



**T.C.
İSTANBUL ÜNİVERSİTESİ-CERRAHPAŞA
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



[DOKTORA TEZİ]

**[Mobil Kablosuz Ağların Optimizasyonunda Yapay Zeka
Yöntemlerinin Kullanılarak Operasyonel Verimliliğin Artırılması]**

[Yakup Tarık KRANDA]

**[DANIŞMAN
Doç. Dr. Rüya ŞAMLI]**

[Bilgisayar Mühendisliği Anabilim Dalı]

[Bilgisayar Mühendisliği Programı]

[İSTANBUL-2021]

Bu çalışma, [29.07.2021] tarihinde aşağıdaki jüri tarafından [Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Mühendisliği Programı]nda [Doktora tezi] olarak kabul edilmiştir.

Tez Jürisi

[Doç. Dr.] [Rüya ŞAMLI] (Danışman)
[İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi]

[Prof. Dr.] [Abdül Halim ZAIM]
[İstanbul Ticaret Üniversitesi
Mühendislik Fakültesi]

[Prof. Dr.] [Sabri ARIK]
[İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi]

[Prof. Dr.] [Ahmet SERTBAŞ]
[İstanbul Üniversitesi-Cerrahpaşa
Mühendislik Fakültesi]

[Prof. Dr.] [İlhami YAVUZ]
[Maltepe Üniversitesi
Mühendislik ve Doğa Bilimleri Fakültesi]

ÖNSÖZ

Çalışmalarım süresince sabırla bana destek olan canım eşim Tuba'ya, son satırlarımı tamamlamadan hemen önce dünyaya gözlerini açarak bana eşsiz bir enerji aşılayan canım oğlum Arsen Toprak'a, hayatım boyunca bana destek olan ebeveynlerim ve kardeşim Selma, Şenol ve Selen Kranda'ya sonsuz şükranlarımı iletiyorum.

Doktora çalışmalarına odaklanmamı sağlayan, çalışmalarımı organize eden ve bilgilendirici, yönlendirici, motive edici desteğiyle her zaman yanımda olan kıymetli danışmanım Doç. Dr. Rüya Şamlı'ya minnettarım.

Kıymetli vaktini ayırarak, tavsiye ve yönlendirmeleriyle tez izleme jürimde bana destek olan değerli hocalarım Prof. Dr. Abdül Halim Zaim ve Prof. Dr. Sabri Arık'a teşekkürü bir borç bilirim.

Ayrıca, hayatımın her alanında elini omzumda hissettiğim, birlikte büyürken koşulsuzca bana motivasyon kaynağı olan en büyük destekçim, canım kardeşim Engin Tamer'e de sonsuz teşekkürlerimi iletiyorum. |

|Yakup Tarık KRANDA|

Temmuz 2021

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	v
İÇİNDEKİLER.....	vi
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ.....	x
SİMGE VE KISALTMA LİSTESİ.....	xi
ÖZET	xvi
SUMMARY	xviii
1. GİRİŞ	1
2. GENEL KISIMLAR.....	6
2.1. MEVSİMSELLİK PROBLEMİ TANIMI	6
2.2. TAHMİNLEME PROBLEMİ TANIMI	15
2.3. MOBİL AĞ TRAFİK TAHMİNLEMESİ LİTERATÜR TARAMASI	17
3. MALZEME VE YÖNTEM.....	31
3.1. VERİ SETLERİ	31
3.1.1. Mevsimsellik Problemi Veri Seti	31
3.1.2. Tahminleme Problemi Veri Seti	33
3.2. YÖNTEM VE KÜTÜPHANELER	35
3.2.1. Tekrarlayan Yapay Sinir Ağları	35
3.2.2. Hiyerarşik Kümeleme Algoritması.....	37
3.2.3. SARIMAX Tahminleme Algoritması	39
3.2.4. STL-ETS Tahminleme Algoritması	40
3.2.5. LSTM Bazlı Tahminleme Algoritmaları	41
3.3. ARAÇLAR.....	43
3.3.1. Visual Studio Code Entegre Geliştirme Ortamı	43
3.3.2. Python Programlama Dili ve Ortamı	44
3.3.3. PostgreSQL Veritabanı Yönetim Sistemi.....	48
3.3.4. ClickHouse Veritabanı Yönetim Sistemi	49
4. BULGULAR.....	51
4.1. MEVSİMSELLİK PROBLEMİ BULGULARI.....	51
4.2. TAHMİNLEME PROBLEMİ BULGULARI.....	79

4.2.1. Kümeleme Algoritması ve Kullanılan Nitelikler	79
4.2.2. Uygulanan LSTM Model Mimarileri	83
4.2.3. Çözümün İş Akışı	86
4.2.4. Deneysel Çıktılar	88
5. TARTIŞMA VE SONUÇ	94
KAYNAKLAR.....	98
ÖZGEÇMİŞ	106

|



ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1. 2025 Global Kullanıcı Başına Mobil Veri Kullanım Dağılımı (GSMA, 2020).....	2
Şekil 2.1. 2025 İtibariyle Mobil Internet Abonelerinin Nüfus Yüzdesi	6
Şekil 2.2. MNO'ların Ağ Dönüşüm Stratejisini Yönlendiren Birincil Hedef	7
Şekil 2.3. Mobil Ağ Kapasite Planlama Yaşam Döngüsü	8
Şekil 2.4. CUSUM Logaritmik Olabilirlik Oranı Davranışı, Sabit Varyanslı Bir Gauss Dizisinin Ortalamasındaki Değişim Örneği	12
Şekil 3.1. Mevsimsellik Etiketli Test Baz İstasyonlarının Dağılımı	32
Şekil 3.2. Eğitim, Test ve Geriye Dönük Veri Zaman Aralıkları	34
Şekil 3.3. Hiyerarşik Kümeleme için Bir Dendrogram Örneği	38
Şekil 3.4. Standart Uzun Ömürlü Kısa Bellek Sinir Ağı Mimarisi	42
Şekil 4.1. Algoritmanın Ana İş Akışı	55
Şekil 4.2. Baz İstasyonu Mevsimsel Güç Endeksi (SPI) Parametreleri	61
Şekil 4.3. Çarpımsal Mevsimselliğe Sahip Bir Baz İstasyonu Örneği	62
Şekil 4.4. Çarpımsal Mevsimsel Esnek Modelin Oransal Değişim Dağılımı	63
Şekil 4.5. Çarpımsal Mevsimsel Esnek Modelin Değişim Noktası Konumları	63
Şekil 4.6. Çarpımsal Mevsimsel Davranışla İlgili Daha Az Esnek Modelin Bileşenleri	64
Şekil 4.7. Çarpımsal Mevsimselliğin Günlüğe Dönüştürülmüş Trafik Verileri Üzerinde Daha Az Esnek Modeldeki Oransal Değişim Dağılımı.....	65
Şekil 4.8. Doygunluk Eğilimi Bileşeniyle Çarpımsal Mevsimsellik İçin Ayırıştırılmış Bileşenler.....	66
Şekil 4.9. Seviye Kayması Nedeniyle Değişim Noktası Oluşmuş Baz İstasyonu Örneği	67
Şekil 4.10. Çarpımsal Mevsimsel Davranışta Daha Az Esnek Modelde Değişim Noktası Konumları.....	67
Şekil 4.11. Eğilim Değişiminin Mevsimsel Bileşene Adreslenmesiyle STL Ayırıştırmasının Hatalı Hale Gelmesi	68

Şekil 4.12. Değişim Noktası Davranışında Daha Esnek Modelin Bulduğu Değişim Noktası Konumları	68
Şekil 4.13. Seviye Kayması Nedeniyle Değişim Noktası Olan İstasyonun Ayırıştırılmış Bileşenleri.....	69
Şekil 4.14. Üstel Artan Eğilim Bileşeni Nedeniyle Değişim Noktası İçeren Baz İstasyonu Örneği.....	70
Şekil 4.15. Parça Bazında Uydurma İle Daha Esnek Modelin Değişim Noktası Konumları ve Üstel Uydurulmuş Eğilim Bileşeni.....	71
Şekil 4.16. Üstel Eğilim Nedeniyle Değişim Noktasına Sahip Baz İstasyonu İçin Ayırıştırılmış Seri Bileşenleri.....	71
Şekil 4.17. Mevsimsel Olmayan Özelliklere Sahip Baz İstasyonu Örneği	72
Şekil 4.18. Mevsimsel Olmayan Özelliklere Sahip Bir Baz İstasyonu İçin Ayırıştırılmış Bileşenler.....	73
Şekil 4.19. Test Seti SPI Değerleri Çıktısı Dağılımı	76
Şekil 4.20. Test Seti Mevsimsel Dönem Uzunluğunun Dağılımı	76
Şekil 4.21. Tüm Veri Setinin Etiketli Baz İstasyonlarının Dağılımı	78
Şekil 4.22. Tüm Veri Seti SPI Değerleri Çıktısının Dağılımı	79
Şekil 4.23. Tüm Veri Seti Mevsimsel Dönem Uzunluğunun Dağılımı.....	79
Şekil 4.24. Eş Lokasyonlu Olmayan Toplam Komşu Sayısı Dağılımı	80
Şekil 4.25. Kümeleme Öncesi Niteliklerin Değer Dağılımlarına Dair Bir Örnek.....	82
Şekil 4.26. Göreceli Azimut Değeri için Örnekler	82
Şekil 4.27. Dördüncü Seviyeden Kesilerek Oluşturulmuş 5 Kümeli Bir Dendrogram Örneği.....	83
Şekil 4.28. Hiperparametre İyileştirme Sonuçları	84
Şekil 4.29. Kullanılan LSTM Mimarileri	85
Şekil 4.30. SARIMAX Yönteminin Limitli Veri Üzerinde Tahminleme Hataları	91
Şekil 4.31. SARIMAX Yönteminin Sentetik Veri Üzerinde Tahminleme Hataları	91
Şekil 4.32. STL-ETS Yönteminin Limitli Veri Üzerinde Tahminleme Hataları	92
Şekil 4.33. STL-ETS Yönteminin Sentetik Veri Üzerinde Tahminleme Hataları	92
Şekil 4.34. Farklı LSTM Mimarilerinin Sentetik Veri Üzerinde Tahminleme Hataları	93

TABLO LİSTESİ

Sayfa No

Tablo 2.1. Ağ T trafiğinin Tahminlemesinde Kullanılan Yapay Zeka Yöntemleri Konulu Literatür Çalışmaları.....	27
Tablo 3.1. Mevsimsellik Veri Seti İstatistikleri.....	31
Tablo 3.2. Tahminleme Veri Seti İstatistikleri	33
Tablo 4.1. Karmaşıklık Matrisi Değerleri	75
Tablo 4.2. Metrik Formülleri.....	75
Tablo 4.3. Karmaşıklık Matrisi	77
Tablo 4.4. Doğruluk Sonuçları ve Karşılaştırma.....	77
Tablo 4.5. Yetersiz ve Sentetik Geçmişin Tahminleme Doğruluk Sonuçları	90

SİMGE VE KISALTMA LİSTESİ

Kısaltmalar	Açıklama
2G	:Second-Generation Cellular Network – İkinci Nesil Kablosuz Telefon Teknolojisi
3G	:Third-Generation Cellular Network – Üçüncü Nesil Kablosuz Telefon Teknolojisi
AARIMA	:Adjusted ARIMA – Düzeltilmiş ARIMA
ACE	:Absolute Cumulative Error – Mutlak Kümülatif Hata
ACF	:Auto-Correlation Function – Otomatik Korelasyon Fonksiyonu
ADNN	:All Direct Neighbor Link – Tüm Doğrudan Komşu Bağlantısı
AIC	:Akaike Information Criterion – Akaike Bilgi Kriteri
ALZ	:Active LeZi – Aktif LeZi
ANN	:Artificial Neural Networks – Yapay Sinir Ağları
APE	:Absolute Percentage Error – Mutlak Yüzde Hatası
API	:Application Programming Interface – Uygulama Programlama Arayüzü
AR	:Autoregressive – Otoresif
ARFIMA	:Autoregressive Fractionally Integrated Moving Average – Otoresif Kesirli Entegre Hareketli Ortalama
ARIMA	:Autoregressive Integrated Moving Average – Otoresif Entegre Hareketli Ortalama
ARIMAX	:Autoregressive Integrated Moving Average with External Variables – Dış Değişkenlerle Otoresif Entegre Hareketli Ortalama
ARMSE	:Average RMSE – Ortalama RMSE
ARMA	:Autoregressive Moving Average – Otoresif Hareketli Ortalama
BBU	:Base-Band Unit – Baz-Bant Birimi
BIC	:Bayesian Information Criterion – Bayes Bilgi Kriteri
BLRNN	:BiLinear Recurrent Neural Networks – Çift Doğrusal Tekrarlayan Sinir Ağları
BPNN	:Back-Propagation Neural Network – Geri-Yayılmalı Sinir Ağları
BPWNN	:Back-Propagation Wavelet Neural Network – Geri-Yayılmalı Dalgacık Sinir Ağları

BS	:Base Station – Baz İstasyonu
BSC	:Base Station Controller – Baz İstasyonu Denetleyicisi
CE	:Cumulative Error – Kümülatif Hata
CN	:Core Network – Çekirdek Ağı
CNN	:Convolutional Neural Network – Evrişimsel Sinir Ağları
CO	:Covariation Orthogonal Prediction – Kovaryasyon Ortogonal Tahmin
ConvLSTM	:Convolutional LSTM – Evrişimsel LSTM
C-RAN	:Cloud RAN – Bulut RAN
DANN	:Dijkstra-Assisted NN – Dijkstra Destekli NN
DBLRNN	:Dynamic BLRNN – Dinamik BLRNN
DBN	:Deep Belief Networks – Derin İnanç Ağları
DCRNN	:Diffusion Convolutional Recurrent Neural Network – Difüzyon Evrişimli Tekrarlayan Sinir Ağı
DSARIMA	:Double Seasonal ARIMA – Çift Mevsimsel ARIMA
DSEXP	:Double Seasonal Exponential Smoothing – Çift Mevsimsel Üstel Düzeltme
D-STN	:Deep Spatio-Temporal Networks – Derin Uzay-Zamansal Ağlar
ENT	:Entropy Theory – Entropi Teorisi
ERLS	:Equality Restricted Least Squares – Eşitlik Kısıtlı En Küçük Kareler
ESN	:Echo State Networks – Yankı Durum Ağları
ETS	:Error Trend and Seasonality – Hata Eğilimi ve Mevsimsellik
FARIMA	:Fractionally Autoregressive Integrated Moving Average – Kesirli Otoregresif Entegre Hareketli Ortalama
FCNN	:Fully Convolutional Neural Network – Tam Evrişimsel Sinir Ağları
FFNN	:Feed Forward Neural Networks – İleri Beslemeli Sinir Ağları
FFT	:Fast Fourier Transforming – Hızlı Fourier Dönüşümü
FWA	:Fixed Wireless Network– Sabit Kablosuz Ağ
gaNET	:Generalized Additive Networks – Genelleştirilmiş Katkı Ağları
GARCH	:Generalized Autoregressive Conditional Heteroskedasticity – Genelleştirilmiş Otoregresif Koşullu Heteroskedastisite
GNN	:Graphed Neural Networks – Grafikli Sinir Ağları
GRU	:Gated Recurrent Unit – Geçitli Tekrarlayan Birimler
HMM	:Hidden Markov Model – Gizli Markov Modeli
HW-ETS	:Holt-Winter’s Exponential Smoothing – Holt-Winter’in Üstel Yumuşatması

INGARCH	:Integer-Valued GARCH – Tamsayı Değerli GARCH
ISP	:Internet Service Provider – Internet Servis Sağlayıcısı
kELM	:Kernel Extreme Machine Learning – Çekirdek Ekstrem Makine Öğrenmesi
LRD	:Long-Range Dependence – Uzun Menzilli Bağımlılık
LSTM	:Long Short-Term Memory – Uzun Ömürlü Kısa Süreli Bellek
LTE	:Long Term Evolution – Uzun Vadeli Evrim
MA	:Moving Average – Hareketli Ortalama
MAD	:Mean Absolute Deviation – Ortalama Mutlak Sapma
MAE	:Mean Absolute Error – Ortalama Mutlak Hata
MAPE	:Mean Absolute Percentage Error – Ortalama Mutlak Yüzde Hatası
MC	:Markov Chain – Markov Zinciri
MEC	:Multi-Access Edge Computing – Çoklu Erişimli Uç Hesaplama
MFO	:Moth-Flame Optimizer – Moth-Flame Eniyileyici
MLE	:Maximum Likelihood Estimation – Maksimum Olabilirlik Kestirimi
MLP	:Multi-Layer Perceptron – Çok Katmanlı Algılayıcı
MLPNN	:Multi-Layer Perceptron Neural Network – Çok Katmanlı Algılayıcı Sinir Ağları
MLPWD	:MLP with Weighted Decay – Ağırlıklandırılmış Düşüşlü MLP
MNO	:Mobile Network Operator – Mobil Ağ Operatörü
MSE	:Mean Squared Error – Ortalama Kare Hata
MuLSTM	:Multivariable LSTM – Çok Değişkenli LSTM
MVO	:Multiverse Optimizer – Çoklu Evren Eniyileyici
NMAE	:Normalized Mean Absolute Error – Normalleştirilmiş Ortalama Mutlak Hata
NMSE	:Normalized Mean Square Error – Normalleştirilmiş Ortalama Kare Hatası
NRMSD	:Normalized Root Mean Square Error Deviation – Normalleştirilmiş Kök Ortalama Kare Hata Sapması
NRMSE	:Normalized Root Mean Square – Normalleştirilmiş Kök Ortalama Kareler
NWDAF	:Network Data Analytics Function – Ağ Veri Analitiği İşlevleri
OLS	:Ordinary Least Square – Sıradan En Küçük Kareler
OSPF	:Open Shortest Path First – Önce En Kısa Yolu Aç
PASO	:Particle Swarm Optimizer – Parçacık Sürüsü Eniyileyici
PCA	:Principle Component Analysis – Temel Bileşenler Analizi
PPM	:Prediction by Partial Matching – Kısmi Eşleştirme ile Tahmin

PRB	:Physical Resource Block – Fiziksel Kaynak Bloğu
PRNN	:Primitives with Recurrent Neural Networks – Tekrarlayan Sinir Ağlarına Sahip İlkeller
RAN	:Radio Access Network – Radyo Erişim Ağı
RE	:Relative Error – Göreceli Hata
RF	:Random Forest – Rastgele Orman
RMSE	:Root Mean Square Error – Kök Ortalama Kare Hata
RNC	:Radio Network Controller – Radyo Ağ Denetleyicisi
RNN	:Recurrent Neural Networks – Tekrarlayan Sinir Ağları
RRBFN	:Recurrent Radial Basis Neural Networks – Tekrarlayan Radyal Temel Sinir Ağları
RRH	:Remote Radio Head – Uzaktan Radyo Başlığı
RRMSE	:Relative Root Mean Squared Error – Göreceli Kök Ortalama Kare Hata
RTO	:Recovery Time Objective – Kurtarma Süresi Hedefi
SAE	:Stacked Auto-Encoders – Yığılmış Otomatik Kodlayıcılar
SARIMA	:Seasonal ARIMA – Mevsimsel ARIMA
SLNN	:Single-Link NN – Tekil Bağlantılı Sinir Ağı
SMAPE	:Symmetric Mean Absolute Percentage Error – Simetrik Ortalama Mutlak Yüzde Hatası
SME	:Subject Matter Expert – Konu Uzmanı
SPI	:Seasonal Power Index – Mevsimsel Güç Endeksi
SPM	:Sampled Pattern Matching – Örneklenmiş Desen Eşleştirme
SSE	:Sum of Square Error – Hata Karelerinin Toplamı
SVM	:Support Vector Machines – Destek Vektör Makineleri
STL	:Seasonal and Trend Decomposition by Using Loess – Loess Kullanarak Mevsimsellik ve Eğilim Ayırıştırma
ST-MVL	:Spatio-Temporal Multi-View-based Learning – Uzak-Zamansal Çoklu Görünüm Tabanlı Öğrenme
SVM	:Support Vector Machine – Destek Vektör Makinesi
TCP-IP	:Transmission Control Protocol - Internet Protocol – İletim Kontrol Protokolü - İnternet Protokolü
TSD	:Time Series Decomposition – Zaman Serisi Ayırıştırma
TN	:Transmission Network – İletim Ağı

VAR	:Vector Auto Regression – Vektör Otomatik Bağlanım
VPN	:Virtual Private Network – Sanal Özel Ağ
vSVR	:v-Support Vector Regressor – vDestekli Vektör Bağlanım
WANN	:Windowed Artificial Neural Network – Pencereleli Yapay Sinir Ağı
WF	:Wavelet Filters – Dalgacık Filtreleri
WLAN	:Wireless Local Area Network – Kablosuz Yerel Alan Ağı
WNN	:Wavelet Neural Networks – Dalgacık Sinir Ağları
WT	:Wavelet Transform – Dalgacık Dönüşümü



ÖZET

[DOKTORA TEZİ]

[Mobil Kablosuz Ağların Optimizasyonunda Yapay Zeka Yöntemlerinin Kullanılarak Operasyonel Verimliliğin Artırılması]

[Yakup Tarık KRANDA]

İstanbul Üniversitesi-Cerrahpaşa

Lisansüstü Eğitim Enstitüsü

[Bilgisayar Mühendisliği Anabilim Dalı]

[Danışman : Doç. Dr. Rüya ŞAMLI]

[Mobil iletişim dünyasında meydana gelen yeni gelişmeler, değişen ihtiyaçlar, bilgiye her an, her yerden, yüksek hız ve kapasitede erişim ihtiyacını ortaya çıkarmıştır. Bu doktora çalışması kapsamında, mobil radyo ağ performans ve kapasitesinin artırılması ve buna bağlı olarak operasyon verimliliği ve maliyet kazancı odağında daha akıllı kapasite planlama yapılabilmesi için uzun dönemli zaman serisi ve yapay öğrenme odaklı tahminleme algoritmaları ve bunların performansını olumsuz etkileyen ana iki durum ele alınmıştır. Bu durumlar mobil radyo hücrelerinin değişim noktası farkında mevsimsellik tespiti ve yeni devreye alındığı için yetersiz geçmişe sahip olan mobil radyo hücrelerinin uzun gelecek trafik tahminleme hatalarının kontrol altına alınması şeklinde ifade edilebilir. Çalışma kapsamında, gerçek mobil ağ günlük trafik verileri işlenmektedir ve baz istasyonları üzerindeki uzun vadeli mevsimsellik etkisini belirlemek için yeni bir hibrit çözüm geliştirilmektedir. Bu işlemten sonra, seviye kayması veya üstel eğilim artışı nedeniyle değişim noktası varlığını dikkate alarak baz istasyonları üzerindeki mevsimsellik etkisini çıkarmak için mobil ağ baz istasyonu trafik verilerinin mevsimsellik tespiti ve ayrıştırılması için bir algoritma tasarlanmaktadır. Yerel değişim noktası arama, değişim oranı analizi ve benzerlik kontrol mekanizmaları incelenmektedir. İki yıllık geçmiş mobil trafik verisi kullanarak çarpımsal mevsimsel faktörleri hesaplanmaktadır. Ayrıştırılmış mevsimsellik etkisi ile baz istasyonlarını etiketlemek, gelecek yılın trafik talebini tahmin ederken algoritmaları yeni özellikler ile beslemeye yardımcı olur.

Buna ek olarak çalışma kapsamında, yetersiz geçmiş verilerden kaynaklanan büyük uzun vadeli trafik tahminleme hatalarına odaklanılmakta ve yetersiz geçmişe sahip yeni konuşlandırılmış hücreler için tahmin hatalarını azaltmak ve hatayı kontrol altına almak için hiyerarşik kümeleme tabanlı bir algoritma geliştirilmektedir. Buradaki amaç, hata sayısını önemli ölçüde azaltmak ve en azından bu hücrelerin ilk yılı için tatmin edici hata seviyelerinde tahmin sonuçları üretmektir. Yeni konuşlandırılan hücreler için yetersiz geçmiş verisi, hareketlilik devir ilişkileri, sahadan sahaya uzaklık, hücrelerin servis açısı kullanılarak kümelenmiş komşuların ortak geçmiş özellikleriyle tamamlanır. Hiyerarşik kümeleme tabanlı algoritmamızla seçilen hücrelerin geçmiş özellikleri, bu yeni hücreler için sentetik olarak geriye dönük bir geçmiş oluşturmak için kullanılır ve gelişmiş doğruluk sonuçlarını göstermek için farklı tahminleme algoritmalarına beslenir. SARIMAX, STL-ETS gibi klasik zaman serisi yöntemlerinin ve Vanilla LSTM Sinir Ağları, LSTM Kodlayıcı-Kod Çözücü Sinir Ağları, CNN-LSTM Kodlayıcı-Kod Çözücü Sinir Ağları, ConvLSTM Kodlayıcı-Kod Çözücü Sinir Ağları gibi derin sinir ağı yöntemlerinin tahmin çıktıları bu tez çalışmasında yer alan tamamlayıcı algoritma yardımıyla iyileştirilmektedir. Daha doğru ortalama sonuçlar için hiperparametre ayarlama adımları ve kümeleme için ideal nitelik seçimi de çoklu çalıştırma simülasyonları ile çalışma kapsamında yer almaktadır. |

Temmuz 2021, [105.] sayfa.

Anahtar kelimeler: |Mobil Ağ Kapasite Tahmini, Yetersiz Veri Sorunları, Hiyerarşik Kümeleme, Yapay Sinir Ağları, Zaman Serisi Yöntemleri, Mevsimsellik, Değişim Noktası Tespiti |

SUMMARY

[Ph.D. THESIS]

**[Increasing Operational Efficiency on Mobil Wireless Network
Optimization by Using Artificial Intelligence Methods]**

[Yakup Tarık KRANDA]

Istanbul University-Cerrahpasa

Institute of Graduate Studies

[Department of Computer Engineering]

Supervisor : [Assoc. Prof. Dr.] [Ruya SAMLI]

New developments and needs in the world of mobile communication have revealed the need for access to information at any time, from anywhere, at high speed and capacity. Within the scope of this doctoral study, long-term time series and machine learning-oriented estimation algorithms and two main situations that negatively affect their performance are discussed in order to increase mobile radio network performance and capacity, and accordingly, to make smarter capacity planning focused on operational efficiency and cost savings. These situations can be expressed as the need for seasonality detection in change point awareness and the large estimation errors of mobile cells, which have insufficient history due to newly commissioning. Within the scope of the study, real mobile radio network daily traffic data is processed, and a new hybrid solution is developed to determine the long-term seasonality effect on base stations. After this process, an algorithm is designed for seasonality detection and decomposition of mobile network base station traffic data in order to extract the seasonality effect on base stations, considering the change point existence due to level shift or exponential trend increase. Local change point search, rate of change analysis and similarity control mechanisms are examined. Using two years of historical traffic data, multiplicative seasonal factors are calculated. Tagging base stations with the disaggregated seasonality effect helps the algorithms predict new features when forecasting traffic demand for the next year. In addition, the study focuses on these large prediction errors caused by insufficient historical data, and a hierarchical clustering-based algorithm is developed to reduce traffic prediction errors and control the error

for newly deployed cells with insufficient history. The aim here is to significantly reduce the number of errors and produce prediction results with satisfactory error levels, at least for the first year of these cells.

Insufficient historical data for newly deployed cells is complemented by common historical characteristics of clustered neighbors using mobility cycle relationships, site-to-site distance, cells' service angle. Historical characteristics of cells selected by our hierarchical clustering-based algorithm are used to generate synthetic retrospective history for these new cells and fed into different prediction algorithms to demonstrate improved accuracy results. The prediction outputs of classical time series methods such as SARIMAX, STL-ETS and deep neural network methods such as Vanilla LSTM Networks, LSTM Encoder-Decoder Networks, CNN-LSTM Networks, ConvLSTM Encoder-Decoder are improved with the help of the complementary algorithm in this thesis. Hyperparameter tuning steps and clustering features significance analysis with multi-run simulations are also included for more accurate averaging results. |

July 2021, |105| pages.

Keywords: | Mobile Network Capacity Forecasting, Insufficient Data Problems, Hierarchical Clustering, Neural Networks, Time Series Methods, Seasonality, Change Point Detection |

1. GİRİŞ

Mobil iletişim dünyasında meydana gelen yeni gelişmeler, ihtiyaçlar, bilgiye her an, her yerden, yüksek hız ve kapasitede erişim ihtiyacı mobil radyo ağların yönetilmesinde yeni algoritma ve otomasyon çözümlerine ihtiyacı artırmıştır. Mobil radyo ağların yönetimi zorlaştıran başlıca gelişmeler şu şekilde sıralanabilir:

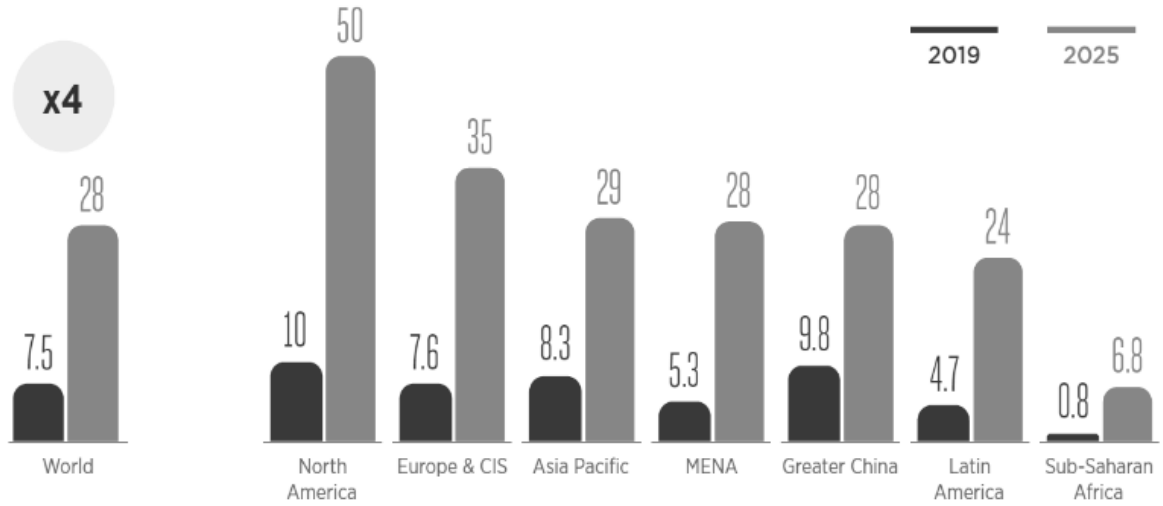
- hızla artan mobil kullanıcı sayısı,
- nesnelerin internet altyapısı üzerinden iletişimlerinin yaygınlaşması (Nesnelerin İnterneti),
- alternatif iletişim teknolojilerinin bir arada kullanımı (2G, 3G, LTE, 5G, WiFi),
- yeni kablosuz iletişim teknolojilerine yönelik çalışmalar (Sanallaştırılmış Ağlar, Programlanabilir Ağlar, 5G Yeni Radyo),
- değerlendirilmek üzere depolanan çok farklı kaynaktan (Radyo Erişim Ağları, Transmisyon Ağları, Çekirdek Ağ, Müşteri Deneyim Yönetim Çözümleri) büyük miktarda verinin anlamlandırılması ihtiyacı.

Mobil Ağ Operatörleri (Mobile Network Operator – MNO) için yeni uygulamalardan doğan çözüm stratejilerinin geliştirilebilmesi, aynı zamanda ağ optimizasyon maliyetlerinin gerek işçilik gerekse donanım tarafıyla düşürülmesini kaçınılmaz kılmaktadır. Kapsamı ve karmaşıklık seviyesi üstel artış gösteren mobil ağ çözümlerinin yönetimi zorlaşmakta, kullanıcıların servis kalitesine yönelik beklentileri artmaktadır. Mobil ağların yönetimi ve optimizasyonunda operasyon verimliliğini artırma, sorunlar oluşmadan tespit edebilme, problemleri noktaların hızla bertaraf edilmesi veya daha az enerji harcama noktalarında çeviklik ve maliyet avantajı sağlayacak yöntemlerin başında makinelerin hızlı işleme kapasitelerinden faydalanma gelmektedir.

Günümüzde internet servislerini mobil cihazlar üzerinden kullanma talebi katlanarak artmaktadır. Her gün kümülatif mobil abone sayısı artmakta ve kullanıcı başına üretilen ağ trafiği miktarı da sürekli olarak büyümektedir. Artan talebin yanı sıra, pandemi koşulları, özellikle uzaktan çalışma ve çevrimiçi iş fırsatları da mobil ağ abone davranışlarını etkilemektedir. Cisco raporuna göre, IP ağlarına bağlı cihaz sayısı 2023 yılına kadar küresel nüfusun üç katından fazla olacak ve 2018'de kişi başına ağa bağlı cihaz sayısı 2,4 adet iken,

2023'te bu sayı 3,6'ya çıkacak, 2018'de 18,4 milyar olan ağı bağlı cihaz sayısı 2023'te 1,4 milyara yükselecektir (Cisco, 2018). GSM Birliği (GSMA) İstihbarat Mobil Ekonomi Araştırma Raporuna göre, 2025 yılına kadar, mobil ağlar tarafından hizmet verilen 1,2 milyar yeni abone olacaktır (GSMA, 2020). 2025 yılına kadar abone başına küresel mobil veri kullanım dağılımı Şekil 1.1'de gösterilmektedir. Dünya çapında mobil veri kullanımının GSMA raporuna göre 2025 yılına kadar 4 kat artması beklenmektedir. Bu artan talebi, değişen abone davranışlarını ve uygun maliyetli ağ yönetimini dikkate almak için mobil ağ operatörlerinin, veri ve makine öğrenimi ile güçlendirilmiş otomatik planlama süreçlerini kullanarak daha akıllı kapasite yatırım planları yapması gerekmektedir.

GB per subscriber per month



Şekil 1.1. 2025 Global Kullanıcı Başına Mobil Veri Kullanım Dağılımı (GSMA, 2020)

Hizmet çeşitliliği, ilgili Hizmet Seviyesi Anlaşma (Service Level Agreement – SLA) gereksinimleri ve hizmet başına deneyim, mobil ağ kapasite planlama sürecinin karmaşıklığını artırmaktadır.

Doğru zamanda doğru yatırımları yapabilmek çok önemlidir. Çünkü, rekabetçi koşullar altında MNO için Toplam Sahip Olma Maliyetini (Total Cost of Ownership – TCO) en aza indirmek önemlidir. Bundan dolayı, yatırım kararı başına Sermaye Harcaması (Capital Expenditure – CAPEX) ve Yatırım Getirisi (Return on Investment – ROI) arasındaki denge dikkate alınmalıdır. Teknoloji ve hizmetlere göre mobil ağı olan talebin uzun vadeli geleceğini tahmin etmek için geçmiş verileri kullanmak, MNO'lardaki merkezi ve bölgesel planlama ekiplerinin

hem kapasite hem de kapsama genişletme amaçlı yatırım kararları almasına yardımcı olmaktadır.

Yanlış yatırım kararları, MNO'ların maliyetlerinde gereksiz artışlara ve kapasite veya kapsama boşluklarından muzdarip abonelerin kaybına neden olabilir. MNO'ların merkezi ve bölgesel planlama ekiplerinin kullanım odaklı Anahtar Performans Göstergelerini (Key Performance Indicator – KPI) toplaması ve ön işleminden geçirmesi gerekir. Değişen koşullara uyum sağlayan, dinamik olarak güncellenen yatırım planları sunmaları gerekmektedir. Bu, ancak ilgili algoritmalarla desteklenen otomatik bir mobil ağ planlama sürecinin varlığı ile mümkündür. Aksi durumda hücre başına elle yapılacak işlemler ile bu süreç çok uzun sürmektedir.

Kapasite planlaması; ağ tıkanıklığı ve yetersiz kullanım alanlarının belirlenmesini ve trafiğin mevcut mobil hücreler arasında adil bir şekilde dağıtılmasını içerir. Ağdaki kullanıcı sayısının çok fazla olduğu göz önüne alındığında, doğru tahminleme sonuçlarının elde edilmesi için operatörlerin büyük çaba göstermesi gerekmektedir. Bu durum büyük önem arz eder, çünkü ağ yatırımlarındaki yatırım getirisi, kaynakların ne kadar verimli kullanıldığına ve aynı zamanda gelecekteki kapasite ihtiyaçlarının ne kadar doğru tahmin edildiğine bağlıdır. Rekabetçi pazar koşulları, kullanıcıların yarattığı trafiğin yanında kullanıcı deneyimini de (Quality of Experience - QoE) dikkate almayı ve özellikle servis farkında planlama yapmayı zorunlu hale getirmektedir. Otomatik planlama sürecinin tüm aşamaları yazılımlarla desteklenir ve aşağıdaki altı adımdan oluşur:

1. **Veri Toplama:** Kullanım KPI'larının ve harici metriklerin (müşteri şikayetleri) alınması gereklidir.
2. **Veri İyileştirme:** Veriler hiçbir zaman temiz ve eksiksiz değildir. Bu sebeple enterpolasyon ve ön işleme gereklidir.
3. **Mevsimsellik Tespiti:** Bölgesel mevsimsellik farkındalığı, doğru kararlar ve tahminler için önemlidir.
4. **Kullanım Tahmini:** Yerel hücrelerin gelecekteki kapasite talebinin tahmin edilmesi, kararlar için önemli bir adımdır.

5. **Kapasite Kontrolleri:** Öngörülen KPI'lar ve diğer kısıtlamalar, yatırım kararlarına öncelik vermek için kullanılır.
6. **Plan Önerisi:** Yatırım kararlarındaki çelişkiler giderilir, öneriler raporlanır ve görselleştirilir.

Birden fazla kilit faktör, genel tahmini ve doğal olarak yatırım kararının doğruluğunu etkiler. Şebeke bölgelerinin uzun vadeli kapasite talebi hakkında bir öngörüye sahip olmak için kullanım KPI'larının gelecek değerlerinin tahmin edilmesi, yatırım karar sürecindeki kritik rolü nedeniyle mümkün olduğunca doğru olmalıdır. Ancak gerçek ağ senaryolarında tahmin sonuçlarının genel doğruluğunu azaltan bazı engeller vardır. Bu nedenlerden biri, yeni konuşlandırılan baz istasyonları için yetersiz geçmiş veriye sahip olmaktır. Tüm ağlar için toplam hücre sayısı yeni eklenen hücreler ile zamanla arttığından, çok uzun geçmişe sahip olmayan henüz devreye alınmış mobil radyo hücreleri görmek yaygındır. Bu yeni hücreler için, sadece birkaç ay ile sınırlı olan verileri kullanarak, bunların yıllık mevsimselliklerini veya özel gün etkilerini (millî/dinî bayramlar, kutlama günleri, yılbaşı vb.) değerlendirmek mümkün değildir. Çünkü ilgili hücre için bazı mevsimler ve özel günler henüz hiç yaşanmamıştır. Ancak otomatikleştirilmiş bir yatırım planlama sürecinde, bir yıldan az geçmiş verisi olan bu tip hücreler, yeterli geçmişe sahip diğer hücreler gibi mümkün olduğunca doğru yönetilmelidir. Uzun dönemli tahminlerde ise 5-6 aydan fazla ileriye dönük tahmin işlemleri dikkate alınır. Tahminleme işlemlerini zorlaştıran faktörlerden bir diğeri ise mevsimsellik ve değişim noktalarının ayrıştırılmış olarak tespitidir. Verilerin kalitesi, hedeflenen mevsimsellik aramasını kapsayacak yeterli veriye sahip olmak, mevsimsel faktörleri belirlerken değişim noktalarının farkında olmak, özel günlerin etkileri vb. durumlar da uzun dönemli tahminleme doğruluğu için oldukça önemlidir. Kısaca, zaman serisi verilerinin özelliklerini anlamak ve doğru bir şekilde ayrıştırmak mobil ağların kapasite planlaması için önemlidir.

Bu doktora çalışması kapsamında, mobil radyo ağ performans ve kapasitesinin artırılması ve buna bağlı olarak operasyon verimliliği ve maliyet kazancı odağında daha akıllı kapasite planlama yapılabilmesi için uzun dönemli zaman serisi ve yapay öğrenme odaklı tahminleme algoritmaları ve bunların performansını olumsuz etkileyen ana iki durum ele alınmıştır. Ele alınan bu durumların bertaraf edilebilmesi için hibrit ve kümeleme tabanlı algoritmalar geliştirilmiş ve canlı ağ verileri üzerinde denenmiştir.

Bu durumlar sırasıyla,

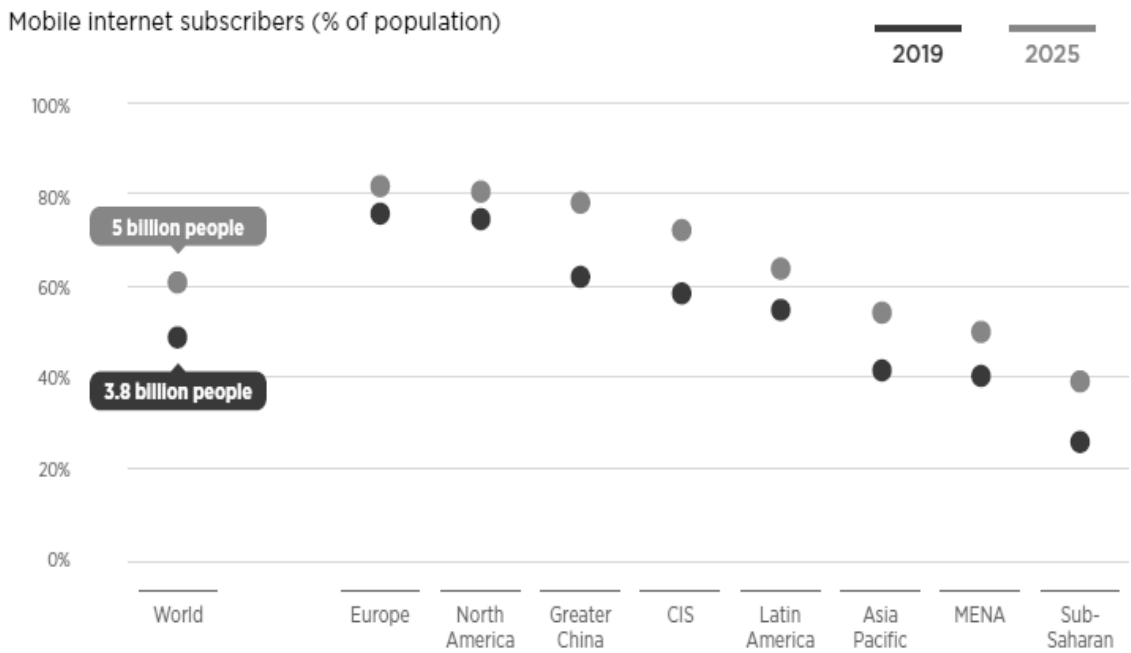
- Mevsimsellik Problemi: mobil radyo hücrelerinin değişim noktası farkında mevsimsellik tespiti ve
- Tahminleme Problemi: yeni devreye alındığı için yetersiz geçmişe sahip olan mobil radyo hücrelerinin uzun gelecek trafik tahminleme hatalarının kontrol altına alınması

şeklinde ifade edilebilir. Çalışma kapsamında, gerçek mobil ağ günlük trafik verileri işlenmektedir ve baz istasyonları üzerindeki uzun vadeli mevsimsellik etkisini belirlemek için yeni bir hibrit çözüm geliştirilmektedir. Bu işlemten sonra, seviye kayması veya üstel eğilim artışı nedeniyle değişim noktası varlığını dikkate alarak baz istasyonları üzerindeki mevsimsellik etkisini çıkarmak için baz istasyonu trafik verilerinin mevsimsellik tespiti ve ayrıştırılması için bir algoritma tasarlanmaktadır. Yerel değişim noktası arama, değişim oranı analizi ve benzerlik kontrol mekanizmaları incelenmektedir. İki yıllık geçmiş trafik verilerini kullanarak çarpımsal mevsimsel faktörleri hesaplanmaktadır. Ayrıştırılmış mevsimsellik etkisi ile baz istasyonlarını etiketlemek, gelecek yılın trafik talebini tahmin ederken daha doğru sonuçlar elde etmeye yardımcı olur. Buna ek olarak çalışma kapsamında, yetersiz geçmiş verilerden kaynaklanan büyük uzun vadeli gelecek tahmini hatalarına odaklanılmakta ve yetersiz geçmişe sahip yeni konuşlandırılmış hücreler için tahmin hatalarını azaltmak ve hatayı kontrol altına almak için hiyerarşik kümeleme tabanlı bir algoritma geliştirilmektedir. Buradaki amaç, hata sayısını önemli ölçüde azaltmak ve en azından bu hücrelerin ilk yılı için tatmin edici hata seviyelerinde trafik tahmin sonuçları üretmektir. Yeni konuşlandırılan hücreler için yetersiz geçmiş verisi, hareketlilik devir ilişkileri, sahadan sahaya uzaklık, hücrelerin servis açısı kullanılarak kümelenmiş komşuların ortak geçmiş özellikleriyle tamamlanır. Geliştirilen hiyerarşik kümeleme tabanlı algoritma ile seçilen hücrelerin geçmiş özellikleri, bu yeni devreye alınmış hücreler için sentetik olarak geriye dönük bir geçmiş oluşturmak için kullanılır ve gelişmiş doğruluk sonuçlarını göstermek için farklı tahminleme algoritmalarına beslenir. SARIMAX, STL-ETS gibi klasik zaman serisi yöntemlerinin ve Vanilla LSTM Sinir Ağları, LSTM Kodlayıcı-Kod Çözücü Sinir Ağları, CNN-LSTM Kodlayıcı-Kod Çözücü Sinir Ağları, ConvLSTM Kodlayıcı-Kod Çözücü Sinir Ağları gibi derin sinir ağı yöntemlerinin tahmin çıktıları bu tez çalışmasında yer alan tamamlayıcı algoritma yardımıyla iyileştirilmektedir. Daha doğru ortalama sonuçlar için hiperparametre ayarlama adımları ve kümeleme için ideal nitelik seçimi de çoklu çalıştırma simülasyonları ile çalışma kapsamında yer almaktadır. |

2. GENEL KISIMLAR

2.1. MEVSİMSELLİK PROBLEMİ TANIMI

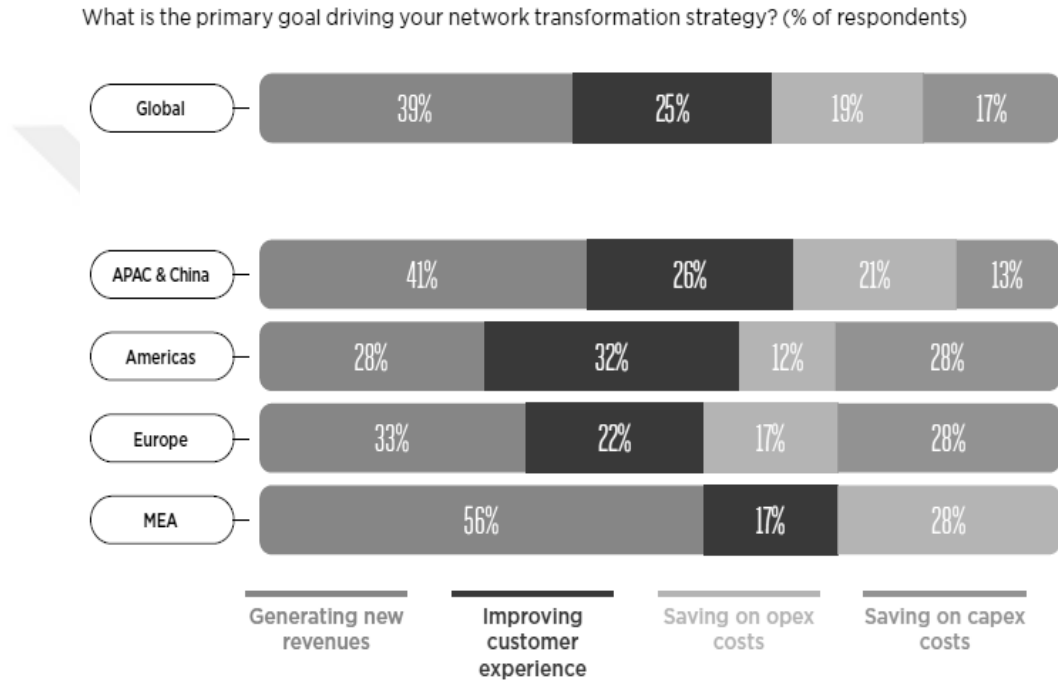
Bilindiği üzere birçok sektör günümüzde dijitalleşmiş süreçlere dönüşmektedir, bu sayede de mobil internet kullanıcılarının sayısı günden güne artmaktadır. Bu dijitalleşme genellikle Makineden Makineye (Machine to Machine - M2M) iletişim ve Nesnelerin İnterneti (Internet of Things - IoT) dönüşümü şeklinde olmakla beraber her geçen gün farklı teknolojiler de ortaya çıkmaktadır. Her gün yeni internet hizmetleri ve uygulamalarının ortaya çıkmasıyla MNO tarafından yönetilecek yük miktarı da artmaktadır. Küresel Covid-19 salgını etkisi de dijitalleştirilmiş işletme sayısının ve ilgili mobil uygulamalara ve hizmetlere olan talebin artmasındaki diğer bir etkidir. GSM Derneği'nin (GSMA) sunduğu Akıllı Mobil Ekonomi Araştırma Raporu'nda, 2025 yılına kadar mobil ağlar tarafından 1,2 milyar yeni aboneye hizmet verileceği ifade edilmiştir (GSMA, 2020). Farklı bölgeler üzerinde mobil abonelerin oranı Şekil 2.1'de gösterilmektedir.



