SATIŞ VE TALEP TAHMİNİ İÇİN DERİN TRANSFER ÖĞRENME METODOLOJİSİNİN GELİŞTİRİLMESİ

Begüm EROL

T.C.
BURSA ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

SATIŞ VE TALEP TAHMİNİ İÇİN DERİN TRANSFER ÖĞRENME METODOLOJİSİNİN GELİŞTİRİLMESİ

Begüm EROL
0000-0001-9131-3317

Prof. Dr. Tülin İNKAYA
(Danışman)

DOKTORA TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2024
Her Hakkı Saklıdır

ÖZET

Doktora Tezi

SATIŞ VE TALEP TAHMİNİ İÇİN DERİN TRANSFER ÖĞRENME METODOLOJİSİNİN GELİŞTİRİLMESİ

Begüm EROL

Bursa Uludağ Üniversitesi
Fen Bilimleri Enstitüsü
Endüstri Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Tülin İNKAYA

Satış ve talep tahmini, tedarik zinciri yönetiminde kritik bir role sahiptir. Hızlı, doğru ve etkin tahminler için gelişmiş tekniklere ihtiyaç duyulmaktadır. Gelişen teknoloji ile birlikte son yıllarda satış ve talep tahmininde derin öğrenme (Deep Learning – DL) yöntemlerinin kullanımı yaygınlaşmıştır. DL ile yapılan tahminlerde başarılı sonuçlar elde edilmektedir. Ancak DL eğitimi için hem büyük miktarda veriye ihtiyaç duyulmaktadır, hem de kullanılan veri miktarı arttıkça modelin eğitim süresi katlanarak artmaktadır. Buna çözüm olarak, kaynak alandaki veri ile ön eğitilmiş ağdan edinilen bilgilerin hedef alana aktarımını sağlayan transfer öğrenme kullanılmaktadır. Bu tez çalışmasında, satış ve talep tahmini için bilgi aktarımını sağlayan derin transfer öğrenme (Deep Transfer Learning – DTL) tabanlı metodolojiler geliştirilmiştir. Satış verileri tek ve çok değişkenli zaman serileri olarak ele alınmıştır. İlk olarak, satış ve talep tahmininde en uygun DL yöntemini seçmek amacıyla dokuz farklı DL yöntemi karşılaştırılmıştır. Parametrik olmayan istatistiksel testler yardımıyla LSTM yöntemi seçilmiştir. İkinci olarak, uzaklık tabanlı kaynak seçimi ile tek kaynaklı DTL modeli önerilmiştir. Sonrasında, transfer öğrenmede aktarım başarısını tahmin eden rastgele orman modeli geliştirilerek kaynak seçiminde açıklanabilirlik sağlanmıştır. Rastgele ormanda seçilen öznitelikler ile derin kümeleme tabanlı DTL modeli geliştirilmiştir ve bu modele topluluk öğrenme uygulanmıştır. Bu metodoloji kümeleme ve topluluk tabanlı DTL metodolojisi olarak adlandırılmıştır. Ayrıca, geliştirilen metodolojide çoklu kaynak seçimi için ağırlıklandırma mekanizması geliştirilmiştir. Son olarak, önerilen metodolojiler çok değişkenli satış verilerine uyarlanmıştır. Çeşitli veri setleri ile yapılan deneysel çalışmalarda, kümelemenin benzer satış verilerinin tespitini sağlayarak tahmin doğruluğunu arttırdığı görülmüştür. Ayrıca, çok kaynaklı DTL yaklaşımlarının tek kaynaklı DTL yaklaşımlarına göre tahmin sonuçlarını iyileştirdiği gözlenmiştir. Geliştirilen metodolojiler; enerji, finans, çevre ve tedarik zinciri gibi çeşitli üretim ve hizmet sektörlerinde uygulanabilir niteliktedir.

Anahtar Kelimeler: Yapay zeka, satış ve talep tahmini, derin öğrenme, transfer öğrenme, kaynak seçimi, derin kümeleme, topluluk öğrenme, açıklanabilirlik
2024, xiii + 208 sayfa.

ABSTRACT

PhD Thesis

DEVELOPMENT OF DEEP TRANSFER LEARNING METHODOLOGY FOR SALES AND DEMAND FORECASTING

Begüm EROL

Bursa Uludag University
Graduate School of Natural and Applied Sciences
Department of Industrial Engineer

Supervisor: Prof. Dr. Tülin İNKAYA

Sales and demand forecasting plays a critical role in supply chain management. Advanced techniques are needed for fast, accurate and effective predictions. With the developing technology, the use of deep learning (DL) methods in sales and demand forecasting has become widespread in recent years. Successful results are obtained in predictions made with DL. However, a large amount of data is needed for DL training, and as the amount of data used increases, the training time of the model increases exponentially. As a solution to this, transfer learning is used, which transfers the data in the source area and the information obtained from the pre-trained network to the target area. In this thesis study, deep transfer learning (DTL) based methodologies have been developed that provide information transfer for sales and demand forecasting. Sales data are treated as univariate and multivariate time series. First, nine different DL methods were compared in order to choose the most appropriate DL method for sales and demand forecasting. The LSTM method was chosen with the help of non-parametric statistical tests. Secondly, a single-source DTL model with distance-based source selection is proposed. Subsequently, a random forest model that predicts transfer success in transfer learning was developed to ensure explainability in source selection. A deep clustering-based DTL model was developed with features selected in random forest and ensemble learning was applied to this model. This methodology has been called clustering and ensemble-based DTL methodology. Additionally, a weighting mechanism for multiple source selection has been developed in the developed methodology. Finally, the proposed methodologies are adapted to multivariate sales data. In experimental studies conducted with various data sets, it has been shown that clustering increases forecast accuracy by enabling the detection of similar sales data. Additionally, it has been observed that multi-source DTL approaches improve prediction results compared to single-source DTL approaches. Developed methodologies; it is applicable in various production and service sectors such as energy, finance, environment and supply chain.

Key words: Artificial intelligence, sales and demand forecasting, deep learning, transfer learning, source selection, deep clustering, ensemble learning, explainability
2024, xiii + 208 pages.

ÖNSÖZ ve TEŞEKKÜR

Lisansüstü eğitim hayatım boyunca bilgi, birikim ve tecrübeleri ile yol gösteren, desteğini hiçbir zaman esirgemeyen, büyük bir özveri ile her konuda yol gösteren ve motive eden, kendisini örnek aldığım çok değerli danışman hocam sayın Prof. Dr. Tülin İNKAYA'ya; lisans hayatımdan bu yana bilgisine, tecrübesine ve eğitmen kimliğine büyük saygı duyduğum ve doktora tez izleme komitemde yer alarak değerli bilgi ve tecrübeleriyle tezime katkı sağlayan sayın hocam Prof. Dr. Erdal EMEL'e; doktora tez izleme komitemde yer alarak verdiği kıymetli fikirlerle çalışmamı zenginleştiren, değerli bilgileri ile tezime katkı sağlayan sayın hocam Prof. Dr. Gürkan ÖZTÜRK'e teşekkürlerimi ve saygılarımı sunarım. Tez savunma sınavımda yer alarak değerli görüşlerini sunan sayın hocalarım Prof. Dr. Seda ÖZMUTLU, Prof. Dr. Alev Taşkın ve Doç. Dr. Arzum ÖZGEN'e teşekkürlerimi ve saygılarımı sunarım.

Bugünlere gelmemde katkısı olan başta saygıdeğer rahmetli Prof. Dr. Cenk ÖZMUTLU hocam olmak üzere adını sayamadığım eğitim hayatıma dokunmuş tüm eğitmenlerime şükranlarımı sunarım.

Beni bugünlere getiren, her daim en büyük destekçilerim olan, varlıklarıyla hayatıma anlam katan kıymetli annem Sevgi EROL'a, sevgili babam Recai EROL'a ve biricik abim Uğur Can EROL'a teşekkürlerimi sunarım. En büyüğünden en küçüğüne adını sayamadığım tüm aile bireylerime varlıkları için teşekkürlerimi sunarım.

Her zaman yanımda olan ve desteklerini esirgemeyen tüm dostlarıma ve arkadaşlarıma sonsuz teşekkürlerimi sunarım.

Bu tez çalışması, Bursa Uludağ Üniversitesi Bilimsel Araştırmalar Projesi Birimi FDK-2021-518 numaralı Lisansüstü Tez Projesi kapsamında desteklenmiştir. Desteklerinden dolayı BAP'a teşekkür eder, saygılarımı sunarım.

Begüm EROL
27/03/2024

İÇİNDEKİLER

Sayfa

ÖZET i
ABSTRACT ii
ÖNSÖZ ve TEŞEKKÜR iii
SİMGELER ve KISALTMALAR DİZİNİ vii
ŞEKİLLER DİZİNİ xii
ÇİZELGELER DİZİNİ xiv
1. GİRİŞ 1
1.1. Tezin Amacı ve Kapsamı 5
1.2. Araştırma Soruları 6
1.3. Tezin Literatüre Katkısı 6
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI 8
2.1. Derin Öğrenme Yöntemleri 8
2.1.1. Derin sinir ağı 8
2.1.2. Derin otokodlayıcı ağı 8
2.1.3. Evrişimli sinir ağı 10
2.1.4. Tekrarlayan sinir ağı 12
2.1.5. Uzun kısa-süreli bellek ağı 13
2.1.6. Kapılı tekrarlayan birim ağı 15
2.1.7. Çift yönlü uzun kısa-süreli bellek ağı 17
2.1.8. CNN-LSTM hibrit ağı 17
2.1.9. Evrişimli uzun kısa-süreli bellek ağı 18
2.2. Transfer Öğrenme 18
2.2.1. Transfer öğrenme yöntemleri 21
2.2.2. Transfer öğrenmenin kullanım amaçları ve potansiyel faydaları 22
2.2.3. Negatif transfer öğrenme 23
2.2.4. Derin transfer öğrenme 24
2.2.5. Transfer öğrenmede kaynak seçimi 25
2.2.6. Transfer öğrenmede açıklanabilirlik 27
2.3. Kümeleme Yöntemleri 33
2.3.1. Derin kümeleme 34
2.3.2. Küme doğrulama yöntemleri 35
2.4. Topluluk Öğrenme Yöntemleri 38
2.4.1. Torbalama 38
2.4.2. Yığınlama 39
2.5. Kaynak Araştırması 40
2.5.1. Derin öğrenme yöntemleri ile satış ve talep tahmini 40
2.5.2. Transfer öğrenme ile satış ve talep tahmini 42
2.5.3. Kümeleme yöntemleri ile satış ve talep tahmini 46
2.5.4. Topluluk öğrenme yöntemleri ile satış ve talep tahmini 50
3. MATERYAL ve YÖNTEM 55
3.1. Zaman Serileri ve Veri Kümeleri 55
3.2. Benzerlik Ölçütleri 57
3.2.1. Tek değişkenli dinamik zaman bükmesi 57
3.2.2. Tek değişkenli gerçek cezalı düzenleme uzaklığı 58
3.2.3. Zaman serisi arasındaki uzaklıkların hesaplanması için bir örnek 59
3.2.3. Çok değişkenli dinamik zaman bükmesi 62

3.3. Derin Otokodlayıcı Tabanlı Derin Kümeleme 64
3.4. Performans Değerlendirme Yöntemleri 66
3.5. Tek Değişkenli Zaman Serileri için Önerilen Çözüm Yaklaşımları 67
3.5.1. Uzaklık tabanlı derin transfer öğrenme 67
3.5.2. Topluluk tabanlı derin transfer öğrenme 69
3.5.3. Kümeleme ve topluluk tabanlı derin transfer öğrenme 70
3.6. Çok Değişkenli Zaman Serileri için Önerilen Çözüm Yaklaşımları 80
3.6.1. Uzaklık tabanlı derin transfer öğrenme 81
3.6.2. Topluluk tabanlı derin transfer öğrenme 82
3.6.3. Kümeleme ve topluluk tabanlı derin transfer öğrenme 82
4. BULGULAR ve TARTIŞMA 84
4.1. Deneysel Koşullar ve Parametre Optimizasyonu 84
4.2. Derin Öğrenme Yöntemlerinin Karşılaştırılması 90
4.2.1. Deney sonuçları 92
4.2.2. Parametrik olmayan istatistiksel test sonuçları 93
4.2.3. Tartışma 101
4.3. Tek Değişkenli Zaman Serileri için Deneysel Çalışmalar 102
4.3.1. Uzaklık tabanlı derin transfer öğrenme yönteminin karşılaştırılması 102
4.3.2. Önerilen çözüm yaklaşımlarının karşılaştırılması 111
4.3.3. Yeni veri setleri ile önerilen çözüm yaklaşımlarının performanslarının değerlendirilmesi 125
4.4. Çok Değişkenli Zaman Serileri için Deneysel Çalışmalar 139
4.5. Tek ve Çok Değişkenli Satış Verilerinin Deneysel Sonuçlarının Karşılaştırılması 151
4.6. Tartışma 155
5. SONUÇ 157
KAYNAKLAR 159
EKLER 170
EK 1 Temel DL Yöntemleri Kullanarak Satış ve Talep Tahmini Yapan Çalışmalar 171
EK 2 DL Yöntemlerde Teknik Bir İyileştirme veya Geliştirme Sağlayarak Satış ve Talep Tahmini Yapan Çalışmalar 172
EK 3 Birleşik DL Yöntemleri veya Topluluk Tabanlı DL Yöntemleri Kullanarak Satış ve Talep Tahmini Yapan Çalışmalar 174
EK 4 Zaman Serisi Öznitelikleri ve Açıklamaları 176
EK 5 Öznitelikler ve Açıklamaları 186
EK 6 Rastgele Orman için Deneysel Sonuçlar 187
EK 7 Seçilen Öznitelikler ve Önem Dereceleri ile Oluşturulmuş Ağırlıklı Öklid Uzaklıkları Matrisi 188
EK 8 Tek Değişkenli Satış Zaman Serileri Arasındaki DTW Uzaklıkları 189
EK 9 Tek Değişkenli Satış Zaman Serileri Arasındaki ERP Uzaklıkları 179
EK 10 Normalize Edilmiş DTW Uzaklıkları 179
EK 11 Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş DTW Uzaklıkları ($wDTW = 0{,}26$) 179
EK 12 Normalize Edilmiş ERP Uzaklıkları 179
EK 13 Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş ERP Uzaklıkları ($wERP = 0{,}23$) 179
EK 14 Tek Değişkenli Zaman Serileri için Tek Kaynaklı TL'deki RMSE Yüzde İyileşme Oranları 179

EK 15 Pharma-Seri_18 Hedef Verisi için Learnx-Seri_4 Kaynak Veri Ağırlıkları 179
EK 16 Pharma-Seri_18 Hedef Verisi için Learnx-Seri_5 Kaynak Veri Ağırlıkları 179
EK 17 Pharma-Seri_18 Hedef Verisi için Pharma-Seri_17 Kaynak Veri Ağırlıkları ..179
EK 18 Pharma-Seri_18 Hedef Verisi için Yöntemlerin Performanslarının Karşılaştırılması 199
EK 19 Tüm Satış Verileri için Seçilen Özniteliklerle Önem Dereceleri ile Oluşturulmuş Ağırlıklı Öklid Uzaklıkları Matrisi 200
EK 20 Tüm Satış Verileri için Normalize Edilmiş DTW Uzaklıkları 201
EK 21 Tüm Satış Verileri için Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş DTW Uzaklıkları ($wDTW = 0{,}26$).....202
EK 22 Tüm Satış Verileri için Normalize Edilmiş ERP Uzaklıkları 203
EK 23 Tüm Satış Verileri için Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş ERP Uzaklıkları ($wERP = 0{,}23$).....204
EK 24 Çok Değişkenli Satış Zaman Serileri Arasındaki DTW_D Uzaklıkları 205
EK 25 Çok Değişkenli Zaman Serileri için Tek Kaynaklı TL'deki RMSE Yüzde İyileşme Oranları 206
ÖZGEÇMİŞ 207

SİMGELER ve KISALTMALAR DİZİNİ

Simgeler	Açıklama
x_k^l	CNN ağının *l*. evrişim katmanında *k*. nörondaki girdi vektörü
b_k^l	CNN ağının *l*. evrişim katmanında *k*. nörondaki sapma vektörü
w_{ik}^{l-1}	CNN ağının (*l*-1). evrişim katmanındaki *i*. nörondan *l*. evrişim katmanındaki *k*. nörona kadar evrişim çekirdeği
s_i^{l-1}	CNN ağının (*l*-1). evrişim katmanında *i*. nörondaki çıktı vektörü
N_{l-1}	CNN ağının (*l*-1). evrişim katmanındaki nöron sayısı
$conv1D$	Evrişim operatörü
W_{xi}	RNN ağı girdisi için ağırlık matrisi
W_{hi}	RNN ağı gizli durumu için ağırlık matrisi
t	Zaman adımı
x_t	RNN, LSTM ve GRU ağının *t* zaman adımındaki girdi vektörü
y_t	RNN, LSTM ve GRU ağının *t* zaman adımındaki çıktı vektörü
h_t	RNN, LSTM ve GRU ağının *t* zaman adımındaki gizli durum vektörü
i	LSTM ağının girdi/güncelleme kapısı
f	LSTM ağının unutma kapısı
c	LSTM ağının hafıza hücresi
o	LSTM ağının çıktı kapısı
W_f	LSTM ağının unutma kapısı için ağırlık matrisi
W_i	LSTM ağının girdi/güncelleme kapısı için ağırlık matrisi
W_c	LSTM ağının hafıza hücresi için ağırlık matrisi
W_o	RNN ağı çıktısı için ağırlık matrisi LSTM ağının çıktı kapısı için ağırlık matrisi
b_f	LSTM ağının unutma kapısı için sapma vektörü
b_i	RNN ağı girdisi için sapma vektörü LSTM ağının girdi/güncelleme kapısı için sapma vektörü
b_c	LSTM ağının hafıza hücresi için sapma vektörü
b_o	RNN ağı çıktısı için sapma vektörü LSTM ağının çıktı kapısı için sapma vektörü
f_t	LSTM ağının *t* zaman adımında unutma kapısı aday değeri
i_t	LSTM ağının *t* zaman adımında girdi kapısı aday değeri
$\tilde{c}_t$	LSTM ağının *t* zaman adımında hafıza hücresinin güncellenmiş değeri
c_t	LSTM ağının *t* zaman adımında hafıza hücresi durumu
o_t	LSTM ağının *t* zaman adımında çıktı kapısı aday değeri
z_t	GRU ağının *t* zaman adımında güncelleme kapısı aday değeri
r_t	GRU ağının *t* zaman adımında sıfırlama kapısı aday değeri
$\tilde{h}_t$	GRU ağının *t* zaman adımında geçici gizli durum değeri
W_z	GRU ağının güncelleme kapısı için ağırlık matrisi
W_r	GRU ağının sıfırlama kapısı için ağırlık matrisi
W_h	GRU ağının gizli durumu için ağırlık matrisi
U_z	GRU ağının güncelleme kapısı için bir önceki zaman adımından gelen ağırlık matrisi
U_r	GRU ağının sıfırlama kapısı için bir önceki zaman adımından gelen ağırlık matrisi

Simgeler	**Açıklama**
U_h	GRU ağının gizli durumu için bir önceki zaman adımından gelen ağırlık matrisi
b_z	GRU ağının güncelleme kapısı için sapma vektörü
b_r	GRU ağının sıfırlama kapısı için sapma vektörü
b_h	GRU ağının gizli durum için sapma vektörü
$\times$	Noktasal çarpım
$\otimes$	Eleman bzaında çarpım
σ	Sigmoid fonksiyonu
$tanh$	Hiperbolik tanjant fonksiyonu
$p(j\|t)$	Karar ağacının *t* düğümündeki *j* sınıfının göreceli frekansı
a_i	*i.* veri noktası ile aynı kümede yer alan veri noktaları arasındaki ortalama mesafe
b_i	*i.* veri noktası ile en yakın başka bir kümede yer alan veri noktaları arasındaki ortalama mesafe
S_i	*i.* veri noktası (gözlem) için Silhouette katsayısı
$\bar{S}$	Bütün veri noktaları için Silhouette katsayısının ortalaması
k	Küme sayısı
$B(k)$	Küme merkezlerinin birbirlerine olan uzaklıklarının toplamı
$W(k)$	Küme içindeki veri noktalarının küme merkezlerine olan uzaklıklarının toplamı
C_i	*i.* küme
n_i	*i.* kümedeki veri noktalarının sayısı
c_i	*i.* kümenin merkezi
x	Veri noktası
c	Bütün veri noktalarının genel merkezi
ch	Calinski-Harabasz indeksi
$d(C_i, C_j)$	C_i ile C_j kümeleri arasındaki uzaklık
$diameter(C_l)$	C_l kümesinin çapı
D	Dunn indeksi
Y_i	Tek değişkenli veya çok değişkenli bir zaman serisi
n	Bir zaman serisindeki gözlem sayısı
T	Gözlem zamanı (Zaman serisinin uzunluğu)
F	Öznitelikler kümesi
y_{it}	*T* gözlem zamanı süresince *i.* zaman serisinin *t* zaman adımındaki gözlem değeri (satış miktarı)
y_{itf}	*T* gözlem zamanı süresince *i.* zaman serisinin *t* zaman adımındaki *f.* değişkenin (özniteliğin) gözlem değeri
R	Tek değişkenli bir zaman serisi
S	Tek değişkenli bir zaman serisi
r_i	*R* zaman serisinin *i.* elemanı
s_j	*S* zaman serisinin *j.* elemanı
$dist_{dtw}(r_i, s_j)$	r_i ve s_j arasındaki DTW uzaklığı
$dist_{erp}(r_i, s_j)$	r_i ve s_j arasındaki ERP uzaklığı
$DTW(R, S)$	*R* ve *S* zaman serileri arasındaki DTW uzaklığı
$ERP(R, S)$	*R* ve *S* zaman serileri arasındaki ERP uzaklığı
g	Sabit bir sayı

m	Değişken (öznitelik, boyut) sayısı
Q	Çok değişkenli (*M* boyutlu) bir zaman serisi
C	Çok değişkenli (*M* boyutlu) bir zaman serisi
$DTW(Q_m, C_m)$	*Q* zaman serisinin *m.* boyutu ile *C* zaman serisinin *m.* boyutu arasındaki DTW uzaklığı
$DTW_I(Q, C)$	*Q* ile *C* zaman serileri arasındaki boyuttan bağımsız olarak ölçülen DTW uzaklığı
$DTW_D(Q, C)$	*Q* ile *C* zaman serileri arasındaki boyut bağımlı olarak ölçülen DTW uzaklığı
x_i	DAE ağının *i.* girdi değeri
z_i	DAE ağının *i.* girdisinin gizli alan (latent space) değeri
$\varphi_e(x_i)$	DAE ağının kodlayıcı ağı
$\varphi_r(z_i)$	DAE ağının kod çözücü ağı
L_{rec}	DAE ağının yeniden yapılandırma kaybı
$\hat{y}_{it}$	*i.* zaman serisinin *t* zaman adımındaki tahmini satış değeri
X_K	Kaynak veri kümesi
y_h	Hedef veri
D_k	Kaynak veri kümesi
D_h	Hedef veri kümesi
R^2	Determinasyon katsayısı
$d_{WE}(A, B)$	*n* adet özniteliğe sahip (*n* boyutlu) *A* ve *B* verileri arasındaki ağırlıklı Öklid uzaklığı
A_i	*A* verisindeki *i.* özniteliğin değeri
B_i	*B* verisindeki *i.* özniteliğin değeri
$d(A, B)$	A ile B verileri arasındaki toplam uzaklık
p	Kaynak veri (satış serisi) sayısı
N	Kaynak modeli eğitme sayısı (topluluk modeldeki tekrar sayısı)
e_{ij}	*i.* kaynak veriye ait *j.* modelin eğitim hatası
$oran_{ij}$	*i.* kaynak veriye ait *j.* modelin maksimum eğitim hatasına göre topluluktaki katkı oranı
$ağırlık_{ij}$	*i.* kaynak veriye ait *j.* modelin topluluktaki önemi (ağırlığı)
t_{ij}	*i.* modelin *j.* topluluk yöntemi için toplam eğitim süresi
t_k	*k.* kaynak verisi ile eğitilen LSTM modelinin eğitim süresi
t_{hk}	*h.* hedef verisi için *k.* kaynak verisi ile ön eğitilmiş LSTM-TL modelinin eğitim süresi
t_j	*j.* topluluk yönteminin eğitim süresi

Kısaltmalar	**Açıklama**
AE	Otomatik Kodlayıcı (Autoencoder)
ANN	Yapay Sinir Ağı (Artificial Neural Network)
ARIMA	Otoregresif Bütünleşik Hareketli Ortalama (Autoregressive Integrated Moving Average)
ARIMAX	Dışsal Değişkenlerle Otoregresif Bütünleşik Hareketli Ortalama (Autoregressive Integrated Moving Average with Exogenous Variables)
AS	Araştırma Soruları

Kısaltmalar	Açıklama
Bi-LSTM	Çift Yönlü Uzun Kısa Süreli Bellek (Bidirectional Long Short Term Memory)
CART	Sınıflandırma ve Regresyon Ağaçları (Classification and Regression Trees)
CNN	Evrişimli Sinir Ağı (Convolutional Neural Network)
ConvLSTM	Evrişimli Uzun Kısa Süreli Bellek (Convolutional Long Short Term Memory)
ÇAT	Çok Adımlı Tahmin
DAE	Derin Otomatik Kodlayıcı (Deep Autoencoder)
DBSCAN	Gürültülü Uygulamaları Yoğunluğa Dayalı Konumsal Kümeleme (Density Based Spatial Clustering of Applications with Noise)
DeepAR	Derin Otoregresif (Deep AutoRegressive)
DeepTCN	Derin Zamansal Evrişimli Sinir Ağı (Deep Temporal Convolutional Neural Network)
DL	Derin Öğrenme (Deep Learning)
DNN	Derin Sinir Ağı (Deep Neural Network)
DT	Karar Ağacı (Decision Tree)
DTL	Derin Transfer Öğrenme (Deep Transfer Learning)
DTW	Dinamik Zaman Bükme (Dynamic Time Warping)
EDR	Gerçek Diziler için Düzenleme Uzaklığı (Edit Distance for Real Sequences)
ERP	Gerçek Cezalı Düzenleme Uzaklığı (Edit Distance with Real Penalty)
ELM	Aşırı Öğrenme Makinesi (Extreme Learning Machine)
ETS	Hata, Eğilim, Mevsimsellik (Error, Trend, Seasonality)
GAN	Üretken Çekişmeli Ağ (Generative Adversarial Network)
GBRT	Gradyan Güçlendirme Regresyon Ağaçları (Gradient Boosted Regression Trees)
GNN	Üretken Çekişmeli Ağ (Generative Adversarial Network)
GRU	Kapılı Tekrarlayan Birim (Gated Recurrent Unit)
KNN	K-En Yakın Komşuluk (K-Nearest Neighbor)
KNN-R	K-En Yakın Komşuluk Regresyonu (K-Nearest Neighbor Regression)
LCSS	En Uzun Ortak Dizi (Longest Common Subsequence)
LightGBM	Hafif Gradyan Güçlendirme Makinesi (Light Gradient Boosting Machine)
LOOCV	Birini Dışarıda Bırak Çapraz Doğrulama (Leave-One-Out Cross Validation)
LR	Lineer Regresyon (Linear Regression)
LSTM	Uzun Kısa Süreli Bellek (Long Short Term Memory)
MAE	Ortalama Mutlak Hata (Mean Absolute Error)
MAPE	Ortalama Mutlak Yüzde Hata (Mean Absolute Percentage Error)
ML	Makine Öğrenmesi (Machine Learning)
MLP	Çok Katmanlı Algılayıcı (Multi-Layer Perceptron)
MS	Model Sayısı
MSE	Ortalama Karesel Hata (Mean Square Error)
NN	Sinir Ağı (Neural Network)
OSD	Ortalama Sıralama Değeri

Kısaltmalar	Açıklama
PCA	Temel Bileşen Analizi (Principal Component Analysis)
PAM	Medoidlerin Etrafında Bölümleme (Partitioning Around Medoids)
RF	Rastgele Orman (Random Forest)
RT	Regresyon Ağacı (Regression Tree)
RMSE	Ortalama Karesel Hatanın Karekökü (Root Mean Square Error)
RNN	Tekrarlayan Sinir Ağı (Recurrent Neural Network)
SARIMA	Mevsimsel Bütünleşik Otoregresif Hareketli Ortalama (Seasonal Autoregressive Integrated Moving Average)
SES	Basit Üstel Düzeltme (Simple Exponential Smoothing)
SMA	Basit Hareketli Ortalama (Simple Moving Average)
SVM	Destek Vektör Makinesi (Support Vector Machine)
SVR	Destek Vektör Regresyonu (Support Vector Regression)
SSE	Hata Kareleri Toplamı (Sum of Squared Errors)
SST	Toplam Kareler Toplamı (Sum of Squared Total)
TADA	Çift Dikkatli Çok Görevli RNN ile Trend Hizalama (Trend Alignment with Dual-Attention Multi-Task RNN)
TAT	Tek Adımlı Tahmin
TL	Transfer Öğrenme (Transfer Learning)
TWED	Zaman Çarpıtma Düzenleme Uzaklığı (Time Warp Edit Distances)
TZ	Tedarik Zinciri
TZY	Tedarik Zinciri Yönetimi
VAE	Değişken Otomatik Kodlayıcı (Variational Autoencoder)
XGBoost	Aşırı Gradyan Güçlendirme (Extreme Gradient Boosting)
1D-CNN	Bir Boyutlu CNN (One Dimensional Convolutional Neural Network)

ŞEKİLLER DİZİNİ

		Sayfa
Şekil 1.1.	Geleneksel ML ile TL arasındaki farklar	4
Şekil 2.1.	Sinir ağı mimarileri	9
Şekil 2.2.	Derin otokodlayıcı mimarisi	10
Şekil 2.3.	1D-CNN sinir ağı mimarisi	11
Şekil 2.4.	Tekrarlayan sinir ağı mimarisi	13
Şekil 2.5.	LSTM sinir ağı mimarisi	14
Şekil 2.6.	GRU sinir ağı mimarisi	16
Şekil 2.7.	Bi-LSTM sinir ağı mimarisi	17
Şekil 2.8.	CNN-LSTM sinir ağı mimarisi	17
Şekil 2.9.	ConvLSTM sinir ağı mimarisi	19
Şekil 2.10.	Transfer öğrenme yöntemlerinin sınıflandırılması	21
Şekil 2.11.	TL performans özeti	23
Şekil 2.12.	Negatif transfer öğrenme	24
Şekil 2.13.	Karar ağacı yapısı	30
Şekil 2.14.	Rastgele orman yapısı	32
Şekil 2.15.	Topluluk öğrenme yöntemleri	39
Şekil 2.16.	Yıl bazında yayın sayıları	42
Şekil 2.17.	Derin öğrenme yöntemlerine göre yayın sayılarının dağılımı	42
Şekil 3.1.	R ve S zaman serilerinin grafikleri	60
Şekil 3.2.	DTW maliyet matrisi ve uzaklık hesaplamaları	60
Şekil 3.3.	DTW yörüngesi ile R ve S serilerinin eşleşme grafiği	60
Şekil 3.4.	ERP maliyet matrisi ve uzaklık hesaplamaları	61
Şekil 3.5.	ERP yörüngesi ile R ve S serilerinin eşleşme grafiği	61
Şekil 3.6.	İki adet çok boyutlu zaman serisi arasındaki çok değişkenli DTW uzaklıkları	64
Şekil 3.7.	Sıralı çok adımlı derin kümeleme	64
Şekil 3.8.	Derin otokodlayıcı yapısı	65
Şekil 3.9.	k-ortalamalar içeren DAE tabanlı derin kümeleme	66
Şekil 3.10.	ERP'ye dayalı LSTM tabanlı derin transfer öğrenme yaklaşımının akış şeması	68
Şekil 3.11.	ERP-LSTM-TL algoritması	69
Şekil 3.12.	Torbalama tabanlı tek kaynaklı derin transfer öğrenme algoritmasının sözde kodu	70
Şekil 3.13.	Yığınlama tabanlı tek kaynaklı derin transfer öğrenme algoritmasının sözde kodu	70
Şekil 3.14.	Tek değişkenli zaman serileri için kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımının akış şeması	71
Şekil 3.15.	Öznitelik çıkarımı	73
Şekil 3.16.	Öznitelik seçimi süreci	75
Şekil 3.17.	Benzemezlik matrisinin oluşturulması ve zaman serilerinin kümelenmesi	76
Şekil 3.18.	Torbalama tabanlı çok kaynaklı derin transfer öğrenme algoritmasının sözde kodu	77
Şekil 3.19.	Yığınlama tabanlı çok kaynaklı derin transfer öğrenme algoritmasının sözde kodu	78
Şekil 3.20.	Kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin	78

	oluşturulması	
Şekil 3.21.	Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin oluşturulması	80
Şekil 3.22.	M-DTW$_D$-LSTM-TL algoritması	82
Şekil 3.23.	Çok değişkenli zaman serileri için kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımının akış şeması	82
Şekil 3.24.	Benzemezlik matrisinin oluşturulması	83
Şekil 3.25.	Zaman serilerinin kümelenmesi	83
Şekil 4.1.	DL yöntemlerinin performanslarının karşılaştırılması için akış şeması	91
Şekil 4.2.	Pharma veri kümesine ait grafikler	91
Şekil 4.3.	Pharma test verisinde en başarılı ilk üç derin öğrenme yönteminin tahmin performanslarının karşılaştırılması	95
Şekil 4.4.	Nemenyi testi sonuç diyagramları	98
Şekil 4.5.	Isı haritaları	106
Şekil 4.6.	LearnX-Seri_4 hedef verisi için sonuç grafikleri	106
Şekil 4.7.	Nemenyi testi sonuç diyagramları	108
Şekil 4.8.	Transfer öğrenme aktarım başarısının tahmininde kullanılan veri kümesi	113
Şekil 4.9.	Pharma-Seri_18 verisi için topluluk yöntemlerin RMSE değerleri	120
Şekil 4.10.	Pharma-Seri_18 verisi için topluluk yöntemlerin MAE değerleri	120
Şekil 4.11.	Pharma-Seri_18 verisi için topluluk yöntemlerin eğitim süreleri	121

ÇİZELGELER DİZİNİ

		Sayfa
Çizelge 3.1.	Veri kümeleri ve özellikleri	56
Çizelge 3.2.	Öznitelikler ve açıklamaları	56
Çizelge 4.1.	Derin öğrenme modellerindeki hiperparametreler	86
Çizelge 4.2.	Karşılaştırmalı değerlendirme yöntemlerinin parametreleri	87
Çizelge 4.3.	Rastgele orman hiperparametreleri	88
Çizelge 4.4.	DAE tabanlı derin kümeleme hiperparametreleri	88
Çizelge 4.5.	Derin öğrenme yöntemlerinin test verilerindeki performansları	94
Çizelge 4.6.	Friedman testleri sonuçları	96
Çizelge 4.7.	Wilcoxon testlerine ait p-değerleri	99
Çizelge 4.8.	Kazanç, kayıp ve eşitlik sayıları	100
Çizelge 4.9.	Veri kümeleri ve kullanılan seriler	103
Çizelge 4.10.	Tahmin yöntemlerinin karşılaştırılması	107
Çizelge 4.11.	Friedman test sonuçları	108
Çizelge 4.12.	Wilcoxon eşleştirilmiş işaretli sıra testi testlerinin p-değerleri	110
Çizelge 4.13.	Rastgele orman modellerinde seçilen öznitelikler ve önem dereceleri	114
Çizelge 4.14.	Seçilen öznitelikler ve değerleri	114
Çizelge 4.15.	Benzemezlik matrisi	115
Çizelge 4.16.	Derin kümeleme sonucu küme atamaları	116
Çizelge 4.17.	Temsil alan boyutu 3, küme sayısı 3 ve 4 iken oluşan küme atamaları	116
Çizelge 4.18.	Serilerin benzemezlik matrisine göre en yakın komşuları	117
Çizelge 4.19.	Favorita-Seri_26 hedef verisi için yöntemlerin performanslarının karşılaştırılması	124
Çizelge 4.20.	Benzemezlik matrisi	125
Çizelge 4.21.	Derin kümeleme sonucu küme atamaları	127
Çizelge 4.22.	Temsil alan boyutu 3, küme sayısı 3 ve 4 iken oluşan küme atamaları	127
Çizelge 4.23.	Serilerin benzemezlik matrisine göre en yakın komşuları	128
Çizelge 4.24.	Tek değişkenli Walmart-Seri_3 hedef verisi için yöntemlerin performanslarının karşılaştırılması	132
Çizelge 4.25.	Tek değişkenli LearnX-Seri_6 hedef verisi için yöntemlerin performanslarının karşılaştırılması	135
Çizelge 4.26.	Tek değişkenli Supermarket-Seri_27 hedef verisi için yöntemlerin performanslarının karşılaştırılması	138
Çizelge 4.27.	Benzemezlik matrisi (Normalize edilmiş DTW_D uzaklıkları)	139
Çizelge 4.28.	Derin kümeleme sonucu küme atamaları	140
Çizelge 4.29.	Temsil alan boyutu 3, küme sayısı 3 ve 4 iken oluşan küme atamaları	141
Çizelge 4.30.	Serilerin benzemezlik matrisine göre en yakın komşuları	142
Çizelge 4.31.	Çok değişkenli Walmart-Seri_3 hedef verisi için yöntemlerin performanslarının karşılaştırılması	145
Çizelge 4.32.	Çok değişkenli LearnX-Seri_6 hedef verisi için yöntemlerin performanslarının karşılaştırılması	147
Çizelge 4.33.	Çok değişkenli Supermarket-Seri_27 hedef verisi için yöntemlerin performanslarının karşılaştırılması	149

Çizelge 4.34. Tek ve çok değişkenli zaman serilerinin deneysel sonuçlarının karşılaştırması 152

1. GİRİŞ

Tedarik Zinciri (TZ); tedarikçiler, üreticiler, dağıtıcılar ve perakendecilerin oluşturduğu bir ağdır. Tedarik Zinciri Yönetimi (TZY) ise bir malın hammadde temininden, üretimi ve dağıtımına kadar son müşteriye kadar ulaşabilmesi için bu süreçte yer alan tedarikçi, üretici, dağıtıcı, perakendeci ve müşteriler arasındaki malzeme, ürün, para ve bilginin yönetimidir (Chopra vd., 2013). Küreselleşmenin ve rekabet ortamının artması ile birlikte TZY'nin önemi ve dolayısıyla karmaşıklığı da artmaktadır.

