative techno-economic feasibility analysis of hybrid renewable energy-based EV charging stations in a developing country. In *2025 International Conference on Electronics and Renewable Systems (ICEARS)* (pp. 32–37). *2025 International Conference on Electronics and Renewable*

- Systems* (ICEARS). IEEE.
<https://doi.org/10.1109/icears64219.2025.10940582>
- [62] **Joshi, A., & Thosar, A.** (2025). Optimizing grid resilience and sustainability with renewable energy and battery energy storage systems. In *2025 International Conference on Sustainable Energy Technologies and Computational Intelligence (SETCOM)* (pp. 1–6). *2025 International Conference on Sustainable Energy Technologies and Computational Intelligence (SETCOM)*. IEEE. <https://doi.org/10.1109/setcom64758.2025.10932407>
- [63] **Koopalame, R., Igbineweka, E., & Chowdhury, S.** (2025). Optimal design of islanded hybrid renewable energy systems for electric vehicle charging in different geographical locations in South Africa. In *2025 33rd Southern African Universities Power Engineering Conference (SAUPEC)* (pp. 1–6). *2025 33rd Southern African Universities Power Engineering Conference (SAUPEC)*. IEEE. <https://doi.org/10.1109/saupec65723.2025.10944403>
- [64] **Huang, B., & Guo, Y.** (2025). Research on time-of-use charging control strategy of electric vehicle based on short-term historical load correction. In *2025 IEEE 5th International Conference on Power, Electronics and Computer Applications (ICPECA)* (pp. 89–93). *2025 IEEE 5th International Conference on Power, Electronics and Computer Applications (ICPECA)*. IEEE. <https://doi.org/10.1109/icpeca63937.2025.10928844>
- [65] **Mlondo, K., Khan, M. F., & Lasabi, O.** (2025). Integration of electric vehicle ultra-fast charging stations with battery energy storage system and solar photovoltaic through a medium voltage direct current distribution network. In *2025 33rd Southern African Universities Power Engineering Conference (SAUPEC)* (pp. 1–5). *2025 33rd Southern African Universities Power Engineering Conference (SAUPEC)*. IEEE. <https://doi.org/10.1109/saupec65723.2025.10944349>
- [66] **Hyndman, R. J., & Athanasopoulos, G.** (2018). *Forecasting: Principles and practice* (2nd ed.). OTexts. ISBN: 978-0987507112.
- [67] **van Etten, T., Degeler, V., & Luo, D.** (2024). Large-scale forecasting of electric vehicle charging demand using global time series modeling. In *Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems* (pp. 40–51). *10th International Conference on Vehicle Technology and Intelligent Transport Systems*. SCITEPRESS – Science and Technology Publications. <https://doi.org/10.5220/0012555400003702>
- [68] **Alyami, S.** (2024). Ensuring sustainable grid stability through effective EV charging management: A time and energy-based approach. *Sustainability*, 16 (14), 6149. <https://doi.org/10.3390/su16146149>
- [69] **Kene, R. O., & Olwal, T. O.** (2023). Energy management and optimization of large-scale electric vehicle charging on the grid. *World Electric Vehicle Journal*, 14 (4), 95. <https://doi.org/10.3390/wevj14040095>
- [70] **Singh, A. R., Kumar, R. S., Madhavi, K. R., Alsaif, F., Bajaj, M., & Zaitsev, I.** (2024). Optimizing demand response and load balancing in smart EV charging networks using AI integrated blockchain framework. *Scientific Reports*, 14 (1). <https://doi.org/10.1038/s41598-024-82257-2>
- [71] **Yi, Z., Liu, X. C., Wei, R., Chen, X., & Dai, J.** (2021). Electric vehicle charging demand forecasting using deep learning model. *Journal of Intelligent Transportation Systems*, 26 (6), 690–703. <https://doi.org/10.1080/15472450.2021.1966627>

- [72] **Çetin, M. S., Gençoğlu, M. T., & Şahin, H.** (2024). A review of electric vehicles: Their impact on the electricity grid and artificial intelligence-based approaches for charging load management. *International Journal of Energy and Smart Grid*, 9 (1), 51–59. <https://doi.org/10.55088/ijesg.1598117>
- [73] **Buzna, L., De Falco, P., Khormali, S., Proto, D., & Straka, M.** (2019). Electric vehicle load forecasting: A comparison between time series and machine learning approaches. In *2019 1st International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED)* (pp. 1–5). *2019 1st International Conference on Energy Transition in the Mediterranean Area (SyNERGY MED)*. IEEE. <https://doi.org/10.1109/synergy-med.2019.8764110>
- [74] **Eddine, M. D., & Shen, Y.** (2022). A deep learning based approach for predicting the demand of electric vehicle charge. *The Journal of Supercomputing*, 78 (12), 14072–14095. <https://doi.org/10.1007/s11227-022-04428-0>
- [75] **Kim, Y., & Kim, S.** (2021). Forecasting charging demand of electric vehicles using time-series models. *Energies*, 14 (5), 1487. <https://doi.org/10.3390/en14051487>
- [76] **Yang, X., Peng, Z., Wang, P., & Zhuge, C.** (2023). Seasonal variance in electric vehicle charging demand and its impacts on infrastructure deployment: A big data approach. *Energy*, 280, 128230. <https://doi.org/10.1016/j.energy.2023.128230>
- [77] **Baghali, S., Guo, Z., & Hasan, S.** (2022). Investigating the spatiotemporal charging demand and travel behavior of electric vehicles using GPS data: A machine learning approach. In *2022 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1–5). *2022 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. <https://doi.org/10.1109/pesgm48719.2022.9917046>
- [78] **Gao, H. O.** (2021). *An agent-based travel and charging behavior model for forecasting high-resolution spatio-temporal battery electric vehicle charging demand* [Project report]. U.S. Department of Transportation. <https://hdl.handle.net/1813/110952>
- [79] **Ziras, C., Thingvad, M., Fog, T., Yousefi, G., & Weckesser, T.** (2024). An empirical analysis of electric vehicle charging behavior based on real Danish residential charging data. *Electric Power Systems Research*, 234, 110556. <https://doi.org/10.1016/j.epsr.2024.110556>
- [80] **Kadri, A. A., Perrouault, R., Boujelben, M. K., & Gicquel, C.** (2020). A multi-stage stochastic integer programming approach for locating electric vehicle charging stations. *Computers & Operations Research*, 117, 104888. <https://doi.org/10.1016/j.cor.2020.104888>
- [81] **Amara-Ouali, Y., Goude, Y., Doumèche, N., Veyret, P., Thomas, A., Hebenstreit, D., Wedenig, T., Satouf, A., Jan, A., Deleuze, Y., Berhaut, P., Treguer, S., & Phe-Neau, T.** (2023). Forecasting electric vehicle charging station occupancy: Smarter Mobility Data Challenge (Version 1). *arXiv*. <https://doi.org/10.48550/ARXIV.2306.06142>
- [82] **Xing, Q., Chen, Z., Zhang, Z., Xu, X., Zhang, T., Huang, X., & Wang, H.** (2020). Urban electric vehicle fast-charging demand forecasting model based on data-driven approach and human decision-making behavior. *Energies*, 13 (6), 1412. <https://doi.org/10.3390/en13061412>
- [83] **Akshay, K. C., Grace, G. H., Gunasekaran, K., & Samikannu, R.** (2024). Power consumption prediction for electric vehicle charging stations and forecasting

- income. *Scientific Reports*, 14 (1). <https://doi.org/10.1038/s41598-024-56507-2>
- [84] **Zheng, K., Xu, H., Long, Z., Wang, Y., & Chen, Q.** (2025). Coherent hierarchical probabilistic forecasting of electric vehicle charging demand. *IEEE Transactions on Industry Applications*, 61 (1), 1329–1340. <https://doi.org/10.1109/tia.2023.3344544>
 - [85] **Bandara, K., Hyndman, R., & Bergmeir, C.** (2022). MSTL: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns. *International Journal of Operational Research*, 1 (1), 1. <https://doi.org/10.1504/ijor.2022.10048281>
 - [86] **Kontopoulou, V. I., Panagopoulos, A. D., Kakkos, I., & Matsopoulos, G. K.** (2023). A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks. *Future Internet*, 15 (8), 255. <https://doi.org/10.3390/fi15080255>
 - [87] **Dudek, G.** (2023). STD: A seasonal-trend-dispersion decomposition of time series. *IEEE Transactions on Knowledge and Data Engineering*, 35 (10), 10339–10350. <https://doi.org/10.1109/tkde.2023.3268125>
 - [88] **Lem, K. H.** (2024). The STL-ARIMA approach for seasonal time series forecast: A preliminary study. *ITM Web of Conferences*, 67, 01008. <https://doi.org/10.1051/itmconf/20246701008>
 - [89] **Sohrabbeig, A., Ardakanian, O., & Musilek, P.** (2023). Decompose and conquer: Time series forecasting with multiseasonal trend decomposition using Loess. *Forecasting*, 5 (4), 684–696. <https://doi.org/10.3390/forecast5040037>
 - [90] **Lin, S., Lin, W., Hu, X., Wu, W., Mo, R., & Zhong, H.** (2024). CycleNet: Enhancing time series forecasting through modeling periodic patterns (Version 2). *arXiv*. <https://doi.org/10.48550/ARXIV.2409.18479>
 - [91] **Yoh, N., Haley, C. L., & Burivalova, Z.** (2024). Time series methods for the analysis of soundscapes and other cyclical ecological data. *Methods in Ecology and Evolution*, 15 (7), 1158–1176. <https://doi.org/10.1111/2041-210x.14361>
 - [92] **Brockwell, P. J., & Davis, R. A.** (2016). *Introduction to time series and forecasting*. In *Springer Texts in Statistics*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-29854-2>
 - [93] **Tsoku, J. T., Metsileng, D., & Botlhoko, T.** (2024). A hybrid of Box-Jenkins ARIMA model and neural networks for forecasting South African crude oil prices. *International Journal of Financial Studies*, 12 (4), 118. <https://doi.org/10.3390/ijfs12040118>
 - [94] **Yakubu, U. A., & Saputra, M. P. A.** (2022). Time series model analysis using autocorrelation function (ACF) and partial autocorrelation function (PACF) for e-wallet transactions during a pandemic. *International Journal of Global Operations Research*, 3 (3), 80–85. <https://doi.org/10.47194/ijgor.v3i3.168>
 - [95] **Parreño, S. J.** (2022). Forecasting electricity consumption in the Philippines using ARIMA models. *International Journal of Machine Learning and Computing*, 12 (6). Available at SSRN: <https://ssrn.com/abstract=4419537>
 - [96] **Chodakowska, E., Nazarko, J., & Nazarko, L.** (2021). ARIMA models in electrical load forecasting and their robustness to noise. *Energies*, 14 (23), 7952. <https://doi.org/10.3390/en14237952>
 - [97] **Pinheiro, M. G., Madeira, S. C., & Francisco, A. P.** (2023). Short-term electricity load forecasting—A systematic approach from system level to secondary substations. *Applied Energy*, 332, 120493. <https://doi.org/10.1016/j.apenergy.2022.120493>

- [98] Szostek, K., Mazur, D., Dralus, G., & Kuszniar, J. (2024). Analysis of the effectiveness of ARIMA, SARIMA, and SVR models in time series forecasting: A case study of wind farm energy production. *Energies*, 17 (19), 4803. <https://doi.org/10.3390/en17194803>
- [99] Alharbi, F. R., & Csala, D. (2022). A seasonal autoregressive integrated moving average with exogenous factors (SARIMAX) forecasting model-based time series approach. *Inventions*, 7 (4), 94. <https://doi.org/10.3390/inventions7040094>
- [100] Amini, M. H., Kargarian, A., & Karabasoglu, O. (2016). ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. *Electric Power Systems Research*, 140, 378–390. <https://doi.org/10.1016/j.epsr.2016.06.003>
- [101] Sreekumar, A. V., & Lekshmi, R. R. (2024). Electric vehicle charging station demand prediction model deploying data slotting. *Results in Engineering*, 24, 103095. <https://doi.org/10.1016/j.rineng.2024.103095>
- [102] Baur, L., Ditschuneit, K., Schambach, M., Kaymakci, C., Wollmann, T., & Sauer, A. (2024). Explainability and interpretability in electric load forecasting using machine learning techniques – A review. *Energy and AI*, 16, 100358. <https://doi.org/10.1016/j.egyai.2024.100358>
- [103] Giudici, P., Piergallini, A., Recchioni, M. C., & Raffinetti, E. (2024). Explainable artificial intelligence methods for financial time series. *Physica A: Statistical Mechanics and Its Applications*, 655, 130176. <https://doi.org/10.1016/j.physa.2024.130176>
- [104] Kim, J., Kim, H., Kim, H., Lee, D., & Yoon, S. (2025). A comprehensive survey of deep learning for time series forecasting: Architectural diversity and open challenges. *Artificial Intelligence Review*, 58 (7). <https://doi.org/10.1007/s10462-025-11223-9>
- [105] Tolun, Ö. C., Zor, K., & Tutsoy, O. (2025). A comprehensive benchmark of machine learning-based algorithms for medium-term electric vehicle charging demand prediction. *The Journal of Supercomputing*, 81 (3). <https://doi.org/10.1007/s11227-025-06975-8>
- [106] Louie, H. M. (2017). Time-series modeling of aggregated electric vehicle charging station load. *Electric Power Components and Systems*, 45 (14), 1498–1511. <https://doi.org/10.1080/15325008.2017.1336583>
- [107] Al-Ogaili, A. S., Tengku Hashim, T. J., Rahmat, N. A., Ramasamy, A. K., Marsadek, M. B., Faisal, M., & Hannan, M. A. (2019). Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: Challenges and recommendations. *IEEE Access*, 7, 128353–128371. <https://doi.org/10.1109/access.2019.2939595>
- [108] Lai, Y., & Dzombak, D. A. (2020). Use of the autoregressive integrated moving average (ARIMA) model to forecast near-term regional temperature and precipitation. *Weather and Forecasting*, 35 (3), 959–976. <https://doi.org/10.1175/waf-d-19-0158.1>
- [109] Zafra-Mejía, C. A., Rondón-Quintana, H. A., & Urazán-Bonells, C. F. (2024). ARIMA and TFARIMA analysis of the main water quality parameters in the initial components of a megacity's drinking water supply system. *Hydrology*, 11 (1), 10. <https://doi.org/10.3390/hydrology11010010>
- [110] Hardiyanti, D. Y., Novianty, H., & Lestarini, D. (2019). Forecasting river water quality using autoregressive integrated moving average (ARIMA). In *Proceedings of the International Conferences on Information System and*