Şekil 2.1. 2025 İtibariyle Mobil Internet Abonelerinin Nüfus Yüzdesi

Ağ üzerindeki çeşitli hizmetler, ağ üzerinde farklı miktarda trafik üretimine karşılık gelmektedir ve her birinin doğası gereği farklı müşteri deneyimi gereksinimleri vardır. Örneğin, aynı mobil baz istasyonundan hem geniş bant video akışı gibi verimlilik duyarlı

uygulama kullanıcıları hem de çevrimiçi oyunlar gibi gecikme duyarlı uygulama kullanıcıları hizmet alıyor olabilirler. Birden fazla hizmet tipindeki farklılıklar ve müşteri beklentileri, Sermaye Harcaması (Capital Expenditure - CAPEX) ve Yatırım Getirisi (Return on Investment - ROI) arasındaki dengeyi yönetmek için doğru zamanda doğru yatırım yapmayı amaçlayan mobil ağ kapasite planlama sürecinin karmaşıklığını arttırmaktadır. Şekil 2.2’de gösterildiği gibi, MNO’ların altyapı yenileme ve ağ dönüştürme kararları için ikinci en yüksek neden olarak, müşteri deneyiminin iyileştirilmesi belirlenmiştir (GSMA, 2020).

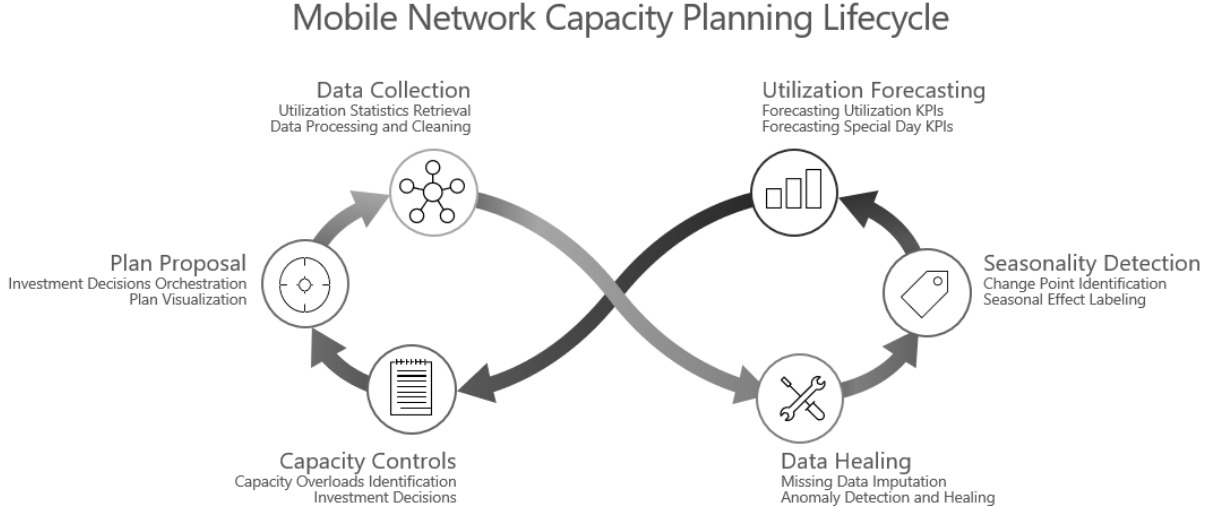


Şekil 2.2. MNO’ların Ağ Dönüşüm Stratejisini Yönlendiren Birincil Hedef

Artan talep, akıllı ROI yönetimi, müşteri deneyimi gereksinimleri, müşteri kaybının önlenmesi gibi nedenler, mobil operatörleri dahili hizmetlerini dijitalleştirmeye ve veri merkezli akıllı kararlar almaya itmektedir. MNO organizasyonlarında, yukarıdaki tüm kısıtları dikkate alarak kapasite yatırım kararlarından sorumlu bölgesel ve merkezi ağ yatırım planlama ekipleri bulunmaktadır. Bu ekiplerin takip ettiği mobil radyo ağ kapasitesi planlama yaşam döngüsünün geliştirilmiş bir süreci Şekil 2.3’te gösterilmektedir.

Ağ planlaması amacıyla işlenen veri kaynaklarının sayısı zamanla artmaktadır. Karar sürecini daha veri merkezli yaklaşımla dijitalleştirmek için tüm süreç, veri analizi ve talep tahmininden gücünü alacak şekilde yazılım tabanlı otonom bir hale dönüştürülmektedir. Tüm süreç, diğer

veri odaklı çözümlere benzer şekilde, birden çok veri kaynağından veri toplama işlemleriyle başlamaktadır.



Şekil 2.3. Mobil Ağ Kapasite Planlama Yaşam Döngüsü

En yaygın olarak kullanılan veri kaynağı, kaynak kullanım istatistikleri hakkında Temel Performans Göstergelerini (Key Performance Indicator - KPI) hesaplamak için performans sayaçları sağlayan Öge Yönetim Sistemleri (Element Management Systems - EMS) günlükleridir. Örneğin, Baz İstasyonuna (Base Station - BS) bağlı kullanıcı sayısı, megabayt cinsinden trafik miktarı, Uzun Vadeli Evrim (Long-Term Evolution - LTE) hücreleri için Fiziksel Kaynak Bloğu (Physical Resource Block - PRB) kullanımı vb. tüm istatistikler, belirli zaman aralıklarını (örneğin her 5, 15, 30 veya 60 dakikada bir) gösteren zaman damgalarıyla etiketlenmektedir.

Ağ KPI'larının yanı sıra, Müşteri İlişkileri Yönetimi (Customer Relationship Management - CRM) sistemlerinden coğrafi konumlandırılmış müşteri şikayetleri, abone bazlı Kullanıcı Ekipmanından (User Equipment - UE) toplanan kitle kaynaklı kalite örnekleri, sinyal yayılım modelleme aracı dışı aktarmaları (kapsama sinyal seviyeleri), taşıma katmanı istatistikleri trafik tahmini için kullanmak üzere EMS tarafından oluşturulan KPI'ların yanında kullanılabilir. Trafik ve talep için yüksek kaliteli tahmin işlemi, veri temizleme, eksik veri tamamlama, anormallik giderme ve uzun vadeli tahminleme için günlük veya haftalık veri birleştirme gibi bazı ön işleme operasyonlarını gerektirmektedir.

Genel tahmini ve doğası gereği yatırım kararı doğruluğunu birden çok anahtar faktör etkiler. Bu faktörleri (verilerin kalitesi, mevsimsellik tespiti için yeterli veriye sahip olma, mevsimsel faktörleri belirlerken değişim noktalarının farkında olma, özel günlerin etkileri vb.) doğru şekilde ayrıştırmak önemlidir. MNO politikaları ve iş kuralları, öngörülen trafik ve diğer kullanım KPI'larına uygulanır ve yatırım önerileri ile takip edilir. Yaşam döngüsünün son aşamasında, tüm öneriler önceden tanımlanmış kısıtlamalara göre değerlendirilir ve düzenlenir. Düşük mesafelerde birbirini etkileyebilecek çelişkili öneriler, MNO politikaları ve öncelikleri ile ilgili olarak çözülür. Geçmişteki değişim noktalarının farkında olarak baz istasyonlarının mevsimsellik etkisinin tespiti, yatırım öneri kararlarının doğruluğunu artırır. Özellikle mevsimsel bölgelerde geçici mobil hücreler, BBU ve akümülatörler ile telafi edilebilecek lokasyonların belirlenmesi, sürekli yatırımdan daha fazla tasarruf sağlamaktadır. Kalıcı ve kalabalık bölgelerde ise yatırımın en geç ne zaman yapılabileceğinin ROI odaklı olarak tespit edilebilmesi MNO için tasarruf olanağı sağlamaktadır.

Bazı sektörel kapasite genişletmeleri için yüksek mevsimsellik gösteren baz istasyonlarında doğru yatırım zamanını bulmak, MNO'lar için başka bir maliyet tasarrufu faktörüdür. Böylelikle müşteri deneyimini korumak için doğru zamanda doğru baz istasyonlarına genişletmeler yapılmış olur. Gereksiz yatırım, karbon ayak izi artışı nedeniyle çevre üzerinde başka bir maliyete neden olur. Doğru sayıda istasyona sahip bir bölgenin kapasitesinin yönetilmesi, MNO'nun karbon ayak izini azaltır ve enerji tasarrufu sağlar. Benzeri şekilde merkezî enerji tasarrufu algoritmalarının da tahmin doğruluğu için yüksek mevsimsellik başlangıç-bitiş tarihlerinden haberdar olmaları gerekmektedir.

Mobil ağ kapasite planlama yaşam döngüsü için, yazılım odaklı bir planlama yaklaşımı geliştirebilmek için değişim noktalarının farkında olma ve bir baz istasyonundaki mevsimsel etkiyi belirleme işlemlerinin tek bir çözüm altında birleştirilmiş olması çok önemlidir. Böylece çözüm, çevrimdışı analiz ve önceki veriler üzerinde öğrenme yetenekleriyle denetimsiz bir çözüm olarak çalışabilir. Algoritmaların çıktıları (baz istasyonlarının mevsimsel etiketlenmesi ve mevsimsel etkinin yüzdeye dayalı miktarı) tahmin algoritmalarına girdi olarak sağlanabilir.

Mevsimsel hareketli ortalama, STL (Seasonal and Trend Decomposition by Using Loess – Loess Kullanarak Mevsimsel ve Trend Ayrıştırma) veya Prophet kütüphanesi gibi yaygın olarak kullanılan zaman serisi ayrıştırma algoritmaları, doğru tanımlanmadıkları takdirde değişim noktalarından muzdariptir. Bu durum, dramatik değişim noktası değişikliğinin eğilim

bileşeni yerine mevsimsel bileşene adreslenmesi, mevsimsel özellikleri gereksiz yere düzleştirerek eğilim bileşenini yanlış uydurma veya gerçek değişim noktalarını yüksek oranda çarpımsal mevsimsellik etkisiyle karıştırma gibi bazı yanlış ayrıştırmalara neden olur. Değişen müşteri davranışlarının veya devam eden yatırım temelli ağ genişletmelerinin etkisi altındaki dinamik yapıları nedeniyle mobil radyo ağlarına bakıldığında bu tür durumların oldukça yaygın olduğu görülmektedir. Bu nedenle, otomatik kapasite planlama çözümleri tasarlayabilmek için, her bir baz istasyonunun değişen hikayesini ayrı ayrı ele almak bir zorunluluktur. Baz istasyonunda net bir mevsimsellik veya değişim noktası olmayan daha düşük ölçekli çarpımsal mevsimsellik mevcutsa, tüm yaygın ayrıştırma yöntemleri için mevsimsellik ve eğilim bileşenlerini ayrıştırmak kolay olacaktır. Ancak veri setindeki iki yıllık geçmişe sahip hücrelerde olduğu gibi zaman serisi kısa olduğunda ve birçok baz istasyonunda karakteristik varyans yüksek seviyede olduğunda, bu durumların farkında olmak ve bazı ek kontroller ve ayarlamalarla iyileştirmek için ortak yöntemler gerekmektedir. Değişim noktaları, stokastik süreçlerin çıktıları üzerindeki olasılık dağılımına ait istatistiksel değişimlerinin gerçekleştiği anlardır.

$X = \{X_t, t = 1, \dots, T\}$, gözlemlenen trafik hacmi örnekleri dizisi ise, bu durumda (2.1) ile trafikte değişiklik olmadığı ifade edildiğinde, alternatif hipotez (2.2) ile bir değişim noktası mevcudiyeti gözlemlenebilecektir (Chen ve Zhang, 2015).

$$H_0: P_{X_1} = P_{X_2} = \dots = P_{X_n} \quad (2.1)$$

Burada $\{1, \dots, n\}$ X_t serisinin aralıkları ve P_{X_n} , n 'in olasılıksal yoğunluk fonksiyonudur.

$$H_1: \exists 1 \leq k < n, \begin{cases} P_{X_k}, & 1 \leq k \\ P_{X_n}, & k < n \end{cases} \quad P_{X_k} \neq P_{X_n} \quad (2.2)$$

Literatürde serideki beklenmedik değişiklikleri belirlemek için zaman serisi verilerinde değişim noktası tespiti veya anormallik algılama algoritmalarını ele alan çalışmalar mevcuttur (Aminikhangahi ve Cook, 2017). Değişim noktası algılama algoritmaları farklı kategorilere ayrılmıştır. Çevrimiçi ve çevrimdışı algoritmalar ana kategorilerdir. Çevrimiçi algoritmalar, izledikleri süreçle eşzamanlı olarak çalışır, her veri noktasını kullanılabilir hale geldikçe işler ve gerçek zamanlı bir çözüm olarak işlem, bir sonraki veri noktası gelmeden önce tamamlanmalıdır. Öte yandan, çevrimdışı algoritmalar tüm veri setini bir kerede analiz eder ve çalışma zamanı süresince gerçek zamanlı bir karar yoktur (Downey, 2008). Burg ve Williams

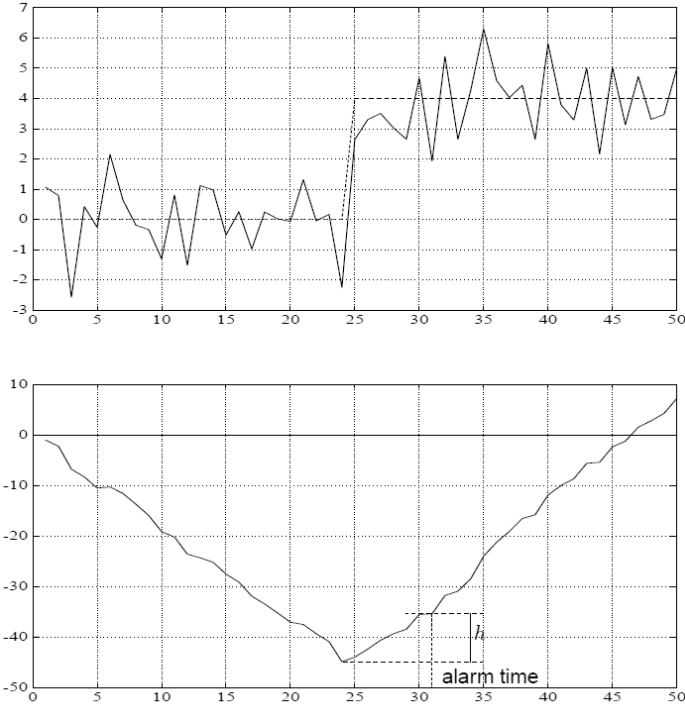
(2020) tarafından listelendiği üzere tek değişkenli, çok değişkenli, model tabanlı, parametrik olmayan gibi başka değişim noktası algılama algoritma kategorileri de vardır. İlk yaklaşımlardan biri, Page (1954) tarafından yaygın olarak kullanılan ve önerilen Kümülatif Toplam Kontrolü (Cumulative Sum Control - CUSUM) grafik yöntemidir. Son k gözlemin ağırlıklı toplamı olarak dağılımın ortalamasındaki bir değişikliği tespit etmektir. Kümülatif toplam değişiminin bir eşiğin üzerinde olduğunu düşünür. CUSUM algoritmasının birden fazla türevi vardır ve bunlardan biri, değişim noktasından önce negatif bir sapmayı ve değişim noktasından sonra pozitif bir kaymayı gösteren logaritmik olabilirlik oranı S_k (5) tipik davranışına sahip sezgisel bir yaklaşımdır. Bu durum Şekil 2.4'te görselleştirilmiştir. $H_0: \theta = \theta_0$ ve $H_1: \theta = \theta_1$ ise (2.3) ve (2.4) y_j 'den y_k 'ye gözlemler için log oranıdır.

$$S_j^k = \sum_{i=j}^k s_i \quad (2.3)$$

$$s_i = \ln \frac{p_{\theta_1}(y_i)}{p_{\theta_0}(y_i)} \quad (2.4)$$

İkili bölümlleme ile olasılık oranı üzerine inşa edilen başka yaklaşımlar da vardır. Scott ve Knott (1974), örnek işleme araçlarını dengeli bir tasarımda bölmek için iyi bilinen bir küme analizi yöntemini kullanmanın sonuçlarını incelemiş ve karşılık gelen bir olasılık oranı testinin, elde edilen gruplar arasındaki farklılıkların önemini değerlendirme yöntemini nasıl verdiğini göstermiştir.

Olasılıklı yöntemler, zaman serilerini bölümlere ayırarak ve değişim noktası varlığı için her bölümün olasılık çıktılarını oluşturarak değişim noktası tespiti için de kullanmaktadır. Downey, algoritmanın çekirdeğinin, her bir i indeksi için, son (en son) değişim noktasının i 'de meydana gelme olasılığını hesaplayan bir denklemler sistemi olarak Bayes yaklaşımlı bir algoritma önermiştir. Bu algoritmayı değişim noktası algılama problemine uygulayarak değerlendirmiş ve Genelleştirilmiş Olabilirlik Oranı (Generalized Likelihood Ratio - GLR) algoritması ile karşılaştırmıştır (Downey, 2008). Bu araştırmada özel bir değişim noktası probleminin izleme olduğu belirtilmiştir. İzleme probleminin amacı, bir zaman serisini sabit aralıklara bölmek ve her aralıktaki sürecin parametrelerini tahmin etmektir. Basit bir yaklaşım ile değişim noktasını tespit etmek, değişikliklerin yerini tahmin etmek için hipotez testini kullanmak ve ardından her aralıktaki sürecin parametrelerini tahmin etmek için geleneksel teknikleri kullanmaktır.



Şekil 2.4. CUSUM Logaritmik Olabilirlik Oranı Davranışı, Sabit Varyanslı Bir Gauss Dizisinin Ortalamasındaki Değişim Örneği

Zaman serisi sayısındaki ve veri akışındaki artışa paralel olarak, birden çok değişim noktasının yerini verimli ve doğru bir şekilde tahmin edebilmek için yükselen bir talep vardır (Killick ve Eckley, 2014). Birden fazla değişim noktası algılama problemi için de Bayes kuralları kullanılabilir (2.5). T serisinin θ parametresinin olasılık dağılımı ile oluşturulmasına izin verilmiştir. Bu durumda,

$$P(\theta|T) = \frac{P(T|\theta) P(\theta)}{P(T)} \quad (2.5)$$

Burada $P(\theta|T)$ sonsal olasılık, $P(T|\theta)$ önsel olasılık, $P(\theta)$ önceki bilinen durumdur. Buna göre C 'de tekil bir değişim noktası tespit eden bir Bayes Modeli (ayrık bir parametre olarak $C \sim Uniform(1, T)$) C noktasındaki hipotezi doğrulamak için sonsal bir olasılık bulmayı hedefler. İş zaman serileri genellikle insan davranışının bir sonucu olarak çok dönemli mevsimselliğe sahiptir (Taylor ve Letham, 2018). Radyo erişim ağlarının kapasite planlaması amacıyla trafik serilerinin otomatik tahmini için seriyi seviye, eğilim ve mevsimsel bileşenlere ayırmak zaruridir (Hyndman ve Khandakar, 2008).

Kısa vade de farklı değildir ve mobil ağ trafik davranışı tarafından oluşturulan uzun vadeli zaman serileri mevcuttur (Nikraves, vd., 2016; Cortez, vd., 2006; Tikunov ve Nishimura, 2007; Sciancalepore, vd. 2017). Serinin mevsimsellik bileşenini modellemenin farklı yolları vardır. Bunlardan biri, periyodik etkilerin standart Fourier serisi olarak modellenmesidir (Harvey ve Shephard, 1993). Box-Cox dönüşümü, Otoregresif Hareketli Ortalama (Autoregressive Moving Average - ARMA) hataları, Eğilim ve Mevsimsel Bileşenler (Trend and Seasonal Components - TBATS) modellemesi ile Trigonometrik Üstel Yumuşatma Durum-Uzay modeli için de benzerdir (Livera, vd., 2011). Bu durum (2.6)'da gösterilmektedir.

$$s(t) = \sum_{n=1}^N \left(a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right) \quad (2.6)$$

Burada P periyot uzunluğudur. Örneğin $P = 12$ olduğunda yıllık veri aylık olarak toplanır, $P = 24$ olduğunda yıllık veri yarım aylık (15 günlük) olarak toplanır. Kısmî bağlanım çizgisi uydurma ile filtreleme tabanlı ayırıştırma teknikleri vardır. Mevsimsel Hareketli Ortalamalar alternatif olarak ağırlıklı ortalamayı içerenlerinden biridir. Ayrıca, X-11 yöntemi, ARIMA Zaman Serilerinde Mevsimsel Çıkarma (Seasonal Extraction in ARIMA Time Series - SEATS) yöntemi (Dagum ve Bianconcini, 2016) veya bunların karma versiyonu gibi aylık veya üç aylık olarak toplanan verilerden oluşan uzun serilerin mevsimsel ayırıştırılması için kullanılan yöntemler de vardır. X-11 ve SEATS yöntemleri, güçlü mevsimsellik modellerini eğilimden temiz şekilde ayırabilmek için yeterince uzun seriler sağlanırsa, eğilim bileşenindeki beklenmedik ani düşüşlerin farkında olabilmektedirler. Ancak, aylık toplanan veri noktaları, veri setimizdeki iki yıllık geçmiş verilere çok fazla benzemediğinde, yalnızca X-11 veya SEATS yöntemlerini kullanarak çarpımsal yıllık mevsimselliği modellemek kolay değildir. Bu yöntemlere bir alternatif olarak, X-11 ve SEATS yöntemlerinden farklı olarak verilerin toplanma çözünürlüğüne bakılmaksızın doğrusal olmayan ilişkilerdeki eğilim ve mevsimsel bileşenlerin esnek kontrolünü sağlayan STL yöntemi vardır. STL, zaman serilerini sırasıyla eğilim, mevsimsel ve kalan bileşenlerine ayırmak için bir filtreleme prosedürüdür. STL, bir dizi daha az pürüzsüz uygulamadan oluşan basit bir tasarıma sahiptir. Basitlik, prosedürün özelliklerinin analizine izin verir ve çok uzun süreli seriler ve büyük miktarlarda eğilim ve mevsimsel düzeltme için bile hızlı hesaplamaya izin verir. STL'nin diğer özellikleri, neredeyse sürekli bir şekilde, çok az miktarda yumuşatarak ya da çok fazla yumuşatarak, güçlü eğilim tahminleri ve mevsimsel bileşenlere kadar değişen,

mevsimsel ve eğilim yumuşatma miktarlarının özelleştirilmiş halidir (Cleveland, vd. 1990). STL, seri bileşenlerini yalnızca toplamsal olarak modelleyebilir. Çarpımsal mevsimselliği modelleyebilmek için, mobil ağlar için veri seti serisinde sıklıkla bulunan log dönüşümünün uygulanması gerekmektedir. Açıklayıcı değişkenler üzerinde, tepkisi tahmin edilen noktaya yakın değerler olacak şekilde, daha yumuşak bir yerel ağırlıklı polinom bağlanımı uygulanır. STL prosedürü için parametreler, belirli bir zaman serisinin özdeğer ve frekans tepkisi analizinden elde edilir. Yinelenen döngü, iç ve dış döngü olmak üzere iki özyinelemeli prosedürden oluşur. Y , trafik hacmi verileri olmak üzere ve T , eğilim bileşenini veya uzun vadeli yönü temsil edecek şekilde, S mevsimsel bileşeni veya periyodik olarak tekrar eden bir modeli ve ϵ rastgele hatayı temsil ediyor dersek, bu durumda, STL iç döngüsü aşağıdaki altı temel adımı gerçekleştirir (Theodosiou, 2011):

- 1- **Eğrilme:** k , bileşenlerin algoritmanın k . geçişindeki değeri anlamına gelmek üzere, ilk yinelemede eğilimin sıfır olduğu varsayılır. $k + 1$ iterasyonunda, trafik verilerinden önceki geçişin bileşenleri çıkartılır. Bu durum (2.7)'de verilmiştir.

$$Y_t - T_t^{(k)} \quad (2.7)$$

- 2- **Mevsimsel düzeltme:** Ön mevsimsel bileşeni almak için ilk adımın geri kalanına uygulanan Loess yumuşatma, $S_t^{(k+1)}$ şeklindedir.
- 3- **Düzeltilmiş mevsimselliğin filtrelenmesi:** Ön mevsimsel bileşene hareketli bir ortalama uygulayarak ve daha sonra başka bir Loess pürüzsüz, $\widetilde{T_t^{(k+1)}}$ geri kalan ön eğilim bileşeni olarak hesaplanır.
- 4- **Düzeltilmiş mevsimselliğin çıkartılması:** İlave mevsimsel bileşeni, bu iki ön bileşen arasındaki fark haline gelir. Bu durum (2.8)'de verilmiştir.

$$S_t^{(k+1)} = \widetilde{S_t^{(k+1)}} - \widetilde{T_t^{(k+1)}} \quad (2.8)$$

- 5- **Mevsimselleştirme:** $k + 1$ iterasyonu için düzeltilmemiş eğilim bileşeninin elde için mevsimsel bileşen orijinal seriden çıkarılır. Bu durum da (2.9)'da verilmiştir.

$$Y_t - S_t^{(k+1)} \quad (2.9)$$

- 6- **Eğilim düzeltme:** Son adımda, mevsimsellikten arındırılmış kalan serileri, $T_t^{(k+1)}$ eğilim bileşeninin bir tahminini elde etmek için Loess ile yumuşatılır.

Bu nedenle, iç döngünün her geçişi, mevsimsel bileşeni güncelleyen bir mevsimsel yumuşatma uygular ve ardından eğilim bileşenini güncelleyen eğilim yumuşatma adımını uygular.

Serideki mevsimsel kalıplara göre deęişim noktaları yerine anormallikleri ele almak için ESD algoritması (Rosner, 1975) üzerine inşa edilen Mevsimsel Hibrit Aşırı Eğitilen Sapma (S-H-ESD) gibi hibrit çözümler vardır.

2.2. TAHMİNLEME PROBLEMİ TANIMI

Her geçen gün kümülatif mobil hizmet abone sayısı artmakta ve kullanıcı başına üretilen ağ trafięi miktarı da sürekli olarak büyümektedir. Mobil internet hizmetinin yanında yeni hizmet türleri de devreye alınmaktadır. Örneęin kurumsal sanal ağlar (Virtual Private Network – VPN) ya da sabit kablosuz ağlar (Fixed Wireless Networks – FWA) bu yeni servislere örnek olarak söylenebilir. Artan bu talebi, deęişen abone davranışlarını ve uygun maliyetli ağ yönetimini sağlayabilmek için MNO planlama ekipleri, veri ve makine öğrenimi ile güçlendirilmiş otomatik planlama süreçlerini kullanarak daha akıllı kapasite yatırım planları yapmaya ihtiyaç duymaktadır. Teknoloji ve hizmetlere göre mobil ağ talebinin uzun vadeli geleceęini tahmin edebilmek, MNO merkezi ve bölgesel planlama ekiplerinin hem kapasite hem de kapsama yatırım kararları dahil olmak üzere daha doğru kararlar almalarına yardımcı olur. Şebeke bölgelerinin uzun vadeli kapasite talebi hakkında bir öngörüde bulunmak için KPI tahmini, karar sürecindeki kritik rolü nedeniyle mümkün olduęunca doğru olmalıdır. Ağlar, artan radyo erişim gereksinimini karşılamak için eklenen yeni hücreler ve baz istasyonlarıyla doğal olarak büyüdüęünden, bu yeni konuşlandırılan hücreler, uzun vadeli kapasite taleplerini doğru bir şekilde tahmin etmek için maalesef yeterli uzunlukta geçmiş veriye sahip deęildir. Bu da tahminleme algoritmalarının bu durumdaki hücrelerin KPI tahminlerinde ciddi hatalar yapmasına sebebiyet vermektedir. Oysaki bu hücreler de yeterli geçmişe sahip komşuları gibi ne kadar doğru tahmin edilirse geleceęe dair yatırım planlarının doğruluęu o kadar artacaktır. Bu sorunun üstesinden gelmek ve bu hücreler için de tatmin edici bir doğruluk seviyesine sahip olmayı mümkün kılarak ilgili hücrelerin uzun vadeli tahminlerini iyileştirmek için sentetik geçmiş oluşturulabilir. Bu veriyi doğru oluşturmak adına çok deęişkenli hiyerarşik kümeleme tabanlı bir algoritma geliştirdik. Dışsal Regresyonlu Mevsimsel Otomatik Regresif Entegre Hareketli Ortalama (SARIMAX) gibi klasik zaman serisi yöntemleri, Loess ile Mevsimsellik Trend Ayırıştırması ve Üstel Düzeltme (STL-ETS) gibi zaman serisi ayırıştırması tabanlı yöntemler dahil olmak üzere farklı tahmin yöntemlerinin tahminini iyileştirmeyi hedefledik. Bunlara ek olarak daha güncel tahminleme yaklaşımları olan Vanilya Uzun Ömürlü Kısa Süreli Bellek (LSTM) ağları, LSTM Kodlayıcı-Kod Çözücü Sinir Ağları, CNN-LSTM Kodlayıcı-Kod

Çözücü Sinir Ağları ve Evrişimli LSTM (ConvLSTM) Kodlayıcı-Kod Çözücü Sinir Ağları gibi derin sinir ağı yöntemleri üzerinde de test ettik. Çözüm sonuçlarımızı, 2 yıllık günlük geçmiş verilerle bir Kademe-1 Avrupa mobile operatörünün canlı ağından ölçülen 70 LTE hücresi üzerinde elde ettik. 10 ay sonrasına ait tahmin gereksinimindeki büyük tahmin hatalarını yaklaşık %35 Ortalama Mutlak Yüzde Hatası (MAPE) seviyeleriyle sınırlandırmayı başarabildik.

Mobil ağı KPI tahminlemede kullanılan literatür çalışmaları sonraki bölümde detaylıca özetlenmiştir. Bizim yaklaşımımız olan kümeleme tabanlı sentetik geçmiş verisi oluşturarak tahminleme iyileştirme çözümüne yakınsayabilecek, farklı kümeleme algoritmaları ve nitelikler kullanan önceki çalışmalar bu kısımda listelenmektedir.

Zamansal örüntülerle uzamsal ilişkileri araştırmak veya farklı yönlerden tahmin doğruluğunu geliştirmek için yapılan bazı çalışmalar, hibrit bir çözümün parçası olarak bazı kümeleme algoritmalarını içerir. Bildiğimiz kadarıyla, genellikle K-Ortalama (K-Means) kümeleme algoritması kullanılmıştır.

Xu vd. (2018) çalışmasında kümelenmiş baz istasyonları arasında tıkanıklık ve kapasite yönetimi için tatil trafiğini tahmin etmek amacıyla k-ortalama yöntemi kullanılmıştır. Baz istasyonu düzeyinde 15 gün sonrası tahmin edilmeye çalışılmıştır. Yazarlar, baz istasyonunun tatil dönemindeki sapmaları üzerinde k-ortalama yöntemini kullandılar ve bu nedenle küme başına belirli rastgele orman tahmin modellerini eğitmek için yetersiz veri sorunlarını çözdüler. K-ortalama kümeleme algoritmasını ele alan bir başka çalışma da Zhang vd. (2019) çalışmasıdır. Bu çalışmada, yazarlar ConvLSTM Sinir Ağları, K-Ortalama, Derin Transfer Öğrenmeyi (Deep Transfer Learning) birlikte kullanarak trafik tahmini için karmaşık gizli kalıpları yakalamak adına uzay-zamansal etki alanları arası korelasyonlara sahip bir derin öğrenme mimarisi geliştirdiler. Hücre düzeyinde trafik 150 saat sonrasının tahmini hedeflendi. Yazarlar, Telekom Italia veri setinden 61 gün uzunluğundaki veriler üzerinde çalışmışlardır. Birbirleri arasındaki benzer uzay-zamansal kalıpları kontrol ederek hücre tahmin modellerini daha kolay ve daha hızlı eğitmek amacıyla transfer öğrenme için k-ortalama kümeleme algoritmasını kullanmışlardır. İçeride kümeleme kullanılarak yapılan bir diğer kısa dönemli tahmin çalışması da Zang vd. (2015) çalışmasıdır. Elman Sinir Ağı'nı (Elman Neural Network - ENN) eğitmek için uzaklık tabanlı k-ortalama kümelemenin geliştirilmesi ve bileşenlerinin dalgacık dönüşümü ile çıkarılması araştırmacılar tarafından incelenmiştir. 358

baz istasyonundan 7 günlük saatlik veril toplandı ve sonraki 24 saati tahmin etmek için küme düzeyinde tahmin yapılmıştır. Doğruluk kıyaslaması için farklı boyutlarda kümeler üzerinde çalışılmıştır. Narmanlioglu vd. (2018) çalışmasında yazarlar, genellikle sadece konum tabanlı mesafeleri veya mekansal hareketleri ele alan diğer çalışmalardan farklı olarak, k-ortalama algoritmasını mobilite (hareketlilik) bilinciyle genişleterek geliştirilen yeni bir kümeleme yönteminin ayrıntılı bir analizini yapmışlardır. Algoritmalarını bazı Kendi Kendini Düzenleyen Ağ (Self Organizing Network - SON) modüllerini geliştirmek ve geleneksel SON işlevleri için fırsatları araştırmak için kullanabileceklerini paylaşmışlardır. Bu çalışmaların hiçbirisi uzun dönemli tahminlerde yetersiz veri sorununu ortadan kaldıracak bir kümeleme algoritmasını ele almamakta ve hiçbirisi bizim çalışmamızda olduğu gibi mobilite ilişkileri, siteden siteye uzaklık ve hizmet alanı benzerlikleri gibi özelliklerle hiyerarşik kümeleme üzerinde bir kümeleme çözümü sağlamamaktadır.

2.3. MOBİL AĞ TRAFİK TAHMİNLEMESİ LİTERATÜR TARAMASI

Bu bölümde, bu tezde ele alınan “Mobil Kablosuz Ağların Optimizasyonunda Yapay Zeka Yöntemlerinin Kullanılarak Operasyonel Verimliliğin Artırılması” probleminin önemli bir parçası olan ağ trafiğinin tahminlenmesi ile ilgili, literatürde gerçekleştirilen çalışmalar incelenmiştir. Bir mobil ağdaki uçtan uca ağ trafiği, RAN (Radio Access Network–Radyo Erişim Ağı), TN (Transmission Network–İletim Ağı) ve CN (Core Network–Çekirdek Ağ) olmak üzere üç katmandan oluşur. Literatür taraması gerçekleştirilirken bu 3 katmanın özellikleri ve TN’nin yönlendiriciler, anahtarlar ve mikro bağlantılar içermesi nedeniyle baz istasyonu trafiğinin yanında, Ethernet tabanlı trafiğin tahmin edildiği çalışmalar da ele alınmıştır. Çalışmalar, kullanılan yöntemlere göre “Doğrusal Zaman Serisi Yöntemleri Kullanan Çalışmalar”, “Doğrusal Olmayan İstatistiksel ve Makine Öğrenmesi Tabanlı Yöntemleri Kullanan Çalışmalar”, “Derin Öğrenme Dahil Yapay Sinir Ağı (Artificial Neural Networks – ANN) Tabanlı Yöntemleri Kullanan Çalışmalar”, “Hibrit Yöntemler Kullanan Çalışmalar” şeklinde gruplandırılarak incelenmiş, sonrasında da toplu bir tablo halinde sunulmuştur.

Doğrusal Zaman Serisi Yöntemleri Kullanan Çalışmalar aşağıdaki şekilde ifade edilmiştir. Jung vd. (2006) çalışmasında kısa vadeli tahmin yapmak için tüm ağların bir yıllık saatlik ethernet paketi veri trafiği günlük, haftalık, aylık olarak toplanmış, AR (Autoregressive–Otoregresif), MA (Moving Average–Hareketli Ortalama), ARMA (Autoregressive Moving

Average–Oto regresif Hareketli Ortalama) ve ARIMA (Autoregressive Integrated Moving Average–Oto regresif Entegre Hareketli Ortalama) modelleri durağan trafiğe uygulanmış ve AR(1) modeliyle günlük veri toplamının ethernet trafiğinin kısa vadeli tahmini için daha uygun olduğu öne sürülmüştür. Ancak bu çalışmada çalışma serilerin doğrusal olmayan yapısıyla mevsimsel veya mekansal farkındalık mevcut olmadığından yeterli doğruluk sonuçları ve değerlendirme ölçütleri sağlanamamıştır. Yu vd. (2010) çalışmasında Heilongjiang eyaleti için Çin’deki gerçek mobil ağdan veri toplamış ve aylık toplu ortalama günlük trafik üzerine tahminlerde bulunulmuştur. Aylık verilerin durağan olmaması eğilimle değişen homojen varyansla açıklanabildiğinden, Çoklu Mevsimsel ARIMA yöntemi kullanılmış ve tahmin aralığı 1–6 ay arasında değiştirilerek sonuçlar elde edilmiştir. %1,38 ile %6 arasında değişen bir MAPE (Mean Absolute Percentage Error–Ortalama Mutlak Yüzde Hatası) değerine sahip olan sonuçlarda yıllık mevsimsellik, ACF (Auto-Correlation Function–Otomatik Korelasyon Fonksiyonu) çıktısının bir incelemesi olarak kabul edilmiştir. Ayrıca bu çalışmada geçmişteki verilerin kendi kendine benzerliğini Pearson korelasyon katsayısı üzerinden değerlendirerek daha büyük hata oluşma nedenleri de kontrol edilmiştir. Miao vd. (2014) çalışmasında Çin’in iki vilayeti için 2G/3G (2./3. Nesil Kablosuz Telefon Teknolojisi) arama izlerinden toplanan trafik verilerine göre çarpımsal SARIMA (Seasonal ARIMA–Mevsimsel ARIMA) kullanılmıştır. Burada 4200 BS’de (Base Station – Baz İstasyonu) toplam 38 gün için saatlik veri modelleme yapılmış ve sonraki saatlerin değerleri %2,2 ile %7,5 arasında değişen bir MAPE değerine sahip olarak tahmin edilmiştir. BS trafikleri, BSC (Base Station Controller–Baz İstasyonu Denetleyicisi) seviyesinde bir araya getirilerek yüksek, düşük, orta seviye trafik seviyeleri olarak sınıflandırılmış ve kalan kontrollü modellerin uygunluğu için istatistiksel testlerden yararlanılmıştır. Tikunov ve Nishimura (2007) çalışmasında arama tıkanıklıklarını proaktif olarak azaltmak için Erlang olarak sonraki 7 gün için saatlik arama trafiğini tahmin etmek amacıyla, çalışmadaki katsayıların otomatik ayarlanmasını sağlayan HW-ETS (Holt-Winter’s Exponential Smoothing–Holt-Winter’in Üstel Yumuşatması) yöntemi kullanılmıştır. Burada yoğun saatler için sadece hafta içi günler kullanılmış ve son derece dalgalanan bir trafiğe sahip olan hafta sonu değerleri elimine edilmiştir. 7 günlük tahmin için 18 günlük verinin ele alındığı çalışmada doğruluk ölçütü olarak NRMSE (Normalized Root Mean Square–Normalleştirilmiş Kök Ortalama Kareler) kullanılmış ve 0,12 ile 0,28 arasında değerler ortaya çıkmıştır. Jia vd. (2009) çalışmasında Tikunov ve Nishimura (2007) ile aynı kullanım durumunu ele alınmış ve sonraki 7 günün tahminini saat bazında doğrulamak için gerçek bir

ağın 269 hücresinden 28 günlük saatlik veri toplanmıştır. Ancak NRMSE ile çarpımsal SARIMA uygulanmış ve analizler değişen trafik düzenleri aracılığıyla; bölgeler, caddeler, üniversite kampüsleri ve ticari bölgeler olarak dört konum kategorisine göre kategorize edilerek 0,10 ile 0,26 arasında puanlar elde edilmiştir. Cao vd. (2018) çalışmasında entropi ağırlık yöntemi ile tanımlanan sıcak nokta alanının trafik hacmini tahmin etmek için HW-ETS ve ARMA kullanılmıştır. Gerçek veriler üzerinde, 4 günlük geçmiş kullanılarak ertesi günün tahminindeki MAPE değerleri, HW-ETS için %53,43 ve %13,40 seviyesinde gözlemlenmiştir.

Literatürdeki bazı çalışmalar girdi trafiği serilerine bazı dönüşümler uygulayarak doğrusal modellerin performansını artırmayı hedeflemektedir. Syed vd. (2010) çalışmasında, veriler üzerinde SARIMA kullanmadan önce %20 sıkıştırma, yeniden yapılandırma ve WF (Wavelet Filters – Dalgacık Filtreleri) ile dönüştürme işlemleri uygulanmış ve bu sayede genel tahminlerdeki doğruluk oranlarında gelişme olduğu ifade edilmiştir. Uzun vadeli trafik tahmini için omurga yönlendiriciler üzerinde analizlerin yapıldığı Guang vd. (2005) çalışmasında ARIMA yönteminde ayrıştırılmış modelleme (seviye, eğilim, mevsimsellik vb.) kullanılmıştır. Wang vd. (2015) çalışmasında ise BS trafik verileri FFT (Fast Fourier Transforming – Hızlı Fourier Dönüşümü) gibi zaman serisi teknikleri kullanılarak üç kategoriye ayrılmış ve normal dağılımlı bölgelerin mekansal modellemesi yapılmıştır. El Hag ve Sharif (2007) çalışmasında Box-Jenkins metodolojisi tarafından bulunan ARIMA bileşenlerinin üstüne bir geribildirim bileşenini uygulamak için, ARIMA üzerinde AARIMA (Adjusted ARIMA – Düzeltilmiş ARIMA) olarak adlandırılan özel bir ARIMA türü üzerinde de çalışılmış ve doğruluğu MAE açısından karşılaştırılan sonuçların ARIMA puanlarına göre %50 iyileşmeye sahip olduğu görülmüştür. Ntalampiras ve Fiore (2018) çalışmasında MEC (Multi-Access Edge Computing–Çoklu Erişimli Uç Hesaplama) tesislerinde mobil hizmet (oyun, sosyal medya, video akışı ve mesajlaşma) talebini tahmin etmek için ARIMAX (Autoregressive Integrated Moving Average with External Variables–Dış Değişkenlerle Otoresif Entegre Hareketli Ortalama) kullanılmıştır. Samulevicius vd. (2015) çalışmasında sınıflandırma temelli bir yaklaşım kullanılmış ve yüksek trafik ve düşük trafik olmak üzere iki sınıf tahmin edilerek planlama ve enerji optimizasyonu için etkinlik trafiğinin tahminini iyileştirmeye odaklanılmıştır.

“Doğrusal Olmayan İstatistiksel ve Makine Öğrenmesi Tabanlı Yöntemleri Kullanan Çalışmalar” aşağıdaki şekilde ifade edilmiştir. Bazı seriler, doğrusal tekniklerle modellenemeyecek kadar yüksek oranda karmaşıklık içerir. Mobil ağ trafiği tahmin

problemlerinde doğrusal olmayan kalıpları modellemek ve tahmin etmek için, makine öğrenmesine dayalı doğrusal olmayan modeller veya GARCH (Generalized Autoregressive Conditional Heteroskedasticity–Genelleştirilmiş Otoregresif Koşullu Heteroskedastisite) gibi koşullu varyansı modellemede başarılı olan yöntemlerin kullanıldığı ve başarılı sonuçlar elde edilen çalışmalar da mevcuttur. Yamada vd. (2018) çalışmasında merkezi izleme sistemlerine aktarılan trafik günlüğü miktarının azaltılması amacıyla trafik tahmin doğruluğundaki değişim araştırılmıştır. Bu amaçla, Gini kazançlarına bakılarak başarılı trafik tahmini yapabilmek için RF (Random Forest–Rastgele Orman) makine öğrenme yöntemi kullanılmış ve 6,5 ile 10,5 arasında değişen bir Alfa parametresi tahmin doğruluğu elde edilmiştir. Chmieliauskas ve Gursnys (2019) çalışmasında her endüstri ile ilgili tahmin kabiliyetinin ölçeklendirilmesi amacıyla zaman serisi tahmini için kolay kullanımlı API (Application Programming Interface–Uygulama Programlama Arayüzü) sağlayan Facebook Prophet kütüphanesinin verimliliğinin araştırılması amaçlanmıştır. 5 aylık günlük verilerle orta vadeli (sonraki 28 gün) ve kısa vadeli (sonraki 24 saat) tahminler yapılmış ve MAPE puanı %44,22 olmuştur. Bu da Prophet kullanımlarının trafikteki dalgalanmaları çok iyi karşılamadığı anlamına gelmektedir. Diğer çalışmalardan farklı olarak, bu çalışmada PRB (Physical Resource Block–Fiziksel Kaynak Bloğu) kullanımı ile trafik hacmi arasındaki korelasyon hakkında da araştırmalar yapılmıştır. Miao vd. (2016) çalışmasında hücresel ağın her bir mekansal-zamansal ölçeğinde hangi tahmin yönteminin uygun olduğunun analiz edilmesi için çalışılmıştır. Bu amaçla 340 3G hücresi üzerinde çalışılarak hücre seviyesi, baz istasyonu seviyesi ve RNC (Radio Network Controller–Radyo Ağ Denetleyicisi) seviyesi dahil olmak üzere farklı ağ ögesi seviyesinde kümelenmiş trafiğin tahmini için SARIMA, HMM (Hidden Markov Model–Gizli Markov Modeli) ve WNN (Wavelet Neural Networks–Dalgacık Sinir Ağları) kullanılmıştır. Ayrıca, bu yöntemlerin doğruluğu farklı kısa vadeli tahmin ufuklarında (15 dakika ila 60 dakika) kontrol edilmiştir. RNC seviyesinde trafik tahmini için WNN ve SARIMA'nın iyi performans gösterebileceği, baz istasyonu ve hücre seviyesinde ise HMM'nin farklı tahmin aralıklarında diğerlerinden daha iyi performans gösterebileceği tespit edilmiştir. Bireysel mobil trafiğin tahminini ve bunun üzerindeki mekansal etkiyi araştırmak için gerçekleştirilen Chen (2018) çalışmasında uzamsal bağlam bilgileri ve olaylar, makine öğrenimi modelleri gibi doğrusal olmayan modellerin performans tahminine göre %1-11 oranında iyileştirme gösterebileceği düşünülerek MC (Markov Chain–Markov Zinciri), PPM (Prediction by Partial Matching–Kısmî Eşleştirme ile Tahmin), SPM (Sampled Pattern Matching–Örneklenmiş Desen Eşleştirme) ve ALZ (Active

LeZi–Aktif LeZi) gibi Markov Modelleri, MLP (Multi-Layer Perceptron–Çok Katmanlı Algılayıcı) gibi farklı modellerle karşılaştırılmıştır ve %70 doğruluk sağlayan MLP’nin, %55-65 arasında doğruluk sağlayan Markov Modellerinden daha iyi performans gösterdiği görülmüştür. Anand vd. (2008) çalışmasında, internet trafiği varyansın sabit olmadığı yerlerde ani trafik değişimleri sergilendiğinden, bu dinamik koşullarda iyi bir trafik modellemesi yapmak için GARCH kullanılmıştır. Ayrıca bu çalışmada en iyi model siparişlerini seçmek için AIC (Akaike Information Criterion–Akaike Bilgi Kriteri) ve BIC (Bayesian Information Criterion–Bayes Bilgi Kriteri) kullanılmış ve MLE (Maximum Likelihood Estimation–En Büyük Olabilirlik Kestirimi) ile optimum parametre değeri seçimi yapılmıştır. Kim vd. (2020) çalışmasında, ARIMA ve GARCH gibi diğer istatistiksel yöntemlere göre avantajlarını ve kıyaslamalar için LSTM (Long Short-Term Memory–Uzun Ömürlü Kısa Süreli Bellek) gibi sezgisel yöntemler analiz edilmiştir. Sonraki 20 dakikada tahmin doğruluğunu test etmek için pasif monitörlerden 45 dakika boyunca toplanan verilerle ilgili olarak, INGARCH (Integer-Valued GARCH–Tamsayı Değerli GARCH)’ın, GARCH ve LSTM’e göre trafik serilerinin özelliklerini yakalamada daha iyi sonuçlar ürettiği görülmüştür.