TZY'deki önemli görevlerden biri talep tahminidir. TZ ağındaki üretici, perakendeci ve distribütör firmalar talep tahminleri yaparlar (Chopra vd., 2013). Siparişe göre üretim yapan bir fabrika, deposunda bulundurması gereken malzeme miktarlarını tespit etmek için, stoğa göre üretim yapan bir fabrika ise kendi üretim ve stok seviyelerini belirlemek için tahminlere ihtiyaç duyar. Doğru talep tahmini firmanın kârını ve verimliliğini artırırken eş zamanlı olarak stoklama ve işletim maliyetlerini azaltmaktadır. Tedarik ile talebin dengelenmemesi durumunda ise firmalar yok satmaya bağlı gelir kaybına veya fazla üretim sonucu artan stok maliyetlerine katlanmaktadır. Bu durumların neticesinde firmalar müşteri kaybı veya düşen müşteri memnuniyetiyle karşılaşmaktadır. Talep tahmini, tüketici davranışlarının belirsizliği ve son dönemde tüketicilerin kendileri için özelleştirilmiş ürünlere artan talepleri nedeniyle çözümü zor bir problemdir.

Yukarıda bahsedilen problemlerin yaşanmaması için tedarik ile talebin dengelenmesi yani talebin doğru tahmin edilmesi gerekmektedir. Üretimin kısıtlı olduğu dönemlerde tüketici bulabildiği ürünü satın aldığı ya da ürünü beklemeyi kabul ettiğinden dolayı tedarik ile talebin dengelenmesine önem verilmemiştir (Chopra vd., 2013). Rekabet artıp piyasaya arz edilen ürün miktarı ve çeşitliliği artınca talep tahmini için veriye dayalı nitel (anket, uzman görüşü, vb.) ve nicel (istatistiksel yöntemler, zaman serisi analizleri, regresyon, sezgisel algoritmalar, yapay sinir ağları, vb.) birçok yöntem geliştirilmiştir. Ancak nitel tahmin yöntemleri ve nicel yöntemlerdeki sapmalar firmalara ciddi maliyetlere neden olmaktadır (Ingle vd., 2021). Yapay sinir ağlarının ortaya çıkması ve yüksek tahmin doğruluğu sağlaması nedeniyle talep tahmininde kullanımı artmaktadır (Kochak ve Sharma, 2015).

1950'lerde ortaya çıkan yapay zekâ, makinaların çeşitli faaliyetleri insanlar kadar kabiliyetli bir şekilde yapabilmesini sağlamaktadır. Yapay zeka, veri üzerinde algoritmik hesaplarda bulunarak programlananı iyileştirebilen, verilen hedeflere farklı yollarla varabilen ve hatalardan öğrenebilen sistemlerdir (Ongsulee, 2017). 1980'lerde ortaya çıkan makine öğrenmesi (Machine Learning – ML), veriler ve parametreler ile benzetimler yaparak, insanlardan daha iyi tespitlerde bulunan, programlanmayanı da açığa çıkarabilen ve kendi kendini eğitebilen sistemlerdir (Ongsulee, 2017). 2010'lu yıllarda ise büyük veri ile tek bir katmanda değil, birçok katmanda makine öğrenmesinde kullanılan hesapları tek bir seferde yapan ve tanımlanması gereken parametreleri kendisi keşfeden bir sistem olan derin öğrenme (Deep Learning – DL) kavramı ortaya çıkmıştır (Bengio, 2009; Ongsulee, 2017). Derin öğrenme, makine öğrenmesinin; makine öğrenmesi ise yapay zekânın alt dalı olarak özetlenebilir (Ongsulee, 2017).

DL, büyük ve zorlu veri kümelerinde karmaşık yapıların öğrenildiği bir ML sınıfıdır (Bengio, 2009). DL verinin birden fazla anlam kazandıran özellik seviyesinden veya temsilinden öğrenmeye dayalıdır (Bengio, 2009). Üst düzey özellikler, alt düzey özelliklerden türetilerek hiyerarşik bir temsil oluşturulur. Elle çıkarılan özellikler yerine veriyi en iyi temsil eden bu hiyerarşik özellik çıkarımı için etkin algoritmalar kullanılmaktadır (Bengio, 2009). DL gözetimli (sınıflandırma), yarı gözetimli (genetik sıralama, ağ sayfası sınıflandırma) veya gözetimsiz (kümeleme, örüntü analizi) olarak gerçekleştirilebilir. DL modelleri farklı düzeylerdeki verilerden aşamalı olarak özellikleri öğrenebildiği için özellikle yapılandırılmamış veriler üzerinde mevcut ML tekniklerinden daha iyi performans göstermektedir (Goodfellow vd., 2016; Dharshini ve Vijila, 2021). Derin öğrenmede makine öğrenmesine kıyasla eğitim için veri ihtiyacı daha fazladır ve buna bağlı olarak daha uzun eğitim sürelerine sahiptir (Dharshini ve Vijila, 2021; Ingle vd., 2021). Eğitim süresinin uzun olması ML yöntemlerine kıyasla hesaplama gücüne olan ihtiyacı da arttırmaktadır. Ek olarak derin öğrenme modelleri kara kutu modeller olarak adlandırılmaktadır, başka bir ifadeyle modellerin karmaşıklığı nedeniyle açıklanabilirliği makine öğrenmesi modellerine göre daha zordur (Ingle vd., 2021).

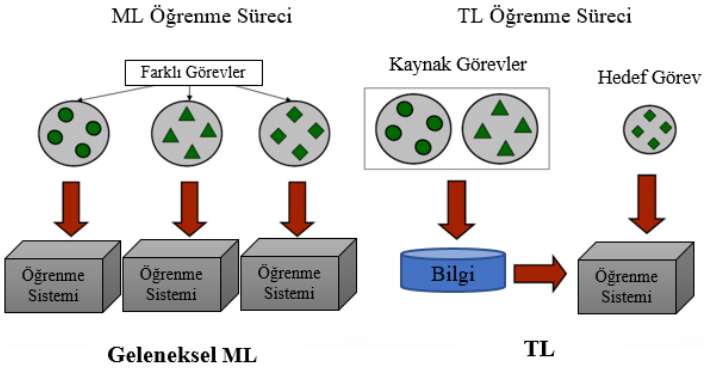
Günümüzde veri üretimi ve depolamadaki gelişmeler, verileri analiz edebilecek güçlü donanımların varlığı sayesinde derin öğrenmenin popülaritesi her geçen gün artmaktadır

(Dharshini ve Vijila, 2021; Acı ve Doğansoy, 2022; Eglite ve Birzniece, 2022). DL; yüz veya ses tanıma sistemlerinde, siber tehdit analizlerinde, güvenlik ve sağlık gibi sistemlerin anomali tespitinde, tavsiye sistemlerinde (müzik, film, vb.), sürücüsüz araç tasarımlarında, tedarik zinciri yönetiminde satış, talep ve müşteri kaybı tahmini gibi birçok alanda kullanılmaktadır (Şeker vd., 2017). Büyük veri ile çalışabildiği ve yüksek tahmin doğrulukları elde edebildiği için tahmin problemlerinin çözümünde DL yöntemleri sıklıkla tercih edilmektedir (Dharshini ve Vijila, 2021; Ingle vd., 2021; Eglite ve Birzniece, 2022).

Derin öğrenme yöntemlerinin kullanımı bazı zorlukları da beraberinde getirmektedir. Çok katmanlı yapıları, modellerin karmaşıklığı ve büyük veri gereksinimleri nedeniyle uzun eğitim süreleri gerektirmektedir. Yeni veri gelişiyle tekrarlayan uzun eğitim sürelerine katlanmak yerine transfer öğrenme kavramı ortaya çıkmıştır.

Öğrenme aktarımı veya transfer öğrenme (Transfer Learning – TL), bir problemi çözerken elde edilen bilgiyi saklayan ve daha sonra farklı ama ilgili bir probleme uygulamayı sağlayan makine öğrenmesinin bir alt alanıdır. Normalde sinir ağlarında her bir görev için ayrı bir öğrenme süreci gerçekleştirilir. Ancak bazı görevlerden öğrenilen bilgileri başka görevlerde kullanmak mümkündür ve avantajlı olmaktadır. Bu nedenle, kaynak görevlerden elde edilen bilgi, hedef görevin çözümü için kullanılmaktadır. TL ile önceki bilgiler kullanılarak daha az eğitim verisi ile daha yüksek başarı gösteren ve daha hızlı öğrenen modeller elde edilebilir. TL ile ML arasındaki farklar Şekil 1.1'de gösterilmektedir. DL modellerini büyük ve zorlu veri kümeleri ile eğitmek için gerekli kaynaklar göz önünde bulundurulduğunda, TL derin öğrenmede önem kazanmaktadır (Pan ve Yang, 2009).

Transfer öğrenmede, önce kaynak veri kümesi ve kaynak görevde temel bir ağ hazırlanır. Daha sonra, öğrenilen özellikler (ağırlıklar, parametreler, vb.) olduğu gibi veya yeniden tasarlanarak bir hedef veri kümesi ve görevi üzerinde eğitilmek üzere ikinci bir hedef ağa aktarılır. Bu yöntemin işe yaraması için aktarılacak bilgilerin genel bilgiler olması gerekir, yani kaynak göreve özgü olmak yerine hem kaynak hem de hedef görevler için uygun olan bilgiler aktarılır (Pan ve Yang, 2009).



Şekil 1.1. Geleneksel ML ile TL arasındaki farklar

ML, DL ve zaman serisi analizi gibi veriye dayalı yöntemler satış ve talep tahmininde yaygın olarak kullanılmaktadır. Yeni piyasaya sürülen ürünlerde doğru modelleri eğitmek için yeterli eğitim verisi bulunmamaktadır. Bu durumda, tahmin performansını iyileştirmek için uzman sistemler uygulanabilmektedir. Uzmanlar, bilgi ve deneyimlerine dayanarak yeni ürün satışlarını tahmin etmek için benzer ürünlerin geçmiş satışları hakkında bilgi aktarımını kullanır. TL yaklaşımı, alanlar arasında analitik bilgi paylaşımına olanak tanır. TL, uzman sistemleri desteklemek veya ikame etmek için kullanılabilir (Pan ve Yang, 2009). Örneğin, eğitim verisinin az olması nedeniyle geleneksel tahmin teknikleri, ML ve DL modellerinin etkili olmaması durumunda transfer öğrenme uygulanabilir. Derin sinir ağı tabanlı TL, listelenen stok ürünler ve yeni ürünler arasında bilgi aktarımı için analitik bir yaklaşım sunarak satış ve talep tahmininde uzmanlar kadar doğru ve güvenilir sonuçlar verebilir (Karb vd., 2020).

Görüntü işleme alanında GoogleNet, ImageNet, ResNet vb. birçok ön eğitilmiş hazır ağlar bulunmaktadır. Özellikle görüntü işleme alanında oldukça yaygın kullanılan TL yaklaşımının zaman serisi verileri üzerinde kullanımı yaygın değildir (Iman vd., 2023). Zaman serisi problemlerinde ise özellik çıkarmak ve sınıflandırma yapmak için kullanılan ön eğitilmiş bir ağ olan TimeNet (Malhotra vd., 2017) ve TimeGPT-1 (Garza ve Mergenthaler-Canseco, 2023) bulunmaktadır (Iman vd., 2023). Bilindiği kadarıyla TZY'de satış ve talep tahmini için zaman serileri ile ön eğitilmiş bir ağ bulunmamaktadır. Bu motivasyon ile bu tez kapsamında satış ve talep tahmini için derin öğrenme tabanlı transfer öğrenme yöntemleri geliştirilmiştir.

Yukarıda verilen bilgiler doğrultusunda tez kapsamında satış ve talep tahmini için derin transfer öğrenme metodolojisi geliştirilmesi hedeflenmiştir. Bu hedef doğrultusunda satış ve talep verileri hem tek değişkenli hem de çok değişkenli zaman serileri olarak ele alınmıştır. Bu kapsamda, ilk olarak tek değişkenli satış zaman serileri için çeşitli derin öğrenme modelleri geliştirilmiş ve doğrulanmıştır. İkinci olarak, belirlenen derin öğrenme yöntemi ile tek kaynaklı transfer öğrenme modelleri geliştirilmiş ve doğrulanmıştır. Daha sonra, transfer öğrenme ile topluluk stratejileri entegre edilmiştir. Üçüncü olarak transfer öğrenmede hedefe uygun kaynak seçimi ve bu seçimin açıklanabilirliğine yönelik benzerlik analizi, öznitelik çıkarımı ve öznitelik seçimi ile kümeleme çalışmaları gerçekleştirilmiştir. Dördüncü olarak, kümeleme tabanlı çoklu kaynak seçimi ile tek kaynaklı transfer öğrenme modelleri çok kaynaklı transfer öğrenme modelleri geliştirilmiştir. Böylece tek değişkenli satış zaman serileri için kümeleme ve topluluk tabanlı derin transfer öğrenme metodolojisi geliştirilmiştir. Son olarak, geliştirilen metodoloji çok değişkenli zaman serileri için genişletilmiştir.

1.1. Tezin Amacı ve Kapsamı

Bu tezin amacı tedarik zinciri yönetiminde satış ve talep tahmini için bilgi aktarımını sağlayan derin transfer öğrenme tabanlı bir metodoloji geliştirmektir. Bu metodoloji aşağıda belirtilen üç ana başlık altında ele alınmıştır:

1. Satış ve talep tahmini için tek kaynaklı derin transfer öğrenme tabanlı modelin geliştirilmesi,
2. Çok kaynaklı derin transfer öğrenme tabanlı modelin geliştirilmesi,
3. Tahmin doğruluğunu iyileştirebilmek ve transfer öğrenmede etkili bir şekilde bilgi aktarımını sağlayabilmek için transfer öğrenme öncesi girdi verilerini kümeleyen modelin geliştirilmesidir.

1.2. Araştırma Soruları

Tez çalışması ile cevaplanması planlanan araştırma soruları (AS) aşağıda belirtilmektedir:

AS-1: Tedarik zincirinde satış ve talep tahmininde tek kaynaklı ve çok kaynaklı derin transfer öğrenme tahmin performansını iyileştirir mi?
AS-2: Doğru ve etkin bir tahmin yapabilmek için transfer öğrenmede hangi bilgiler, ne zaman ve nasıl aktarılabilir?
AS-3: Transfer öğrenmede bilgi aktarımını sağlayacak ağ mimarisi seçimi ne olmalıdır?
AS-4: Transfer öğrenme kararının verilmesinde hedef ve kaynak alan arasında benzerlik analizi kullanılabilir mi?
AS-5: Transfer öğrenme kararının verilmesinde topluluk yöntemler (torbalama, yığınlama, vb.) kullanılabilir mi?
AS-7: Veri setlerinin kümelenmesi satış ve talep tahmininde transfer öğrenmenin performansını iyileştirilebilir mi?
AS-8: Farklı uzunluktaki zaman serilerinin kümelenmesinde derin kümeleme yöntemleri kullanılabilir mi? Ürüne ait özellikler (ürün fiyatı, markası, bozulabilirlik durumu, vb.) ve çevresel faktörler (hava durumu, özel günler, ekonomik göstergeler, promosyon vb.) gibi ek özniteliklere sahip zaman serilerinin kümelenmesinde derin kümeleme yöntemleri kullanılabilir mi?
AS-9: Çok değişkenli zaman serilerinde transfer öğrenme tahmin performansını iyileştirir mi?

1.3. Tezin Literatüre Katkısı

Tez kapsamında gerçekleştirilen çalışmaların literatüre katkıları aşağıda maddeler halinde belirtilmektedir:

- Satış ve talep tahmininde hem tek değişkenli hem de çok değişkenli satış zaman serileri için derin transfer öğrenmeye dayalı çözüm yaklaşımları geliştirilmiştir.

- Transfer öğrenme kararlarında ve negatif transfere karşı benzerlik analizi, derin kümeleme yöntemi ve topluluk yöntemler ele alınmıştır. Farklı uzunluklarda satış ve talep zaman serileri kullanılmıştır ve küme sayısının bilinmediği durumlar ele alınmıştır.
- Öncelikle farklı uzunluklardaki tek değişkenli satış zaman serileri ile satış tahmininde ERP metriğini kullanarak uzaklık tabanlı tek kaynaklı derin transfer öğrenme modeli geliştirilmiştir. Satış tahmini alanında bilindiği kadarıyla ERP metriğini kullanan ilk çalışmadır (Erol ve İnkaya, 2024).
- Tek değişkenli satış zaman serileri için geliştirilen metodolojide transfer öğrenmede kaynak seçimi, bilginin aktarılabilirliği ve açıklanabilirlik çalışmaları entegre bir şekilde ele alınmıştır. Bu entegre çözüm yaklaşımı ile literatüre katkı sağlanmaktadır.
- Çok değişkenli satış zaman serileri için çok boyutlu benzerlik ölçütüne ve derin kümelemeye dayalı bir çözüm metodolojisi geliştirilmiştir.
- Kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımı ile transfer öğrenmede çoklu kaynak seçimi için bir çözüm yaklaşımı sunulmuştur. Ek olarak, tahmin performansını arttırmak adına hedef ağdaki eğitim hatalarına dayalı çoklu kaynakların ağırlıklandırılmasına yönelik bir strateji geliştirilmiştir. Satış tahmini alanında bilindiği kadarıyla kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı geliştiren ilk çalışmadır.

2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Bu bölümde, tez kapsamında kullanılan temel yöntemler anlatılmaktadır ve ardından konu ile ilgili literatürdeki çalışmalar özetlenmektedir. Bölüm 2.1'de derin öğrenme yöntemleri, Bölüm 2.2'de transfer öğrenme yöntemleri, Bölüm 2.3'te kümeleme yöntemleri ve Bölüm 2.4'te topluluk öğrenme yöntemleri anlatılmaktadır. Bölüm 2.5'te literatürde derin öğrenme, transfer öğrenme, kümeleme ve topluluk öğrenme yöntemlerini kullanarak satış ve talep tahmini yapan çalışmalara yer verilmektedir.

2.1. Derin Öğrenme Yöntemleri

Bu bölümde, tez çalışması kapsamında ele alınan dokuz adet derin öğrenme yöntemi açıklanmıştır.

2.1.1. Derin sinir ağı

Derin sinir ağı (Deep Neural Network – DNN), girdi ve çıktı katmanları arasında çok sayıda katman bulunan yapay bir sinir ağı yapısıdır. Ağın derinleşmesi ile genelleme yeteneğinin sığ ağlara göre daha iyi olduğu öne sürülmektedir (Goodfellow vd., 2016). Şekil 2.1'de sığ ve derin sinir ağı mimarileri gösterilmektedir. DNN, öznitelik mühendisliği gereksinimini azaltabilme ve karmaşık problemleri birden çok katmanla çözebilme özelliklerine sahiptir (Kaya ve Yıldırım, 2020). Ayrıca DNN mimarisi sayesinde geçmiş ve mevcut verileri kullanarak geleceği tahmin etme konusunda iyi performans göstermektedir (Hossain vd., 2015; Gashler ve Ashmore, 2016; Hu vd., 2016).

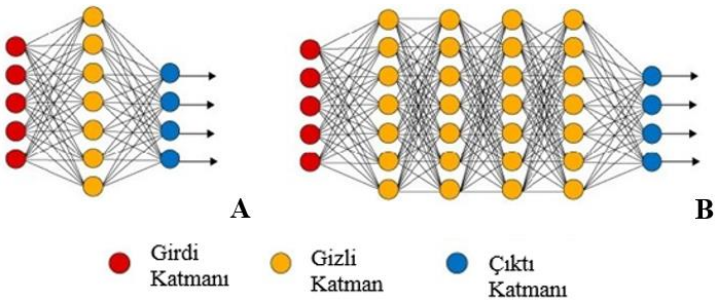
2.1.2. Derin otokodlayıcı ağı

Otokodlayıcı (Autoencoders – AE), karmaşık veri temsillerini öğrenmek için eğitilmiş denetimsiz bir sinir ağıdır. Otokodlayıcı, girdi veri kümesinin boyutunu azaltarak en az kayıp ile girdiye en yakın çıktıyı üreten, öznitelikleri ve yapısal bilgileri öğrenebilen bir yapay sinir ağı mimarisidir (Canchumuni vd., 2019).

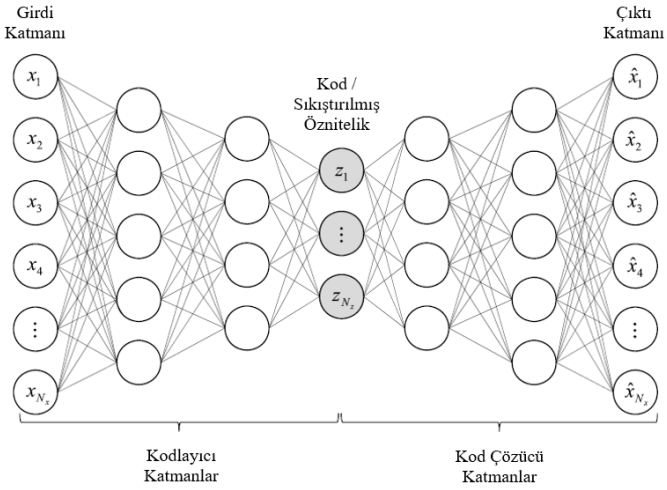
Otokodlayıcı genellikle veri sıkıştırma ve gürültü azaltma amaçlarıyla kullanılmaktadır. Bununla birlikte son yıllarda derin üretken modellerin yapı taşları olarak da sıklıkla kullanılmaya başlanmıştır (Goodfellow vd., 2016).

Her bir katmandaki çıktıların ardışık katmanın girdilerine bağlandığı ve birden fazla otokodlayıcı katmanından oluşan sinir ağı, derin otokodlayıcı (Deep Autoencoders – DAE), olarak adlandırılmaktadır (Canchumuni vd., 2019). Şekil 2.2'de altı adet tam bağlantılı katmanından oluşan standart bir derin otokodlayıcı ağı gösterilmektedir. İlk üç katman kodlayıcıdır ve girdi uzayını bir öznitelik uzayına eşlemekle sorumludur. Son üç katman ise kod çözücüdür ve ters eşlemeye karşılık gelir. Merkezi katmana ise kod denir. Eğitim süreci, girdi ile çıktı arasındaki farkı ölçen bir kayıp fonksiyonunun minimize edilmesini içerir. Bu fark değeri ile otokodlayıcının başarısı değerlendirilir.

Eğitim sonrasında girdinin (x) en önemli özellikleri kod (z) ile temsil edilebilir. Kod çözücü doğrusal olduğunda ve kayıp fonksiyonu ortalama karesel hata olduğunda, otomatik kodlayıcı temel bileşen analizi (Principal Component Analysis – PCA) ile aynı alt uzayı kapsayacak niteliktedir (Hossain vd., 2015). Bu nedenle, otokodlayıcılar PCA'nın doğrusal olmayan genellemeleri olarak yorumlanabilir (Deng vd., 2017; Canchumuni vd., 2019).



Şekil 2.1. Sinir ağı mimarileri
A) Sığ sinir ağı mimarisi **B)** Derin sinir ağı mimarisi



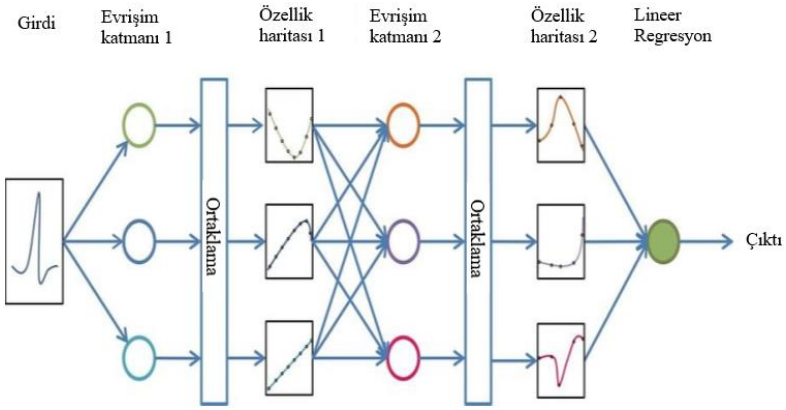
Şekil 2.2. Derin otokodlayıcı mimarisi (Kraus vd., 2020).

2.1.3. Evrişimli sinir ağı

Evrişimli sinir ağı (Convolutional Neural Network – CNN) görüntü sınıflandırma ve nesne tanıma gibi genellikle görsel görevlerde kullanılmak üzere geliştirilmiş bir sinir ağı türüdür (LeCun vd., 2015). Özellikle girdi verisindeki özniteliklerin çıkarılmasında ve karmaşık temsillerin öğrenilmesinde etkilidir. Bir CNN mimarisi girdi katmanı, evrişim katmanı, ortaklama katmanı, tam bağlantılı katman ve çıktı katmanı içermektedir.

İşlenen girdi verisinin boyutuna göre CNN üç gruba ayrılır: Tek boyutlu, iki boyutlu ve üç boyutlu CNN. Tek boyutlu CNN (One Dimensional Convolutional Neural Network – 1D-CNN) zaman serisi ya da ses sinyali gibi tek boyutlu ilişkilerin olduğu veride kullanılır. İki boyutlu CNN görüntülerin, üç boyutlu CNN ise video tanıma ya da üç boyutlu medikal görüntülerin işlenmesinde kullanılır (LeCun vd., 2015).

1D-CNN girdi dizisindeki zamansal ilişkilerin öğrenilmesini sağlar. Şekil 2.3'te 1D-CNN'nin ağ yapısı görülmektedir (Vangala ve Vadlamani, 2020). Buna göre, evrişim ve ortaklama (havuzlama) katmanları girdi dizisinden öznitelik çıkarma için kullanılır. Bu katmanlar sonucunda oluşan çıktı dizisi tam bağlantılı katmana gönderilerek nihai çıktı elde edilir (Ragab vd., 2020).



Şekil 2.3. 1D-CNN ağ mimarisi (Vangala ve Vadlamani, 2020).

1D-CNN'de evrişim işlemi çekirdeğin girdi verisi üzerinde kaydırılmasını ve girdinin her bir alt dizisi ile çekirdeğin iç çarpımının hesaplanmasını içerir. Elde edilen çıktı dizisi, ileri seviyeli öznitelikleri çıkarmak ve modelin nihai çıktısını oluşturmak için ek katmanlardan geçirilen bir öznitelik haritasıdır. Evrişim katmanındaki ağırlık paylaşımı eğitimdeki parametre sayısını azaltarak 1D-CNN'nin daha erken ve daha hızlı yakınsamasını sağlar (Gashler ve Ashmore, 2016). 1D-CNN'de evrişim işlemi Denklem (2.1) ile ifade edilir (Kiranyaz vd., 2021):

$$x_k^l = \sum_{i=1}^{N_{l-1}} conv1D(w_{ik}^{l-1}, s_i^{l-1}) + b_k^l \tag{2.1}$$

Burada sırasıyla x_k^l ve b_k^l l. evrişim katmanında k. nörondaki girdi ve sapma, w_{ik}^{l-1} (l-1). evrişim katmanındaki i. nörondan l. evrişim katmanındaki k. nörona kadar evrişim çekirdeği, s_i^{l-1} (l-1). evrişim katmanında i. nörondaki çıktı, N_{l-1} (l-1). evrişim katmanındaki nöron sayısı ve $conv1D$ ise evrişim operatörüdür. x_k^l girdisine aktivasyon fonksiyonu uygulandığında ara çıktı elde edilir (Kiranyaz vd., 2021).

Evrişim katmanından sonra gelen ortaklama katmanı, ana öznitelikleri korurken boyut azaltma amacıyla öznitelik haritasından alt örneklem oluşturur. Bu amaçla, ortaklamada ortalama ya da maksimum yaklaşımları kullanılabilir (Wang vd., 2017). Ortalama ortaklama yaklaşımında belirli komşulukta ortalama değer alınır. Maksimum ortaklama yaklaşımında ise komşuluktaki en büyük değer alınır. Ortaklama yaklaşımları aynı zamanda aşırı uyum sorununu çözmeyi amaçlamaktadır (Wang vd., 2017). Sonrasında

elde edilen öznitelikler tam bağlantılı katmana girdi olarak verilir (Swapna vd., 2018). Tam bağlantılı katman düzleştirilmiş girdi ile çalışır ve girdiler bu katmandaki tüm nöronlar ile bağlıdır (Fukuoka vd., 2018). Nihai çıktının elde edilmesi için uygun bir aktivasyon fonksiyonu seçilir. Buna göre sınıflandırma problemlerinde sınıf etiketlerinin olasılık dağılımı, tahmin problemlerinde ise sürekli bir çıktı değeri oluşturulur.

2.1.4. Tekrarlayan sinir ağı

Tekrarlayan sinir ağı (Recurrent Neural Network – RNN) sıralı verinin işlenmesinde kullanılır. Geçmiş girdi verisini hafızada tutarak mevcut verinin işlenmesini sağlayan bir yapay sinir ağıdır (Acı ve Doğansoy, 2022). Bu amaçla, tekrarlayan bağlantı adı verilen bir yapı kullanılır ve bir zaman adımındaki çıktının bir sonraki zaman adımına girdi olarak kullanılmasına izin verir.

Klasik sinir ağı mimarisinde girdi-çıktı bağımsızlığı olduğu varsayılır. Fakat RNN sıralı şekilde gelen verileri kullanır ve dizideki her verinin çıktısı önceki hesaplamalara bağlıdır (Acı ve Doğansoy, 2022). Böylece ileri beslemeli sinir ağından farklı olarak RNN hafızaya sahiptir. Bir dizinin her elamanı için aynı görevi yerine getirmeleri ve çıktının önceki hesaplamalara bağlı olması nedeniyle tekrarlayan olarak adlandırılır (Acı ve Doğansoy, 2022). Şekil 2.4'te örnek bir RNN ve ağın açılımı gösterilmektedir.

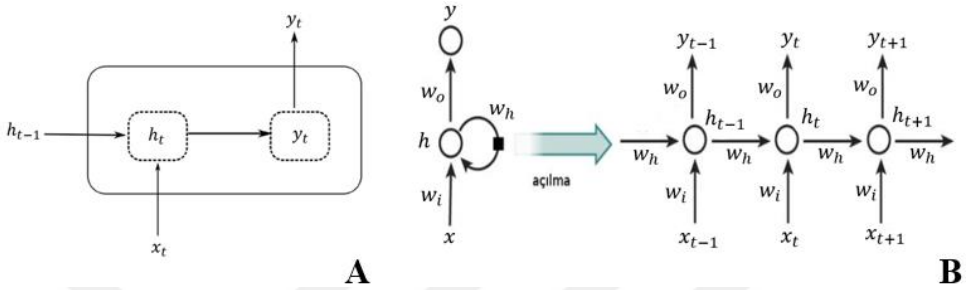
RNN'nin tekrarlayan birimlerine ait matematiksel işlemler Denklem (2.2)-(2.3)'te verilmiştir. h_t, t zaman adımındaki gizli durumu gösterir ve bu yapı RNN hücresinin hafızasıdır. RNN hücresinin t zaman adımındaki girdi ve çıktısını sırasıyla x_t ve y_t göstermektedir. W_{xi}, W_{hi} ve W_o sırasıyla girdi, gizli durum ve çıktı için ağırlık matrislerini; b_i ve b_o ise sapma vektörlerini ifade etmektedir. Mevcut gizli durum, önceki zaman adımının gizli durumuna ve mevcut girdiye bağlıdır. Ayrıca, mevcut durumu bir sonraki duruma bağlayan geri bildirim döngüleri ile desteklenir.