- Technology (pp. 158–163). *International Conferences on Information System and Technology*. SCITEPRESS – Science and Technology Publications. <https://doi.org/10.5220/0009907201580163>
- [111] **Hyndman, R. J., & Khandakar, Y.** (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27 (3). <https://doi.org/10.18637/jss.v027.i03>
- [112] **Du, H., Zhao, Z., & Xue, H.** (2020). ARIMA-M: A new model for daily water consumption prediction based on the autoregressive integrated moving average model and the Markov chain error correction. *Water*, 12 (3), 760. <https://doi.org/10.3390/w12030760>
- [113] **Kaur, J., Parmar, K. S., & Singh, S.** (2023). Autoregressive models in environmental forecasting time series: A theoretical and application review. *Environmental Science and Pollution Research*, 30 (8), 19617–19641. <https://doi.org/10.1007/s11356-023-25148-9>
- [114] **Mado, I., Budiman, A., & Triwiyatno, A.** (2023). Short-term forecasting daily electricity loads using seasonal ARIMA patterns of generation units at PT. PLN (Persero) Tarakan City. *Jurnal Ilmiah Kursor*, 12 (2), 93–100. <https://doi.org/10.21107/kursor.v12i2.348>
- [115] **Laitos, V., Vontzos, G., Paraschoudis, P., Tsampasis, E., Bargiotas, D., & Tsoukalas, L. H.** (2024). The state of the art electricity load and price forecasting for the modern wholesale electricity market. *Energies*, 17 (22), 5797. <https://doi.org/10.3390/en17225797>
- [116] **Fatima, S. S. W., & Rahimi, A.** (2024). A review of time-series forecasting algorithms for industrial manufacturing systems. *Machines*, 12 (6), 380. <https://doi.org/10.3390/machines12060380>
- [117] **Van Krieking, G., De Cauwer, C., Sapountzoglou, N., Coosemans, T., & Messagie, M.** (2021). Day-ahead forecast of electric vehicle charging demand with deep neural networks. *World Electric Vehicle Journal*, 12 (4), 178. <https://doi.org/10.3390/wevj12040178>
- [118] **Cheng, C., Sa-Ngasoongsong, A., Beyca, O., Le, T., Yang, H., Kong, Z. (James), & Bukkapatnam, S. T. S.** (2015). Time series forecasting for nonlinear and non-stationary processes: A review and comparative study. *IIE Transactions*, 47 (10), 1053–1071. <https://doi.org/10.1080/0740817x.2014.999180>
- [119] **Gilleran, M., Bonnema, E., Woods, J., Mishra, P., Doebber, I., Hunter, C., Mitchell, M., & Mann, M.** (2021). Impact of electric vehicle charging on the power demand of retail buildings. *Advances in Applied Energy*, 4, 100062. <https://doi.org/10.1016/j.adapen.2021.100062>
- [120] **Wang, S., Li, C., & Lim, A.** (2019). Why are the ARIMA and SARIMA not sufficient (Version 3). *arXiv*. <https://doi.org/10.48550/ARXIV.1904.07632>
- [121] **Ren, F., Tian, C., Zhang, G., Li, C., & Zhai, Y.** (2022). A hybrid method for power demand prediction of electric vehicles based on SARIMA and deep learning with integration of periodic features. *Energy*, 250, 123738. <https://doi.org/10.1016/j.energy.2022.123738>
- [122] **Paudel, A., Mandal, P., & Ravikumar, G.** (2024). Forecasting charging demand patterns of electric vehicles using hybrid deep learning approaches. In *2024 IEEE Power & Energy Society General Meeting (PESGM)* (pp. 1–5). *2024 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. <https://doi.org/10.1109/pesgm51994.2024.10761077>
- [123] **Majidpour, M., Qiu, C., Chu, P., Gadh, R., & Pota, H. R.** (2014). Modified pattern sequence-based forecasting for electric vehicle charging stations. In *2014*

- IEEE International Conference on Smart Grid Communications (SmartGridComm)* (pp. 710–715). 2014 *IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE. <https://doi.org/10.1109/smartgridcomm.2014.7007731>
- [124] Akhtar, S., Shahzad, S., Zaheer, A., Ullah, H. S., Kilic, H., Gono, R., Jasiński, M., & Leonowicz, Z. (2023). Short-term load forecasting models: A review of challenges, progress, and the road ahead. *Energies*, 16 (10), 4060. <https://doi.org/10.3390/en16104060>
- [125] Hamed, M. M., Mustafa, A., Al-Sharif, M., & Shawaqfah, M. (2021). Modeling the households' satisfaction level with the first electric vehicle and the time until the purchase of the second electric vehicle. *International Journal of Sustainable Transportation*, 17 (1), 52–64. <https://doi.org/10.1080/15568318.2021.1983677>
- [126] Alansari, M., Al-Sumaiti, A. S., & Abughali, A. (2024). Optimal placement of electric vehicle charging infrastructures utilizing deep learning. *IET Intelligent Transport Systems*, 18 (8), 1529–1544. <https://doi.org/10.1049/itr2.12527>
- [127] Xin, F., Yang, X., Beibei, W., Ruilin, X., Fei, M., & Jianyong, Z. (2024). Research on electric vehicle charging load prediction method based on spectral clustering and deep learning network. *Frontiers in Energy Research*, 12. <https://doi.org/10.3389/fenrg.2024.1294453>
- [128] Sulandari, W., Subanar, S., Suhartono, S., & Utami, H. (2016). Forecasting electricity load demand using hybrid exponential smoothing–artificial neural network model. *International Journal of Advances in Intelligent Informatics*, 2 (3), 131. <https://doi.org/10.26555/ijain.v2i3.69>
- [129] Waheed, W., & Qingshan, X. (2024). An efficient load forecasting technique by using Holt-Winters and Prophet algorithms to mitigate the impact on power consumption in COVID-19. *IET Energy Systems Integration*, 6 (4), 364–374. <https://doi.org/10.1049/esi2.12132>
- [130] Trubetskoy, G. (2016, January 29). *Triple exponential smoothing (forecasting)*. <https://grisha.org/blog/2016/01/29/triple-exponential-smoothing-forecasting/>
- [131] Djakaria, I., & Saleh, S. E. (2021). Covid-19 forecast using Holt-Winters exponential smoothing. *Journal of Physics: Conference Series*, 1882 (1), 012033. <https://doi.org/10.1088/1742-6596/1882/1/012033>
- [132] Cihan, P. (2022). Impact of the COVID-19 lockdowns on electricity and natural gas consumption in the different industrial zones and forecasting consumption amounts: Turkey case study. *International Journal of Electrical Power & Energy Systems*, 134, 107369. <https://doi.org/10.1016/j.ijepes.2021.107369>
- [133] Cihan, P. (2023). Time-series forecasting of energy demand in electric vehicles and impact of the COVID-19 pandemic on energy demand. *Sakarya University Journal of Computer and Information Sciences*, 6 (1), 10–21. <https://doi.org/10.35377/saucis...1209519>
- [134] Mortezaejad, S. A. F., & Wang, R. (2025). Addressing challenges in time series forecasting: A comprehensive comparison of machine learning techniques (Version 1). *arXiv*. <https://doi.org/10.48550/ARXIV.2503.20148>
- [135] Almaghrebi, A., Aljuheshi, F., Rafaie, M., James, K., & Alahmad, M. (2020). Data-driven charging demand prediction at public charging stations using supervised machine learning regression methods. *Energies*, 13 (16), 4231. <https://doi.org/10.3390/en13164231>

- [136] **Jahangir, H., Tayarani, H., Ahmadian, A., Golkar, M. A., Miret, J., Tayarani, M., & Gao, H. O.** (2019). Charging demand of plug-in electric vehicles: Forecasting travel behavior based on a novel Rough Artificial Neural Network approach. *Journal of Cleaner Production*, 229, 1029–1044. <https://doi.org/10.1016/j.jclepro.2019.04.345>
- [137] **Zhu, J., Yang, Z., Mourshed, M., Guo, Y., Zhou, Y., Chang, Y., Wei, Y., & Feng, S.** (2019). Electric vehicle charging load forecasting: A comparative study of deep learning approaches. *Energies*, 12 (14), 2692. <https://doi.org/10.3390/en12142692>
- [138] **Friedman, J. H.** (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29 (5). <https://doi.org/10.1214/aos/1013203451>
- [139] **Chen, T., & Guestrin, C.** (2016). XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). *KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. <https://doi.org/10.1145/2939672.2939785>
- [140] **Breiman, L.** (2001). Random forests. *Machine Learning*, 45 (1), 5–32. <https://doi.org/10.1023/a:1010933404324>
- [141] **Lu, Y., Li, Y., Xie, D., Wei, E., Bao, X., Chen, H., & Zhong, X.** (2018). The application of improved random forest algorithm on the prediction of electric vehicle charging load. *Energies*, 11 (11), 3207. <https://doi.org/10.3390/en11113207>
- [142] **Smola, A. J., & Schölkopf, B.** (2004). A tutorial on support vector regression. *Statistics and Computing*, 14 (3), 199–222. <https://doi.org/10.1023/b:stco.0000035301.49549.88>
- [143] **Xydas, E. S., Marmaras, C. E., Cipcigan, L. M., Hassan, A. S., & Jenkins, N.** (2013). Forecasting electric vehicle charging demand using support vector machines. In *2013 48th International Universities' Power Engineering Conference (UPEC). 2013 48th International Universities' Power Engineering Conference (UPEC)*. IEEE. <https://doi.org/10.1109/upec.2013.6714942>
- [144] **Chen, W., Chen, X., Liao, Y., Wang, G., Yao, J., & Chen, K.** (2013). Short-term load forecasting based on time series reconstruction and support vector regression. In *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)* (pp. 1–4). *TENCON 2013 – 2013 IEEE Region 10 Conference*. IEEE. <https://doi.org/10.1109/tencon.2013.6718960>
- [145] **Zhang, X.** (2018). Short-term load forecasting for electric bus charging stations based on fuzzy clustering and least squares support vector machine optimized by wolf pack algorithm. *Energies*, 11 (6), 1449. <https://doi.org/10.3390/en11061449>
- [146] **Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y.** (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
- [147] **Hochreiter, S., & Schmidhuber, J.** (1997). Long short-term memory. *Neural Computation*, 9 (8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>

- [148] **Elahe, M. F., Kabir, M. A., Mahmud, S. M. H., & Azim, R.** (2022). Factors impacting short-term load forecasting of charging station to electric vehicle. *Electronics*, 12 (1), 55. <https://doi.org/10.3390/electronics12010055>
- [149] **Xue, M., Wu, L., Zhang, Q. P., Lu, J. X., Mao, X., & Pan, Y.** (2021). Research on load forecasting of charging station based on XGBoost and LSTM model. *Journal of Physics: Conference Series*, 1757 (1), 012145. <https://doi.org/10.1088/1742-6596/1757/1/012145>
- [150] **Wang, Z., Wan, B., Ling, T., et al.** (2021). Electric bus charging load forecasting method based on spectral clustering and LSTM neural network. *Electric Power Construction*, 42 (6), 58–66. <https://doi.org/10.12204/j.issn.1000-7229.2021.06.006>
- [151] **Gerossier, A., Girard, R., & Kariniotakis, G.** (2019). Modeling and forecasting electric vehicle consumption profiles. *Energies*, 12 (7), 1341. <https://doi.org/10.3390/en12071341>
- [152] **Dabbaghjamanesh, M., Moeini, A., & Kavousi-Fard, A.** (2021). Reinforcement learning-based load forecasting of electric vehicle charging station using Q-learning technique. *IEEE Transactions on Industrial Informatics*, 17 (6), 4229–4237. <https://doi.org/10.1109/tii.2020.2990397>
- [153] **Taylor, S. J., & Letham, B.** (2018). Forecasting at scale. *The American Statistician*, 72 (1), 37–45. <https://doi.org/10.1080/00031305.2017.1380080>

EKLER

EK A: V1 Model Performans Çıktıları

EK B: V2 Model Performans Çıktıları

EK C: V3 Model Performans Çıktıları

EK D: V4 Model Performans Çıktıları

EK A

Çizelge A.1 : V1 veri seti ARIMA model performansları.

Model İsmi	p	d	q	AIC	BIC	MSE	RMSE	MAE	MAPE	SMAPE	MBE
Random Walk	0	1	0	662.6576	670.1627	29.4122	5.4233	4.1047	25.1524	20.9729	-1.6973
AR(1)	1	0	0	600.5099	611.7677	182.4091	13.5059	12.3335	60.8154	89.4287	12.3335
AR(2)	2	0	0	545.3918	560.4021	173.2935	13.1641	11.7615	57.2913	83.5399	11.7615
MA(1)	0	0	1	673.7367	684.9944	185.5536	13.6218	12.4920	61.7283	91.1360	12.4920
MA(2)	0	0	2	633.9932	649.0035	185.6239	13.6244	12.4895	61.7293	91.1640	12.4895
Durağan Olmayan AR(1)	1	1	0	566.4765	577.7343	71.7208	8.4688	7.1407	44.7852	33.0870	-6.4910
Durağan Olmayan MA(1)	0	1	1	473.2890	484.5467	61.5467	7.8452	6.5232	40.9750	30.7617	-5.6623
Karma Model	1	1	1	475.2876	490.2979	61.4859	7.8413	6.5194	40.9516	30.7475	-5.6571
Hata Terimleriyle Model	0	1	2	475.2879	490.2982	61.4917	7.8417	6.5198	40.9540	30.7490	-5.6576
Genişletilmiş Karma	1	1	2	472.7207	491.4836	57.7251	7.5977	6.2871	39.5731	29.9041	-5.3591
Daha Karmaşık Karma	2	1	2	474.1375	496.6529	59.1311	7.6897	6.3769	40.1266	30.2417	-5.4832

Çizelge A.2 : V1 veri seti SARIMA model performansları.