Derin Öğrenme Dahil Yapay Sinir Ağları Tabanlı Yöntemleri Kullanan Çalışmalar aşağıdaki şekilde ifade edilmiştir. Chen vd. (2017) çalışmasında trafik tahminlerinin tamamlayıcılığının kümelenmesine ilişkin aynı BBU (Base-Band Unit – Baz-Bant Birimi) havuzu altında RRH (Remote Radio Head – Uzaktan Radyo Başlığı)’leri organize ederek C-RAN (Cloud RAN–Bulut RAN) dağıtımını ve enerji tüketimi maliyetlerini azaltmak için ARIMA, WANN ve MuLSTM (Multivariable LSTM – Çok Değişkenli LSTM) yöntemleri, kodlayıcı ve kod çözücü olarak iki LSTM katmanı ile karşılaştırılmıştır. ARIMA, 182 RRH’nin saatlik olarak 61 gün boyunca Telekom Italia’dan toplanan verilere ilişkin kısa vadeli tahminlerinde, örüntülerin temel özelliklerini öğrenmiş, ancak uzamsal korelasyonları dikkate almamıştır. WANN, tek RRH özelliklerini içeren gizli kalıpları öğrenmede ARIMA’ya kıyasla çok daha iyi bir performans göstermiştir, ancak özellikle Cuma ve Cumartesi günleri için istikrarlı sonuçlar elde edememiştir. MuLSTM, 1 üzerinden 0,074 MAE puanına erişirken ARIMA ve WANN için bu değer sırasıyla sırasıyla 0,202 ve 0,175’tir. Çalışma çıktıları, sonuçların genel C-RAN yatırım maliyetlerinde yaklaşık %50 azaltma gösterdiğini ifade etmektedir. Wang vd. (2017) çalışmasında, uzamsal korelasyonları modellemek için, SAE (Stacked Auto-Encoders–Yığılmış Otomatik Kodlayıcılar) kullanılmış ve diğer hücrelerin hem genel hem de yerel

özellikleri dikkate alınmıştır. Otomatik kodlama katmanlarının çıktılarını ve tahmin hücrelerinin geçmiş örneklerini alan LSTM birimleriyle elde edilen zamansal modellere karşılık baz model olarak ARIMA ve SVM (Support Vector Machine–Destek Vektör Makinesi) kullanılmıştır. Her iki model de genel ortalama özellikleri öğrenirken, SVM spesifik varyansı dikkate almakta daha iyi bir performans göstermiştir. Çözümlerin MSE, MAE ve Logaritmik Kayıp (Log Loss) sonuçları sırasıyla 0,042, 0,165 ve 0,583 olarak ortaya çıkmış ve temel modellere kıyasla yaklaşık %30 iyileşme elde edilmiştir. Zhang ve Patras (2018) çalışmasında ARIMA, HW-ETS ve MLP’ye kıyasla ConvLSTM (Convolutional LSTM–Evrişimsel LSTM) ve D-STN (Deep Spatio-Temporal Networks–Derin Uzay-Zamansal Ağlar) kullanarak trafik tahmininin doğruluğu artırılmaya çalışılmıştır. D-STN yaklaşımı, NRMSE açısından temel modellere kıyasla %61 daha doğru sonuç ortaya koymuştur. Zheng vd. (2020) çalışmasında, özel bir gizli katmanlı FFNN (Feed Forward Neural Networks–İleri Beslemeli Sinir Ağları) olan ve çok hızlı eğitilebilen kELM (Kernel Extreme Machine Learning–Çekirdek Ekstrem Makine Öğrenmesi) ağları ile yeni bir yöntem önerilmiştir. Bu çalışmada Gauss, polinom, doğrusal ve Sigmoid olmak üzere 4 fonksiyon kullanılmış ve optimizasyon için PASO (Particle Swarm Optimizer – Parçacık Sürüsü Eniyileyici), MVO (Multiverse Optimizer – Çoklu Evren Eniyileyici) ve MFO (Moth-Flame Optimizer – Moth-Flame Eniyileyici) kullanılmıştır. MFO ile kELM, vSVR’dan (v-Support Vector Regressor – vDestekli Vektör Bağlanım) yaklaşık 9 kat daha hızlı eğitilmiş ve %11,15 MAPE skoru ile aynı tatmin edici doğruluk düzeyini sunmuştur. Lopez vd. (2019) çalışması, sinir ağları yaklaşımından yararlanmak için gradyan artırmaya dayalı toplu yaklaşım içeren gaNET (Generalized Additive Networks – Genelleştirilmiş Katkı Ağları) adlı yeni bir sinir ağı tabanlı model önermiştir. Farklı gaNET mimarileri, Rastgele Orman, Doğrusal Bağlanım, LSTM gibi farklı RNN mimarileri ve seq2seq modelleri gibi makine öğrenimi modelleriyle karşılaştırılmıştır. Spesifik gaNET mimarileri MSE, R-kare ve NRMSD’de (Normalized Root Mean Square Error Deviation – Normalleştirilmiş Kök Ortalama Kare Hata Sapması) diğerlerinden daha iyi performans gösterirken seq2seq modelleri MAE ve MAD (Mean Absolute Deviation – Ortalama Mutlak Sapma) değerlerinde daha iyi bir sonuç göstermiştir. Park ve Woo (2009) çalışmasında doğrusal olmayan trafik serilerine yaklaşma avantajlarını kullanmak için BLRNN (BiLinear Recurrent Neural Networks – Çift Doğrusal Tekrarlayan Sinir Ağları) uygulanmış ve dinamik öğrenmeyle, DBLRNN’nin (Dynamic BLRNN – Dinamik BLRNN) daha avantajlı olduğu görülmüştür. NMSE değerlerine bakıldığında DBLRNN, gelecek 100 saniyelik ethernet

trafiğini tahmin etmede MLP ve PRNN'den (Primitives with Recurrent Neural Networks – Tekrarlayan Sinir Ağlarına Sahip İlkeller) daha iyi performans göstermiştir. Dalgitsis vd. (2018) çalışmasında, klasik zaman serileri ve makine öğrenimi yöntemleriyle LSTM ağlarının kapsamlı bir karşılaştırması ele alınmıştır. LSTM'ler, aktivasyon fonksiyonlarına göre ağırlıkları hatırlama ve unutma yeteneği sayesinde, serinin geri kalanı ile korelasyona göre ağırlıklarını unutabilir ve yeniden değerlendirebilir. Bu yetenek, LSTM'leri daha çok yönlü ve hatalara, gürültüye ve örnek boşluklarına karşı dirençli hale getirir. LTE baz istasyonu trafiğini milisaniye cinsinden toplayan Trinh vd. (2018) çalışmasında kaynak optimizasyonu amacıyla 15 adım ilerisi tahmin edilmiştir. Bu çalışmada LSTM tahminleri ARIMA'ya kıyasla NRMSE açısından 5 kat daha iyi sonuç göstermiştir. Andreoletti vd. (2019) DCRNN (Diffusion Convolutional Recurrent Neural Network – Difüzyon Evrişimli Tekrarlayan Sinir Ağı) kullanılarak yeni grafik modelleme ile ilişkisel korelasyonlar üzerinde çalışılmıştır. Çözümler 12 düğüm ve birbirleri arasında 30 bağlantı içeren gerçek omurga ağ verileri üzerinde LSTM ve FCNN ile karşılaştırılmıştır. Hem LSTM hem de FCNN için %200'ün üzerinde olan MAPE değeri DCRNN için %43,2 olarak elde edilmiştir. Karşılaştırmalı olarak LSTM ve ARIMA ile uygulama düzeyinde trafik tahmini yapan Azari vd. (2019) çalışmasında, derin öğrenmeye dayalı yöntemler kullanılmıştır. Wang vd. (2018) çalışmasında, metropollerde uzun mesafe mekansal ilişkiler, kullanıcıların hareketliliği, özellikle 3 km'nin üzerindeki mesafelerden yarım saatte gelen toplu taşıma hareketleri kullanılarak doğruluğun artırılması için GNN (Graphed Neural Networks – Grafikli Sinir Ağları) ve LSTM üzerine bir uzay-zamansal model oluşturulmuştur. Ancak mekansal ilişkilerin zamansal modellemesi için, seriler eğilim ve mevsimsellik olarak bileşenlerine ayrıştırılmıştır. Cui vd. (2015) çalışmasında, bir Çin operatörünün xDR dosyalarından çıkarılan ve boyut olarak çok büyük olduğu için Hadoop ile işlenen trafiği tahmin etmek için ESN (Echo State Networks – Yankı Durum Ağları) kullanılmıştır. Tahmin doğruluğu, sadece bir gün öncesindeki verilerin kullanılmasına rağmen, NRMSE ve MAE açısından tatmin edici olarak elde edilmiştir. Uzun vadeli tahmin için Telekom Italia veri kümesi üzerinde gerçekleştirilen Jaffry (2020) çalışması, FFNN ve ARIMA'ya kıyasla LSTM'yi kullanmıştır. Burada LSTM, MAE açısından diğerlerinden daha iyi performans göstermiş ve eğitim süresini azaltmıştır. Feng vd. (2018) çalışması, uzaysal korelasyonları kullanmak için bir özellik çıkarıcıdan ve trafik patlaması gibi uzun vadeli geçici karmaşık özellikleri kapsamak içinse seq2seq modelleyiciden oluşan kendi çok katmanlı sinir mimarisini geliştirmiştir. DeepTP olarak adlandırılan ve içerisinde dahili olarak LSTM

kullanılan bu modelin diğer modellerle kıyaslanmasında ARIMA, SVM ve RNN'den yaklaşık %12 daha iyi sonuç elde edilmiştir. Sevgican vd. (2020) çalışmasında, sentetik olarak oluşturulmuş ve GitHub'da paylaşılan veriler üzerinde, tek tek hücrelerdeki farklı abone grupları için dilim trafiğinin yük performansını tahmin etmek için NWDAF'in (Network Data Analytics Function – Ağ Veri Analitiği İşlevleri) bir alt işlevi olarak LSTM ve RNN tabanlı yöntemler kullanılmıştır. Peng ve Yuan (2008) çalışmasında BPNN (Back-Propagation Neural Network–Geri-Yayılmalı Sinir Ağları) için bir dalgacık geliştirmesi oluşturulmuş ve BPNN'nin yavaş ve yerel yakınsama sorunlarını iyileştirmek için BPWNN (Back-Propagation Wavelet Neural Network–Geri-Yayılmalı Dalgacık Sinir Ağları) mimarisi sağlanmış ve MSE açısından, doğruluk %50 artırılmıştır. Junsong vd. (2009) çalışması, Elman Sinir Ağları'na dayanan ve tahminini doğrusal modeller ve ANN mimarîsi ile karşılaştıran bir çalışmadır. Gowrishankar ve Satyanarayana (2009) çalışmasında, kısa vadeli tahmin çözümü olarak sonraki saniye ve dakikalar için erişim noktası düzeyinde trafik tahmininde RRBFN (Recurrent Radial Basis Neural Networks – Tekrarlayan Radyal Temel Sinir Ağları), ESN ve FARIMA (Fractionally Autoregressive Integrated Moving Average – Kesirli Otoregresif Entegre Hareketli Ortalama) karşılaştırılmıştır. ESN, RRBFN'ye kıyasla biraz daha iyi sonuç üretmiştir ancak daha fazla eğitim örneği ve daha uzun eğitim süresi gerektirir. Her iki ANN tabanlı yaklaşım yaklaşık %97 doğruluk elde ederken, temel model olan FARIMA'nın doğruluk düzeyi %80 civarlarında kalmıştır. Cortez vd. (2006) çalışmasında ANN'lerin TCP-IP (Transmission Control Protocol - Internet Protocol – İletim Kontrol Protokolü - İnternet Protokolü) trafik tahmini, iki büyük ISP (Internet Service Provider–İnternet Servis Sağlayıcısı) trafik serisinde HW-ETS ve ARIMA'ya karşı kıyaslanmıştır. ANN'ler, serilerin doğrusal olmayan özelliklerini modellemede daha yüksek başarı ortaya çıkarmıştır. Cortez vd. (2008) çalışmasında, benzer bir konu daha ayrıntılı olarak ele alınmış ancak geçmiş kalıplar üzerinde basit tek değişkenli model olan SLNN (Single-Link NN–Tekil Bağlantılı Sinir Ağı), tümünün geçmiş değerlerini içeren ADNN (All Direct Neighbor Link–Tüm Doğrudan Komşu Bağlantısı) gibi bazı farklı sinir ağları karşılaştırılmıştır. Yüksel vd. (2019) çalışmalarında, bazı testlerle özellikleri tanımlamak ve parametreleri otomatik olarak tahmin etmek için bir otomatik makine öğrenme çözümü olarak MLP için otomatikleştirilmiş mimari tanımlama ve hiperparametre ayarı için bir yazılım geliştirmek üzere çalışılmıştır.

Hibrit yöntemler kullanan çalışmalar aşağıdaki şekilde ifade edilmiştir. Bazı çalışmalar, hedeflenen sorunların ayrı bölümlerini çözen birden fazla tekniği bir araya getiren hibrit yaklaşımlar önermiştir. Bu amaçla genellikle zaman serisi teknikleriyle birlikte makine öğrenimi yöntemleri veya derin sinir ağları kullanılmıştır. Xu vd. (2018) çalışmasında, tatil etkisi tahmini için baz istasyonlarının ayrıştırılmış tatil davranışları ve Rastgele Ormanın ayrıştırılmış tatil davranışları üzerinden k-ortalamları kümelenmesi birleştirilmiştir. BS verileri tatil kalıntılarına göre kümelenmiştir. Bu sayede küme başına belirli Rastgele Orman tahmin modellerini eğitmek için yetersiz veri sorunu çözülmüştür. Tahmin edilen tatil artıklarının tahmin edilen eğilim ve mevsimsel bileşenlerinin üstüne eklendiği çalışmada Facebook Prophet'e kıyasla daha iyi APE (Absolute Percentage Error – Mutlak Yüzde Hatası) değeri alınmış ve tatil trafiği 15 gün önceden tahmin edilmiştir. Sonuçlarda veri noktalarının %84'ü için hata oranı Prophet'te %50 iken yeni yöntemde %20'nin altında bulunmuştur. Xiang vd. (2010) çalışmasında, WLAN trafiğindeki patlama nedeniyle oluşan yüksek değişkenlik sorunu çözülmüştür. CO (Covariation Orthogonal Prediction – Kovaryasyon Ortogonal Tahmin), BPNN ile seri olarak birleştirilmiş ve WLAN trafiği CE (Cumulative Error – Kümülatif Hata) ve ACE (Absolute Cumulative Error – Mutlak Kümülatif Hata) kullanılarak tahmin edilmiştir. Sonuç olarak birleştirilmiş çözümün CO doğrusal tahmininden daha iyi performans gösterdiği görülmüştür. Au vd. (2011) çalışmasında, otomatikleştirilmiş bir DSARIMA (Double Seasonal ARIMA – Çift Mevsimsel ARIMA) algoritması geliştirerek ve DSEXP (Double Seasonal Exponential Smoothing – Çift Mevsimsel Üstel Düzeltme) uygulayarak mobil arama trafiğinin iki mevsimlik özellikleri araştırılmış, hem modelin tahmin çıktıları üzerinde hem de basit ortalama, OLS (Ordinary Least Square – Sıradan En Küçük Kareler), ERLS (Equality Restricted Least Squares – Eşitlik Kısıtlı En Küçük Kareler) ve ENT (Entropy Theory – Entropi Teorisi) ile kombinasyonları üzerinde deneysel analizler yapılmıştır. DSARIMA ile DSEXP'ten daha başarılı bir şekilde 973 BS'den 20 haftalık saatlik veri kullanılarak 7,32 MAPE ile iki hafta öncesinde yoğun saat periyodu çağrı bağlantı hacmine ulaşılmıştır. Nikravesht vd. (2016) çalışmasında, STL (Seasonal and Trend Decomposition by Using Loess – Loess Kullanarak Mevsimsellik ve Eğilim Ayırıştırma) tekniği kullanılarak, serisini mevsimsel ve trend ayırıştırma ile ayırıştırarak trafik tahmini için aktif kullanıcı sayısını tahmin etmek için ticari MNO verileri üzerinde çalışılmıştır. Bu çalışmada bir hücre üzerindeki verilerle birkaç gün öncesindeki eğitim 4 günlük saatlik verilerle karşılaştırılmış ve hibrit model 8,56 MAE ve 11,45 RMSE sağlayarak üç ayrı modelden daha iyi performans göstermiştir.

Zhang vd. (2019) çalışmasında hem uzamsal hem de zamansal korelasyonları yakalayan ve eksikleri olduğu tespit edilen Doğrusal Bağlanım, SVR, klasik LSTM ve DenseNet ile kıyaslama için ConvLSTM ve CNN'e (Convolutional Neural Network–Evrişimsel Sinir Ağları) dayalı bir derin öğrenme modeli geliştirmiştir. Bu çalışmada hücreler k-ortalama ile kümelenmiş ve yeni modelin çok daha kolay ve hızlı bir şekilde eğitilmesi sağlanmıştır. Zang vd. (2015) çalışmasında, farklı küme boyutu boyutları için, 6 günlük saatlik geçmiş verileri kullanarak sonraki 24 saatlik veri tahmin edilmeye çalışılırken, Elman Sinir Ağları ile eğitmek için küme seviyesindeki serilere uygulanan WT (Wavelet Transform–Dalgacık Dönüşümü) ve küme baz istasyonlarına uzaklığa dayalı k-ortalama kümelemesi incelenmiştir. Bu hibrit çözümün, FFNN ve kapsamlı algılama yöntemlerinden daha iyi performans gösterdiği tespit edilmiştir. Katris ve Daskalaki (2015) çalışmasında, sinir ağları ile zaman serisi yöntemleri (FARIMA ve ANN) birleştirilmiş ve bu hibrit yöntem her iki yöntemin ayrıık versiyonlarıyla karşılaştırılmıştır. Bu birleşim temelde basit ortalama, ağırlıklı ortalama, doğrusal bağlanım veya sağlanan verilerde doğrusal olup olmadığını kontrol eden sinir ağı testi ile yapılmıştır. Bu teste göre doğrusal olmama durumunda ANN, doğrusal olma durumunda ise FARIMA daha iyi çözüm olarak değerlendirilmiştir. Zhang vd. (2017) çalışmasında tahminleri geliştirmek için bir Rastgele Orman tabanlı makine öğrenimi hattı oluşturulmuş bu hat üzerinde SARIMA, Karar Ağacı ve Bağlanım) gibi farklı modeller kullanılmıştır. Bu şekilde, model ortalama RMSE (Root Mean Square Error – Kök Ortalama Kare hata) puanını HMM (Hidden Markov Model – Gizli Markov Modeli), VAR (Vector Auto Regression – Vektör Otomatik Bağlanım), ST-MVL (Spatio-Temporal Multi-View-based Learning – Uzay-Zamansal Çoklu Görünüm Tabanlı Öğrenme), SARIMA gibi kıyaslama yöntemlerine kıyasla %40 artırmıştır. Zhou vd. (2006) çalışmasında, trafikteki patlamalar gibi koşullu varyanslı trafik serilerinin genel özelliklerini göz önünde bulundurularak LAN (Local Area Network – Yerel Ağ) trafiğini tahmin etmek için ARIMA ve GARCH'ı birleştirerek bir başka hibrit model geliştirilmiştir. Dawoud vd. (2014) çalışmasında, enerji tasarrufu amacıyla baz istasyonu trafiğini tahmin etmek için, hibrit bir çözüm geliştirmiştir.

Literatürde gerçekleştirilen bu çalışmalar, kullanılan yöntemler, mevsimsellik, tahmine konu olan ağ kapsamı, veri seti, amaç, değerlendirme metriği ve doğruluk sonucu gibi kriterlerine göre toplu olarak Tablo 2.1'de verilmiştir.

Tablo 2.1. Ağ Trafikinin Tahminlemede Kullanılan Yapay Zeka Yöntemleri Konulu Literatür Çalışmaları

Çalışma	Yöntemler	Mevsimsel	Amaç	Ağ Kapsamı	Veri Seti	Değerlendirme Metriği	Doğruluk Sonucu
DOĞRUSAL ZAMAN SERİSİ YÖNTEMLERİ							
Zhou vd. (2012)	ENT	E	Zamansal, mekansal ve hizmet tabanlı bilgilerin ses ve metin veri tahmini üzerindeki etkisini analiz	Baz istasyonu toplu trafik	7000 baz istasyonu-3 milyon abone-1 aylık trafik verisi	CDF	N/A
Jung vd. (2006)	AR, MA, ARMA, ARIMA	H	Tüm intranet için ethernet'ten toplam paket verilerini tahmin etme	Tüm ağ trafiği	İntranet 1 yıllık saatlik ethernet paketleri (günlük, haftalık, aylık)	N/A	N/A
Yu vd. (2010)	SARIMA	E	Ötümüzdeki 6 aya kadar günlük ortalama trafiği tahmin etme	Alt-bölge toplu trafik	Çin-Heilongjiang'da hizmet veren MNO'nun 4 yıllık ortalama günlük mobil trafiği	MAPE	%1,38-%6
Miao vd. (2014)	SARIMA	E	Çarpımsal SARIMA'nın farklı hacimler için BS'lerin saatlik trafiğine iyi uyup uymadığını gösterme	Baz istasyonu ve BSC toplu trafik	Çin'in iki eyaletindeki 4200 Baz İstasyonundan 38 Gün için saatlik 2G / 3G verileri	MAPE	%2,2-%7,5
Tikunov ve Nishimura (2007)	HW-ETS	E	Arama tıkanıklıklarını azaltmak için Erlang'da kısa vadeli CS arama trafiğini tahmin etme	Hücre düzeyinde trafik	77 GSM hücresinden 25 günlük saatlik çağrı trafiği (18 eğitim + 7 test)	NRMSE	0,12-0,28
Jia vd. (2009)	SARIMA	E	Tıkanıklık uyarısı ve planlaması için kısa vadeli tahmin yapma	Hücre düzeyinde trafik	Hwlongjiang'daki 269 hücreden 35 günlük saatlik çağrı trafiği (28 eğitim + 7 test)	NRMSE	0,10-0,26
Cao vd. (2018)	ARMA, HW-ETS	E	Entropi ağırlık yöntemi ile belirlenen sıcak nokta alanlarının trafiğini tahmin etme	Bölge düzeyi trafiği	5 gün boyunca 80360 baz istasyonundan 15 dakikalık aralıklı trafik verisi	MAPE	%53,46, %13,40
Sciancalepore vd. (2017)	WF, SARIMA	E	Giriş serilerini db4 dalgacık filtreleriyle denoize ederek SARIMA tahmin doğruluğunu iyileştirme (% 20 sıkıştırma)	Tüm LAN ağ trafiği	Karaçi Üniversitesi'nde, 5 gün veri üretmek için kullanılan yüksek hızlı fiber LAN ağı istatistikleri	SSE, R-kare, AIC	689,98, 0,912, 6,742
Syed vd. (2010)	TSD, ARIMA	E	Ayrıştırılmış modelleme ile omurga yönlendirici ağ trafiği tahmininin ARIMA modelinden daha iyi performans gösterdiğini kanıtlama	Bir omurga yönlendirici trafiği	CERNET01'in 11 günlük saatlik yönlendirici trafiği, günde 121 gün CERNET02 ve haftada 5 yıllık NSFNET ağları	MSE	6,81, 209,31
Jiang vd. (2015)	FFT	E	Doğruluk iyileştirmesi için uzamsal-zamansal modelleme için bir yaklaşım sağlama	Baz istasyonu trafiği	Çin'de 5 dakikalık aralıklarla 185 baz istasyonundan alınan 20 günlük veri	MSE	%5,50
Li vd. (2009)	AARIMA, ARIMA	H	ARIMA doğruluğunu, birinci seviye farkına dayalı ek bir geri bildirim bileşeni ile iyileştirme	Tüm ağ trafiği	% 80-20 eğitim ve test bölümü ile 4 genel etki alanı internet trafiği	MAE	44, 211, 9, 73, 234, 15
Guang vd. (2015)	ARMAX	E	MEC'de mobil hizmet taleplerini tahmin etme, ARMAX'ı uzamsal korelasyon ile kullanmayı kolaylaştırma	MEC tesis düzeyinde trafik	Bir metropol bölgesindeki 123 MEC tesisinden 5 dakika aralıklarla toplanan 3G / 4G mobil hizmet trafiği	RE, AE	0,06, 0,000006
El Hag ve Sharif (2007)	Sınıflandırma	H	Olaylar veritabanındaki olayları ilişkilendirerek planlanan olay hücreleri trafik seviyesini (yüksek, düşük) tahmin etme	Hücre düzeyi trafiği	Danimarka mobil ağı, 45 gün için 660 hücre saatlik trafik verisi	Sınıflandırma	%48'dan %100'e
Zhang vd. (2019)	WT, PCA, TomoG	E	Mevsimselliği ve eğilimi modellemek için Dalgacık ayrıştırmayı kullanma	Yönlendirici düzeyi trafiği	Üniversite kampüsünde 5 dakika aralıklarla 3 günlük 6 Yönlendirici verileri	MSE	0,8, 0,15
Zang vd. (2015)	WT, ARIMA	E	Düzenli dalgacık dönüşümü ile klasik tahmin yöntemlerinin kombinasyonu	Tüm ağ trafiği	Üniversite kampüsünden 5 dakikalık aralıklarla 13 günlük yönlendirici veri trafiği	N/A	N/A
Katris C, Daskalaki (2015)	HW-ETS	E	Kullanıcı mobilite modelini dikkate alarak dilim trafik talebini tahmin ederek 5G ağ dilimlerini ayarlama	Dilim düzeyi trafiği	Matlab simülasyonu, 7 baz istasyonu, 10-30 istemci ve istemci başına 100 kullanıcı şeklindeki veri seti	N/A	N/A
DOĞRUSAL OLMAYAN İSTATİSTİKSEL YÖNTEMLER VE MAKİNE ÖĞRENMESİ TABANLI YÖNTEMLER							
Box ve Jenkins (1970)	RF	H	Transfer için gerekli trafik günlük miktarını azaltmak için trafik tahmininin özellik önemini bulma	Baz istasyonu düzeyi trafiği	10 dakikalık bir ilden. 100 BS'de 50 günlük aralıklarla Telecom Italia Big Data Challenge verileri	RMSE	6,5-10,5

Çalışma	Yöntemler	Mevsimsel	Amaç	Ağ Kapsamı	Veri Seti	Değerlendirme Metriği	Doğruluk Sonucu
Ntalampiras ve Fiore (2018)	Prophet	E	Facebook Prophet Library'nin mobil trafik tahmini için araştırma doğruluğunun analizi	Hücre düzeyi trafiği	141 LTE 1800 MHz, 237 LTE 800 MHz hücrelerden 180 günlük günlük trafik verileri ve 30 günlük saatlik veriler	R-kare, MAPE	%44,22
Samulevicius vd. (2015)	SARIMA, HMM, WNN	E	Hücresel ağı her bir mekansal-zamansal ölçeğinde uygun tahmin yöntemini analiz etme	Hücre, BS, RNC düzeyi trafiği	China Mobile ağında 340 3G hücresinden 15 dakikalık aralıklarla 3 haftalık veri	SMAPE	1,04, 0,86, 0,169
Yamada vd. (2018)	Markov Yöntemleri, MC, PPM, SPM, ALZ, MLP	E	Bireysel mobil trafiğin tahminini ve bunun üzerindeki mekansal etkiyi araştırmak için bir rapor hazırlama	Tüm ağ trafiği ve bireysel abone trafiği	Büyük bir şehirdeki bir MNO'dan 1 yıllık CDR ve oturum kayıtlarına sahip iki veri kümesi	MAPE	%70, %67, %55- %65%
Chmieliauskas ve Gursnys (2019)	GARCH, ARCH, ARIMA	E	GARCH ve yoğun kuyruklu dağıtım kullanarak iyi bir trafik modeli oluşturma	Tüm ağ trafiği	USA Tier-1 ISP'nin 5 dakikalık aralıklarla 24 saat internet omurgası trafiği	RTO	N/A
Miao vd. (2016)	INGARCH, GARCH, ARIMA, LSTM	E	Trafiğin özelliklerini yakalamak için INGARCH'ı gösterme	Tüm ağ trafiği	CAIDA'nın 130.000 UTC'den toplanan 4 günlük dakika taneciklik pasif izleme verileri	NMAE, NMSE	0,024, 0,0453
DERİN ÖĞRENME DAHİL YSA TABANLI YÖNTEMLER							
Chen (2018)	MuLSTM, ARIMA, WANN	H	RRH trafiğini tahmin ederek ve bunları mesafe ve entropi temelli tamamlayıcı özelliklere göre bir araya getirerek şehirlerdeki BBU havuzlarının dağıtım maliyet ve enerjisini azaltma	Hücre düzeyi trafiği	İtalya'daki 182 hücre koordinatlarına eşlenen 61 günlük (% 70 eğitim +% 30 test) saatlik trafik hacmi içeren Telekom Italia Big Data Challenge veri kümesi	MAE	0,074-0,083
Anand vd. (2008)	SAE, LSTM	E	Gelişmiş trafik tahmin doğruluğu için uzay-zamansal modellemenin önemini gösterme	Hücre düzeyi trafiği	China Mobile 2844 baz istasyonu için koordinatlarıyla birlikte 4 aylık saatlik veriler	MSE, MAE	0,042, 0,165, 0,583
Kim (2020)	ConvLSTM, 3D-ConvNet, D-STN, HW-ETS, ARIMA, MLP	H	Kaynak yönetimi için Deep Spatio-Temporal Networks ile trafik talep tahmini	Hücre düzeyi trafiği	60 gün saatlik Telecom Italia Big Data Challenge veri kümesi	NRMSE	0,71
Chen vd. (2017)	kELM, vSVR	E	RNN ve SVR ile karşılaştırıldığında doğruluğunu ve eğitim hızını göstermek için kELM tabanlı trafik tahmini	Konuma dayalı trafik	Çin'in Huainan kentindeki ticari bir caddeye ait özel kutlama günü dahil 163 günlük saatlik trafik verileri	MAPE	%11,15
Huang ve Chen (2019)	RF, Seq2Seq, LSTM	E	Gradyan artırma kullanılarak kaldırılan artık bir sinir ağı mimarisi önerme	Cihaz düzeyi trafiği	6214 cihazdan toplanan 26 günlük saatlik veri trafiği	MSE, MAE, R-kare, MAD,	3,0217, 0,8, 0,1478, 0,7061, 9,94
Wang vd. (2017)	D-BLNN, BLNN, MLPNN	E	BLNN'yi Dinamik BLNN olarak tanıtarak dinamik öğrenme kontrolü	Tüm ağ trafiği	1000 saniyelik geçmiş BellCore ethernet trafik verileri	NMSE	1,2
Zhang ve Patra (2018)	LSTM, SVM, ARIMA, SARIMAX	E	RNN-klasik zaman serisi yöntemleri ve makine öğrenimi modeli karşılaştırma	Hücre düzeyi trafiği	Vodafone ağından toplanan 573 baz istasyonunun 4 aylık günlük istatistikleri	MSE	1,68, 3,24, 6,53, 11,26
Zheng vd. (2020)	LSTM, FFNN, ARIMA	E	Sonraki milisaniyeler için kaynak ayarlamaları için bir LTE baz istasyonu trafiğini tahmin etme	Hücre düzeyi trafiği	LTE baz istasyonundan milisaniye ayrıntı düzeyinde toplanan 4 haftalık veri	NRMSE	0,01, 0,05, 0,21
Lopez vd. (2019)	DCRNN, LSTM, FCNN	E	DCRNN kullanarak trafiği tahmin etme	Düğüm düzeyi trafiği	5 dakika aralıklarla 6 ay boyunca 30 bağlantıda 12 düğümden toplanan gerçek omurga ağı	MAPE	%43,2, %210, %220
Park ve Woo (2009)	LSTM, ARIMA	E	Derin öğrenme modellerinin performansını akılcıca seçilmiş özellikler ve parametrelerle doğrusal modellerle karşılaştırma	Tüm ağ uygulama trafiği	Ağ koklama araçlarından 24 gün boyunca toplanan 10 saniyelik uygulama paketi trafiği	RMSE	60, 62
Dalgakis vd. (2018)	GNN, LSTM, ARIMA, TSD	E	1,5 milyon aboneye ve 30 dakikada 5929 baz istasyonuna sahip bir Çin şehrinde 13 günlük arama izleme verilerini gösterme	Hücre düzeyi trafiği	Kullanıcı mobilitesi ve hücre içi ve hücre içi trafik ayrıştırması ile eğitim trafiği	MAE, MAPE	1,051, 0,79
Trinh vd. (2018)	ESN	E	Hadoop ve ESN kullanarak trafik tahmini için büyük xDR verilerini paralel şekilde işleme	Tüm ağ düzeyinde uygulama trafiği	Bir Çin büyük şehrinin 6 günlük xDR verileri	NRMSE, RMSE, MAE	0,1263, 0,0605, 0,0440
Andreoletti vd. (2019)	LSTM, FFNN, ARIMA	E	LSTM ile gelecekteki trafiği tahmin etme ve klasik FFNN ve ARIMA ile karşılaştırma	Tüm ağ trafiği	60 gün saatlik Telecom Italia Big Data Challenge veri kümesi	MAE	0,10
Azari vd. (2019)	LSTM + Seq2Seq, ARIMA, SVM, GRU	E	Çok katmanlı DL mimarisi ile mekansal bağımlılıkların ve harici bilgilerin tahmini	Rastgele seçilen 15 baz istasyonu trafiği	Şangay, Çin'deki 1 milyon kullanıcı ile 9600 baz istasyonundan 1 aylık 10 dakikalık aralıklı trafik verileri	NRMSE	0,06

Çalışma	Yöntemler	Mevsimsel	Amaç	Ağ Kapsamı	Veri Seti	Değerlendirme Metriği	Doğruluk Sonucu
Wang vd. (2018)	RNN, LSTM	E	Ağ dilim yükünü 5G NWDAF'ın dahili bir işlevi olarak tahmin etme	Hücre düzeyinde abone kategorisi trafiği	5 RRU, 3 kullanıcı kategorisi ve 5 cihaz türü için 15 dk aralıklarla 6 aylık veriler	MAPE, MAE	0,544, 0,202
Cui vd. (2015)	BPWNN, BPNN	E	BPWNN ile, adaptif öğrenmeyle yavaş ve yerel yakınsamasının üstesinden gelmek için öngörütüde bulunma	Tüm ağ trafiği	15 günlük saatlik gerçek ağ trafiği	MSE	0,0051, 0,0121
Jaffry (2020)	Elman Ağları, ARIMA, FARIMA, ANN	E	Elman Sinir Ağları ile ağ trafiğini geçmiş verileri dikkate alarak tahmin etme	Tüm ağ trafiği	Simulasyon verisi	MAE, MSE, NMSE	11,669, 230,59, 0,65
Feng vd. (2018)	RRBFN, ESN, FARIMA	E	Avantajları karşılaştırmak için RBFNN, ESN ve FARIMA ile kablosuz ağ trafiğini modelleme	Cihaz düzeyi trafiği	476 erişim noktasından saniye ve dakikalar için toplanan CRAWDAD deposundan 11 haftalık izleme verileri	MAPE, MSE	%97,8, 0,0125
Sevgican vd. (2020)	ANN, HW-ETS, ARIMA	E	İki büyük ISP'nin TCP-IP trafik tahmininde YSA'nın doğruluk performansını değerlendirme	Tüm ağ trafiği	İki büyük ISS'den gelen 5 dakikalık ve saatlik toplanmış 1 haftalık TCP-IP trafik verileri	SSE, MAPE	%12-23, %3-5
Nie vd. (2018)	CNN, RNN	E	DQN karar verme ve trafik talebi tahminini uygulayarak enerjiye duyarlı bir mobil trafik boşaltma planı önerme	Tüm ağ trafiği	61 günlük (55 günlük eğitim + 6 günlük test) 10 dakikalık aralıklarla CDR verilerini içeren Telecom Italia veri seti	N/A	N/A
Feng vd. (2016)	FFNN, ARIMA	E	FFNN ve ARIMA türleriyle trafik özelliklerini karşılaştırma	Tüm ağ trafiği	Sırasıyla saatlik 365 gün ve ikinci olarak 2 günlük gözlemlere sahip iki izleme veri kümesi	N/A	N/A
Peng ve Yuan (2008)	SLNN, ADNN, DANN, HW-ETS	E	Bağlantıların TCP / IP trafiğinde farklı NN mimarisini birbirleriyle ve HW-ETS yöntemleriyle karşılaştırma	Bağlantı düzeyi trafiği	UKERNA'nın 39 bağlantısından toplanan saatlik veriler	RRMSE	19,7, 19,4, 15,4
Junsong vd. (2009)	MLP	E	MLP sinir ağı için otomatik hiper parametre ayarı ve mimari seçimi yaparak otomatik bir tahmin yazılımı geliştirme	Tüm ağ trafiği	5 dakika ve saatlik ayrıntı düzeyinde iki ISS'den toplanan 4 veri kümesi	MAPE	3,24, 4,30
Zhang vd. (2017)	ESN, LSTM, CNN, SARIMA	E	ESN kullanarak uzun ve kısa vadede hızlı ve doğru bir iş hacmi ve paket oranı tahmini sağlama	Tüm ağ seviyesi ve paket hızı	Bir ISP'nin 15 dakika aralıklarla 12 yıl 96 saat toplanan MAWI veri kümesi	RMSE	0,098, 169,9, 4,664
HİBRİT YÖNTEMLER							
Gowrishankar ve, Satyanarayana (2009)	TSD, k-Means, RF, Prophet	E	Baz istasyonlarının kümelmesiyle tıkanıklık ve kapasite yönetimi için tatil trafiğini tahmin etme	Baz istasyon seviye trafiği	7 tatil dönemi dahil olmak üzere Çin'in 3 eyaletinden 1 yıldan az günlük BS trafik verisi	APE	%20, %50
Adeleke (2019)	BPNN	H	Kendine benzerlik modellerini yakalamak için hibrit model ile WLAN trafik tahmininin tahmin doğruluğunu iyileştirme	Tüm WLAN trafiği	Bir olay sırasında 0,1 sn aralıklarla toplanan 1 günlük WLAN izlemeleri	CE, ACE	~0, 35
Cortez vd. (2006)	DSARIMA, DSEXP	E	Çift mevsimsel mobil ağ trafiğini tahmin etmek ve modelleri birleştirmek için otomatik bir algoritma geliştirme	Baz istasyon seviye trafiği	973 Cell Tower'dan 22 haftalık saatlik çağrı bağlantı hacmi (20 eğitim + 2 test)	MAPE	7,32
Cortez vd. (2008)	MLP, MLPWD, SVM, MLP + SVM	E	Aktif kullanıcıların ayrıştırılmış bileşenlerini tahmin etmede YSA makine öğrenimi tabanlı modellerle kıyaslama yöntemleri	Hücre düzeyinde birleştirilmiş veriler	Ticari MNO'dan 24 özelliğe sahip 1 hücreden elde edilmiş 1 haftalık saatlik veriler (168 örnek - 100 eğitim + 68 test)	MAE, RMSE	8,56, 11,45
Rutka (2008)	ConvLSTM, k-Means	E	Trafik tahmini için karmaşık gizli kalıpları yakalamak üzere Spatio-Temporal Cross-Domain korelasyonlarıyla derin öğrenme mimarisi geliştirme	Hücre düzeyi trafiği	61 günlük (55 günlük eğitim + 6 günlük test) 10 dakikalık aralıklarla CDR verilerini içeren Telecom Italia veri seti	RMSE, MAE, R-kare	164,31, 90,94, 0,95
Yüksel vd. (2019)	k-Means + WT, FFNN	E	Elman Neural Network'ü eğitmek için mesafe tabanlı K-Means kümelemesini geliştirme ve bileşenlerini dalgacık dönüşümü ile çıkarma	Küme düzeyi trafiği	358 baz istasyonunun 7 günlük saatlik verileri	NMSE, NMAE, MAPE	0,0194, 0,0008, 0,0238
Xu vd. (2018)	FARIMA, FARIMA + ANN	H	Farklı zaman serisi yöntemlerini birleştirerek tahmin doğruluğunu iyileştirme	Tüm ağ trafiği	Saniyeler ve çerçeve düzeyinde ethernet paketi trafiği içeren 9 video trafiği veri kümesi	RMSE, MAE	2,28, 2,67

Çalışma	Yöntemler	Mevsimsel	Amaç	Ağ Kapsamı	Veri Seti	Değerlendirme Metriği	Doğruluk Sonucu
Xiang vd. (2010)	RF, SARIMA, VAR, HMM, ST-MVL	E	Topluluk Rastgele Orman modelinde zamansal ve mekansal korelasyonlarla trafiği tahmin ederek enerji tasarrufu	Baz istasyonu düzeyi trafiği	Bir Çin şehriden 181 günlük günlük baz istasyonu trafik verileri	ARMSE	13,2
Au vd. (2011)	ARIMA, GARCH	E	Ani yükselme gibi koşullu varyanslı trafik serilerinin genel özelliklerini kapsayan model tespit etme	Tüm ağ trafiği	35 seri ve 2 seri sample ile saniye seviyesinde toplanan iki gerçek LAN trafiği	NMSE	0,01, 0,001
Nikravesht vd. (2016)	Kalman Filtresi	E	Gözlem penceresindeki eğilime göre Doğrusal Bağlanım ve Naive yöntemini kullanarak daha az parametrelili hibrit modelle trafiği tahmin etme	Baz istasyonu düzeyi trafiği	Open Mobile Network platformundan toplanan 1 haftalık saatlik trafik istatistikleri	MAD, MSE, MAPE	0,53, 0,66, 32,21
Zhou vd. (2006)	WNN	E	Ağ karakterini tahmin etmek için yeni bir ağ modeli keşfetme	Tüm ağ trafiği	BellCore Laboratuvarı yüksek çözünürlüklü trafik verileri	N/A	N/A
Dawoud vd. (2014)	WT, Gauss, DBN, PCA, SRMF, TomoG	E	DBN ve Gauss Modeli tabanlı bir ağ trafiği tahmin modeli önerme	Tüm ağ trafiği	Gerçek ağdan alınan 5 dakikalık aralıklı trafik verilerinin 2016 örnekleri	N/A	N/A, 0,2

Burada TSD (Time Series Decomposition – Zaman Serisi Ayrıştırma), PCA (Principle Component Analysis – Temel Bileşenler Analizi), DBN (Deep Belief Networks – Derin İnanç Ağları), CDF (Cumulative Distribution Function – Birikimli Dağılım Fonksiyonu), SSE (Sum of Square Error – Hata Karelerinin Toplamı), RE (Relative Error – Göreceli Hata), AE (Absolute Error – Mutlak Hata), SMAPE (Symmetric Mean Absolute Percentage Error – Simetrik Ortalama Mutlak Yüzde Hatası), RTO (Recovery Time Objective – Kurtarma Süresi Hedefi), NMAE (Normalized Mean Absolute Error – Normalleştirilmiş Ortalama Mutlak Hata), NMSE (Normalized Mean Square Error – Normalleştirilmiş Ortalama Kare Hatası), RRMSE (Relative Root Mean Squared Error – Göreceli Kök Ortalama Kare Hata), ARMSE (Average RMSE – Ortalama RMSE)’dir. |

3. MALZEME VE YÖNTEM

3.1. VERİ SETLERİ

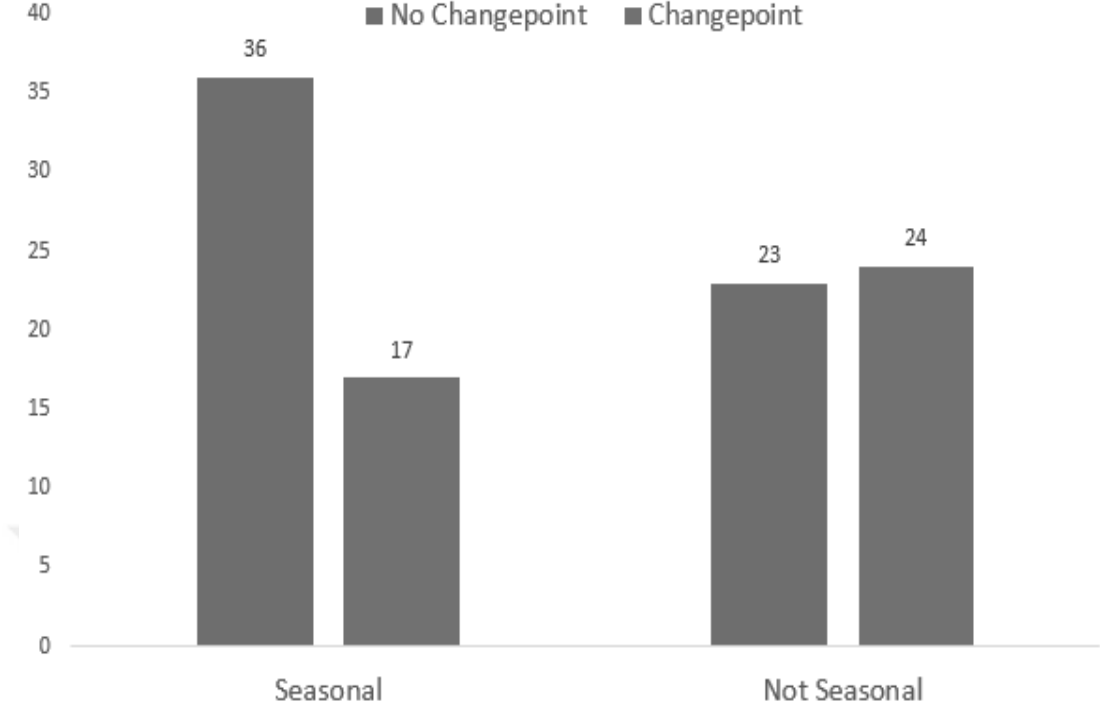
3.1.1. Mevsimsellik Problemi Veri Seti

Harici erişim için izin verilen kayıtlı geçmiş verilerin eksikliği nedeniyle mobil ağ verilerinin uzun vadeli geçmiş modellerini çalışmak ve analiz etmek o kadar kolay değildir. Hackathonlar ve akademik amaçlarla paylaşılan Telekom Italia veya China Mobile gibi mobil operatörlerden bazı kamuya açık veri kümeleri mevcuttur, ancak bunlar da iki yıl ve daha fazlası gibi yeterince uzun geçmiş verileri içermemektedir. Son yıllarda birçok MNO, uzun vadeli stratejik analitik amaçlar için veri ambarları veya veri havuzları inşa etmeye başlamıştır. Algoritmanın analiz edilmesi ve geliştirilmesi için, Kademe-1 Avrupa MNO'nun tescilli bir veri kümesi üzerinde çalışılmıştır. Toplanan veriler, megabayt cinsinden iki yıllık geçmiş hücresel LTE trafik hacmi verilerini, baz istasyonu kimlikleri, hücre kimlikleri, sektör kimlikleri, şehir, bölge, bant ve üretici bilgilerini içerir. Toplanan veriler Tablo 3.1'de sunulmuştur.