Uygulama alanlarına ve kullanılan veri türlerine göre aktivasyon fonksiyonları değişebilmektedir. Genellikle aktivasyon fonksiyonu olarak gizli katmanda sigmoid

fonksiyonu (σ), çıktı katmanında ise hiperbolik tanjant fonksiyonu (*tanh*) kullanılmaktadır.

$$h_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (2.2)$$

$$y_t = tanh(W_o h_t + b_o) \quad (2.3)$$



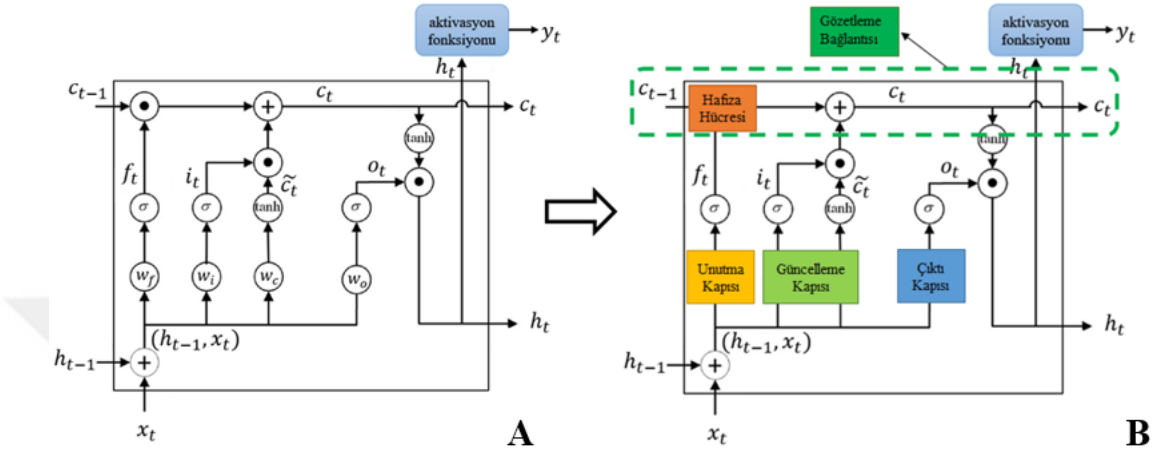
Şekil 2.4. Tekrarlayan sinir ağı mimarisi
A) Tekrarlayan sinir ağı modeli **B)** Tekrarlayan sinir ağının açılımı

RNN'nin eğitimi için geriye yayılım algoritması kullanılır. Ancak uzun dizilerde geriye yayılımdaki türev işlemleri esnasında türevlerin yok olması ya da aşırı büyümesi ile karşılaşılabilir. Bu problemlere kaybolan gradyan (vanishing gradient) ve patlayan gradyan (exploding gradient) adı verilir. Bu durumda ağın doğru bir şekilde eğitimi zorlaşmaktadır. RNN'de yaşanan bu problemlerin üstesinden gelmek için uzun kısa-süreli bellek ağı (Long Short Term Memory – LSTM) ve kapılı tekrarlayan birim ağı (Gated Recurrent Unit – GRU) gibi derin öğrenme yöntemleri önerilmiştir (Hochreiter ve Schmidhuber, 1996).

2.1.5. Uzun kısa-süreli bellek ağı

RNN'de uzun zaman dizilerinin eğitiminde karşılaşılabilen kaybolan/patlayan gradyan problemlerine çözüm olarak LSTM ağı önerilmiştir (Bengio vd., 1994; Hochreiter ve Schmidhuber, 1996). LSTM uzun dönemli zaman bağımlılığı bilgilerini depolayabilen ve girdi ile çıktı dizilerini uygun şekilde eşleştirebilen tekrarlayan sinir ağı mimarisidir (Hochreiter ve Schmidhuber, 1996; Greff vd., 2016).

LSTM birimleri bir blok içinde yer alır ve bir veya birden fazla blok bir araya gelerek LSTM mimarisini oluşturur. Şekil 2.5'te örnek bir LSTM bloğu gösterilmektedir. LSTM bloğu veri girişi, çıkışı ile akışını kontrol eden üç kapı ve bilgiyi tutan bir hafıza hücresi içermektedir (Greff vd., 2016).



Şekil 2.5. LSTM sinir ağı mimarisi
A) LSTM bloğu **B)** LSTM kapıları

LSTM ağındaki işlemler Denklem (2.4)-(2.10) ile ifade edilebilir. Burada zaman adımı *t*, girdi kapısı *i*, unutma kapısı *f*, hafıza hücresi *c*, çıktı kapısı *o*, aktivasyon fonksiyonları *tanh* ve σ, noktasal çarpım $\times$, eleman bazında çarpım ise $\otimes$ ile gösterilmektedir. Ayrıca, *t* zaman adımında girdi vektörü x_t, çıktı vektörü y_t, hafıza hücresi durumu c_t, girdi kapısının aday değeri i_t, hafıza hücresinin güncellenmiş değeri $\tilde{c}_t$, unutma kapısının aday değeri f_t, çıktı kapısının aday değeri o_t ve gizli katmanın aday değeri h_t ile gösterilmektedir. W_f, W_i, W_c, W_o ağırlık matrislerini; b_f, b_i, b_c, b_o ise sapma vektörlerini temsil etmektedir (Greff vd., 2016; Erol ve İnkaya, 2024).

$$f_t = \sigma(W_f \otimes [h_{t-1}, x_t] + b_f) \quad (2.4)$$

$$i_t = \sigma(W_i \otimes [h_{t-1}, x_t] + b_i) \quad (2.5)$$

$$\tilde{c_t} = tanh\,(W_c \otimes [h_{t-1}, x_t] + b_c) \quad (2.6)$$

$$c_t = f_t \times c_{t-1} + i_t \times \tilde{c_t} \quad (2.7)$$

$$o_t = \sigma(W_o \otimes [h_{t-1}, x_t] + b_o) \quad (2.8)$$

$$h_t = o_t \times tanh(c_t) \quad (2.9)$$

$$y_t = \sigma(h_t) \quad (2.10)$$

LSTM ağında Denklem (2.4)'te yer alan unutma kapısı ile veri silinmesi, Denklem (2.5)'te yer alan girdi (güncelleme) kapısı ile veri güncellemesi, Denklem (2.6) ile ise hücre durumuna eklenecek yeni aday değerlerin oluşturulması sağlanır. Denklem (2.7)'de unutma kapısı ile mevcut durum, girdi kapısı ile aday girdiler güncellenerek mevcut hafıza hücresinin durumu oluşturulur. Denklem (2.8)-(2.10) ile ağın nihai çıktısı; çıktı kapısı, önceki ve mevcut hafıza hücresi durumlarına bağlı olarak hesaplanır (Greff vd., 2016). Özetle, LSTM ağının ileri yayılım işlemleri Denklem (2.4)-(2.10)'da verilmiştir.

LSTM ağının eğitimi için geriye yayılım algoritması kullanılır. Çıktı katmanından girdi katmanına doğru geriye yayılırken ağdaki hatayı en küçükleyecek şekilde ağırlıklar üzerinde güncellemeler yapılır. Tahmin problemleri için kayıp fonksiyonu olarak genellikle ortalama karesel hata (Mean Squared Error – MSE) kullanılmaktadır. Her iterasyonda kayıp fonksiyona ait gradyanlar (kısmi türevler) hesaplanır ve ardından her bir katmanın ağırlıkları ile sapma değerleri güncellenir. İleri yöndeki ve geri yöndeki bu döngü belirlenen bir durdurma kriteri elde edilene kadar tekrarlanır (Sherstinsky, 2020).

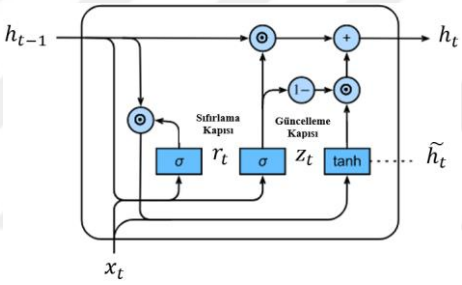
LSTM'de RNN'nin aksine sabit bir W matrisi yerine Denklem (2.7)'deki gibi her bir adımda değişen farklı bir unutma kapısı değeri kullanılmaktadır. Böylece kaybolan ve patlayan gradyan problemlerinin önüne geçilebilmektedir. Hata terimine bağlı olarak her kapının ağırlığı bir optimizasyon algoritması kullanılarak güncellenmektedir (Hochreiter ve Schmidhuber, 1996; Greff vd., 2016).

2.1.6. Kapılı tekrarlayan birim ağı

RNN, kısa süreli hafızaya sahip olduğu için uzun dönemli bağımlılıkları taşıyamaz. Bu duruma çözüm olarak GRU yöntemi sunulmuştur. Şekil 2.6'da mimarisi gösterilen GRU, Cho vd. (2014) tarafından önerilmiştir. Her tekrarlayan birimin farklı zaman adımlarındaki bağımlılıkları yakalaması amacıyla GRU birim içindeki bilgi akışını ayarlayan, ancak ayrı bir hafıza hücresine sahip olmayan kapı adı verilen dahili mekanizmalara (LSTM birimine benzer) sahiptir (Chung vd., 2014).

GRU bilgi aktarmak için gizli durumu kullanır ancak hücre durumu içermez. Bu nedenle GRU, LSTM'ye kıyasla daha az tensör işlemine sahiptir ve eğitim süresi daha hızlıdır (Chung vd., 2014). GRU yapısında bulunan kapılar, bir dizideki hangi verilerin saklanması veya unutulmasının önemli olduğunu öğrenir ve tahmin için yalnızca ilgili verileri tutmayı, ilgisiz verileri ise unutmayı sağlayabilir. Böylece uzun diziler boyunca ilgili bilgileri iletebilir (Chung vd., 2014).

GRU, güncelleme ve sıfırlama adında iki kapı içerir. Güncelleme kapısı, önceki gizli durumun ne kadarının saklanması gerektiğini ve onu güncellemek için yeni girdinin ne kadarının kullanılması gerektiğini belirler. Sıfırlama kapısı ise yeni gizli durum hesaplanırken önceki gizli durumun ne kadarının unutulması gerektiğini ve yeni girdinin ne kadarının dikkate alınması gerektiğini belirler (Chung vd., 2014).



Şekil 2.6. GRU sinir ağı mimarisi

GRU'daki işlemler Denklem (2.11)-(2.15) ile ifade edilebilir (Chung vd., 2014). Burada *t* bir zaman serisinin zaman adımlarını, *z* ve *r* sırasıyla güncelleme ve sıfırlama kapılarını gösterir. σ ve *tanh* aktivasyon fonksiyonlarını, × ise noktasal çarpımı ifade etmektedir. x_t, y_t, $\tilde{h}_t$, h_t, z_t, r_t sırasıyla *t* zaman adımındaki girdi, çıktı, geçici gizli durum, gizli durum, güncelleme kapısı ve sıfırlama kapısı vektörlerini göstermektedir. b_z, b_r, b_h sapma vektörlerini ve $W_z, W_r, W_h, U_z, U_r, U_h$ ağırlık vektörlerini ifade etmektedir.

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (2.11)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (2.12)$$

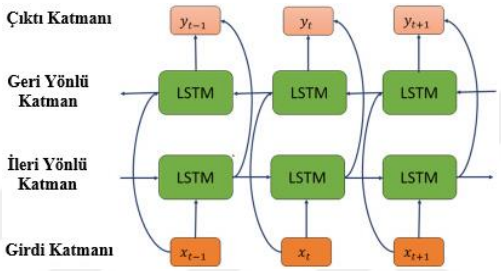
$$\tilde{h_t} = tanh\,(W_h x_t + U_h(r_t \times h_{t-1}) + b_h) \quad (2.13)$$

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times \tilde{h}_t \quad (2.14)$$

$$y_t = \sigma(h_t) \quad (2.15)$$

2.1.7. Çift yönlü uzun kısa-süreli bellek ağı

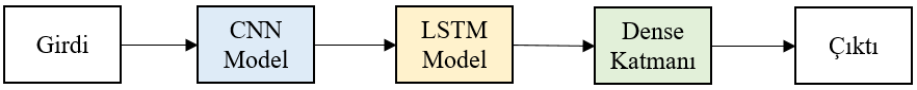
Çift yönlü LSTM (Bidirectional LSTM – Bi-LSTM) iki ayrı LSTM modelinden oluşur. LSTM modellerinden biri girdiyi ileri yönde alır, diğeri ise geri yönde alır. Böylece, ağdaki bilgi miktarını arttırarak zaman serisindeki iki yönlü uzun vadeli bağımlılıkları öğrenebilir (Graves ve Schmidhuber, 2005). Şekil 2.7'de Bi-LSTM mimarisi gösterilmektedir. Bi-LSTM sıralı verinin işlenmesinde etkilidir ancak girdi dizisi ileri ve geri yönlü iki defa işlendiği için hesap karmaşıklığı yüksektir.



Şekil 2.7. Bi-LSTM sinir ağı mimarisi

2.1.8. CNN-LSTM hibrit ağı

CNN-LSTM sıralı girdi verisinin işlenmesi için CNN ve LSTM ağını birleştiren bir sinir ağı mimarisidir. CNN evrişim işlemleri ile sıralı verideki özniteliklerin çıkarılmasını sağlar. CNN ile elde edilen öznitelikler LSTM ağına girdi olarak verilir ve LSTM sıralı verideki uzun dönemli bağımlılıkların öğrenilmesini sağlar. Şekil 2.8'de CNN-LSTM'nin ağ yapısı verilmiştir (Wang vd., 2016).



Şekil 2.8. CNN-LSTM sinir ağı mimarisi

Hibrit modelde, her bir girdi dizisi CNN modeli tarafından işlenen çoklu alt dizilere bölünür. Örneğin, 36 zaman adımı içeren bir dizi 12 zaman adımı içeren üç alt diziye bölünerek aynı CNN katmanında işlenir. Bu amaçla, zamana göre yayılan (TimeDistributed) katman sarmalayıcı olarak kullanılabilir. CNN alt modelinin çıktısı

LSTM katmanı tarafından işlenir. Bunu, tam bağlantılı bir katman ve tek adımlı tahminler yapmak için bir çıktı katmanı izleyebilir. CNN-LSTM modeli için alt dizi sayısı ve uzunluğu iki önemli parametredir (Wang vd., 2016).

2.1.9. Evrişimli uzun kısa-süreli bellek ağı

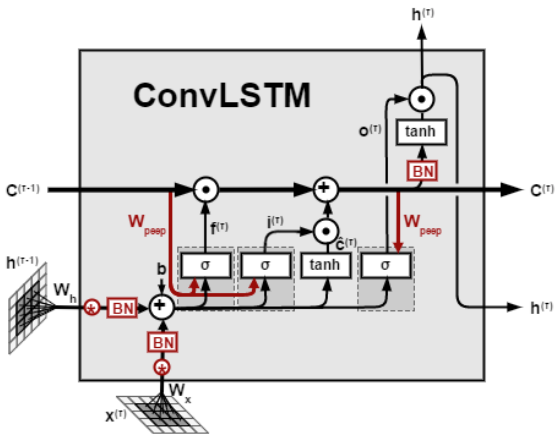
Evrişimli (convolutional) LSTM veya kısaca ConvLSTM, LSTM hücresi içinde evrişim işlemi içeren bir LSTM katmanı türüdür. ConvLSTM'de LSTM hücresinde yer alan her kapıdaki matris çarpımı evrişim işlemiyle değiştirilir. Böylece, çok boyutlu verilerde uzamsal özellikler yakalanır. ConvLSTM, girdiden duruma ve durumdan duruma geçişlerde evrişimli yapılar içeren, zaman-mekansal (spatiotemporal) tahmin için kullanılabilen tekrarlayan bir sinir ağı türüdür (Shi vd., 2015). Örnek bir ConvLSTM mimarisi Şekil 2.9'da gösterilmektedir.

ConvLSTM ve LSTM arasındaki temel fark, girdi boyutlarının sayısıdır. LSTM girdi verileri tek boyutlu olduğundan video, uydu, radar görüntü veri seti gibi mekansal dizi verileri için uygun değildir. ConvLSTM ise üç boyutlu veriler için tasarlanmıştır. Durumların girdilerle aynı sayıda satıra ve aynı sayıda sütuna sahip olmasını sağlamak için evrişim işlemini uygulamadan önce dolgu yapılması gerekebilir. CNN-LSTM'de olduğu gibi, girdi verileri her bir dizinin sabit sayıda zaman adımına sahip olduğu alt dizilere bölünür, ancak her bir alt dizideki satır sayısı da belirtilmelidir. CNN'dekine benzer şekilde ConvLSTM katmanı, girdi dizilerini okurken filtre sayısını ve kullanılan çekirdeğin boyutunu belirlemeye olanak verir. Bu katmanın çıktısı, bir öznitelik haritası olup önce düzleştirilir ve sonrasında bir çıktı katmanına girdi olarak verilir (Shi vd., 2015).

2.2. Transfer Öğrenme

Transfer öğrenme, bir problemin çözümü ile elde edilen bilginin ilgili fakat farklı bir problemde kullanılmasını sağlayan bir makine öğrenmesi yöntemidir (Pan ve Yang, 2009). Makine öğrenmesi ve derin öğrenmede her bir görev için sıfırdan öğrenme gerçekleştirilir. Ancak transfer öğrenme kavramı ile kaynak görevlerden öğrenilen

birtakım bilgiler farklı hedef görevlere aktarılarak öğrenme gerçekleştirilir. TL, hedef görevi modellerken hızlı ilerleme veya gelişmiş performans sağlar.



Şekil 2.9. ConvLSTM sinir ağı mimarisi (Shi vd., 2015)

Transfer öğrenme temelde iki aşama içerir. (Pang ve Yang, 2009). İlk aşamada kaynak veri kümesi ve kaynak görev ile temel bir ağ oluşturulur. İkinci aşamada ise ilk aşamada kaynak ağdan öğrenilen özellikler (ağırlıklar, parametreler, vb. tüm bilgiler) olduğu gibi veya yeniden tasarlanarak bir hedef veri kümesi ve görevi üzerinde eğitilmek üzere ikinci bir hedef ağa aktarılır. Transfer öğrenme sürecinde aşağıda verilen üç soru sorulur (Pang ve Yang, 2009):

Neyin aktarılacağı: TL yapılırken uygulanan ilk ve en önemli aşamadır. Kaynak veriden hedefe hangi bilgi aktarılınca hedef görevin başarısının artacağına cevap aranan bölümdür. TL'nin amacına ulaşabilmesi için aktarılacak bilgilerin genel bilgiler olması gerekir, yani kaynak göreve özgü olmak yerine hem kaynak hem de hedef görevler için uygun olan bilgiler aktarılmalıdır. Bazı bilgilerin kullanılan veriye/göreve özgü olabileceği ve bazısının kullanılan alan ne olursa olsun ortak olabileceğine dikkat edilmelidir.

Ne zaman aktarılacağı: Bu aşamada verinin hangi durumlarda aktarılacağı ya da hangi durumlarda aktarılmaması gerektiği belirlenir. Bazı durumlarda kaynak veri ve hedef veri birbirinden çok farklı olabilir. Bu gibi durumlarda aktarım başarılı olmayabilir. Hatta

hedef veri üzerindeki eğitimin başarısını düşürebilir. Bu duruma *olumsuz aktarım/negatif transfer* denir.

Nasıl aktarılacağı: Burada öğrenmenin aktarımı için hangi yöntemlerin kullanılacağına karar verilir. Var olan yöntemlerin direkt kullanılması veya değiştirilmesi yaklaşımına gidilebilir.

Transfer öğrenmede neyin aktarılacağı ile ilgili aşağıdaki yaklaşımlardan bahsedilmektedir (Pang ve Yang, 2009):

Örnek transferi (Instance transfer): Kaynak verideki bazı örneklerin hedef veride kullanılmasıdır. Kaynak verideki belirli örneklere uygun ağırlık değerleri verilmesi ve hedef veri üzerinde kullanılmasıyla başarı artırılabilir. Kaynak ve hedef verisinin ortak birçok özelliğe sahip olması durumunda geçerlidir.

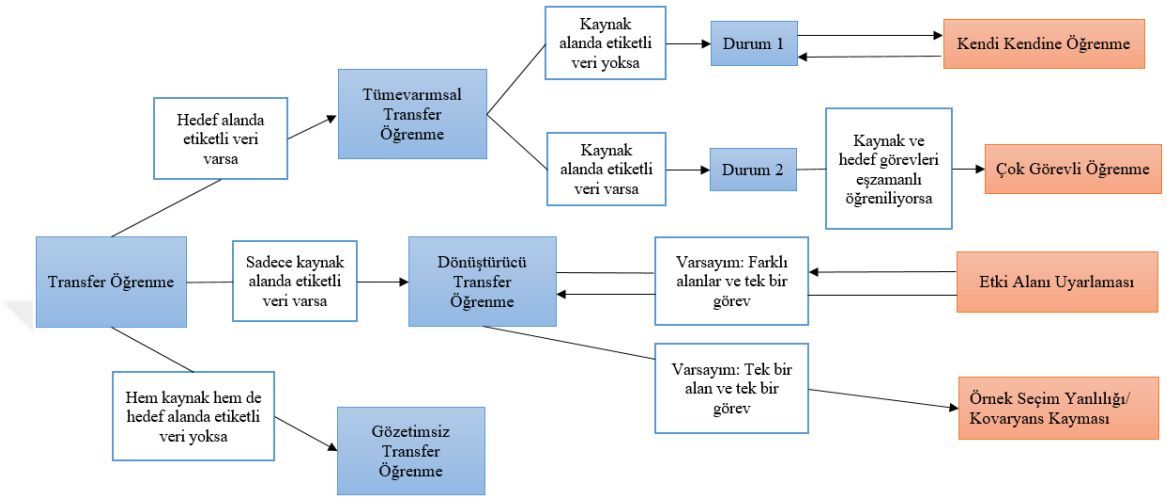
Özellik temsili transferi (Feature representation transfer): Kaynak verideki özellik temsillerinin hedef veride kullanılmasıdır. Kaynak ve hedef verisinin ortak özelliklere sahip olması durumunda geçerlidir. Ortak özelliklerin yanı sıra yalnızca kaynak veya yalnızca hedefte olan özellikler de bulunabilir. Kaynak verisinden hedef verisine daha iyi özellik gösterimleri bularak hedef görevdeki performansın artması beklenir.

Parametre transferi (Parameter transfer): Birbirine yakın hedef ve kaynak görevleri için benzer parametrelerin kullanılması durumudur. Transfer edilecek bilgi paylaşılan ortak parametreler ile kaynak görevden hedef göreve aktarılır.

İlişki kurma tecrübesinin transferi (Relational knowledge transfer): Kaynak verideki bazı ilişkilerin hedefteki ilişkilere benzetilmesi durumudur. Kaynak veri ve hedef veri arasında bazı ilişkilerin olduğu varsayılır, veri kaynakları arasında ilişkisel bilginin eşleştirilmesi yapılır.

2.2.1. Transfer öğrenme yöntemleri

Transfer öğrenme kaynak/hedef görev ve verilerine göre Şekil 2.10'da görüldüğü gibi üç ana başlık altında sınıflandırılmaktadır (Pang ve Yang, 2009).



Şekil 2.10. Transfer öğrenme yöntemlerinin sınıflandırılması (Pan ve Yang, 2009)

Tümevarımsal Transfer Öğrenme (Inductive Transfer Learning): Hedef görevin kaynak görevden farklı ancak ilgili olduğu durumlardır. Görevlerdeki verilerin etiketli veya etiketsiz olması bakımından ikiye ayrılır. *Durum 1*, kaynak verinin etiketli veri içermemesi durumudur. Bu durumda TL kendi kendine öğrenme (self taught learning) gibi çalışır. Kendi kendine öğrenmede kaynak ve hedef verilerin etiket uzayı farklıdır, bu da kaynak verisinin doğrudan kullanılamayacağı anlamına gelir. *Durum 2* ise kaynak görevde etiketli verinin bulunmasıdır. Bu durumda, TL çok görevli öğrenme (multi-task learning) gibi çalışır. Ancak çok görevli öğrenmenin eğitildiği tüm görevlerde başarı göstermesi beklenirken TL'de amaç hedef görev başarısının yüksek olmasıdır. Kaynak görev için geliştirilen bir model hedef görev için başlangıç noktası olarak kullanılır.

Dönüştürücü Transfer Öğrenme (Transductive Transfer Learning): Kaynak ve hedef görevlerin aynı ancak verilerinin farklı olmasıdır. Hedef görevde etiketli verinin bulunmadığı, kaynak görevde ise bol miktarda etiketli verinin olduğu durumlarda kullanılır. Öznitelik uzayının kaynak ve hedef veri kümeleri için farklı olması ve aynı olması durumlarına göre dönüştürücü transfer öğrenme iki farklı şekilde ifade edilebilir.

Gözetimsiz Transfer Öğrenme (Unsupervised Transfer Learning): Tümevarımsal öğrenme aktarımına benzemektedir. Farklı olarak etiketsiz görevler üzerine odaklanmaktadır. Buradaki kaynak ve hedef veri kümeleri benzerdir ancak görevler birbirinden farklıdır. Kümeleme, boyut küçültme vb. görevlerde kullanılır.

2.2.2. Transfer öğrenmenin kullanım amaçları ve potansiyel faydaları

Transfer öğrenme, zamandan tasarruf etmek veya daha iyi bir performans elde etmek için kullanılmaktadır. Torrey ve Shavlik (2010), transfer öğrenmeyi kullanırken göz önünde bulundurulması gereken üç olası faydayı Şekil 2.11'deki gibi tanımlamaktadır:

- **Daha yüksek başlangıç:** Transfer öğrenmede hedef modelin başlangıçtaki performansı sıfırdan öğrenme olduğu durumdan daha yüksektir.
- **Daha yüksek eğim:** Transfer öğrenmede kaynak modelin eğitim hızı sıfırdan öğrenme olduğu durumdan daha diktir.
- **Daha yüksek asimptot:** Transfer öğrenmede eğitimli modelin yakınsaması sıfırdan öğrenme olduğu durumdan daha iyidir.

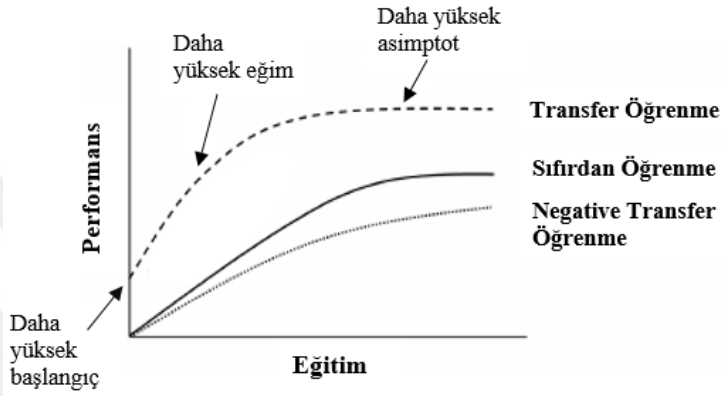
CNN, RNN, LSTM gibi derin öğrenme yöntemlerini sıfırdan eğitmek yerine transfer öğrenme kullanmanın olası yararları yukarıda bahsedilen üç fayda ile ilişkili olarak aşağıdaki gibi belirtilebilir (Brownlee, 2017):

- **Daha hızlı eğitim süresi:** Önceden eğitilmiş modellerdeki ağırlıklar birçok bilgiyi barındırır. Dolayısıyla bu bilgileri kullanıp ince ayar yapılarak yeni model daha hızlı eğitilir. Hatta bazı problemler için TL kullanılarak çok küçük dönem (2-4 epoch) sayılarında bile yüksek başarı elde edilebilmektedir (Brownlee, 2017).
- **Daha az veri:** Sıfırdan oluşturulan modellerin en büyük dezavantajı eğitim için büyük ölçekli bir veri kümesinin gerekmesidir. Bu veri kümelerinin oluşturulması için ciddi bir zaman gerekir. Bunun yerine önceden eğitilmiş modellere ince ayar yapılması daha az veri kullanarak yüksek performanslar elde edilmesini sağlar. Öte yandan her problem için yeterli veriye sahip olunamayabilir. Bu tarz

durumlarda da TL ile önceden geliştirilemeyen yüksek performanslı modeller geliştirilebilir.

- **Daha iyi performans:** Önceden eğitilmiş modellere yeni tam bağlantılı katman(lar) eklenerek yapılan işlemin başarıyı iyileştirdiği görülmektedir.

İdeal olarak, yukarıda bahsedilen tüm faydalar başarılı bir TL uygulamasında görülmektedir.



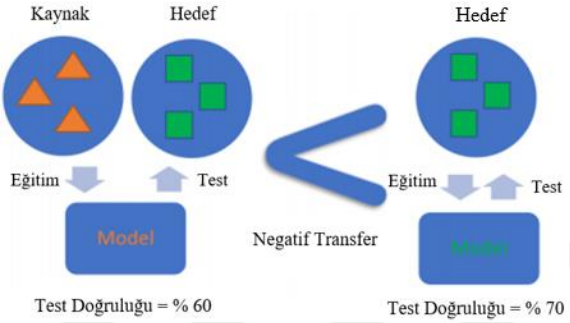
Şekil 2.11. TL performans özeti

2.2.3. Negatif transfer öğrenme

Transfer öğrenme ile gerçek hayat uygulamalarındaki eksik veya yetersiz eğitim verisi durumlarının birçoğunda başarılı sonuçlar elde edilmektedir (Niu vd., 2020). Ancak TL her zaman performansı arttırmayabilir hatta bazen mevcut performansın düşmesine de sebep olabilir. Bu duruma *negatif transfer* (*olumsuz aktarım*) denir.

TL'de başarının her zaman garanti edilememesinin nedenleri arasında temel varsayımların ihlal edilmesi olabilmektedir: 1) görev korelasyonu, hem kaynak hem de hedef alanındaki öğrenme görevleri ilişkili/benzerdir; 2) alan korelasyonu, kaynak alan ve hedef alan veri dağılımları çok farklı değildir; ve 3) uygun bir model her iki alana da uygulanabilir. Bu varsayımların ihlali negatif transfere yol açabilir. Kaynak alanlardan ilgisiz bilgilerin aktarılması ve bu bilgilerin istenmeyen bir şekilde hedefe dahil edilmesi Şekil 2.12'de gösterildiği gibi hedef alandaki öğrenme performansını azaltır (Pang ve Yang, 2009; Zhang vd., 2022).

Negatif transfer, TL'de araştırılmaya devam edilen zor bir problemdir. Negatif transfer öğrenmeyi önlemek veya azaltmak amacıyla kaynak ile hedef alanların benzer kısımlarını bulmak, farklı alanların/görevlerin/modellerin/özelliklerin aktarılabilirliğini değerlendirmek üzere birçok araştırma yapılmıştır (Ye ve Dai, 2018; Hirt vd., 2020; Zhang vd., 2022).



Şekil 2.12. Negatif transfer öğrenme

2.2.4. Derin transfer öğrenme

Derin öğrenmede kullanılan transfer öğrenme şekli, tümevarımsal transfer öğrenmedir. Derin öğrenme tabanlı transfer öğrenme (derin transfer öğrenme), verilen bir hedef alan ve hedef görevde bir derin öğrenme modelinin performansını arttırmak için kaynak alan ve kaynak görevde yer alan bilginin keşfedilerek hedefe aktarımını ifade eder (Niu vd., 2020). Derin transfer öğrenmeyi tahmin problemlerinde kullanabilmek için iki yaklaşım vardır (Brownlee, 2017):

a) Gelişmiş Model Yaklaşımı: Dört adımdan oluşmaktadır:

1. **Kaynak görev seçimi:** Büyük miktarda veriye sahip kaynak görev seçilir. Kaynak göreve ait girdi ve çıktı verileri elde edilir.
2. **Kaynak model geliştirme:** Kaynak görev için kaynak model geliştirilir.
3. **Modeli hedef alanda yeniden kullanma:** Kaynak görevine uyan model, daha sonra hedef görev modeli olarak başlangıçta kullanılabilir. Uygulanan modelleme tekniğine bağlı olarak, modelin tüm bölümleri veya parçaları başlangıç için kullanılabilir.

4. **Modeli ayarlama:** İsteğe bağlı olarak, hedef görev için modelin mevcut girdi-çıktı çifti verilerine adapte edilmesi veya parametrelerin ince ayarının yapılması gerekebilir. Kısacası kaynak model hedef alan ve görev için ayarlanmalıdır.

b) Ön-Eğitimli Model Yaklaşımı: DL alanında yaygın olarak kullanılmaktadır ve üç adımdan oluşmaktadır:

1. **Kaynak model seçimi:** Mevcut modellerden önceden eğitilmiş bir kaynak modeli seçilir. Örneğin; görüntü işleme alanında GoogleNet, AlexNet, vb.; doğal dil işleme alanında ise GloVe, Microsoft ResNet, vb. gibi ön eğitilmiş ağlar bulunmaktadır (Iman vd., 2023). Bu kaynak modellerden hedef göreve uygun olacak şekilde ağ seçimi yapılır.
2. **Modeli hedef alanda yeniden kullanma:** Önceden eğitilmiş model, daha sonra ilgili hedef görevdeki bir model için başlangıç noktası olarak kullanılabilir. Uygulanan modelleme tekniğine bağlı olarak, modelin tüm bölümleri veya parçaları başlangıç için kullanılabilir.
3. **Modeli ayarlama:** Hedef görev için mevcut olan girdi-çıktı çifti verilerine adapte edilmeli veya parametrelerin ince ayarı yapılmalıdır.

2.2.5. Transfer öğrenmede kaynak seçimi

Transfer öğrenme, bilgiyi bir kaynak görevden bir hedef göreve aktarmayı amaçlamaktadır. Makine öğrenmesi ve derin öğrenme yöntemlerine transfer öğrenme uygulanabilir (Pan ve Yang, 2009). Özellikle DL modellerinin eğitim için büyük veriye ihtiyaç duyması nedeniyle TL derin öğrenmede popüler hale gelmiştir (Iman vd., 2023). Bu tezde, transfer öğrenme içeren derin öğrenme yöntemleri ile zaman serisi tahmin problemlerinin çözümü ele alınmaktadır.