Model İsmi	p	d	q	P	D	Q	s	AIC	BIC	MSE	RMSE	MAE	MAPE	SMAPE	MBE
Basit SARIMA	1	1	1	1	1	1	7	716.4989	739.0144	48.9578	6.9970	5.5560	33.2548	27.2622	-2.9802
Mevsimsel SARIMA	2	1	2	1	1	1	7	703.7695	726.2850	137.2229	11.7142	9.7977	58.4507	40.9371	-8.7753
Yüksek Dereceli SARIMA	3	1	3	2	1	2	7	903.9762	926.4917	67.3162	8.2046	6.7757	41.3879	32.0196	-4.6339
Aylık SARIMA	1	1	1	1	1	1	12	1105.6760	1128.1914	65535	65535	---	---	197.7823	----
Hibrit SARIMA	2	1	1	1	1	2	7	703.7695	726.2850	137.2205	11.7141	9.7976	58.4503	40.9369	-8.7752

Çizelge A.3 : V1 veri seti Holt-Winters model performansları.

Model İsmi	beta	gamma	MSE	RMSE	MAE	MAPE	SMAPE	MBE
Holt Doğrusal Trend	1	0	119.9644	10.9528	9.5511	45.4845	61.1778	9.5111
Holt-Winters Katı Mevsimsel	0	1	119.9644	10.9528	9.5511	45.4845	61.1778	9.5111
Holt-Winters Adaptif Mevsimsel	1	1	119.9644	10.9528	9.5511	45.4845	61.1778	9.5111
Basit Üstel Düzgünleştirme	0	0	119.9644	10.9528	9.5511	45.4845	61.1778	9.5111

Çizelge A.4 : V1 veri seti LSTM model performansları.

Katman	Gizli Birim	Öğrenme Oranı	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE	SMAPE	MBE
1	100	0.001	0	YANLIŞ	2.1942	1.4813	1.2145	7.1897	7.1834	-0.0326
1	100	0.001	0	DOĞRU	2.0380	1.4276	1.1791	6.4308	6.4852	0.0464

Çizelge A.4 (devam) : V1 veri seti LSTM model performansları.

Katman	Gizli Birim	Öğrenme Oranı	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE	SMAPE	MBE
1	100	0.001	0.2	YANLIŞ	1.7618	1.3273	1.0970	6.3165	6.3366	0.0828
1	100	0.001	0.2	DOĞRU	2.1472	1.4653	1.1524	6.4422	6.5072	0.0352
1	100	0.005	0	YANLIŞ	2.4517	1.5658	1.2787	7.2958	7.5328	0.3227
1	100	0.005	0	DOĞRU	2.2566	1.5022	1.2453	6.8893	6.8716	-0.1428
1	100	0.005	0.2	YANLIŞ	1.5320	1.2377	0.9667	5.7785	5.8658	0.1483
1	100	0.005	0.2	DOĞRU	1.4935	1.2221	0.9885	5.6165	5.6280	0.1228
1	200	0.001	0	YANLIŞ	1.9053	1.3803	1.1302	6.2030	6.2725	0.1972
1	200	0.001	0	DOĞRU	1.7783	1.3335	1.0695	6.2112	6.1066	-0.2505
1	200	0.001	0.2	YANLIŞ	1.6423	1.2815	1.0775	6.0477	5.9528	-0.2028
1	200	0.001	0.2	DOĞRU	1.6041	1.2665	0.9994	5.5100	5.4604	-0.1235
1	200	0.005	0	YANLIŞ	1.8650	1.3657	1.1051	6.3076	6.3316	0.0280
1	200	0.005	0	DOĞRU	1.4219	1.1924	0.9383	5.2518	5.2360	0.0397
1	200	0.005	0.2	YANLIŞ	1.7131	1.3089	1.0460	5.8438	5.9600	0.1370
1	200	0.005	0.2	DOĞRU	2.1229	1.4570	1.1872	6.7497	6.7877	0.0268
2	100	0.001	0	YANLIŞ	2.3359	1.5284	1.1799	6.5971	6.6174	-0.1082
2	100	0.001	0	DOĞRU	1.9161	1.3842	1.1297	5.9955	6.0329	0.1342
2	100	0.001	0.2	YANLIŞ	1.8030	1.3427	1.0541	5.8544	5.8848	-0.0597
2	100	0.001	0.2	DOĞRU	1.7466	1.3216	1.0482	5.8419	5.7578	-0.3228
2	100	0.005	0	YANLIŞ	1.7059	1.3061	1.0439	5.8843	5.8774	0.1836
2	100	0.005	0	DOĞRU	1.8614	1.3643	1.0750	6.0687	5.9675	-0.3407
2	100	0.005	0.2	YANLIŞ	1.7414	1.3196	1.1038	6.0740	6.0511	-0.0784
2	100	0.005	0.2	DOĞRU	1.9634	1.4012	1.1220	6.2034	6.1105	-0.3502
2	200	0.001	0	YANLIŞ	1.6347	1.2785	1.0454	6.1010	6.0179	-0.1218
2	200	0.001	0	DOĞRU	1.9846	1.4088	1.1329	6.4387	6.4430	0.0842
2	200	0.001	0.2	YANLIŞ	1.5614	1.2496	0.9884	5.5794	5.6436	0.0640
2	200	0.001	0.2	DOĞRU	2.3185	1.5227	1.1957	6.9746	7.0315	-0.0344
2	200	0.005	0	YANLIŞ	1.6756	1.2945	1.0324	5.8944	5.8011	-0.1389

Çizelge A.4 (devam-2) : V1 veri seti LSTM model performansları.

Katman	Gizli Birim	Öğrenme Oranı	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE	SMAPE	MBE
2	200	0.005	0	DOĞRU	1.6994	1.3036	0.9978	5.8329	5.7806	0.0144
2	200	0.005	0.2	YANLIŞ	1.4060	1.1858	0.9594	5.2824	5.3098	0.0456
2	200	0.005	0.2	DOĞRU	1.9868	1.4096	1.1622	6.4933	6.5156	-0.0089

Çizelge A.5 : V1 veri seti Random Forest model performansları.

Ağaç	Derinlik	Yaprak (Min)	MSE	RMSE	MAE	MAPE	SMAPE	MBE
50	3	1	40.6320	6.3743	4.8616	24.0650	25.8767	3.1364
50	3	5	40.8093	6.3882	4.8110	23.7671	25.5953	3.0478
50	5	1	39.9199	6.3182	4.7873	23.7829	25.4173	2.9436
50	5	5	39.2162	6.2623	4.7259	23.6088	25.0690	2.8851
100	3	1	40.1924	6.3397	4.8272	23.9524	25.6626	3.0566
100	3	5	39.1399	6.2562	4.7335	23.4930	25.0660	2.9453
100	5	1	40.1615	6.3373	4.8086	23.7562	25.5181	3.0557
100	5	5	41.3918	6.4336	4.8648	23.9756	25.8539	3.1825

Çizelge A.6 : V1 veri seti SVR model performansları.

Kutu (C)	Hata Payı	Çekirdek	Ölçek	MSE	RMSE	MAE	MAPE	SMAPE	MBE
0.1	0.01	linear	auto	38.6266	6.2150	4.7412	23.1911	25.4119	3.3530
0.1	0.01	linear	1	30.6083	5.5325	4.1024	22.1240	21.5720	1.0859
0.1	0.01	gaussian	auto	113.3887	10.6484	9.0659	42.7413	56.9311	8.9319
0.1	0.01	gaussian	1	135.7784	11.6524	10.3313	49.7420	68.4634	10.3090
0.1	0.1	linear	auto	39.4221	6.2787	4.8032	23.4137	25.7897	3.4732
0.1	0.1	linear	1	30.0950	5.4859	4.0742	22.1639	21.4202	0.9092
0.1	0.1	gaussian	auto	115.3043	10.7380	9.1664	43.2607	57.8022	9.0439

Çizelge A.6 (devam) : V1 veri seti SVR model performansları.

Kutu (C)	Hata Payı	Çekirdek	Ölçek	MSE	RMSE	MAE	MAPE	SMAPE	MBE
0.1	0.1	gaussian	1	135.7784	11.6524	10.3313	49.7420	68.4634	10.3090
1	0.01	linear	auto	30.9634	5.5645	4.1461	21.9977	21.8488	1.4451
1	0.01	linear	1	30.5399	5.5263	4.0981	22.1224	21.5504	1.0749
1	0.01	gaussian	auto	83.2241	9.1227	7.2204	33.6897	42.4061	6.6172
1	0.01	gaussian	1	135.8054	11.6536	10.3325	49.7490	68.4757	10.3103
1	0.1	linear	auto	31.1786	5.5838	4.1595	22.0737	21.9077	1.4432
1	0.1	linear	1	30.4086	5.5144	4.1107	22.1005	21.6507	1.1677
1	0.1	gaussian	auto	82.3918	9.0770	7.1784	33.4693	42.0623	6.5803
1	0.1	gaussian	1	135.8984	11.6575	10.3369	49.7731	68.5181	10.3148

Çizelge A.7 : V1 veri seti LightGBM model performansları.

Yaprak Sayısı	Öğrenme Oranı	Tahminci Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
15	0.01	100	1.1196	0.8073	1.0877E+17	-0.0503	149.2428	-0.3242
15	0.01	200	1.1067	0.7969	1.1518E+17	-0.0261	146.4812	-0.2850
15	0.01	300	1.1028	0.7949	1.1799E+17	-0.0191	145.6506	-0.2675
15	0.05	100	1.1048	0.7971	1.2030E+17	-0.0227	145.4836	-0.2557
15	0.05	200	1.1151	0.8017	1.2048E+17	-0.0418	146.1417	-0.2560
15	0.05	300	1.1247	0.8089	1.2179E+17	-0.0599	146.8935	-0.2542
15	0.1	100	1.1146	0.8020	1.2071E+17	-0.0408	146.2015	-0.2557
15	0.1	200	1.1299	0.8124	1.2185E+17	-0.0697	147.4755	-0.2552
15	0.1	300	1.1404	0.8204	1.2264E+17	-0.0897	148.3767	-0.2557
31	0.01	100	1.1196	0.8070	1.0980E+17	-0.0503	149.0870	-0.3193
31	0.01	200	1.1096	0.7981	1.1632E+17	-0.0317	146.5535	-0.2802
31	0.01	300	1.1097	0.7987	1.1846E+17	-0.0317	146.1956	-0.2673
31	0.05	100	1.1117	0.8011	1.2059E+17	-0.0356	146.1452	-0.2550
31	0.05	200	1.1295	0.8125	1.2178E+17	-0.0689	147.5170	-0.2549

Çizelge A.7 (devam) : V1 veri seti LightGBM model performansları.

Yaprak Sayısı	Öğrenme Oranı	Tahminci Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
31	0.05	300	1.1423	0.8206	1.2225E+17	-0.0932	148.6208	-0.2554
31	0.1	100	1.1318	0.8141	1.2201E+17	-0.0733	147.6962	-0.2546
31	0.1	200	1.1530	0.8296	1.2373E+17	-0.1139	149.3766	-0.2546
31	0.1	300	1.1680	0.8402	1.2492E+17	-0.1431	150.5056	-0.2538
50	0.01	100	1.1219	0.8084	1.1014E+17	-0.0546	149.2458	-0.3192
50	0.01	200	1.1137	0.8020	1.1720E+17	-0.0392	146.9125	-0.2786
50	0.01	300	1.1136	0.8022	1.1952E+17	-0.0391	146.4876	-0.2640
50	0.05	100	1.1214	0.8083	1.2169E+17	-0.0537	146.9722	-0.2534
50	0.05	200	1.1405	0.8220	1.2332E+17	-0.0899	148.6629	-0.2525
50	0.05	300	1.1553	0.8334	1.2477E+17	-0.1183	149.7975	-0.2514
50	0.1	100	1.1444	0.8238	1.2253E+17	-0.0974	149.1480	-0.2585
50	0.1	200	1.1687	0.8422	1.2479E+17	-0.1444	150.8290	-0.2573
50	0.1	300	1.1854	0.8538	1.2634E+17	-0.1773	151.8317	-0.2578

Çizelge A.8 : V1 veri seti XGBoost model performansları.

Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
100	0.01	3	0.7	0.7	1.1258	0.8129	1.0591E+17	-0.0619	150.6305	-0.3425
100	0.01	3	0.7	1	1.1216	0.8089	1.0694E+17	-0.0540	149.9057	-0.3340
100	0.01	3	1	0.7	1.1264	0.8133	1.0594E+17	-0.0631	150.6747	-0.3427
100	0.01	3	1	1	1.1245	0.8105	1.0638E+17	-0.0594	150.2752	-0.3378
100	0.01	5	0.7	0.7	1.1253	0.8125	1.0657E+17	-0.0610	150.4124	-0.3392
100	0.01	5	0.7	1	1.1201	0.8084	1.0866E+17	-0.0512	149.4430	-0.3257
100	0.01	5	1	0.7	1.1263	0.8133	1.0678E+17	-0.0628	150.4611	-0.3389
100	0.01	5	1	1	1.1213	0.8098	1.0901E+17	-0.0535	149.5514	-0.3241
100	0.05	3	0.7	0.7	1.0961	0.7921	1.1973E+17	-0.0067	144.7235	-0.2548

Çizelge A.8 (devam) : V1 veri seti XGBoost model performansları.

Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
100	0.05	3	0.7	1	1.0985	0.7936	1.2038E+17	-0.0111	144.8092	-0.2524
100	0.05	3	1	0.7	1.0989	0.7947	1.1942E+17	-0.0119	145.0465	-0.2597
100	0.05	3	1	1	1.1022	0.7965	1.1919E+17	-0.0179	145.3350	-0.2618
100	0.05	5	0.7	0.7	1.1035	0.7974	1.1972E+17	-0.0203	145.4867	-0.2564
100	0.05	5	0.7	1	1.1081	0.8018	1.2162E+17	-0.0287	145.7119	-0.2491
100	0.05	5	1	0.7	1.1062	0.7998	1.1999E+17	-0.0253	145.7306	-0.2578
100	0.05	5	1	1	1.1065	0.8007	1.2077E+17	-0.0258	145.6676	-0.2506
200	0.01	3	0.7	0.7	1.1084	0.8014	1.1318E+17	-0.0293	147.2901	-0.2980
200	0.01	3	0.7	1	1.1051	0.7976	1.1413E+17	-0.0233	146.6866	-0.2897
200	0.01	3	1	0.7	1.1099	0.8027	1.1305E+17	-0.0321	147.4400	-0.2996
200	0.01	3	1	1	1.1089	0.8006	1.1296E+17	-0.0303	147.3047	-0.2975
200	0.01	5	0.7	0.7	1.1094	0.8017	1.1372E+17	-0.0313	147.2596	-0.2956
200	0.01	5	0.7	1	1.1078	0.8001	1.1568E+17	-0.0283	146.7329	-0.2837
200	0.01	5	1	0.7	1.1110	0.8034	1.1395E+17	-0.0343	147.3691	-0.2952
200	0.01	5	1	1	1.1098	0.8019	1.1533E+17	-0.0319	146.9516	-0.2847
200	0.05	3	0.7	0.7	1.0986	0.7938	1.2074E+17	-0.0112	144.7560	-0.2479
200	0.05	3	0.7	1	1.1020	0.7963	1.2064E+17	-0.0176	145.1374	-0.2505
200	0.05	3	1	0.7	1.1009	0.7962	1.2045E+17	-0.0156	144.9864	-0.2529
200	0.05	3	1	1	1.1045	0.7983	1.2051E+17	-0.0222	145.2545	-0.2545
200	0.05	5	0.7	0.7	1.1116	0.8025	1.2098E+17	-0.0354	145.8981	-0.2495
200	0.05	5	0.7	1	1.1201	0.8079	1.2268E+17	-0.0513	146.4919	-0.2461
200	0.05	5	1	0.7	1.1123	0.8039	1.2082E+17	-0.0367	146.0351	-0.2526
200	0.05	5	1	1	1.1163	0.8066	1.2186E+17	-0.0442	146.1829	-0.2456

Çizelge A.9 : V1 veri seti Prophet model performansları.

Değişim Nokt. Ölç.	Mevsimsellik	Tatil Ölçeği	RMSE	MAE	MAPE	R2	SMAPE	MBE
0.01	additive	5	2.3711	2.0346	6.0480E+17	-3.7131	134.7533	1.8282
0.01	additive	10	2.3711	2.0346	6.0480E+17	-3.7131	134.7533	1.8282
0.01	multiplicative	5	2.3675	2.0249	6.0054E+17	-3.6989	134.6907	1.8116
0.01	multiplicative	10	2.3675	2.0249	6.0054E+17	-3.6989	134.6907	1.8116
0.05	additive	5	3.0144	2.5954	7.4392E+17	-6.6175	139.9455	2.4432
0.05	additive	10	3.0144	2.5954	7.4392E+17	-6.6175	139.9455	2.4432
0.05	multiplicative	5	2.9995	2.5730	7.3703E+17	-6.5420	139.7347	2.4157
0.05	multiplicative	10	2.9995	2.5730	7.3703E+17	-6.5420	139.7347	2.4157

Çizelge A.10 : V1 veri seti GBR model performansları.

Tahminci Sayısı	Maks. Derinlik	Öğrenme Oranı	Min. Yaprak Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
50	3	0.05	5	1.1041	0.7970	1.1567E+17	-0.0214	146.3127	-0.2814
50	3	0.05	10	1.1040	0.7965	1.1558E+17	-0.0211	146.2880	-0.2814
50	3	0.1	5	1.1024	0.7969	1.1911E+17	-0.0183	145.4870	-0.2625
50	3	0.1	10	1.1018	0.7960	1.1989E+17	-0.0171	145.2218	-0.2591
50	5	0.05	5	1.1105	0.8010	1.1789E+17	-0.0332	146.5364	-0.2728
50	5	0.05	10	1.1090	0.8015	1.1759E+17	-0.0305	146.5972	-0.2743
50	5	0.1	5	1.1102	0.8012	1.1954E+17	-0.0328	146.3000	-0.2600
50	5	0.1	10	1.1085	0.8008	1.2020E+17	-0.0295	146.1273	-0.2581
100	3	0.05	5	1.1014	0.7961	1.1901E+17	-0.0163	145.4411	-0.2623
100	3	0.05	10	1.1021	0.7964	1.1936E+17	-0.0178	145.3943	-0.2619
100	3	0.1	5	1.1042	0.7987	1.1990E+17	-0.0216	145.6761	-0.2583
100	3	0.1	10	1.1058	0.7994	1.2103E+17	-0.0245	145.4575	-0.2548
100	5	0.05	5	1.1116	0.8041	1.2103E+17	-0.0353	146.1904	-0.2551
100	5	0.05	10	1.1117	0.8030	1.1977E+17	-0.0355	146.4874	-0.2620
100	5	0.1	5	1.1165	0.8048	1.2017E+17	-0.0445	146.7524	-0.2561
100	5	0.1	10	1.1128	0.8049	1.2133E+17	-0.0375	146.5134	-0.2529

Çizelge B.1 : V2 veri seti ARIMA model performansları.

Model İsmi	p	d	q	AIC	BIC	MSE	RMSE	MAE	MAPE
Random Walk	0	1	0	662.6576	670.1627	29.4122	5.4233	4.1047	25.1524
AR(1)	1	0	0	600.5099	611.7677	182.4091	13.5059	12.3335	60.8154
AR(2)	2	0	0	545.3918	560.4021	173.2935	13.1641	11.7615	57.2913
MA(1)	0	0	1	673.7367	684.9944	185.5536	13.6218	12.4920	61.7283
MA(2)	0	0	2	633.9932	649.0035	185.6239	13.6244	12.4895	61.7293
Durağan Olmayan AR(1)	1	1	0	566.4765	577.7343	71.7208	8.4688	7.1407	44.7852
Durağan Olmayan MA(1)	0	1	1	473.2890	484.5467	61.5467	7.8452	6.5232	40.9750
Karma Model	1	1	1	475.2876	490.2979	61.4859	7.8413	6.5194	40.9516
Hata Terimleriyle Model	0	1	2	475.2879	490.2982	61.4917	7.8417	6.5198	40.9540
Genişletilmiş Karma	1	1	2	472.7207	491.4836	57.7251	7.5977	6.2871	39.5731
Daha Karmaşık Karma	2	1	2	474.1375	496.6529	59.1311	7.6897	6.3769	40.1266

Çizelge B.2 : V2 veri seti SARIMA model performansları.

Model İsmi	p	d	q	P	D	Q	s	AIC	BIC	MSE	RMSE	MAE	MAPE
Basit SARIMA	1	1	1	1	1	1	7	716.4989	739.0144	48.9578	6.9970	5.5560	33.2548
Mevsimsel SARIMA	2	1	2	1	1	1	7	703.7695	726.2850	137.2229	11.7142	9.7977	58.4507
Yüksek Dereceli SARIMA	3	1	3	2	1	2	7	903.9762	926.4917	67.3162	8.2046	6.7757	41.3879
Aylık SARIMA	1	1	1	1	1	1	12	1105.6760	1128.1914	65535	65535	4.5304E+155	2.5168E+156
Hibrit SARIMA	2	1	1	1	1	2	7	703.7695	726.2850	137.2205	11.7141	9.7976	58.4503

Çizelge B.3 : V2 veri seti Holt-Winters model performansları.

Model İsmi	beta	gamma	MSE	RMSE	MAE	MAPE
Holt Doğrusal Trend	1	0	119.9644	10.9528	9.5511	45.4845
Holt-Winters Katı Mevsimsel	0	1	119.9644	10.9528	9.5511	45.4845
Holt-Winters Adaptif Mevsimsel	1	1	119.9644	10.9528	9.5511	45.4845
Basit Üstel Düzgünleştirme	0	0	119.9644	10.9528	9.5511	45.4845

Çizelge B.4 : V2 veri seti LSTM model performansları.

Katman	Gizli Birim	Öğrenme Oranı	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE
1	100	0.001	0	YANLIŞ	2.4120	1.5530	1.2249	7.0725
1	100	0.001	0	DOĞRU	2.1323	1.4602	1.1561	6.5062
1	100	0.001	0.2	YANLIŞ	1.5715	1.2536	0.9520	5.3549
1	100	0.001	0.2	DOĞRU	1.8525	1.3611	1.0693	6.1041
1	100	0.005	0	YANLIŞ	1.5315	1.2375	0.9956	5.7132
1	100	0.005	0	DOĞRU	1.6111	1.2693	0.9864	5.6804
1	100	0.005	0.2	YANLIŞ	1.4196	1.1915	0.9786	5.5363
1	100	0.005	0.2	DOĞRU	1.7340	1.3168	1.0477	5.9604
1	200	0.001	0	YANLIŞ	1.9205	1.3858	1.1331	6.2522
1	200	0.001	0	DOĞRU	1.9152	1.3839	1.1132	6.0389
1	200	0.001	0.2	YANLIŞ	1.9073	1.3811	1.0748	6.1498
1	200	0.001	0.2	DOĞRU	1.9410	1.3932	1.1028	6.4237
1	200	0.005	0	YANLIŞ	1.8487	1.3597	1.1253	6.1815
1	200	0.005	0	DOĞRU	1.9135	1.3833	1.0964	6.1769
1	200	0.005	0.2	YANLIŞ	1.8474	1.3592	1.0285	5.7745
1	200	0.005	0.2	DOĞRU	1.6821	1.2970	1.0037	5.5700
2	100	0.001	0	YANLIŞ	1.6545	1.2863	1.0219	5.4496
2	100	0.001	0	DOĞRU	1.9249	1.3874	1.0614	5.9465
2	100	0.001	0.2	YANLIŞ	1.9490	1.3961	1.0940	6.1814

Çizelge B.4 (devam) : V2 veri seti LSTM model performansları.

Katman	Gizli Birim	Öğrenme Oranı	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE
2	100	0.001	0.2	DOĞRU	1.7987	1.3412	1.0577	5.9944
2	100	0.005	0	YANLIŞ	1.8501	1.3602	1.0908	5.8377
2	100	0.005	0	DOĞRU	1.4666	1.2110	1.0288	5.8707
2	100	0.005	0.2	YANLIŞ	2.2140	1.4879	1.1439	6.6092
2	100	0.005	0.2	DOĞRU	1.6442	1.2823	1.0531	5.8481
2	200	0.001	0	YANLIŞ	1.9385	1.3923	1.1493	6.3413
2	200	0.001	0	DOĞRU	1.5986	1.2644	1.0007	5.5208
2	200	0.001	0.2	YANLIŞ	1.5302	1.2370	1.0115	5.8389
2	200	0.001	0.2	DOĞRU	1.6431	1.2818	0.9935	5.6724
2	200	0.005	0	YANLIŞ	2.3422	1.5304	1.1935	6.8420
2	200	0.005	0	DOĞRU	1.9103	1.3821	1.0927	6.0698
2	200	0.005	0.2	YANLIŞ	1.4053	1.1855	0.9099	4.7693
2	200	0.005	0.2	DOĞRU	2.1238	1.4573	1.1572	6.6603

Çizelge B.5 : V2 veri seti Random Forest model performansları.

Ağaç	Derinlik	Yaprak (Min)	MSE	RMSE	MAE	MAPE	SMAPE	MBE
50	3	1	39.9837	6.3233	4.8297	23.9698	39.9837	6.3233
50	3	5	40.5796	6.3702	4.8396	23.9392	40.5796	6.3702
50	5	1	40.1208	6.3341	4.7604	23.6282	40.1208	6.3341
50	5	5	38.6505	6.2170	4.7051	23.6158	38.6505	6.2170
100	3	1	40.8788	6.3937	4.8262	23.6886	40.8788	6.3937
100	3	5	41.0122	6.4041	4.8604	23.9426	41.0122	6.4041
100	5	1	41.1525	6.4150	4.8426	23.7644	41.1525	6.4150
100	5	5	40.1832	6.3390	4.7734	23.4830	40.1832	6.3390

Çizelge B.6 : V2 veri seti SVR model performansları.

Kutu (C)	Hata Payı	Çekirdek	Ölçek	MSE	RMSE	MAE	MAPE
0.1	0.01	linear	auto	39.1845	6.2598	4.7854	23.3447
0.1	0.01	linear	1	30.6083	5.5325	4.1024	22.1240
0.1	0.01	gaussian	auto	113.1642	10.6379	9.0545	42.6828
0.1	0.01	gaussian	1	135.7784	11.6524	10.3313	49.7420
0.1	0.1	linear	auto	39.4309	6.2794	4.8038	23.4140
0.1	0.1	linear	1	30.0950	5.4859	4.0742	22.1639
0.1	0.1	gaussian	auto	115.1942	10.7329	9.1608	43.2323
0.1	0.1	gaussian	1	135.7784	11.6524	10.3313	49.7420
1	0.01	linear	auto	30.7100	5.5417	4.1248	22.0318
1	0.01	linear	1	30.5399	5.5263	4.0981	22.1224
1	0.01	gaussian	auto	83.0575	9.1136	7.2124	33.6485
1	0.01	gaussian	1	135.8054	11.6536	10.3325	49.7490
1	0.1	linear	auto	31.6647	5.6271	4.2030	22.0714
1	0.1	linear	1	30.4086	5.5144	4.1107	22.1005
1	0.1	gaussian	auto	80.9096	8.9950	7.1019	33.0937
1	0.1	gaussian	1	135.8984	11.6575	10.3369	49.7731

Çizelge B.7 : V2 veri seti LightGBM model performansları.

Yaprak Sayısı	Öğrenme Oranı	Tahmini Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
15	0.01	100	8.3324	6.8126	31.5705	-1.2419	37.9532	-6.1309
15	0.01	200	7.3591	5.7979	27.4000	-0.7488	31.1338	-4.6433
15	0.01	300	7.0945	5.5066	26.4665	-0.6252	29.3065	-4.0392
15	0.05	100	6.9915	5.4426	26.4995	-0.5784	28.9010	-3.6981
15	0.05	200	7.2181	5.5967	27.2273	-0.6824	29.8270	-3.7649
15	0.05	300	7.2155	5.6326	27.5318	-0.6812	30.0466	-3.7185
15	0.1	100	7.1418	5.5427	27.1729	-0.6470	29.5361	-3.6313
15	0.1	200	7.3002	5.6991	27.9804	-0.7209	30.5104	-3.7297

Çizelge B.7 (devam): V2 veri seti LightGBM model performansları.