Tablo 3.1. Mevsimsellik Veri Seti İstatistikleri

Veri Türü	Sayı
Bölge	3
Şehir	4
Hücre	20680 LTE hücresi
Saha	4119 baz istasyonu
Frekans Katmanı	4 bant
Hücre Üretici	3
Zaman Aralığı	2018-Ağu-1 / 2020-Ağu-1

Genel veri seti içerisinden mevsimsellik, değişim noktası mevcudiyeti açısından farklı özelliklere sahip 150 baz istasyonu seçilmiş ve her bir baz istasyonunun özelliği üç SME (Subject Matter Expert – Konu Uzmanı) tarafından etiketlenmiştir. Toplam 50 tanesi keşif analizi ve denetim için parametre ayarı amacıyla kullanılmış, kalan 100 baz istasyonu, model doğruluk testi amacıyla ayrılmış ve herhangi bir analiz veya parametre ayarı amacıyla kullanılmamıştır. SME etiketlemesine sahip kategorilere ayrılmış alanların dağıtılmış sayısı Şekil 3.1'de gösterilmektedir.



Şekil 3.1. Mevsimsellik Etiketli Test Baz İstasyonlarının Dağılımı

SME'ler, bir sahayı mevsimsel olarak etiketlemek için eşik olarak %30 mevsimsel etki seviyesini sınır olarak kabul etmişlerdir. $\{1, \dots, n\}$ baz istasyonunun $X_n^{(k)}$ trafik değerleri serisinin aralıkları iken belirli bir zaman için (n) mevsimsel bileşeni $S_n^{(k)}$ ve eğilim bileşenini $T_n^{(k)}$ olarak belirlenirse, SME'ler λ ile gösterilen ve 0,30 olarak tanımlanan yatırım politikalarına göre bir oran eşiği tanımlar. Görüldüğü gibi mevsimsel olarak işaretlenmiş 53 baz istasyonu vardır ve 47'si ise mevsimsel değildir. Mevsimsel baz istasyonlarının 17'sinde, pandemi karantina uygulamaları veya başka bir yakın hücre yatırımının trafik çalması etkisi gibi farklılık gösterebilecek nedenlerle değişim noktası vardır. Değişim noktaları, seviye değişimleri veya eğilim artışı veya düşüşü şeklinde olabilir.

$$Is_Sea^{(k)} = \begin{cases} 1, & \exists n \ni \frac{S_n^{(k)}}{T_n^{(k)}} \geq \lambda \\ 0, & \forall n \ni \frac{S_n^{(k)}}{T_n^{(k)}} < \lambda \end{cases}$$

3.1.2. Tahminleme Problemi Veri Seti

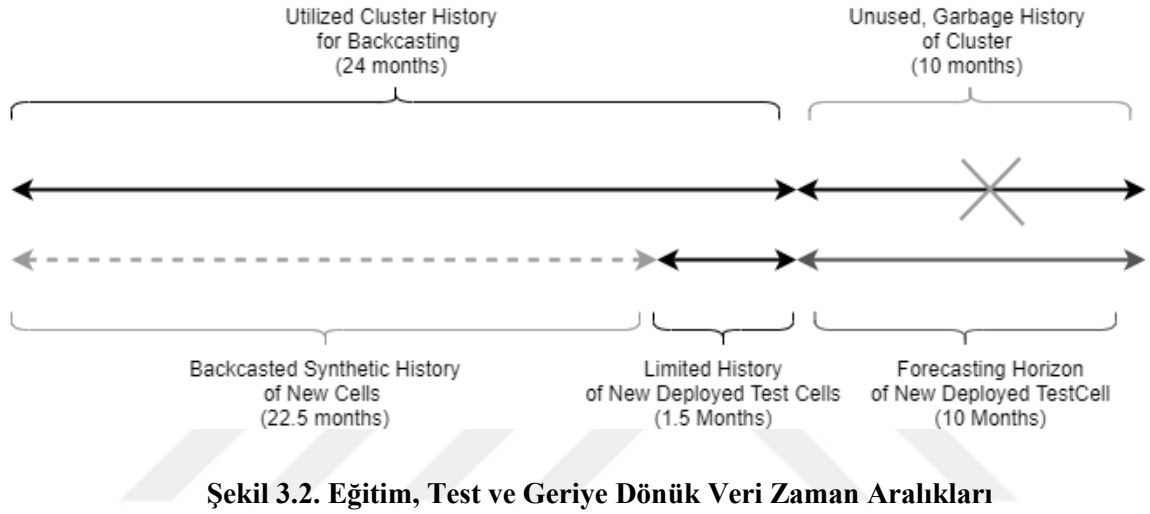
Çalışmamızda, Kademe-1 Avrupa mobil ağ operatöründen toplanan canlı ağ verileri üzerinde çalıştık. Çalışmamızın veri kaynağı, desteklenen baz istasyonu satıcılarının Öge Yönetim Sistemi (Element Management System - EMS) günlüklerinden oluşmaktadır. Ericsson ve Huawei, veri setimizdeki radyo ekipmanı sağlayıcılarıdır. Toplanan veriler, geçmiş veriler olarak maksimum 3 yıllık günlük trafik hacmine sahip 3461 LTE hücresi ve LTE hücreleri arasındaki devir faaliyetlerine dayalı son mobilite KPI'ları, hücrelerin koordinatları, hizmet azimut açısı, bant türleri, satıcı ve bölge bilgileri gibi site planlama verilerini içerir. Hücre düzeyinde trafik tahmini üzerinde çalıştık, bu nedenle sektör, eNodeB veya bölge için trafik hacmi toplamı yoktur. 11,5 ay önce devreye alınan 70 LTE hücresi seçtik. Bunların gelecek 10 ayın trafik hacmini tahmin etmek için ilk 1,5 aylık geçmişlerini kullandık. Bu 70 hücre, yeni devreye alınan eNodeB'ler üzerindedir. Bu nedenle kapasite genişletme yerine daha çok kapsama alanı iyileştirme amacıyla devreye alınmışlardır. 3461 LTE hücresi ve seçilen 70 test LTE hücresi dahil olmak üzere veri kümesinin ayrıntıları Tablo 3.2'de paylaşılmaktadır.

Tablo 3.2. Tahminleme Veri Seti İstatistikleri

Veri Türü	Sayı
<i>Bölge</i>	2 bölge
<i>Şehir</i>	3 büyük şehir
<i>Hücre</i>	3461 LTE hücresi
<i>Saha</i>	494 baz istasyonu
<i>Band</i>	3 bant
<i>Hücre Üretici</i>	2 üretici
<i>Tarih Aralığı</i>	2018-Tem-25 / 2021-May-18 (yaklaşık 3 sene)
<i>Sentetik Geçmiş Üretimi için Kullanılan Tarih Aralığı</i>	2018-Tem-25 / 2020-Tem-31 (2 sene)
<i>Yeni Devreye Alınan Hücre</i>	70 LTE hücresi
<i>Yeni Devreye Alınan Saha</i>	70 Baz istasyonu (hücre sayısına eşit)
<i>Test Tahminlemesi için Kullanılan Tarih Aralığı</i>	2020-Haz-15 / 2020-Tem-31 (1,5 ay)
<i>Test Doğruluğu için Kullanılan Tahminleme Ufku</i>	2020-Ağu-01 / 2021-May-18 (10 ay ilerisi)

3 büyük şehirdeki LTE hücrelerinin 3 yıllık geçmiş trafik hacmi verilerini ve son mobilite KPI'larını topladık. 11,5 ay önce devreye alınan 70 LTE hücresi seçildi. Devreye alım tarihinin 2020-Ağustos-01 olduğunu varsayıyoruz ve hücre başına gelecek 10 aylık trafik hacmini tahmin ederek test skorlarımızı elde ediyoruz. Çalışma kapsamında sadece ilk 1,5 aylık geçmiş

trafik bilgisini kullanarak gelecek 10 aylık trafiği tahmin etmek hedeflenmiştir. 10 aylık gerçek ölçümleri, MAPE olarak hata hesaplaması için öngörülen değerlerle karşılaştırdık. 1,5 aylık yeni hücreler için yetersiz geçmiş sorunu çözmek için kümelenmiş komşu hücrelerin önceki 2 yıllık geçmiş trafik özelliklerinden faydalandık ve yeni hücreler için sentetik bir geçmiş oluşturma yoluna gittik. Simülasyonumuzu gerçek hayat şartlarına uygun hale getirmek için kümelenmiş komşuların son 10 aylık trafik verilerini kullanmadık. Şekil 3.2'de eğitim, test ve geriye dönük veri aralıkları görselleştirilmiştir.



Herhangi bir tahmin veya kümeleme algoritması uygulamadan önce, toplanan ham verilerin ön işlenmesi çoğu zaman gereklidir. EMS sistemlerinin doğası veya planlı bakım sorunları nedeniyle, veri kümelerinde eksik verilerin veya geçerli aralığın dışında olan KPI'ların yanlış değerlerinin (yani negatif trafik hacmi, %100'den fazla Fiziksel Kaynak Blokları (PRB) kullanımı gibi) olması normaldir. Eksik veya yanlış veri sorunlarının yanı sıra, bazı harici nedenlerden dolayı (yani planlanmamış bir olayın etkisi, afetler vb. nedeniyle) KPI değerlerinde anormallikler de olasıdır. Anormallik durumunda, bunlar tespit edilmeli ve daha uygun değerle değiştirilmelidir.

Çalışmamızda betikler geliştirilerek ham verilere aşağıdaki altı adım uygulanmaktadır:

- 1- Veri temizleme ve toplama: Aralık dışı olan değerlerin temizlenmesi ve 15 günlük trafik hacmi olarak birleştirilerek yeniden örneklenmesi. Bu nedenle, bir yıllık tarih, 24 zaman adımına karşılık gelir.

- 2- Eksik veri interpolasyonu: Eksik verilerin tamamlanması veya interpolasyonlu değerlerle atanması
- 3- Anomali iyileştirme: Mevsimsel ölçekli Medyan Mutlak Sapma (MAD) algoritması ile aykırı değerlerin tespiti, anormalliklerin temizlenmesi ve 2. adıma benzer uygun değerlerle boşlukların interpolasyonu.
- 4- Devreye alım tarihlerinin çıkarılması: PM kullanılabilirliğini kontrol ederek ve test örnekleri oluşturmak için genç hücreleri seçerek hücre başına devreye alım tarihlerinin hesaplanması.
- 5- Mobilite ilişkisi gücünün hesaplanması: Hücreden hücreye hareketlilik ilişkileri elde etmek için ilişkisel geçiş KPI'larını işleme.
- 6- İşlenen verilerin saklanması: İleride yapılacak analizler için işlenen çıktılar veri tabanında saklanır.

3.2. YÖNTEM VE KÜTÜPHANELER

3.2.1. Tekrarlayan Yapay Sinir Ağları

Tekrarlayan Yapay Sinir Ağları (Recurrent Neural Networks – RNN) gizli durumlara sahipken önceki çıktıların girdi olarak kullanılmasına izin veren bir sinir ağları sınıfıdır. Bu yapıda düğümler arasındaki bağlantılar, yönlendirilmiş bir döngü oluşturmaktadır (Stanford, 2021). Bu döngü yapısı, dinamik zamansal davranış sergilemesine izin vermektedir. İleri beslemeli sinir ağlarından farklı olarak, RNN'ler kendi giriş belleğini girdilerin keyfi sıralarını işlemek için kullanabilirler. Tekrarlayan sinir ağlarının asıl amacı ardışık bilgileri kullanmaktır. Geleneksel bir sinir ağında tüm girişlerin (ve çıkışların) birbirinden bağımsız olduğu varsayılır. RNN'ler ise tekrarlayan olarak adlandırılır çünkü bir dizinin her ögesi için aynı görevi yerine getirir, çıktılar önceki hesaplamalara bağlıdır. RNN'lerin diğer bir özelliği bilgi toplayan “bellek” taşımalarıdır. Teorik olarak, RNN'ler keyfi uzun dizilerdeki bilgileri kullanabilir, ancak pratikte yalnızca birkaç adım geriye dönmekle sınırlıdır. Her bir t zamanında, $a^{<t>}$ aktivasyonu ve $y^{<t>}$ çıktısı aşağıdaki gibi ifade edilir (Bayramli, 2021):

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a) \quad y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

Burada W_{ax} , W_{aa} , W_{ya} , b_a , b_y geçici olarak paylaşılan katsayılardır ve g_1 , g_2 aktivasyon fonksiyonlarıdır.

RNN’de, tüm zaman dilimlerindeki $\mathcal{L}$ kayıp fonksiyonu, her zaman dilimindeki kaybı temel alınarak aşağıdaki gibi tanımlanır:

$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})$$

RNN’de geri yayılım zamanının her noktasında yapılır. T zaman diliminde, ağırlık matrisi W ’ye göre $\mathcal{L}$ kaybının türevi aşağıdaki gibi ifade edilir:

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^T \frac{\partial \mathcal{L}^{(T)}}{\partial W} \Big|_{(t)}$$

RNN’in başlıca uygulama alanları ve bu alanlarda kullanılan yapılar aşağıdaki gibi ifade edilebilir:

- Geleneksel Sinir Ağları: Bu uygulama alanında $T_x = T_y = 1$ şeklindeki bire-bir yapılar kullanılır.
- Müzik Üretimi: Bu uygulama alanında $T_x = 1$, $T_y = 1$ şeklindeki bire-çok yapılar kullanılır.
- Duygu Sınıflandırma: Bu uygulama alanında $T_x > 1$, $T_y = 1$ şeklindeki çoka-bir yapılar kullanılır.
- İsim Varlık Tanıma: Bu uygulama alanında $T_x = T_y$ şeklindeki çoka-çok yapılar kullanılır.
- Makine Çevirisi: Bu uygulama alanında $T_x \neq T_y$ şeklindeki çoka-çok yapılar kullanılır.

RNN yapılarında en yaygın kullanılan aktivasyon fonksiyonları aşağıdaki gibi ifade edilebilir:

- Sigmoid Aktivasyon Fonksiyonu: $g(z) = \frac{1}{1+e^{-z}}$ şeklinde tanımlanır.
- Tanh Aktivasyon Fonksiyonu: $g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ şeklinde tanımlanır.
- RELU Aktivasyon Fonksiyonu: $\max(0, z)$ şeklinde tanımlanır.

RNN yapılarının başlıca avantajları aşağıdaki gibi ifade edilebilir:

- Herhangi bir uzunluktaki girdilerin işlenmesi imkanı
- Girdi büyüklüğüyle artmayan model boyutu
- Geçmiş bilgileri dikkate alarak hesaplama
- Zaman içinde paylaşılan ağırlıklar

RNN yapılarının başlıca dezavantajları aşağıdaki gibi ifade edilebilir:

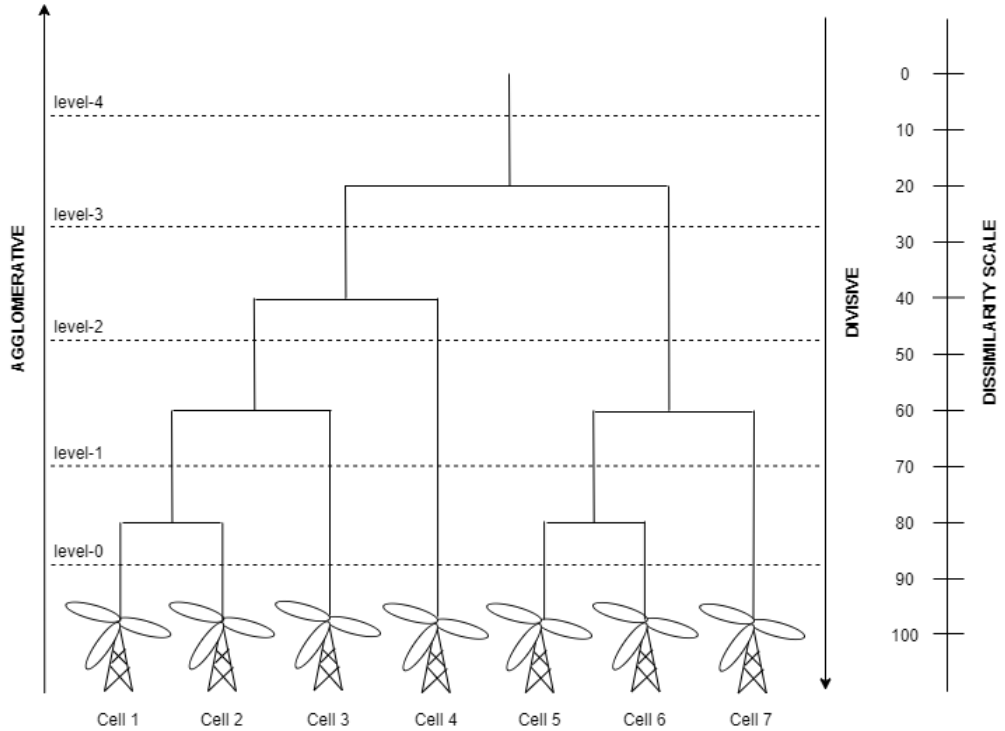
- Yavaş hesaplama
- Uzun zaman önceki bilgiye erişme zorluğu
- Mevcut durum için gelecekteki herhangi bir girdinin düşünülmemesi

3.2.2. *Hiyerarşik Kümeleme Algoritması*

Kümeleme, örüntüleri (gözlemler, veri elemanları veya özellikli vektörleri) gruplar (kümeler) halinde sınıflandıran, denetimsiz bir yöntemdir. Kümeleme problemi birçok alandaki farklı problemlerde kullanılmış ve üzerinde çalışılmış bir problemidir. Bu durum bu problemin, veri analizi konusunda ne kadar faydalı bir yöntem olduğunun göstergesidir (Jain vd., 1999). Kümeleme işlemi için kullanılan algoritmalar; sıralı algoritmalar, hiyerarşik algoritmalar, genetik algoritma, stokastik yöntemler ve yoğunluk-tabanlı algoritmalar gibi maliyet fonksiyonu optimizasyonu temelli algoritmalar şeklinde kategorize edilebilirler (Koutroumbas vd., 2009). Rafsanjani vd. (2012) çalışmasında, araştırmacıları bazı hiyerarşik kümeleme algoritmalarını ve bu algoritmaların özelliklerini ele almış ve bu algoritmalar arasında bir karşılaştırma işlemi gerçekleştirmiştir. Hiyerarşik kümeleme algoritmasının k-ortalama gibi diğer denetimsiz diğer algoritmalarla karşılaştırıldığında, eğitim verisinin hacmine göre daha uzun bir “en kötü durum karmaşıklığı” na sahip olabileceği görülmüştür. Fakat bu tez çalışmasında gerçekleştirilen işlemlerde operatörlerin tercihine göre yerel alanlara veya bölgelere bölünebildiğinden, kümelemenin toplam zamanını azaltmak mümkün olabilmektedir. Ayrıca, kümeleme bir yığın işlem gibi düşünülebilir ve böylece seçilen kümeleme algoritmasının zaman karmaşıklığı ile ilgili sorunlar paralellik vb yöntemler kullanılarak elimine edilebilir. Hiyerarşik kümeleme algoritmaları, yaprak düğümlerden başlanan ve kümelerin beraber birleşerek ilerlediği aglomeratif ve kök düğümden başlanan ve özyineli olarak kümeleri bölen bölücü algoritmalar şeklinde ifade edilebilir. Algoritmalarındaki gözlem

çiftlerindeki benzerliği tespit etmek için herhangi bir geçerli metrik kullanılabilmektedir (Rahmat Widia vd., 2012).

Aglomeratif algoritmalar her elemanı ayrı bir küme olarak alır ve sonrasında bu elemanları daha büyük kümeler şeklinde birleştirir. Bölücü algoritmalar tüm küme ile başlar ve bu büyük kümeyi daha küçük kümelere bölerek ilerler (Rahmat Widia vd., 2012). Bir veri setindeki tüm gözlemlerdeki ilişkileri anlatan bir dendrogram, verilerin arasındaki hiyerarşik ilişkileri anlamak için çok faydalıdır, birçok durumda belli bir sayıda spesifik kümeye ihtiyaç bulunmaktadır (Myatt, 2009). Şekil 3.3'te örnek bir dendrogram verilmiştir, bu şekilde aglomeratif modelde dendrogramın düşük seviyeleri için farklılık seviyesi artmaktadır. Bu tez çalışmasında yapılan işlemlerde de ilişkili hücrelerin yetersiz geçmişini geriye dönük olarak ele almak için komşu hücrelerin seçimi amacıyla aglomeratif hiyerarşik kümeleme modeli kullanılmıştır. Seçilen kümelere toplu trafik karakteristik özellikleri yetersiz geçmişi çözümlemek için kullanılmaktadır.



Şekil 3.3. Hiyerarşik Kümeleme için Bir Dendrogram Örneği

3.2.3. SARIMAX Tahminleme Algoritması

SARIMAX, mevsimsel karakteristik bileşenlerden oluşan popüler bir klasik zaman-serisi modelleme yöntemidir. Yakın ölçümlerin oto-regressif işlemleri ve bir dış tahminleyicinin modellenmesi ile ilgili oluşan hataların yok edilmesi üzerinde temellendirilmiştir (Pankratz, 1991).

AR'ın doğru değerlerini tanımlayarak, mevsimsel olan ve olmayan bileşenler için MA komutları, tahminleme doğruluğunda kritik bir role sahiptir ve uygun model parametreleri için grid arama-tabanlı yaklaşımlar kullanılmaktadır. Bu tez çalışmasında da parametre tahminleme için ızgara-arama yaklaşımı kullanılmıştır.

$X = \{X_t, t = 1, \dots, T\}$ gözlemlenmiş trafik hacim örneklerinin kümesi olarak kabul edilirse, bu durumda bir ARIMA işlemi (3.1) - (3.11) arasında modellenir.

Fark operatörü (3.1)'de tanımlanmıştır.

$$Y_t = \nabla^d X_t \quad (3.1)$$

Ardından ARMA işlemi (3.2)'deki gibi verilmiştir.

$$Y_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \dots + \phi_p Y_{t-p} + Z_t + \theta_1 Z_{t-1} + \dots + \theta_q Z_{t-q} \quad (3.2)$$

Burada Z_t 'nin beyaz gürültülü rastgele salınımlı $Z_t \sim iid(0, \sigma^2)$ prosesidir. Geri (lag) operatörü (3.3)'teki gibi tanımlanmıştır.

$$B^p X_t = X_{t-p} \quad (3.3)$$

Burada s mevsimsel periyodikliği ifade ettiğinde, mevsimsel fark (3.4)'teki gibi ifade edilebilir.

$$\nabla_s = (1 - B^s) \quad (3.4)$$

Buna göre (3.5) eşitliği sağlanır.

$$Y_t = \nabla_s^D X_t \quad (3.5)$$

SARIMA(p, d, q, P, D, Q)_s modeli (3.6)'daki gibi bir geri operatörü şeklinde ifade edilebilir.

$$\Phi_p(B^s)\phi_p(B)(1 - B^s)^D(1 - B)^d X_t = \Theta_q(B^s)\theta_q(B)Z_t \quad (3.6)$$

Burada (3.7) - (3.10) arasındaki eşitlikler sağlanır.

$$\theta_q(B) = 1 + \theta_1 B + \dots + \theta_q B^q \quad (3.7)$$

$$\theta_Q(B^S) = 1 + \theta_1 B^S + \theta_2 B^{2S} \dots + \theta_Q B^{QS} \quad (3.8)$$

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (3.9)$$

$$\Phi_P(B^S) = 1 - \Phi_1 B^S - \Phi_2 B^{2S} - \dots - \Phi_P B^{PS} \quad (3.10)$$

Sonuç olarak, SARIMAX (3.11)'de verilen v dış gerici ile genişletilebilir.

$$\Phi_P(B^S)\phi_p(B)(1 - B^S)^D(1 - B)^d X_t = \theta_Q(B^S)\theta_q(B)Z_t + \sum_{i=1}^n B_i v_t^i \quad (3.11)$$

3.2.4. STL-ETS Tahminleme Algoritması

Holt Winter'ın ETS algoritması, Holt (1957) ve Winters (1960)'da trend ve mevsimsel bileşenlerin her ikisi için ağırlıklandırılmış ortalama bir modda yakın zamandaki gözlemleri ele alarak tahmin sağlamaktadır. Ayrıca, ek olarak seviye bileşenlerini içermektedir. Modelimiz toplamsal bir zaman serisi modelidir ve formülü (3.12) - (3.15) arasında gösterilmektedir. Böylece, tüm bu bileşenler (3.15)'teki $\hat{y}$ 'nin gelecek değerini tahmin etmek için toplanır. Mevcut gözlem t ile ifade edildiğinde bu tahminleme işlemi h ile gösterilmektedir.

$$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \quad (3.12)$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \quad (3.13)$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \quad (3.14)$$

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)} \quad (3.15)$$

Burada α , β^* ve γ , sırasıyla seviye, eğilim ve mevsimsellik ifade eden yumuşatma katsayılarıdır. Yumuşatma katsayıları, bileşen gözlemi ve önceki tahminler arasındaki yakın zamanlı değişimlerin ağırlıklarını tanımlamak için kullanılır. Daha yüksek bir yumuşatma katsayısı yakın zamandaki değişiklikler üzerinde daha çok önem anlamına gelmektedir, dolayısıyla bileşen tahmini ve tüm tahmin, trafikteki ani değişikliklere daha hassas olur. Holt'un ETS algoritması yalnızca trend bileşenleri modellerken Winter'ın ETS algoritması ise yalnızca

mevsimsel bileşenleri modeller. Buna göre, Holt Winter'ın ETS algoritması her iki bileşen türü üzerine de yapılandırılmış olur.

Bu tez çalışmasında, mevsimsel bileşenler STL algoritması ile modellenmiştir. Bu çalışmada orijinal trafik serilerinden mevsimsel bileşenler çıkartılmış ve kalan seriler Holt'un ETS versiyonu ile modellenmiştir. Mevsimsel bileşenler halihazırda STL ile modellendiği için Holt'un ETS algoritması kalan trend bileşeni modellemek için kullanılmıştır. Bu da bu tez çalışmasındaki modelin neden HW-ETS gibi değil de STL-ETS gibi adlandırıldığını açıklamaktadır.

STL, sırasıyla trend, mevsimsel ve kalan bileşenler içinde zaman serileri oluşturan bir filtreleme prosedürüdür. STL Loess yumuşatma uygulamalarının dizisini içeren basit bir tasarıma sahiptir. Buradaki basitlik, uzun zaman serileri ve trend ve mevsimsel yumuşatma için dahi, prosedürün özelliklerini analiz etmeye ve hızlı hesaplama yapmaya izin vermektedir. STL'nin diğer özellikleri, çok küçük bir seviyedeki yumuşatmadan geniş seviyedeki yumuşatmaya kadar mevsimsel ve trend yumuşatmanın miktarlarını ifade eder (Cleveland vd., 1990).

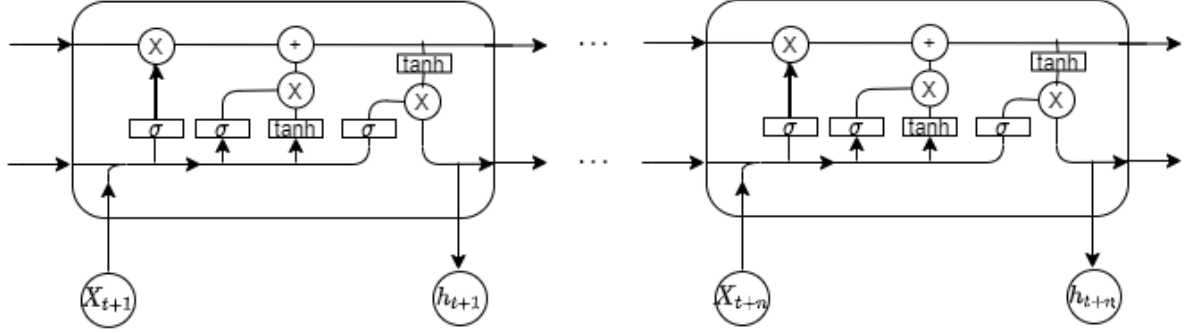
STL yalnızca toplamsal bileşenleri modelleyebilir. Çarpımsal mevsimseli modellemek için, mobil ağırları için veri setimizde sıklıkla yer alan log-dönüşümlerine gereksinim duyulmaktadır. Bir Loess yumuşatıcı veri setindeki her noktada yerel olarak ağırlıklandırılmış polinomsal regresyon olarak uygulanmaktadır. STL prosedürü için olan parametreler verilen bir zaman serisinin frekans cevap analizi ve özdeğerlerden elde edilmiştir (Theodosiou, 2011).

3.2.5. LSTM Bazlı Tahminleme Algoritmaları

LSTM ağırları, Hochreiter vd. (1997)'de tanımlanmış olan RNN'nin özel bir türüdür. LSTM, geri yayılım algoritmasının doğası gereği, kaybolan gradyan problemi tarafından oluşan uzun-aralık bağımlılık problemini çözümlmek için kullanılır (Bengio vd., 1994). Özellikle ağ kapasitesi genişletme kararları için uzun vadeli mobil ağ trafiği kalıpları dikkate alındığında, hücre sektörlerinin yıllık mevsimselliği gibi uzun vadeli özelliklerin tahmin modelleriyle kapsanması gerekir.

Vanilla RNN mimarisi, ANN modelini uzun menzilli bağımlılıkta güçlendirmek için kapılar konseptiyle genişletilmiştir, bu nedenle önceden çıkarılan bazı bilgileri unutmak ve hücre

durumları olarak taşınan daha yeni bilgilerle değiştirmek mümkündür. Unutma, giriş ve çıkış kapıları olan bir LSTM mimarisi Şekil 3.4'te gösterilmiştir.



Şekil 3.4. Standart Uzun Ömürlü Kısa Bellek Sinir Ağı Mimarisi

Aşağıdaki matematiksel formüller sırasıyla unutma kapıları, giriş kapıları, çıkış kapıları, hücre, gizli durumlar ve t zamanının örnek bir değerini ifade eden f_t , i_t , o_t , C_t , h_t ve X_t kapılarıdır. W_z vektörleri, eğitime modeli boyunca öğrenen kapıların ağırlık katsayılarıdır ve b_z LSTM hücresinin ilgili katmanları için bias değerleridir. Kapılar için, sigmoid fonksiyonu (3.16) kullanılmıştır ve bir LSTM'nin gizli durumunda neyin korunması gerektiğine karar vermek için tanh fonksiyonu (3.17) kullanılmıştır. Diğer formüller (3.18) - (3.23) arasında verilmiştir.

$$\sigma = S(z) = \frac{1}{1 + e^{-z}} \quad (3.16)$$

$$\text{Tanh} = T(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (3.17)$$

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \quad (3.18)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \quad (3.19)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, X_t] + b_C) \quad (3.20)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (3.21)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \quad (3.22)$$

$$h_t = o_t * \tanh(C_t) \quad (3.23)$$

Bu tez çalışmasında, Vanilla LSTM modelinde dahi, aktivasyon fonksiyonu olarak tanh yerine, ReLU (Doğrultucu Lineer Birim) kullanılmıştır (3.24). ReLU aktivasyon fonksiyonu kaybolan gradyan probleminin üstesinden gelmektedir. Bu sayede, genellikle çok-katmanlı derin ağların varsayılan aktivasyon fonksiyonu olmaktadır. Ayrıca, tez çalışmasında ReLU aktivasyon fonksiyonuna sahip olan Vanilla LSTM modelin yanında, LSTM Kodlayıcı-Kod Çözücü Sinir Ağları, CNN-LSTM Kodlayıcı-Kod Çözücü Sinir Ağları, Evrişimli LSTM (ConvLSTM) Kodlayıcı-Kod Çözücü Sinir Ağları geliştirilmiştir.

$$\text{ReLU} = R(z) = \begin{cases} z, & z > 0, \\ 0, & z \leq 0 \end{cases} \quad (3.24)$$

3.3. ARAÇLAR

3.3.1. Visual Studio Code Entegre Geliştirme Ortamı

Visual Studio Code, günümüz programlama dillerinin bağımsızlık ve farklı işletim sistemlerinde çalışma ihtiyacı gibi özelliklerine uyum gösterecek şekilde Microsoft, Linux ve Mac işletim sistemlerinde çalışabilen bir araçtır. Kurulum yapıldığında basit bir metin editörü görüntüsüne sahip olan ürün, eklentiler sayesinde Node JS, Ruby, Python, C/C++, C#, Javascript gibi birçok programlama dilini desteklemektedir. Diğer bir deyişle klasik Visual Studio ürünleri gibi her şeyi bünyesinde barındırmak yerine çekirdek bir yapı ile başlatılıp lazım olan parçaların eklentiler halinde ürüne eklenmesi sağlanmış ve eklentilerin indirilebildiği bir market ortamı oluşturulmuştur (Visual Studio Code, 2021; Çetinkaya, A., 2021).

Visual Studio Code, hata ayıklama özellikleriyle, gelişmiş web ve bulut uygulamaları üstünde kodları düzenlemeye, yeniden tanımlamaya ve optimize etmeye yaramaktadır. Visual Studio Code tamamen ücretsiz ve açık kaynak kodlu bir hizmettir. Visual Studio Code, Mac OS X, Linux ve Windows işletim sistemlerinde CoreCLR ve ASP.NET 5 uygulamaları geliştirmek ve diğer dilleri desteklemek için sunulan bir kod düzenleme yazılımıdır. Hata ayıklama, sürüm kontrol ve ASP.NET 5 uygulamaları en önemli özellikleri arasında yer alır. Visual Studio Code, bir kaynak kodu düzenleyicisinin basitliğini kod tamamlama ve hata ayıklama gibi güçlü geliştirici araçlarıyla birleştirir. Hafif olma ve hızlı başlamanın yanı sıra, Visual Studio Code, değişkenler, yöntemler ve içe aktarılan modüller için kod tamamlama özelliğine sahiptir;

grafiksel hata ayıklama, çoklu imleç düzenleme, parametre ipuçları ve diğer güçlü düzenleme özellikleri; şık kod gezintisi ve yeniden düzenleme ve Git desteğini içeren yerleşik kaynak kodu kontrolünü içerisinde bulundurur. Bir dizi programlama dili ve belirli bir dilde bulunabilen veya bulunmayacak özellikler kümesini destekler. Visual Studio Code merkezi bir depoda bulunan eklentiler aracılığıyla genişletilebilir. Bu, editör ve dil desteğine eklemeler içerir. Dikkat çeken bir özellik de kodu analiz eden, statik analiz için linters ve araçlar gibi uzantıları oluşturma olanağıdır.

Visual Studio Code ürününün başlıca özellikleri aşağıdaki gibidir:

- Zengin bir kod gözden geçirme desteği vardır.
- Platform bağımsız kod geliştirebilmeyi destekler.
- Zengin bir kod tanıma içeriği vardır.
- Kod hata ayıklama desteği vardır.
- Hızlıdır.

Visual Studio, esas olarak .NET geliştirme için kullanılan tam bir IDE'dir (Integrated Development Environment - Entegre Geliştirme Ortamı) ve yalnızca Windows işletim sisteminde kullanılır. Öncelikle Windows makineleri/sunucuları için kod geliştirmede kullanılır. Visual Studio Code ise, birçok isteğe bağlı eklentiye sahip bir metin düzenleyicidir.

3.3.2. Python Programlama Dili ve Ortamı

Python son birkaç yılda modern yazılım geliştirme, altyapı yönetimi ve veri analizinde birinci sınıf bir programlama dili olarak ön plana çıkmaktadır (Lutz, 2001; Oliphant, 2007; Raschka, 2016). Özellikle web uygulaması oluşturma ve sistem yönetiminde önemli rol alma, veri analizleri ve makine öğrenimi gibi konularda sıklıkla kullanılmaktadır. Python'un önemli avantajları aşağıdaki gibidir (Bilginç, 2021):

- Öğrenmesi ve Kullanması Oldukça Kolay Olan Bir Dildir
- Geniş Kullanım ve Uygulama Alanlarına Sahip Olan Bir Dildir
- Geniş Çaplı Bir Dildir

Python'un kullanım alanları aşağıdaki gibi ifade edilebilir:

- Ansible ve Salt gibi araçlarda sistem gereksinimlerini ve yapılandırmasını sağlayan, bunlara ek olarak web tarayıcıları veya uygulama GUI'leriyle etkileşimleri otomatikleştiren özellikleri bulunmaktadır.
- Python programlama dili ile konsol uygulamaları ve çoklu platformlara GUI uygulamaları oluşturulabilir.
- Python makine öğrenmesi ve veri madenciliği için elverişli bir programlama dili olmuştur. Python arayüzündeki kütüphanelerin birçoğu makine öğrenimi ve veri bilimi üzerine elverişlidir.
- Python ile web servisleri ve RESTful API'ler kodlanabilir.
- Diğer diller ile karşılaştırıldığında Python'da yer alan bütün modüller ve kütüphaneler birer nesne olarak görev yapar. Bunun sayesinde Python, etkili bir kod derleyici olarak ön plana çıkmaktadır.
- Python'un çoklu platform uygulamaları oluşturması kod derleyici sistemlerinde etkili kodlar oluşturmak için uygun olmasını da sağlar.
- Python ile, Cython veya Numba gibi projeler kullanılarak C'ye göre biraz daha yavaş ancak hafıza anlamında C ile yarışabilecek potansiyele sahip yazılımlar geliştirilebilir.

Python'un dezavantajları aşağıdaki şekilde ifade edilebilir:

- Yüksek düzeyde bir programlama dili olduğu için sistem düzeyinde programlama için uygun değildir.
- Platformlar arası bağımsız ikili dosyalar için çağrı yapan durumlarda ideal değildir. Windows, MacOS ve Linux işletim sistemleri için bir uygulama geliştirilebilir ancak oluşturulacak uygulama fazla görkemli bir uygulama olmaz.
- Genel olarak Python programlama dilinde geliştirme hızı ön planda olduğu için yüklü programlarda fazla işlevsellik sağlamaz.

Python'un son zamanlarda oldukça başarılı bir dil olmasının sebeplerinden biri kendine özgü zengin kütüphanelere sahip olmasıdır. Python, hem güçlü bir standart kütüphaneden hem de üçüncü taraf geliştiricilerden kolayca elde edilen, edinilen bilgilerin de kolayca kullanılabilen kütüphanelerden destek almaktadır. Çalışmalarımız kapsamında kullandığımız kütüphaneler listelenmiştir. Bu kütüphaneler programlama dilini oldukça zenginleştirmiştir. (Öztürk, 2021):

1. NumPy (Numerical Python – Sayısal Python) : Veri bilimi çalışmaları için olmazsa olmaz olarak ifade edilen, çok büyük ölçekteki matematiksel-bilimsel hesaplamaları kolay, hızlı ve esnek bir şekilde gerçekleştirmeyi sağlayan çözümleri içeren kütüphanedir. Ağırlıklı olarak Veri Madenciliği, Makine Öğrenmesi, Veri Analizi, Tek-Çok Boyutlu dizi içeren hesaplama yapılarında kullanılır.
2. SciPy (Scientific tools for Python – Python için Bilimsel Araçlar) : Veri Bilimi çalışmaları için olmazsa olmaz olarak ifade edilen bir diğer kütüphanedir. NumPy'e göre matematiksel çözümlerden ziyade daha bilimsel çözümlerin ağırlıklı olarak ele almaktadır.
3. Matplotlib (Mathematical Plot Library – Matematiksel Çizim Kütüphanesi): Veri Bilimi çalışmaları için sayısal matematik hesaplamalarını iki ya da üç boyutlu görsel çıktılar haline getiren bir çizim (görselleştirme) kütüphanesidir.

4. SciKit-Learn (Science Kit/Kits Learn – Bilim Öğrenim Kiti) : Veri Bilimi ve Makine Öğrenmesi çalışmaları (Veri İşleme, Bağlanım, Küme Analizi vb.) için kullanılan çok yönlü bir kütüphanedir.
5. Keras: Birçok derin öğrenme modelini tanımlamak, anlamlı hale getirmek ve eğitmek için uygun bir çözüm yolu sağlayan derin öğrenme kütüphanesidir.
6. TensorFlow (Tensör Akışı) : Bir dizi görev arasında veri akışı ve türevlenebilir programlama için kullanılan ücretsiz ve açık kaynaklı bir kütüphanedir. Ağırlıklı olarak sembolik matematik işlemleri ve sinir ağları gibi makine öğrenimi uygulamaları için kullanılır.
7. Pandas: Veri işleme ve analizi için kullanılan bir kütüphanedir. Temel olarak zaman etiketli serileri ve sayısal tabloları işlemek için bir veri yapısı oluşturmakta ve yapıyı çeşitli işlemler ile analiz etmektedir.
8. Statsmodels (Statistical Models – İstatistiksel Modeller) : İstatistiksel modelleri tahmin etmek ve istatistiksel testler yapmayı sağlayan bir kütüphanedir. Farklı veri türleri işleyebilmek, geniş tanımlayıcı istatistik bir yapıya sahip olmak, istatistiksel test çözümleri, çizim fonksiyonu ve sonuç istatistikleri listesi geliştirebilmek gibi özelliklere sahiptir.
9. Seaborn: Matplotlib, Numpy, Pandas, Scipy, Statsmodel ve PyData gibi çevik ve çok yönlü kütüphaneler üzerine alt yapısı yapıları oluşturulan, istatistiksel hesaplamalara yeni bir anlam ve farklı bir bakış açısı kazandırmak için kullanılan oldukça efektif bir veri görselleştirme kütüphanesidir.
10. SQL Alchemy (SQL Simyası) : Açık kaynaklı bir SQL araç takımı ve ORM (Object Relational Mapping – Nesne İlişkisel Eşleştirici) kütüphanesidir. Veritabanı tablolarını Python objeleri ile eşleştirip, objeler üzerinden veritabanı veri işleme işlemleri yapmayı sağlar.

3.3.3. PostgreSQL Veritabanı Yönetim Sistemi

PostgreSQL, güçlü özellikler ve avantajlara sahip, açık kaynaklı ve tamamen ücretsiz nesne tabanlı ilişkisel bir veri tabanı sistemidir. SQL dilinin güvenlik, depolanabilirlik ve ölçeklendirilebilme özelliklerinden faydalanan PostgreSQL, birçok alanda veri tabanı yöneticisi olarak da kullanılmaktadır. 1986 yılında Berkeley'deki Kaliforniya Üniversitesi'nde POSTGRES projesinin bir parçası olarak doğan PostgreSQL birçok alanda veri tabanı yöneticisi olarak da kullanılmaktadır. PostgreSQL, kullanıcılara sunduğu başarılı veri mimarisi, doğruluğu, sağlam özellik seti, esneklik ve açık kaynak avantajıyla sektörde en çok kabul gören veri tabanı yönetim sistemlerinden biridir. UNIX ve UNIX türevi sistemler başta olmak üzere PostgreSQL; Linux, MacOS ve Windows gibi birçok işletim sistemi tarafından desteklenmektedir (Drake ve Worsley, 2002).

PostgreSQL özellikleri aşağıdaki gibi ifade edilebilir (Matthew ve Stones, 2005):

- Veri tabanı mimarisine ihtiyaç duyulan birçok alanda kullanılabilir.
- Birçok majör işletim sistemi tarafından desteklenir, Windows, MacOS, Solaris, SuSE ve OpenSuSE, Ubuntu Linux ve varyantlarında, Debian GNU/Linux ve varyantlarında, Red Hat Linux ailesinde (CentOS/Fedora/Scientific/Oracle ve varyantlarında), diğer Linux dağıtımlarında, OpenBSD ve FreeBSD üzerinde kullanılabilir.
- Birçok programlama dili tarafından desteklenir. C (libpq), Embedded C (ecpg), Python, Java (jdbc) ve TCL (libpgtcl) ile kullanılabilir.
- Veri tabanı içerisinde yer alan farklı işlemleri desteklemek için teknolojilere sahiptir. Performans açısından diğer ticarî veya açık kaynak kodlu veri tabanlarıyla kıyaslandığında başarılı metrikler elde etmektedir.

3.3.4. ClickHouse Veritabanı Yönetim Sistemi

ClickHouse, Yandex'in geliştirdiği ve kendi bünyesinde bulunan çeşitli servislerin altyapısında başarıyla çalışan, OLAP (Online Analytical Processing - Çevrimiçi Analitik İşleme) için kullanılabilen, Haziran 2016 yılında Apache 2 Lisansı ile hizmete sunulan açık kaynak kodlu bir Veritabanı Yönetim Sistemi'dir. Sütun bazlı bir veri tabanı yönetim sistemi olan ClickHouse, büyük miktardaki veriden gerçek zamanlı analitik raporlar alınabilmesini sağlayarak geliştiricilere büyük faydalar sağlamaktadır. Metrica başta olmak üzere Yandex'in e-posta, reklam ve daha birçok servisinde kullanılmakta olan sistem, 2011'de işlem başına 1000 özellik içeren 10 milyar işlemin saklanması ve işlenmesi için CERN'in Büyük Hadron Çarpıştırıcısı deneylerinde de başarıyla kullanılmıştır. ClickHouse web ve mobil uygulamalar için geliştirilen analiz araçlarının yanı sıra büyük miktarda verinin hızlı, etkin, güvenilir ve ölçeklenebilir şekilde işlenmesini gerektiren telekomünikasyon, dijital reklamcılık, artırmalı satış sistemleri, e-ticaret, online oyunlar ve bilgi güvenliği alanlarında da kullanılabilen, coğrafi olarak dağıtılmış yüzlerce bileşenden oluşan sunucu sistemlerinin yanı sıra tek bir sunucu ve hatta sanal makinelerde de çalışabilen bir veri tabanı yönetim sistemidir (Imashev vd. 2019).

ClickHouse'ın ana özellikleri şu şekilde ifade edilebilir (Argedor, 2021):

- Gerçek sütun odaklı bir veri tabanı yönetim sistemidir. Bu sistemde değerlerle hiçbir şey saklanmamaktadır. Örneğin, sabit uzunluklu değerler, uzunluk "sayı"larının değerlerin yanında saklanmasını önlemek için desteklenmektedir.
- Doğrusal ölçeklenebilirlik özelliği mevcuttur. Sunucu ekleyerek bir kümeyi genişletmek mümkündür.
- Hata toleransı özelliği mevcuttur. Sistem, her bir parçanın bir grup kopya olduğu bir parça kümesidir. ClickHouse, zaman uyumsuz çoklu ana kopya çoğaltma kullanmaktadır. Veriler herhangi bir mevcut kopyaya yazılmakta ve ardından kalan tüm kopyalara dağıtılmaktadır.
- Petabaytlarca veriyi saklama ve işleme yeteneği mevcuttur.

- SQL desteđi mevcuttur. ClickHouse, dizileri ve iç içe geçmiş veri yapılarını, yaklaşık ve URI (Uniform Resource Identifier - Tekdüzen Kaynak Tanımlayıcı) işlevlerini ve harici bir anahtar-değer deposunu bağlamak için kullanılabilirliği içeren genişletilmiş SQL benzeri bir dili desteklemektedir.
- Yüksek performanslıdır. Vektör hesaplamaları kullanılır. Veriler yalnızca sütunlara göre depolanmaz, aynı zamanda vektörler (sütunların bölümleri) tarafından da işlenmektedir. Bu yaklaşım, yüksek işlemci performansı elde etmeyi sağlamaktadır.
- Veri sıkıştırma özelliđi mevcuttur.
- Sabit disk sürücüsü optimizasyonu mevcuttur. Sistem, rastgele erişim belleđine sığmayan verileri işleyebilmektedir.

ClickHouse için yaygın durumlardan biri sunucu günlüğü analizidir. ClickHouse'a düzenli veri yüklemeleri ayarladıktan sonra, anlık sorgularla olayları analiz etmek veya hata oranları, yanıt süreleri gibi bir hizmetin ölçümlerini izlemek mümkündür. ClickHouse, şirket içi analistler için dahili bir veri ambarı olarak da kullanılabilir. ClickHouse, farklı sistemlerden (Hadoop veya belirli günlükler gibi) verileri depolayabilir ve analistler verilerle dahili gösterge tabloları oluşturabilir veya iş amaçlı gerçek zamanlı analiz gerçekleştirebilir (ClickHouse, 2021). |

4. BULGULAR

4.1. MEVSİMSELLİK PROBLEMİ BULGULARI

Bu tez kapsamındaki uygulamalardan ilkinde, seviye kayması veya üstel eğilim artışı nedeniyle değişim noktası varlığı dikkate alınarak baz istasyonları üzerindeki mevsimsel etkiyi ortaya çıkarmak için baz istasyonu trafik verilerinin mevsimsellik tespiti ve ayrıştırılmasına yönelik hibrit bir algoritma tasarlanmış ve sunulmuştur. Geliştirilen uygulamadaki hedef, benzersiz çıktısı olan, çevrimdışı ve toplu bir çözüm olarak canlı mobil ağ geçmiş verileri üzerinde çalışan tatmin edici bir algoritma bulmaktır. Algoritma, geçmişinde değişim noktası içerse bile baz istasyonu trafiği üzerindeki mevsimsel etkiyi tespit etmede başarılı olmalıdır. Literatürde değişim noktaları tespiti ve mevsimsellik tespiti ile ilgili Telekom sektörü dışında diğer alanlarda ayrı ayrı çalışmalar bulunduğu ancak mobil ağ trafiğini araştırmak için bir araya getirilmediği görülmüştür.