Derin sinir ağı tabanlı transfer öğrenmede aktarım mekanizması iki adımdan oluşur. İlk adımda kaynak veri kümesi ile ön eğitilmiş uygun bir derin öğrenme modeli seçilir. İkinci adımda hedef veri kümesi ile ön eğitilmiş model parametrelerine ince ayar yapılır. Uygun kaynak verinin seçimi, bilginin başarılı (olumlu) aktarımını kolaylaştırırken negatif

(olumsuz) aktarımını sınırlandırır. Geleneksel zaman serisi tahmin görevlerinde kaynak veri kümesi seçimi manuel olarak gerçekleştirilir. Burada öncelikle mevcut kaynak veri kümeleri eğitilerek ön eğitilmiş çeşitli kaynak modeller elde edilir. Ardından ön eğitilmiş kaynak modellerden hedef veri kümesine transfer öğrenme gerçekleştirilerek tahmin performansları karşılaştırılır. Bu şekilde gerçekleştirilen manuel seçim, zaman ve hesaplama açısından maliyetlidir.

Son yıllarda hedefe uygun kaynak seçimi ile ilgili araştırmalar artmıştır (Ye ve Dai, 2018; Ye ve Dai, 2021; Gu vd., 2021; Gu ve Dai, 2018). Bu araştırmalarda kaynak olarak en büyük benzerliğe sahip zaman serilerini seçmek için zaman serilerinin benzerliği hesaplanmıştır. Benzerlik hesabı için uzaklık tabanlı ve özellik tabanlı (veri dağılımı, veri homojenliği, vb. özellikler) ölçütler kullanılmıştır. Kaynak ve hedef veri kümelerinin benzerliğinin yanı sıra derin transfer öğrenmede kullanılan sinir ağının benzerliğine odaklanan çalışmalar mevcuttur (Ye ve Dai, 2018; Hirt vd. 2020; Zhang vd., 2023).

Transfer öğrenmede negatif transfer öğrenmeyi engellemek ve transfer öğrenmedeki aktarım başarısını arttırmak (aktarılabilirlik) için benzerlik analizi kullanılabilir (Meiseles ve Rokach, 2020). Kaynak ve hedef kümeleri arasındaki benzerlikler uzaklık ölçütleri ile hesaplanabilir. Uzaklık ölçütleri elastik olmayan ve elastik uzaklık ölçütleri olarak ikiye ayrılmaktadır. Elastik olmayan uzaklık ölçütleri aynı uzunluktaki tek veya çok değişkenli zaman serileri arasındaki benzerlikleri hesaplar. Öklid uzaklığı, Manhattan uzaklığı elastik olmayan ölçütlerdir. Elastik uzaklık ölçütleri ise farklı uzunluklardaki tek değişkenli zaman serileri arasında doğrusal olmayan eşleştirmeler yaparak şekilsel benzerliklerini hesaplama olanağı sağlar (Abanda vd., 2019). Literatürde farklı boyutlardaki zaman serileri arasındaki benzerlikleri hesaplamak için Dinamik Zaman Bükmesi (Dynamic Time Warping – DTW), En Uzun Ortak Dizi (Longest common subsequence – LCSS), Gerçek Diziler için Düzenleme Uzaklığı (Edit distance for real sequences – EDR), Gerçek Cezalı Düzenleme Uzaklığı (Edit distance with real penalty – ERP) ve Zaman Çarpıtma Düzenleme Uzaklığı (Time Warp Edit Distances – TWED) gibi ölçütler kullanılmaktadır (Jiang, 2020).

2.2.6. Transfer öğrenmede açıklanabilirlik

Açıklanabilirlik konusu, derin öğrenme ve yapay zeka alanındaki araştırmacılar tarafından giderek daha fazla ilgi görmektedir (Linardatos vd., 2020; Minh vd., 2022). Açıklanabilirlik, öğrenme algoritmalarının kararlarını anlama ve insanlar tarafından yorumlanabilir hale getirme yeteneği olarak tanımlanabilir (Linardatos vd., 2020).

Minh vd., (2022) açıklanabilirliği modelleme öncesi açıklanabilirlik, yorumlanabilir modeller ve modelleme sonrası açıklanabilirlik olmak üzere üç kategoride ele almışlardır. Modelleme öncesi açıklanabilirlik, veri kümesine ilişkin genel bir bakış elde etmek için çeşitli veri ön işleme ve özellik keşfetme yöntemlerini kapsar. Bu yöntemler veri analizi, veri özetleme ve veri dönüştürme yöntemleridir (Minh vd., 2022; Ali vd., 2023).

Yorumlanabilir modeller, basit yapısı nedeniyle yorumlanabilen makine öğrenmesi yöntemlerini ifade etmektedir. Bu modeller, doğası gereği yorumlanabilir modeller ve hibrit yorumlanabilir modeller olmak üzere iki grupta incelenmektedir. Doğası gereği açıklanabilir ve yorumlanabilir makine öğrenmesi yaklaşımları, model eğitim süreci sırasında yorumlanabilirlik kısıtlarını dahil ederek tahminlerin insanlar tarafından anlaşılabilecek açıklamalarını sağlamak üzere tasarlanmıştır. Bu yaklaşımlar doğrusal modeller (lineer regresyon, lojistik regresyon, vb.), ağaç tabanlı modeller (karar ağacı, rastgele orman, vb.) ve kural tabanlı modeller olmak üzere üç kategoriye ayrılabilir. Hibrit yorumlanabilir model yaklaşımı ise hem yüksek performans hem de model yorumlanabilirliğini sağlamak amacıyla yorumlanması zor olan bir kara kutu model ile doğası gereği yorumlanabilir bir modeli birleştirmeye dayalı bir yaklaşımdır. Bu yaklaşımlara klasik makine öğrenmesi modellerinin derin formülasyonları (derin en yakın komşu, derin kalman filtreleri, yapısal değişken otokodlayıcılar, vb.) ve makine öğrenmesi ile derin öğrenme modellerinden oluşturulan topluluk öğrenme yöntemleri örnek verilebilir (Minh vd., 2022; Ali vd., 2023).

Modelleme sonrası açıklanabilirliğe ait yöntemler, bir algoritmanın eğitim sürecinde nasıl performans gösterdiğini veya sağlanan herhangi bir girdi için nasıl tahminler ürettiğini açıklamayı amaçlamaktadır. Bu yöntemler genel olarak özellik önemi yöntemleri, örnek

tabanlı yöntemler, modelden bağımsız yöntemler ve modele özgü yöntemler olmak üzere dört kategoriye ayrılabilir (Minh vd., 2022; Ali vd., 2023).

Transfer öğrenme bağlamında, açıklanabilirlik teknikleri, özellikle bir kaynaktan bir hedefe bilgi aktarımını izlemek ve anlamak için oldukça önemlidir. Bir kaynak görevden öğrenilen özelliklerin hedef görevde nasıl kullanıldığını ve etkin olup olmadığını açıklamak için karar ağaçları ve kural tabanlı yöntemler gibi yorumlanabilir yöntemler kullanılabilir. Transfer öğrenmede hedefe uygun kaynak seçimi ve hangi bilgilerin aktarılacağı (aktarılabilirlik) konusunda ise modelleme öncesi veri analizi ve özellik çıkarımı yöntemleri açıklanabilirliği sağlayabilir. Model sonrası açıklanabilirlik yaklaşımlarında ise aktarılan öznitelik vektörlerinin görselleştirilmesi ile transfer öğrenme modelinde hangi özelliklerin öğrenildiğini ve bu özelliklerin hedef görevde nasıl kullanıldığı açıklanabilir (Ali vd., 2023). Bu tür çalışmalar, transfer öğrenmenin karmaşıklığını azaltmak, modellerin güvenilirliğini artırmak ve insanların modelin kararlarını anlamasına yardımcı olmak için önemlidir (Minh vd., 2022; Ali vd., 2023).

Bu tezde, transfer öğrenmede açıklanabilirlik için modelleme öncesi açıklanabilirlik ele alınmıştır. Bu kapsamda, öznitelik çıkarımı ve öznitelik seçimi ile yorumlanabilir modellerin kullanımı açıklanmıştır.

Öznitelik Çıkarımı: Öznitelik çıkarımı (feaure extraction), ham veriden öznitelik (değişken) türetmektir. Öznitelik çıkarımı, veri bilimi ile makine öğrenmesi süreçlerinde ve model performanslarında oldukça önemlidir. Amaç, çıkarılan bu öznitelikler aracılığıyla bir veri kümesinin ayrıntılı bir temsilini oluşturmaktır (Bonifati vd., 2022).

Öznitelik çıkarımı, mevcut özniteliklerden yeni öznitelikler oluşturarak bir veri kümesindeki özniteliklerin sayısını azaltmayı amaçlar. Bu yeni indirgenmiş öznitelik seti, orijinal öznitelik setinde yer alan bilgilerin çoğunu özetleyebilmelidir. Bu şekilde, orijinal özniteliklerin özet bir versiyonu, orijinal veri setinin kombinasyonundan oluşturulabilir. Öznitelik çıkarımı ile model doğruluğu iyileştirilebilir, aşırı uyum riski azaltılabilir, eğitim daha hızlı yapılabilir ve veri görselleştirmesi geliştirilebilir. Ayrıca en önemlisi

modelin açıklanabilirliğinde ve yorumlanabilirliğinde artış sağlanabilir (Herff ve Krusienski, 2019).

Literatürde temel bileşen analizi, bağımsız bileşen analizi, makine öğrenmesi yöntemleri (otomatik kodlayıcılar) ile öznitelik çıkarımı yapılmaktadır (Herff ve Krusienski, 2019).

Öznitelik Seçimi: Öznitelik seçimi (feature selection), bir veri kümesinde bulunan özelliklerin (özniteliklerin) sayısını azaltarak, modelin basit ve etkili olmasını sağlayan bir veri ön işleme adımıdır (Bonifati vd., 2022). Öznitelik seçimi, veri analizi, makine öğrenmesi ve istatistiksel modelleme gibi birçok alanda yaygın olarak kullanılır (Bonifati vd., 2022). Öznitelik seçimi; veri kümesindeki gereksiz, tekrarlayan veya düşük varyanslı öznitelikleri çıkararak veri setinin boyutunu azaltır. Daha az öznitelik hem modelin daha hızlı eğitilmesini hem de daha az bellek ve hesaplama gücü gerektirmesini sağlar. Bu durum modelin genelleme performansını artırır ve aşırı uyum riskini azaltır. Öznitelik sayısının fazla olması, modelin daha karmaşık ve anlaşılması zor hale gelmesine neden olabilir. Öznitelik seçimi, modelin daha anlaşılır ve yorumlanabilir hale gelmesine yardımcı olur (Guyon ve Elisseeff, 2003).

Literatürde filtreleme yöntemleri (Pearson korelasyonu, Ki-Kare testi, vb.), sarmalayıcı yöntemler ve gömülü yöntemler (lineer regresyon, karar ağacı, rastgele orman, vb.) ile öznitelik seçimi yapılmaktadır (Herff ve Krusienski, 2019). Bu tezde, öznitelik seçimi için gömülü yöntemlerden biri olan rastgele orman yöntemi kullanılmıştır. Rastgele orman, birden fazla karar ağacının birleşiminden meydana geldiği için öncelikle karar ağacı yöntemi açıklanmıştır. Sonrasında, rastgele orman yöntemi verilmiştir.

- **Karar Ağacı**

Karar ağaçları (Decision Tree – DT), sınıflandırma ve regresyon görevleri için kullanılan parametrik olmayan denetimli bir öğrenme yöntemidir (Han vd., 2012). Amacı, veri özniteliklerinden çıkarılan basit karar kurallarını öğrenerek hedef değişkenin değerini tahmin eden bir model oluşturmaktır. Şekil 2.13'te görüldüğü üzere bir karar ağacı kök düğüm, dallar, iç düğümler ve yaprak düğümlerden oluşan hiyerarşik bir yapıya sahiptir

(Géron, 2017). Bir karar ağacı kök düğümle başlar. Kök düğümden çıkan dallar karar düğümleri olarak da bilinen iç düğümlere beslenir. Her bir gözlem düğümler yardımıyla mevcut özniteliklere bağlı olarak yaprak düğümlerde sınıflandırılır. Düğüm sayısı arttıkça modelin karmaşıklığı artar. Yaprak düğümler, veri kümesindeki tüm olası sonuçları temsil eder (Géron, 2017).

Karar ağacı, veri kümesini bir dizi karar kuralları uygulayarak alt bölümlere ayırır. Karar ağacı algoritmalarında bölünmenin nasıl gerçekleşeceği ağacın doğruluğunu etkileyen faktörlerden biridir. Karar ağacında veri kümesini olabildiğince en iyi açıklayan kökün ve özniteliklerin seçimi önemlidir. Sınıflandırma ve regresyon problemleri için bölme kriterleri farklıdır. Karar ağaçlarında bir düğümü iki veya daha fazla alt düğüme bölmeye karar vermek için bölme kriteri kullanır. Bölme işlemi düğümlerin homojenliğini (saflığını) arttırır. Bölme kriterinin seçimi, hedef değişkenin türüne bağlıdır. Karar ağaçlarında kategorik değişkenler için entropi, Gini ve sınıflandırma hatası; sürekli değişkenler için ise en küçük kareler yöntemi kullanılmaktadır (Han vd., 2012).



Şekil 2.13. Karar ağacı yapısı

Gini indeksi bir düğümün saflığını gösterir ve 0 ile 1 arasında bir değer alır. Entropi rastgeleliği, belirsizliği ve beklenmeyen durumun ortaya çıkma olasılığını gösterir. Bir düğümün homojenliğinin ölçütüdür. Entropi daha dengeli bir ağaç çıkarmaya meyilli iken, Gini frekansı fazla olan sınıfı ayrıştırmaya meyillidir. *t* düğümü için Gini indeksi ve entropi değerleri sırasıyla Denklem (2.16) ve Denklem (2.17) ile hesaplanmaktadır. Burada $p(j|t)$, *t* düğümündeki *j* sınıfının göreceli frekansı ifade etmektedir (Han vd., 2012).

$$Gini(t) = 1 - \sum_j [p(j|t)]^2 \quad (2.16)$$

$$Entropy\ (t) = -\sum_j p(j|t) \log_2 p(j|t) \quad (2.17)$$

Karar ağacı algoritması aşağıdaki adımları içermektedir (Han vd., 2012):

Adım 1: Veri kümesindeki örnekleri en iyi ayıran öznitelik belirlenir.

Adım 2: Seçilen öznitelik ile ağacın bir düğümü oluşturulur ve bu düğümden iç (çocuk) düğümler veya ağacın yaprakları oluşturulur. İç düğümlere ait örnekler belirlenir.

Adım 3: Adım 2'de oluşturulan her bir iç düğüm için aşağıdaki durumlar kontrol edilir:

- Belirli bir safsızlık değeri elde edildiğinde işlem sonlandırılır.
    - Tüm nesneler aynı sınıfa ait ise işlem sonlandırılır.
    - Düğüm aynı öznitelik değerlerinden oluşuyor ise işlem sonlandırılır.
- Belirli bir derinlik seviyesinde işlem sonlandırılır.
- Belirli sayıda yaprak düğümü elde edildiğinde işlem sonlandırılır.
- Safsızlık iyileşmediğinde işlem sonlandırılır.
- Diğer durumda Adım 2'ye dönülür.

Literatürde sınıflandırma ağaçları (Hunt, ID3, C4.5, C5.0, CART) ve regresyon ağaçları (C4.5, CART) olmak üzere iki kategoride birçok algoritma önerilmiştir (Han vd., 2012).

- **Rastgele Orman**

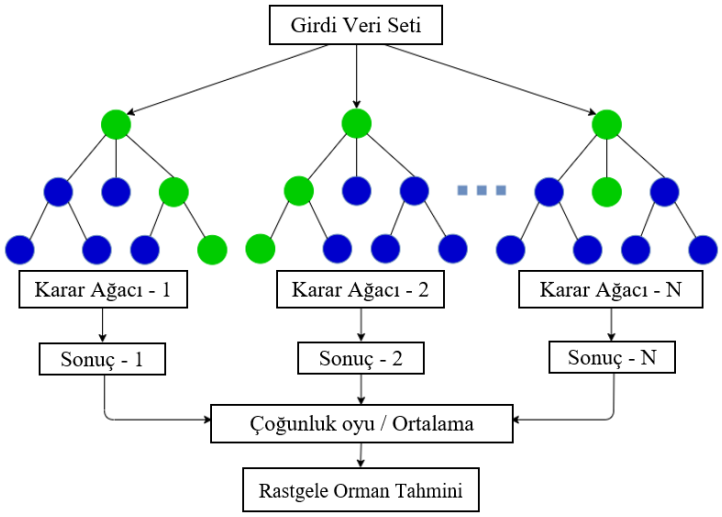
Rastgele orman (Random Forest – RF), Breiman tarafından geliştirilen bir makine öğrenmesi tekniğidir. Sınıflandırma ve regresyon problemleri için kullanılır (Han vd., 2012). Şekil 2.14'te gösterildiği gibi rastgele orman, birden çok karar ağacının ürettiği tahminlerin bir araya getirilerek değerlendirilmesine dayanır. Rastgele ormanda karar ağacından farklı olarak kök düğümü bulma ve düğümleri bölme işlemleri rastgele yapılmaktadır (Han vd., 2012).

Breiman (2001) tarafından geliştirilen CART (Classification and Regression Trees) yöntemi yüksek performansa sahip olmasına rağmen aşırı öğrenmeye meyillidir. Aşırı öğrenmeyi önlemek için Breiman (2001) torbalama ile rastgele alt uzay yöntemlerini birleştirerek rastgele orman tekniğini geliştirmiştir. Rastgele ormanda ağaçlar için

gözlemler yeniden örnekleme yöntemiyle, bölen öznitelikler (değişkenler) ise rastgele uzay yöntemiyle seçilir. Bu sayede genellenebilirlik (rassallık) ve aşırı öğrenme gibi makine öğrenmesinin temel problemlerine çözüm yaklaşımı sunulmuştur. Rastgele orman yöntemi özniteliklerin tahmin üzerindeki nispi önem derecelerini ölçmesi açısından yararlı bir yöntemdir (Breiman, 2001).

Rastgele orman yöntemi her bir özniteliğin (değişkenin) önem derecesini belirlemede kullanılabilir. Önem derecesi, bir özniteliğin ne kadar etkili olduğunu değerlendirmeyi sağlar ve modelin yorumlanabilirliğini artırır (Gregorutti vd., 2017; Breiman, 2001). Rastgele orman yönteminde ortalama azalan safsızlık, permütasyon önemi ve z-skor yöntemlerine göre önem dereceleri hesaplanabilir (Breiman, 2001; Gregorutti vd., 2017). Bu tezde, ortalama azalan safsızlık (mean decrease impurity) yöntemi kullanılmıştır. Bu yöntem, bir özniteliğin karar ağacındaki safsızlığı ne kadar azalttığını ölçer. Tahmin problemlerinde her bir özniteliğin belirsizliği ne kadar azalttığı MSE ile hesaplanır (Gregorutti vd., 2017).

Rastgele ormanlar hatalara ve aykırı değerlere karşı dayanıklıdır. Her bölmede rastgele öznitelik seçimi yapıldığı için daha az sayıda öznitelik dikkate alınır, bu sayede büyük veri tabanlarında etkilidirler (Han vd., 2012).



Şekil 2.14. Rastgele orman yapısı

2.3. Kümeleme Yöntemleri

Veri setinde, etiketlerin (bağımlı/hedef değişken) bulunmadığı durumlarda kısacası algoritmanın verileri öğreneceği bir eğitmeninin olmadığı ve algoritmanın farklı yöntemler kullanarak etiketleme yaptığı durumlara denetimsiz öğrenme (unsupervised learning) adı verilir (Ghahramani, 2003). Kümeleme, bir dizi veri nesnesini (gözlemi) herhangi bir harici bilgi olmadan benzer özelliklerine göre gruplara ayıran bir denetimsiz öğrenme yöntemidir. Aynı kümedeki nesnelerin birbirlerine benzer, farklı kümelerdeki nesnelerin birbirlerinden farklı olması hedeflenir. Kümeleme, verilerde doğal olarak oluşan gruplanmayı keşfetmeyi amaçlar (Han vd., 2012).

Kümelemenin kullanım amaçlarından biri, veriyi benzerliklerine göre gruplayarak küme etiketlerini belirlemektir. Diğer bir kullanım amacı ise veri seti için gruplar oluşturarak kullanıcıya özet bilgi sunmaktır. Kümeleme ayrıca veri ön işleme adımı olarak kullanılabilir (Aggarwal ve Reddy, 2014).

Literatürde merkez tabanlı (bölümlemeli) kümeleme, yoğunluk tabanlı kümeleme, dağılım tabanlı kümeleme, ızgara tabanlı kümeleme, hiyerarşik kümeleme, topluluk kümeleme, spektral kümeleme ve çoklu görünüm kümeleme gibi kümeleme yöntemleri bulunmaktadır (Aggarwal ve Reddy, 2014; Ren vd., 2022). Kümeleme yöntemi seçimi kullanılacak veri türüne ve uygulamanın amacına göre farklılık gösterir (Aggarwal ve Reddy, 2014). Bölümlemeli (k-ortalamalar, k-medoids) ve hiyerarşik kümeleme algoritmaları diğer kümeleme algoritmalarına göre basit ve kolay uygulanabilir olmaları nedeniyle geniş bir uygulama yelpazesinde kullanılmaktadır (Aggarwal ve Reddy, 2014).

Bu tezde zaman serisi verilerinin kümelenmesi ele alınmıştır. Zaman serisi verileri, korelasyon tabanlı veya şekil tabanlı kümelenebilir. Korelasyon tabanlı kümeleme analizinde, belirli bir zaman serisinin ne zaman oluşturulduğuna ilişkin zaman bilgisi dikkate alınarak kümeleme gerçekleştirilir. Sensör verileri, hisse senedi gibi borsa verilerinde kullanılabilir. Şekil tabanlı kümeleme analizinde, iki zaman serisi arasındaki benzerlik hesabı önemlidir (Aggarwal ve Reddy, 2014). Zaman serilerinde benzerlik hesabı için zaman serilerinin uzunluklarına bağlı olarak elastik (DTW, ERP, TWED, vb.)

ve elastik olmayan (Öklid, Minkowski, vb.) benzerlik ölçütleri kullanılmaktadır (Chen ve Ng, 2014).

2.3.1. Derin kümeleme

Derin öğrenme tabanlı kümeleme yaklaşımı (derin kümeleme), verilerden etkili bir şekilde kümelemeye fayda sağlayacak özellikleri çıkarmayı ve öğrenilen özelliklerle kümeleme gerçekleştirmeyi amaçlar (Ren vd., 2022). Ren vd. (2022), kullanılan ağ ve veri yapısı açısından derin kümeleme yöntemlerini tek-görünümlü, çok-görünümlü, yarı denetimli ve transfer öğrenme olmak üzere dört kategoride özetlemişlerdir.

Derin tek-görünümlü kümeleme: Geleneksel kümeleme görevlerinde verilerin tek-görünümlü olarak bilinen aynı biçim ve türe sahip olduğu varsayılır. Bu veriler için temsillerin derin sinir ağları tarafından çıkarılması derin kümelemenin önemli bir özelliğidir. Temsil öğreniminin (representation learning) amacı, veriyi daha anlamlı ve işlenebilir bir formata dönüştürerek makine öğrenmesi modellerinin daha iyi performans göstermesini sağlamaktır (Bengio vd., 2013). Temsil öğrenimi, makine öğrenmesi görevlerinde öznitelik (veya temsil) öğrenmenin önemini göstermektedir (Bengio vd., 2013). Bu kapsamda, derin tek-görünümlü kümeleme algoritmaları derin otomatik kodlayıcı (DAE) tabanlı, derin sinir ağı (DNN) tabanlı, varyasyonel otomatik kodlayıcı (Variational Autoencoder – VAE) tabanlı, üretken çekişmeli ağı (Generative Adversarial Network – GAN) tabanlı ve grafik sinir ağı (Graph Neural Network – GNN) tabanlı derin kümeleme bulunmaktadır (Ren vd., 2022).

Derin çok-görünümlü kümeleme: Çok görünümlü veri, çeşitli kaynaklardan elde edilen farklı formatta veya türde olan verileri ifade eder (Ren vd., 2022). Derin çok-görünümlü kümelemenin amacı, kümeleme performansını artırmak için çok-görünümlü verilerin içerdiği tamamlayıcı bilgileri kullanmaktır. Aynı zamanda bu bilgiler derin tek-görünümlü kümeleme için yol gösterici olabilir. Derin çok-görünümlü kümeleme derin gömülü kümeleme tabanlı, alt uzay kümeleme tabanlı ve grafik sinir ağı tabanlı olmak üzere üç kategoride incelenebilir (Ren vd., 2022).

Yarı denetimli öğrenmeye dayalı derin kümeleme: Yarı denetimli kümelemede amaç fonksiyonu denetimsiz kümeleme kaybı ve kısıtlama kaybının birleşimidir (Ren vd., 2022). Mevcut yarı denetimli kümeleme yöntemlerinden daha iyi kümeleme performansı elde etmek amacıyla modele kısıtlama kaybı olarak ek bilgiler eklenir ve yarı denetimli hale getirilir. Bu şekilde ilişki kısıtlamaları ile sinir ağlarının birleştirilmesine dayanan derin kümeleme yaklaşımıdır (Ren vd., 2022).

Transfer öğrenmeye dayalı derin kümeleme: Etiketlere sahip bir kaynak alandan/görevden öğrenilen bilgi veya kalıpların, farklı ancak ilgili bir hedef alana/göreve aktarılmasını ve bu bilgi transferi ile kümeleme performansını iyileştirmeyi amaçlayan derin kümeleme yaklaşımıdır (Ren vd., 2022).

2.3.2. Küme doğrulama yöntemleri

Kümeleme kalitesinin değerlendirilmesine yönelik çeşitli küme doğrulama (cluster validity) yöntemleri geliştirilmiştir (Maulik ve Bandyopadhyay, 2002). Bu sayede kümeleme çalışmalarında küme kalitesi ve uygun küme sayısı belirlenerek kümeleme işlemleri başarıyla tamamlanabilir (Maulik ve Bandyopadhyay, 2002). Literatürdeki yöntemler, dahili ve harici yöntemler olmak üzere iki grupta incelenebilir (Şenol ve Karacan, 2018). Dahili indeksler, veri setinin yapısal özelliklerine ve kümeleme algoritmasının sonuçlarına dayanarak kümeleme kalitesini değerlendirirler. Özellikle, küme içi benzerlikleri artırmayı, kümeler arası farklılıkları ise maksimize etmeyi hedeflerler. Harici yöntemlerde ise gerçek küme ya da sınıf etiketleri olmak üzere ek bilgilere ihtiyaç vardır.

Bu tezde, küme kalitesini değerlendirmek için aşağıdaki dahili indeksler kullanılmıştır.

Silhouette Katsayısı: Silhouette katsayısı yöntemi Rousseeuw (1987) tarafından bir veri setindeki küme sayısını belirlemek için önerilmiştir. Her veri noktasının bulunduğu kümeye ne kadar iyi uyduğunu ölçer. Burada, a_i, i. veri noktası ile aynı kümede yer alan veri noktaları arasındaki ortalama mesafe; b_i, i. veri noktası ile en yakın başka bir kümede yer alan veri noktaları arasındaki ortalama mesafedir. Buna göre, i. veri noktası (gözlem)

için S_i Silhouette katsayısı Denklem (2.19) ile hesaplanır. Bütün veri noktaları için Silhouette katsayısının ortalaması Denklem (2.20) ile hesaplanır. (Martinez ve Martinez, 2001).

$$S_i = \frac{(b_i - a_i)}{\max(a_i, b_i)} \tag{2.19}$$

$$\bar{S} = \frac{1}{\mathrm{n}} * \sum_{i=1}^{n} S_i \tag{2.20}$$

Silhouette katsayısı -1 ile 1 arasında değerler alabilir. Silhouette katsayısının 1'e yakın değerler alması, i. veri noktasının bulunduğu kümeye diğer kümelere kıyasla daha benzer olduğunu göstermektedir (Martinez ve Martinez, 2001; Tan vd., 2016). Literatürde kümeleme kalitesi açısından ortalama Silhouette katsayısının 0,5'ten büyük olması halinde veri setinin ayrıldığı küme sayısının yeterli olduğu, 0,2'den küçük olması halinde ise veri setinin ayrıldığı küme sayısının yeterli olmadığı kabul edilmektedir (Martinez ve Martinez, 2001; Kaufman ve Rousseeuw, 2009).

Calinski-Harabasz İndeksi: Varyans oranı kriteri olarak da bilinen Calinski-Harabasz indeksi, kümeler arası dağılımın (yayılım) küme içi dağılıma oranıyla ilişkilidir (Calinski ve Harabasz, 1974) Bu indeks yüksek değerlere sahip olduğunda, kümelerin birbirinden daha iyi ayrıldığı ve homojen içeriklere sahip olduğu anlamına gelir. Calinski-Harabasz indeksi Denklem seti (2.21)-(2.23) ile hesaplanır (Calinski ve Harabasz, 1974; Maulik ve Bandyopadhyay, 2002).

$$B(k) = \sum_{i=1}^{k} n_i * \|c_i - c\|^2 \tag{2.21}$$

$$W(k) = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - c_i\|^2 \tag{2.22}$$

$$ch = \frac{B(k) * (n-k)}{W(k) * (k-1)} \tag{2.23}$$

Bu denklem setinde $B(k)$ küme merkezlerinin birbirlerine olan uzaklıklarının toplamını, $W(k)$ küme içindeki veri noktalarının küme merkezlerine olan uzaklıklarının toplamını, k küme sayısını, C_i *i.* kümeyi, n_i *i.* kümedeki veri noktalarının sayısını, c_i *i.* kümenin merkezini, c tüm veri noktalarının genel merkezini, x veri noktasını, n toplam veri noktası

sayısını ve *ch* ise Calinski-Harabasz indeksini ifade etmektedir (Calinski ve Harabasz, 1974; Maulik ve Bandyopadhyay, 2002).

Calinski-Harabasz indeksi, küme sayısını ve küme içi yayılımı dikkate alırken, aynı zamanda küme merkezlerinin birbirlerine olan uzaklıklarını dikkate alarak homojenlik ve bölmeye yönelik bir ölçüt sunar. Bu nedenle, yüksek bir Calinski-Harabasz indeksi, daha iyi bir kümeleme sonucuna işaret eder (Maulik ve Bandyopadhyay, 2002).

Dunn İndeksi: Dunn indeksi, kümeleme algoritmalarının performansını değerlendirmek için kullanılan bir başka ölçüttür. Bu indeks, en küçük kümeler arası mesafenin (iki küme merkezi arasındaki en küçük mesafe) en büyük küme içi mesafeye (bir kümede iki nokta arasındaki en büyük mesafe) bölünmesiyle hesaplanır. Amacı, küme içi yoğunluğun maksimize edilmesi ve kümeler arası mesafelerin minimize edilmesidir. Bu durumda, yüksek Dunn indeksi değerleri homojen ve iyi ayrılmış kümelerin olduğunu gösterir (Ansari vd., 2015; Ncir vd., 2021). Dunn indeksi, Denklem seti (2.24)-(2.26) ile hesaplanır.

$$d(C_i, C_j) = \min_{x_i \in C_i,\, x_j \in C_j} d(x_i, x_j) \tag{2.24}$$

$$diameter(C_l) = \max_{x_{l_1}, x_{l_2} \in C_l} d(x_{l_1}, x_{l_2}) \tag{2.25}$$

$$D = \min_{1 \le i \le k} \left(\min_{\substack{i+1 \le j \le k; \\ j \ne i}} \left(\frac{d(C_i, C_j)}{\max_{1 \le l \le k} diameter(C_l)} \right) \right) \tag{2.26}$$

Burada $d(C_i, C_j)$, C_i ile C_j kümeleri arasındaki uzaklığı; $d(x_i, x_j)$, $x_i \in C_i$ ile $x_j \in C_j$ veri noktaları arasındaki uzaklığı, $diameter(C_l)$, C_l kümesinin çapını ve *D*, Dunn indeksini ifade etmektedir (Dunn, 1974; Ansari vd., 2015; Ncir vd., 2021).

Dunn indeksi kompakt ve iyi ayrılmış kümeleri belirler, ancak tek başına kümeleme doğrulama ölçütü olarak kullanıldığında bazı sınırlamalara sahiptir. Özellikle, veri setinin yapısı ve dağılımı gibi faktörler dikkate alınmalıdır. Ayrıca, büyük veri setleriyle çalışırken hesaplama maliyeti artabilir. Bu nedenle, Dunn indeksinin diğer kümeleme doğrulama ölçütleri ile birlikte değerlendirilmesi daha uygundur (Dunn, 1974).

2.4. Topluluk Öğrenme Yöntemleri

Topluluk öğrenme yöntemleri, verilerin eğitimini birden fazla öğrenme yönteminde gerçekleştirerek tahminlerin birleştirilmesi yoluyla daha yüksek doğruluğa ve daha yüksek performansa sahip tahminler sunmayı amaçlar (Zhou, 2012; Zhou ve Zhou, 2021). Bir uzmanın alacağı kararın hatalı olma riski, birden fazla uzmanın alacağı ortak kararın hatalı olma riskinden daha yüksektir. Bu kapsamda birden çok uzmanın ortak kararı, bir uzmanın kararından daha güvenilir ve daha yüksek doğruluğa sahip olmaktadır. Bu yaklaşım temelinde topluluk öğrenme yöntemi geliştirilmiştir. İstatistik biliminde, 1977 yılında Tukey'in veri setine ikili doğrusal regresyon modelini önermesi ile topluluk çalışmalarına adım atıldığı kabul edilmektedir. Gerçek anlamda ilk topluluk yaklaşımı ise Hansen ve Salamon (1990) tarafından sunulmuştur ve tahmin performansını geliştirmek üzere sinir ağlarından oluşan topluluk tabanlı bir yöntem önermişlerdir (Hansen ve Salamon, 1990; Ganaie vd., 2022).

Topluluk öğrenme yönteminin başarısı, yapısındaki farklı modellerin elde ettiği farklı hatalara ve tahminlere dayanmaktadır. Bu nedenle performansı arttırmak için modellerin hatalarının birbirlerinden farklı olması sağlanmalıdır (Polikar, 2012). Modellerin tahminlerinde farklılıkların elde edilmesi için eğitim veri setinin farklı alt kümeleri kullanılmaktadır. Topluluk öğrenme yönteminde model performansını etkileyen diğer bir husus ise uygun birleştirme yönteminin seçimidir. Birleştirme tekniklerine, işlem adımlarına ve eğitim veri seti için örneklem seçimine göre farklı topluluk öğrenme yöntemleri mevcuttur. Bunlar, torbalama (bagging), güçlendirme (boosting) ve yığınlama (stacking) yöntemleridir (Polikar, 2012; Zhou, 2012).