Yaprak Sayısı	Öğrenme Oranı	Tahminci Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
15	0.1	300	7.3174	5.7251	28.1133	-0.7290	30.6646	-3.7331
31	0.01	100	8.3324	6.8126	31.5705	-1.2419	37.9532	-6.1309
31	0.01	200	7.3591	5.7979	27.4000	-0.7488	31.1338	-4.6433
31	0.01	300	7.0945	5.5066	26.4665	-0.6252	29.3065	-4.0392
31	0.05	100	6.9915	5.4426	26.4995	-0.5784	28.9010	-3.6981
31	0.05	200	7.2181	5.5967	27.2273	-0.6824	29.8270	-3.7649
31	0.05	300	7.2155	5.6326	27.5318	-0.6812	30.0466	-3.7185
31	0.1	100	7.1418	5.5427	27.1729	-0.6470	29.5361	-3.6313
31	0.1	200	7.3002	5.6991	27.9804	-0.7209	30.5104	-3.7297
31	0.1	300	7.3174	5.7251	28.1133	-0.7290	30.6646	-3.7331
50	0.01	100	8.3324	6.8126	31.5705	-1.2419	37.9532	-6.1309
50	0.01	200	7.3591	5.7979	27.4000	-0.7488	31.1338	-4.6433
50	0.01	300	7.0945	5.5066	26.4665	-0.6252	29.3065	-4.0392
50	0.05	100	6.9915	5.4426	26.4995	-0.5784	28.9010	-3.6981
50	0.05	200	7.2181	5.5967	27.2273	-0.6824	29.8270	-3.7649
50	0.05	300	7.2155	5.6326	27.5318	-0.6812	30.0466	-3.7185
50	0.1	100	7.1418	5.5427	27.1729	-0.6470	29.5361	-3.6313
50	0.1	200	7.3002	5.6991	27.9804	-0.7209	30.5104	-3.7297
50	0.1	300	7.3174	5.7251	28.1133	-0.7290	30.6646	-3.7331

Çizelge B.8 : V2 veri seti XGBoost model performansları.

Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
200	0.05	3	1	0.5	6.7107	5.3737	26.7213	-0.4542	28.7004	-3.3727
100	0.05	3	0.7	0.7	6.7455	5.3225	26.1086	-0.4693	28.1793	-3.5682
200	0.05	3	0.7	0.7	6.7507	5.3492	26.8375	-0.4716	28.3803	-3.2190

Çizelge B.8 (devam) : V2 veri seti XGBoost model performansları.

Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
100	0.1	3	0.5	0.5	6.7688	5.2602	26.1055	-0.4794	27.8397	-3.3093
100	0.05	3	1	0.5	6.8203	5.4442	26.6351	-0.5021	29.0239	-3.7489
200	0.1	3	0.7	0.7	6.8229	5.4241	27.4352	-0.5032	28.9158	-3.1324
100	0.1	3	0.7	0.7	6.8425	5.4584	27.2020	-0.5119	29.0787	-3.3896
100	0.1	3	0.5	0.7	6.8498	5.4371	27.2151	-0.5151	28.9742	-3.2596
100	0.05	3	1	0.7	6.8633	5.5121	26.8681	-0.5211	29.4335	-3.8563
100	0.05	3	0.7	0.5	6.8636	5.4183	26.1662	-0.5212	28.8318	-3.8823
100	0.05	3	0.5	1	6.8644	5.3806	26.2432	-0.5216	28.5156	-3.6750
300	0.05	3	1	0.5	6.8767	5.4846	27.2635	-0.5270	29.3842	-3.4202
200	0.05	3	1	0.7	6.8851	5.5832	27.6840	-0.5307	29.9468	-3.5372
300	0.1	5	1	1	6.8862	5.4779	27.7816	-0.5312	29.3375	-2.9958
300	0.1	3	0.7	0.7	6.8972	5.4851	28.0048	-0.5361	29.2955	-3.0133
300	0.05	3	0.7	0.7	6.8975	5.4314	27.2511	-0.5362	28.8370	-3.2760
200	0.1	5	1	1	6.8982	5.4787	27.7360	-0.5366	29.3448	-3.0264
100	0.1	3	0.7	0.5	6.9032	5.3364	25.6807	-0.5388	28.3959	-3.7673
300	0.05	3	0.5	0.5	6.9043	5.4171	26.8070	-0.5393	28.8529	-3.4573
200	0.05	3	0.7	0.5	6.9214	5.4479	26.7324	-0.5469	29.0210	-3.6132
100	0.1	5	1	1	6.9304	5.5058	27.6252	-0.5509	29.5261	-3.2400
300	0.01	3	0.5	1	6.9377	5.4157	25.9050	-0.5542	28.7072	-4.0464
300	0.05	3	1	0.7	6.9387	5.6186	28.0453	-0.5547	30.1507	-3.3850
100	0.05	3	0.5	0.5	6.9513	5.4380	26.1906	-0.5603	28.8757	-3.9407
100	0.05	3	0.7	1	6.9546	5.4899	26.7859	-0.5618	29.2465	-3.7718
200	0.05	3	0.5	0.5	6.9693	5.4527	26.6376	-0.5684	28.9849	-3.7142
200	0.1	3	0.7	0.5	6.9870	5.4633	26.9337	-0.5764	29.2501	-3.4897

Çizelge B.9 : V2 veri seti Prophet model performansları.

Değişim Nokt. Ölç.	Mevsimsellik	Tatıl Ölçeği	RMSE	MAE	MAPE	R2	SMAPE	MBE
0.001	additive	1	5.7864	4.3875	21.8480	-0.0850	22.7015	-2.5611
0.001	additive	5	5.7864	4.3875	21.8480	-0.0850	22.7015	-2.5611
0.001	additive	10	5.7864	4.3875	21.8480	-0.0850	22.7015	-2.5611
0.001	multiplicative	1	5.8409	4.3991	21.9371	-0.1055	22.7097	-2.5539
0.001	multiplicative	5	5.8409	4.3991	21.9371	-0.1055	22.7097	-2.5539
0.001	multiplicative	10	5.8409	4.3991	21.9371	-0.1055	22.7097	-2.5539
0.01	additive	1	5.3368	4.0809	21.6453	0.0771	21.0219	-1.3103
0.01	additive	5	5.3368	4.0809	21.6453	0.0771	21.0219	-1.3103
0.01	additive	10	5.3368	4.0809	21.6453	0.0771	21.0219	-1.3103
0.01	multiplicative	1	5.1668	4.0718	22.1496	0.1350	21.0514	-0.8185
0.01	multiplicative	5	5.1668	4.0718	22.1496	0.1350	21.0514	-0.8185
0.01	multiplicative	10	5.1668	4.0718	22.1496	0.1350	21.0514	-0.8185
0.05	additive	1	5.2076	4.0113	21.9882	0.1212	20.6710	-0.7161
0.05	additive	5	5.2076	4.0113	21.9882	0.1212	20.6710	-0.7161
0.05	additive	10	5.2076	4.0113	21.9882	0.1212	20.6710	-0.7161
0.05	multiplicative	1	5.1338	4.0511	22.2414	0.1459	20.9357	-0.6484
0.05	multiplicative	5	5.1338	4.0511	22.2414	0.1459	20.9357	-0.6484
0.05	multiplicative	10	5.1338	4.0511	22.2414	0.1459	20.9357	-0.6484
0.1	additive	1	5.1956	4.0050	22.0355	0.1253	20.6400	-0.6480
0.1	additive	5	5.1956	4.0050	22.0355	0.1253	20.6400	-0.6480
0.1	additive	10	5.1956	4.0050	22.0355	0.1253	20.6400	-0.6480
0.1	multiplicative	1	5.1305	4.0495	22.2425	0.1470	20.9271	-0.6386
0.1	multiplicative	5	5.1305	4.0495	22.2425	0.1470	20.9271	-0.6386
0.1	multiplicative	10	5.1305	4.0495	22.2425	0.1470	20.9271	-0.6386

Çizelge B.10 : V2 veri seti GBR model performansları.

Tahminci Sayısı	Maks. Derinlik	Öğrenme Oranı	Min. Yaprak Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
50	3	0.2	1	6.5696	5.2320	26.2122	-0.3937	27.6194	-2.6774
100	3	0.1	1	6.6532	5.2606	26.3183	-0.4294	27.9793	-2.9771
50	3	0.1	1	6.6778	5.2864	26.0472	-0.4399	28.1939	-3.4838
150	3	0.1	1	6.7122	5.3257	26.9275	-0.4548	28.3595	-2.7173
100	3	0.05	1	6.7675	5.3272	26.1197	-0.4789	28.4115	-3.6007
50	3	0.1	5	6.8095	5.3450	26.1061	-0.4973	28.3922	-3.6330
100	3	0.05	10	6.8407	5.4099	26.4057	-0.5110	28.8144	-3.7292
50	3	0.2	10	6.8448	5.4697	27.2310	-0.5129	29.2465	-3.3808
100	3	0.05	5	6.8486	5.3883	26.0932	-0.5145	28.6456	-3.8009
150	3	0.05	1	6.8750	5.3993	26.5544	-0.5263	28.8589	-3.4855
50	3	0.1	10	6.8767	5.4190	26.5950	-0.5270	28.9166	-3.6477
100	3	0.2	1	6.8869	5.5754	28.3855	-0.5315	29.5819	-2.4529
50	5	0.1	10	6.9052	5.4725	27.0008	-0.5397	29.2352	-3.6128
100	5	0.05	10	6.9072	5.4683	26.7885	-0.5406	29.2021	-3.6809
150	3	0.05	10	6.9175	5.4797	26.9710	-0.5452	29.3053	-3.6150
50	5	0.1	5	6.9272	5.4000	26.4428	-0.5495	28.7834	-3.6445

Çizelge C.1 : V3 veri seti ARIMA model performansları.

Model İsmi	p	d	q	AIC	BIC	MSE	RMSE	MAE	MAPE	R2
Random Walk	0	1	0	101.2264	105.0100	19104.5900	138.2194	137.0068	100.1887	-56.3963
AR(1)	1	0	0	55.1075	60.7829	19014.5033	137.8931	136.6796	99.9466	-56.1256
AR(2)	2	0	0	52.2987	59.8660	19014.7802	137.8941	136.6798	99.9461	-56.1265
MA(1)	0	0	1	52.0847	57.7602	19013.5800	137.8897	136.6752	99.9424	-56.1229
MA(2)	0	0	2	50.8680	58.4352	19012.8513	137.8871	136.6721	99.9398	-56.1207
Durağan Olmayan AR(1)	1	1	0	87.5509	93.2264	19068.3719	138.0883	136.8744	100.0899	-56.2875
Durağan Olmayan MA(1)	0	1	1	64.1457	69.8212	19075.2392	138.1131	136.9002	100.1098	-56.3081
Karma Model	1	1	1	63.6349	71.2022	19064.0089	138.0725	136.8590	100.0789	-56.2744
Hata Terimleriyle Model	0	1	2	63.8405	71.4078	19056.8402	138.0465	136.8327	100.0593	-56.2528
Genişletilmiş Karma	1	1	2	65.6104	75.0695	19061.3783	138.0630	136.8493	100.0717	-56.2665
Daha Karmaşık Karma	2	1	2	60.0978	71.4487	19069.5728	138.0926	136.8797	100.0947	-56.2911

Çizelge C.2 : V3 veri seti SARIMA model performansları.

Model İsmi	p	d	q	P	D	Q	s	AIC	BIC	MSE	RMSE	MAE	MAPE	R2
Basit SARIMA	1	1	1	1	1	1	7	103.3370	114.6879	19001.9301	137.8475	136.6386	99.9199	-56.0879
Mevsimsel SARIMA	2	1	2	1	1	1	7	87.4306	98.7816	18953.1241	137.6703	136.4681	99.7995	-55.9412
Yüksek Dereceli SARIMA	3	1	3	2	1	2	7	128.8172	140.1682	19172.5869	138.4651	137.2415	100.3537	-56.6006
Aylık SARIMA	1	1	1	1	1	1	12	79.1385	90.4895	18998.8443	137.8363	136.6193	99.8994	-56.0786
Hibrit SARIMA	2	1	1	1	1	2	7	87.4306	98.7816	18953.1241	137.6703	136.4681	99.7995	-55.9412

Çizelge C.3 : V3 veri seti Holt-Winters model performansları.

Model İsmi	beta	gamma	MSE	RMSE	MAE	MAPE	R2
Holt Doğrusal Trend	1	0	616.7767	24.8350	19.9667	13.5765	-0.8530
Holt-Winters Katı Mevsimsel	0	1	616.7767	24.8350	19.9667	13.5765	-0.8530
Holt-Winters Adaptif Mevsimsel	1	1	616.7767	24.8350	19.9667	13.5765	-0.8530
Basit Üstel Düzgünleştirme	0	0	616.7767	24.8350	19.9667	13.5765	-0.8530

Çizelge C.4 : V3 veri seti LSTM model performansları.

Katman	Gizli Birim	Öğrenme Oranı	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE	R2
1	100	0.001	0	YANLIŞ	39.3970	6.2767	5.6892	4.3144	0.8816
1	100	0.001	0	DOĞRU	85.9592	9.2714	7.5332	5.6073	0.7418
1	100	0.001	0.2	YANLIŞ	72.6439	8.5231	6.7505	4.8189	0.7818
1	100	0.001	0.2	DOĞRU	30.4950	5.5222	4.5851	3.5097	0.9084
1	100	0.005	0	YANLIŞ	51.6148	7.1843	6.3479	4.9738	0.8449
1	100	0.005	0	DOĞRU	96.7854	9.8380	7.8449	5.7768	0.7092
1	100	0.005	0.2	YANLIŞ	35.8123	5.9843	5.1740	3.8088	0.8924
1	100	0.005	0.2	DOĞRU	53.7751	7.3331	6.4462	4.9716	0.8384
1	200	0.001	0	YANLIŞ	53.4546	7.3113	6.0289	4.4004	0.8394
1	200	0.001	0	DOĞRU	53.6655	7.3257	6.4028	4.6793	0.8388
1	200	0.001	0.2	YANLIŞ	22.9825	4.7940	3.8275	2.8191	0.9310
1	200	0.001	0.2	DOĞRU	54.7279	7.3978	6.5672	4.8255	0.8356
1	200	0.005	0	YANLIŞ	48.2985	6.9497	5.8059	4.4123	0.8549
1	200	0.005	0	DOĞRU	44.8840	6.6995	5.7239	4.2046	0.8652
1	200	0.005	0.2	YANLIŞ	79.0931	8.8934	7.7631	6.0712	0.7624
1	200	0.005	0.2	DOĞRU	35.1627	5.9298	4.7128	3.6345	0.8944
2	100	0.001	0	YANLIŞ	33.1149	5.7546	4.6348	3.5031	0.9005
2	100	0.001	0	DOĞRU	17.0386	4.1278	2.9164	2.2663	0.9488
2	100	0.001	0.2	YANLIŞ	46.3866	6.8108	4.8675	3.4112	0.8606

Çizelge C.4 (devam): V3 veri seti LSTM model performansları.