Baz istasyonlarının mevsimsellik etkisiyle etiketlenmesi, sonraki yılların trafik talebinin tahmin edilmesine yardımcı olan bir işlemdir. Mobil ağlar için otomatik kapasite planlama, geçmiş kalıpları kullanarak trafik talebinin uzun vadeli tahminini gerektirir. Doğru yatırım zamanına ve doğru kapasite genişletme boyutuna karar vermek veya dışsal özelliklere sahip tahmin algoritmalarının doğruluğunu iyileştirmek için hem mevsimsel ayrıştırma hem de mevsimsel dönem tanımlama önemlidir. Geliştirilen algoritmada da bu özelliklerden faydalanılmaya çalışılmıştır. Algoritma, yerleşik değişim noktası arama, değişim oranı analizi ve içindeki benzerlik kontrol mekanizmalarından oluşmaktadır ve iki yıllık geçmiş trafik verilerini kullanarak çarpımsal mevsimsel faktörleri hesaplayabilmektedir.

Mobil trafik talebinin sağlıklı bir şekilde tahmin edilmesi kapasite yatırım kararlarının doğruluğu için faydalıdır. Çünkü, birçok baz istasyonu, mobil ağların dinamizmi nedeniyle geçmişlerinde değişim noktalarına sahiptir. Örneğin, yeni konuşlandırılan bir baz istasyonu, mevcut komşu baz istasyonlarından trafik çalmaya açıktır. Bir başka değişim noktası nedenine örnek olarak, pandemi karantina kısıtlamaları verilebilir. Bu kısıtlamalar iş alanları veya eğlence yerleri gibi belirli bölgelerde beklenmedik seviye değişimlerine neden olabilirler. Bu uygulamada da veri setinde benzer davranışlar gözlemlenmiş ve parça bazında Bayes tipi bağlanım tabanlı değişim noktası dağılım analizi, benzerlik araması ve esnek STL ayrıştırmayı birleştiren bir algoritma geliştirilmiştir. Bu uygulamadaki amaç, başka bir değişim noktası

algılama algoritması tanımlamak değil, Prophet'in Bayesci yaklaşımı üzerine inşa edilmiş Bayes temelli bir yaklaşım kullanmaktır. Ancak bu yaklaşımın yetersiz kaldığı durumları kotarabilecek genişletmeler yapmaktır.

Algoritma tasarlanırken gerçekleştirilen uygulamada, gerçek ağların günlük trafik verileri ele alınmıştır. Kullanılan veri seti 4119 baz istasyonundan alınan veriden oluşmaktadır. Bunlardan 150 tanesi merkezi planlama ekiplerinden olan üç SME tarafından etiketlenmiş verilerdir. Bu 150 baz istasyonundan 100'ü, çözümleri tekil Prophet kütüphanesi ve tekil STL çözümü ile karşılaştırmak için test seti olarak kullanılmıştır. Elde edilen sonuçlar 100 baz istasyonu için bir Kademe-1 Avrupa mobil operatöründen alınan iki yıllık geçmiş verisi üzerinde uygulanmıştır. Yapılan literatür araştırmalarında, mobil baz istasyonlarının uzun dönemli değişim noktası farkında olarak etiketlenmesine odaklanan ve SME'lerin gözetiminde gerçek ağ verileri ile test edilen başka bir çalışma tespit edilmemiştir.

Bu özellikleri canlı ağ verilerinde, parçalı STL ayrıştırması ve Prophet'in Laplace ile bağlanımını kullanan gelişmiş doğrulukla hesaplamak için hibrit bir algoritma tasarlanmıştır. Her iki yöntemin de zayıf ve güçlü parçalarının farkındalığıyla birleştirme ve değişim noktası ve benzerlik analizi için bu birleşimden faydalanma sonucunda doğruluk değeri bu yöntemlerin tek başlarına kullanımlarının ortalamasına göre %18,6 artmıştır. Bu genel artış dışında algoritmanın problem karmaşıklığını artıran ve ayrıştırma doğruluğunu azaltan bazı özel durumlara (agresif çarpımsal mevsimsellik durumu, geçmişteki seviye kaydırma şeklindeki değişim noktaları, geçmişteki eğilim artışı şeklindeki değişim noktaları, zayıf mevsimsellik durumu veya mevsimsel olmama durumu) karşı gösterdiği performans da incelenmiştir. Tüm bu inceleme işlemleri ve işlemlerin çıktıları, canlı ağ verilerindeki sorunlu noktalar üzerinden görselleştirilmiştir. Gerekli tüm ön işleme, veri temizleme ve anormallik giderme işlemleri uygulamadan önce tamamlanmıştır. Bu uygulamada geliştirilen algoritma formül (4.1)'in ikinci eşitliğinde gösterildiği gibi çarpımsal ayrıştırmaya dayalı olarak tanımlanmıştır. İlk eşitlik toplamsal mevsimsellik ifade etmektedir.

$$Y = T + S + \epsilon \quad \text{veya} \quad Y = T.S.\epsilon \quad (4.1)$$

Burada T eğilimi veya uzun vadeli yönü, S mevsimselliği veya periyodikliği tekrarlanan bir modeli ve ϵ rastgele hatayı temsil etmektedir.

Uygulamada iki yıllık geçmiş veriler için birçok ampirik analiz tamamlanmış ve trafik verilerinde çarpımsal mevsimsel özellikler gözlemlenmiştir. Mobil ağ trafiği yüksek olduğundan durağan olmayan mevsimsel özellikler, değişim noktalarının yanı sıra hem değişim noktalarını hem de mevsimsel hareketleri birlikte değerlendirmek için mobil trafikteki mevsimsel etkiyi ortaya çıkaran bir algoritma tasarlamayı gerektirmiştir. Algoritmanın iç bölümlerinde çarpımsal özellikleri daha kolay hesaplayabilmek ve değişim noktaları varoluş kontrolleri için değerlendirme amaçlı toplamsal özelliklere dönüştürmek için log-dönüşüm uygulanmıştır. Uygulamada çok sayıda değişim noktası belirlenmiş ve Laplace $(0, \tau)$ olarak seyrek bir öncelik kullanılmıştır. τ parametresi, modelin hızını değiştirmedeki esnekliğini doğrudan kontrol etmektedir (Taylor ve Letham, 2018). Ancak uygulamada yıllar arasında önemli değişim oranlarının dağılımını karşılaştırarak otomatik değişim noktası algılama çıktıları geliştirildiğinden, çarpımsal mevsimsel değişikliklerden ayrıştırılmış ve bu, gerçek değişim noktalarını ayırt etmeye yardımcı olmuştur. Değişim noktasının eğilim üzerindeki etkisi tanımlanmış ve başarıyla ayrıştırılmıştır. Normalde bu yaklaşım, tüm çarpımsal mevsimsel değişiklikleri, önemli bir değişim oranına sahip değişim noktalarının yüksek bir ihtimali olarak işaretler ve değişim noktası algılama algoritmasına özel olarak tasarlanmadığı için gelecekteki tahminlerde gerçek değişim noktalarını tekrar eder. CUSUM log-olabilirlik oranı davranışı, sabit varyanslı bir normal dağılımlı dizisinin ortalamasındaki bir değişikliğe neden olabilmektedir (Basseville ve Nikiforov, 1993).

Bu uygulamada bir baz istasyonunu mevsimsel ($Is_Sea^{(k)} = 1$) veya mevsimsel değil ($Is_Sea^{(k)} = 0$) olarak etiketlemek için (4.2) kullanılmıştır.

$$Is_Sea^{(k)} = \begin{cases} 1, & \exists n \ni \frac{S_n^{(k)}}{T_n^{(k)}} \geq \lambda \\ 0, & \forall n \ni \frac{S_n^{(k)}}{T_n^{(k)}} < \lambda \end{cases} \quad (4.2)$$

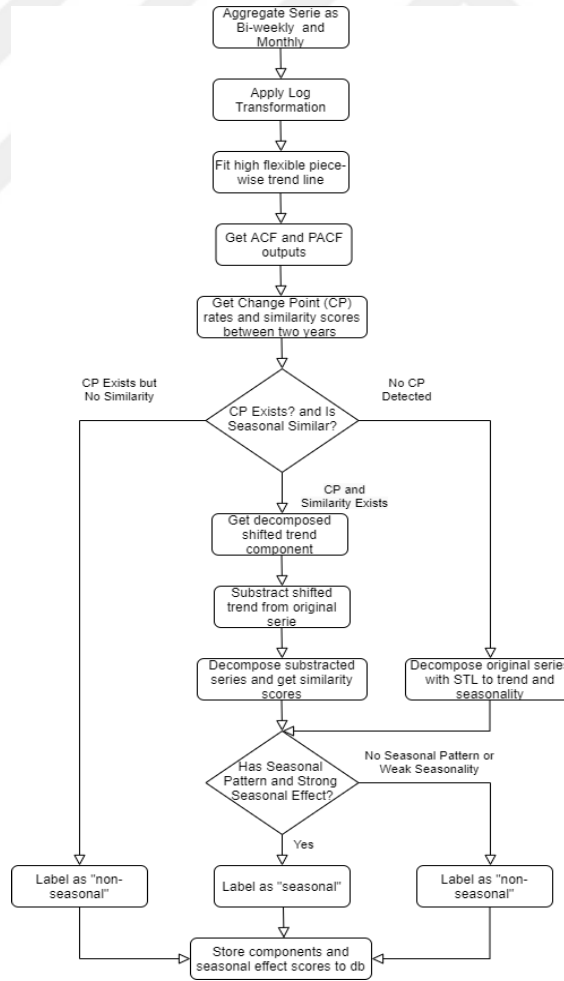
Bir baz istasyonunun bileşenlerinin (mevsimsel bileşen ve eğilim bileşeni) birbirine oranı herhangi bir zaman aralığında %30'un üzerine çıkarsa, bu baz istasyonu mevsimsel olarak etiketlenmiş, aksi takdirde daha düşük seviyelerde mevsimsellik gösterse bile mevsimsel olmayan olarak etiketlenmiştir. Ham verileri uzun vadeli mevsimsellik ve değişim noktası analizi için daha uygun hale getirmek için bazı ön işleme operasyonları uygulanmıştır. Toplanan veriler, hücre düzeyinde günlük trafik hacmi ve ilgili hücre planlama verilerinden oluşmuştur. Yanıltıcı aykırı değerlerden kurtulmak için istatistikler bir araya getirildiğinde uzun

vadeli mevsimsellik ve eğilim değişiklikleri daha net analiz edilebilmektedir. İlgili baz istasyonlarını yeni sektörel hücre genişletmeleri veya doğru zamanda mobil baz istasyon ataması ile genişletmek için, günlük çözünürlükte veri kullanımı çok anlamlı değildir. Ayrıca yıllık mevsimlik özelliklerin de korunması gerekmektedir. Bu nedenle, verileri algoritmaya beslemeden önce aşağıdaki adımlar uygulanmıştır:

- 1- **Veri Temizliği:** Veri setinde EMS sunucularındaki bazı problemlerden dolayı negatif değerli trafik hacmi verileri veya eksik hücre kimliği bilgileri gibi bazı anormal değerler bulunabilmektedir. Algoritmanın doğru bir şekilde çalışabilmesi için bu veriler temizlenmiştir.
- 2- **Eksik Veri Enterpolasyonu:** Bazı EMS sunucu kesintileri nedeniyle kimi zaman günlük trafik verilerinin eksik olması söz konusu olabilmektedir. Algoritmanın doğru bir şekilde çalışabilmesi için bu değerler, haftanın aynı günlerinin ortalaması alınarak enterpolasyon aracılığıyla tamamlanmıştır.
- 3- **Anormallik İyileştirme:** Sayaçların yanlış okunması veya çeşitli EMS problemleri nedeniyle kimi zaman beklenmedik şekilde yüksek veya düşük olan trafik değerleri söz konusu olabilmektedir. Algoritmanın doğru bir şekilde çalışabilmesi için bu değerlerin tespit edilmesi ve iyileştirilmesi gerekmektedir. Bu aykırı değerler, Mevsimsel Ölçekli Medyan Mutlak Sapma (Seasonally Scaled Median Absolute Deviation – Seasonally Scaled MAD) ile tespit edilmiş, kaldırılmış ve ikinci adımdaki gibi doğru değerlerle güncellenmiştir.
- 4- **Baz İstasyonu Düzeyinde Toplama:** Algoritmanın daha doğru bir şekilde çalışabilmesi, işleme süresinin azaltılması, bölge bazında odaklanabilmek için hücre düzeyindeki veriler baz istasyonu seviyesinde günlük verilere dönüştürülecek şekilde birleştirilmiştir. Toplama şeklinde bir birleştirme gerçekleştirilmiş ve günlük baz istasyonu trafik seviyeleri elde edilmiştir.
- 5- **Yeni Baz İstasyonlarının Kaldırılması:** Algoritmanın doğru bir şekilde çalışabilmesi için iki yıldan az geçmişe sahip genç baz istasyonları, yıllık mevsimsellik analizi için veri setinin dışında tutulmuştur.

- 6- **İşlenen Verilerin Saklanması:** Algoritmanın doğru bir şekilde çalışabilmesi için ileride yapılacak analizler için işlenen çıktılar veri tabanında saklanmıştır. Böylece tekrarlı denemelerde, aynı işlemlerin gereksiz tekrar önlenmiştir. Bazı özel Python ve SQL betikleri

Sunulan hibrit algoritmanın iş akışı Şekil 4.1’de verilmiştir. Hibrit çözümün geliştirilmesi ve test edilmesi için, akademi ve endüstride yaygın olarak kullanılan Python’da kodlanmış iki haricî kütüphane kullanılmıştır. Seyrek öncel olasılıklı Bayes bağlantımı, arka ucunda Stan kütüphanesini (Stan Geliştirme Ekibi, 2020) kullanarak zaman içindeki değişikliklerin oranını tespit eden Facebook’un Prophet kütüphanesinin (Taylor ve Letham, 2018; Erp vd. 2019) üzerine inşa edilmiştir. Diğer kullanılan kütüphane ise, son ayrıştırma adımında STL algoritması için kullanılan Statsmodel kütüphanesidir. (Seabold, vd., 2010).



Şekil 4.1. Algoritmanın Ana İş Akışı

Algoritma, günlük dalgalanmalardan kurtulmak için günlük baz istasyonu trafik hacmi verilerini iki haftalık ve aylık seriler halinde birleştirerek başlar ve uzun vadeli yatırım kararları için daha kritik olan yıllık mevsimsel özelliklere daha fazla odaklanır. Her iki birleştirilmiş seri, mobil radyo ağlarının çarpımsal mevsimsel özellikleri nedeniyle log dönüşümüne tabi tutulmuştur. Dolayısıyla, baz istasyonları üzerindeki mevsimsel etki, eğilimin bir fonksiyonudur ve açıklayıcı analizin bir sonucu olarak çarpımsal etki göstermektedir. Değişim oranı dağılımındaki potansiyel anormal davranışları elde etmek için logaritmik dönüştürülmüş trafik serilerinde Prophet modelini kullanarak oldukça esnek bir Bayesci Lasso bağlanımı uygulanmıştır. Limitli mevsimsellik sayesinde değişim oranı eğilim eğrisi içerisinde açıklanmakta, böylelikle değişim oranları açısından mevsimsel değişimler de geri alınmaktadır. İki yılın her ayı için, değişim noktası dağılımının daha iyi araştırılması amacıyla eğri yerleştirildiğinde belirli aralıklarla potansiyel bir değişim noktası varlığı da varsayılır ve Laplace dağılımındaki bir seyrek önsel ile bunlardan zayıf olanların yutulması sağlanır. Hem iki haftalık hem de aylık seriler, eğilim ve mevsimsel bileşen değişikliklerini kontrol etmek için STL modeliyle ayrıştırılır. Bu göstergelerin her ikisi de trafik serilerinin değişim noktaları içerip içermediğine ve içeriyorsa bile mevsimsel bir özellik gösterip göstermediğine karar vermek için kullanılır. Algoritma içinde bu amaçla geliştirilmiş iki adet gösterge bulunmaktadır. Bunlardan ilk gösterge bir değişim noktasına işaret ediyor ve ikinci gösterge de mevsimsel karakteristik olarak pozitifse bu hücrenin geçmişinden değişim noktası barındıran eğilim bileşeni çıkartılmalıdır. Bunun için STL ayrışmasından önce değişim noktası yumuşatma amacıyla içerideki değişim noktasını kapsayacak ve ana eğilim bileşeninden çıkarılacak esnek bir eğilim eğrisi oturtulmalıdır. Bu, değişim noktası etkisinin mevsimsel hareketlerden ayrılmasına ve eğilim bileşeni içinde açıklanmasına yardımcı olur. Prophet kütüphanesi, yüksek çarpımsal mevsimsellik etkisini potansiyel bir değişim noktası olarak etiketleme eğilimindedir. Benzeri şekilde STL kütüphanesi de değişim noktalarını mevsimsellik bileşeni içinde açıklama eğilimindedir. Bunlar bu iki modelin tespit edilen önemli zayıf özellikleridir ve bahsi geçen ayrıştırmaların hatalı olmasına sebebiyet vermektedir. Ayrıca Prophet kütüphanesi, mevsimsel bileşeni Fourier serisi ile açıklamaktadır. Bu da daha düşük değerler içeren eğilim bileşeninin ilk senesini, ikinci yıl için de atama eğilimindedir. Bu, sonraki yılın mevsimsel artış seviyesinin zirvesinin yanlış uydurulmasına neden olur. Bu nedenle özellikle çarpımsal mevsimselliğin güçlü olduğu bazı hücrelerde kapasitenin ne kadar aşılacağını Prophet ile tespit etmek güçleşmektedir.

Uygulamanın sonucunda elde edilen algorithmada beş adet çıktı üretilmektedir:

- 1- (4.7) ile hesaplanan Mevsimsel Güç Endeksi (Seasonal Power Index - SPI) değerleri açısından olumlu aylık mevsimsel etki gücü,
- 2- mevsimsel dönemlerin gün cinsinden uzunluğu,
- 3- baz istasyonunun (4.9)'a göre genel SPI değeri,
- 4- geçmişteki değişim noktalarının varlığı,
- 5- maksimum SPI dönemini SME'lerin %30 eşik tanımıyla karşılaştırarak baz istasyonunun “mevsimsel” veya “mevsimsel değil” olarak etiketlenmesi.

Günlükteki potansiyel değişim noktalarının varlığını ve mevsimselliğini belirlemek için Algoritma 4.1'deki kontroller uygulanır.

Algoritma 4.1 *değişim noktası kontrolleri*

```

procedure exec_change_ratio_controls (bw_series, sig_thr, rat_thr)
  set c_prio_sc  $\leftarrow$  1.5, c_rng  $\leftarrow$  1.0, cp_in_y1  $\leftarrow$  false, cp_in_y2  $\leftarrow$  false
  set model  $\leftarrow$  init Prophet with no_sea, c_prio_sc, c_rng
  call model. fit with bw_series
  set rc_y1[ ], rc_y2[ ]  $\leftarrow$  split model.rate_changes[ ] yearly
  set max_y1  $\leftarrow$  Max(Abs(rc_y1)), max_y2  $\leftarrow$  Max(Abs(rc_y2))
  set num_sig_y1  $\leftarrow$  0.1, num_sig_y2  $\leftarrow$  0.1
  for each change ratio c1, c2 in rc_y1[ ], rc_y2[ ]
    if Abs(c1) > sig_thr then
      increment num_sig_y1
    end if
    if Abs(c2) > sig_thr then
      increment num_sig_y2
    end if
  end for
  if num_sig_y1 / num_sig_y2 > rat_thr or max_y1 / max_y2 > rat_thr then
    set cp_in_y1  $\leftarrow$  true
  elif num_sig_y2 / num_sig_y1 > rat_thr or max_y2 / max_y1 > rat_thr then
    set cp_in_y2  $\leftarrow$  true
  return cp_in_y1, cp_in_y2
end procedure

```

Prophet'in kendisi, iki tür büyüme modeline sahiptir:

- doğrusal olmayan doymuş büyüme ve
- değişim noktaları ile doğrusal eğilim.

İkinci model, genelleştirilmiş bir toplamsal model olarak değişim noktalarını adresleyebilmek için bir ölçek parametresine sahiptir. Bu uygulamada, iki yıl arasındaki değişim oranlarının karşılaştırılmasına da odaklanılmıştır. Her iki yıl birbirine benzer mi veya mevsimsel değişimleri ikisi de içeriyor mu araştırılır. Bu adımda, (4.4)'te verilen büyüme fonksiyonu, büyüme oranı olarak k , oran ayarlamaları olarak δ ve ofset parametresi olarak m 'den oluşur.

$$\delta_j \sim \text{Laplace}(0, \tau) \quad (4.3)$$

$$g(t) = (k + a(t)^\tau \delta) t + (m + a(t)^\tau \gamma) \quad (4.4)$$

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t, \quad (4.5)$$

$$y(t) = g(t) + \epsilon_t, \quad (4.6)$$

s_j , $j = 1, \dots, S$ zamanlarında S değişim noktası olduğu varsayılırsa, δ ve δ_j gibi oran ayarlamalarının vektörü s_j anındaki değişim oranını temsil edebilir. Büyüme modeline esnek bir parametre atandıktan sonra (varsayılan Prophet değeri 0,05'tir), model, değişikliklerin net oranını elde etmek için bu aşamada bir seyrek öncül kullanır. (4.4)'de olduğu gibi bu öncül ikili üstel (diğer bir ifadeyle Laplace) dağılımı olduğu için seyrek öncül olarak adlandırılır. Burada τ parametresi, Laplace formu nedeniyle modelin esnekliğini doğrudan kontrol eder. Bu nedenle, bu önceki değeri artırmak, eğri uydurmayı daha esnek hale getirir. Mevsimsel etiketli baz istasyonları üzerindeki ampirik analize dayanarak, bu değer $\tau = 1,5$ ve değişim noktası aralığı 1,0 olarak belirlenerek, geçmişte logaritmik dönüştürülmüş trafik serilerinin tamamı adreslenebilmiştir. Bulunan her değişim noktasının 0,8'in üzerindeki değeri, trafik hacminin güçlü çarpımsal mevsimsel yukarı ve aşağı hareketlerini de kapsayan önemli değişim noktaları olarak tanımlanmaktadır. Birinci ve ikinci yıldaki değişikliklerin oranı, önemli oranların sayısı ve oranların maksimum mutlak değerleri ile karşılaştırılmıştır. Bu göstergelerin her ikisinde de oran dağılımının farklı olması durumunda potansiyel bir değişim noktasının var olabileceği kanısına varılır. Test veri setinin dışında kalan çarpımsal mevsimsel baz istasyonları

da dahil olmak üzere deęişim noktası için ampirik bir analiz yapılmıř ve bu eřik 3 olarak tanımlanmıřtır. P mevsimsel dönem ve $X = \{X_{S1}, X_{S2}, \dots, X_{SN} \mid N \leq P\}$ mevsimsel bileřen endeksi ise ve $F_S(X)$, $F_T(X)$ sırasıyla mevsimsellik ve eęilim bileřenlerinin ayrıřtırılmıř fonksiyonları olur. Bunların aylık deęerleri kullanarak (4.7)'deki formül ile bir ayın SPI olarak mevsimsel etkisinin gücü hesaplanabilir. Burada $X = M$ 'dir ve M belirli bir ayı ifade etmektedir.

$$SPI_M = \frac{F_S(M)}{F_T(M)} \quad (4.7)$$

(4.8)'de Jaccard benzerlik puanının (Jaccard, 1912) tamamlayıcısı olarak Jaccard mesafesini kullanarak, mevsimsel benzerlikleri olan baz istasyonlarını aylık SPI benzerlięi açısından karřılařtırmaya yardımcı olan bařka bir gösterge oluřturulmuřtur.

$$d_j(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \quad (4.8)$$

Veri setinde yüksek mevsimsel etki gösteren bölgelerde yer alan baz istasyonları vardır. Ancak bunların bazılarının geęmiřlerinde seviye deęiřimi bazlı deęiřim noktaları da mevcuttur. Deęiřim noktalarına raęmen mevsimsel özelliklerini korudukları için sunulan algoritma, Algoritma 2'deki akıř ile SPI'lar açısından karakteristik benzerliklerini de kontrol eder.

Algoritma 2 mevsimsel karakteristik kontrolleri

```

procedure exec_sea_char_control (m_series, delta_thr, spi_sim_thr, jacc_thr)
  set num_similars  $\leftarrow$  0, num_match_dir  $\leftarrow$  0, num_match_pos  $\leftarrow$  0
  set has_sea_pattern  $\leftarrow$  false
  set model  $\leftarrow$  init STL with yearly_sea
  set trend_m, sea_m  $\leftarrow$  call model.fit with m_series get decompose_res
  set spi_m[len(m_series)]  $\leftarrow$  null
  for each component tre_of_month, sea_of_month in trend_m, sea_m
    add spi_m  $\leftarrow$  sea_of_month / tre_of_month
  end for
  set spi_y1[, spi_y2[]  $\leftarrow$  split spi_m[, ] yearly
  for each index ix in spi_y1, spi_y2
    set spi_m_y1  $\leftarrow$  spi_y1[ix], spi_m_y2  $\leftarrow$  spi_y2[ix]
    set spi_m_y1_prev  $\leftarrow$  spi_y1[ix - 1], spi_m_y2_prev  $\leftarrow$  spi_y2[ix - 1]
    if Abs(spi_m_y1 - spi_m_y2)  $\leq$  delta_thr then
      increment num_similars
    end if
    if (spi_m_y1_prev - spi_m_y1) * (spi_m_y2_prev - spi_m_y2)  $>$  0 then
      increment num_match_dir
    end if

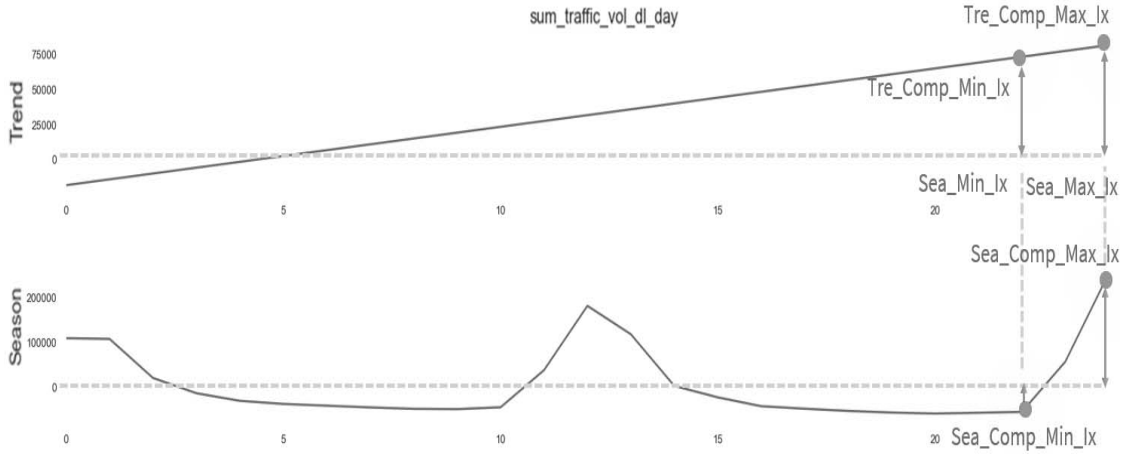
```

```

    if spi_m_y1 * spi_m_y2 > 0 then
        increment num_match_pos
    end if
end for
set jacc_sc_dir ← num_match_dir / 12, jacc_sc_pos ← num_match_pos / 12
if num_similars ≥ spi_sim_thr and jacc_sc_dir ≥ jacc_thr
    and jacc_sc_pos ≥ jacc_thr then
        has_sea_pattern ← true
    return has_sea_pattern
end procedure

```

İki yıllık aylık SPI'leri hesaplamak için öncelikle STL ayrıştırma algoritması ile log dönüşümlü trafik serileri ayrıştırılmıştır. Dolayısıyla değişim noktasının varlığı mevsimsel bileşeni bozsa dahi trafik serisi eğilim ve mevsimsel bileşenlere sahiptir. Ardışık ayları arasındaki SPI değişimleri, yukarı ve aşağı yönlü olmasına göre ikili olarak eşleştirilir ve her iki ikili yön vektörleri ve SPI değişim güçleri olarak iki yıl arasında karşılaştırılır. Yönlü karşılaştırma için Jaccard mesafesi, iki yılın aynı ayları benzer yönlerde hareket ediyorsa elde edilir. Ayrıca, bir başka gösterge olarak, iki yıllık aylık SPI'ların değişimi, bir delta eşiği ile birbirine benzemektedir, ardından baz istasyonu da mevsimsel özelliklere sahip olarak etiketlenir. Aylık değişimler yıllar arasında karşılaştırılmış ve SPI değişim yönü (yukarı yönlü mü yoksa aşağı yönlü mü?) ve pozisyon (eğilim bileşenin altında veya üstünde mi?) açısından benzerlikleri hesaplanmıştır. Her iki durum da Jaccard mesafesi skoru elde edilecek şekilde karşılaştırılmıştır. Bir baz istasyonunun aylık ayrıştırılmış benzerlik örüntüsü, %25'in üzerinde ($spi_sim_thr = 0,25$) ve %50'den az Jaccard mesafesi ($jacc_thr = 0,5$) gösteriyorsa, o zaman uygulamada değişim noktasından bağımsız olarak potansiyel bir mevsimsel desen kabul edilmiştir. Değişim noktası göstergesi de pozitifse, o zaman ilgili değişim noktasını da kapsayan esnek bir eğilim bileşeni oturtulmuş ve STL ayrışmasından önce orijinal seriden çıkarılmıştır. Bu nedenle, eğilim değişiminin etkisi mevsimsel bileşenden izole edilerek Prophet'in zayıf yanı iyileştirilebilmiştir. Daha önce belirtildiği gibi, aylık ve iki haftalık serilerin STL ayrıştırması ayrı ayrı yapılmıştır. Mevsimsel güç endeksi, (4.7) ile aylık olarak hesaplanmıştır. Ancak ortak bir mevsimsel etki puanını hesaplamak için (4.9)'da belirtilen daha genel başka bir formül kullanılmaktadır. Bu formül, aynı dönemlere denk gelen eğilim değerleri ile mevsimsel maksimum ve minimum değerlerinin oranından oluşmaktadır. Şekil 4.2'de baz istasyonu mevsimsellik güç endeksi parametreleri gösterilmektedir.



Şekil 4.2. Baz İstasyonu Mevsimsel Güç Endeksi (SPI) Parametreleri

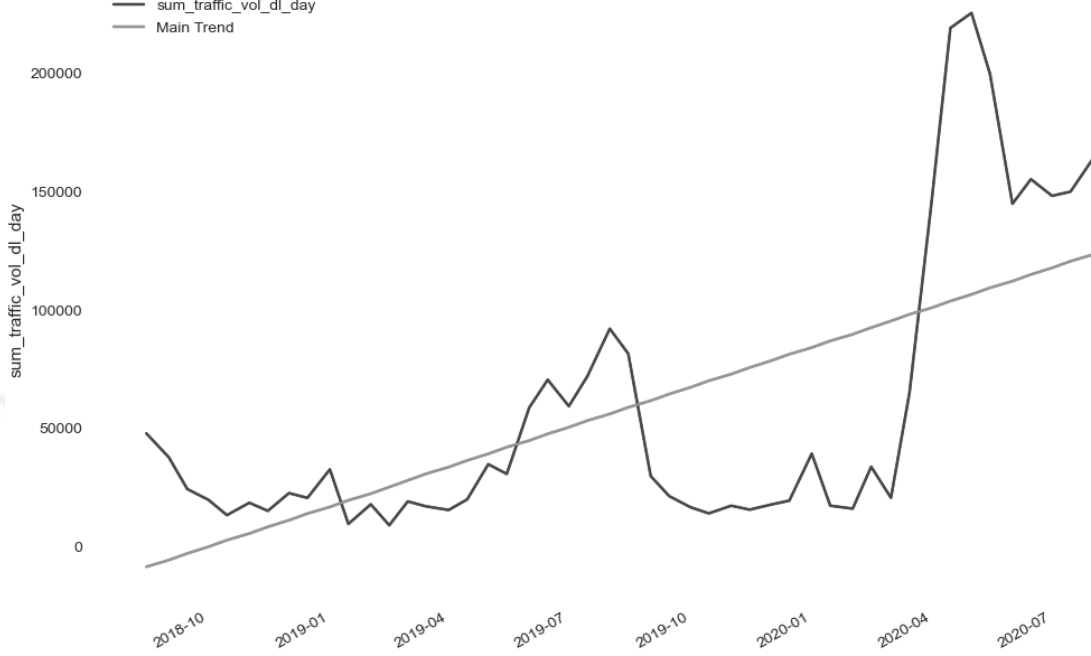
Yine, P mevsimsel dönem ve $X = \{X_{S1}, X_{S2}, \dots, X_{SN} \mid N \leq P\}$ mevsimsel bileşen endeksi ise $F_S(X)$, $F_T(X)$ sırasıyla mevsimsellik ve eğilim bileşenlerinin ayrıştırılmış işlevleridir. Bunlara göre, (4.9)'da bir baz istasyonuna ait genel SPI yani mevsimsel etkinin ilgili baz istasyonunda ne kadar güçlü olduğu hesaplanmıştır.

$$SPI = \frac{Sea_Comp_Max_Ix + |Sea_Comp_Min_Ix|}{Tre_Comp_Max_Ix + Tre_Comp_Min_Ix} \quad (4.9)$$

Algoritma geliştirildikten sonra baz istasyonu trafiğinin farklı durumları için hibrit algoritmanın saf Prophet ve STL tabanlı yaklaşımların zayıf yanlarını nasıl iyileştirdiği incelenmiştir. Sonunda SME'ler tarafından hazırlanan test seti için algoritmanın doğruluk sonuçları ve tüm ağ için elde edilen sonuçlar paylaşılmıştır.

Agresif Çarpımsal Mevsimsellik Durumu: Öngörülmesi gereken ilk ve yaygın durum, bazı baz istasyonları için agresif çarpımsal mevsimsellik etkisidir. Şekil 4.3'te görüldüğü gibi, bu tür baz istasyonlarında eğilim bileşenin bir fonksiyonu olarak hareket eden bir mevsimsel bileşen gözlemlenmektedir. Şekilde 4.3'te ana eğilim bileşeni, doğrusal eğilim büyümesi hakkında fikir sağlamak açısından paylaşılmış ve kırmızı olarak gösterilmiştir. Temel olarak, baz istasyonu trafik eğiliminin gerçek doğrusal olmayan değerleri SME değerlendirmelerine dayalı olarak uygun şekilde belirlenmelidir. Bu nedenle eğilimin nihaî ayrışmasını, Prophet'in lojistik büyümesinden ziyade STL ayrıştırmasından elde edilen lojistik büyüme ile uyumlu kılmak baz istasyonları için daha uygun görünmektedir. Bu örnekte baz istasyonu trafiği, her yıl Nisan, Mayıs, Haziran dönemleri civarında zirve yapmakta ve mevsimselliğin agresifliği

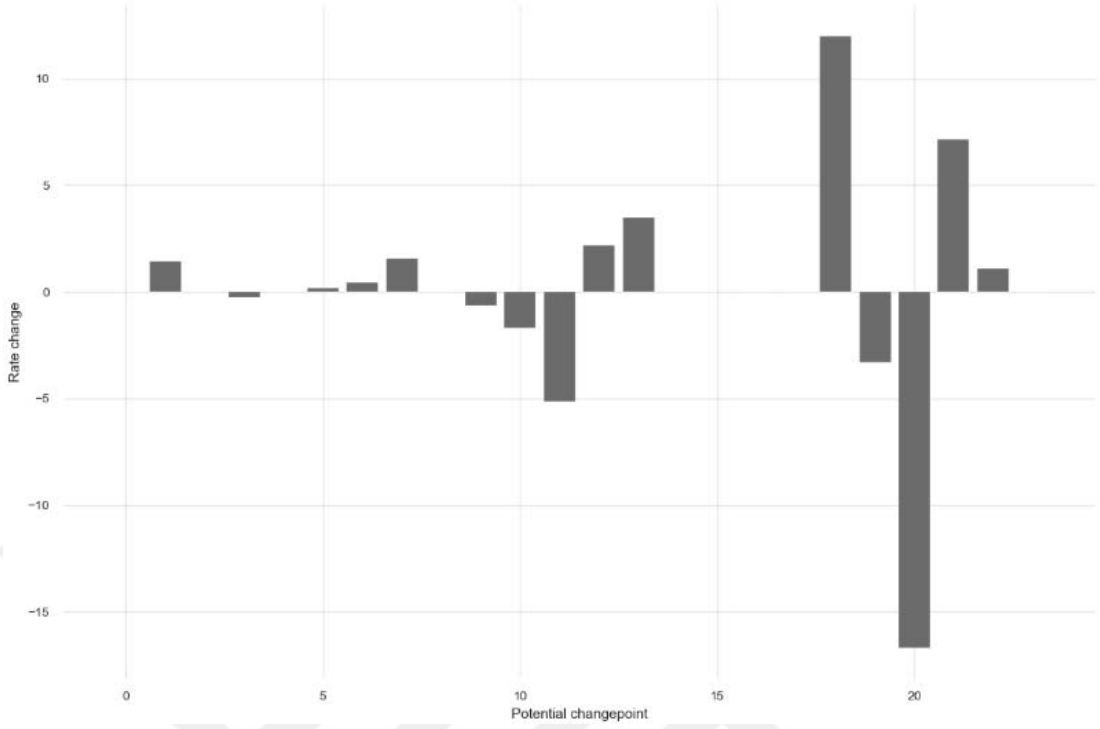
mevcut kaynak blokları nedeniyle zamanın bir fonksiyonu olarak doyma noktasına kadar artan bir eğilimin fonksiyonu olarak davranış göstermektedir.



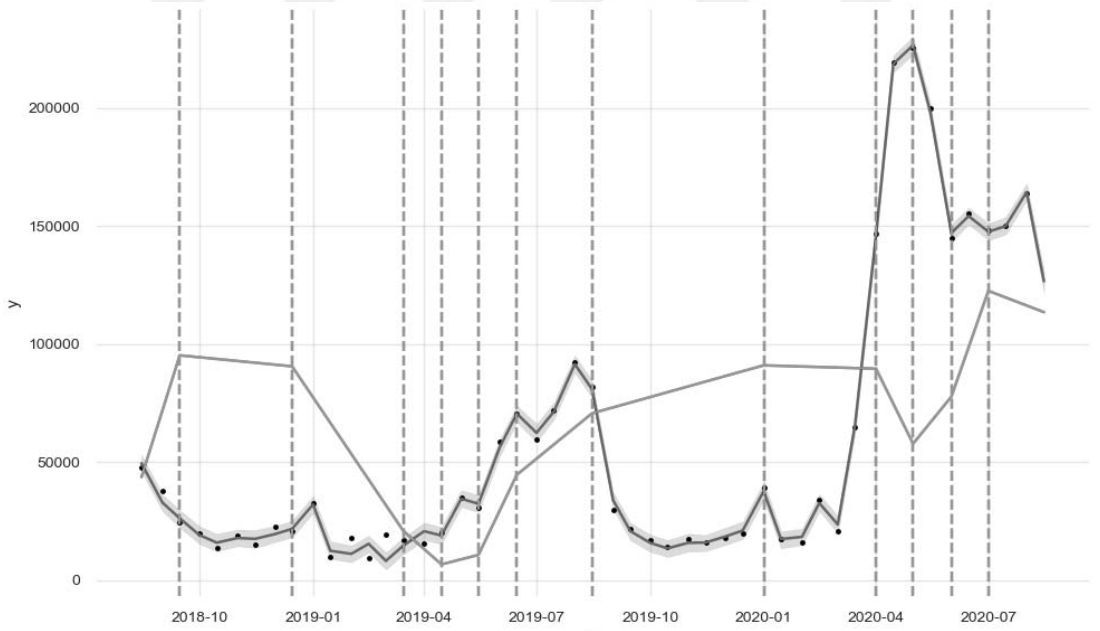
Şekil 4.3. Çarpımsal Mevsimselliğe Sahip Bir Baz İstasyonu Örneği

Değişim noktalarını tespit edebilmek için, daha yüksek esnekliğe sahip bir Laplace öncül parametresi belirlemek bu agresif çarpımsal mevsimsel artışın, güçlü değişim noktaları olarak yanlış ifade edilmesine neden olabilmektedir. Prophet kütüphanesinde oluşturulan çarpımsal yıllık mevsimsellik modeli olarak Laplace (0, 1,5) kullanıldığında, değişim oranlarının zamana göre dağılımı Şekil 4.4 ve Şekil 4.5'teki gibi gözlemlenmektedir.

Herhangi bir potansiyel değişim noktasını yakalamak için eğilim bileşeni sapmalarını gereksiz yere çok esnek tutmak, mevsimsel hareketlerin potansiyel değişim noktaları olarak işaretlenmesine neden olur ve bu da, yıllık çarpımsal mevsimselliği uydurmak için beklenmedik değişim noktaları tespitine neden olur. Burada mevsimsel bileşen, Fourier serisi uydurmanın bir çıktısıdır. Bu örneğe özel olarak, çarpımsal yıllık mevsimsellik modeline daha uygun bir Prophet modeli için, varsayılan değer olan Laplace (0, 0,05) gibi daha az esnek bir öncül parametre değeri belirlemek işe yararmaktadır.



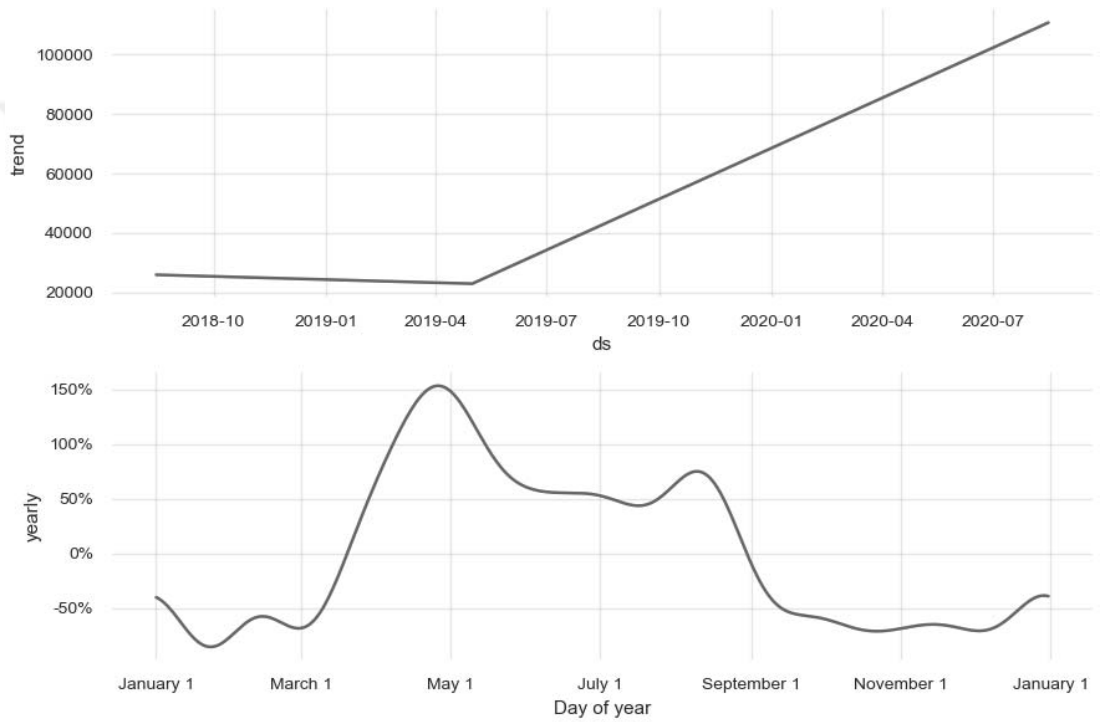
Şekil 4.4. Çarpımsal Mevsimsel Esnek Modelin Oransal Değişim Dağılımı



Şekil 4.5. Çarpımsal Mevsimsel Esnek Modelin Değişim Noktası Konumları

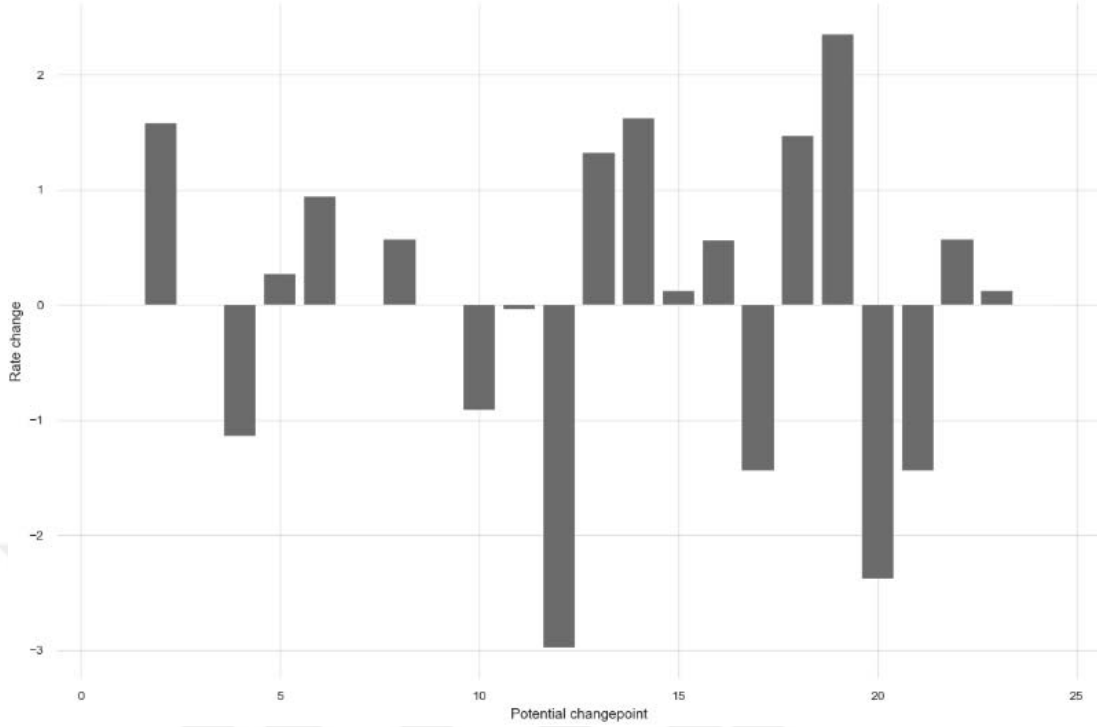
Daha az esnek eğilim bileşeni ile ayrıştırılmış bileşenler Şekil 4.6'da görselleştirilmiştir. Değişim noktası önceliklerinin ayarlanması, eğilim ve mevsimsel bileşenler arasında trafik miktarının doğru şekilde dağıtılmasında kritik bir rol oynamaktadır. Şekil 4.6'daki tek bir değişim noktası, genel serinin ortasına yakın bir yerde bulunur ve eğilim üzerindeki eğim

değişikliğinin ayrıştırılmış bileşenlere bakıldığında daha iyi uyduğu gözlemlenmektedir. Ancak daha düşük veya varsayılan öncül Laplace parametrelerini tutmak, eğilim değişikliklerini göz ardı etme eğilimindedir. Bu da modeli gerçek değişim noktalarına karşı daha az hassas hale getirmektedir. Otomatik bir çözüm, agresif olanlar da dahil olmak üzere çarpımsal mevsimsel hareketleri de diğer bazı nedenlerin neden olduğu seviye değişimlerinden otomatik olarak ayırt edebilmelidir.