2.4.1. Torbalama

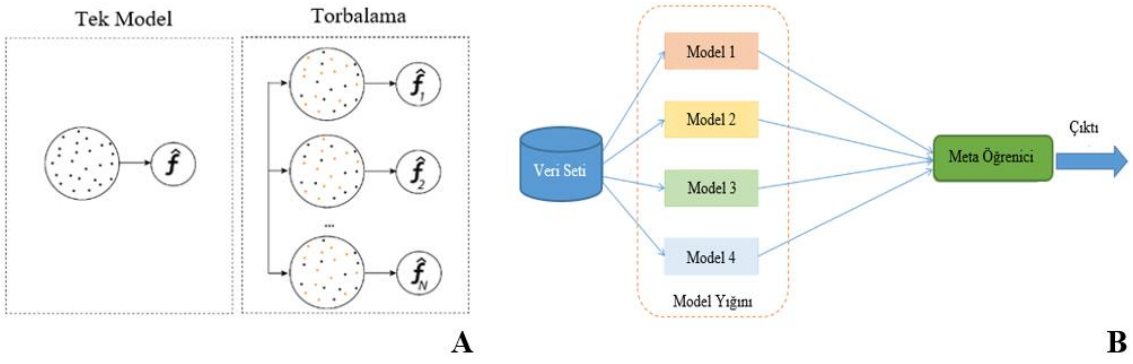
Torbalama (Bootstrap aggregating – Bagging), yeniden örnekleme (bootstrap) yöntemini esas almaktadır. Breiman tarafından geliştirilen yöntem eğitim veri setinin bootstrap örnekleme ile farklı alt kümelerinin oluşturulması ve eğitilmesi esasına dayanmaktadır (Breiman, 2001). Oluşturulan her alt eğitim seti bir temel model ile eğitilmektedir. Paralel olarak tüm modeller alt eğitim setlerini tahmin etmektedir. Modellerin tahminleri

sınıflandırma probleminde çoğunluk oyu tekniği ile regresyon problemlerinde ise ortalama tekniği kullanılarak birleştirilir ve hesaplanan bu tahmin, topluluk yönteminin tahmini olarak kabul edilmektedir.

Torbalama yönteminde farklı modellerden ziyade alt eğitim veri seti sayısı kadar temel model eş zamanlı olarak farklı alt veri setleri ile eğitilmektedir. Ardından her modelin tahminleri birleştirilmektedir. Farklı eğitim setlerinin seçilmesindeki amaç tahmin farklılıkları elde ederek tahmin başarısını yükseltmektir. Torbalama yöntemi Şekil 2.15 A'da gösterilmektedir.

2.4.2. Yığınlama

Wolpert tarafından geliştirilen yığınlama (stacking), farklı modellerin tahminlerini meta-öğrenici için girdi kabul ederek bu tahminlerden daha yüksek performanslı bir tahmin üretme esasına dayanmaktadır (Wolpert, 1992). Yığınlamada öğrenme süreci iki aşamalıdır. İlk aşamada, aynı eğitim veri setinden farklı modeller ile tahminler elde edilmektedir. İkinci aşamada ise ilk aşamadan elde edilen tahminler meta-öğrenicide eğitilerek nihai tahmin yapılmaktadır. Yığınlama yönteminde meta-öğrenici olarak sınıflandırmada lojistik regresyon, regresyon problemlerinde ise doğrusal regresyon (Linear Regression – LR), destek vektör regresyonu (Support Vector Regression – SVR), vb. yöntemler kullanılmaktadır (Ribeiro ve Coelho, 2020; Ribeiro vd. 2022). Yığınlama yöntemi Şekil 2.15 B'de gösterilmektedir (Wolpert, 1992).



Şekil 2.15. Topluluk öğrenme yöntemleri
A) Torbalama yöntemi (Polikar, 2012) **B)** Yığınlama yöntemi (Wolpert, 1992)

2.5. Kaynak Araştırması

Bu bölümde tez kapsamında gerçekleştirilen literatür taramalarına yer verilmektedir. Her alt bölümde sırasıyla derin öğrenme, transfer öğrenme, kümeleme ve topluluk öğrenme yöntemlerini kullanarak satış ve talep tahmini yapan çalışmalar anlatılmaktadır.

2.5.1. Derin öğrenme yöntemleri ile satış ve talep tahmini

Tez kapsamında öncelikle literatürdeki derin öğrenme yöntemleri ile satış ve talep tahmini yapan çalışmalar araştırılmıştır. Scopus, Elsevier, Science Direct, Web of Science, Taylor & Francis, Emerald, Hindawi, EBSCHO ve ULAKBİM veri tabanlarında aşağıda belirtilen anahtar kelimelerle tarama gerçekleştirilmiştir. Taramalarda "başlık, anahtar kelimeler, özet, makalenin tamamı" seçenekleri sırasıyla seçilerek tarama gerçekleştirilmiştir. Bu taramaya araştırma makaleleri, inceleme/tarama makaleleri ve konferans bildirileri dahil edilmiştir. İlgili alanda yapılan lisansüstü tezler ve kitap bölümleri kapsam dışında tutulmuştur. Aşağıda belirtilen anahtar kelimelerle literatür taraması yapılmıştır:

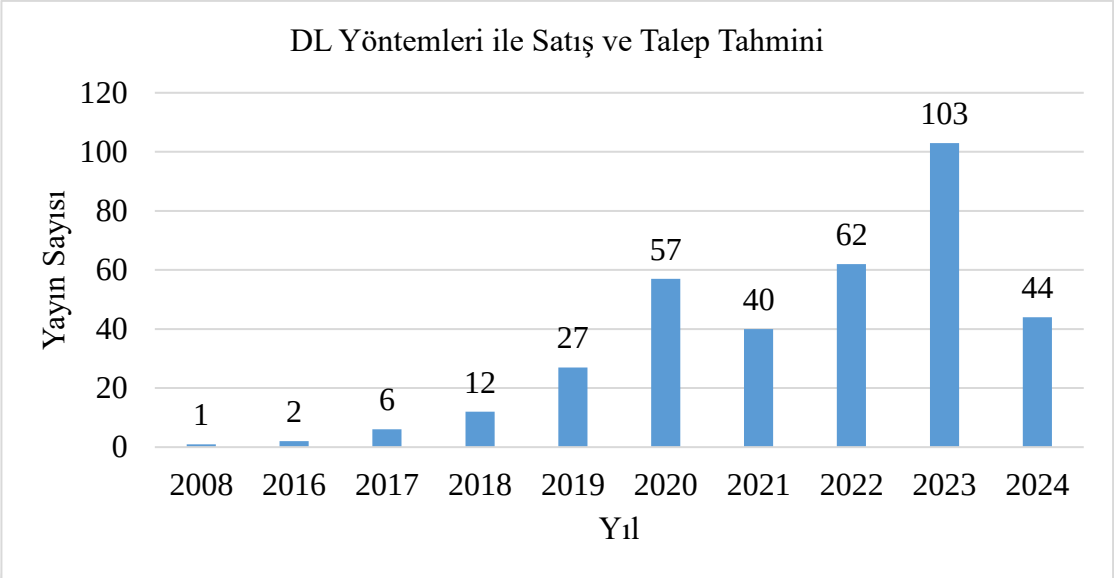
- "Deep Learning" AND ["Sales Forecasting" OR "Demand Forecasting"]
- "Deep Learning" AND ["Sales Forecasting" OR "Demand Forecasting"] AND ["Supply Chain (SC)" OR "Supply Chain Network" OR "Supply Chain Management (SCM)"]
- "Deep Learning" AND ["Sales Forecasting" OR "Demand Forecasting"] AND "Supply Chain Management (SCM)" AND ["Survey" OR "Review"]
- "Deep Learning" AND "Supply Chain Management (SCM)" AND ["Survey" OR "Review"]
- Yukarıdaki tüm aramalarda "Forecasting" yerine "Prediction" ve "Estimation" kelimeleri yazılarak da aynı tür aramalar gerçekleştirilmiştir.

Literatür taraması sonucunda yıllara göre yayın sayılarının dağılımı Şekil 2.16'da gösterilmektedir. Bu alanda derin öğrenme yöntemlerinin kullanımı 2008 yılında

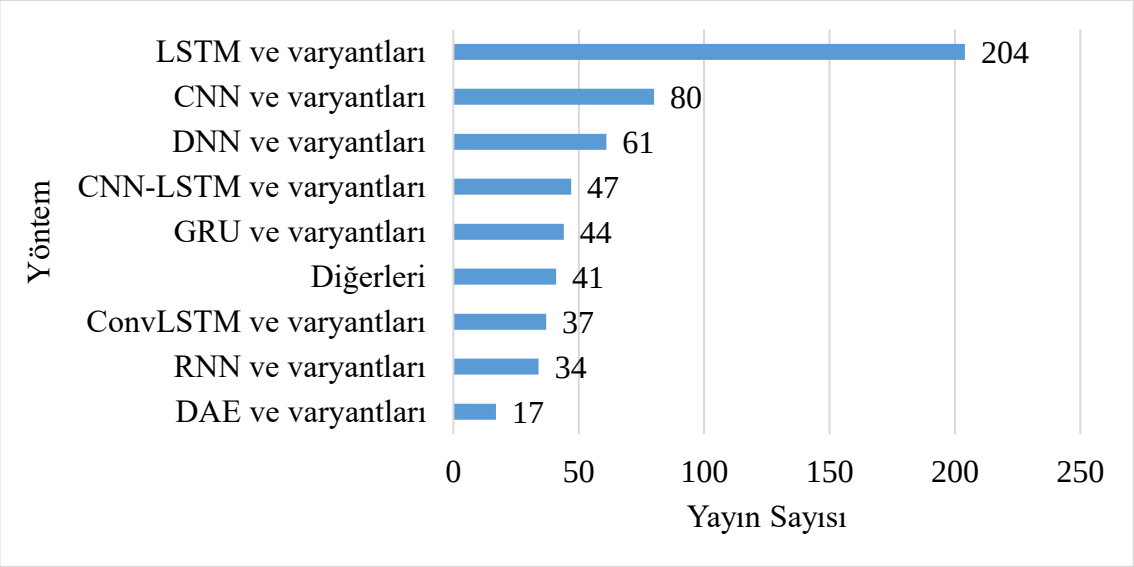
başlamış ve gün geçtikçe kullanımı artmıştır. Literatür taraması 21 Mart 2024 tarihine kadar olan çalışmaları kapsamaktadır.

Derin öğrenme yöntemleri ile satış ve talep tahmini yapan çalışmalarda kullanılan derin öğrenme yöntemlerinin dağılımları Şekil 2.17'de verilmektedir. Bir çalışmada derin öğrenme yöntemi ile birlikte çeşitli mekanizmalar (dikkat, gömme, vb.) kullanıldıysa ya da bu yöntem üzerinde iyileşme yapıldıysa, bu çalışma ilgili yöntemin varyantı olarak ele alınmıştır. Diğerleri ile belirtilen yöntemler ise temel derin öğrenme yöntemlerinin dışında geliştirilerek literatüre kazandırılan yeni yöntemleri (DeepAR, DeepState, TADA, vb.) ifade etmektedir. Ayrıca birden fazla yöntemin uygulandığı çalışmalarda, her yöntem için ilgili çalışma sayılmıştır. Şekil 2.17'ye göre satış ve talep tahmini için en çok tercih edilen yöntem LSTM ve varyantlarıdır. İkinci sırada CNN ve varyantları, üçüncü sırada ise DNN ve varyantları yer almaktadır. Ayrıca, satış verilerinin tek değişkenli veya çok değişkenli zaman serileri olması ve tahmin ufku uzunluğunun tek adımlı veya çok adımlı olmasına göre de istatistikler tutulmuştur. Buna göre taranan 354 makaleden 249'u tek değişkenli, 97'si çok değişkenli ve 8'i ise hem tek değişkenli hem çok değişkenlidir. Tahmin ufku uzunluğuna göre ise 354 makaleden 255'i tek adımlı, 79 tanesi çok adımlı ve 20 tanesi ise hem tek hem de çok adımlı tahmin gerçekleştirmiştir.

Literatürdeki çalışmalar kullanılan yönteme göre üç ana başlık altında sınıflandırılmıştır: (i) Mevcut derin öğrenme yöntemlerini kullanan çalışmalar, (ii) teknik iyileştirme veya geliştirme sağlayan çalışmalar, (iii) birleşik veya topluluk yöntemleri kullanan çalışmalar. Her bir sınıfa ait çalışmalar sırasıyla Ek 1, Ek 2 ve Ek 3'te verilmektedir. Taranan tüm çalışmalar incelendiğinde son yıllarda derin öğrenme yöntemlerini kullanarak satış ve talep tahmini yapan çalışmalarda artış yaşanmıştır. Ek olarak, son yıllarda temel derin öğrenme yöntemlerinin kullanımının azaldığı, teknik açıdan iyileştirilmiş veya geliştirilmiş yöntemlerin öne çıktığı görülmektedir (Ingle vd., 2021; Dharshini ve Vijila, 2021; Acı ve Doğansoy, 2022; Eglite ve Birzniece, 2022).



Şekil 2.16. Yıl bazında yayın sayıları



Şekil 2.17. Derin öğrenme yöntemlerine göre yayın sayılarının dağılımı

2.5.2. Transfer öğrenme ile satış ve talep tahmini

Literatürde satış ve talep tahmininde derin transfer öğrenme yöntemlerini kullanan çalışma sayısı sınırlıdır. Bu kapsamda ilk çalışma Zhao vd. (2017)'ne aittir. Bu çalışmada CNN kullanılarak yapılandırılmış verilerden etkili özellikleri öğrenen ve ardından bu bilgileri satış tahmini için kullanan yeni bir yaklaşım önerilmiştir. Ayrıca, tahmin doğruluğunu iyileştirmek için satış tahmini kapsamında, bir satış bölgesindeki öğrenilmiş

kalıpları başka bir satış bölgesine aktarma suretiyle TL uygulanmıştır. Bir e-ticaret firmasına ait veri kümesi alınmış, ilk olarak sinir ağı tüm satış bölgelerindeki eğitim örneklerini içeren tüm eğitim kümesi üzerinde eğitilmiş, ardından her bir satış bölgesine ait test kümeleri ile TL gerçekleştirilmiştir.

Pan ve Zhou (2020) etkin öznitelikleri çıkararak e-ticaret satış hacmini tahmin etmek için CNN uygulamışlardır. Satış tahminlerinin doğruluğunu artırmak için numune ağırlığı azaltma ve transfer öğrenme tekniklerini kullanmışlardır. Bir e-ticaret firmasına ait satış verileri ile yapılan deneylerle önerilen yöntemin TL yoluyla farklı satış bölgeleri arasındaki bilgi aktarımı ile satış tahmin doğruluğunun etkili bir şekilde iyileştirebileceğini göstermişlerdir.

Karb vd. (2020) yeni ürün satış tahmini için DL tabanlı bir TL yaklaşımı sunmuşlardır. Parametre transferi yapılmış ve TL modellerini eğitmek için üç farklı ağ yapısı tanımlanmıştır. Farklı ağ yapıları ile hedef alana hangi bilgilerin aktarılabildiği ve kaynak ile hedef alanlar arasındaki farklılıkların tahmin performansını nasıl etkilediği araştırılmıştır. Ek bilgiye sahip olma, eğitimde yeterli büyüklükte veriye sahip olma veya olmama vb. durumlarda farklı ağ konfigürasyonlarının seçilmesi gerektiği sonucuna varılmıştır. Bu çalışmada ayrıca fiyatların benzerliğine dayalı olarak alt gruplar belirlenmiş ve topluluk stratejisi uygulanmıştır. Aynı fiyat kümesinin alanların benzerliği için iyi bir gösterge olduğu, ancak kaynak modellerin seçim kriterlerinin sadece fiyatla sınırlandırılmasının yetersiz olduğu sonucu çıkarılmıştır. Optimal model olarak tek bir kaynak modelinin seçilemeyeceği belirtilmiştir. TL ile yetersiz eğitim verisi ve dönemsel talep artışının olduğu durumlarda tahmin performansının önemli ölçüde iyileştirilebileceği gösterilmiştir.

Literatürde bilindiği kadarıyla satış tahmini alanında benzerlik analizini dahil ederek TL uygulaması yapan bir adet çalışma bulunmaktadır. Hirt vd. (2020)'na ait olan bu çalışmada satış tahmini alanında farklı restoranların satış verileri arasında çok adımlı tahminde transfer öğrenmenin faydasını araştırmayı ve geliştirmeyi hedeflemişlerdir. Her sinir ağı aktarımı başarılı olmadığından, hedef performans üzerindeki etkisini (aktarılabilirliği) ölçmek için veri, görev ve sinir ağı benzerliklerine dayalı olarak

potansiyel göstergeleri hesaplamışlardır. Veri benzerliği için temel bileşen analizi ile çok boyutlu ölçeklendirme yöntemlerini, sinir ağı benzerliği için tekil değer kanonik korelasyon analizini kullanmışlardır. Aktarılabilirlik ve test edilen göstergeler arasında önemli negatif korelasyonlar elde edilmiştir. Aynı anda daha az model aktarımı gerektiren ve model performansını artıran bu göstergelere dayalı aktarım yolunun seçilebileceği bulunmuştur.

Literatür taraması sonucu elde edilen bu çalışmalarda genellikle kaynak ile hedef alan ve görevleri aynıdır. Hem kaynak hem de hedef görevler satış tahminidir. Tek kaynaklı veride parametre transferi yapılarak derin transfer öğrenme modelleri geliştirilmiştir. Bir başka ifadeyle çalışmalarda yazarlar ellerinde bulunan verileri belli oranlarda kaynak ve hedef olarak ikiye ayırmaktadır. Zhao vd. (2017) ile Pan ve Zhou (2020) bir firmaya ait bölgesel satışları, Karb vd. (2020) bir gıda firmasına ait satışları, Hirt vd. (2020) ise farklı restoranların satış verileri olmak üzere aynı sektörden farklı ürün veya hizmetlere yönelik satış verileri alarak TL uygulamışlardır. Ek olarak aktarılabilirlik kapsamında farklı ağ mimarileri oluşturmuşlardır.

Bu tezde, literatürdeki çalışmalardan farklı olarak farklı sektörlerdeki firmalara ait ürün ve hizmet satış verileri ile TL yapılmaktadır. Böylece, farklı sektörler arasındaki bilgi aktarımının sonuçları ele alınarak tartışılmaktadır. Öte yandan, literatürde satış tahmini alanındaki bu çalışmalar içinde sadece Hirt vd. (2020)'na ait çalışmada TL için aktarılabilirlik konusu ve buna bağlı olarak kaynak-hedef alan benzerliği konusuna odaklanılmıştır. Bu tezde, TL'de aktarılabilirlik için kaynak ile hedef veri benzerliğine dayalı bir seçim gerçekleştirilmektedir.

Literatürde satış tahmini dışında TL çalışmalarında kaynak-hedef alan benzerliğine odaklanan çalışmalar mevcuttur. Bu çalışmalardan biri Ye ve Dai (2021)'e aittir. Ye ve Dai (2021) yetersiz sayıda etiketli eğitim verisinin olduğu durumu ele alarak zaman serisi tahmini için CNN mimarisine sahip bir TL çerçevesi sunmuştur. Farklı veri kümelerinde TL sürecini uygulayarak mevcut etiketli veri yokluğunu etkili bir şekilde azaltmak ve mevcut tahmin için faydalı bilgileri aktarmak amaçlanmıştır. DTW ve Jensen-Shannon yakınsamasının bir birleşimi kullanılarak hedef ve kaynak etki alanları arasındaki

benzerlik dereceleri elde edilmiş ve bu uygun kaynak alanının seçiminde kullanılmıştır. Çalışma kapsamında ele alınan zaman serilerinin boyutu eşittir ve tek kaynaklı TL uygulanmıştır. TL'de kaynak seçimi ve aktarılabilirlik konusu için veri, görev ve sinir ağı benzerliklerine odaklanılmıştır.

Meiseles ve Rokach (2020)'a ait diğer bir çalışmada ise zaman serisi sınıflandırma problemi için hedef verisine uygun kaynak verisi seçmek yerine uygun bir kaynak model seçimi ele alınmıştır. Gizlilik veya telif hakkı kısıtlamaları nedeniyle eğitim verisi erişilebilir olmadığında, kaynak modeli seçimi problemine odaklanılmıştır. Hedef veriler için kaynak modellerin hedef verileri ne kadar iyi kümelediğini Silhouette katsayısı ile hesaplayarak bir kaynak model seçimi yapılmıştır. Bu çalışmada TL'de kaynak seçimi için kümelemeye dayalı bir seçim gerçekleştirilmiştir.

Jiang (2020)'a ait çalışmada ise zaman serisi sınıflandırma problemi için en yakın komşu ve DL modelleri arasında kapsamlı bir karşılaştırma yapılmıştır. Bu çalışmada, zaman serileri arasındaki benzerlikler için elastik, elastik olmayan ve eşik tabanlı olmak üzere sekiz farklı uzaklık ölçütü kullanılmıştır. Üç DL modeli ile bir boyutlu sınıflandırıcılar karşılaştırılmıştır. DL modellerinin gerçek ceza ve dinamik zaman bükmeli düzenleme uzaklığına sahip bir boyutlu sınıflandırıcılardan anlamlı derecede farklılaşmadığı gösterilmiştir.

Literatürde satış tahmini dışında TL çalışmalarında kaynak-hedef alan benzerliğine odaklanan bu çalışmalarda uzaklığa dayalı benzerlik ölçütleri ve kümeleme yöntemleri kullanılmıştır. Bu tez çalışmasında, literatürden farklı olarak TL'de kaynak seçimi için hem benzerlik ölçütleri hem de kümeleme yöntemleri entegre bir şekilde ele alınmıştır.

Bu tezde ele alınan bir diğer konu açıklanabilirliktir, bu nedenle bu konu üzerine yapılan çalışmalar da araştırılmıştır. Bu kapsamda, öncelikle zaman serisi tahmini çalışmalarında yapılan çalışmalar incelenmiştir. Tek veya çok değişkenli zaman serisi problemlerinin (sınıflandırma, kümeleme, vb.) çözümü birçok gerçek dünya uygulamasında kritik bir görevdir. Literatürde yer alan birçok yöntem etkinliği, verimliliği ve ölçeklenebilirliği en

üst düzeye çıkarmayı amaçlamaktadır, ancak sonuçların açıklanabilirliğini sağlayamamaktadır (Bonifati vd., 2022).

Zaman serisi problemlerinin çözümünde açıklanabilirliği ve yorumlanabilirliği sağlayacak özniteliklerin çıkarılması önem arz etmektedir. Bu kapsamda, Henderson ve Fulcher (2021) zaman serisi öznitelik çıkarımı kütüphanelerini hesaplama hızlarına, her kütüphanede bulunan özniteliklerin miktarına ve farklı kütüphanelerden elde edilen öznitelik kümelerindeki benzerliklere göre karşılaştırarak değerlendirmişlerdir. Bu kapsamda hctsa (7730 öznitelik: Matlab) (Fulcher ve Jones, 2017), feasts (42 öznitelik: R) (O'Hara-Wild vd., 2021), tsfeatures (63 öznitelik: R) (Hyndman vd., 2020), Kats (40 öznitelik: Python) (Facebook, 2021), tsfresh (1558 öznitelik: Python) (Christ vd., 2018), TSFEL (390 öznitelik: Python) (Barandas vd., 2020) ve catch22 (22 öznitelik, Matlab, R, Python, and Julia) (Lubba vd., 2019; Henderson, 2021; Harris, 2021) olmak üzere yedi farklı zaman serisi öznitelik çıkarımı kütüphanesini kullanmışlardır. En büyük öznitelik kümesine sahip olan hctsa'nın en kapsamlı kütüphane olduğunu, diğer kümelerde daha yüksek seviyelerde özetlenen çok sayıda Fourier katsayısını bir araya getirmesi nedeniyle tsfresh'in en ayırt edici kütüphane olduğunu tespit etmişlerdir (Henderson ve Fulcher, 2021).

Bonifati vd. (2022) çok değişkenli zaman serisi kümeleme problemi için öznitelik çıkarımı ve seçimine dayalı makine öğrenmesi tabanlı yeni bir yöntem geliştirmişlerdir. Yöntem, zaman serilerinden çıkarılan sinyaller arası yorumlanabilir özniteliklere dayanır. Yöntemin etkinliği ve yorumlanabilirliği çeşitli deneylerle gösterilmiştir.

Literatürde satış ve talep tahmini alanında derin transfer öğrenmede açıklanabilirlik konusunu ele alan bir çalışma bilindiği kadarıyla yoktur.

2.5.3. Kümeleme yöntemleri ile satış ve talep tahmini

Satış ve talep tahmini çalışmalarında tahmin performansını iyileştirmek için kümeleme yöntemleri kullanılmaktadır. Buradaki amaç, benzer özellikler gösteren zaman serilerini kümeleyerek anlamlı örüntülerin çıkarılmasını sağlamak ve zaman serileri arasındaki çapraz ilişkileri ortaya çıkarabilmektir.

Literatürde DL yöntemleri ile satış ve talep tahmini yapan çalışmalar içerisinde kümeleme analizi yapan çalışmalar bulunmaktadır. Bandara vd. (2019) ürün satışlarının seyrek olduğu durumlarda güçlü bir tahmin yapabilmek için çeşitli ürün gruplandırma stratejileri ile e-ticaret ürün çeşitleri arasındaki mevcut doğrusal olmayan ilişkilerin kullanılmasına dayalı bir tahmin çerçevesi önermişlerdir. K-ortalamalar ile kümeleme, satış sıralaması ve sıfır satış yüzdesine göre ürün gruplandırma stratejilerini kullanarak LSTM ağı ile tahmin gerçekleştirmişlerdir. Bu gruplandırma stratejilerinin bir ürün portföyündeki satış modellerinin farklı olduğu durumlarda LSTM eğitim süreçlerini iyileştirdiğini belirtmişlerdir.

Yin vd. (2020), yeni ürünlerin talep tahmini için kümeleme algoritması ile DL yöntemlerini birleştirmişlerdir. Bulanık kümeleme yöntemi ile benzer ürünleri kümelemişlerdir ve bu yolla yeni ürün satış tahmininde daha yüksek doğruluk elde edildiğini göstermişlerdir. Kümeleme algoritması tarafından tanımlanan zaman serisi gruplarının sayısı modelin performansını etkilediği için optimal küme sayısına ulaşmanın zor ancak önemli bir faktör olduğunu belirtmişlerdir. Zaman serilerinin kümelenmesi sonucunda bilgi kaybı olabileceğini ifade etmişlerdir.

Diğer bir çalışma Bandara vd. (2020)'na aittir. Bandara vd. (2020) zaman serilerini kümeleyerek ve her küme için ayrı modeller oluşturarak bir dizi zaman serisindeki çapraz seriler bilgisini kullanan LSTM tabanlı bir tahmin çerçevesi önermişlerdir. K-ortalamalar, Gürültülü Uygulamaları Yoğunluğa Dayalı Konumsal Kümeleme (Density Based Spatial Clustering of Applications with Noise - DBSCAN) ve Medoidlerin Etrafında Bölümleme (Partitioning Around Medoids - PAM) olmak üzere üç farklı kümeleme yöntemi kullanmışlardır. Zaman serilerindeki en uygun küme sayısını belirlemek için bu kümeleme tekniklerinden yararlanmışlardır ve büyük veriye dikkat çekmişlerdir. Kümeleme sürecinin, eğitilebilir zaman serilerinin homojenliğini ve paralelliğini geliştirdiğini belirtmişlerdir.

Wang (2021) turizm sektöründe talep tahmini için k-ortalamalar kümeleme ve LSTM tabanlı bir tahmin çerçevesi önermişlerdir. Seyrek verilerden kaynaklanan hatalı tahminleri önlemek amacıyla birden çok otelin verilerini kümeleyerek konuk türlerine

göre (eğlence ve iş amaçlı) misafir akışını tahmin etmişlerdir. Tek bir otel verisi ile yapılan tahmin, otel gruplarının tahminleri ile karşılaştırıldığında; seyrek verilerin neden olduğu yanlışlıkların etkili bir şekilde önlenebildiği ve LSTM'nin tahmin performansını artırabildiği gösterilmiştir.

Kaya vd. (2022) haftalık otel talebini dört hafta önceden tahmin edebilen ve dikkat ağı içeren LSTM ağı tabanlı talep tahmin modeli sunmuşlardır. Öncelikle sinir ağlarını kullanarak zaman serisi talep verilerinden özellikler çıkarmışlardır ve bu özelliklere göre benzer kriterleri karşılayan otelleri kümelemek için k-ortalamalar ile kümeleme yapmışlardır. Özellik çıkarma ve kümelemeden elde edilen bilgilerle yapılan tahminin dikkat ağı ile birlikte tahmin hatasını azalttığı belirtilmiştir.

Literatürde derin transfer öğrenme ile satış ve talep tahmini yapan çalışmalar içerisinde kümeleme analizi yapan bir çalışmaya ise rastlanılmamıştır.

Bu tez çalışmasında geliştirilen çözüm yaklaşımında derin otokodlayıcı tabanlı çok aşamalı derin kümeleme yöntemi kullanılmıştır. Çok aşamalı derin kümeleme, iki modülün ayrı ayrı optimize edildiği ve sırayla bağlandığı yöntemleri ifade eder. Burada, öncelikle veri örneklerinin temsillerini öğrenmek (yerleştirme) için derin temsil öğrenme teknikleri (derin sinir ağı, derin otokodlayıcı, vb.) kullanılır, sonrasında nihai kümeleme sonuçlarını elde etmek için öğrenilen temsilleri klasik kümeleme modellerine (k-ortalamalar, hiyerarşik kümeleme, vb.) beslenir. Veri işleme ve kümelemenin bu şekilde ayrılması, araştırmacıların kümeleme analizi yapmasını kolaylaştırır.

Bu tez çalışmasında, derin otokodlayıcı ve k-ortalamalardan oluşan çok aşamalı derin kümeleme yöntemi kullanılmıştır. Literatürde satış ve talep tahmini alanında bu yöntemi kullanan bir çalışmaya rastlanmamıştır. Derin kümeleme alanındaki çeşitli derleme çalışmaları incelendiğinde görüntü, metin, video ve grafik verilerinin kümelenmesinde derin kümeleme yöntemlerinin sıklıkla kullanıldığı görülmektedir (Nutakki vd., 2019; Zhou vd., 2022; Ren vd. 2022). Bu nedenle, zaman serilerinde derin kümeleme yapan çalışmalar özetlenmiştir.

Alqahtani vd. (2021) derin zaman serisi kümeleme alanında kapsamlı bir inceleme çalışması sunmuşlardır. Ek olarak, derin konvolüsyonel otokodlayıcı kümeleme yöntemi ile hayvan davranışı (hareketi) kümeleme vaka çalışması sunmuşlardır.

Yang vd. (2022) ise fotovoltaik güç tahminini gerçekleştirmek için ikili kümeleme ve evrişimli sinir ağından oluşan bileşik bir tahmin çerçevesi önermişlerdir. Fotovoltaik gücünün zamansal belirsizliğini ve sayısal hava tahmininin yüksek boyutlu karmaşıklığını önlemek için ham veriler, veri fazlalığını azaltmak amacıyla sırasıyla ayrık dalgacık dönüşümü ve derin otomatik kodlayıcı tarafından işlenir. İkinci olarak, güç eğrisinin dinamik özelliklerini ve hava durumu bilgisi verilerinin sayısal kümelenmesini aşamalı olarak gerçekleştirmek için dinamik zaman bükmesi mesafesi ve bulanık ortalama kümelemeyi temel alan ikili bir kümeleme modeli önerilmektedir. Bu çalışmada, derin otokodlayıcı ve bulanık ortalama kümeleme yöntemleri birleştirilerek çok aşamalı derin kümeleme uygulanmıştır. Bu ikili kümeleme modelinin diğer geleneksel kümeleme modellerinden daha iyi performansa sahip olduğu gösterilmiştir.

Lai vd. (2021) güneş radyasyonu tahmini için derin öğrenme tabanlı hibrit bir yöntem önerilmektedir. Düzensiz modelleri daha iyi tanımlamak ve böylece daha iyi bir kümeleme performansı sağlamak için derin zaman serisi kümeleme kullanılmıştır. Kümeleme sonrası tahminleri oluşturmak üzere her küme için özellik-dikkatli derin sinir ağı oluşturulmuştur. Önerilen yöntem dinamik olarak farklı özelliklere farklı önemler vermektedir ve bir sonraki saatteki radyasyonu tahmin etmek için ağırlıklı özelliklerden yararlanmaktadır.

Bu tez kapsamında, satış ve talep tahmininde derin kümelemeye dayalı derin transfer öğrenme metodolojisi geliştirilmiştir. Bilindiği kadarıyla literatürde derin transfer öğrenme ile satış ve talep tahmini yapan çalışmalar içerisinde kümeleme analizi yapan bir çalışma bulunmamaktadır. Ayrıca derin öğrenme ile satış ve talep tahmini yapan çalışmalar içerisinde derin kümeleme kullanan bir çalışmaya rastlanmamıştır. Bu tez, bu yönleriyle literatüre katkı sağlamaktadır.

2.5.4. Topluluk öğrenme yöntemleri ile satış ve talep tahmini

Literatürde DL yöntemleri ile topluluk öğrenme yöntemini kullanan çalışmalar sınırlıdır. Bu çalışmalardan biri Kilimci vd. (2019)'ne aittir. Kilimci vd. (2019) DNN, SVR algoritması ve farklı zaman serisi yöntemlerini birleştiren topluluk tabanlı bir talep tahmini yöntemi geliştirmişlerdir. En iyi, ortalama ve ağırlıklı ortalama olmak üzere üç farklı topluluk stratejisi belirleyerek torbalama uygulamışlardır. Önerilen yöntem Türkiye'deki bir perakende şirketinde uygulanarak tedarik zincirindeki talep tahmini performansını iyileştirdiği gösterilmiştir.

Arora vd. (2020) distribütör kuruluşların talep tahmini problemine odaklanmıştır. Bu amaçla, tek değişkenli ve çok değişkenli zaman serileri modelleri ile LSTM ağını birleştiren bir topluluk öğrenme yöntemi sunmuşlardır. Ortalama ve ağırlıklı ortalama olmak üzere iki farklı topluluk stratejisi belirleyerek torbalama uygulamışlardır.

Ma vd. (2021) perakende satış tahmini için yeni geliştirilen derin evrişimli sinir ağlarına dayalı bir meta-öğrenme çerçevesi sunmuşlardır. Yedi farklı meta-öğrenici (Üstel Düzeltme, ARIMAX, GBRT, SVR, RF, ELM, CNN) ve üç topluluk yöntemi (torbalama, yığınlama, güçlendirme) ile karşılaştırmalı analizler gerçekleştirmişlerdir.