Katman	Gizli Birim	Öğrenme Oranı	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE	R2
2	100	0.001	0.2	DOĞRU	88.0353	9.3827	6.6719	5.1684	0.7355
2	100	0.005	0	YANLIŞ	49.6167	7.0439	5.5157	4.2368	0.8509
2	100	0.005	0	DOĞRU	22.0189	4.6924	3.9275	2.9529	0.9338
2	100	0.005	0.2	YANLIŞ	56.2656	7.5010	5.9173	4.4466	0.8310
2	100	0.005	0.2	DOĞRU	65.1259	8.0701	6.8745	5.2702	0.8043
2	200	0.001	0	YANLIŞ	52.4347	7.2412	5.7436	4.3716	0.8425
2	200	0.001	0	DOĞRU	60.3230	7.7668	6.4406	5.0445	0.8188
2	200	0.001	0.2	YANLIŞ	59.4155	7.7081	6.4706	4.8255	0.8215
2	200	0.001	0.2	DOĞRU	46.7266	6.8357	4.4973	3.5397	0.8596
2	200	0.005	0	YANLIŞ	36.3672	6.0305	5.3217	4.0204	0.8907
2	200	0.005	0	DOĞRU	68.7413	8.2910	6.3405	4.8613	0.7935
2	200	0.005	0.2	YANLIŞ	36.6674	6.0554	5.0861	3.8390	0.8898
2	200	0.005	0.2	DOĞRU	59.1148	7.6886	7.0657	5.4133	0.8224

Çizelge C.5 : V3 veri seti Random Forest model performansları.

Ağaç	Derinlik	Yaprak (Min)	MSE	RMSE	MAE	MAPE	R2
50	3	1	830.4187	28.8170	23.5571	15.9664	-1.4948
50	3	5	970.7803	31.1573	25.7416	17.4641	-1.9165
50	5	1	984.0041	31.3688	26.1042	17.7482	-1.9563
50	5	5	1066.3531	32.6551	27.3268	18.6002	-2.2037
100	3	1	965.9692	31.0800	25.7255	17.4632	-1.9021
100	3	5	922.1660	30.3672	24.9452	16.9011	-1.7705
100	5	1	893.9738	29.8994	24.4946	16.5908	-1.6858
100	5	5	959.1772	30.9706	25.6551	17.4244	-1.8817

Çizelge C.6 : V3 veri seti SVR model performansları.

Kutu (C)	Hata Payı	Çekirdek	Ölçek	MSE	RMSE	MAE	MAPE	R2
0.1	0.01	linear	auto	3437.0123	58.6260	55.9627	39.9544	-9.3259
0.1	0.01	linear	1	232.7828	15.2572	12.8536	9.0855	0.3006
0.1	0.01	gaussian	auto	4205.3411	64.8486	62.2014	44.5057	-11.6342
0.1	0.01	gaussian	1	4138.5103	64.3313	61.6900	44.1348	-11.4334
0.1	0.1	linear	auto	3346.5816	57.8496	55.1799	39.3828	-9.0542
0.1	0.1	linear	1	230.4399	15.1802	12.7985	9.0457	0.3077
0.1	0.1	gaussian	auto	4191.7315	64.7436	62.0941	44.4266	-11.5933
0.1	0.1	gaussian	1	4127.4142	64.2450	61.6000	44.0678	-11.4001
1	0.01	linear	auto	594.5938	24.3843	19.5943	13.3196	-0.7863
1	0.01	linear	1	231.2796	15.2079	12.8247	9.0659	0.3052
1	0.01	gaussian	auto	4440.7693	66.6391	63.8410	45.6496	-12.3415
1	0.01	gaussian	1	4083.1918	63.8999	61.2400	43.7999	-11.2672
1	0.1	linear	auto	596.5495	24.4244	19.6403	13.3544	-0.7922
1	0.1	linear	1	229.6621	15.1546	12.7847	9.0414	0.3100
1	0.1	gaussian	auto	4441.4650	66.6443	63.8515	45.6591	-12.3436
1	0.1	gaussian	1	4072.1767	63.8136	61.1500	43.7329	-11.2341

Çizelge C.7 : V3 veri seti LightGBM model performansları.

Yaprak Sayısı	Öğrenme Oranı	Tahminci Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
15	0.01	100	56.1626	53.6013	37.5856	-10.2194	46.7883	-53.6013
15	0.01	200	47.5881	44.5363	30.9885	-7.0551	37.2182	-44.5363
15	0.01	300	43.3042	39.9263	27.6335	-5.6702	32.6318	-39.9263
15	0.05	100	40.1969	36.5329	25.1639	-4.7473	29.3680	-36.5329
15	0.05	200	39.6181	35.8950	24.6997	-4.5830	28.7648	-35.8950
15	0.05	300	39.1826	35.4137	24.3494	-4.4609	28.3117	-35.4137
15	0.1	100	39.4537	35.7135	24.5676	-4.5367	28.5937	-35.7135

Çizelge C.7 (devam) : V3 veri seti LightGBM model performansları.

Yaprak Sayısı	Öğrenme Oranı	Tahminci Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
15	0.1	200	38.8265	35.0194	24.0624	-4.3621	27.9418	-35.0194
15	0.1	300	38.4888	34.6446	23.7897	-4.2692	27.5914	-34.6446
31	0.01	100	56.1626	53.6013	37.5856	-10.2194	46.7883	-53.6013
31	0.01	200	47.5881	44.5363	30.9885	-7.0551	37.2182	-44.5363
31	0.01	300	43.3042	39.9263	27.6335	-5.6702	32.6318	-39.9263
31	0.05	100	40.1969	36.5329	25.1639	-4.7473	29.3680	-36.5329
31	0.05	200	39.6181	35.8950	24.6997	-4.5830	28.7648	-35.8950
31	0.05	300	39.1826	35.4137	24.3494	-4.4609	28.3117	-35.4137
31	0.1	100	39.4537	35.7135	24.5676	-4.5367	28.5937	-35.7135
31	0.1	200	38.8265	35.0194	24.0624	-4.3621	27.9418	-35.0194
31	0.1	300	38.4888	34.6446	23.7897	-4.2692	27.5914	-34.6446
50	0.01	100	56.1626	53.6013	37.5856	-10.2194	46.7883	-53.6013
50	0.01	200	47.5881	44.5363	30.9885	-7.0551	37.2182	-44.5363
50	0.01	300	43.3042	39.9263	27.6335	-5.6702	32.6318	-39.9263
50	0.05	100	40.1969	36.5329	25.1639	-4.7473	29.3680	-36.5329
50	0.05	200	39.6181	35.8950	24.6997	-4.5830	28.7648	-35.8950
50	0.05	300	39.1826	35.4137	24.3494	-4.4609	28.3117	-35.4137
50	0.1	100	39.4537	35.7135	24.5676	-4.5367	28.5937	-35.7135
50	0.1	200	38.8265	35.0194	24.0624	-4.3621	27.9418	-35.0194
50	0.1	300	38.4888	34.6446	23.7897	-4.2692	27.5914	-34.6446

Çizelge C.8 : V3 veri seti XGBoost model performansları.

Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
300	0.05	7	1	1	16.2397	13.1577	9.3851	0.0619	9.3937	-0.8391

Çizelge C.8 (devam) : V3 veri seti XGBoost model performansları.

Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
200	0.05	7	1	1	16.2688	13.0364	9.2665	0.0586	9.3051	-1.2836
300	0.05	5	0.5	1	16.3122	12.3087	8.5577	0.0535	8.7427	-3.5066
300	0.05	7	0.5	1	16.3631	12.5622	8.7862	0.0476	8.9377	-2.9318
300	0.1	7	1	1	16.7633	13.8214	9.9514	0.0005	9.8745	0.1403
200	0.1	7	1	1	16.7633	13.8211	9.9511	0.0005	9.8743	0.1390
100	0.1	7	1	1	16.7657	13.6979	9.8307	0.0002	9.7860	-0.3126
300	0.1	7	0.5	0.7	17.0478	12.2942	8.3183	-0.0337	8.6973	-6.6358
200	0.1	7	0.5	0.7	17.1850	12.3799	8.3608	-0.0505	8.7603	-6.9635
300	0.1	7	0.7	1	17.1941	13.6283	9.6220	-0.0516	9.7342	-2.1640
200	0.1	7	0.7	1	17.2017	13.6246	9.6158	-0.0525	9.7315	-2.2150
200	0.1	5	1	1	17.3126	14.0654	10.0721	-0.0661	10.0496	-0.6033
300	0.1	5	1	1	17.3137	14.0666	10.0730	-0.0662	10.0504	-0.6024
200	0.05	7	0.5	1	17.3163	12.5615	8.5601	-0.0666	8.9206	-6.1582
300	0.05	7	0.7	1	17.3173	13.6793	9.7076	-0.0667	9.7718	-1.6253
100	0.1	5	1	1	17.3346	13.9848	9.9921	-0.0688	9.9922	-0.9199
300	0.05	7	0.5	0.7	17.3808	12.6233	8.6007	-0.0745	8.9673	-6.1428
200	0.05	5	0.5	1	17.3905	12.6098	8.5715	-0.0757	8.9532	-6.5732
100	0.1	7	0.7	1	17.4463	13.5575	9.4843	-0.0826	9.6827	-3.4153
300	0.05	5	1	1	17.4895	14.2041	10.1802	-0.0880	10.1487	-0.5509
200	0.05	5	1	1	17.4952	14.1145	10.0947	-0.0887	10.0849	-0.8553
200	0.05	7	0.7	1	17.5655	13.5500	9.5156	-0.0975	9.6782	-3.0547
300	0.1	7	0.5	1	17.6096	13.5233	9.4756	-0.1030	9.6503	-3.0461
100	0.05	7	1	1	17.6150	13.2542	9.0387	-0.1037	9.4451	-6.8748
300	0.1	5	0.7	1	17.6687	13.5244	9.4198	-0.1104	9.6586	-3.9239

Çizelge C.9 : V3 veri seti Prophet model performansları.

Değişim Nokt. Ölç.	Mevsimsellik	Tatil Ölçeği	RMSE	MAE	MAPE	R2	SMAPE	MBE
0.001	additive	1	22.5998	17.7138	12.1161	-0.4706	13.1855	-14.0927
0.001	additive	5	22.5998	17.7138	12.1161	-0.4706	13.1855	-14.0927
0.001	additive	10	22.5998	17.7138	12.1161	-0.4706	13.1855	-14.0927
0.001	multiplicative	1	22.5998	17.7138	12.1161	-0.4706	13.1855	-14.0927
0.001	multiplicative	5	22.5998	17.7138	12.1161	-0.4706	13.1855	-14.0927
0.001	multiplicative	10	22.5998	17.7138	12.1161	-0.4706	13.1855	-14.0927
0.01	additive	1	23.4949	18.2649	12.4264	-0.5894	13.6285	-15.4545
0.01	additive	5	23.4949	18.2649	12.4264	-0.5894	13.6285	-15.4545
0.01	additive	10	23.4949	18.2649	12.4264	-0.5894	13.6285	-15.4545
0.01	multiplicative	1	23.4949	18.2649	12.4264	-0.5894	13.6285	-15.4545
0.01	multiplicative	5	23.4949	18.2649	12.4264	-0.5894	13.6285	-15.4545
0.01	multiplicative	10	23.4949	18.2649	12.4264	-0.5894	13.6285	-15.4545
0.05	additive	1	23.6263	18.3437	12.4709	-0.6073	13.6922	-15.6485
0.05	additive	5	23.6263	18.3437	12.4709	-0.6073	13.6922	-15.6485
0.05	additive	10	23.6263	18.3437	12.4709	-0.6073	13.6922	-15.6485
0.05	multiplicative	1	23.6263	18.3437	12.4709	-0.6073	13.6922	-15.6485
0.05	multiplicative	5	23.6263	18.3437	12.4709	-0.6073	13.6922	-15.6485
0.05	multiplicative	10	23.6263	18.3437	12.4709	-0.6073	13.6922	-15.6485
0.1	additive	1	23.6324	18.3473	12.4729	-0.6081	13.6951	-15.6576
0.1	additive	5	23.6324	18.3473	12.4729	-0.6081	13.6951	-15.6576
0.1	additive	10	23.6324	18.3473	12.4729	-0.6081	13.6951	-15.6576
0.1	multiplicative	1	23.6324	18.3473	12.4729	-0.6081	13.6951	-15.6576
0.1	multiplicative	5	23.6324	18.3473	12.4729	-0.6081	13.6951	-15.6576
0.1	multiplicative	10	23.6324	18.3473	12.4729	-0.6081	13.6951	-15.6576

Çizelge C.10 : V3 veri seti GBR model performansları.