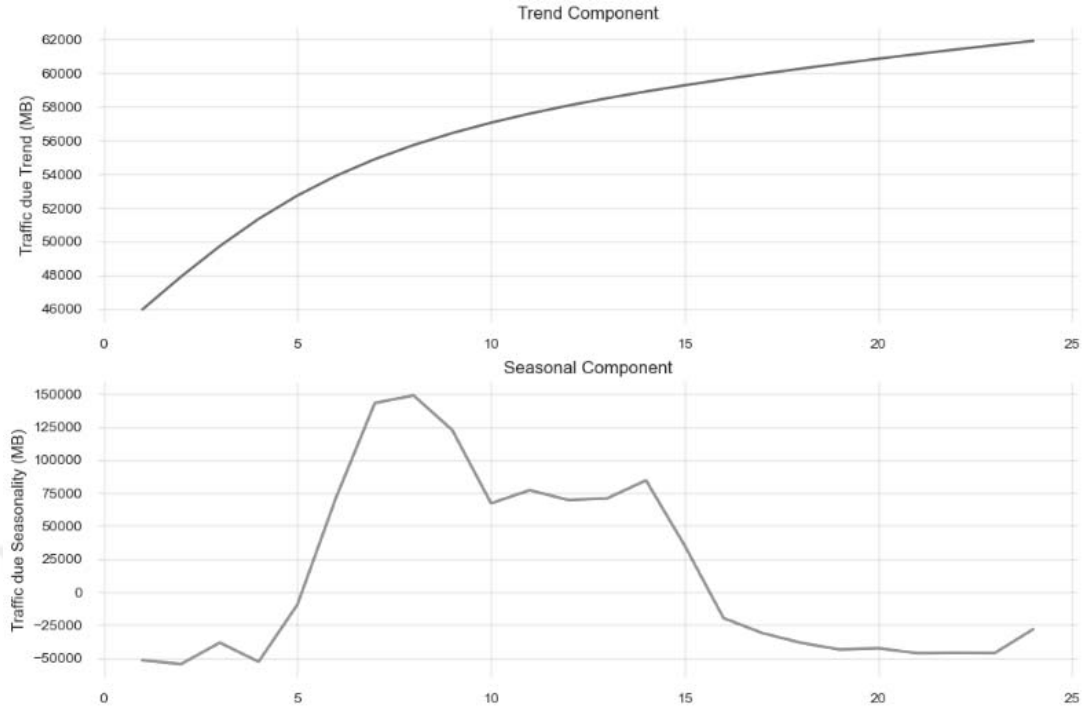
Şekil 4.6. Çarpımsal Mevsimsel Davranışla İlgili Daha Az Esnek Modelin Bileşenleri

Orijinal serinin log-dönüşümü, çarpımsal mevsimsel etkiye sahip baz istasyonlarının her iki yıllık değerleri arasındaki mesafeyi Jaccard mesafesi açısından azaltmaktadır. Bu da, iki yıl arasındaki hareket benzerliklerini karşılaştırmayı kolaylaştırmaktadır. Log-dönüşüm seriyeye uygulandığında ve tüm olası değişim oranlarını almadan önce esnek olarak bırakıldığında, Şekil 4.7'de görselleştirilmiş oranların dağılımı elde edilmektedir. İki yıl arasındaki değişim oranlarının dağılımı miktar ve pozisyon olarak daha benzerdir ve daha düşük Jaccard mesafeleri anlamına gelmektedir.



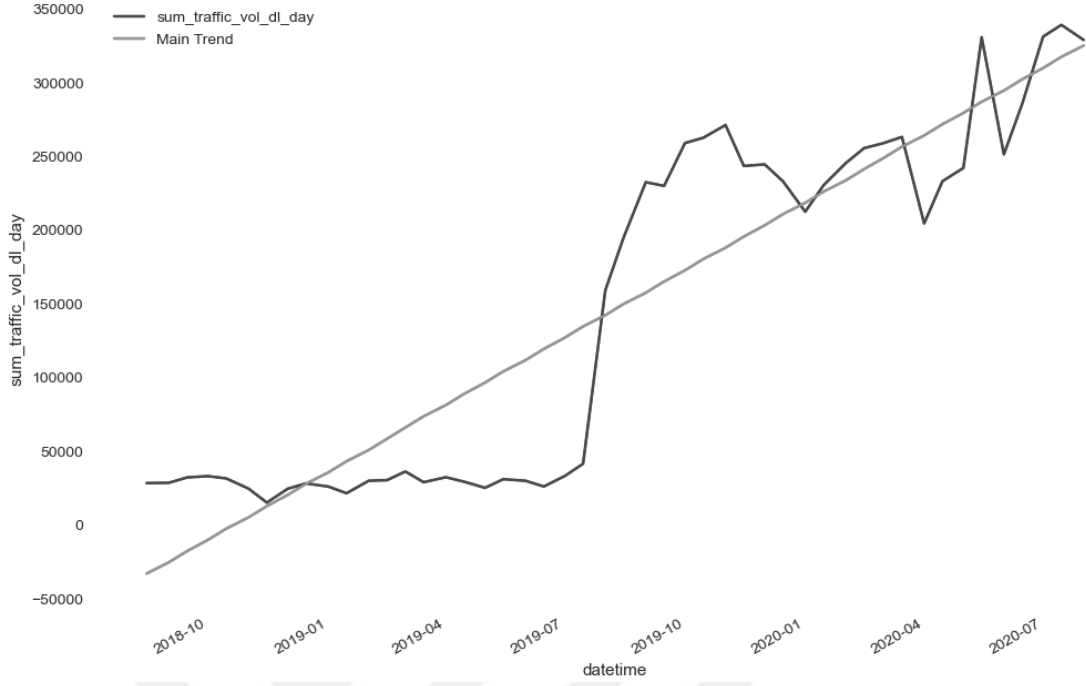
Şekil 4.7. Çarpımsal Mevsimselliğin Günlüğe Dönüştürülmüş Trafik Verileri Üzerinde Daha Az Esnek Modeldeki Oransal Değişim Dağılımı

Algoritma 1 bazı değişim noktası kontrolleri için, sonraki yıllardaki oran değişiklikleri arasında önemli bir fark olmadığından, bazı baz istasyonu için tek başına yanlış sonuçlar döndürebilir. Her iki yılda da benzer tutarlarda sırasıyla 2,7 ve 2,3 olan beş önemli oran değişikliği mevcuttur. Ayrıca, aylık hesaplanan SPI değerleri için benzerlik puanlarının da yüksek olduğu gözlemlenmektedir. Algoritma 2 ile açıklanan Jaccard mesafeleri, iki yıl arasında SPI yön değiştirme, pozisyon değiştirme ve aylık SPI miktarı karşılaştırmaları için %38, %12 ve %38 olarak hesaplanmıştır. Bu nedenle, Jaccard mesafesine göre değişim noktalarının varlığı ve değişim benzerlikleri için müteakip kontroller, mevsimsellik olduğunu ancak belirtilen baz istasyonu için değişim noktası olmadığını ifade etmektedir. Değişim noktaları varlığında algoritma, seviye değişimini kapsayacak şekilde esnek bir eğilim eğrisi oluşturmaya çalışmakta ve doğru mevsimsel ayrıştırma için bunu orijinal seriden çıkarmaktadır. Doygunluk eğilimi bileşeniyle çarpımsal mevsimsel için algoritmanın bileşenleri Şekil 4.8’de verilmiştir.

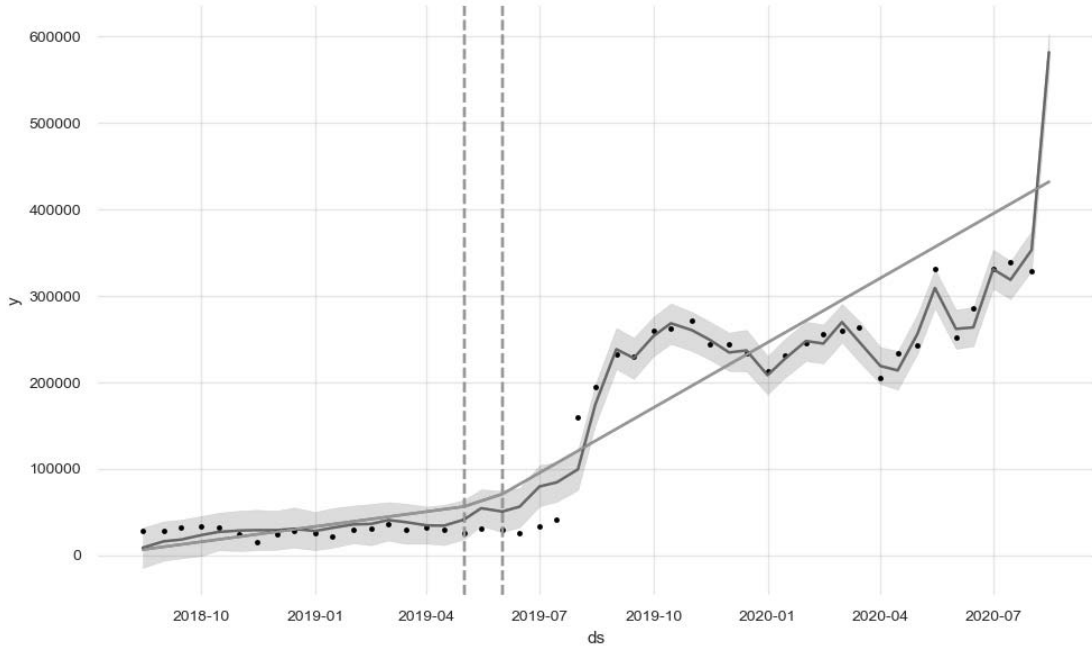


Şekil 4.8. Doygunluk Eğilimi Bileşeniyle Çarpımsal Mevsimsellik İçin Ayrıştırılmış Bileşenler

Geçmişteki Seviye Kayması Bazlı Değişim Durumları: Mobil ağlar, özellikle yüksek yoğunluk bölgeleri için çok hızlı büyümektedir. Bu nedenle, yeni konuşlandırılan veya kapatılan baz istasyonları, SME'lerin beklediği gibi komşu baz istasyonlarında önemli eğilim değişikliklerine neden olabilir. Kapsama optimizasyonu için anteni yukarı eğme veya aşağı eğme eylemleri ya da servis parametre konfigürasyonları gibi hücrelerin yapılandırma değişiklikleri gibi başka değişim nedenleri de olabilir. Bundan dolayı, otomatik mevsimsel etki analizi tasarlanırken, bu tür seviye değişimleri dikkate alınmalı, özerk olarak tespit edilebilmeli ve trafik hacminin mevsimsel hareketlerinden ayırt edilebilmelidir. Eğilim açısından kaydırılmış seviyeye sahip örnek bir baz istasyonu Şekil 4.9'da ve 4.10'da görselleştirilmiştir. Düşük önsel parametrelili ve değişim noktalı (örneğin Laplace (0, 0,05)) parça bazında Bayes eğrisi uydurma, şekil 4.10'da görselleştirildiği gibi bu değişim noktasını yakalamak için yeterli olamamıştır. Belirtilen baz istasyonunun geçmişi, Temmuz 2019'dan sonra güçlü seviye kayması içermektedir. Ancak önsel parametre düşük bir değere ayarlandığından, eğrinin parça bazında eşlenmesi yalnızca birkaç değişim noktasını yakalar. Dolayısıyla, değişim noktaları Mayıs civarında yanlış yerleştirilmiş ve bu da trafik değişikliğinin mevsimsel bileşene atfedilmesine neden olmuştur.



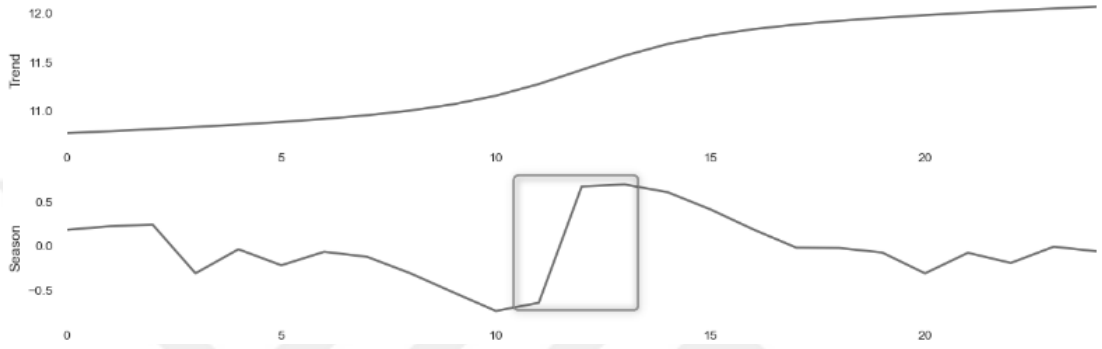
Şekil 4.9. Seviye Kayması Nedeniyle Değişim Noktası Oluşmuş Baz İstasyonu Örneği



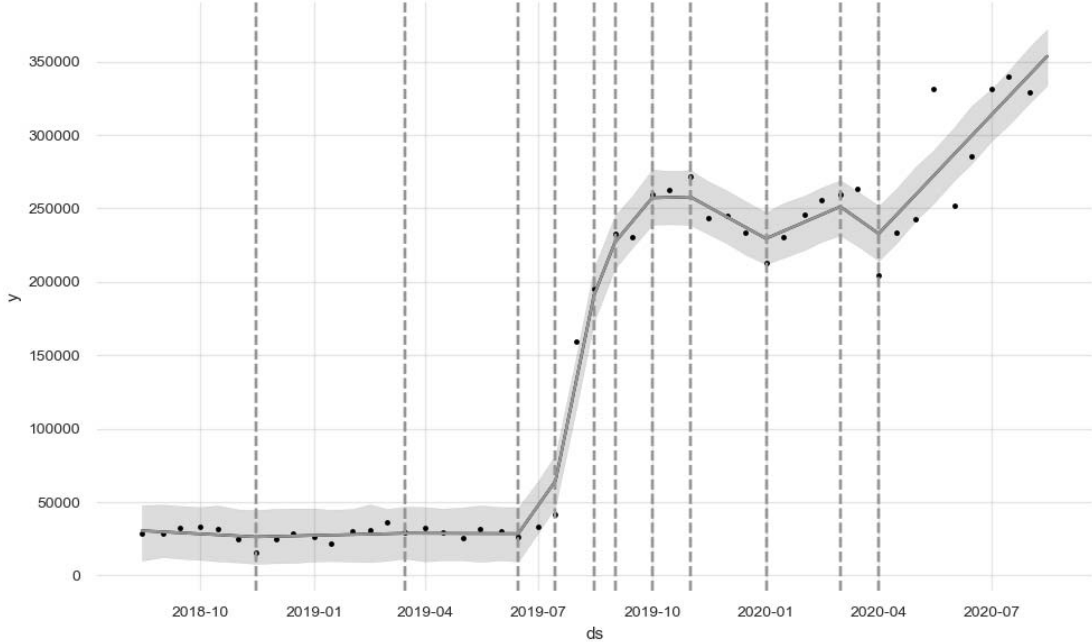
Şekil 4.10. Çarpımsal Mevsimsel Davranışta Daha Az Esnek Modelde Değişim Noktası Konumları

Serilerin STL tabanlı ayrışması için de benzer hatalı ayrıştırma çıktısı gözlenmiştir. Şekil 4.11’de görüldüğü gibi, iki yıllık geçmiş bilgisi STL ile ayrıştırıldığında (mevsimsel derece= 1, eğilim derecesi= 0), trafikteki anormal eğilim değişimi yanlış bir atamayla mevsimsel

bileşen kapsamında açıklanmıştır. Bu değişimi mevsimsellik ile açıklayan sadece STL değildir. Yanlış değişim noktası seçimi durumunda, Prophet'in mevsimsellik bileşeni de oldukça baskındır. Bu nedenle, önceden doğru değişim noktasının tespiti için uygun esneklik parametresini belirlemek kolay değildir. Örneğin bu özel durum için daha yüksek esneklik (örneğin Laplace (0, 1,5)) çok daha iyi performans gösterir ve Şekil 4.12'de gösterildiği gibi trafiği nispeten daha makul bir formatta eğilim bileşenindeki değişime adresler.



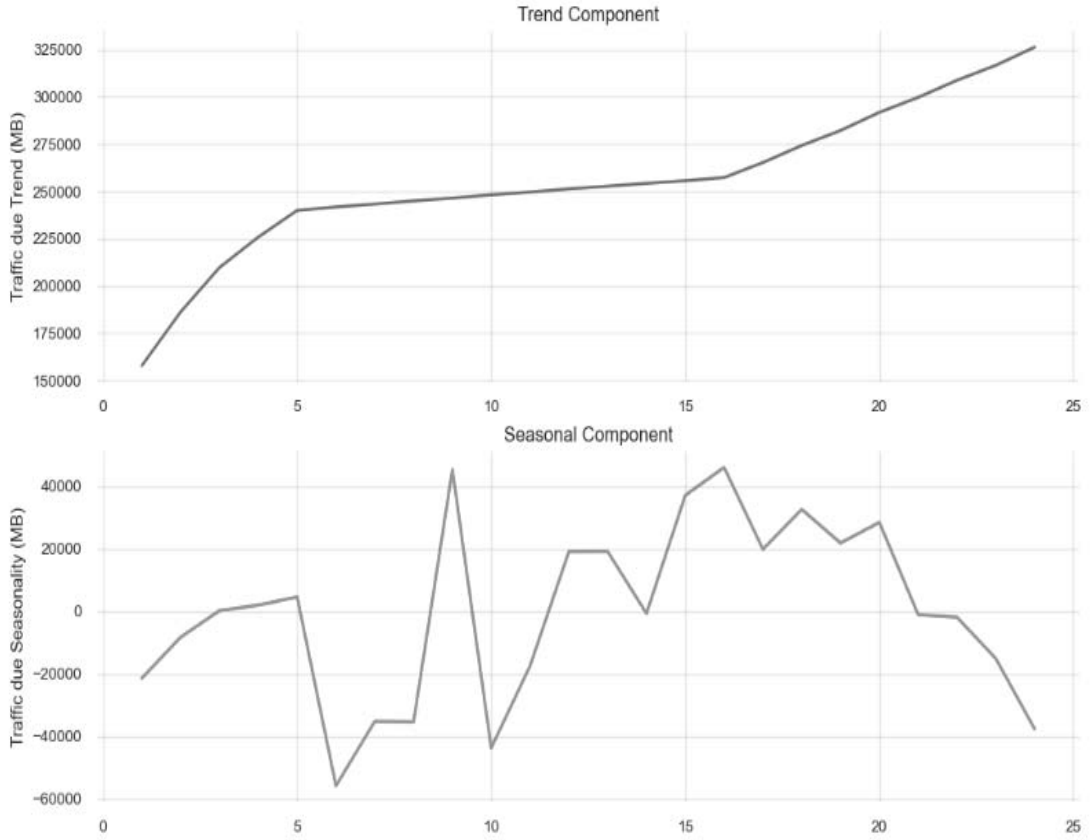
Şekil 4.11. Eğilim Değişiminin Mevsimsel Bileşene Adreslenmesiyle STL Ayırıştırmasının Hatalı Hale Gelmesi



Şekil 4.12. Değişim Noktası Davranışında Daha Esnek Modelin Bulduğu Değişim Noktası Konumları

Tez çalışması kapsamında geliştirdiğimiz hibrit çözüm, değişim noktasının potansiyel konumunu kontrol eder ve ayırıştırma için geçmişin doğru bölümlerini dikkate alır. Dolayısıyla,

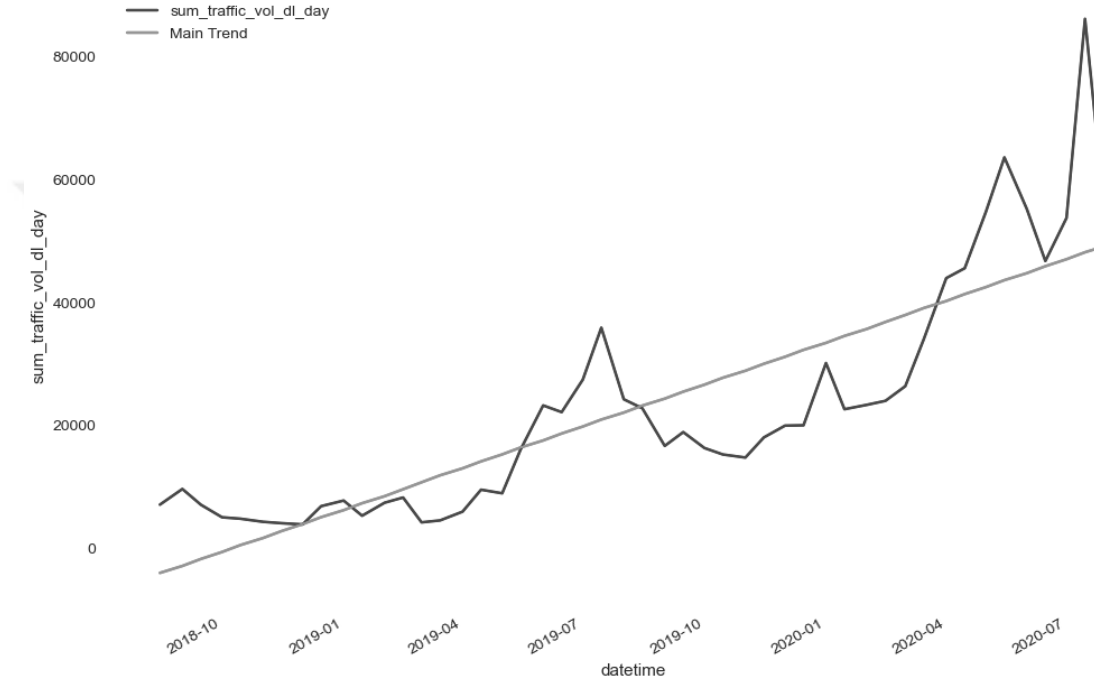
Şekil 4.13'te görüldüğü gibi değişim gerçekleşikten sonraki son yılın davranışını kendisine referans alır ve baz istasyonunun bileşenlerini ayrıştırmada en güncel davranışlar değerlendirilmiş olunur. Baz istasyonunun eski karakteristiği önemini yitirmiş olur. Nitekim önceki yılın karakteristiği tahminleme için değişim noktası sonrası değersizleşmiştir. 2020 yılına ait nispeten daha yumuşak bir pozitif eğilim bileşeni ve bunun üzerindeki mevsimsel bileşen başarıyla damıtılır. Elbette ilgili baz istasyonunda hem 2019 hem de 2020 yıllarında bazı tekrarlayan yükselişler ve düşüşler gözlemlenmektedir. Ancak bu örnekte mevsimsel hareketlerin eğilim bileşenine göre gücü oransal olarak zayıftır. Bu nedenle baz istasyonu mevsimsel olmayan şeklinde etiketlenmiştir. Çünkü algoritma, baz istasyonlarının herhangi bir anında eğilim bileşeninin mevsimsel bileşene oranı %30 dan fazla ise ilgili baz istasyonunu mevsimsel olarak etiketlemektedir. Bu baz istasyonu örneği için ilgili eşik değeri (%30) aşılamadığından “mevsimsel olmayan” baz istasyonu olarak işaretlenmiştir.



Şekil 4.13. Seviye Kayması Nedeniyle Değişim Noktası Olan İstasyonun Ayrıştırılmış Bileşenleri

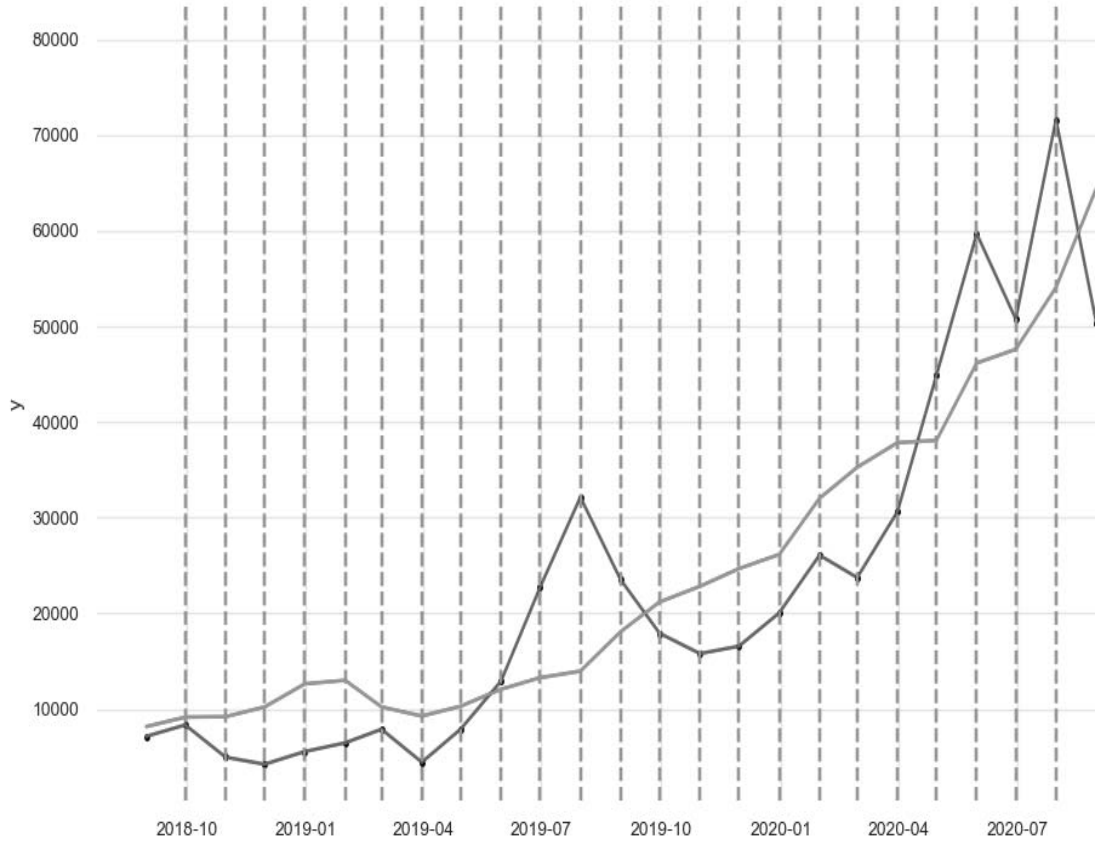
Geçmişteki Eğilim Artışı Bazlı Değişim Durumları: Bazı baz istasyonları için nüfus artışı, müşteri davranış değişiklikleri veya diğer bazı nedenlerden dolayı talep artışı bulunmaktadır.

Algoritmanın bu eğilim değişikliklerini yakalaması ve bu eğilim değişiklikleri ile ilgili mevsimsel etki faktörlerini ayarlaması gerekmektedir. Değişim noktaları sadece seviye değişimlerinden kaynaklanmamaktadır. Literatürde artan veya azalan eğilim bileşeninin durağan olmayan davranışı da değişim noktaları olarak ele alınmaktadır. Yeni yıllarla birlikte eğilimin Şekil 4.14'te görüldüğü gibi üstel olarak büyümeye başladığı baz istasyonları da mevcuttur.

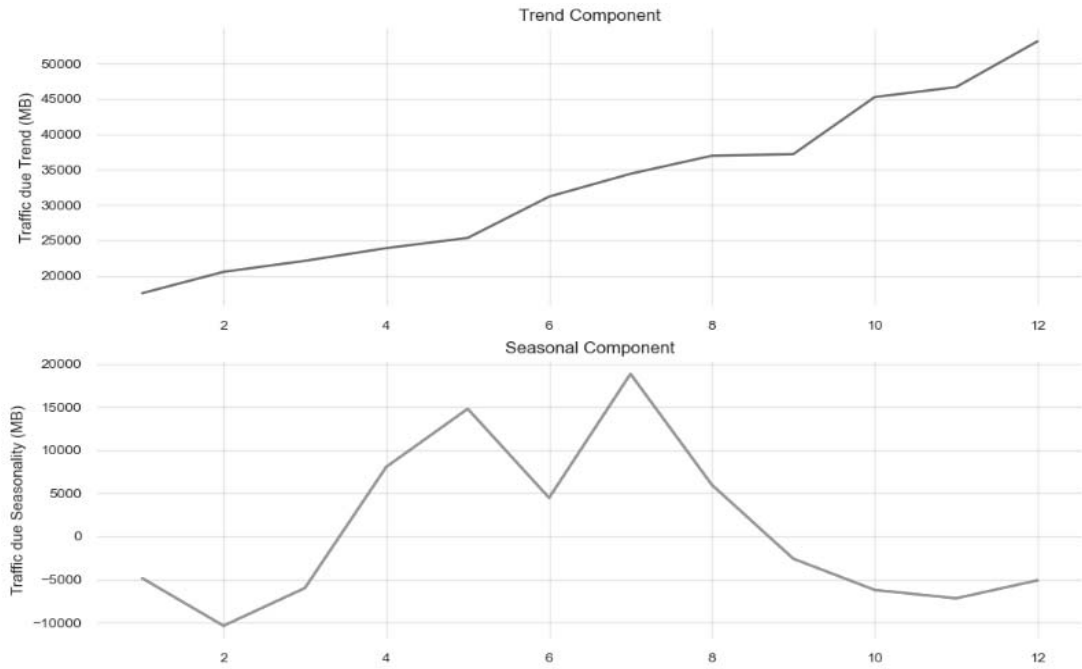


Şekil 4.14. Üstel Artan Eğilim Bileşeni Nedeniyle Değişim Noktası İçeren Baz İstasyonu Örneği

Algoritma bu gibi durumlarda bir değişim noktası tespit eder ve önceden esnek bir değişim noktası olan parça bazında bir eğri uydurmaya çalışır. Bu eğri, Şekil 4.15'te gösterilmiştir. Elde edilen bu üstel eğilim eğrisi orijinal trafik serisinden çıkarılır. Kalan seri ise, Şekil 4.16'daki algoritmanın son bileşenlerini hesaplamak üzere STL ile ayrıştırılır. Mayıs, Haziran ve Temmuz dönemleri için mevsimsel etki daha yüksektir ve Temmuz ayı, aylık trafiğin en yüksek olduğu dönemdir. Dolayısıyla, kapasite planlama önerileri için Mayıs ayından önceki herhangi bir tarih yatırım için daha uygun görünmektedir.

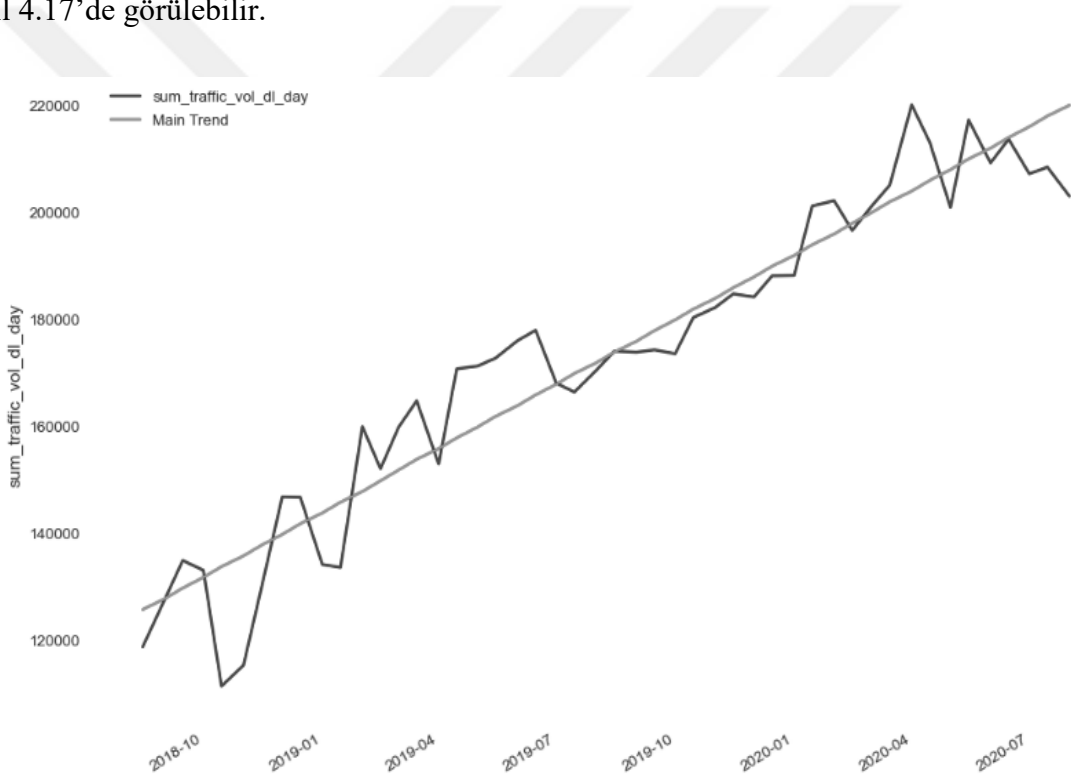


Şekil 4.15. Parça Bazında Uydurma İle Daha Esnek Modelin Değişim Noktası Konumları ve Üstel Uydurulmuş Eğilim Bileşeni



Şekil 4.16. Üstel Eğilim Nedeniyle Değişim Noktasına Sahip Baz İstasyonu İçin Ayrıştırılmış Seri Bileşenleri

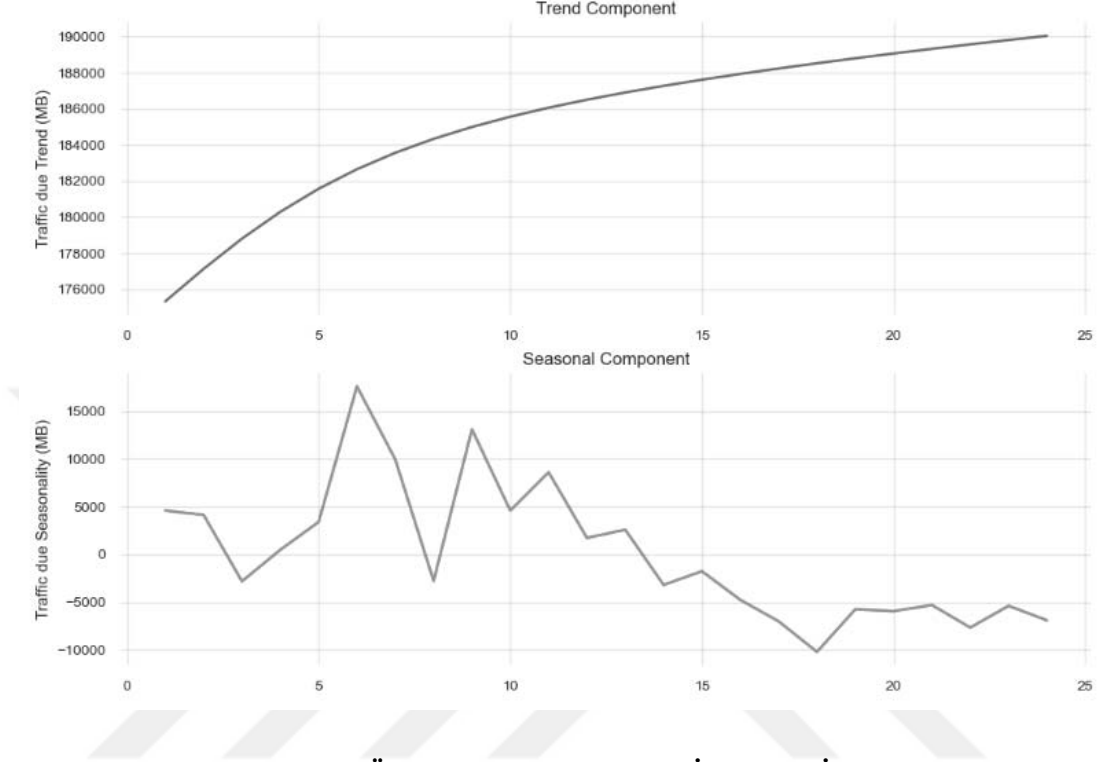
Zayıf Mevsimsellik Veya Mevsimsel Olmama Durumu: Her baz istasyonunun güçlü mevsimsel özellikler göstermesi beklenmemektedir. Algoritmanın baz istasyonunu etiketlemek için mevsimsel etkiye göre eğilim oranının farkında olması gerekir. Bazı baz istasyonlarının rastgele yürüyüşe benzer mevsimsel bir bileşeni olabilir. Bazı diğerlerinin ise mevsimlerde güçlü zirveleri olmayabilir. Bu baz istasyonları mevsimsel olmayan olarak etiketlenmekte ve mevsimsel zirvedeki kapasite yetersizliği nedeniyle bu sahalara yatırım yapılması önlenebilmektedir. Bu bölgeler için, SME'ler sadece eğilim bileşenlerindeki değişiklikleri dikkate almaktadır. Örnek bir mevsimsel olmayan, güçlü eğilim bileşenine sahip baz istasyonu Şekil 4.17'de görülebilir.



Şekil 4.17. Mevsimsel Olmayan Özelliklere Sahip Baz İstasyonu Örneği

Bu kategorideki baz istasyonu trafik özellikleri, eğilim bileşeni ve bazı mevsimsel desenlerle açıklanamayan kalıntı bileşeni ile modellenenmektedir. Nihai algoritmanın görsel çıktısı, içindeki kalıntıları da içeren eğilim bileşenine kıyasla çok zayıf bir mevsimsel bileşene sahiptir. En yüksek nokta, eğilim bileşeni için 150 GB olarak tespit edilmişken, bu döneme karşılık gelen mevsimsel bileşen ise yaklaşık 15 GB seviyesinde kalmıştır. Yani oransal olarak %10'dan daha azdır. Ayrıca, benzerlik ve ilişkili hareketleri kıyaslamak amacıyla Jaccard mesafesi çok düşük

olduğundan, bu baz istasyonu beklendiği gibi algoritma tarafından mevsimsel olmayan olarak etiketlenmiştir. Durum Şekil 4.18’de gösterilmiştir.



Şekil 4.18. Mevsimsel Olmayan Özelliklere Sahip Bir Baz İstasyonu İçin Ayrıştırılmış Bileşenler

Sonuç olarak geliştirilen algoritma, aşağıdaki beş çıktıyı sağlamıştır:

- 1- **Mevsimsel etiketleme:** Algoritma, baz istasyonunu mevsimsel desen ve etki düzeyine göre mevsimsel olan veya mevsimsel olmayan şeklinde etiketlemektedir. Baz istasyonu mevsimselse doğru sonuç elde edildiği düşünülmektedir.
- 2- **Mevsimsel etkinin toplam gücü:** Algoritma mevsimsel trafik miktarının eğilim trafik miktarına oranına ilişkin genel bir SPI bazlı bağlı etki skorunu hesaplamaktadır.
- 3- **Mevsimsel etkinin ay bazında gücü:** Algoritma mevsimsel trafik miktarının eğilim trafik miktarına oranına ilişkin aylık olarak SPI bazlı bağlı etki skorunu hesaplamaktadır.

- 4- **Değişim noktasının varlığı:** Algoritma baz istasyonunun geçmiş çevrimdışı verilerinde değişim noktaları olup olmadığını tespit etmektedir. Değişim noktaları içeriyorsa doğru sonuç elde edildiği düşünülmektedir.
- 5- **Mevsim süresi:** Algoritma, mevsimsel etki altındaki gün cinsinden süreyi hesaplamaktadır. Bu süre boyunca eğilim ve mevsimsellik oranı sürekli olarak %30'un üzerindedir.

Algoritma tüm test baz istasyonlarında çalışmakta ve bu beş ana çıktıyı sağlamaktadır. Bunlardan ikisi, mevsimsel etiketleme ve değişim noktasının varlığı şeklinde ifade edilen ikili çıktılarıdır. Ancak diğer üçü, mevsimsel etkinin gücü ve mevsimin gün biriminde olası süresini belirten sayısal değerlerdir. Genel bir doğruluk puanı elde etmek için, SME'lerin bakış açısıyla bir kabul kriteri tanımlayarak bu sayısal çıktılar kategorik çıktılara dönüştürülmüştür. Baz istasyonu, mevsimsellik ve değişim noktası durumu için doğru şekilde etiketlenmişse, daha sonra mevsimsel etki miktarı ve süresinin yanı sıra (4.10)'da ifade edildiği gibi Mutlak Yüzde Hata (Absolute Percentage Error - APE) açısından da kontrol edilir. SPI, mevsimsel etkinin en yoğun olduğu ay dikkate alınarak hesaplanır. Ayrıca, baz istasyonu başına SME'ler tarafından tanımlanan gerçek bir değer bulunmaktadır. Bu iki ölçüm için kabul edilebilir APE değeri SME'ler için %15'in altıdır. Bu seviye, planlama mühendisleri için ağ kapasitesi yatırım kararlarını zamanında vermeleri açısından tatmin edici hata seviyesidir. Hesaplanan SPI ile gerçekleşen SPI ve hesaplanan süre ile gerçekleşen süre açısından bir baz istasyonu test verisi üzerinde APE değerinin %15 altına gözlemlenmesi durumunda, her iki tahmin de doğru olarak işaretlenmiştir.

$$APE = \frac{|Calculated\ Value - Actual\ Value|}{Actual\ Value} \quad (4.10)$$

Test setindeki her baz istasyonu algoritmaya göre kategorize edilmiş ve Tablo 4.1. doğruluk tablosuna göre etiketlenmiştir. Problem ikili sınıflandırmaya dönüştürüldüğü için, Tablo 4.2'de ifade edildiği gibi F1 skoru ile genel bir doğruluk ölçütü elde edilmiştir. Zamanında verilen yatırım kararlarını kaçırmamak ve müşteri deneyiminin azalmasına sebebiyet vermemek için algoritma tarafından kaç tane gerçek pozitifin tespit edildiği kritiktir. Bu durum gerçek pozitif değerlerin oranına (recall) karşılık gelmektedir. Üstelik CAPEX'in artmasına neden olan gereksiz yatırım kararlarına neden olmamak da algoritmanın kabul görmesi açısından

önemlidir. Bu durum da hassasiyet (precision) değerine karşılık gelmektedir. Bu iki ölçüm arasında bir denge kurmak için, hassasiyet ve gerçek pozitif değerlerin oranı arasında harmonik ortalama hesabına karşılık gelen F1 skoru kullanılabilir. Her iki puana da eşit ağırlık verilir, bu nedenle betayı F ölçüsü için varsayılan değer olan 1,0 olarak tutmak gereklidir.

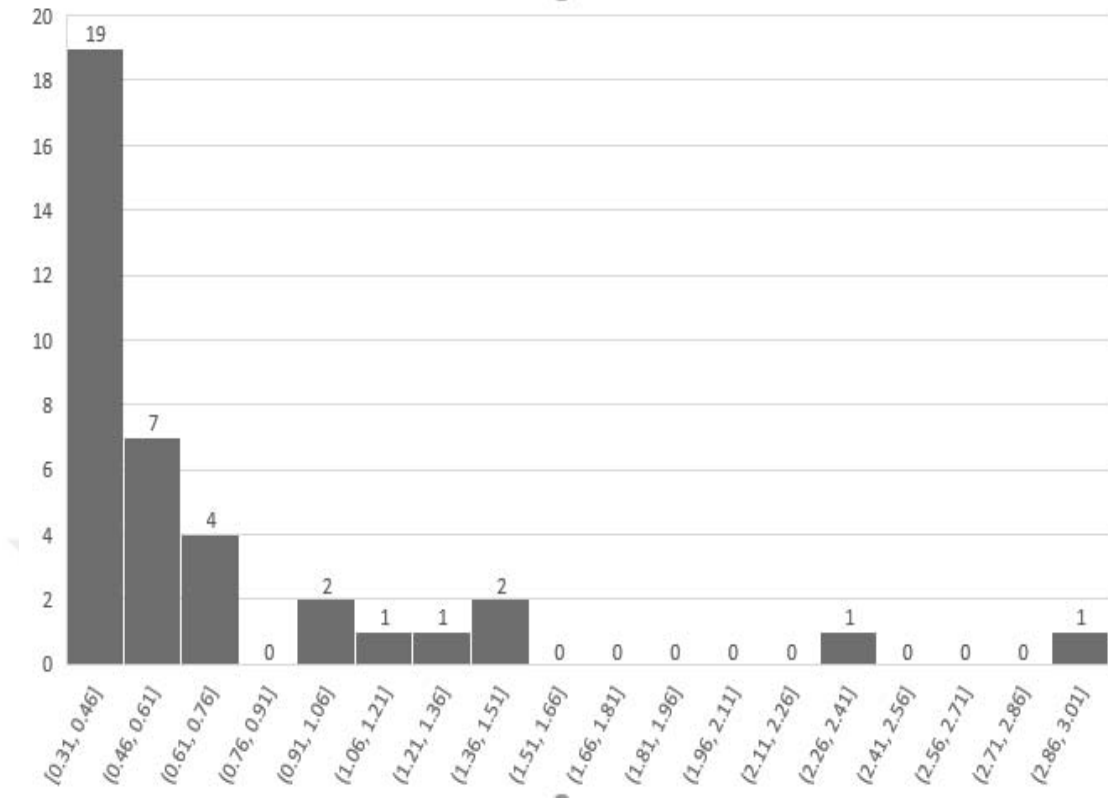
Tablo 4.1. Karmaşıklık Matrisi Değerleri

Değer	Açıklama
TP (True-Positive / Doğru-Pozitif)	Dört çıktının hepsinden, sonucun doğru olduğu ve baz istasyonunun doğru ayrışmayla mevsimsel etiketlendiği anlamına gelmektedir.
TN (True-Negative / Doğru-Negatif)	Dört çıktının hepsinden, sonucun doğru olduğu ve baz istasyonunun mevsimsel olmayan olarak etiketlendiği anlamına gelmektedir.
FN (False-Negative / Yanlış-Negatif)	Gerçek bir mevsimsel baz istasyonunu mevsimsel olmayan olarak etiketlemek anlamına gelmektedir.
FP (False-Positive / Yanlış-Pozitif)	Mevsimsel olmayan gerçek bir baz istasyonunu mevsimsel olarak etiketlemek anlamına gelir.

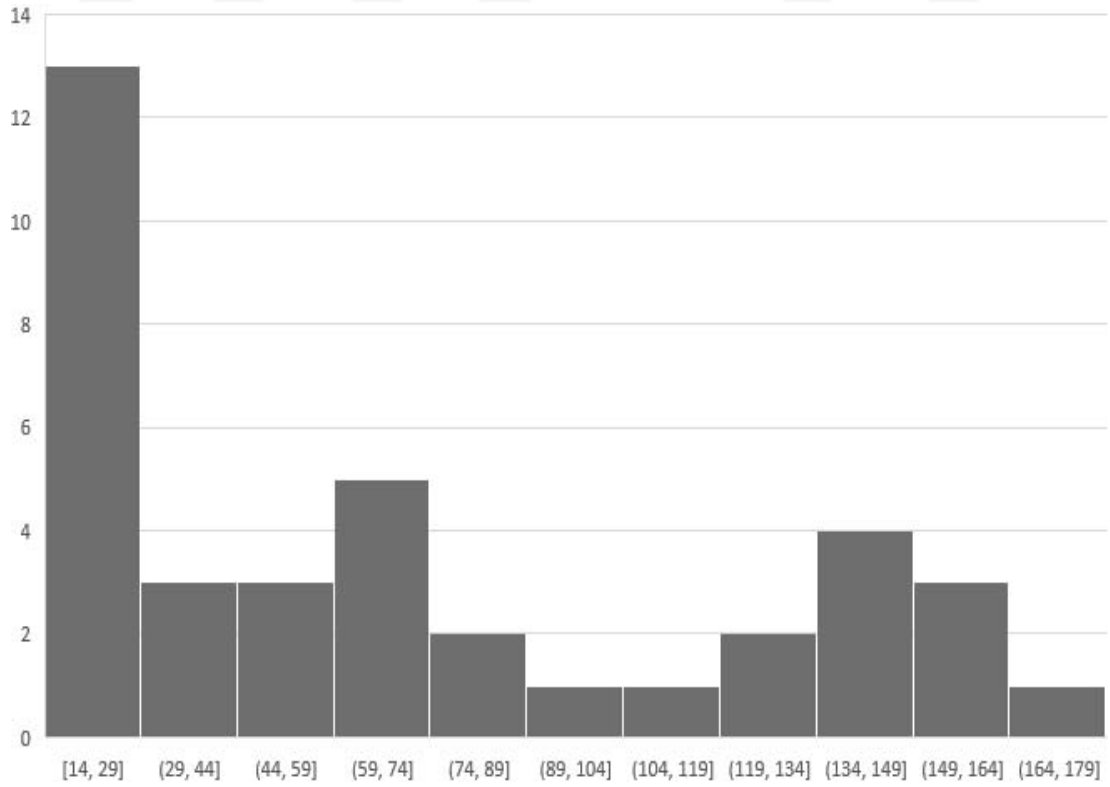
Tablo 4.2. Metrik Formülleri

Değer	Formül
Doğruluk (Accuracy)	$(TP + TN) / (TP + FP + FN + TN)$
Gerçek Pozitif Değerlerin Oranı (Recall)	$TP / (TP + FN)$
Hassasiyet (Precision)	$TP / (TP + FP)$
F1 Skoru	$2 \times \text{Hassasiyet} \times \text{Gerçek Pozitif Değerlerin Oranı} / (\text{Hassasiyet} + \text{Gerçek Pozitif Değerlerin Oranı})$

Test setinde algoritma tarafından tespit edilen SPI ve dönemsel uzunluk değerlerinin dağılımı sırasıyla Şekil 4.19 ve 4.20’de görselleştirilmiştir.