Okwuchi vd. (2020) çilek verimini ve akabinde çilek satış fiyatını tahmin etmek için dikkat ağı içeren CNN-LSTM ve Conv-LSTM modelleri ile LR ve SVR tabanlı yığın topluluk yöntemleri önermişlerdir. SVR tabanlı yığınlama modelleri ile ConvLSTM modellerinin daha başarılı sonuçlar verdiği belirtilmiştir.

Literatürde satış ve talep tahmini alanında TL tabanlı topluluk yöntemi uygulayan bir çalışmaya rastlanmamıştır. Bu nedenle, farklı alanlarda derin TL yöntemleriyle topluluk modeli kullanan çalışmalar incelenmiştir. Literatür taraması sonucunda bu alanda sekiz adet makale elde edilmiştir. Bu çalışmalardan biri Sun vd. (2018)'ne aittir. Sun vd. (2018) finansal zaman serilerini tahmin etmek için AdaBoost algoritmasını ve LSTM ağını birleştiren hibrit bir topluluk öğrenme yaklaşımı önermişlerdir. Günlük döviz kuru ve borsa endeksi veri setleri ile yapılan deneyler önerilen AdaBoost-LSTM topluluk

öğrenme yaklaşımının diğer tahmin modellerinden (ARIMA, SVR, MLP, LSTM) ve diğer topluluk öğrenme yaklaşımlarından (AdaBoost-MLP, AdaBoost-SVR) daha iyi performansa sahip olduğu gösterilmiştir. AdaBoost-LSTM yaklaşımının, özellikle finansal zaman serileri gibi doğrusal olmayan ve düzensizliğe sahip zaman serisi verilerinin tahmini için oldukça umut verici bir yaklaşım olduğu belirtilmiştir.

Xiao vd. (2019), günlük deniz yüzey sıcaklığını tahmin etmek için LSTM-AdaBoost birleşik yöntemini önermişlerdir. İlk olarak, LSTM ve AdaBoost modellerini zaman serileri ile bağımsız olarak eğitmişlerdir. Ardından her iki modelin tahmin sonucunun ortalamasını alarak nihai tahminleri oluşturmuşlardır. Buradaki amaç, iki güçlü ve heterojen modelin birleştirilmesiyle tahmin hatalarını azaltmak ve nihai sonuçları iyileştirebilmektir. Ortalama alma stratejisini kullanan LSTM-AdaBoost modelinin kısa ve orta vadeli günlük deniz yüzey sıcaklığı tahminlerinde SVR, ANN, LSTM ve AdaBoost yöntemlerinden daha iyi performans gösterdiği gözlenmiştir.

Busari vd. (2021) finansal zaman serisi tahmini için AdaBoost-LSTM ve AdaBoost-GRU birleşik modellerini önermişlerdir. Hisse senedi fiyatı, petrol fiyatı ve dolar borsa verileri ile yapılan deneylerle AdaBoost-GRU ile en iyi tahmin sonuçlarına ulaşıldığı ve topluluk modellerinin (AdaBoost-GRU, AdaBoost-LSTM) tekli modellerden (GRU, LSTM) daha iyi performansa sahip olduğu gösterilmiştir.

Busari ve Lim (2021), ham petrol fiyatını tahmin etmek ve tahmin performansını iyileştirmek için AdaBoost-LSTM ve AdaBoost-GRU modellerini sunmuşlardır. LSTM/GRU modelini sklearn içine yerleştirerek AdaBoost Regressor ile AdaBoost-LSTM/GRU modelini oluşturmuşlardır. Sekiz aylık günlük ham petrol fiyatları kullanılarak yapılan deneylerle önerilen AdaBoost-GRU modelinin LSTM, GRU ve AdaBoost-LSTM modellerine kıyasla daha iyi olduğu gösterilmiştir.

Bai vd. (2021), üretim kalitesinin tahmini için kaba küme teorisini kullanan bir topluluk öğrenme çerçevesi (AdaBoost-LSTM modeli) önermişlerdir. Öncelikle farklı üretim aşamalarındaki önemli parametreler, öznitelik indirgeme ve kaba küme teorisine dayalı olarak seçilmiştir. Seçilen önemli parametreler kullanılarak kaba bilginin varlığında

LSTM ile üretim kalitesi tahmin edilmiştir ve LSTM'nin öğrenme performansı AdaBoost yaklaşımıyla güçlendirilmiştir. Önerilen modelin SVR, ANN ve LSTM yöntemlerinden daha iyi performansa sahip olduğu ve ayrıca bu modellerden istatistiksel olarak anlamlı bir farklılığa sahip olduğu gösterilmiştir. Kaba bilgiyi eşzamanlı olarak bütünleştiren topluluk öğrenme stratejisinin, modelin girdi değişkenlerine duyarlılığını ve tahmin performansını geliştirdiği belirtilmiştir.

Zhang vd. (2021), kısa vadeli rüzgâr enerjisi tahmini için AdaBoost-LSTM modelini sunmuşlardır. Rüzgâr gücü çıktısının potansiyel özelliklerini ve ilgili faktörlerini etkin bir şekilde araştırmak için zaman serisi ayrıştırma ve yeniden yapılandırma tekniklerini kullanmışlardır. Orijinal zaman serisi, farklı spektrum özelliklerini yansıtabilen alt serilere (dizi) ayrıştırılmıştır. LSTM ile ayrıştırılan her bir alt dizide belirlenen etki faktörleri (meteorolojik veriler, vb.) dahil edilerek tahmin gerçekleştirilmiştir ve Adaboost-LSTM'yi optimize etmek için kullanılmıştır. Nihai tahmin sonucu, tüm alt dizilerin tahmin sonuçlarının entegre edilmesiyle elde edilmiştir. Deneysel çalışmalarla LSTM'nin en iyi temel tahmin modeli olduğu ve ayrıştırma tekniği ile Adaboost'un LSTM'nin performansını iyileştirdiği gösterilmiştir. Ayrıca ayrıştırma tekniğinin iyileştirmeye olan katkısının Adaboost'tan daha fazla olduğu ve her ikisinin birlikte kullanımı ile daha fazla iyileştirme sağlanabildiği vurgulanmıştır. Ek olarak, LSTM'yi geliştirmek için Adaboost'un kullanılması ile tahmin doğruluğunun arttırıldığı ancak gradyan patlaması ve gradyan kaybolması problemlerini tamamen çözmediği belirtilmiştir.

Wang vd. (2021) hidroelektrik santral akışlarının tahmini için torbalama algoritmasına dayalı LSTM modelini (Bagging-LSTM) önermişlerdir. Önerilen modelin LSTM, SVR ve RF yöntemlerine göre daha yüksek doğruluk oranları sağladığı, özellikle tahmin adımı uzadıkça daha düşük hataya yol açtığı gösterilmiştir.

Dang vd. (2022) güneş lekesi sayısı üzerinden güneş aktivitesini tahmin etmek için dört DL modelini (LSTM, GRU, transformer, informer) birleştiren iki seviyeli doğrusal olmayan bir topluluk yöntemi olarak XGBoost-DL adını verdikleri modeli önermişlerdir. XGBoost-DL modeli üç geleneksel model (SARIMA, üstel düzeltme, Prophet), bileşeni

olan dört DL modeli ve NASA tahminleri ile karşılaştırılmıştır. XGBoost-DL'nin en iyi tahmin performansına sahip olduğu gösterilmiştir. XGBoost-DL modelinin üstün performansı, DL modelleri üzerine kurulu karar ağaçlarının oluşturduğu iki seviyeli doğrusal olmayan güçlü topluluk mimarisinden kaynaklandığı vurgulanmıştır. Ek olarak, geleneksel modellerle ve tüm modellerin tahmin sonuçlarına ayrı ayrı beş topluluk öğrenme yöntemi (ortalama, medyan, hata tabanlı yöntem, doğrusal regresyon ve XGBoost aracılığıyla) uygulanmıştır. DL modellerine dayalı topluluk modellerinin, geleneksel modellere veya tüm temel modellere dayalı olanlardan daha doğru tahminlere sahip olduğu gösterilmiştir.

Literatür taraması sonucu elde edilen bu çalışmalarda DL yöntemleri ile topluluk yöntemleri birleştirilmiştir. TL yöntemleri ile topluluk yöntemlerini birleştiren çalışmalar ise sınıflandırma problemi olarak görüntü işleme alanında yoğunlaşmaktadır. Zaman serisi tahmini gibi regresyon problemlerinde yapılan çalışma sayısı ise sınırlıdır. Bu çalışmalardan ilki, Ye ve Dai (2018) tarafından sunulmuştur. Ye ve Dai (2018), zaman serisi tahmininde yeterli eğitim ve test verisinin mevcut olmadığı durumlarda bilginin uzun bir zaman aralığında nasıl aktarılacağı problemini çözmek için transfer öğrenme, Aşırı Öğrenme Makinesi (Extreme Learning Machine – ELM) ve topluluk öğrenmeye dayalı hibrit bir algoritma önermişlerdir. ELM modeline topluluk öğrenme şeması dahil edilmiştir. Topluluk öğrenme şemasında kurucu modellerin ağırlıkları, yeni örnekler üzerindeki performanslarına göre uyarlanabilir bir şekilde güncellenmektedir. Önerilen algoritma, önceki dönemlerdeki gizli bilgiyi etkin bir şekilde kullanabilir. Ayrıca az sayıda yeni eğitim örneği mevcut olmasına rağmen, etkili modeller üretmeyi amaçlamışlardır. Üç sentetik ve altı gerçek veri seti üzerindeki deneysel sonuçlar ile önerilen algoritmanın etkinliğini gösterilmiştir.

Shen vd. (2020) lityum iyon (Li-ion) şarj edilebilir pillerin kapasite tahmininde transfer öğrenme ve topluluk öğrenme kavramlarını içeren derin öğrenmeye dayalı bir yöntem sunmuşlardır. Bu çalışmada, küçük bir eğitim veri setinin mevcut olduğu uygulamalar hedeflenmiştir. İlk olarak, sekiz Li-ion hücresinden alınan 10 yıllık günlük şarj/deşarj döngüsü verileri kaynak veri seti olarak kullanılarak sekiz farklı derin evrişimli sinir ağı (Deep CNN – DCNN) modeli eğitilmiştir. Daha sonra, ön eğitilmiş DCNN modellerinin

öğrenilen parametreleri TL yoluyla kaynak görevden hedef göreve aktarılarak DCNN-TL modelleri elde edilmiştir. Son adımda, sekiz DCNN-TL modeliyle bir topluluk modeli oluşturulmuştur. Bir hedef veri kümesi üzerinde yapılan deneylerle önerilen topluluk modelinin etkinliği gösterilmiştir. Hedef görevdeki Li-ion hücrelerinin kapasitelerini tahmin etmede diğer veriye dayalı yöntemlerden (RF, Gauss süreci) daha yüksek doğruluk ve sağlamlık üretebileceği gösterilmiştir.

Zhou vd. (2020) demiryolu işletmeciliğinde çeşitli nedenlerle (şiddetli hava koşulları, altyapı arızası, insan faktörleri vb.) meydana gelen tren gecikmelerini ve gecikmenin yayılımını tahmin etmek için transfer öğrenme, derin öğrenme ve topluluk öğrenmesine dayalı çok aşamalı akıllı bir yöntem önermişlerdir. İlk olarak, tek gizli katmanlı ileri beslemeli sinir ağı kullanılarak yığınlama topluluk modeli geliştirilmiştir. Ardından, önerilen topluluk modelinin bir gecikmeden sonraki ilişkili gecikmeleri tahmin edebilmesi için bir sınıflandırma çalışması yapılmıştır. Etiketlenmemiş ve dengesiz tren verilerini doğru sınıflandırarak ilk gecikmenin ilişkili gecikmelere yol açıp açmayacağına karar vermek için sinir ağına dayalı bir transfer öğrenme sınıflandırıcısı geliştirilmiştir. Sınıflandırma ve tahmin prosedürleri ardışık olarak çalıştırılarak, yayılma aralığı ve tren gecikmeleri zinciri elde edilebilmektedir. Farklı demiryolu hatlarından alınan gerçek tren verileri ile yapılan deneylerde 0,5 dakikalık kabul edilebilir hata oranında gecikme değişiklikleri tahmininde %90'lara varan doğruluk oranları elde edildiği gösterilmiştir. Ayrıca, ilişkili gecikmelerin tahmin modellemesinde torbalama ve AdaBoost topluluk modellerinden daha iyi olduğu gösterilmiştir.

Bu tezde literatürdeki çalışmalardan farklı olarak satış tahmini alanında kümeleme ve topluluk yöntemine dayalı bir tahmin metodolojisi geliştirilmiştir. Bu kapsamda TL'de aktarılabilirlik için hedef veriye uygun kaynak verisi seçim kararı hem benzerlik ölçütleri hem de kümelemeye dayalı olarak ele alınmıştır. Ayrıca, transfer öğrenmede açıklanabilirlik kavramı üzerinde durulmuştur. Çalışmanın çok kaynaklı transfer öğrenmeye genişletilmesi için kümeleme yöntemi ele alınmıştır. Tek ve çok kaynaklı transfer öğrenme başarısını iyileştirmek için ise topluluk öğrenme yaklaşımı entegre edilmiştir.

3. MATERYAL ve YÖNTEM

Bu bölümde, tez kapsamında yapılan çalışmalarda kullanılan materyal ile geliştirilen çözüm yaklaşımları verilmektedir. Bölüm 3.1'de zaman serilerine ilişkin tanımlar verilmektedir ve bu tezde kullanılan veri kümeleri tanıtılmaktadır. Bölüm 3.2'de benzerlik ölçütleri, Bölüm 3.3'te derin otokodlayıcı tabanlı derin kümeleme yöntemi, Bölüm 3.4'te çalışmada kullanılan performans değerlendirme metrikleri verilmektedir. Bölüm 3.5'te tek değişkenli zaman serileri için önerilen çözüm yaklaşımları ve Bölüm 3.6'da ise çok değişkenli zaman serileri için önerilen çözüm yaklaşımları açıklanmaktadır.

3.1. Zaman Serileri ve Veri Kümeleri

Tek değişkenli zaman serisi, belirli bir zaman diliminde bir değişkenin değerlerinin zaman adımına göre kaydedildiği veri dizisidir. $Y_i = \{y_{it}: \ t \in T\}, i \in D$ tek değişkenli zaman serisi olsun. Burada, y_{it} T gözlem zamanı süresince *i.* zaman serisinin *t* zaman adımındaki gözlem değerini (satış miktarı) ve *D* farklı zaman serilerini içeren kümeyi ifade etmektedir.

Çok değişkenli zaman serisi, birden fazla değişkenin (özniteliğin) zaman içinde nasıl değiştiğini gösteren bir veri dizisidir. Her bir zaman adımında, değişken değerleri kaydedilir. $Y_i = \{y_{itf}: \ t \in T, f \in F\}, i \in D$ çok değişkenli zaman serisinde; y_{itf}, T gözlem zamanı süresince *i.* zaman serisinin *t* zaman adımındaki *f.* değişkenin (özniteliğin) gözlem değerini (hafta, ay, yıl, resmi tatil ve satış miktarı), *D* farklı zaman serilerini içeren kümeyi, *F* ise özniteliklerin kümesini ifade etmektedir.

Çok değişkenli bir zaman serisinin öznitelikleri kategorik ve sayısal olmak üzere iki türde incelenebilir. Tamsayılı ve aralık/oran ölçekli öznitelikler sayısal öznitelikler iken sınıfsal, ikili ve sıralı öznitelikler ise kategorik özniteliklerdir (Bramer, 2007). Özniteliklerin türüne göre yöntem seçimi yapılması oldukça önemlidir. Çıktı (hedef) değişkeni, kategorik değişken ise sınıflandırma yöntemleri, çıktı değişkeni sayısal değişken ise tahmin yöntemleri kullanılmaktadır (Han vd., 2012).

Bu tezde, ele alınan satış veri kümelerinin özellikleri Çizelge 3.1'de verilmektedir. Bu veri kümeleri çok değişkenli zaman serileridir ve çeşitli platformlardan (Kaggle, Github, vb.) elde edilmiştir. Veri kümelerine ait öznitelikler Çizelge 3.2'de verilmektedir. Bu tez çalışmasında, çıktı değişkeni satış miktarı olup sayısal değişkendir. Uygulamalarda tüm veri kümelerine ait satış değerleri haftalık olacak şekilde düzenlenmiştir. Tez kapsamında gerçekleştirilen çalışmalarda Çizelge 3.1'de yer alan veri kümeleri hem tek değişkenli hem de çok değişkenli olarak ele alınmıştır.

Çizelge 3.1. Veri kümeleri ve özellikleri

Veri Kümesi	Zaman Serisinin Uzunluğu	Verinin Özellikleri	Açıklama
Walmart	143	Mevsimsellik vardır.	Farklı bölgelerdeki 45 Walmart mağazasına ait 2010-2012 yılları arasındaki geçmiş haftalık satış verileri (Kaggle, 2020)
Supermarket	209	Mevsimsellik ile önce artma sonra azalma eğilimi vardır.	Amerika'nın farklı bölgelerindeki yerel bir süpermarkete ait 2014-2017 yılları arasındaki geçmiş haftalık satış verileri (Github, 2021)
Forklift	228	Çok sayıda sıfır içermektedir ve azalma eğilimi vardır.	Forklift üreticisine ait 2016-2020 yılları arasındaki günlük satış verileri (Puspita vd., 2018)
LearnX	882	Mevsimsellik vardır.	LearnX çevrimiçi öğrenme platformuna ait 600 kursun yıllık satış verileri (Kaggle, 2020)
Rossmann	942	Mevsimsellik vardır.	3000'den fazla Rossman mağazasına ait günlük satış verileri (Kaggle, 2020)
Tv_Sales	974	Artma eğilimi vardır.	Bangladeş'in günlük TV satış verileri (Kaggle, 2020)
Favorita	1688	Mevsimsellik vardır.	Favorita marketlerine ait haftalık satış verileri (Kaggle 2020)
Pharma	2106	Mevsimsellik ile önce azalma sonra artma eğilimi vardır.	Eczane günlük ilaç satış verileri (Kaggle 2020)

Çizelge 3.2. Öznitelikler ve açıklamaları

Öznitelik Adı	Veri Tipi	Açıklaması
Hafta	Sayısal veri (Tamsayılı değişken)	Satışın yapıldığı yılın hafta bilgisi
Ay	Sayısal veri (Tamsayılı değişken)	Satışın yapıldığı tarihin ay bilgisi
Yıl	Sayısal veri (Tamsayılı değişken)	Satışın yapıldığı tarihin yıl bilgisi
Resmi Tatil	Sayısal veri (İkili değişken)	Ülkelerin resmi tatilleri (yılbaşı haftası, 1 Mayıs İşçi Bayramı, ülkelerin bağımsızlık günü) ve Dünya genelinde tüm ülkeler tarafından kutlanan özel günler (anneler günü, babalar günü ve sevgililer günü)
Satış/Talep Miktarı	Sayısal veri (Tamsayılı değişken)	Satış hacmi

3.2. Benzerlik Ölçütleri

Bu tez çalışmasında, satış verileri arasındaki benzerlikleri hesaplamak için zaman serisi benzerlik ölçütleri kullanılmıştır. Satış verileri farklı uzunluklarda olabilmektedir. Bu nedenle, veri kümeleri arasındaki benzerlik hesaplarında elastik ölçütlerden DTW ile ERP ölçütleri kullanılmıştır. Bu bölümde, öncelikle tek değişkenli zaman serileri için kullanılan DTW ve ERP ölçütleri açıklanarak örnekler üzerinde gösterilmiştir. Ardından, çok değişkenli zaman serileri için kullanılan çok değişkenli DTW versiyonu açıklanmıştır.

3.2.1. Tek değişkenli dinamik zaman bükmesi

Dinamik zaman bükmesi (DTW), iki zaman serisi arasında en uygun hizalamayı bulmak için kullanılan bir tekniktir (Sakoe ve Chiba, 1978; Müller, 2007). Zaman serileri, birbiriyle eşleşmesi için doğrusal olmayan bir şekilde bükülür. DTW iki zaman serisini (dizi) hizalamak için bir dizideki boşlukları ilgili boşluktan bir önceki noktayı kopyalamak suretiyle doldurur. Buna DTW boşlukları adı verilir.

$R = [r_1, r_2, \dots, r_{T_1}]$ ve $S = [s_1, s_2, \dots, s_{T_2}]$ sırasıyla T_1 ve T_2 uzunluğunda iki farklı zaman serisi olsun. Bu zaman serileri arasındaki DTW uzaklığı Denklem (3.1)'de formüle edilmiştir. Zaman serileri R ve S arasındaki DTW değeri $DTW(R,S)$ ile gösterilir ve Denklem (3.2)'ye göre hesaplanır (Chen ve Ng, 2014). İki zaman serisi arasındaki dinamik çarpıtma mesafesi (DTW) L_1-norm veya L_2-norm kullanılarak hesaplanır. Her eşleşen çift için $(r_1,\ s_1)$ için $L_1 - \text{norm} = |r_1 - s_1|$ uzaklığını hesaplarken $L_2 - \text{norm} = (r_1 - s_1)^2$ uzaklığını hesaplar. Eğer eşleşmedeki bir nokta boş ise bu boşluğa bir önceki ögeyi kopyalayarak L_1-norm veya L_2-norm uzaklığını hesaplar (Sakoe ve Chiba, 1978; Müller, 2007). Denklem (3.1) ve Denklem (3.2) L_1-norm uzaklığına dayalı olarak verilmiştir (Chen ve Ng, 2014).

$$dist_{dtw}(r_i, s_j) = \begin{cases} |r_i - s_j| & eğer\ r_i, s_i\ boşluk\ değil\ ise; \\ |r_i - s_{j-1}| & eğer\ s_i\ boşluk\ ise; \\ |s_j - r_{i-1}| & eğer\ r_i\ boşluk\ ise. \end{cases} \quad (3.1)$$

$$DTW(R,S) = \begin{cases} 0 & eğer\ T_1 = T_2 = 0, \\ \infty & eğer\ T_1 = 0\ veya\ T_2 = 0, \\ dist_{dtw}(r_i, s_j) + \min \begin{cases} DTW\big(Kalan(R), Kalan(S)\big), \\ DTW(Kalan(R), S), \\ DTW\big(R, Kalan(S)\big) \end{cases} & \text{d.d.} \end{cases} \quad (3.2)$$

DTW bir metrik değildir ve üçgen eşitsizliğini sağlamayı garantilemez (Chen ve Ng, 2014).

3.2.2. Tek değişkenli gerçek cezalı düzenleme uzaklığı

DTW'nun bir versiyonu olan ERP ise hem zaman kaymasını içeren bir uzaklık ölçüm metriğidir, hem de üçgen eşitsizliğini garantiler. ERP, $L_1 -$ normunu ve gerçek sıralı düzenleme uzaklığını (Levenshtein, 1966) birleştiren bir uzaklık ölçüsüdür. ERP, önceki noktaların kopyalanmaması bakımından DTW'dan farklıdır. ERP, düzenleme uzaklığına benzer olarak örnek nokta eşleştirme yöntemini kullanır ve zamansal hizalama yaparak seriler arasındaki benzerliği değerlendirebilir (Chen ve Ng, 2014).

$R = [r_1, r_2, \dots, r_{T_1}]$ ve $S = [s_1, s_2, \dots, s_{T_2}]$ sırasıyla T_1 ve T_2 uzunluğunda iki farklı tek değişkenli zaman serisi olsun. Zaman serileri R ve S arasındaki ERP değeri $ERP(R,S)$ ile gösterilir ve Denklem (3.3) ve Denklem (3.4)'e göre hesaplanır (Chen ve Ng, 2014). ERP eşleşen çiftleri belirlemek için düzenleme uzaklığını kullanır ve her eşleşen çift (r_1, s_1) için $L_1 - \text{norm} = |r_1 - s_1|$ uzaklığını hesaplar. ERP'nin hesaplanmasında zaman serisi R'nin i. elemanı ile S'nin j. elemanı boşluk değil ise gerçek ceza kullanılır. Eğer eşleşmedeki bir nokta boş ise bu boşluğa sabit bir değer atayarak $L_1 -$ norm uzaklığını hesaplar. Denklem (3.4)'te g ile belirtilen sabit bir sayı olup herhangi bir değer alabilir. Chen ve Ng (2014), g'nin 0 seçilmesini önermektedir.

$$dist_{erp}(r_i, s_j) = \begin{cases} |r_i - s_j| & eğer\ r_i\ ve\ s_i\ boşluk\ değil\ ise; \\ |r_i - g| & eğer\ s_i\ bir\ boşluk\ ise; \\ |s_j - g| & eğer\ r_i\ bir\ boşluk\ ise. \end{cases} \quad (3.3)$$

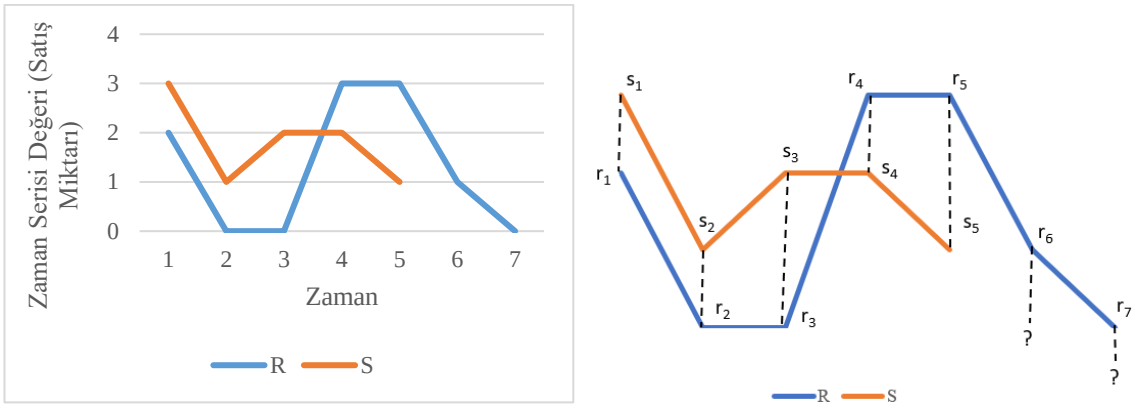
$$ERP(R,S) = \begin{cases} \sum_1^{T_1} |r_i - g| & \text{eğer } T_2 = 0, \\ \sum_1^{T_2} |s_j - g| & \text{eğer } T_1 = 0, \\ \min \begin{cases} ERP\big(Kalan(R), Kalan(S)\big) + dist_{erp}(r_1, s_1), \\ ERP(Kalan(R), S) + dist_{erp}(r_1, [\,.\,]), \\ ERP\big(R, Kalan(S)\big) + dist_{erp}(s_1, [\,.\,]) \end{cases} & \text{d.d.} \end{cases} \quad (3.4)$$

Burada, [.] zaman serisinin ilgili elemanının boşluk olduğunu gösterir.

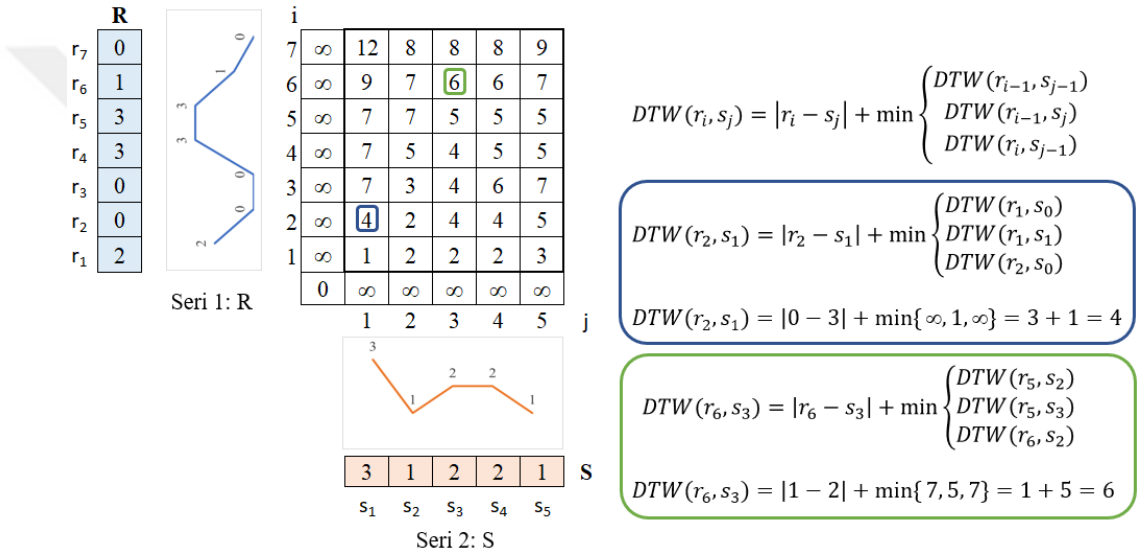
3.2.3. Zaman serisi arasındaki uzaklıkların hesaplanması için bir örnek

Bu bölümde, iki zaman serisi arasındaki uzaklığın DTW ve ERP metrikleri ile nasıl hesaplandığı bir örnek ile açıklanmıştır. Şekil 3.1'de grafikleri verilen R = [2, 0, 0, 3, 3, 1, 0] ve S = [3, 1, 2, 2, 1] farklı uzunluklarda iki farklı zaman serisi olsun. Şekil 3.1'de görüldüğü üzere R serisinde yer alan r_6 ve r_7 zaman adımları için S serisinde karşılık gelen zaman adımları bulunmamaktadır.

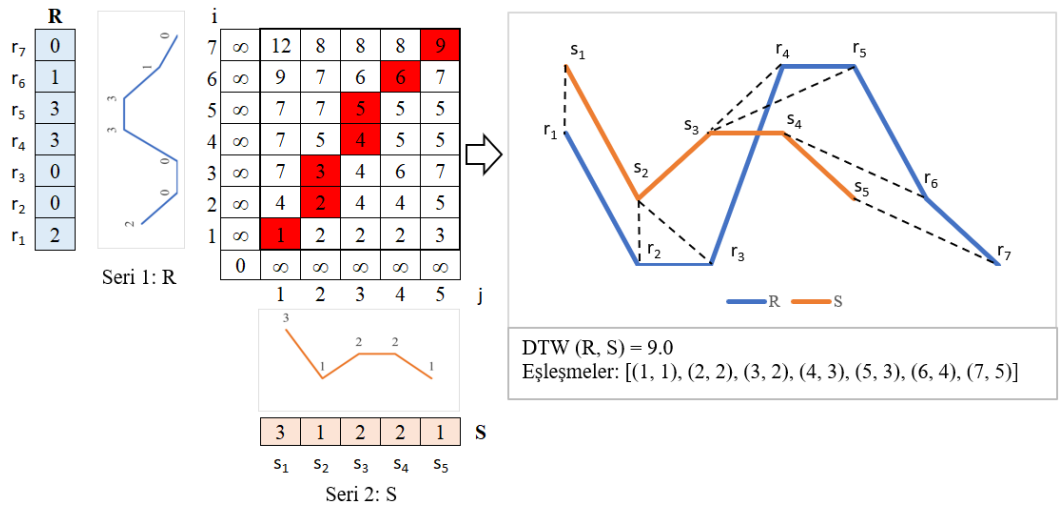
İki zaman serisi arasındaki uzaklık hesabı için DTW uygulandığında Şekil 3.2'deki gibi maliyet (uzaklık) matrisi oluşturulur. Bu matris (3.1)-(3.2) denklemleri kullanılarak yapılan hesaplamalar neticesinde oluşturulur. Bu hesaplamalara örnek olarak matristeki rastgele iki değerin hesaplaması Şekil 3.2'de gösterilmiştir. R ve S zaman serilerinin DTW uzaklığı 9 olarak bulunmuş ve oluşan DTW yörüngesi [(1, 1), (2, 2), (3, 2), (4, 3), (5, 3), (6, 4), (7, 5)] şeklindedir. Bu yörünge, her iki zaman serisinin değerleri arasında oluşturulmuş eşleşmelerdir. Bu eşleşmenin grafiğe dökülmüş hali Şekil 3.3'te gösterilmektedir.