Tahminci Sayısı	Maks. Derinlik	Öğrenme Oranı	Min. Yaprak Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
50	5	0.2	1	16.1055	13.0925	9.3030	0.0774	9.3437	-1.1276
150	5	0.2	1	16.1057	13.0925	9.3029	0.0774	9.3437	-1.1280
100	5	0.2	1	16.1057	13.0925	9.3029	0.0774	9.3437	-1.1280
150	5	0.1	1	16.4160	13.1659	9.3296	0.0415	9.3998	-1.7295
100	5	0.1	1	16.4163	13.1652	9.3287	0.0414	9.3992	-1.7350
50	5	0.1	1	16.4569	13.1439	9.2928	0.0367	9.3831	-2.0330
150	5	0.05	1	16.5694	13.1872	9.3110	0.0235	9.4142	-2.2919
100	5	0.05	1	16.6320	13.1694	9.2700	0.0161	9.4004	-2.6967
150	7	0.2	1	17.2906	12.8387	8.6264	-0.0634	9.0989	-7.5347
100	7	0.2	1	17.2906	12.8387	8.6264	-0.0634	9.0989	-7.5347
50	7	0.2	1	17.2910	12.8389	8.6265	-0.0635	9.0991	-7.5357
150	7	0.05	1	17.7058	13.4356	9.0875	-0.1151	9.5650	-7.5012
100	7	0.05	1	17.8760	13.5429	9.1396	-0.1366	9.6436	-7.8945
50	5	0.05	1	18.1205	14.0008	9.5366	-0.1679	10.0071	-7.6826
50	3	0.1	1	18.4149	14.3651	9.9272	-0.2062	10.2818	-6.2068
100	3	0.05	1	18.6636	14.5967	10.0375	-0.2390	10.4579	-7.0135

Çizelge D.1 : V4 veri seti ile 24 saatlik en iyi ARIMA model performansları.

Saat	Model İsmi	p	d	q	MSE	RMSE	MAE	MAPE
0	Durağan Olmayan MA(1)	0	1	1	2.0263	1.4235	1.2269	65535
1	Durağan Olmayan MA(1)	0	1	1	1.1241	1.0602	0.7794	65535
2	Durağan Olmayan MA(1)	0	1	1	0.4485	0.6697	0.5930	65535
3	Durağan Olmayan MA(1)	0	1	1	0.3135	0.5599	0.3435	65535
4	Durağan Olmayan MA(1)	0	1	1	0.3374	0.5808	0.3381	65535
5	MA(1)	0	0	1	0.0454	0.2130	0.0922	65535
6	Durağan Olmayan MA(1)	0	1	1	0.2116	0.4601	0.3655	65535
7	Durağan Olmayan MA(1)	0	1	1	0.3591	0.5992	0.5493	65535
8	Durağan Olmayan MA(1)	0	1	1	3.4567	1.8592	1.5955	65535
9	Durağan Olmayan MA(1)	0	1	1	8.1336	2.8519	2.3498	65535
10	Hata Terimleriyle Model	0	1	2	6.3118	2.5123	1.9790	65535
11	Durağan Olmayan MA(1)	0	1	1	4.8305	2.1978	1.6668	65535
12	Durağan Olmayan MA(1)	0	1	1	7.5666	2.7507	2.3872	65535
13	Genişletilmiş Karma	1	1	2	4.1793	2.0443	1.7550	52.6067
14	Durağan Olmayan MA(1)	0	1	1	11.3011	3.3617	2.5475	52.8234
15	AR(2)	2	0	0	18.6619	4.3199	3.1815	65535
16	Durağan Olmayan MA(1)	0	1	1	22.8302	4.7781	4.2002	117.4252
17	Durağan Olmayan MA(1)	0	1	1	27.4552	5.2398	4.3960	65535
18	Durağan Olmayan MA(1)	0	1	1	10.6531	3.2639	2.6586	88.5241
19	Durağan Olmayan MA(1)	0	1	1	30.6249	5.5340	4.6439	65535
20	Durağan Olmayan MA(1)	0	1	1	11.8044	3.4358	2.8200	71.3162
21	Durağan Olmayan MA(1)	0	1	1	13.4682	3.6699	3.1088	158.6117
22	Durağan Olmayan MA(1)	0	1	1	8.5950	2.9317	2.3686	43.7641
23	Durağan Olmayan MA(1)	0	1	1	5.3451	2.3119	2.0862	65535

Çizelge D.2 : V4 veri seti ile 24 saatlik en iyi SARIMA model performansları.

Saat	Model İsmi	p	d	q	P	D	Q	s	MSE	RMSE	MAE	MAPE
0	Aylık SARIMA	1	1	1	1	1	1	12	3.7473	1.9358	1.5344	65535
1	Mevsimsel SARIMA	2	1	2	1	1	1	7	1.3948	1.1810	1.0219	65535
2	Mevsimsel SARIMA	2	1	2	1	1	1	7	0.4786	0.6918	0.3918	65535
3	Mevsimsel SARIMA	2	1	2	1	1	1	7	0.4279	0.6541	0.4525	65535
4	Basit SARIMA	1	1	1	1	1	1	7	0.7996	0.8942	0.5631	65535
5	Hibrit SARIMA	2	1	1	1	1	2	7	0.0486	0.2204	0.1003	65535
6	Mevsimsel SARIMA	2	1	2	1	1	1	7	0.3303	0.5747	0.3633	65535
7	Hibrit SARIMA	2	1	1	1	1	2	7	0.5964	0.7722	0.5366	65535
8	Aylık SARIMA	1	1	1	1	1	1	12	49.6159	7.0439	4.8490	65535
9	Mevsimsel SARIMA	2	1	2	1	1	1	7	5.5536	2.3566	1.7231	65535
10	Hibrit SARIMA	2	1	1	1	1	2	7	5.1466	2.2686	1.8527	65535
11	Hibrit SARIMA	2	1	1	1	1	2	7	4.3216	2.0788	1.5812	65535
12	Aylık SARIMA	1	1	1	1	1	1	12	150.4212	12.2646	9.4450	65535
13	Mevsimsel SARIMA	2	1	2	1	1	1	7	11.0118	3.3184	2.8290	75.7489
14	Hibrit SARIMA	2	1	1	1	1	2	7	19.0132	4.3604	3.6939	76.5718
15	Aylık SARIMA	1	1	1	1	1	1	12	17.3551	4.1659	2.9882	65535
16	Hibrit SARIMA	2	1	1	1	1	2	7	10.3682	3.2200	2.6304	64.7523
17	Aylık SARIMA	1	1	1	1	1	1	12	81.9447	9.0523	7.2712	65535
18	Aylık SARIMA	1	1	1	1	1	1	12	34.1895	5.8472	4.7512	135.1936
19	Mevsimsel SARIMA	2	1	2	1	1	1	7	118.0173	10.8636	9.0067	65535
20	Hibrit SARIMA	2	1	1	1	1	2	7	54.0330	7.3507	5.4236	111.6600
21	Aylık SARIMA	1	1	1	1	1	1	12	34.2197	5.8498	4.8651	230.5548
22	Mevsimsel SARIMA	2	1	2	1	1	1	7	17.0056	4.1238	3.6310	75.6220
23	Aylık SARIMA	1	1	1	1	1	1	12	15.6801	3.9598	3.0794	65535

Çizelge D.3 : V4 veri seti ile 24 saatlik en iyi Holt-Winters model performansları.

Saat	alpha	beta	MSE	RMSE	MAE	MAPE
13	0.20	0.80	9.7685	3.1255	2.6475	87.4066
14	0.20	0.20	7.9963	2.8278	2.4978	78.0262
16	0.80	0.00	9.8393	3.1368	2.6198	45.2479
18	0.40	0.60	10.6684	3.2662	2.5185	60.9988
20	0.20	0.00	2.4876	1.5772	1.3490	32.9680
21	0.20	0.80	7.2957	2.7011	2.1609	62.8576
22	0.80	0.20	4.5537	2.1339	1.6332	33.0638

Çizelge D.4 : V4 veri seti ile 24 saatlik en iyi LSTM model performansları.

Saat	Katman	Gizli	Öğrenme	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE
0	2	100	0.001	0.0	1	0.0190	0.1379	0.1107	65535
1	2	200	0.005	0.2	0	0.0202	0.1420	0.0964	65535
2	2	100	0.001	0.2	1	0.0163	0.1278	0.1002	65535
3	1	100	0.005	0.0	1	0.0046	0.0676	0.0454	65535
4	2	200	0.005	0.0	1	0.0063	0.0793	0.0673	65535
5	2	100	0.005	0.0	1	0.0047	0.0688	0.0599	65535
6	1	100	0.005	0.0	1	0.0032	0.0565	0.0488	65535
7	1	200	0.005	0.0	0	0.0132	0.1149	0.1041	65535
8	2	100	0.005	0.2	0	0.0557	0.2360	0.1930	65535
9	2	100	0.001	0.0	1	0.0965	0.3106	0.2461	65535
10	2	200	0.005	0.2	1	0.0329	0.1815	0.1533	65535
11	2	200	0.005	0.0	1	0.0475	0.2179	0.1735	65535
12	2	100	0.001	0.2	1	0.0464	0.2154	0.1730	65535
13	1	200	0.005	0.2	0	0.1374	0.3707	0.2851	7.2205
14	2	100	0.001	0.0	1	0.0984	0.3137	0.2648	8.4231
15	2	200	0.001	0.2	0	0.1020	0.3194	0.2312	65535
16	1	100	0.005	0.0	0	0.2864	0.5351	0.4445	10.5316

Çizelge D.4 (devam) : V4 veri seti ile 24 saatlik en iyi LSTM model performansları.

Saat	Katman	Gizli	Öğrenme	Düşürme Oranı	BiLSTM	MSE	RMSE	MAE	MAPE
17	1	100	0.001	0.0	0	0.2478	0.4978	0.4041	65535
18	1	100	0.005	0.2	0	0.1209	0.3478	0.2690	5.3138
19	2	100	0.001	0.2	1	0.1130	0.3362	0.2573	65535
20	2	200	0.005	0.0	1	0.0907	0.3011	0.2525	5.5616
21	2	100	0.001	0.2	1	0.0889	0.2982	0.2692	9.3795
22	1	200	0.001	0.0	0	0.1106	0.3326	0.2748	7.2219
23	2	200	0.001	0.0	1	0.0629	0.2508	0.2107	65535

Çizelge D.5 : V4 veri seti ile 24 saatlik en iyi Random Forest model performansları.

Saat	Ağaç	Derinlik	Yaprak (Min)	MSE	RMSE	MAE	MAPE
0	100	5	1	2.0583	1.4347	1.2090	65535
1	100	3	5	1.3175	1.1478	0.8461	65535
2	50	3	1	0.4044	0.6359	0.5293	65535
3	100	5	5	0.2584	0.5083	0.4209	65535
4	100	5	1	0.3401	0.5831	0.3228	65535
5	100	5	5	0.0457	0.2137	0.1065	65535
6	50	3	5	0.2330	0.4827	0.3523	65535
7	50	5	5	0.4227	0.6501	0.5647	65535
8	50	3	1	3.6121	1.9005	1.5555	65535
9	100	3	5	2.8644	1.6924	1.4039	65535
10	100	3	1	3.1911	1.7864	1.3514	65535
11	50	3	5	6.5289	2.5552	2.0124	65535
12	100	5	1	8.4784	2.9118	2.1232	65535
13	100	3	5	4.3831	2.0936	1.7486	41.8649
14	50	5	1	12.6526	3.5571	2.6588	48.7897
15	50	5	1	13.8907	3.7270	2.7875	65535
16	100	5	1	6.3390	2.5177	2.1551	49.2478
17	50	5	5	11.4307	3.3809	2.7258	65535

Çizelge D.5 (devam): V4 veri seti ile 24 saatlik en iyi Random Forest model performansları.

Saat	Ağaç	Derinlik	Yaprak (Min)	MSE	RMSE	MAE	MAPE
18	50	5	1	12.4935	3.5346	2.7501	67.9271
19	50	3	1	8.3729	2.8936	2.4454	65535
20	50	3	1	3.1801	1.7833	1.4269	29.5305
21	100	3	1	6.1921	2.4884	2.2135	83.6237
22	50	3	1	3.9670	1.9917	1.6050	33.1810
23	50	3	5	6.1239	2.4747	2.0650	65535

Çizelge D.6 : V4 veri seti ile 24 saatlik en iyi SVR model performansları.

Saat	Kutu (C)	Hata Payı	Çekirdek	Ölçek	MSE	RMSE	MAE	MAPE
0	1	0.01	Linear	Auto	1.9741	1.4050	1.1096	65535
1	1	0.01	Gaussian	1	1.2720	1.1278	0.8030	65535
2	1	0.10	Gaussian	1	0.4586	0.6772	0.5189	65535
3	1	0.10	Gaussian	Auto	0.2530	0.5030	0.3679	65535
4	0.1	0.10	Linear	Auto	0.3338	0.5778	0.3381	65535
5	0.1	0.01	Linear	Auto	0.0468	0.2163	0.0567	65535
6	1	0.10	Gaussian	Auto	0.2324	0.4820	0.3293	65535
7	1	0.10	Gaussian	1	0.4455	0.6675	0.5593	65535
8	0.1	0.01	Linear	Auto	4.3524	2.0863	1.6184	65535
9	1	0.10	Gaussian	Auto	3.1247	1.7677	1.3729	65535
10	1	0.01	Linear	1	3.0829	1.7558	1.4088	65535
11	1	0.10	Linear	1	4.9676	2.2288	1.7810	65535
12	1	0.01	Linear	1	7.0184	2.6492	1.8063	65535
13	1	0.10	Linear	1	5.0109	2.2385	1.7940	39.5425
14	1	0.01	Linear	Auto	13.4824	3.6718	2.8905	59.7036
15	1	0.10	Gaussian	Auto	14.2635	3.7767	2.6962	65535
16	1	0.01	Linear	Auto	6.8413	2.6156	2.3187	53.9180
17	1	0.10	Linear	Auto	13.1851	3.6311	2.9936	65535
18	1	0.01	Linear	Auto	13.4939	3.6734	2.7391	71.3197

Çizelge D.6 (devam) : V4 veri seti ile 24 saatlik en iyi SVR model performansları.

Saat	Kutu (C)	Hata Payı	Çekirdek	Ölçek	MSE	RMSE	MAE	MAPE
19	0.1	0.01	Linear	Auto	8.6198	2.9359	2.4076	65535
20	1	0.10	Linear	Auto	4.2078	2.0513	1.8414	40.9568
21	0.1	0.01	Linear	Auto	6.1894	2.4878	2.0660	65.0600
22	0.1	0.01	Linear	1	5.3563	2.3144	1.9124	40.6558
23	1	0.01	Linear	Auto	6.8065	2.6089	2.2869	65535

Çizelge D.7 : V4 veri seti ile 24 saatlik en iyi LightGBM model performansları.