Şekil 4.19. Test Seti SPI Değerleri Çıktısı Dağılımı



Şekil 4.20. Test Seti Mevsimsel Dönem Uzunluğunun Dağılımı

Etiketleme sonuçlarının toplam sayısı Tablo 4.3'te bir karmaşıklık matrisi olarak sunulmuştur. Algoritmanın etiketlediği 45 baz istasyonu gerçek mevsimsel baz istasyonları olarak tatmin edici bir süre uzunluğu hesaplaması ve mevsimsel etki değerleri ayrıştırmasını doğru şekilde sağlamaktadır (Doğru-Pozitif). Yanlış etiketleme veya yetersiz ayrıştırma seviyesi 8 adettir (Yanlış-Negatif). Bu 8 yanlış-negatiften 5 tanesi eğilim bileşeninin yanlış ayrıştırılmasından, 3 tanesi ise daha kısa dönem uzunluğu hesaplamalarından kaynaklanmaktadır. Burada daha kısıdan kastedilen, periyot uzunluğunun %15'ten fazla hata ile hesaplanması anlamına gelmektedir. Algoritma 44 baz istasyonunu mevsimsellik dışı olarak doğru bir şekilde etiketlemiştir (Doğru-Negatif). Bazı değişim noktası etkilerinden dolayı, SPI dağılım benzerliği doğru adreslenememiş dolayısıyla, gerçekte mevsimsel olmayan baz istasyonlarından 3'ü, mevsimsel olarak etiketlenmiştir (Yanlış-Pozitif).

Tablo 2.3. Karmaşıklık Matrisi

	Doğru Mevsimsel	Doğru Mevsimsel Olmayan
Tahmin Edilmiş Mevsimsel	45	3
Tahmin Edilmiş Mevsimsel Olmayan	8	44

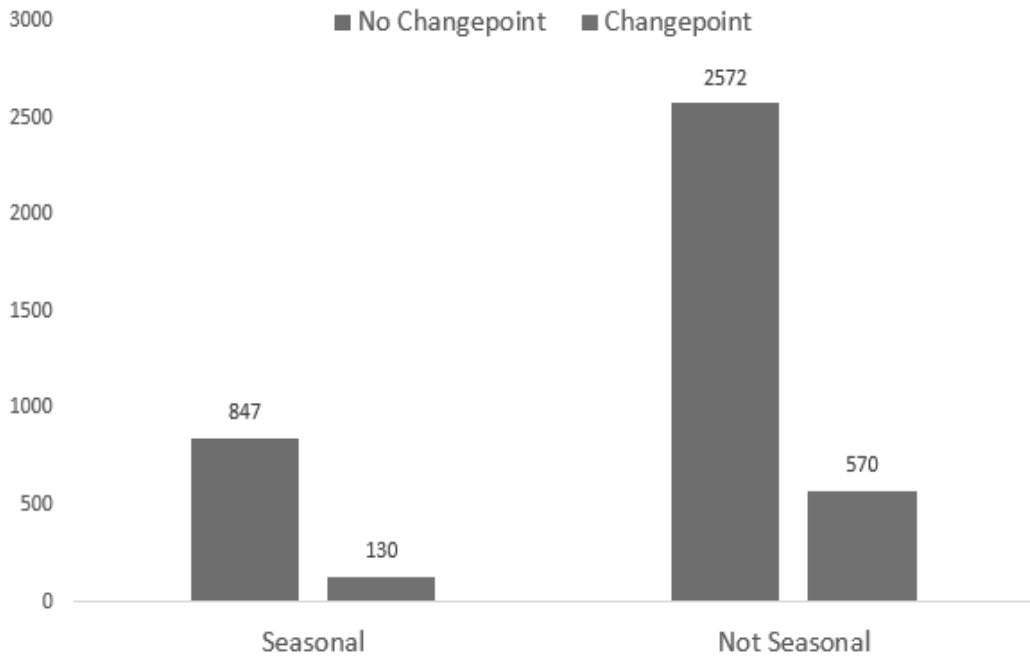
Hassasiyet, gerçek pozitif değerlerin oranı ve bunların harmonik ortalaması olan F1 skoru genel doğruluk sonuçları Tablo 4.4'te listelenmiştir. Algoritma sonuçları, STL ve Prophet'in kendi başına sağladıkları ayrıştırma yaklaşımlarıyla karşılaştırılmıştır ve bu sayede çok daha iyileştirilmiş doğruluğu gösterilmiştir. Algoritma, değişim noktası olasılığı ve dağıtımı üzerinde bazı ek otomatik analizler yaparak, bu çerçevelerin her ikisini de kullanmakta, trafik geçmişindeki kritik değişiklikleri anlamakta ve gelişmiş ayrıştırma performansı için doğru parametreleri ve bölümleri seçmektedir. STL ve Prophet yaklaşımlarının ortalama F1 skorunu %14 oranında geride bırakan algoritmanın kendi F1 skoru 0,89 olarak gözlemlenmiştir.

Tablo 4.4. Doğruluk Sonuçları ve Karşılaştırma

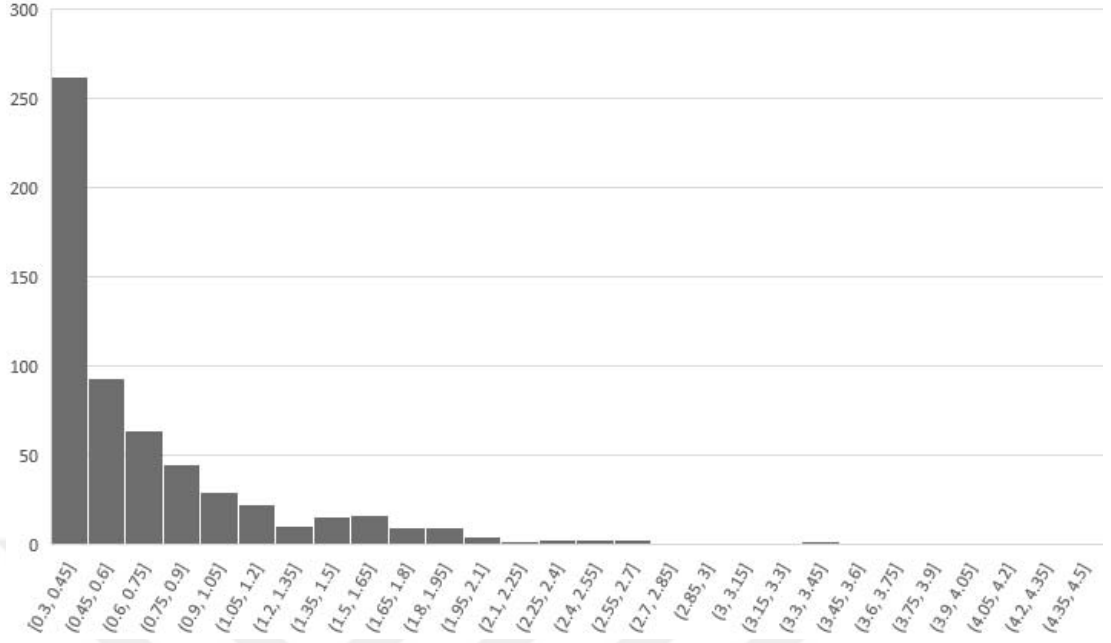
	Sunulan Algoritma	STL	Prophet
Hassasiyet	0,93	0,72	0,77
Gerçek Pozitif Değerlerin Oranı	0,84	0,73	0,79
F1-Skoru	<u>0,89</u>	0,72	0,78

Değişim noktası içeren geçmiş veriler için STL bu değişiklikleri, değişim noktasının eğilim

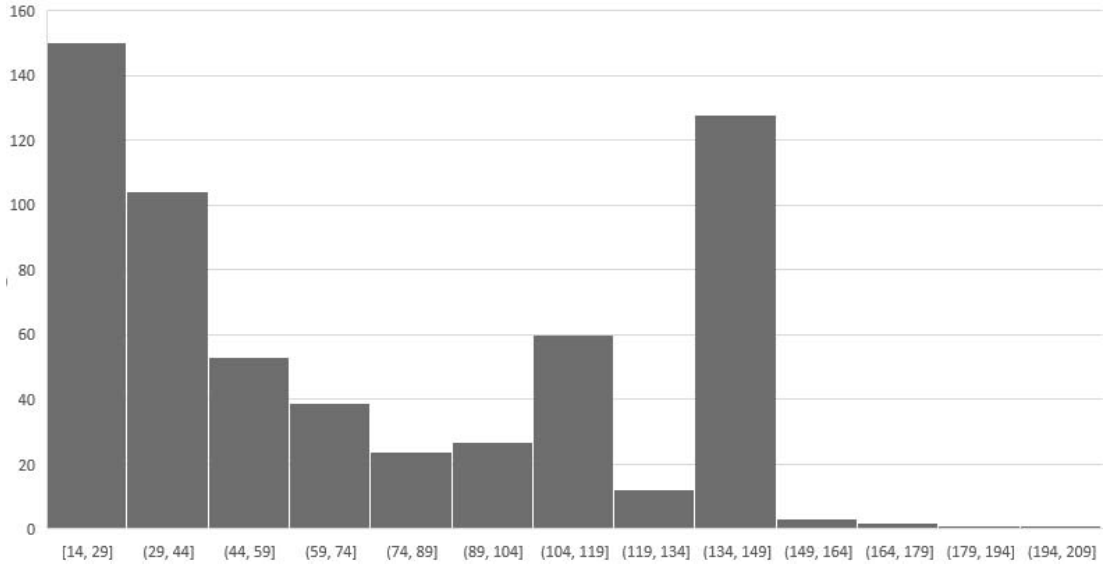
bileşeni yerine mevsimsel bileşenle açıklama eğilimindedir. STL'den farklı olarak, Prophet otomatik değişim noktası algılama özelliği olarak varsayılan parametrelerle çalıştırıldığında iyi bir performans sergilemektedir. Ancak ciddi eğilim değişikliklerini kavrayamamakta ve eğilim bileşenini yeterince uyduramamaktadır. Bununla birlikte, Prophet çarpımsal mevsimselliği açıklamada STL'e göre daha başarılıdır. Her iki çözüm de nihai karar için değişim noktasının sonrasını dikkate almamaktadır ki bu da SME'lerin karşılanamayan bir başka beklentisidir. Çünkü değişim noktasından sonraki hücre karakteristiğini doğru yorumlamak daha değerli ve güncel olmaktadır. Değişim noktası zamanından önceki hücre karakteristiğinin etkisi altında kalan algoritmalar uzun süreli hafıza olarak bu karakteristiği muhafaza etmektedirler. Bundan dolayı geçmişinde değişim noktası içeren baz istasyonlarının bileşenlerine ayrıştırılmasında geliştirdiğimiz algoritmanın kullanılması daha doğru sonuç elde edilmesini sağlayabilmektedir. Tablo 4.4'te görüldüğü gibi kıyaslamada iyileştirilmiş hassasiyet ve gerçek pozitif değerlerin oranı puanları elde edilir. Veri setindeki tüm baz istasyonlarını kategorize etmek için nihai algoritma tüm veri setine uygulandığında oluşan sınıflandırma dağılımı Şekil 4.21'de görülmektedir. Tüm baz istasyonlarının toplam %31'i mevsimseldir ve baz istasyonlarının %20'sinin geçmişinde geliştirilen algoritmaya göre değişim noktaları bulunmaktadır. SPI değerlerinin dağılımı ve mevsimsel dönem uzunluğu sırasıyla Şekil 4.22 ve Şekil 4.23'te gösterilmiştir.



Şekil 4.21. Tüm Veri Setinin Etiketli Baz İstasyonlarının Dağılımı



Şekil 4.22. Tüm Veri Seti SPI Değerleri Çıktısının Dağılımı



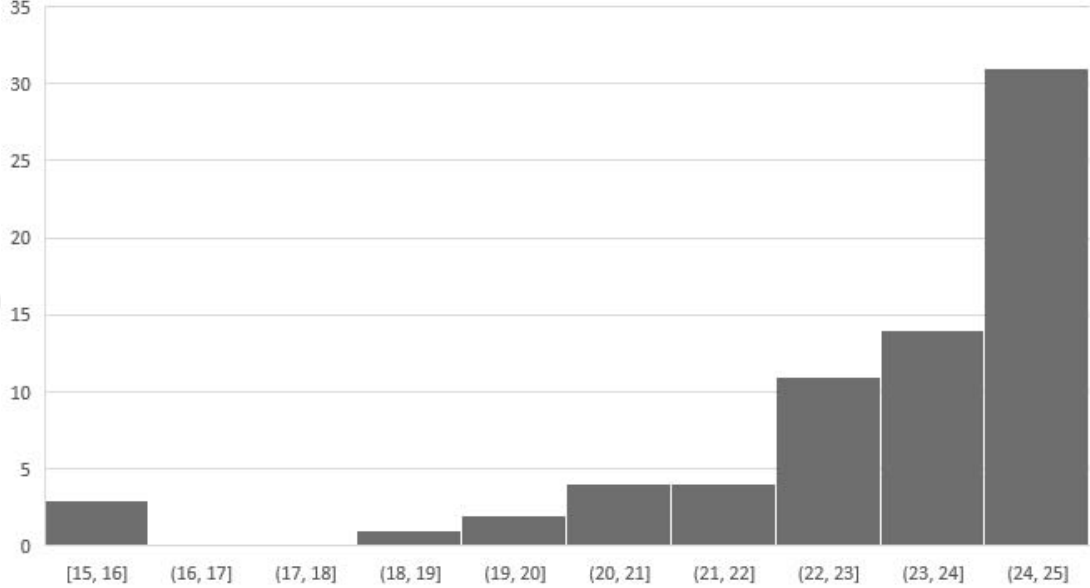
Şekil 4.23. Tüm Veri Seti Mevsimsel Dönem Uzunluğunun Dağılımı

4.2. TAHMİNLEME PROBLEMİ BULGULARI

4.2.1. Kümeleme Algoritması ve Kullanılan Nitelikler

Aglomeratif Hiyerarşik Kümeleme algoritması, geçmişte sentetik olarak oluşturulacak her hücre için ortak trafik karakteristiğini miras almak için uygun komşu hücreleri seçmek için uygulanır. Her hücre için, kümeleme amacıyla dikkate alınması gereken ortalama olarak yaklaşık 25

komşusu vardır. Bunlar aynı baz istasyonu altında olmayan (eş lokasyonlu olmayan) komşu hücrelerdir. Her bir test hücresi için toplam eş lokasyonlu olmayan komşu sayısının histogramı Şekil 4.24'te görselleştirilmiştir.



Şekil 4.24. Eş Lokasyonlu Olmayan Toplam Komşu Sayısı Dağılımı

Test setindeki her hücre için 25 komşusunu kullanırız ve küme özellikleri olarak üç nitelik ile bu hücreler üzerinde hiyerarşik kümeleme uygularız. Bu nitelikler sırasıyla;

- İki hücre arasında günlük ortalama geçiş denemeleri sayısı(hareketlilik aktivitesi),
- metre cinsinden iki hücre arasındaki coğrafi mesafe,
- iki hücrenin nispi azimutu (hizmet alanı örtüşme göstergesi)

olarak listelenebilir. Burada belirtilen "iki hücre", kaynak hücre (yetersiz geçmişe sahip hücre) ve onun eş lokasyonlu olmayan yani aynı sahadan hizmet vermeyen komşularından biri anlamına gelir.

Çalışmamızda iki hücrenin bağıl azimutu ve aralarındaki coğrafi uzaklık aşağıdaki fonksiyonlarla hesaplanmıştır. Böylece, kaynak hücreye (cellid1) ilişkin komşu hücreler (cellid2) için yukarıda belirtilen üç nitelik değeri get_features işlevi çağrılarak elde edilir.

Fonksiyon gerekli tüm hesaplamaları dahili olarak yapar ve ölçeklendirildikten sonra kümeleme amaçlı kullanılacak özellik değerlerini döndürür. Bunun için de geliştirdiğimiz diğer fonksiyonları çağırır.

Fonksiyon 1 *get_bearing*

```

procedure get_bearing(pointA, pointB)
  set lat1  $\leftarrow$  call math.radians(pointA[0])
  set lat2  $\leftarrow$  call math.radians(pointB[0])
  set diffLong  $\leftarrow$  call math.radians(pointB[1] - pointA[1])
  set x  $\leftarrow$  math.sin(diffLong) * math.cos(lat2)
  set y  $\leftarrow$  math.cos(lat1) * math.sin(lat2) - (math.sin(lat1) *
  math.cos(lat2) * math.cos(diffLong))

  set initial_bearing  $\leftarrow$  math.atan2(x, y)
  set initial_bearing  $\leftarrow$  math.degrees(initial_bearing)
  set compass_bearing  $\leftarrow$  (initial_bearing + 360) % 360

  return compass_bearing
end procedure

```

Fonksiyon 3 *get_distance*

```

procedure get_distance(lat1, lon1, lat2, lon2)
  set R  $\leftarrow$  6373.0
  set lat1  $\leftarrow$  math.radians(lat1)
  set lon1  $\leftarrow$  math.radians(lon1)
  set lat2  $\leftarrow$  math.radians(lat2)
  set lon2  $\leftarrow$  math.radians(lon2)

  set a  $\leftarrow$  math.sin((lat2 - lat1) / 2) ** 2 + math.cos(lat1) *
  math.cos(lat2) * math.sin((lon2 - lon1) / 2) ** 2
  set c  $\leftarrow$  2 * math.atan2(math.sqrt(a), math.sqrt(1 - a))

  return R * c
end procedure

```

Fonksiyon 2 *get_relative_azimuth*

```

procedure get_relative_azimuth(lat1, lon1, lat2, lon2, azi1,
azi2)
  set Point_A  $\leftarrow$  (lat1, lon1), Point_B  $\leftarrow$  (lat2, lon2)
  set Alfa  $\leftarrow$  call get_bearing(pointA, pointB)
  set Beta  $\leftarrow$  call get_bearing(pointB, pointA)

  if abs(azi1 - Alfa) > 180 and abs(Beta - azi2) < 180 then
    return (360 - abs(azi1 - Alfa)) + abs(Beta - azi2)
  else if abs(azi1 - Alfa) < 180 and abs(Beta - azi2) > 180 then
    return abs(azi1 - Alfa) + (360 - abs(Beta - azi2))
  else if abs(azi1 - Alfa) > 180 and abs(Beta - azi2) > 180 then
    return (360 - abs(azi1 - Alfa)) + (360 - abs(Beta - azi2))
  else
    return abs(azi1 - Alfa) + abs(Beta - azi2)
  end if
end procedure

```

Fonksiyon 4 *get_features*

```

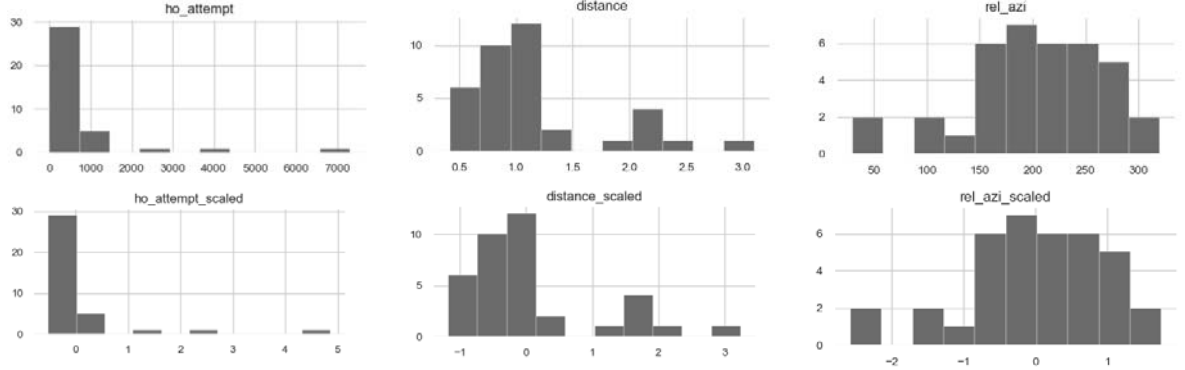
procedure get_features(cellid1, cellid2, lat1, lon1, lat2, lon2,
azi1, azi2)
  set ho_attempt  $\leftarrow$  call get_ho_attempt(cellid1, cellid2)
  set distance  $\leftarrow$  call get_distance(lat1, lon1, lat2, lon2)
  set rel_azi  $\leftarrow$  360 - call get_relative_azimuth(lat1, lon1,
lat2, lon2, azi1, azi2)

  return (ho_attempt, distance, rel_azi)
end procedure

```

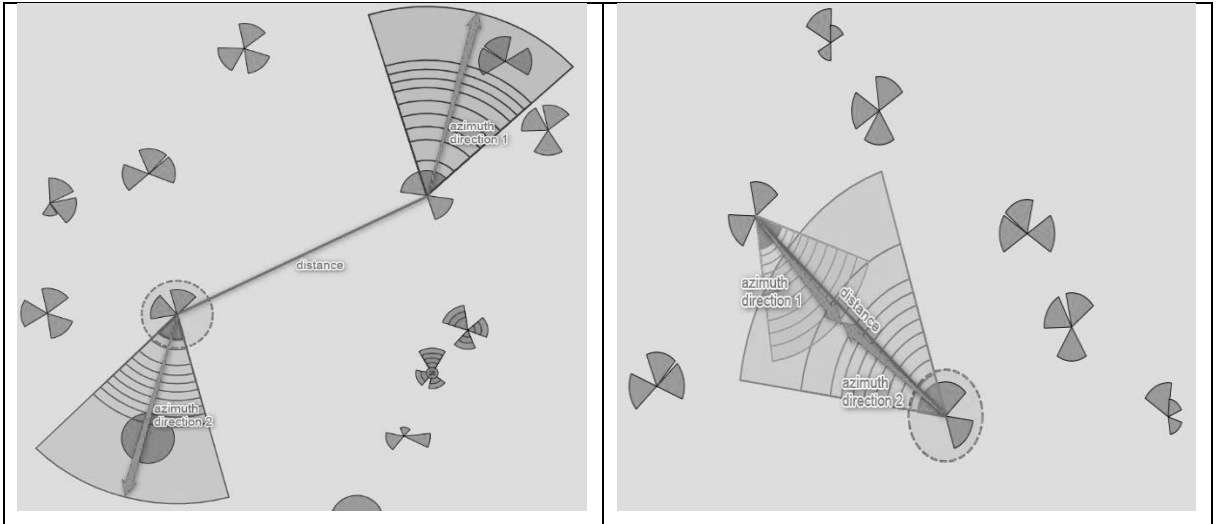
Mesafe, günlük geçiş girişimleri veya iki hücre arasındaki göreceli azimut, birim olarak ve hücre başına yüksek varyansa sahip olduğundan, mesafe tabanlı bir kümeleme algoritması uygulamadan önce, bu üç özelliğin tümüne standardizasyon uygularız. Bu nedenle, kümeleme yaklaşımını yönlendirmek için eşit şansa sahip sıfır ortalama, bir birim varyans özelliklerine

sahip olurlar. Örneğin, bir kaynak hücre ile komşuları arasındaki nitelik değerleri dağılımı Şekil 4.25'te görselleştirilmiştir.



Şekil 4.25. Kümeleme Öncesi Niteliklerin Değer Dağılımlarına Dair Bir Örnek

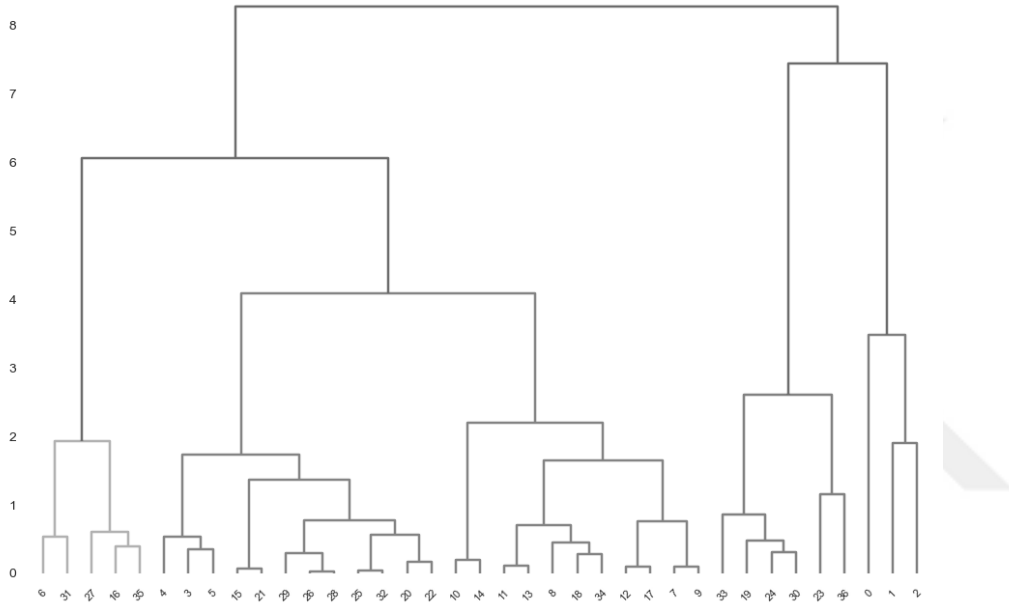
Göreceli azimut özelliği, iki hücre arasındaki açıya bağlıdır. Bu iki hücre birbirine belirli bir açıyla bakıyorsa, daha düşük bir bağıl azimut değeri elde edilir. Çünkü örtüşen bölgelere hizmet etme potansiyelleri daha fazladır. Değer olarak 0 ile 360 arasındadır. Ancak kümeleme algoritmamıza beslemek için bunu bir özellik haline getirdiğimizde 360 değerinden çıkartarak kullandık. Dolayısıyla birbirine bakan hücreler için bu nitelik daha yüksek değer alır ve bunun tersi de geçerlidir. Şekil 4.26, solda daha yüksek bir göreceli azimut (niteliğe dönüştürmeden önceki haliyle) değerini ve sağ pencerede daha düşük bir değeri görselleştirir.



Şekil 4.26. Göreceli Azimut Değeri için Örnekler

Hiyerarşik kümeleme algoritmasının çıktısı, verilerin kümeler ağaç görünümüyle ayrılması olan dendrogramdır. Ağacı kesmeye karar verdiğimiz seviyeye göre farklı sayıda küme elde

edilir. Bir hücre için komşuların kümelenmesi amacıyla örnek bir dendrogram görünümü Şekil 4.27’de görselleştirilmiştir. Çalışmamızda komşu hücreleri beş kümeye ayırdık. Her kümenin, kaynak hücreyle günlük ortalama geçiş denemeleri açısından ortalama bir mobilite etkinliği vardır. Mobilite niteliğinin farklı etkilerini karşılaştırmak için, küme merkezlerinin ortalama mobilite seviyelerini göz önünde bulundurarak bu beş küme arasından ortalama mobiliteye sahip ilk kümeyi seçiyoruz ve kaynak hücre için sentetik geçmişi oluşturmak için bu küme kapsamındaki komşu hücreleri kullanıyoruz.



Şekil 4.27. Dördüncü Seviyeden Kesilerek Oluşturulmuş 5 Kümeli Bir Dendrogram Örneği

4.2.2. Uygulanan LSTM Model Mimarileri

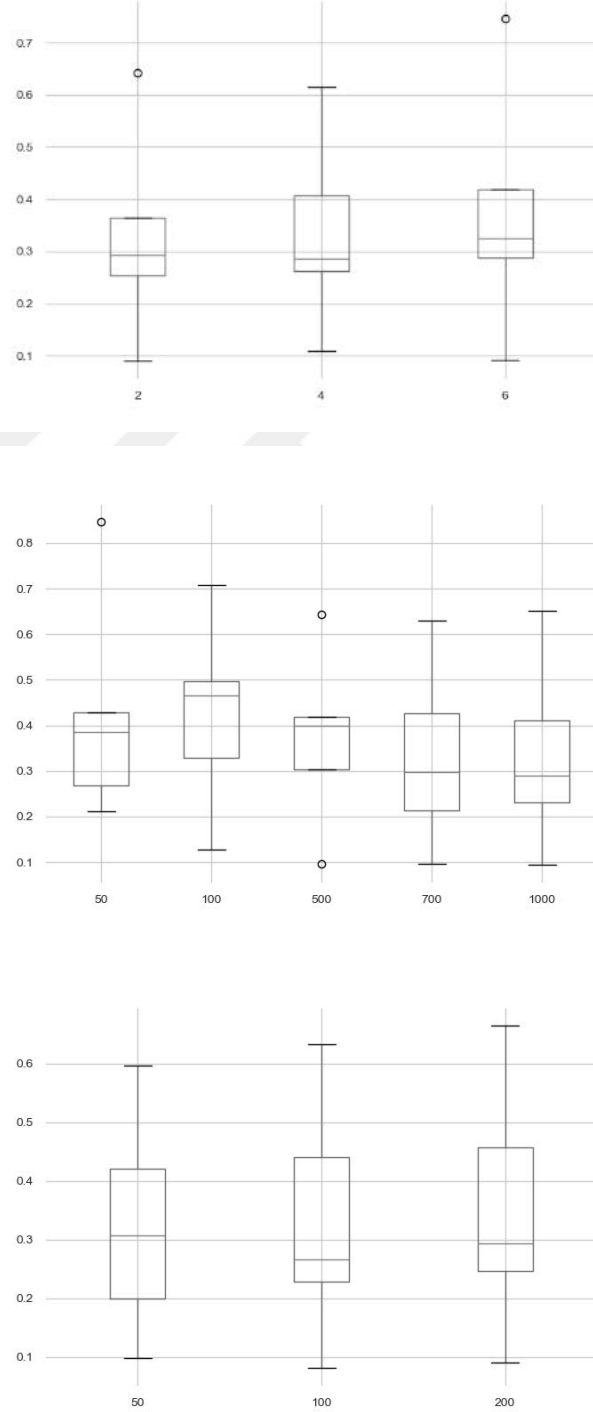
Çalışmamızda sonuçları karşılaştırmak için dört farklı tek değişkenli LSTM sinir ağı kullanıyoruz. Bu modeller sırasıyla Vanilla LSTM, LSTM kodlayıcı-kod çözücü, CNN-LSTM kodlayıcı-kod çözücü ve Evrişimli LSTM kodlayıcı-kod çözücü ağlarıdır. Farklı mimariler de farklı hiperparametre değerleri ile test ederek sonuçları paylaştık. Modeller, [örnekler, zaman adımları, nitelikler] biçiminde girdi verileri bekler. Bizim durumumuz için, nitelikler boyutu tek değişkenli bir yaklaşımdır. Vanilla LSTM modeli sadece katmanlarında LSTM birimlerinden oluşmaktadır. Ancak LSTM kodlayıcı-kod çözücü modelinin iki alt modeli vardır. İlk alt model girdi vektörünü kodlar ve aşağıdaki alt model, tahmin edilen değerleri çıktı olarak dahil ederek kodlanmış çıktının kodunu çözer. CNN-LSTM kodlayıcı kod çözücü mimarisi, kodlayıcı alt modelinin ön katmanında ek bir evrişim katmanı içerir. CNN'ler frekans

değişimlerini azaltmada iyidir Sainath vd. (2015). Özellikle uzamsal-zamansal analiz için özneteliklerin çıkarılması yararlıdır. 1D CNN, çalışmamızda seri girişini okur ve LSTM için özellikleri çıkartırız.

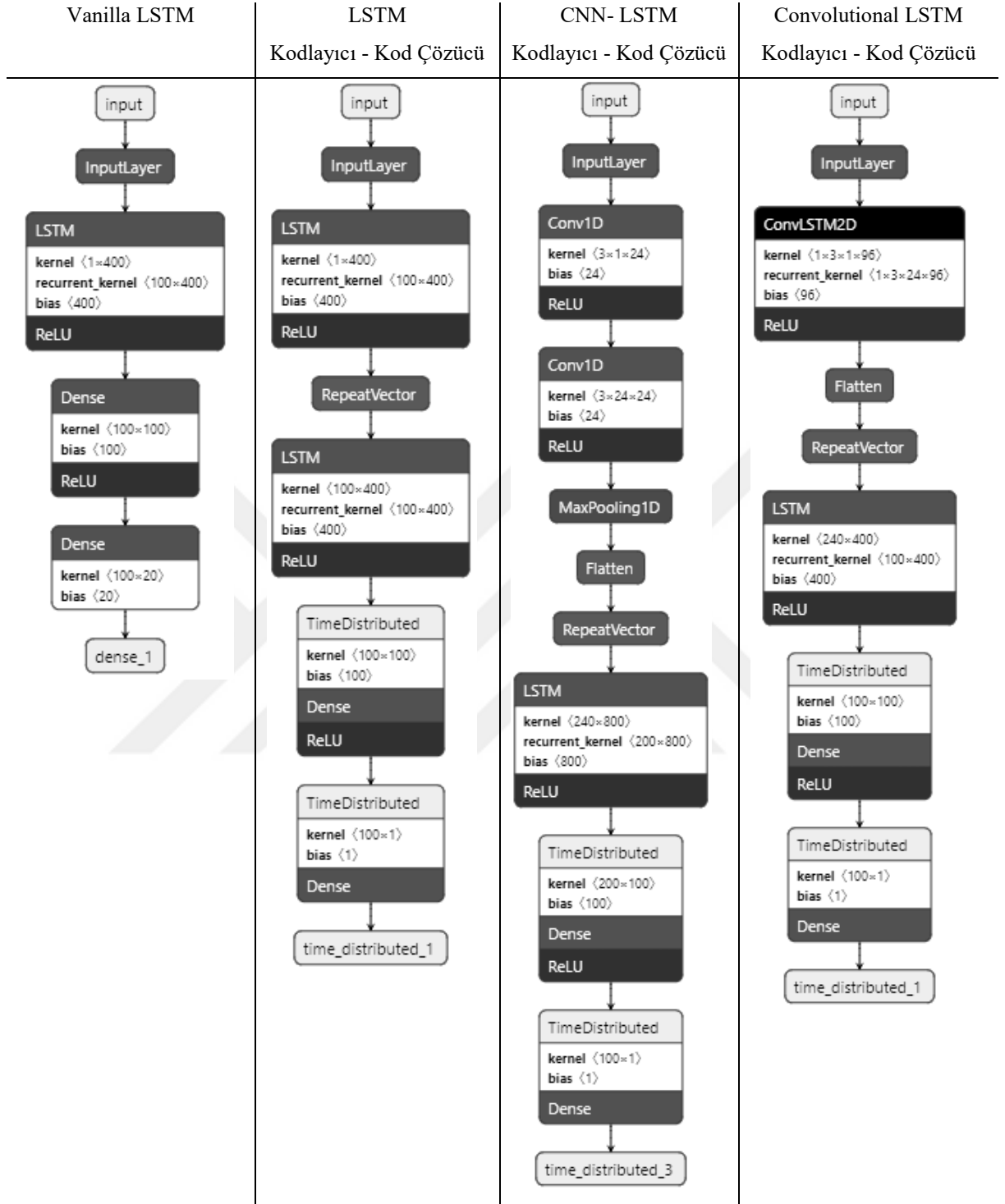
En iyi yığın büyüğü olarak 2 görülmektedir.

En iyi yineleme sayısı olarak 700 görülmektedir.

Katmanlardaki LSTM birimi sayısı olarak 100 ideal sayı olarak görülmektedir.



Şekil 4.28. Hiperparametre İyileştirme Sonuçları



Şekil 4.29. Kullanılan LSTM Mimarileri

ConvLSTM aynı zamanda 2B evrişimleri uygulamak için özel genişletilmiş bir CNN türüdür. Benzer şekilde, birden çok ConvLSTM katmanını istifleyerek ve bir kodlama-kod çözme yapısı oluşturarak, uzay-zamansal dizi tahmin problemleri için bir ağ modeli oluşturan çalışmalar da

vardır Xingjian vd. (2015). LSTM, giriş verilerini doğrudan okur. CNN-LSTM modelleri, CNN modellerinden gelen çıktıyı yorumlar ve ConvLSTM modelleri, LSTM birimlerini beslemek için girdi verilerine katlamalar uygular. LSTM modellerinin hiperparametre ayarına yönelik çalışmamızda, yığın boyutu, yineleme sayısı ve LSTM birim sayısı parametreleri için ızgara araması yapıyoruz. Metodoloji olarak beş hücre seçiyoruz ve her parametre kombinasyonu için modelleri beş kez ayrı ayrı çalıştırıyoruz ve ortalama MAPE sonuçlarını alıyoruz. Ardından, ortalama MAPE sonuçlarını kontrol ederek en iyi parametre kombinasyonunu seçiyoruz. Hiperparametre kombinasyonlarının ortalama MAPE dağılımı Şekil 4.28'de görselleştirilmiştir.

Dört farklı LSTM mimarisi için katmanların detaylı açıklaması Şekil 4.29'da Netron aracı kullanılarak görselleştirilmiştir. RepeatVector katmanı, kodlayıcı-kod çözücü mimarileri için girişi n kez tekrarlar. Bu nedenle, çıktı katmanları, tek çıktılı TimeDistributed katmanıdır. Kodlayıcı-kod çözücü mimarilerinden farklı olarak, Vanilla LSTM, 15 günlük toplu trafik hacmi çıkışı ile 10 ay öncesinden trafik tahmini için 100 nöron içeren yoğun bir katman içerir.

4.2.3. Çözümün İş Akışı

POC ortamımızı Python dili ve scipy, statsmodels, sklearn, keras ve multiprocessing paketlerini Visual Studio Code IDE'si üzerinde kullanarak hazırladık. Hücre istatistiklerini ve saha planlama niteliklerini toplamak ve depolamak için PostgreSQL veri tabanını ve sinir ağı mimarisi görselleştirmesi için Netron aracını kullandık. Çalışmamızda sonuçları karşılaştırmak için dört farklı tek değişkenli LSTM sinir ağı kullanıyoruz.

Uygulamamız iki aşamadan oluşmaktadır:

- Hiperparametre ayarı için bir kıyaslama çözümü,
- Doğru hiperparametrelerle tahmin iyileştirmeleri için bir kıyaslama çözümü.

Hiperparametre ayarlama aşaması için iş akışı adımları aşağıda listelenmiştir. Daha iyi uyan hiperparametreleri anlamak için beş örnek hücre seçtik. Yığın boyutu, yineleme sayısı ve katmanlardaki LSTM birim sayısı dahil olmak üzere her bir hiperparametre kombinasyonu, numune hücreler başına beş kez çalıştırılır ve kıyaslama için ortalama MAPE hata değerleri hesaplanır. Her kombinasyonun görselleştirilmiş etkisi önceki bölümde paylaşılmıştır. SARIMAX için Box-Cox dönüşümü ve STL-ETS için ızgara araması için uygulanmıştır.

Hiperparametre ayarlaması için iş akışı

- Select 5 sample cells with 2 years history
 - Retrieve all sample cells pm and cm
 - Aggregate data into 15 days intervals
 - for each batch size candidate (2, 4, 6)
 - * for each LSTM unit size candidate (50, 100, 500, 700, 1000)
 - + for each epoch candidate (50, 100, 200)
 - o for each sample cell
 - # for 5 times
 - > Forecast next 8 months with Vanilla LSTM with current hyperparameter combination
 - > Forecast next 8 months with LSTM Encoder-Decoder with current hyperparameter combination
 - > Forecast next 8 months with CNN-LSTM Encoder-Decoder with current hyperparameter combination
 - > Forecast next 8 months with ConvLSTM Encoder-Decoder with current hyperparameter combination
 - > Forecast next 8 months with ConvLSTM Encoder-Decoder with current hyperparameter combination
 - > Calculate MAPE for selected sample cell for each model
 - # end for
 - # Calculate average MAPE for selected cell
 - o Calculate average MAPE of all sample cells
 - o Save results to DB with hyperparameter combination
 - o end for
 - + end for
 - * end for
 - end for
 - Retrieve all results from DB for each architecture and hyperparameter
 - List the best combination as hyperparameter with lowest MAPE level.
-

Kıyaslama için genel iş akışının adımları aşağıda listelenmiştir. Algoritmamız, başlangıçta sınırlı verilerle ve daha sonra sentetik geçmiş verilerle her tahmin algoritmasını kullanarak 70 test hücresi için ayrı ayrı çalışır. Böylece algoritmamızın tahmin hatalarını kontrol altına almaya nasıl yardımcı olduğunu görme şansımız olmuştur. Her hücre için, kümeleme amacıyla aday komşular alınır ve aglomeratif bir şekilde bağımsız olarak beş hiyerarşik küme oluşturulur. Bu beş küme arasından, kaynak hücre ile en yüksek mobilite aktivitesinde olan küme seçilmiştir. Yetersiz yeni hücre geçmişini tamamlamak için seçilen küme hücrelerinin birleştirilmiş trafik özelliklerini kullanılmıştır. Yeni hücrenin sınırlı geçmişinin seviyesini ve birleştirilmiş küme trafiğinin seviyesini karşılaştırdık. Böylece yeni hücre için sentetik bir geçmiş oluştururken orantılı bir karakteristik uyarlanabilmiştir. Bununla beraber karşılaştırma amacıyla en iyi ikinci yüksek mobilite seviyesine sahip küme de ele alınmıştır. Böylece, kıyaslama adına iki küme aldığımız için, "küme 1" en yüksek hareketliliğe sahip kümeyi (ortalama günlük el değiştirme

sayısı olarak), "küme 2" ise aynı kapsamda en yüksek ikinci kümeyi temsil eder. Tahmin algoritması ve kullanılan geçmiş başına ortalama MAPE hata değeri elde edilmiştir.

Kıyaslama aşaması için iş akışı

- Retrieve all test cell pm and cm
 - Aggregate data into 15 days intervals
 - for each test cell
 - * Forecast next 8 months with SARIMAX with grid-search and insufficient history
 - * Forecast next 8 months with STL-ETS with insufficient history
 - * Forecast next 8 months with Vanilla LSTM with selected hyperparameters and insufficient history
 - * Forecast next 8 months with LSTM Encoder-Decoder with selected hyperparameters and insufficient history
 - * Forecast next 8 months with CNN-LSTM Encoder-Decoder with selected hyperparameters and insufficient history
 - * Forecast next 8 months with ConvLSTM Encoder-Decoder with selected hyperparameters and insufficient history
 - * Calculate MAPE of each forecast and save results to DB
 - * Get all non-cosite cell neighbors
 - * Calculate site to site distance
 - * Calculate relative azimuth
 - * Get ho attempts
 - * Scale three features
 - * Apply clustering for 5 clusters
 - * Select top two clusters by ho attempt
 - * for each cluster (cluster 1 and cluster 2)
 - + Aggregate cluster cells traffic data
 - + Aggregate traffic into 15 days intervals
 - + Backcast synthetic history to cell from this aggregated traffic proportionally
 - + Forecast next 8 months with SARIMAX with grid-search and synthetic history
 - + Forecast next 8 months with STL-ETS with synthetic history
 - + Forecast next 8 months with Vanilla LSTM with selected hyperparameters and synthetic history
 - + Forecast next 8 months with LSTM Encoder-Decoder with selected hyperparameters and synthetic history
 - + Forecast next 8 months with CNN-LSTM Encoder-Decoder with selected hyperparameters and synthetic history
 - + Forecast next 8 months with ConvLSTM Encoder-Decoder with selected hyperparameters and synthetic history
 - + Calculate MAPE of each model and save results to DB for cluster
 - * end for
 - end for
 - Retrieve all MAPE results for each forecasting method, cluster type and used history
 - List average MAPE for each forecasting method, cluster type and used history
-

4.2.4. Deneysel Çıktılar

Bu bölümde, kullanılan her bir tahmin algoritması için toplam doğruluk sonuçlarını paylaşılmıştır. Ayrıca, yetersiz seviyedeki verileri kullandığımızda durumun nasıl daha da kötüye gidebileceği konusunda okuyuculara bir fikir vermek için görselleştirilmiş tahmin

hataları paylaşılmıştır. Sonuçlar, algoritmamızın MAPE skorunu %30 civarında tatmin edici bir seviyede tutmaya yardımcı olduğunu ve yeni konuşlandırılmış hücreler için de planlama mühendisleri için güven sağladığını göstermektedir. Bu nedenle, sınırlı kullanılabilir geçmiş nedeniyle son derece yanlış tahmin sonuçlarının otomatik planlama çözümlerinin yanlış yatırımlara sebep olmasını önleyebilmektedir.

Her model ve kullanılan geçmiş verisi için doğruluk sonuçları Tablo 4.5'te listelenmiştir. Sınırlı olan mevcut veriler yalnızca 1,5 aylık günlük trafik verileri içerdiğinden, hücrenin yıllık mevsimsel özelliklerini çıkarmak mümkün değildir. Bu yüzden hücrenin 10 ay sonrası gibi uzun vadeli tahminlerde yüksek hatalara neden olabilecek seviye veya eğilim bileşenleri gözlemlenebilmektedir. SARIMAX gibi geleneksel doğrusal zaman serisi modelleri ve STL-ETS gibi zaman serilerini bileşenlerine ayırarak tahminleyen modeller için ortalama hata seviyeleri, 1,5 aylık verileri kullanarak gelecek 10 ayı tahmin etmede sırasıyla yaklaşık %133 ve %124'tür. Bu yöntemler için devasa MAPE seviyeleri, yeni hücreler için yalnızca eğilim veya seviye bileşenlerinin modellenmesinden kaynaklanmaktadır.

Bu hücreler için sentetik geçmiş oluşturduğumuzda ve geriye dönük tahmin yaparak mevcut verilerini 1,5 aydan 24 aya çıkardığımızda, aynı tahmin yöntemlerinin mevsimsel bileşeni öğrenmesi mümkündür. Dolayısıyla, MAPE seviyelerinin SARIMAX ve STL-ETS için sırasıyla %27 ve %32 seviyelerine düşebildiği gözlemlenmiştir. En azından bu MAPE seviyesi, otomatikleştirilmiş kapasite planlama algoritmaları için gereksiz yatırım kararlarını önleyebilmektedir.

Ayrıca dört farklı mimari kullanarak çağdaş LSTM modellerinde sentetik geçmiş yaklaşımının hata düzeyini de test ettik. Bir sinir ağı modeli eğiterek 1,5 aylık verilerle bir tahmin elde etmek mümkün olmadığı için sadece geriye dönük geçmiş kullanım senaryosunu kontrol ettik. CNN-LSTM Kodlayıcı-Kod Çözücü, Conv-LSTM Kodlayıcı-Kod Çözücü, LSTM Kodlayıcı-Kod Çözücü ve Vanilla LSTM ağları için ortalama MAPE seviyeleri sırasıyla %35, %37, %37 ve %38'dir. Her hücre için trafik verilerindeki yüksek değişkenlik nedeniyle, verileri saha veya bölge seviyelerinde bileştirmeden zaman serisi örüntülerini kavramak zordur. Çalışmamızda elde edilen hata değerlerini karşılaştırdığımızda, evrişim katmanlarının yumuşatma etkisi, kodlayıcı-kod çözücü modellerine göre küçük avantajlar sağlamıştır. Ancak, sinir ağlarının veri açlığı nedeniyle iki yıllık geçmişe kadar genişlese de bu seviye sinir ağı modellerini eğitmek

için hala çok sınırlıdır. Doğrusal zaman serisi modellerinin yaklaşımı, sinir ağı modellerine göre daha düzgündür. Bu nedenle hata seviyeleri de LSTM modellerinden biraz daha düşüktür.