Şekil 3.1. R ve S zaman serilerinin grafikleri

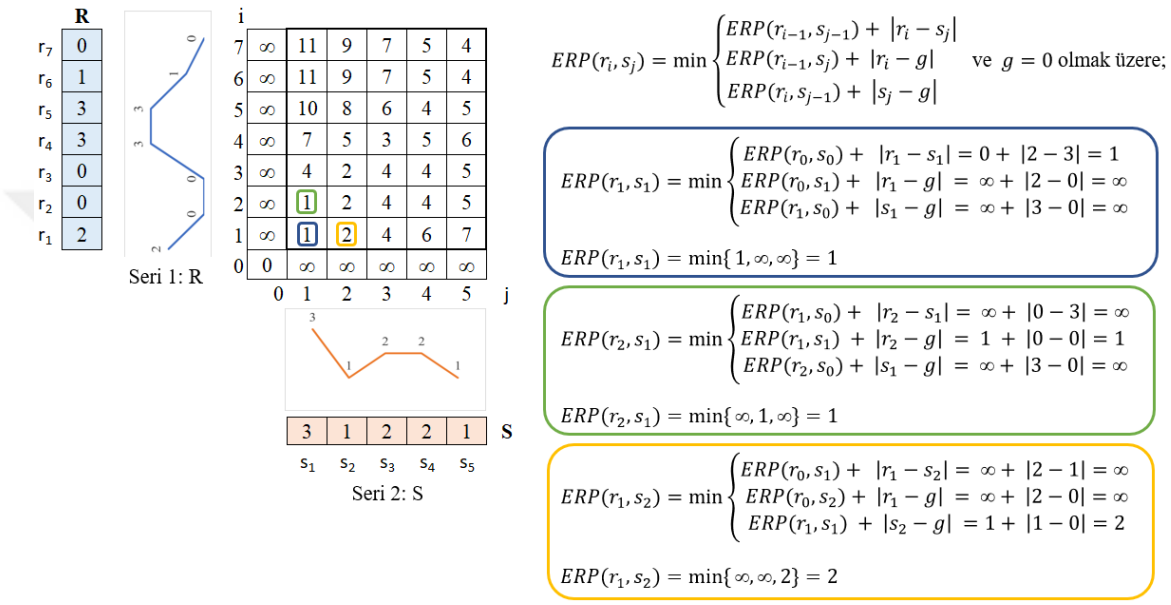


Şekil 3.2. DTW maliyet matrisi ve uzaklık hesaplamaları

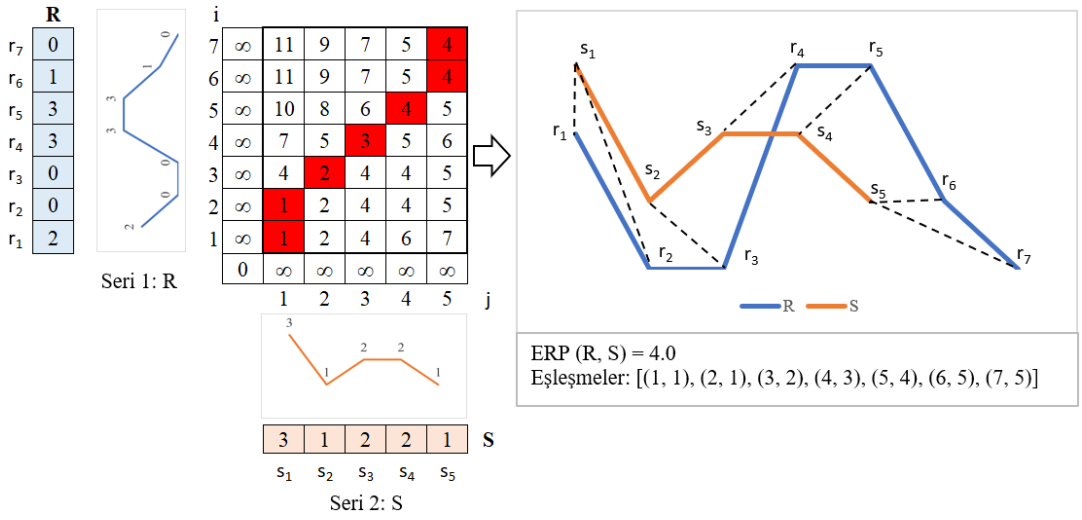


Şekil 3.3. DTW yörüngesi ile R ve S serilerinin eşleşme grafiği

R ve S zaman serileri arasındaki uzaklığı ERP ile hesaplamak için (3.3)-(3.4) denklemleri kullanılır. Bu hesaplamalara örnek olarak matristeki rastgele iki değerin hesaplaması Şekil 3.4'te gösterilmiştir. R ve S zaman serilerinin ERP uzaklığı 4 olarak bulunmuş ve oluşan ERP yörüngesi [(1, 1), (2, 1), (3, 2), (4, 3), (5, 4), (6, 5), (7, 5)] şeklindedir. Bu yörünge her iki zaman serisinin değerleri arasında oluşturulmuş eşleşmelerdir. Bu eşleşmenin grafiğe aktarılmış hali Şekil 3.5'te gösterilmektedir.



Şekil 3.4. ERP maliyet matrisi ve uzaklık hesaplamaları



Şekil 3.5. ERP yörüngesi ile R ve S serilerinin eşleşme grafiği

R ve S zaman serileri arasındaki DTW uzaklığı 9 ve ERP uzaklığı 4'tür. DTW, zaman serilerini elastik bir biçimde hizalayarak hesaplanır. Elastik hizalama, her bir zaman serisinin zaman ölçeğini ve hızını ayarlamaya izin verir. Böylece zaman serileri arasındaki benzerlik ölçülebilir. ERP, her iki zaman serisinin kesin olarak hizalanmasıyla hesaplanır. Hizalama işlemi sırasında her bir öğenin değiştirilmesi, ekleme veya silme işlemleri için sabit bir ceza uygulanır. Bu nedenle, ERP uzaklığı, zaman serilerinin kesin hizalaması için en az maliyetli yolu bulmaya çalışır. Bu nedenle, ERP uzaklığı DTW uzaklığına göre genellikle daha küçük bir değer alır (Chen ve Ng, 2014).

3.2.3. Çok değişkenli dinamik zaman bükmesi

Elastik olmayan uzaklık ölçütleri tek değişkenli zaman serileri için geliştirilmiştir. Denklem (3.1) ve Denklem (3.2)'de formüle edilen DTW uzaklığı yalnızca tek boyutlu (değişkenli) zaman serileri için geçerlidir. Shokoohi-Yekta vd. (2017) DTW uzaklığını çok değişkenli zaman serisi versiyonuna genişletmişlerdir. Burada, zaman serisinin her bir boyutu zaman serisinin bir değişkenini bir diğer ifadeyle özniteliğini ifade etmektedir.

Q ve *C* iki *M*-boyutlu zaman serisi olmak üzere bu zaman serileri arasındaki DTW uzaklığını hesaplamak için DTW_I ($DTW_{Independent}$) ve DTW_D ($DTW_{Dependent}$) olmak üzere iki olası yaklaşım önermişlerdir.

DTW_I, DTW kapsamında bağımsız olarak ölçülen tüm boyutların kümülatif uzaklıklarıdır. $DTW(Q_m, C_m)$, *Q*'nun *m.* boyutu ile *C*'nin *m.* boyutu arasındaki DTW uzaklığı olmak üzere DTW_I uzaklığı Denklem (3.5)'teki gibi hesaplanabilir (Shokoohi-Yekta vd., 2017).

$$DTW_I(Q, C) = \sum_{m=1}^{M} DTW(Q_m, C_m) \tag{3.5}$$

DTW_I uzaklık hesabında her boyutun bağımsız olduğu kabul edilir ve DTW'ya her boyutu diğerlerinden bağımsız olarak çarpıtma özgürlüğü verilir. Diğer taraftan çok boyutlu DTW, tek bir çarpıtma matrisinde tüm boyutları aynı şekilde eğrilmeye zorlayacak şekilde hesaplanabilir. Başka bir ifadeyle, boyutların bağımsızlığına izin

verilmediği ve tüm boyutlar arasında karşılıklı bağımlılık olduğu varsayılır. Bu durumda hesaplanan DTW uzaklığı, DTW_D uzaklığı olarak tanımlanır (Shokoohi-Yekta vd., 2017).

DTW_D, tek boyutlu zaman serileri için hesaplanan DTW'ya benzer şekilde hesaplanır. Denklem (3.1)'de verilen $dist_{dtw}(q_i, c_j)$ hesabı tek bir veri noktası yerine M veri noktasının kümülatif kareli Öklid uzaklıkları ($L_2 - norm$) olarak Denklem (3.6)'daki gibi tanımlanır. $q_{i,m}$, Q'nun m. boyutunda i. veri noktası ve $c_{j,m}$, C'nin m. boyutunda j. veri noktası olmak üzere $dist_{dtw}(q_i, c_j)$ Denklem (3.6)'daki gibi güncellenir (Shokoohi-Yekta vd., 2017).

$$dist_{dtw}(q_i, c_j) = \sum_{m=1}^{M} \begin{cases} (q_{i,m} - c_{j,m})^2 & eğer\ q_{i,m}, c_{j,m}\ boşluk\ değil\ ise; \\ (q_{i,m} - c_{j-1,m})^2 & eğer\ c_{j,m}\ boşluk\ ise; \\ (c_{j,m} - q_{i-1,m})^2 & eğer\ q_{i,m}\ boşluk\ ise. \end{cases} \quad (3.6)$$

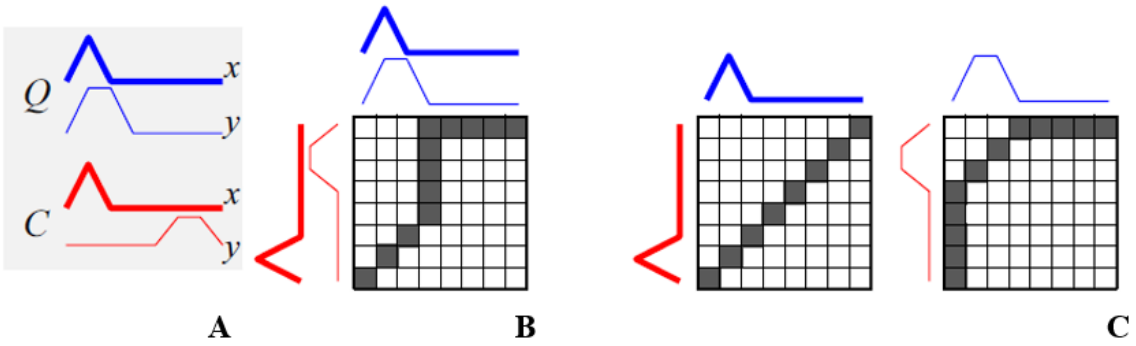
$DTW_D(Q, C)$ uzaklığı ise tek boyutlu zaman serileri için hesaplanan Denklem (3.2) ile benzer şekilde hesaplanır (Shokoohi-Yekta vd., 2017).

DTW_I ve DTW_D hesabına örnek olması açısından için M =2 iki olduğu durum Şekil 3.6'da gösterilmiştir. Burada Q ve C iki boyutlu zaman serileri olmak üzere DTW_I hesabı Denklem (3.7) ve DTW_D hesabı Denklem (3.8)'deki gibi hesaplanır (Shokoohi-Yekta vd., 2017).

$$DTW_I(Q, C) = \sum_{m=1}^{M=2} DTW(Q_m, C_m) = DTW(Q_x, C_x) + DTW(Q_y, C_y) \quad (3.7)$$

$$DTW_D(Q, C) = DTW(\{Q_x, C_x\}, \{Q_y, C_y\}) \quad (3.8)$$

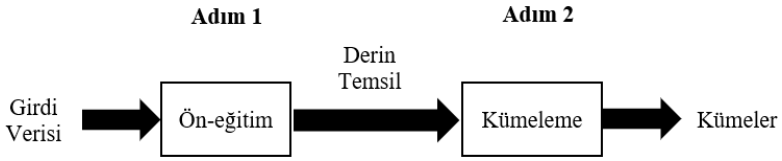
Burada, x zaman serilerinin birinci boyutunu (özniteliği) ve y ise ikinci boyutunu (özniteliği) ifade eder.



Şekil 3.6. İki adet çok boyutlu zaman serisi arasındaki çok değişkenli DTW uzaklıkları **A)** İki adet çok boyutlu zaman serisi **B)** Seriler arasındaki DTW_D uzaklığı **C)** Seriler arasındaki DTW_I uzaklığı (Shokoohi-Yekta vd., 2017)

3.3. Derin Otokodlayıcı Tabanlı Derin Kümeleme

Bu tezde, zaman serilerinin kümelenmesi için derin kümeleme yöntemi ele alınmıştır. Zaman serileri tek görünümlü veriler olduğu için derin tek-görünümlü kümeleme yöntemleri üzerine odaklanılmıştır. Nutakki vd. (2018) derin tek görünümlü kümeleme yöntemlerini eğitim stratejilerine göre çok adımlı ardışık derin kümeleme, birleşik derin kümeleme ve kapalı döngü çok adımlı derin kümeleme olmak üzere üç kategoriye ayırmaktadır. Sıralı çok adımlı derin kümeleme (sequential multistep deep clustering) yaklaşımları Şekil 3.7'de gösterildiği gibi iki ana adımdan oluşur. İlk adım, girdi verilerinin derin (gizli) temsilini öğrenirken, ikinci adım bu derin veya gizli temsil üzerinde kümeleme gerçekleştirir. Bu yaklaşımlardan biri DAE ağı tabanlı derin kümeleme yaklaşımıdır (Nutakki vd., 2018; Ren vd., 2022).



Şekil 3.7. Sıralı çok adımlı derin kümeleme (Nutakki vd., 2018)

Derin otomatik kodlayıcı ağı, verilerin denetimsiz temsili öğrenimi için tasarlanmıştır ve doğrusal olmayan bir haritalama fonksiyonunu öğrenebilir. Şekil 3.8'de derin otokodlayıcı yapısı gösterilmektedir. Derin otomatik kodlayıcı derin kümeleme yöntemlerinin geliştirilmesinde sıklıkla kullanılır. DAE, Denklem (3.10)'da tanımlanan

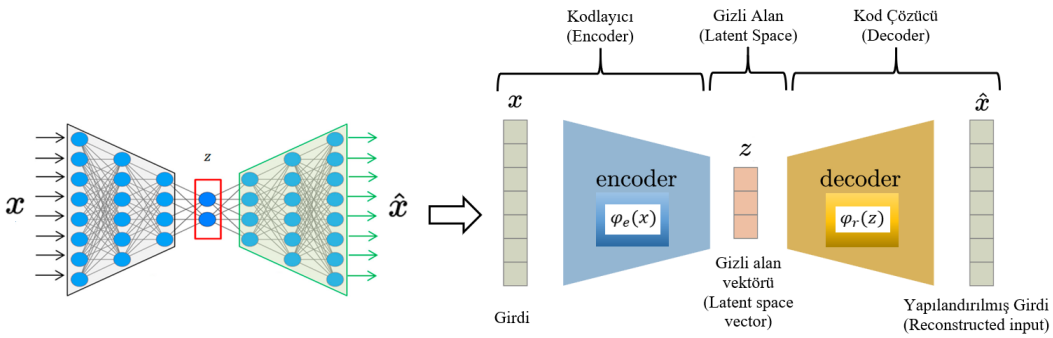
ağın yeniden yapılandırma kaybını (L_{rec}) en aza indirerek düşük boyutlu bir gömme özellik uzayını öğrenmeyi amaçlamaktadır:

$$z_i = \varphi_e(x_i) \tag{3.9}$$

$$L_{rec} = \min \frac{1}{n} \sum_{i=1}^{n} \|x_i - \varphi_r(z_i)\|^2 ; \tag{3.10}$$

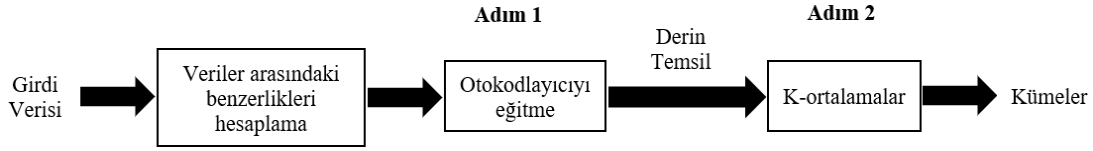
Denklem (3.9)'da x_i *i.* girdiyi, $\varphi_e(.)$ otomatik kodlayıcının kodlayıcı ağını ve z_i *i.* girdinin gizli alan (latent space) değerini temsil etmektedir (Ren vd., 2022). Denklem (3.10)'da $\varphi_r(.)$ otomatik kodlayıcının kod çözücü ağını ve n nesne sayısını ifade etmektedir (Ren vd., 2022). Kodlayıcıyı bir özellik çıkarıcı olarak kullanarak çeşitli kümeleme amaç fonksiyonları önerilmiştir.

DAE tabanlı derin kümelemede iki temel yaklaşım vardır. İlk yaklaşım, verileri düşük boyutlu özniteliklere yerleştirir ve daha sonra gömülü öznitelikleri k-ortalamalar algoritması gibi geleneksel kümeleme yöntemleriyle kümeler. Bu yaklaşımda, DAE ve kümeleme amaç fonksiyonları ayrı ayrı optimize edilir. İkinci yaklaşım, özellik çıkarıcıyı ve kümeleme sonuçlarını birlikte optimize eder. Bu tezde, ilk yaklaşım kullanılmıştır.



Şekil 3.8. Derin otokodlayıcı yapısı

Şekil 3.9'da bu tez çalışmasında kullanılan DAE tabanlı derin kümeleme yapısı gösterilmektedir (Nutakki vd., 2018). Bu yöntemde otomatik kodlayıcıyı doğrudan girdi verileri üzerinde eğiten çok adımlı yöntemlerden farklı olarak önceden hesaplanmış bir benzerlikten bir temsil (gömme) çıkarılır ve sonrasında otomatik kodlayıcı bu temsilleri kodlamak için eğitilir. Eğitim sonrası elde edilen derin temsiller k-ortalamalar yöntemine girdi olarak verilerek kümeleme işlemi gerçekleştirilir.



Şekil 3.9. k-ortalamalar içeren DAE tabanlı derin kümeleme (Nutakki vd., 2018)

3.4. Performans Değerlendirme Yöntemleri

Tahmin problemlerinde model performansları, performans metrikleri ile değerlendirilmektedir. Performans metrikleri, tahmini değer ile gerçekleşen değer arasındaki fark (hata) üzerinden hesaplanmaktadır. Gerçekleşen değerlere ne kadar yakın tahminler elde edilirse performans metrikleri de aynı ölçüde küçük değerler almaktadır. Bu tez kapsamında, tahmin sonuçları Denklem (3.11) ve Denklem (3.12)'de verilen Ortalama Karesel Hatanın Karekökü (Root Mean Square Error – RMSE) ve Ortalama Mutlak Hata (Mean Absolute Error – MAE) performans metrikleriyle değerlendirilmiştir. Bu denklemlerde $\hat{y}_{it}$ tahmini satış değeri, y_{it} gerçek satış değerini ve n test veri kümesindeki örneklem sayısını ifade etmektedir.

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(\hat{y}_{it} - y_{it})^2}, \quad \forall i \in D \tag{3.11}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|\hat{y}_{it} - y_{it}|, \quad \forall i \in D \tag{3.12}$$

RMSE, büyük hataların küçük hatalardan daha fazla cezalandırıldığı bir ölçüttür. MAE metriğinde ise büyük hatalar küçük hatalarla aynı ağırlığa sahiptir. Büyük hatalara daha fazla vurgu yapmak istendiğinde veya veri setinde aykırı değerler olduğunda RMSE, tüm hataları eşit önemde tutmak istendiğinde ve aykırı değerlerin etkisi azaltmak istendiğinde MAE metriği tercih edilir. Bu metriklerin sıfıra yakın değerler alması, model tahminlerinin gerçek değerlere yakın olduğunu göstermektedir.

Bu tez kapsamında, kullanılan veri setleri sıfır değerini içerdiği için Ortalama Mutlak Yüzde Hata (Mean Absolute Percentage Error – MAPE) performans metriği kullanılmamıştır.

3.5. Tek Değişkenli Zaman Serileri için Önerilen Çözüm Yaklaşımları

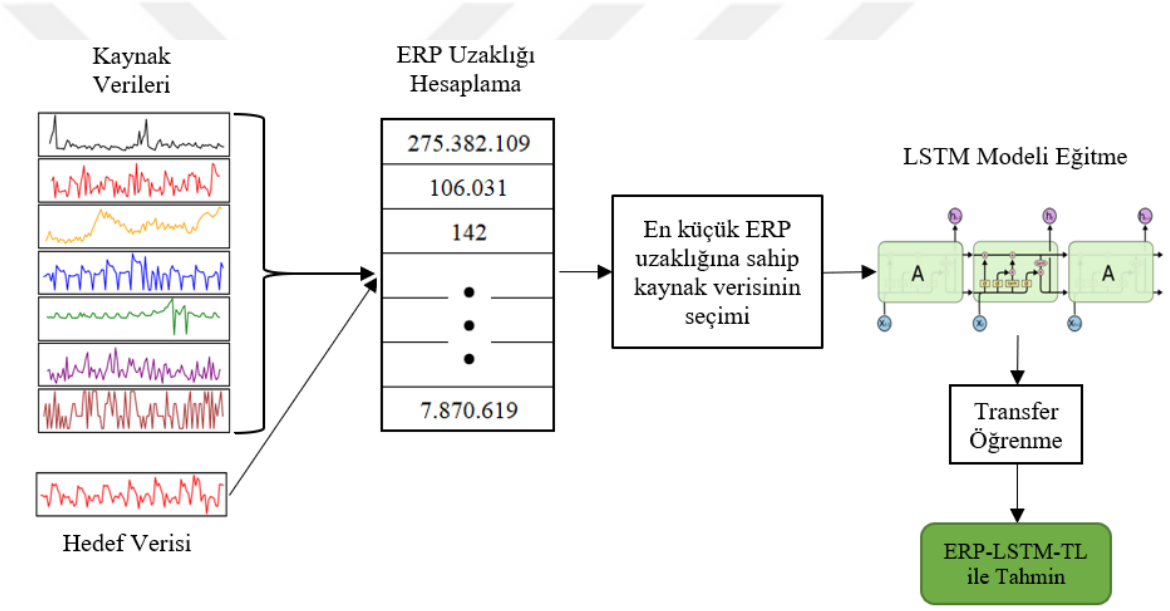
Bu bölümde satış tahmin problemi tek değişkenli zaman serisi problemi olarak ele alınmıştır. Tez kapsamında tek değişkenli zaman serileri için üç farklı çözüm yaklaşımı sunulmaktadır. Bölüm 3.3.1'de yer alan ilk yaklaşımda uzaklık tabanlı derin transfer öğrenme yöntemi geliştirilmiştir. Bu yöntemde tek kaynaklı transfer öğrenmede kaynak seçimi için uzaklık tabanlı bir seçim yaklaşımı uygulanmıştır. Bölüm 3.3.2'de verilen ikinci yaklaşımda, birinci yaklaşımda geliştirilen yönteme topluluk öğrenme yöntemleri entegre edilmiştir. Bu kapsamda torbalama ve yığınlama topluluk yöntemleri kullanılmıştır. Böylece, tek değişkenli zaman serileri için tek kaynaklı topluluk tabanlı derin transfer öğrenme yöntemleri elde edilmiştir. Bölüm 3.3.3'te yer alan üçüncü yaklaşımda ise tek kaynaklı transfer öğrenme yöntemi çok kaynaklı transfer öğrenme yöntemine genişletilmiştir. Bu kapsamda çoklu kaynak seçimi için k-en yakın komşuluk tabanlı, kümeleme tabanlı ve ağırlıklandırılmış kümeleme tabanlı seçim yaklaşımları geliştirilmiştir.

3.5.1. Uzaklık tabanlı derin transfer öğrenme

Farklı sektörler arasında bilgi aktarımını sağlayarak tahmin performansını arttırmak amacıyla uzaklık tabanlı derin transfer öğrenme yaklaşımı geliştirilmiştir. Bu yaklaşımda, derin öğrenme yöntemlerinden doğrusal olmayan uzun süreli bağımlılıkları yakalayabilen ve zaman serisi tahmininde etkili sonuçlar veren LSTM ağı kullanılmıştır. Transfer öğrenme sürecinde ise ağ bazlı aktarım ele alınmıştır (Niu vd., 2020).

Kaynak ve hedef veri arasında benzerlik olması transfer öğrenme için fayda sağlamaktadır (Meiseles ve Rokach, 2020). Bu nedenle, aktarılabilirlik ve kaynak seçiminde benzerlik analizi kullanılmıştır. Satış verileri arasındaki benzerlik analizinde ERP metriği uygulanmıştır. Özetle, ERP'ye dayalı LSTM tabanlı derin transfer öğrenme yaklaşımı (ERP-LSTM-TL), bir hedef satış verisi için en benzer kaynak satış verisine LSTM ile derin transfer öğrenme uygulayarak tahmin doğruluğunu arttırmayı amaçlamaktadır. Önerilen yaklaşımın genel akış şeması Şekil 3.10'da verilmiştir.

Kaynak veri kümesi $X_K = \{x_1, x_2, .., x_n\}$, n adet potansiyel kaynak satış verisinden oluşmaktadır ve l. potansiyel kaynak satış verisinde $x_l = \{x_{lt} \mid t = 1, 2, .., T_l\}$, T_l adet dönem bulunmaktadır. Hedef satış verisi $y_h = \{y_{ht} \mid t = 1, 2, .., T_h\}$, T_h döneme ait satış verisini içermektedir. Buna göre, ilk olarak, potansiyel kaynak ve hedef satış verileri [0, 1] aralığına normalize edilir. Sonrasında, hedef ve potansiyel kaynak verileri arasında ERP uzaklığı hesaplanır ve hedef veriye en küçük uzaklığa sahip (en benzer) satış verisi kaynak veri olarak seçilir. Ardından, seçilen kaynak veri ile kaynak LSTM ağı eğitilir. Son olarak, parametre transferi uygulanır ve ön eğitilmiş kaynak LSTM ağı hedef veri ile tekrar eğitilir. Böylece, hedef veri için LSTM tahmin modeli elde edilir. ERP-LSTM-TL algoritması Şekil 3.11'de verilmiştir.



Şekil 3.10. ERP'ye dayalı LSTM tabanlı derin transfer öğrenme yaklaşımının akış şeması

Bu tez çalışması, farklı sektörler ve farklı uzunluklardaki satış verileri arasında kaynak seçimi içeren LSTM tabanlı derin transfer öğrenme yaklaşımı sunması itibariyle literatüre katkı sağlamaktadır. Ayrıca, bilindiği kadarıyla farklı uzunluktaki zaman serilerinde aktarılabilirlik ölçütü olarak metrik bir uzunluk olan ERP uzaklığını kullanan ilk çalışmadır (Erol ve İnkaya, 2024).

Algoritma 1: ERP-LSTM-TL algoritması

Girdi: $X_K = \{x_1, x_2, .., x_n\}$ kaynak veri kümesi ve $y_h = \{y_{ht} \mid t = 1, 2, .., T_h\}$ hedef veri

Çıktı: Hedef veri için satış tahmini modeli

Yöntemin adımları:

Adım 1. Kaynak veri kümesi X_K ve hedef satış verisi y_h, [0, 1] aralığına normalize edilir.

Adım 2. Hedef veri ve potansiyel kaynak veriler arasındaki ERP uzaklığı $ERP(x_l, y_h), \forall\, x_l \in X_K$ hesaplanır.

Adım 3. Hedef veriye en küçük ERP uzaklığına sahip kaynak veri x_{l^*}, $l^* = \arg\min_l \{ERP(x_l, y_h)\}$ seçilir.

Adım 4. Kaynak LSTM modeli, seçilen kaynak veri x_{l^*} ile hiperparametre optimizasyonu yapılarak eğitilir.

Adım 5. Kaynak LSTM modelindeki hiperparametre ve ağırlıklar hedef LSTM modeline aktarılır. LSTM-TL modeli elde edilir.

Adım 6. Hedef LSTM-TL modeli, hedef veri y_h ile eğitilir.

Şekil 3.11. ERP-LSTM-TL algoritması

3.5.2. Topluluk tabanlı derin transfer öğrenme

Tahmin doğruluğunu arttırmak amacıyla LSTM-TL modeli ile torbalama ve yığınlama topluluk yöntemleri kullanılmıştır. Torbalama yönteminde tahmin sonuçlarının birleştirilmesi için ortalama alma yöntemi uygulanmıştır. Önerilen torbalama algoritmasının sözde kodu Şekil 3.12'de verilmiştir.

Yığınlama yönteminde ise farklı LSTM-TL modelleri istiflenerek yığın LSTM-TL modeli oluşturulmuştur. Yığınlama yönteminde temel öğrenici LSTM-TL yöntemi iken meta-öğrenici olarak literatürde sıklıkla tercih edilen SVR yöntemi kullanılmıştır (Ribeiro ve Coelho, 2020; Ribeiro vd. 2022). Yığınlama yönteminde, temel ve meta öğrenicilerin eğitilmesi için veri eğitim ve test olarak ikiye ayrılmıştır. Aynı eğitim seti ile tüm temel öğreniciler eğitilmiştir. Ardından bu tahminler meta öğreniciye girdi (özellik) olarak verilerek meta-öğrenicinin eğitim işlemi gerçekleştirilmiştir. Son olarak, test verisiyle meta öğrenici test edilerek topluluk modelin tahmin sonuçları elde edilmiştir (Livieris vd., 2020). Önerilen yığınlama algoritmasının sözde kodu Şekil 3.13'te verilmektedir.

Algoritma 2: Torbalama tabanlı tek kaynaklı derin transfer öğrenme algoritması

Girdi: $D_k = \{y_t: \ t \in T_k\}$ kaynak ve $D_h = \{y_t: \ t \in T_h\}$ hedef veri kümeleri
Çıktı: Torbalama topluluk modeli
Adım 1. Kaynak ve hedef satış verilerine [0, 1] normalizasyonu uygulanır.
Adım 2. Kaynak veri kümesi ile LSTM kaynak modeli geliştirilir ve eğitilir.
Adım 3. *Adım 2*'de ön eğitilmiş LSTM ağı ve ağın ağırlık değerleri alınır. Hedef veri kümesine ağ yapısı değiştirilmeden parametre transferi yaklaşımı ile TL uygulanır.
Adım 4. *Adım 2* ve *Adım 3 N* kez tekrarlanarak *N* adet LSTM-TL modeli oluşturulur.
Adım 5. Her bir LSTM-TL modelinde hedef test verisi tahmin edilir ve tahmin değerlerinin ortalamasının alınması suretiyle topluluk modelinin sonuçları elde edilir.

Şekil 3.12. Torbalama tabanlı tek kaynaklı derin transfer öğrenme algoritmasının sözde kodu

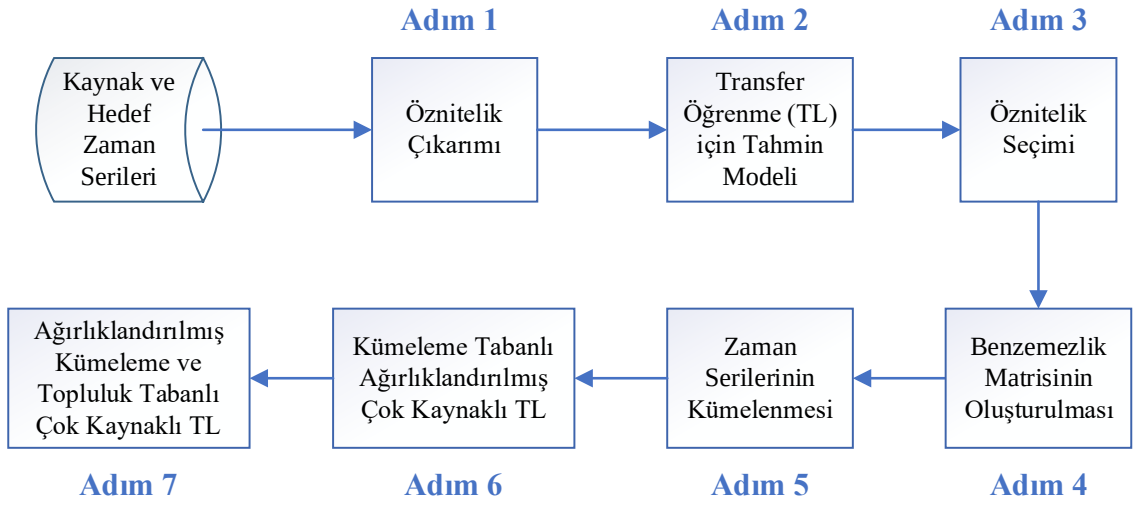
Algoritma 3: Yığınlama tabanlı tek kaynaklı derin transfer öğrenme algoritması

Girdi: $D_k = \{y_t: \ t \in T_k\}$ kaynak ve $D_h = \{y_t: \ t \in T_h\}$ hedef veri kümeleri
Çıktı: Yığınlama topluluk modeli
Adım 1. Kaynak ve hedef satış verilerine [0, 1] normalizasyonu uygulanır.
Adım 2. Kaynak veri kümesi ile LSTM kaynak modelini geliştirilir ve eğitilir.
Adım 3. *Adım 2*'de ön eğitilmiş LSTM ağı ve ağın ağırlık değerleri alınır. Hedef veri kümesine ağ yapısı değiştirilmeden parametre transferi yaklaşımı ile TL uygulanır.
Adım 4. *Adım 2* ve *Adım 3 N* kez tekrarlanarak *N* adet LSTM-TL modeli oluşturulur.
Adım 5. Her bir LSTM-TL modelinde hedef eğitim verisi tahmin edilir ve eğitim tahmin değerleri ile destek vektör regresyonu modeli eğitilerek yığın topluluk modeli oluşturulur.
Adım 6. Yığın topluluk modeli ile hedef test verisi tahmin edilir.

Şekil 3.13. Yığınlama tabanlı tek kaynaklı derin transfer öğrenme algoritmasının sözde kodu

3.5.3. Kümeleme ve topluluk tabanlı derin transfer öğrenme

Bu tezde, kümeleme ile satış verileri arasındaki benzerlikleri bularak aynı kümeye atanan satış verileri için transfer öğrenmede kaynak ve hedef eşleştirmesinin yapılması amaçlanmıştır. Böylece TL'deki bilgi aktarma başarısının arttırılması hedeflenmiştir. Bu doğrultuda, kümeleme ve topluluk tabanlı derin transfer öğrenme metodolojisi geliştirilmiştir. Şekil 3.14'te akış şeması verilen bu metodolojinin adımları aşağıda özetlenmiştir:



Şekil 3.14. Tek değişkenli zaman serileri için kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımının akış şeması

Adım 1: Özniteliklerin çıkarımı

Transfer öğrenmede hedefe uygun kaynak seçimi, model performansını olumlu yönde etkileyen önemli bir unsurdur. Bu nedenle, zaman serileri arasındaki benzerlikleri hesaplamak ve bu benzerlikleri kullanarak hedefe uygun kaynak seçimi yapabilmek için öznitelik çıkarımı yapılmıştır. Çıkarılan öznitelikler ile veriyi en iyi şekilde tanımlayarak transfer öğrenmede kaynak ve hedef verileri kullanma nedenlerini açıklayabilmek amaçlanmıştır. Buna göre, kaynak ve hedef satış verilerinin öznitelikleri Şekil 3.15'te gösterildiği üzere üç başlıkta incelenmiştir: Zaman serilerinin istatistiksel analizi sonucu elde edilen öznitelikler, satış ile ilgili öznitelikler ve uzaklık tabanlı öznitelikler.

1. **Zaman Serisi Öznitelikleri:** İstatistiksel öznitelikler, zaman serisi verilerindeki trend, dalgalanma, mevsimsellik ve diğer önemli örüntüleri izlemek ve keşfetmek için kullanılmaktadır (Henderson ve Fulcher, 2021). Bu öznitelikler, veri analizinde ve modelleme sürecinde faydalı bilgiler sağlamaktadır (Henderson ve Fulcher, 2021). Bu kapsamda, bu tezde zaman serisine ait istatistiksel özniteliklerin çıkarılması için R programında *tsfeatures* kütüphanesi kullanılmıştır (Hyndman vd., 2018; Henderson ve Fulcher, 2021). Her satış serisi için ortalama, varyans, otokorelasyon, trend, mevsimsellik vb. olmak üzere toplam 22 adet öznitelik çıkarılmıştır. Bu öznitelikler Ek 4'te açıklanmıştır.

2. **Satış ile ilgili Öznitelikler:** Her satış verisi için veri büyüklüğü, başlangıç yılı, satılan ürün grubu ve sıfır satış oranı olmak üzere dört adet öznitelik ele alınmıştır. Ayrıca, kaynak-hedef eşleşmelerine göre sektör bilgisi ile benzer ürün satışlarının olup olmadığına dair iki adet öznitelik oluşturulmuştur. Bu grupta toplamda altı adet öznitelik bulunmaktadır. Bu öznitelikler detaylı bir şekilde aşağıda açıklanmaktadır:

   Seri Büyüklüğü (SeriesSize): Satış verisinin uzunluğunu (büyüklüğünü) ifade eder. Örneğin, Walmart veri kümesi 143 haftalık bir satış verisidir.