Saat	Yaprak Sayısı	Öğrenme Oranı	Tahmini Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
0	15	0.01	100	2.0805	1.7263	2.46518E+16	-1.5249	142.1099	-1.6168
1	15	0.01	100	1.3288	0.9545	4.65628E+16	-0.4262	163.2250	-0.7477
2	15	0.01	100	0.6786	0.4668	6.59972E+16	-0.0802	183.1929	-0.1737
3	15	0.01	100	0.5209	0.3776	3.72394E+16	-0.2210	186.2335	-0.2122
4	15	0.01	100	0.6006	0.3017	1.1147E+16	-0.2050	196.9913	-0.2522
5	15	0.01	200	0.2097	0.0777	1.48351E+16	0.0306	198.9353	-0.0119
6	15	0.01	100	0.5131	0.3034	1.34118E+16	-0.2899	195.3531	-0.2438
7	15	0.01	100	0.6909	0.5156	2.85218E+16	-0.3851	176.6385	-0.3889
8	15	0.01	100	2.8683	2.2957	1.68486E+16	-1.5057	142.2033	-2.2209
9	15	0.01	100	3.2107	2.6202	3.6461E+16	-1.3876	115.5835	-2.4583
10	15	0.01	100	2.8725	2.5244	2.32089E+16	-2.3817	105.9877	-2.4135
11	15	0.01	100	3.4153	2.8410	2.25009E+16	-1.5901	103.8853	-2.7164
12	15	0.01	100	3.9716	3.1368	3.2751E+16	-1.2822	95.8946	-2.9913
13	15	0.01	100	3.5226	2.8643	51.9542	-1.9515	75.8790	-2.8643
14	15	0.01	100	4.5467	3.4760	56.1670	-1.2871	83.1603	-3.4108
15	15	0.01	100	4.3560	3.2490	4.60901E+16	-0.6960	82.0498	-2.8032
16	15	0.01	100	4.3546	3.6641	56.4780	-2.2871	85.2716	-3.6641
17	15	0.01	100	5.3240	4.4223	3.28822E+16	-1.8179	112.2291	-4.2762
18	15	0.01	100	5.5590	4.8265	73.3916	-2.8628	120.7561	-4.7857

Çizelge D.7 (devam) : V4 veri seti ile 24 saatlik en iyi LightGBM model performansları.

Saat	Yaprak Sayısı	Öğrenme Oranı	Tahminci Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
19	15	0.01	100	4.3938	3.3353	5.22216E+16	-0.8961	105.3179	-3.0206
20	15	0.01	100	4.4674	4.1984	75.5119	-7.5618	123.0528	-4.1984
21	15	0.01	100	3.9215	3.1500	62.6523	-1.8187	101.5728	-3.1500
22	15	0.01	100	4.2584	3.8080	77.1537	-3.9919	127.5811	-3.8080
23	15	0.01	100	3.7627	3.0286	2.62377E+16	-1.4935	132.6934	-2.9121

Çizelge D.8 : V4 veri seti ile 24 saatlik en iyi XGBoost model performansları.

Saat	Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
0	100	0.01	5	0.8	0.8	1.9617	1.6131	2.7155E+16	-1.2447	127.4693	-1.4925
1	100	0.01	5	1.0	0.8	1.3182	0.9549	5.3637E+16	-0.4034	159.3786	-0.7167
2	100	0.01	5	1.0	0.8	0.6980	0.4604	5.5035E+16	-0.1427	187.6062	-0.2160
3	100	0.01	3	0.8	1.0	0.5296	0.3662	2.8968E+16	-0.2623	188.6027	-0.2376
4	100	0.01	3	0.8	0.8	0.6013	0.3030	1.1592E+16	-0.2081	197.1102	-0.2515
5	100	0.01	5	1.0	0.8	0.2097	0.0815	1.6576E+16	0.0303	198.9020	-0.0079
6	100	0.01	3	1.0	0.8	0.5028	0.3075	1.8101E+16	-0.2387	193.1136	-0.2271
7	100	0.01	5	1.0	0.8	0.6801	0.5162	3.1520E+16	-0.3421	175.4841	-0.3762
8	100	0.01	3	1.0	0.8	2.7750	2.1947	2.1690E+16	-1.3452	129.2466	-2.0983
9	100	0.01	3	1.0	0.8	2.9962	2.4200	3.7833E+16	-1.0793	102.9041	-2.2519
10	100	0.01	3	0.8	0.8	2.8823	2.5405	2.2223E+16	-2.4050	107.4768	-2.4272
11	100	0.01	3	0.8	0.8	3.4419	2.8940	2.6641E+16	-1.6306	107.3654	-2.7135
12	100	0.01	3	0.8	0.8	4.0583	3.2592	3.2527E+16	-1.3830	102.9389	-3.1148
13	100	0.01	3	1.0	0.8	3.4268	2.7629	49.8390	-1.7932	71.9552	-2.7152
14	100	0.01	5	1.0	0.8	4.5180	3.4413	55.8872	-1.2583	82.0930	-3.3157
15	100	0.01	5	1.0	0.8	4.2696	3.1998	4.4066E+16	-0.6294	81.1624	-2.7753
16	100	0.01	3	1.0	0.8	4.0440	3.3738	51.8675	-1.8349	75.5546	-3.3738

Çizelge D.8 (devam) : V4 veri seti ile 24 saatlik en iyi XGBoost model performansları.

Saat	Tahminci Sayısı	Öğrenme Oranı	Maks. Derinlik	Alt Örnek Oranı	Sütun Oranı	RMSE	MAE	MAPE	R2	SMAPE	MBE
17	100	0.01	5	0.8	0.8	5.1443	4.2323	3.5246E+16	-1.6309	104.1371	-4.0758
18	100	0.01	5	1.0	0.8	5.3275	4.5845	70.6366	-2.5478	110.5516	-4.5054
19	100	0.01	5	0.8	0.8	4.3069	3.2544	5.5370E+16	-0.8219	100.9666	-2.9025
20	100	0.01	5	0.8	0.8	4.2195	3.9569	71.0809	-6.6379	111.9485	-3.9569
21	100	0.01	5	1.0	0.8	3.6999	2.9180	58.0576	-1.5091	88.6451	-2.8613
22	100	0.01	3	1.0	0.8	4.0629	3.5777	71.2423	-3.5441	113.4734	-3.5777
23	100	0.01	5	0.8	0.8	3.6231	2.8853	3.3973E+16	-1.3119	119.3653	-2.7344

Çizelge D.9 : V4 veri seti ile 24 saatlik en iyi Prophet model performansları.

Saat	Değişim Nokt. Ölç.	Mevsimsellik	RMSE	MAE	MAPE	R2	SMAPE	MBE
0	0.1	Additive	1.3630	1.1547	1.4441E+17	-0.0837	65.3362	0.5063
1	0.01	Additive	1.1420	0.9171	2.4774E+17	-0.0533	96.4590	0.3653
2	0.1	Multiplicative	0.7866	0.4743	3.8040E+16	-0.4516	198.5152	-0.4431
3	0.01	Additive	0.5409	0.4757	9.5909E+16	-0.3167	177.4987	-0.0498
4	0.1	Additive	0.5635	0.3736	5.7101E+16	-0.0610	188.1912	-0.1200
5	0.1	Additive	0.2203	0.1373	4.4897E+16	-0.0701	196.6805	0.0621
6	0.01	Additive	0.4450	0.3914	8.6814E+16	0.0297	173.6751	-0.0059
7	0.1	Additive	0.5430	0.4987	1.2802E+17	0.1446	130.0537	0.0743
8	0.01	Additive	1.8749	1.5861	1.0587E+17	-0.0706	68.1767	0.0931
9	0.01	Multiplicative	2.2411	1.7756	8.0303E+16	-0.1633	57.4600	1.4859
10	0.01	Multiplicative	1.6487	1.2742	9.0366E+16	-0.1140	39.7154	0.6527
11	0.01	Additive	2.2537	1.8403	6.4709E+16	-0.1278	55.4512	-0.8132
12	0.01	Multiplicative	2.5421	1.9952	1.1145E+17	0.0650	47.6776	-0.2727
13	0.1	Additive	2.3698	1.8862	59.8471	-0.3358	39.9676	0.7867
14	0.1	Multiplicative	3.0957	2.5227	65.1233	-0.0603	52.8816	-1.0348

Çizelge D.9 (devam) : V4 veri seti ile 24 saatlik en iyi Prophet model performansları.

Saat	Değişim Nokt. Ölç.	Mevsimsellik	RMSE	MAE	MAPE	R2	SMAPE	MBE
15	0.01	Multiplicative	2.6889	2.0015	7.8957E+16	0.3538	48.8460	-0.4247
16	0.01	Additive	2.3522	1.9666	58.4261	0.0409	38.3074	1.1065
17	0.1	Additive	3.4779	2.9218	1.6257E+17	-0.2025	50.9360	1.2887
18	0.01	Additive	2.9534	2.2927	72.3376	-0.0903	41.8588	-0.2163
19	0.1	Additive	4.0015	3.3665	2.5998E+17	-0.5727	79.1603	1.9813
20	0.1	Additive	1.8726	1.4568	37.7592	-0.5043	27.3636	1.0771
21	0.01	Additive	2.8172	2.4040	117.1750	-0.4547	58.1761	1.1962
22	0.01	Additive	2.1159	1.8356	45.0768	-0.2324	39.3103	-0.0749
23	0.1	Additive	2.3691	2.1325	1.5240E+17	0.0115	70.8949	0.3745

Çizelge D.10 : V4 veri seti ile 24 saatlik en iyi GBR model performansları.

Saat	Tahminci Sayısı	Maks. Derinlik	Öğrenme Oranı	Min. Yaprak Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
0	100	3	0.05	1	2.3481	1.9681	1.0663E+15	-2.2163	193.7669	-1.9638
1	100	3	0.05	1	1.4798	1.0226	1.0676E+16	-0.7688	197.4018	-0.9837
2	100	3	0.1	5	0.7501	0.4103	1.4560E+16	-0.3198	197.1730	-0.3702
3	100	3	0.05	5	0.5642	0.3632	1.9418E+16	-0.4324	194.4379	-0.2994
4	100	3	0.05	5	0.6103	0.2927	4.3854E+15	-0.2442	199.0214	-0.2756
5	100	3	0.05	1	0.2136	0.0700	1.0765E+16	-0.0059	199.3950	-0.0384
6	200	3	0.1	1	0.5319	0.2950	4.8926E+15	-0.3863	199.1727	-0.2850
7	100	3	0.05	1	0.7834	0.5439	9.1795E+15	-0.7805	196.3561	-0.5172
8	100	3	0.05	5	3.0866	2.5055	2.0035E+15	-1.9016	178.2996	-2.4966
9	100	3	0.05	5	3.6992	3.1250	9.6E+15	-2.1694	172.6886	-3.0823
10	100	3	0.05	1	3.6917	3.2862	1.0278E+15	-4.5858	172.1209	-3.2862
11	200	3	0.1	5	3.8251	3.3476	1.4085E+16	-2.2489	149.1068	-3.1598
12	200	5	0.05	1	4.5192	3.8203	1.2360E+14	-1.9549	159.5044	-3.8203
13	100	3	0.05	5	4.2220	3.6390	72.2852	-3.2400	120.8518	-3.5596

Çizelge D.10 (devam): V4 veri seti ile 24 saatlik en iyi GBR model performansları.

Saat	Tahminci Sayısı	Maks. Derinlik	Öğrenme Oranı	Min. Yaprak Sayısı	RMSE	MAE	MAPE	R2	SMAPE	MBE
14	100	5	0.05	5	5.2749	4.0935	67.7615	-2.0784	116.2133	-4.0117
15	100	3	0.05	1	4.8159	3.7026	3.3136E+16	-1.0730	104.9984	-3.4106
16	100	3	0.05	1	4.7902	4.3681	78.9502	-2.9777	132.8195	-4.3681
17	100	5	0.05	5	6.1351	5.2901	2.3922E+15	-2.7418	164.0526	-5.2901
18	100	3	0.05	5	6.2623	5.5482	90.3761	-3.9021	167.0216	-5.5482
19	100	3	0.05	1	5.2677	4.1931	3.4446E+15	-1.7255	187.5359	-4.1835
20	100	3	0.05	5	5.3729	5.1672	96.0906	-11.3839	184.7406	-5.1672
21	100	3	0.05	5	4.5040	3.8479	89.2077	-2.7182	163.2586	-3.8479
22	100	3	0.05	5	5.0087	4.5951	96.1790	-5.9059	185.1939	-4.5951
23	100	3	0.05	5	4.2015	3.4825	3.2310E+15	-2.1090	194.1693	-3.4681

ÖZGEÇMİŞ

Ad-Soyad : Hatice Bilge ALGIN

ÖĞRENİM DURUMU:

- **Lisans** : 2005, İnönü Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği
- **Yüksek Lisans** : ---, İnönü Üniversitesi, Mühendislik Fakültesi, Elektrik Elektronik Mühendisliği

MESLEKİ DENEYİM:

- Bilkomp Elektrik Enerji ve Kompanzasyon, Kurucu – Yönetici (2007-2020)
- T.C. Enerji ve Tabii Kaynaklar Bakanlığı – TEİAŞ – ELTEMTEK, Kıdemli İç Denetçi (20.03.2020 – 20.06.2023)
- Full Elektrikli Araçlar Şarj Ağı İstasyonları İşletmeciliği Enerji Tic. ve San. A.Ş., Kurucu Genel Müdür (01.08.2023 - Halen)
- Acarlar Grup Şirketleri, Genel Müdür (15.10.2024 - Halen)

YÜKSEK LİSANS VEYA DOKTORA TEZİNDEN TÜRETİLEN ÇALIŞMALAR (Makaleler, Bildiriler, Patentler v.b.)

- **Algin, H. B. & Köseoğlu, M. (2025, Temmuz).** Coğrafi ve Özellik Ağırlıklı Yaklaşımlarla EV Şarj Talebi Tahmini: Karşılaştırmalı Bir İnceleme. 5. BİLSEL International Kibyra Scientific Researches Congress, Burdur, Türkiye, ss. 8–12. (Bildiri)
- **Algin, H. B., Dağcı B. & Köseoğlu, M. (2025, Kasım).** Forecasting Electric Vehicle Charging Demand Using Time Series and Machine Learning Methods. 9. International Antalya Scientific Research and Innovative Studies Congress, Antalya, Türkiye