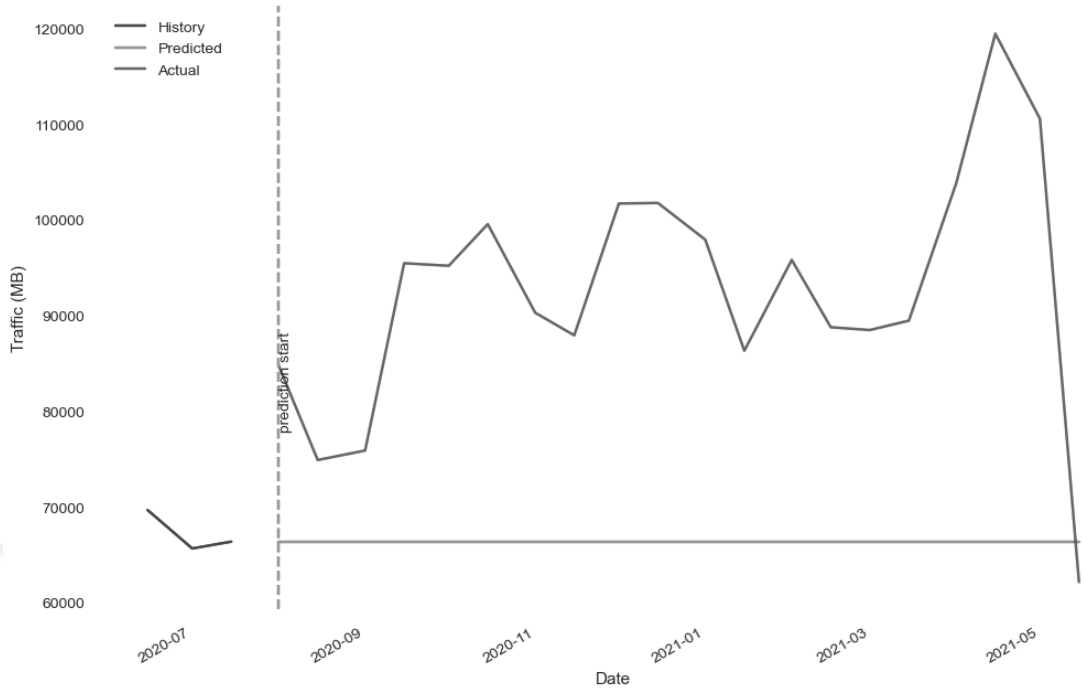
Çalışmamızda ayrıca, mobilite faaliyetlerinin sentetik geçmiş üretimi için hücre seçimi üzerindeki etki gücünü anlamak amacıyla farklı kümelerin tercih edilmesini de kıyasladık. Önceki bölümlerde açıkladığımız gibi, günlük ortalama hücreler arası geçiş talepleri açısından Küme 1 en yüksek küme, Küme 2 ise ikinci en yüksek kümedir. Tablo 4.5'te görüldüğü gibi, ortalama MAPE'de Küme 1 için hata düzeyi %34, Küme 2 için ise %41 seviyesindedir. Dolayısıyla, hücre seçiminde yüksek mobilitenin benzer trafik özelliklerine sahip olması üzerindeki etkisi, mesafe ve göreceli azimut ile karşılaştırıldığında daha belirgindir.

Özetle, yetersiz veriye sahip hücreler için trafik tahmini yaklaşımımızı kullanarak, ortalama hata seviyeleri %90 oranında iyileştirilmiştir ve %128 MAPE seviyelerinden ortalama %34 MAPE seviyelerine düşürülerek kontrol altına alınmıştır. Ortalama MAPE seviyeleri Tablo 4.5'te listelenmiştir.

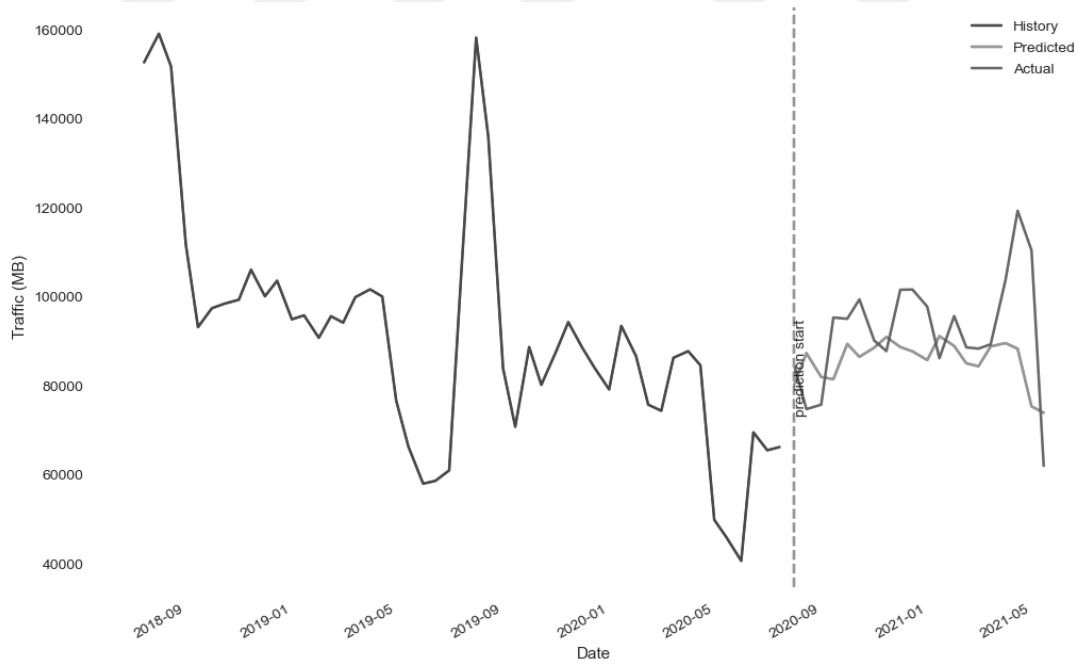
Tablo 4.5. Yetersiz ve Sentetik Geçmişin Tahminleme Doğruluk Sonuçları

METHOD	Top First Mobility Cluster		Top Second Mobility Cluster	
	CLUSTER 1		CLUSTER 2	
	AVG(MAPE)	MEDIAN(MAPE)	AVG(MAPE)	MEDIAN(MAPE)
SARIMAX with Cluster Based Backcasted, Synthetic History	0.27	0.27	0.35	0.31
STL-ETS with Cluster Based Backcasted, Synthetic History	0.32	0.30	0.36	0.31
CNN-LSTM Encoder-Decoder with Cluster Based Backcasted, Synthetic History	0.35	0.33	0.43	0.36
Conv-LSTM Encoder-Decoder with Cluster Based Backcasted, Synthetic History	0.37	0.35	0.43	0.35
LSTM Encoder-Decoder with Cluster Based Backcasted, Synthetic History	0.37	0.32	0.42	0.39
Vanilla LSTM with Cluster Based Backcasted, Synthetic History	0.38	0.34	0.48	0.37
STL-ETS with Insufficient, Limited, Available History	1.24	0.63	1.24	0.63
SARIMAX with Insufficient, Limited, Available History	1.33	0.72	1.34	0.72

Bu bölümde okuyucuya fikir vermesi için bazı tahmin hatası görselleri paylaşıyoruz. Örneğin, Şekil 4.30'da SARIMAX, 1,5 aylık mevcut verilerden sadece seviye bileşeni oluşturabilmiştir. Ancak mevsimsel özellikler çok düzgün ve katı olmasa da hücrenin gerçek trafiğinde pozitif bir eğilim bileşeni vardır. Şekil 4.31'de, sentetik olarak oluşturulmuş geriye dönük 24 aylık geçmişin yardımıyla çok daha iyi bir tahmin elde etmek mümkün olmuştur.



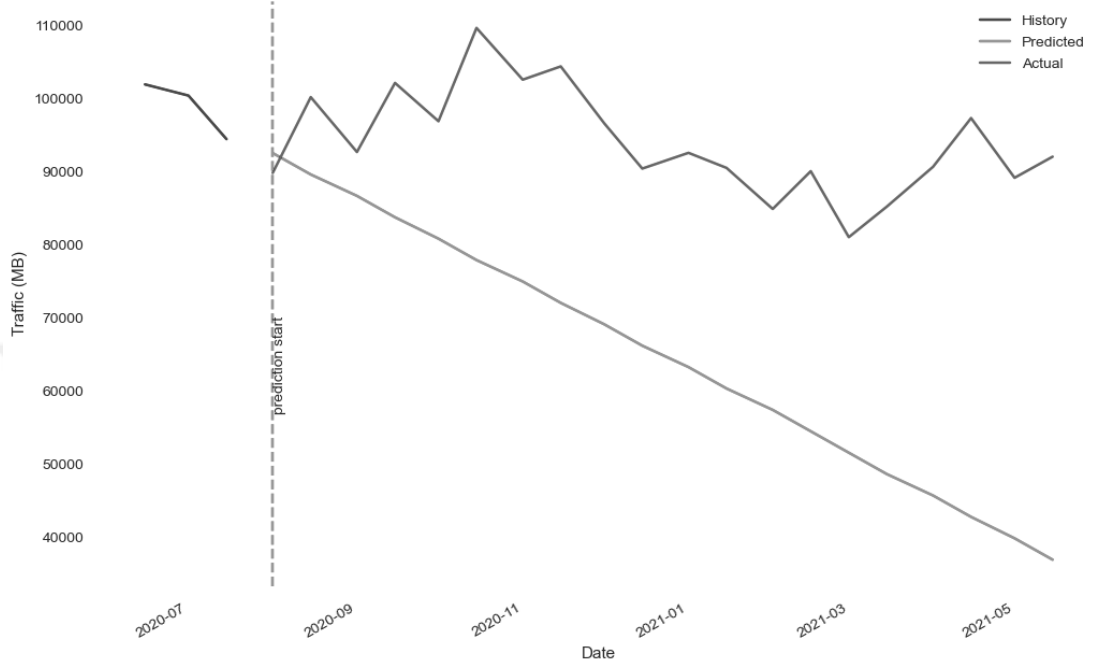
Şekil 4.30. SARIMAX Yönteminin Limitli Veri Üzerinde Tahminleme Hataları



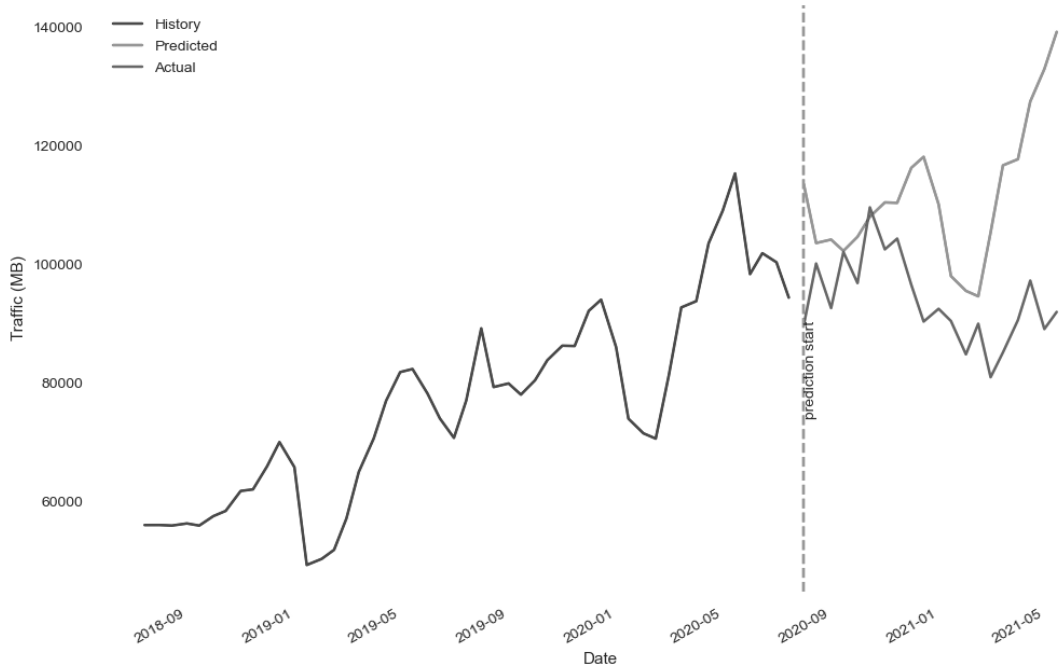
Şekil 4.31. SARIMAX Yönteminin Sentetik Veri Üzerinde Tahminleme Hataları

Benzer şekilde, Şekil 4.32’de STL-ETS, yıl seviyesine yayılması doğru olmayan eğilim bileşeninde hapsolmuştur. Bu nedenle, STL-ETS, oldukça aşağı yönlü bir trafik tahminlemekte

ve büyük bir hata seviyesine neden olabilmektedir. Algoritmamız tarafından üretilen sentetik geçmiş ise hücre için belirli bir seviyeyi tahmin etmesine yardımcı olmuştur. MAPE, Şekil 4.33'te de görüleceği üzere sınırlı olarak mevcut veri tahminine kıyasla oldukça düşürülmüştür.

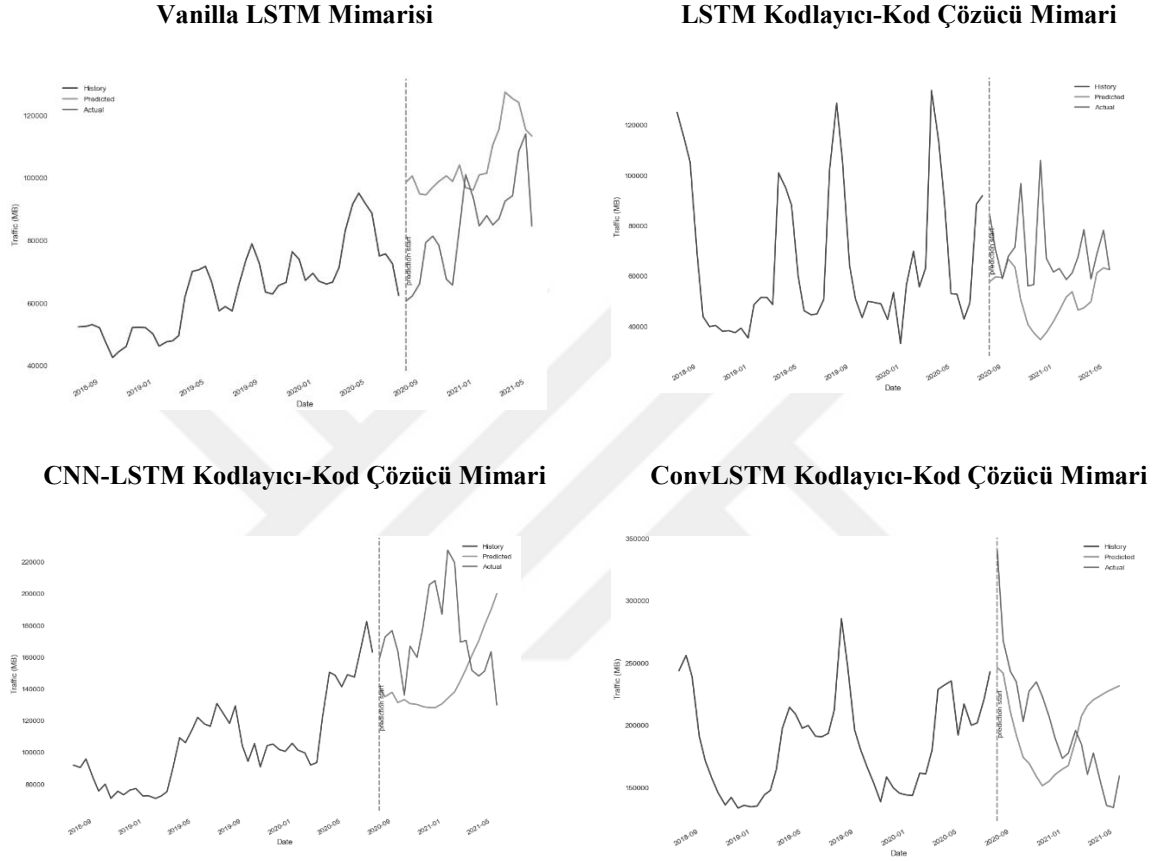


Şekil 4.32. STL-ETS Yönteminin Limitli Veri Üzerinde Tahminleme Hataları



Şekil 4.33. STL-ETS Yönteminin Sentetik Veri Üzerinde Tahminleme Hataları

Şekil 4.34'te LSTM mimarilerinin çıktısı olarak farklı tahmin görselleri bulunmaktadır. Trafik verilerinin yüksek varyansı nedeniyle, düzleştirilmiş seviyeler ve eğilim tahmini için yeterli eğitim verisi yoktur. Ancak hata seviyelerinin sentetik geçmişin yardımıyla kontrol altında tutulduğu gözlemlenmiştir.



Şekil 4.34. Farklı LSTM Mimarilerinin Sentetik Veri Üzerinde Tahminleme Hataları

5. TARTIŞMA VE SONUÇ

Bu doktora tez çalışması kapsamında, tahminin nedeni çeşitli amaçlarla da olsa, mobil ağ trafiği tahmini kapsamında literatürde bulunabilen tüm çalışmalar incelenmiş ve karmaşıklık, ağ seviyesi, tahmin ufku uzunluğu, kullanılan teknikler ve korelasyon seçenekleri dahil olmak üzere farklı boyutlarda sınıflandırılmıştır. Çalışmaları karmaşıklık kategorilerinde sınıflandırırken, yalnızca trafik geçmişinin doğrusal özelliklerini değil, aynı zamanda doğrusal olmayan özelliklerin kapsamı da dikkate alınmıştır. Kısa vadeli tahminleri ele alan çalışmalar çok daha yoğundur. Az sayıda çalışma ise orta vadeli veya uzun vadeli tahminleri ele almaktadır. Doğrusal teknikler, eğilim ve mevsimsellik faktörlerinde artan varyans nedeniyle özellikle uzun vadeli tahminlerde daha zayıftır. Ancak aynı zamanda, geliştirilmiş uzun vadeli doğruluk için çoklu mevsimsellik bileşenleri sağlayabilen Mevsimsel ARIMA ve HW-ETS dahil olmak üzere zaman serisi ayrıştırma yaklaşımları gibi doğrusal modeller de vardır. Yine de doğrusal temel modeller serilerin temel özelliklerini takip ederler ve bu da oynak ve doğrusal olmayan ortamlarda doğruluklarını azaltır. Bu noktada, başarılı eğitim için artan veri miktarı ve hesaplama kaynakları için gerekli boyutlandırma gibi farklı ön koşullara sahip olsalar bile, yapay öğrenme modelleri ve derin sinir ağları daha doğru sonuçlar üretebilmektedirler. Zaman serisi tekniklerinin parametrelerini tahmin etmek, ilgili alan üzerinde biraz uzmanlık gerektirir ve rastgele arama yöntemleri sebebiyle zaman alıcı olabilir. Genellikle kısa vadeli tahmin, ağ kaynaklarının dinamik olarak yönetimi için yetersiz veri sorunu olmadan çalışılabilmektedir. Mobil radyo ağlarının uzun vadeli tahminini çalışmanın ana engellerinden biri, akademi için yeterli veri kaynağına sahip olmanın sınırlılığı olabilir. Ayrıca, birçok MNO'nun uzun vadeli istatistikleri tutmak için yatırımı yoktur. Dolayısıyla tüm bu nedenler, mobil radyo ağlarının uzun vadeli tahmin ve analizi üzerinde çalışmak için engeli artırmaktadır. Mobil ağların tahminleme konuları üzerinde çalışacaklar için açık veri kaynakları da tez çalışması ile ortaya çıkarılan literatür taraması yayını kapsamında paylaşılmış, önceki tüm çalışmalar listelenmiştir.

Otonom ve veri odaklı kapasite planlama yaklaşımı mobil ağ bileşenlerinin uzun vadeli talep tahminlemesine ihtiyaç duymaktadır. Bu tez çalışması kapsamında günlük trafik hacmi miktarı üzerinden mobil ağların kullanım taleplerinin uzun vadeli tahminlemesinde karşılaşılan başlıca iki önemli soruna çözüm geliştirecek algoritmalar tasarlanmıştır.

Bu sorunlar;

- Mevsimsellik Problemi: mobil radyo hücrelerinin değişim noktası farkında mevsimsellik tespiti ve
- Tahminleme Problemi: yeni devreye alındığı için yetersiz geçmişe sahip olan mobil radyo hücrelerinin uzun gelecek trafik tahminleme hatalarının kontrol altına alınması

şeklinde ifade edilebilir. İlk sorun için, mevsimsellik tespiti ve mobil ağ baz istasyonu trafik verilerinin ayrıştırılması için, seviye kayması veya üstel eğilim artışı nedeniyle değişim noktası varlığı dikkate alınarak baz istasyonları üzerindeki mevsimsellik etkisini çıkartan bir algoritma geliştirilmiş, canlı mobil ağ verisi ile test edilmiş ve detayları paylaşılmıştır. Mobil ağlar için otomatik kapasite planlaması, geçmiş kalıpları kullanarak trafik talebinin uzun vadeli tahminini gerektirir. Doğru yatırım zamanına ve doğru kapasite genişletme boyutuna karar vermek veya dışsal özelliklere sahip tahmin algoritmalarının doğruluğunu iyileştirmek için hem mevsimsel ayrıştırma hem de mevsimsel dönem tanımlaması, yatırım kararlarının doğruluğunu iyileştirir.

Tez çalışmasında planlama uzmanları tarafından etiketlenmiş 150 baz istasyonu bulunmaktadır ve bu baz istasyonları arasından 100 tanesi doğruluk testi için kullanılmakta, kalan 50 baz istasyonundan oluşan başka bir grup, keşif analizi, eşikler için parametre ayarı ve ilk model testleri için kullanılmaktadır. Algoritmanın dört ana çıktısı vardır;

- 1- **Mevsimsel etiketleme:** Algoritma, baz istasyonunu mevsimsel desen ve etki düzeyine göre “mevsimsel olan” veya “mevsimsel olmayan” şeklinde etiketlemektedir. Baz istasyonun gerçek mevsimselliği ile eşleşirse doğru sonuç elde edildiği düşünülmektedir.
- 2- **Mevsimsel etkinin gücü:** Algoritma mevsimsel trafik miktarının eğilim trafik miktarına oranına ilişkin SPI bazlı bağıl etki skorunu hesaplamaktadır.
- 3- **Değişim noktasının varlığı:** Algoritma baz istasyonunun geçmiş çevrimdışı verilerinde değişim noktaları olup olmadığını tespit etmektedir. Değişim noktalarının gerçek varlığı ile eşleşiyorsa doğru sonuç elde edildiği düşünülmektedir.

- 4- **Mevsim süresi:** Algoritma, mevsimsel etki sırasında gün cinsinden süreyi hesaplamaktadır. Bu süre boyunca eğilim ve mevsimsellik oranı sürekli olarak %30'un üzerindedir.

Geliştirilen algoritma doğru yatırım zamanına ve doğru kapasite genişletme boyutuna karar vermek veya dışsal özelliklere sahip tahmin algoritmalarının doğruluğunu iyileştirmek için hem mevsimsel ayrıştırma hem de mevsimsel dönem tanımlaması karar doğruluğunu iyileştirmektedir. Bu özellikleri canlı ağ verileri üzerinde gelişmiş doğrulukla hesaplamak için, parçalı STL ayrıştırmasını ve kendi içinde Laplace öncül kullanan Prophet kütüphanesinin bağlanımını barındıran hibrit bir algoritma tasarlanmıştır. Bir araya getirilen her iki yöntemin de zayıf ve güçlü yanlarının farkındalığıyla birleştirmekte ve genel çıktıyı değişim noktası ve benzerlik analizi ile güçlendirmektedir. Geliştirilen hibrit algoritma ile bu yöntemlerin tek başlarına kullanım ortalamaları karşılaştırıldığında, elde edilen doğru etiketlemenin F1 karmaşıklık skoru olarak %18,6 oranında iyileştiği ve 0,89 seviyesinde bir skor elde edildiği gözlemlenmektedir. Ayrıca, problemin karmaşıklığını artıran ve ayrıştırma doğruluğunu azaltan bazı özel durumlar da tespit edilmiş ve paylaşılmıştır. Her yöntemin çıktıları canlı ağ verileri üzerinde görselleştirilmiştir. Gerekli tüm ön işleme, veri temizleme ve anomali iyileştirme işlemleri önceden tamamlanmıştır. Yapılan analiz, farklı amaçlar için trafik ayrıştırma gereksinimlerine tek değişkenli, çevrimdışı, parametrik ve denetimli bir şekilde yaklaşır. Daha ileri çalışma konusu olarak, denetimsiz, çevrimiçi, sınırlı geçmişe sahip, parametrik olmayan, çok değişkenli (örneğin yani kaynak bloğu kullanılamaması) farklı bir çözüm de çalışılabilir. Bu tez çalışması kapsamında geliştirilen algoritma ve elde edilen bulgular araştırma makalesi olarak yayına dönüştürülmüştür.

Düşük kaliteli geçmişe sahip yeni konuşlandırılan baz istasyonlarının mevsimsel etkisinin ayrıştırılması da konu uzmanlarıyla yaptığımız tartışmalara dayanan, yukarıda da paylaştığımız, diğer bir araştırma alanı olarak bu tez çalışmasında ele alınmıştır. Ağlar, artan radyo erişim gereksinimini karşılamak için yeni hücreler ve baz istasyonları ile doğal olarak büyüdüğünden, bu yeni konuşlandırılan hücreler, uzun vadeli kapasite talebini doğru bir şekilde tahmin etmek için yeterli geçmiş veriye sahip değildirler. Bu sorunun üstesinden gelmek ve bu hücreler için de yeterli tahminleme doğruluğu seviyesine sahip olmayı mümkün kılmak için, ilgili hücrelerin uzun vadeli tahminlerini iyileştirmek için sentetik geçmiş oluşturmak adına çok değişkenli hiyerarşik kümeleme tabanlı bir algoritma geliştirilmiştir. SARIMAX, STL-ETS gibi klasik

zaman serisi yöntemleri ve Vanilla LSTM Sinir Ağları, LSTM Kodlayıcı-Kod Çözücü Ağları, CNN-LSTM Kodlayıcı-Kod Çözücü Ağları, ConvLSTM Kodlayıcı-Kod Çözücü Ağları gibi derin sinir ağı yöntemleri dahil olmak üzere farklı tahmin yöntemlerinin tahminini iyileştirmek için geliştirilen yaklaşım test edilmiştir. Çözüm, farklı kümeler için canlı bir mobil ağdan ölçülen 70 LTE hücresinin gelecek tahminlemede kullanılmıştır. Hiyerarşik kümeye dahil edebilmek adına komşu hücreler de dahil edilerek toplam 3461 LTE hücre verisi işlenmiş ve hesaba katılmıştır. Toplam 3 yıllık günlük geçmiş verisi kullanılmıştır. Bu probleme özel geliştirdiğimiz algoritma ile 10 ay sonrası için trafik tahminlemede büyük tahmin hataları ortalama %128 MAPE seviyesinden %35 MAPE seviyelerine düşürülebilmektedir. Ayrıca, sınırlı geçmişe sahip hücreler için sentetik geçmiş oluşturmak üzere toplu trafik karakteristiklerini miras alacağımız ve tahmin yapılacak hücreye özel oluşturulan kümeye dahil edeceğimiz komşu hücreleri seçerken de mobilite aktivitelerinin nitelik olarak önemi araştırılmıştır. Mobilite ilişkilerini baskın nitelik olarak göz önünde bulundurmanın, mesafe veya göreceli azimut kriterlerine kıyasla daha faydalı sonuçlar sağladığı gözlemlenmiştir.

Gelecekteki çalışmalar için, tahmin doğruluğunu artırmak adına, daha fazla değişken veya yeni hücrelerin özel günlerde davranışlarını tahmin etmek için diğer bazı dışsal faktörler eklemek mümkündür. LSTM modelleri, sınırlı geçmişe sahip hücrelerin geleceğini tahmin ederken coğrafik uzamsal ilişkilerin daha fazla farkında olmak için Grafikli Sinir Ağları ile genişletilebilir. Trafik hacmi verileri dışındaki diğer KPI'lar, gelecekteki bir çalışma olarak uzun vadeli tahminlerde hücreler için de incelenebilir. Bu ikinci probleme özel geliştirilen tahminleme iyileştirme algoritması da diğer algoritma gibi bulguları ile birlikte araştırma makalesi olarak yayına gönderilmiştir. Tez çalışması kapsamında yapılan çalışmalar birisi gözden geçirme, diğer ikisi ise araştırma makalesi olmak üzere toplam üç akademik yayına içerik oluşturmuştur. |

KAYNAKLAR

- Adeleke, O. A., 2019, Echo-State Networks for Network Traffic Prediction, *IEEE Annual Information Technology, Electronics and Mobile Communication Conference*, 202–206.
- Argedor, <https://www.argedor.com/clickhouse>, Erişim Tarihi: 12 Mayıs 2021
- Aminikhanghahi, S., Cook, D. J., 2017, A Survey of Methods for Time Series Change Point Detection, *Knowledge and Information Systems*, 51(2), 339–367.
- Anand, N., Scoglio, C., Natarajan, B., 2008, GARCH - Non-linear time series model for traffic modeling and prediction, *NOMS - IEEE/IFIP Network Operations and Management Symposium*, 694–697.
- Andreoletti, D., Troia, S., Musumeci, F., Giordano, S., Maier, G., Tornatore, M., 2019, Network Traffic Prediction based on Diffusion Convolutional Recurrent Neural Networks, *INFOCOM WKSHPs, IEEE Conference on Computer Communications Workshops*, 246–251.
- Au, S., Ma, G., Yeung, S., 2011, Automatic Forecasting of Double Seasonal Time Series with Applications on Mobility Network Traffic Prediction, Joint Statistical Meetings, erişim Tarihi: 17 Ekim 2020.
- Azari, A., Papapetrou, P., Denic, S., Peters, G., 2019, Cellular Traffic Prediction and Classification: A Comparative Evaluation of LSTM and ARIMA, *Lecture Notes in Computer Science*, 11828, 129–144.
- Balke, N. S., 1993, Detecting Level Shifts in Time Series, *Journal of Business & Economic Statistics*, 11, 81–92
- Basseville, M., Nikiforov, I., 1993, Detection of Abrupt Change Theory and Application.
- Bayramli, B., 2021, https://burakbayramli.github.io/dersblog/algs/algs_140_rnn/kendini_tekrarlayan_yapay_sinir_aglari_recurrent_neural_network_rnn_.html, Erişim Tarihi: 16 Mayıs 2021.
- Bengio, Y., Simard, P., Frasconi, P., 1994, Learning Long-Term Dependencies with Gradient Descent is Difficult, *IEEE Transactions on Neural Networks and Learning Systems*, vol. 5, no. 2, 157–166.
- Bilginç, 2021, <https://bilginc.com/tr/blog/158/python-nedir-python-hakkinda-hersey>, Erişim Tarihi: 15 Mayıs 2021.
- Box, G. E. P., Jenkins, G. M., 1970, Time series analysis: Forecasting and control. San Francisco: Holden-Day.
- Burg, G. J. J., Williams, C. K. I., 2020, An Evaluation of Change Point Detection Algorithms.

- Cao, A., Qiao, Y., Sun, K., Zhang, H., Yang, J., 2018, Network traffic analysis and prediction of Hotspot in cellular network, *IC-NIDC, IEEE International Conference on Network Infrastructure and Digital Content*, 452–456.
- Chen, G., 2018, Spatiotemporal Individual Mobile Data Traffic Prediction, *Technical Report, RT-0497, INRIA Saclay - Ile-de-France*.
- Chen, H., Zhang, N., 2015, Graph-based change-point detection, *Annals of Statistics*, 43(1), 139–176.
- Chen, L., Yang, D., Zhang, D., Wang, C., Li, J., Nguyen, T., 2017, Deep mobile traffic forecast and complementary base station clustering for C-RAN optimization, *Journal of Network and Computer Applications*, 121, 59–69.
- Chmieliauskas, D., Gursnys, D., 2019, LTE Cell Traffic Grow and Congestion Forecasting, *Open Conference of Electrical, Electronic and Information Sciences*, 1–5.
- Cleveland, R. B., Cleveland, W. S., McRae, J. E., Terpenning, I. J., 1990, STL: A seasonal-trend decomposition procedure based on loess, *Journal of Official Statistics*, 6(1), 3–33.
- ClickHouse, <https://clickhouse.tech/docs/en/introduction/distinctive-features>, Erişim Tarihi: 12 Mayıs 2021.
- Cisco, 2018, Cisco Annual Internet Report (2018–2023), Erişim Tarihi: 12 Mayıs 2021
- Cortez, P., Rio, M., Rocha, M., Sousa, P., 2006, Internet traffic forecasting using neural networks, *IEEE International Conference on Neural Networks*, 2635–2642.
- Cortez, P., Rio, M., Sousa, P., Rocha, M., 2008, Forecasting internet traffic by neural networks under univariate and multivariate strategies, *ANNIP, ICINCO, International. Workshop on Artificial Neural Networks and Intelligent Information Processing*, 61–70.
- Cui, H., Yao, Y., Zhang, K., Sun, F., Liu, Y., 2015, Network traffic prediction based on Hadoop, *International Symposium on Wireless Personal Multimedia Communications*, 29–33.
- Çetinkaya, A., 2021, <https://www.minepla.net/2015/05/visual-studio-code-nedir-ne-degildir/>, Erişim Tarihi: 15 Mayıs 2021.
- Dagum, E. B., Bianconcini, S., 2016, Seasonal adjustment methods and real time trend-cycle estimation in Statistics for Social and Behavioral Sciences. Springer.
- Dalgkisis, A., Louta, M., Karetos, G. T., 2018, Traffic forecasting in cellular networks using the LSTM RNN, *ACM International Conference*, 28–33.
- Dawoud, S., Uzun, A., Gondor, S., Kupper, A., 2014, Optimizing the power consumption of mobile networks based on traffic prediction, *International Computer Software and Applications Conference*, 279–288.

- Downey, A. B., 2008, A novel changepoint detection algorithm.
- Douglas, K., Douglas, S., PostgreSQL, 2003, *A comprehensive guide to building, programming and administering PostgreSQL databases*, Sams Publishing.
- Drake, J. D., Worsley, J. C., 2002, *Practical PostgreSQL*, O'Reilly Media, Inc.
- El Hag, H. M. A., Sharif, S. M., 2007, An adjusted ARIMA model for internet traffic, *IEEE AFRICON Conference*.
- Erp S. V., Oberski D. L., Mulder J., 2019, Shrinkage priors for Bayesian penalized regression, *Journal of Mathematical Psychology*, 89, 31–50.
- Feng, G., 2016, Network traffic prediction based on neural network, *International Conference on Intelligent Transportation, Big Data and Smart City*, 527–530.
- Feng, J., Chen, X., Gao, R., Zeng, M., Li, Y., 2018, DeepTP: An End-to-End Neural Network for Mobile Cellular Traffic Prediction, *IEEE Network*, 32(6), 108–115.
- Gould, P. G, Koehler, A. B., Ord, J. K., Snyder, R. D., Hyndman R. J., Vahid-Araghi, F., 2008, Forecasting time series with multiple seasonal patterns, *European Journal of Operational Research*, 191, 207–222.
- Gowrishankar, S., Satyanarayana, P. S., 2009, A Time Series Modeling and Prediction of Wireless Network Traffic, *International Journal of Interactive Mobile Technologies*, 3(1), 40–52.
- GSMA Intelligence, 2020, Mobile Economy Research Report. Erişim Tarihi: 21 Mart 2021.
- Guang, C., Jian, G., Wei, D., 2005, A time-series decomposed model of network traffic, *Lecture Notes in Computer Science*, 3611, 338–345.
- Harvey, A. C., Shephard, N., 1993, Structural time series models in Handbook of Statistics, *Elsevier*.
- Hochreiter, S., Schmidhuber, J., 1997, Long Short-Term Memory, *Neural Computation*, vol. 9, no. 8, 1735–1780.
- Holt, C. E., 1957, Forecasting seasonals and trends by exponentially weighted averages, *Carnegie Institute of Technology*.
- Huang, C., Chen, P., 2019, Mobile Traffic Offloading with Forecasting using Deep Reinforcement Learning, *IEEE Access*, 8, 1–9.
- Hyndman, R. J., Khandakar, Y., 2008, Automatic Time Series Forecasting: The forecast Package for R, *Journal of Statistical Software*, 27(3), 22.

- Imashev, B., Azamat, N., Sidelkovskiy, A., Sidelkovskaya, A., 2019, The Practice of Moving to Big Data on the Case of the NoSQL Database, Clickhouse, *World Congress on Global Optimization, Optimization of Complex Systems: Theory, Models, Algorithms and Applications*, 991, 820-828
- Jaccard, P., 1912, The Distribution of the Flora in the Alpine Zone, *New Phytologist*, 11(2), 37–50.
- Jaffry, S., 2020, Cellular Traffic Prediction with Recurrent Neural Network, Cornell University, *Computer Science, Networking and Internet Architecture*.
- Jain, A.K., Murty, M.N., Flynn, P.J., 1999, Data clustering: A review, *ACM Computing Surveys*, 264–323.
- Jia, G., Yu, P., Xiyuan, P., Qiang, C., Jiang, Y., Yufeng, D., 2009, Traffic forecasting for mobile networks with multiplicative seasonal ARIMA models, *International Conference on Electronic Measurement and Instruments*, 3377–3380.
- Jiang, D., Xu, Z., Xu, H., 2015, A novel hybrid prediction algorithm to network traffic, *Annals of Telecommunications*, 70, 427–439.
- Jung, S., Kim, C., Chung, Y., 2006, A prediction method of network traffic using time series models, *Lecture Notes in Computer Science*, 3982, 234–243.
- Junsong, W., Jiukun, W., Maohua, Z., Junjie, W., 2009, Prediction of internet traffic based on Elman neural network, *CCDC, Chinese Control and Decision Conference*, 1248–1252.
- Katris, C., Daskalaki, S., 2015, Combining time series forecasting methods for internet traffic, *Springer in Mathematics and Statistics*, 122, 309–317.
- Killick, R., Eckley, I. A., 2014, Changepoint: An R Package for Changepoint Analysis, *Journal of Statistical Software*, 58(3).
- Kim, M., 2020, Network traffic prediction based on INGARCH model. *Wireless Networks*.
- Koutroumbas, K., Theodoridis, S., 2009, Pattern Recognition, Academic Press.
- Lakshmanan, A., Das, S., 2017, Two-stage models for forecasting time series with multiple seasonality.
- Li, J., Shen, L., Tong, Y., 2009, Prediction of network flow based on wavelet analysis and ARIMA model, *International Conference on Wireless Networks and Information Systems*, 217–220.
- Livera, A. M., Hyndman, R. J., Snyder, R. D., 2011, Forecasting time series with complex seasonal patterns using exponential smoothing, *Journal of the American Statistical Association*, 106(496), 1513–1527

- Lopez, M., Carro, B., Sanchez, A., 2019, Neural network architecture based on gradient boosting for IoT traffic prediction, *Future Generation Computer Systems*, 100, 656–673.
- Lutz, M., 2001, *Programming Python*, O'Reilly & Associates Inc.
- Matthew, N., Stones, R., 2005, *Beginning Databases with PostgreSQL From Novice to Professional*, Appress Publishing.
- Miao, D., Qin, X., Wang, W., 2014, The periodic data traffic modeling based on multiplicative seasonal ARIMA model, *WCSP, International Conference on Wireless Communications and Signal Processing*, 2–6.
- Miao, D., Sun, W., Qin, X., Wang, W., 2016, MSFS: Multiple Spatio-temporal Scales Traffic Forecasting in Mobile Cellular Network, *PICom, IEEE International Conference on Pervasive Intelligence and Computing, IEEE International Conference on Big Data*, 787–794.
- Myatt, G. J., 2009, *Making Sense of Data II*, Wiley.
- Narmanlioglu, O. and Zeydan, E., 2018, Mobility-aware cell clustering mechanism for self-organizing networks, *IEEE Access*.
- Network Traffic Prediction, *Joint Statistical Meetings*, Erişim Tarihi: 21 Nisan 2021.
- Nie, L., Wang, X., Wan, L., Yu, S., Song, H., Jiang, D., 2018, Network Traffic Prediction Based on Deep Belief Network and Spatiotemporal Compressive Sensing in Wireless Mesh Backbone Networks, *Wireless Communications and Mobile Computing*.
- Nikravesh, A. Y., Ajila, S. A., Lung, C.H., Ding, W., 2016, An Experimental Investigation of Mobile Network Traffic Prediction Accuracy, *Services Transactions on Big Data*, 3(1), 1–16.
- Ntalampiras, S., Fiore, M., 2018, Forecasting Mobile Service Demands for Anticipatory MEC, *WoWMoM, IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 14–19.
- Oliphant, T. E., 2007, Python for Scientific Computing, *Computing in Science & Engineering*, 9(3), 10–20.
- Öztürk, M., 2021, <https://miracozturk.com/python-kutuphaneleri-ve-ozellikleri/>, Erişim Tarihi: 15 Mayıs 2021
- Page, E. S., 1954, Continuous inspection schemes, *Biometrika*, 41, 100–115.
- Pankratz, A., 1991, *Forecasting with Dynamic Regression Models*, John Wiley and Sons Press.
- Park, D., Woo, D., 2009, Prediction of network traffic by using dynamic bilinear recurrent neural network, *ICNC, International Conference on Natural Computation*, 419–423.

- Peng, W., Yuan, L., 2008, Network Traffic Prediction Based on Improved BP Wavelet Neural Network, *International Conference on Wireless Communications, Networking and Mobile Computing*, 1–5.
- Rafsanjani, M.K., Varzaneh, Z.A., Chukanlo, N.E., 2012, A survey of hierarchical clustering algorithms, *The Journal of Mathematics and Computer Science* vol. 5 no.3, 229–240.
- Rahmat Widia, S., Jasni, M. Z., Embong, A., 2011, A Comparative Agglomerative Hierarchical Clustering Method to Cluster Implemented Course, *Journal of Computing*.
- Raschka, S., 2016, *Python Machine Learning*, Packt Publishing.
- Rosner, B., 1975, On the detection of many outliers, *Technometrics*, 17(2), 221–227.
- Rutka, G., 2008, Network traffic prediction using ARIMA and neural networks models, *Elektron Elektrotech*, 84, 47–52.
- Samulevicius, S., Pedersen, T. B., Sorensen, T. B., 2015, MOST: Mobile broadband network optimization using planned Spatio-temporal events, *Proceeding IEEE Vehicular Technology Conference*, 1–4.
- Sainath, T. N., Vinyals, O., Senior, A. and Sak, H., 2015, Convolutional, long short-term memory, fully connected deep neural networks, in *Proc. ICASSP*, 4580–4584.
- Sciancalepore, V., Samdanis, K., Costa-Perez, X., Bega, D., Gramaglia, M., Banchs, A., 2017, Mobile traffic forecasting for maximizing 5G network slicing resource utilization, *IEEE International Conference on Computer Communications*.
- Scott, A. J., Knott, M., 1974, A Cluster Analysis Method for Grouping Means in the Analysis of Variance. *Biometrics*, 30(3), 507–512.
- Seabold, S., Perktold, J., 2010, Statsmodels: Econometric and statistical modeling with python, *Python in Science Conference*.
- Sevgican, S., Turan, M., Gokarlan, K., Yilmaz, H., Tugcu, T., 2020, Intelligent network data analytics function in 5G cellular networks using machine learning, *Journal Of Communications And Networks*, 22(3), 269–280.
- Smith, G., 2010, *PostgreSQL 9.0: High Performance*, Packt Publishing Ltd,
- Stan Development Team, 2020, *Stan Modeling Language Users Guide and Reference Manual*, Erişim Tarihi: 20 Nisan 2021.
- Stanford, 2021, <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-recurrent-neural-networks>, Erişim Tarihi: 16 Mayıs 2021.

- Syed, A. R., Burney, S. M. A., Sami, B., 2010, Forecasting Network Traffic Load Using Wavelet Filters and Seasonal Autoregressive Moving Average Model, *International Journal of Computer and Electrical Engineering*, 2(6), 979–983.
- Taylor, S. J., Letham, B., 2018, Forecasting at scale. *The American Statistician*, 72(1), 37–45.
- Theodosiou, M., 2011, Disaggregation & aggregation of time series components: A hybrid forecasting approach using generalized regression neural networks and the theta method, *Neurocomputing*, 74(6), 896–905.
- Tikunov, D., Nishimura, T., 2007, Traffic prediction for mobile network using Holt-Winter's exponential smoothing, *SoftCOM, International Conference on Software, Telecommunications and Computer Networks*, 310–314.
- Trinh, H., Giupponi, L., Dini, P., 2018, Mobile Traffic Prediction from Raw Data Using LSTM Networks, *PIMRC IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 1827–1832.
- Visual Studio Code, 2021, <https://code.visualstudio.com/>, Erişim Tarihi: 15 Mayıs 2021.
- Wang, J., Tang, J., Xu, Z., Wang, Y., Xue, G., Zhang, X., Yang, D., 2017, Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach, *INFOCOM, IEEE Conference on Computer Communications*.
- Wang, S., Zhang, X., Zhang, J., Feng, J., Wang, W., Xin, K., 2015, An Approach for Spatial-Temporal Traffic Modeling in Mobile Cellular Networks, *ITC, International Teletraffic Congress*, 203–209.
- Wang, X., Zhou, Z., Xiao, F., Xing, K., Yang, Z., Liu, Y., Peng, C., 2018, Spatio-Temporal Analysis and Prediction of Cellular Traffic in Metropolis, *ICNP, IEEE International Conference on Network Protocols*.
- Winters, P. R., 1960, Forecasting sales by exponentially weighted moving averages, *Management Science*, vol. 6, 324–342.
- Xiang, L., Ge, X., Liu, C., Shu, L., Wang, C., 2010, A new hybrid network traffic prediction method, *GLOBECOM, IEEE Global Telecommunications Conference*.
- Xingjian, S., Zhou, C., Hao, W., Dit-Yan, Y., Wai-kin, W. and Wang-chun, W., 2015, Convolutional LSTM Network: a machine learning approach for precipitation nowcasting, In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1 (NIPS'15)*, 802–810.
- Xu, M., Wang, Q., Lin, Q., 2018, Hybrid holiday traffic predictions in cellular networks, *NOMS, IEEE/IFIP Network Operations and Management Symposium*, 1–6.

- Yamada, Y., Shinkuma, R., Sato, T., Oki, E., 2018, Feature-Selection Based Data Prioritization in Mobile Traffic Prediction Using Machine Learning, *GLOBECOM, IEEE Global Communications Conference*, 1–6.
- Yu, Y., Wang, J., Song, M., Song, J., 2010, Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient, *ISDEA, International Conference on Intelligent System Design and Engineering Application*, 1(1), 980–983.
- Yüksel, M., Akay, M., Çiftçi, S., 2019, Development of Internet Traffic Prediction Software Using Time-Series Multilayer Perceptron, *Gümüşhane University Journal of Science and Technology Institute*, 1–6.
- Zang, Y., Ni, F., Feng, Z., Cui, S., Ding, Z., 2015, Wavelet transform processing for cellular traffic prediction in machine learning networks, *ChinaSIP, IEEE China Summit and International Conference on Signal and Information Processing*, 458–462.
- Zhang, C., Patras, P., 2018, Long-term mobile traffic forecasting using deep Spatio-Temporal neural networks, *MobiHoc, International Symposium on Mobile Ad Hoc Networking and Computing*, 231–240.
- Zhang, C., Zhang, H., Qiao, J., Yuan, D., Zhang, M., 2019, Deep Transfer Learning for Intelligent Cellular Traffic Prediction Based on Cross-Domain Big Data, *IEEE Journal on Selected Areas in Communications*, 37(6), 1389–1401.
- Zhang, S., Zhao, S., Yuan, M., Zeng, J., Yao, J., Lyu, M., King, I., 2017, Traffic Prediction Based Power Saving in Cellular Networks, *SIGSPATIAL ACM International Conference on Advances in Geographic Information Systems*, 1–10.
- Zheng, X., Lai, W., Chen, H., Fang, S., Li, Z., 2020, A study of cellular traffic data prediction by kernel ELM with parameter optimization, *Applied Sciences*, 1–16.
- Zhou, B., He, D., Sun, Z., 2006, Traffic predictability based on ARIMA/GARCH model, *NGI, Conference on Next Generation Internet Design and Engineering*, 200–207.
- Zhou, X., Zhao, Z., Li, R., Zhou, Y., Zhang, H., 2012, The predictability of cellular networks traffic, *ISCIT, International Symposium on Communications and Information Technologies*, 973–978.