   Satış Başlangıç Yılı (SalesStartYear): Satış zaman serisinin başlangıç yılını ifade eder. Örneğin, Walmart veri kümesi 2010-2012 yılları arasındaki satışları içermektedir, bu durumda bu veri için başlangıç yılı 2010'dur.

   Ürün Grubu (ProductGroup): Satış verisinin tek ürün grubu veya çoklu ürün grubu içermesine yönelik oluşturulan özniteliktir. Tek ürün grubu içeren veriler 0, çoklu ürün grubu içeren veriler 1 olacak şekilde ikili kodlama yapılmıştır. Örneğin, Walmart bir perakende zincirine ait veri kümesidir. Bu nedenle, bu veri birden fazla ürün satışlarını içermektedir ve çoklu ürün grubuna dahildir. Bu öznitelik değeri Walmart için 1'dir. Benzer şekilde, Tv_Sales tek ürün grubuna (televizyon) ait satışları içeren bir veri kümesidir ve bu öznitelik değeri Tv_Sales için 0'dır.

   Sıfır Satış Oranı (ZeroSalesPer): Satış seyrekliğini, bir diğer ifadeyle sıfır satış oranını temsil eden özniteliktir. Bir veri kümesinde sıfır satış değerine sahip hafta sayısı, toplam hafta sayısına bölünerek Denklem (3.13) ile sıfır satış oranı hesaplanır.

$$ZeroSalesPer = \frac{Sıfır\ satış\ değerine\ sahip\ hafta\ sayısı}{Toplam\ hafta\ sayısı} \tag{3.13}$$

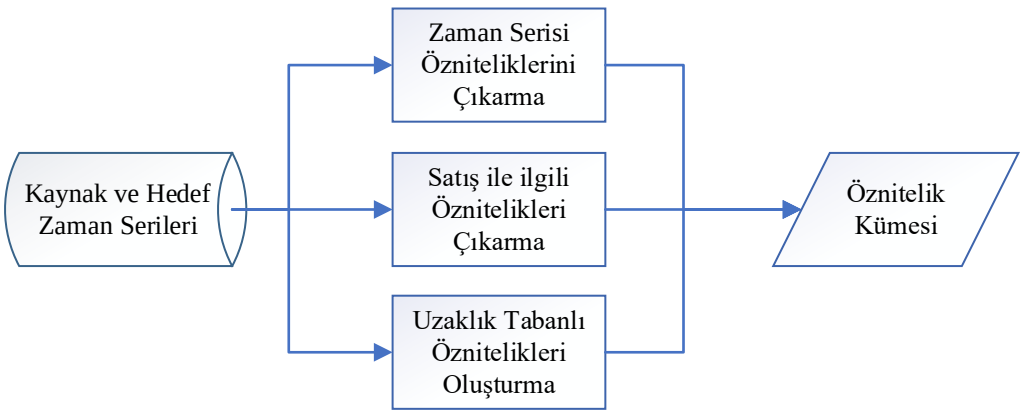
   Sektör Bilgisi (Sector_Info): Kaynak ve hedef satış verilerinin aynı veya farklı sektöre ait olmasına yönelik oluşturulan özniteliktir. Kaynak-hedef veri

eşleşmelerinde aynı sektöre ait veriler 1, farklı sektöre ait veriler 0 olacak şekilde ikili kodlama yapılmıştır. Örneğin, Walmart-LearnX kaynak-hedef veri eşlemelerinde bu iki veri farklı sektörlere ait veriler olduğu için bu öznitelik değeri sıfırdır. Walmart-Rossmann kaynak-hedef eşlemesinde ise her iki veri de perakende sektörüne ait veriler olduğu için bu öznitelik değeri 1'dir.

Benzer Ürün Satışı (SimilarProduct): Kaynak ve hedef satış verilerinin benzer ürünlerin satışını içerip içermemesine yönelik oluşturulan özniteliktir. Kaynak-hedef veri eşleşmelerinde benzer ürünlere ait satışları içeren veriler 1, benzer ürünlere ait satışları içermeyen veriler 0 olacak şekilde ikili kodlama yapılmıştır. Örneğin, Walmart-LearnX kaynak-hedef veri eşlemesinde bu iki veri farklı ürünlere ait satışları içerdiği için bu öznitelik değeri sıfırdır. Walmart-Rossmann kaynak-hedef veri eşlemesinde ise her iki veri de perakende sektörüne ait ve benzer ürün satışlarını içerdiği için bu öznitelik değeri 1'dir.

3. **Uzaklık Tabanlı Öznitelikler:** Transfer öğrenme kararına fayda sağlaması için kaynak ile hedef verileri arasındaki şekilsel uzaklıklar ele alınmıştır. Bu kapsamda, tüm kaynak-hedef veri eşleşmelerinde kaynak-hedef verileri arasındaki ERP ve DTW uzaklıkları alınarak iki adet öznitelik oluşturulmuştur.

Özetle, bu tezde 22 adet zaman serisi öznitelikleri, 6 adet satışla ilgili öznitelikler ve 2 adet uzaklık tabanlı öznitelikler olmak üzere toplam 30 adet öznitelik ele alınmıştır. Tüm öznitelikler ve açıklamaları Ek 5'te sunulmaktadır.



Şekil 3.15. Öznitelik çıkarımı

Adım 2: Transfer öğrenme ile tahmin modelinin oluşturulması

Transfer öğrenmede, kaynak veriden hedef veriye fayda sağlayacak bilgilerin aktarımı için transfer öğrenme aktarım başarısı tahmin problemi tanımlanmıştır. Bu tahmin probleminin girdileri, kaynak-hedef veri eşleşmelerindeki özniteliklerdir, çıktısı ise RMSE/MAE iyileşme oranlarıdır. Bu problemde, kaynak-hedef veri eşleşmeleri ele alındığı için 22 adet zaman serisi öznitelikleri hem kaynak hem de hedef veriler için ayrı ayrı tanımlanmıştır ve bu kapsamda toplam 44 adet öznitelik oluşturulmuştur. Satış ile ilgili özniteliklerden Sector_Info ve SimilarProduct dışında kalan 4 adet öznitelik hem kaynak hem hedef veriler için ayrı ayrı tanımlanmıştır ve bu kapsamda toplam 8 adet öznitelik oluşturulmuştur. Satış ile ilgili özniteliklerden Sector_Info ve SimilarProduct öznitelikleri ile uzaklık tabanlı özniteliklerden her biri, kaynak ve hedef veri eşleşmeleri için hesaplanmaktadır. Bu nedenle, bu kapsamda 4 öznitelik oluşturulmuştur. Sonuç olarak, tahmin probleminin toplam girdi (öznitelik) sayısı 56'dır (44+8+4=56).

Adım 3: Öznitelik seçimi

Transfer öğrenme aktarım başarısı tahmininde, hem yüksek tahmin doğruluğu elde etmek hem de öznitelik seçimi yapmak amacıyla rastgele orman yöntemi kullanılmıştır. Rastgele orman yönteminin uygulanması ile özniteliklerin önem dereceleri elde edilmiştir ve buna bağlı olarak öznitelik seçimi yapılmıştır.

Rastgele orman modelinin performansını değerlendirmek için R^2 ölçütü kullanılmıştır. R^2, tahmin modelinde bağımsız değişkenlerin bağımlı değişken üzerindeki varyansını açıklama oranının istatistiksel bir ölçüsüdür (Cameron ve Windmeijer, 1996). Bu değer, tahmin modelinin verileri açıklama gücünü temsil eder ve 0 ile 1 arasında bir değer alır. R^2 matematiksel olarak Denklem (3.14) ile ifade edilir (Cameron ve Windmeijer, 1996):

$$R^2 = 1 - \frac{SSE}{SST} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (3.14)$$

Burada, *SSE* hata kareleri toplamını, bir diğer ifadeyle modelin gerçek değerlerle tahmin değerleri arasındaki farkların karelerinin toplamını ifade eder. *SST* ise bağımlı değişkenin

gerçek değerleri ile ortalaması arasındaki farkların karelerinin toplamını ifade eder. Bu denklemde, y_i bağımlı değişkenin *i.* gözlem (gerçek) değerini, $\hat{y}_i$ rastgele orman yönteminin *i.* gözlem için yaptığı tahmini değeri, $\bar{y}$ bağımlı değişkenin gözlemlenen değerlerinin ortalaması ve *n* gözlem sayısını ifade eder. R^2 değeri bire ne kadar yakın ise modelin bağımlı değişkeni o ölçüde iyi açıkladığı kabul edilir (Cameron ve Windmeijer, 1996).

Rastgele orman modelinin hataları hesaplanırken, torba dışı hata (out-of-bag) kullanılmıştır (Breiman, 2001). Burada, oluşturulan her karar ağacında, modelin kullanmadığı veri noktaları için tahminler yapılır ve bu tahminlerin hatalarının ortalaması torba dışı hatayı verir. Bu hata ile rastgele orman modelinin genelleme yeteneği değerlendirilmiştir. Modelin torba dışı hatası ne kadar düşük ise modelin genelleme yeteneğinin o ölçüde iyi olduğu kabul edilir (Breiman, 2001).
Adım 2 ve Adım 3'teki alt adımlar Şekil 3.16'da gösterilmiştir.



Şekil 3.16. Öznitelik seçimi süreci

Adım 4: Benzemezlik matrisinin oluşturulması

Adım 3'te belirlenen öznitelik önem derecelerini benzemezlik matrisine yansıtmak için ağırlıklı Öklid uzaklığı (Weighted Euclidean Distance) kullanılmıştır. Ağırlıklı Öklid uzaklığında, her özniteliğin önemini/etkisini gösteren bir ağırlık değeri bulunmaktadır (Merigo ve Casanovas, 2011).

n adet özniteliğe sahip (*n* boyutlu) A ve B verileri arasındaki ağırlıklı Öklid uzaklığı Denklem (3.13) ile hesaplanır. Özniteliklerin önem dereceleri [0, 1] aralığındadır ve $\sum_{i=1}^{n} w_i = 1$ idir.

$$d_{WE}(A,B) = \sqrt{\sum_{i=1}^{n} w_i * (A_i - B_i)^2} \tag{3.13}$$

Burada, $d_{WE}(A,B)$ A ve B verileri arasındaki ağırlıklı Öklid uzaklığını, w_i i. özniteliğin önem derecesini, A_i ve B_i i. özniteliğin sırasıyla A ve B verilerindeki değerlerini ifade etmektedir (Merigo ve Casanovas, 2011). Bu tezde, rastgele orman modelinden elde edilen öznitelik önem dereceleri ağırlık değerleri olarak alınmıştır.

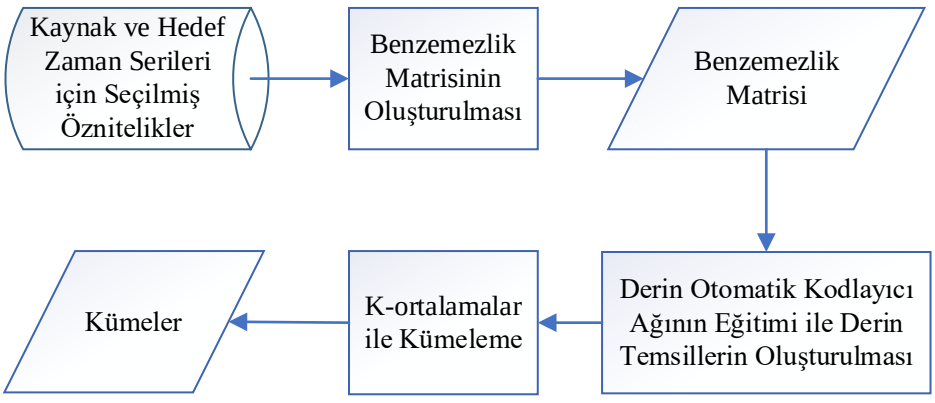
DTW ve ERP öznitelikleri her kaynak-hedef veri eşleşmesi için hesaplanır. Bu nedenle, eğer seçilen öznitelikler arasında DTW ve ERP öznitelikleri var ise benzemezlik matrisi iki adımda oluşturulur. İlk adımda, DTW ve ERP öznitelikleri dışında kalan öznitelikler ile ağırlıklı Öklid uzaklıkları Denklem (3.13) ile hesaplanır. İkinci adımda, DTW ve ERP uzaklıkları ilgili önem dereceleri ile ağırlıklandırılarak Denklem (3.14)'te gösterildiği şekilde toplam uzaklık hesaplanır.

$$d(A,B) = d_{WE}(A,B) + w_{DTW} * DTW(A,B) + w_{ERP} * ERP(A,B) \quad (3.14)$$

Burada, $d(A,B)$ A ile B arasındaki toplam uzaklığı, w_{DTW} DTW özniteliğinin önem derecesini, $DTW(A,B)$ A ile B arasındaki DTW uzaklığını, w_{ERP} ERP özniteliğinin önem derecesini ve $ERP(A,B)$ A ile B arasındaki ERP uzaklığını ifade etmektedir.

Adım 5: Zaman serilerinin kümelenmesi

Adım 4'te oluşturulan benzemezlik matrisi girdi olarak kullanılmıştır ve DAE tabanlı derin kümeleme uygulanmıştır. Adım 4 ve Adım 5'teki alt adımlar Şekil 3.17'de gösterilmiştir.



Şekil 3.17. Benzemezlik matrisinin oluşturulması ve zaman serilerinin kümelenmesi

Adım 6: Kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin oluşturulması

Elde edilen kümeler, Silhouette, Calinski-Harabasz ve Dunn küme doğrulama indekslerine göre değerlendirilmiştir. Burada, her indeks için en yüksek değere sahip küme sayısı ele alınmıştır. Kümeler birbirine benzer satış verilerini içermektedir, bu nedenle küme içindeki verilere çok kaynaklı transfer öğrenme uygulanmıştır. Şöyle ki, verilen bir hedef verisinin içinde bulunduğu kümede yer alan diğer veriler bu hedef verisine benzemektedir ve bu veriler transfer öğrenmede kaynak olarak kullanılmaktadır.

Çok kaynaklı transfer öğrenmede, verilen hedef verisi için nihai tahmin sonuçları topluluk öğrenme yöntemleri ile elde edilir. Bir başka deyişle, her bir kaynak veri ile *N* adet LSTM modeli eğitilir. Bu kapsamda, çok kaynaklı torbalama ve yığınlama topluluk öğrenme modelleri geliştirilmiştir. Geliştirilen algoritmaların sözde kodları sırasıyla Şekil 3.18 ve Şekil 3.19'da verilmiştir. Bu yaklaşım, kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı olarak adlandırılmaktadır. Bu yaklaşıma ilişkin akış şeması Şekil 3.20'de verilmiştir.

Algoritma 4: Torbalama tabanlı çok kaynaklı derin transfer öğrenme algoritması

Girdi: $C = \{C_1, C_2, \dots, C_k\}$ k adet küme, $y_h = \{y_{ht}: t \in T_h\}$ hedef zaman serisi; N topluluk sayısı

Çıktı: Torbalama topluluk modeli

Adım 1. Eğer $y_h \in C_i$ ve $\{y_1, y_2, \dots, y_p\} \in C_i$ ise, p adet kaynak satış serisi ve hedef satış verisine [0, 1] normalizasyonu uygulanır.

Adım 2. Kaynak veri kümesi ile LSTM kaynak modeli geliştirilir ve eğitilir.

Adım 3. *Adım 2*'de ön eğitilmiş LSTM ağı ve ağın ağırlık değerleri alınır. Hedef veri kümesine ağ yapısı değiştirilmeden parametre transferi yaklaşımı ile TL uygulanır.

Adım 4. *Adım 2* ve *Adım 3 N* kez tekrarlanarak *N* adet LSTM-TL modeli oluşturulur.

Adım 5. Her bir kaynak veri kümesi (toplam p adet kaynak veri) için *Adım 2-4* uygulanır.

Adım 6. Tüm kaynak veri kümeleri ile elde edilen her bir LSTM-TL modelinde (toplam $N*p$ adet model) hedef test verisi tahmin edilir ve tahmin değerlerinin ortalamasının alınması suretiyle topluluk modelin sonuçları elde edilir.

Şekil 3.18. Torbalama tabanlı çok kaynaklı derin transfer öğrenme algoritmasının sözde kodu

Algoritma 5: Yığınlama tabanlı çok kaynaklı derin transfer öğrenme

Girdi: $C = \{C_1, C_2, \dots, C_k\}$ k adet küme, $y_h = \{y_{ht}: t \in T_h\}$ hedef zaman serisi; N topluluk sayısı

Çıktı: Yığınlama topluluk modeli

Adım 1. Eğer $y_h \in C_i$ ve $\{y_1, y_2, \dots, y_p\} \in C_i$ ise, p adet kaynak satış serisi ve hedef satış verisine [0, 1] normalizasyonu uygulanır.

Adım 2. Kaynak veri kümesi ile LSTM kaynak modeli geliştirilir ve eğitilir.

Adım 3. *Adım 2*'de ön eğitilmiş LSTM ağı ve ağın ağırlık değerleri alınır. Hedef veri kümesine ağ yapısı değiştirilmeden parametre transferi yaklaşımı ile TL uygulanır.

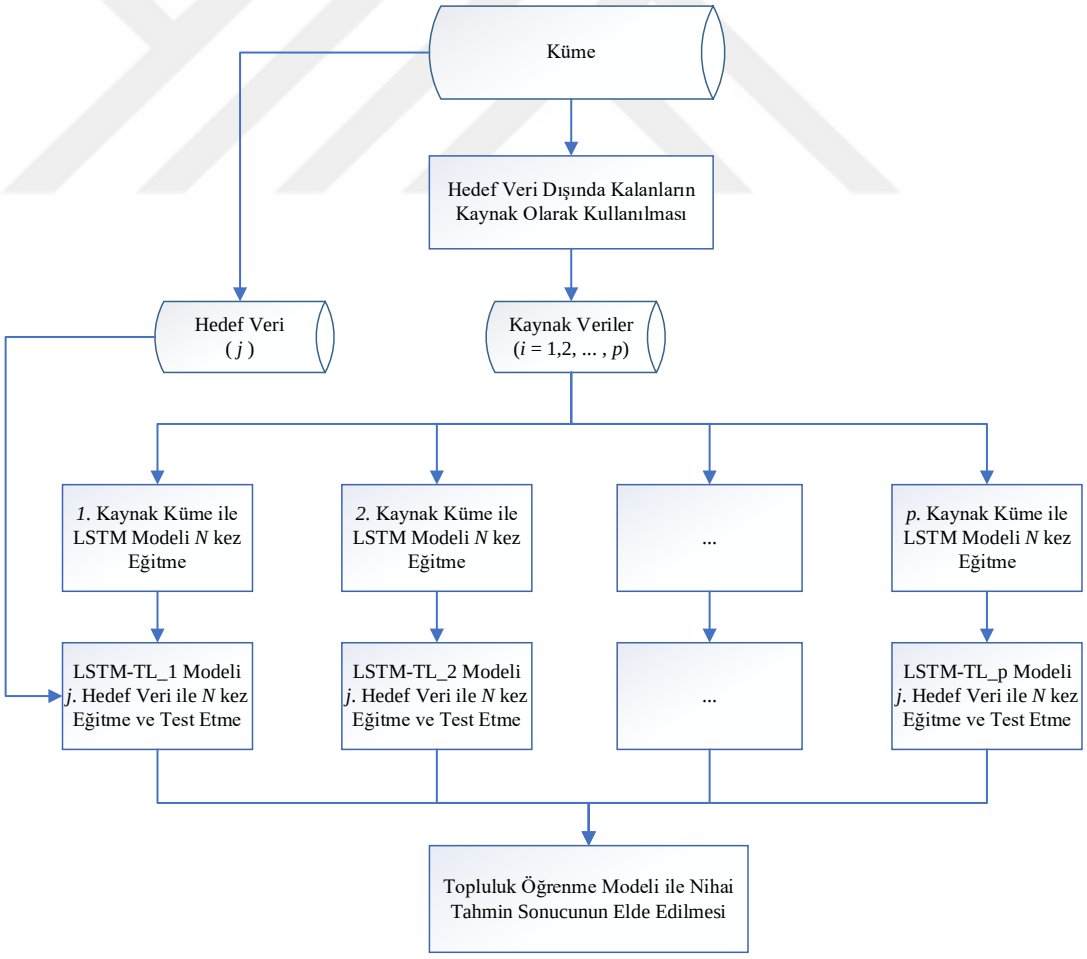
Adım 4. *Adım 2* ve *Adım 3* N kez tekrarlanarak N adet LSTM-TL modeli oluşturulur.

Adım 5. Her bir kaynak veri kümesi (toplam p adet kaynak veri) için *Adım 2-4* uygulanır.

Adım 6. Tüm kaynak veri kümeleri ile elde edilen her bir LSTM-TL modelinde (toplam $N*p$ adet model) hedef eğitim verisi tahmin edilir ve eğitim tahmin değerleri bir meta öğrenicide (destek vektör regresyonu) eğitilerek yığın topluluk modeli oluşturulur.

Adım 7. Yığın topluluk modeli ile hedef test verisi tahmin edilir.

Şekil 3.19. Yığınlama tabanlı çok kaynaklı derin transfer öğrenme algoritmasının sözde kodu



Şekil 3.20. Kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin akış şeması

Adım 7: Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin oluşturulması

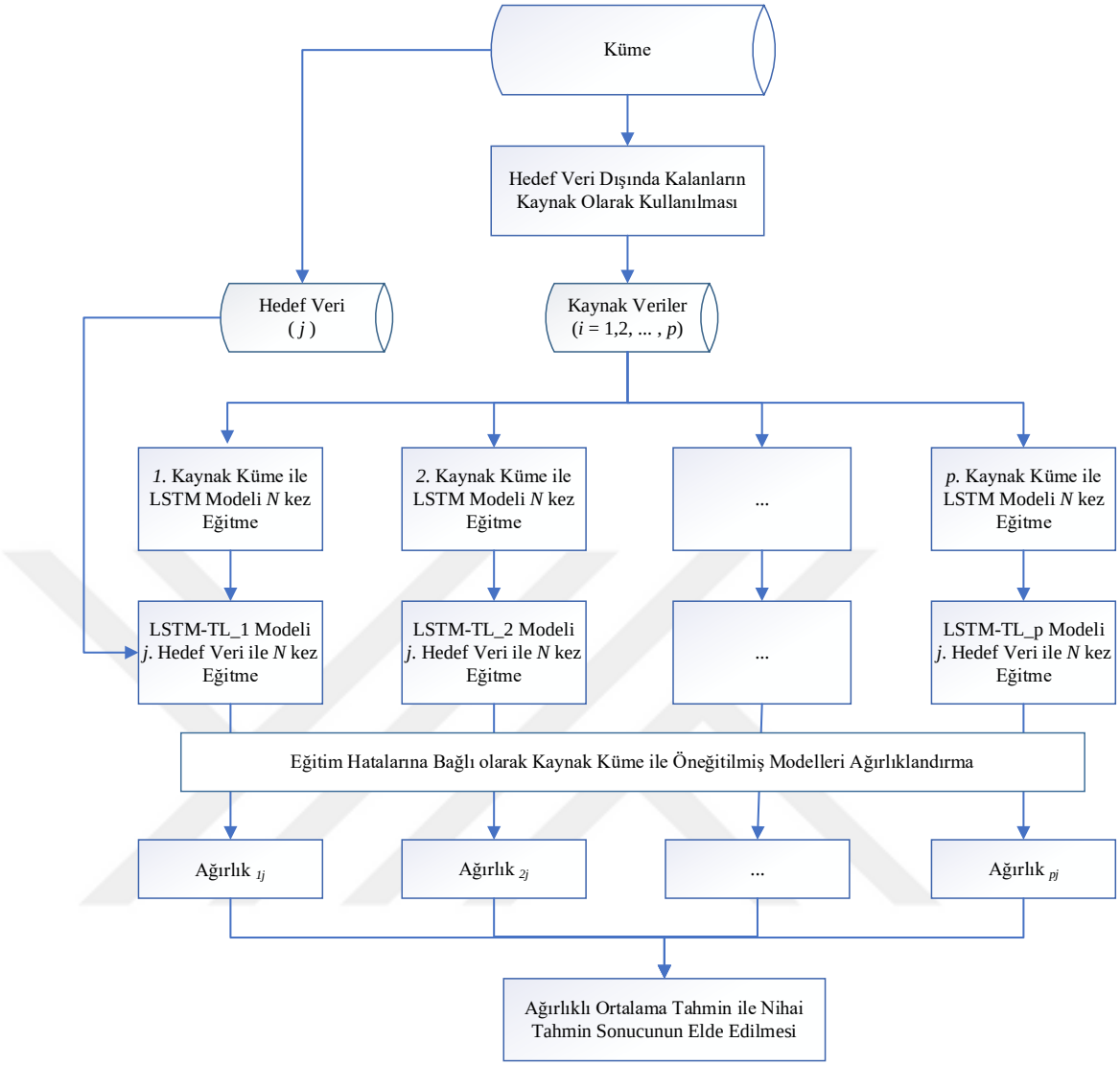
Tahmin doğruluğunu arttırmak için çok kaynaklı transfer öğrenmede hedef verideki eğitim hatalarına bağlı bir ağırlıklandırma stratejisi geliştirilmiştir. Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin TL model yaklaşımı olarak adlandırılan bu yaklaşım, hedef veri kümesi için aynı kümede yer alan tüm verilerin eğitim hatalarına göre ağırlıklandırılarak kaynak olarak kullanılmasını esas almaktadır. Bu yaklaşımda, Denklem seti (3.15)-(3.16)'da verilen ağırlıklandırma mekanizması ile torbalama yöntemi kullanılmıştır.

$$oran_{ij} = 1 - \left(\frac{e_{ij}}{max_l(e_{il})}\right), \qquad i = 1,2,\dots,p;\ j = 1,2,\dots,n \tag{3.15}$$

$$ağırlık_{ij} = \frac{oran_{ij}}{\sum_{i=1}^{p}\sum_{j=1}^{n} oran_{ij}}, \qquad \forall i,j \tag{3.16}$$

Burada; e_{ij}, *i.* kaynak veriye ait *j.* modelin eğitim hatasını; $oran_{ij}$, *i.* kaynak veriye ait *j.* modelin maksimum eğitim hatasına göre topluluktaki katkı oranını; $ağırlık_{ij}$ ise *i.* kaynak veriye ait *j.* modelin topluluktaki önemini ifade etmektedir. Denklem (3.16), toplulukta yer alan tüm kaynak modellerin katkı oranlarının [0-1] aralığına normalize edilmesini sağlamaktadır. Elde edilen model ağırlıkları ile model tahminlerinin ağırlıklı ortalaması alınarak nihai tahmin sonuçları elde edilmiştir.

Bu yaklaşıma ilişkin akış şeması Şekil 3.21'de verilmiştir.



Şekil 3.21. Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin akış şeması

3.6. Çok Değişkenli Zaman Serileri için Önerilen Çözüm Yaklaşımları

Bu bölümde, satış verileri çok değişkenli zaman serileri olarak ele alınmıştır. Bölüm 3.6.1'de uzaklık tabanlı derin transfer öğrenme yöntemi geliştirilmiştir. Bölüm 3.6.2'de, tek kaynaklı topluluk tabanlı derin transfer öğrenme yöntemleri sunulmuştur. Bölüm 3.6.3'te ise tek kaynaklı derin transfer öğrenme yöntemi, çok kaynaklı derin transfer öğrenme yöntemine genişletilmiştir.

3.6.1. Uzaklık tabanlı derin transfer öğrenme

Bu yöntemde, çok değişkenli hedef satış verisine en benzer çok değişkenli kaynak satış verisi ile derin transfer öğrenme uygulayarak tahmin doğruluğunu arttırmak amaçlanmıştır. Çok değişkenli zaman serileri arasındaki uzaklıkları hesaplamak için Bölüm 3.3.3'de açıklanan DTW_D uzaklık ölçütü kullanılmıştır. Tahmin modeli olarak LSTM ağı geliştirilmiştir. Bu yaklaşım, çok değişkenli LSTM tabanlı derin transfer öğrenme yaklaşımı (M-DTW_D-LSTM-TL) olarak adlandırılmıştır.

Kaynak veri kümesi $X_K = \{x_1, x_2, \dots, x_n\}$, n adet potansiyel kaynak satış verisinden oluşmaktadır ve l. potansiyel kaynak satış verisinde $x_l = \{x_{ltf} \mid t = 1, 2, \dots, T_l; f = 1, 2, \dots, m\}$, T_l adet dönem ve m adet öznitelik bulunmaktadır. Hedef satış verisi $y_h = \{y_{htf} \mid t = 1, 2, .., T_h; f = 1,2, \dots, m\}$, T_h adet dönem ve m adet öznitelik içeren satış verisini ifade etmektedir. y_{htf}, T_h dönem boyunca h. hedef satış verisinin t zaman adımındaki f. özniteliğin gözlem değerini ifade etmektedir. Hem kaynak hem de hedef satış verileri aynı öznitelik kümesine başka bir ifadeyle aynı özniteliklere sahiptir. Buna göre, ilk olarak, potansiyel kaynak ve hedef satış verileri [0, 1] aralığına normalize edilir. Sonrasında, hedef ve potansiyel kaynak verileri arasında DTW_D uzaklığı hesaplanır ve hedef veriye en küçük uzaklığa sahip (en benzer) satış verisi kaynak veri olarak seçilir. Ardından, seçilen kaynak veri ile kaynak LSTM ağı eğitilir. Son olarak, parametre transferi uygulanır ve ön eğitilmiş kaynak LSTM ağı (LSTM-TL) hedef veri ile tekrar eğitilir. Böylece, hedef veri için LSTM tahmin modeli elde edilir. M-DTW_D-LSTM-TL algoritması Şekil 3.22'de verilmiştir.

Algoritma 6: M-DTW$_D$-LSTM-TL algoritması

Girdi: $X_K = \{x_1, x_2, \dots, x_n\}$ kaynak veri kümesi ve $y_h = \{y_{htf} \mid t = 1, 2, \dots, T_h; f = 1,2, \dots, m\}$ hedef veri
Çıktı: Hedef veri için satış tahmini modeli
Yöntemin adımları:
Adım 1. Kaynak veri kümesi X_K ve hedef satış verisi y_h, [0, 1] aralığına normalize edilir.
Adım 2. Hedef veri ve potansiyel kaynak veriler arasındaki DTW$_D$ uzaklığı $DTW_D(x_l, y_h), \forall x_l \in X_K$ hesaplanır.
Adım 3. Hedef veriye en küçük DTW$_D$ uzaklığına sahip kaynak veri x_{l^*}, $l^* = \underset{l}{\operatorname{argmin}}\{DTW_D(x_l, y_h)\}$, seçilir.
Adım 4. Kaynak LSTM modeli, seçilen kaynak veri x_{l^*} ile hiperparametre optimizasyonu yapılarak eğitilir.
Adım 5. Kaynak LSTM modelindeki hiperparametre ve ağırlıklar hedef LSTM modeline aktarılır. Böylece LSTM-TL modeli elde edilir.
Adım 6. Hedef LSTM modeli, hedef veri y_h ile eğitilir.

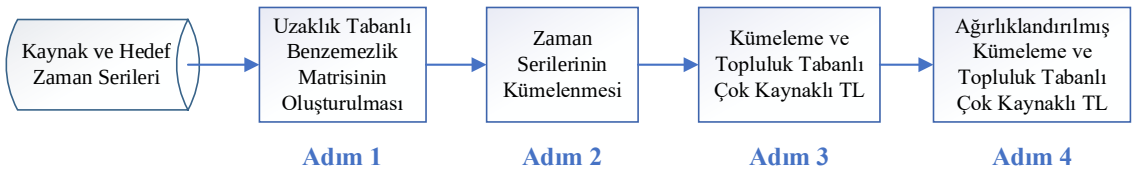
Şekil 3.22. M-DTW$_D$-LSTM-TL algoritması

3.6.2. Topluluk tabanlı derin transfer öğrenme

Bu bölümde, Şekil 3.12 ve Şekil 3.13'te sözde kodları verilen torbalama ve yığınlama topluluk öğrenme algoritmaları çok değişkenli zaman serilerinde uygulanmıştır.

3.6.3. Kümeleme ve topluluk tabanlı derin transfer öğrenme

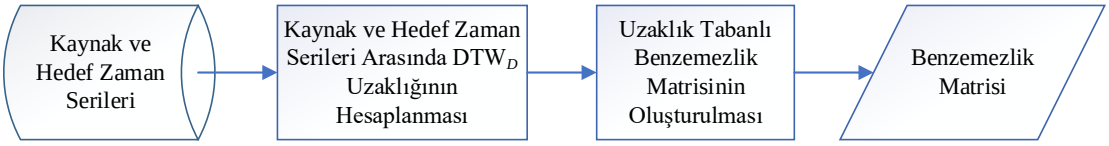
Çok değişkenli zaman serileri için geliştirilen kümeleme ve topluluk tabanlı derin transfer öğrenme metodolojisinin akış şeması Şekil 3.23'te verilmiştir. Ayrıca, bu metodolojinin adımları aşağıda açıklanmıştır:



Şekil 3.23. Çok değişkenli zaman serileri için kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımının akış şeması

Adım 1: Uzaklık tabanlı benzemezlik matrisinin oluşturulması

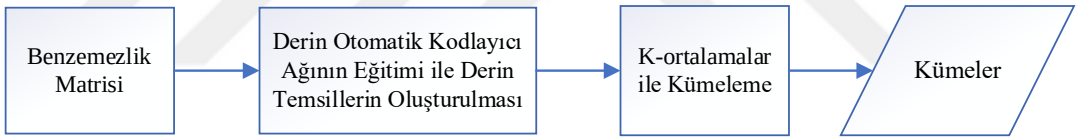
Bu adımda, çok değişkenli kaynak ve hedef satış verileri arasındaki DTW_D uzaklığı hesaplanmıştır. DTW_D uzaklıkları kullanılarak seriler arasındaki uzaklık tabanlı benzemezlik matrisi oluşturulmuştur. Buradaki alt adımlar Şekil 3.24'te gösterilmektedir.



Şekil 3.24. Benzemezlik matrisinin oluşturulması

Adım 2: Zaman serilerinin kümelenmesi

Adım 1'de elde edilen benzemezlik matrisi girdi olarak kullanılarak kaynak ve hedef satış verilerine DAE tabanlı derin kümeleme yöntemi uygulanmıştır. Adım 2'deki alt adımlar Şekil 3.25'te gösterilmektedir.



Şekil 3.25. Zaman serilerinin kümelenmesi

Adım 3: Kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin oluşturulması

Bu adımda, Bölüm 3.5.3'te tek değişkenli zaman serileri için verilen Adım 6 çok değişkenli zaman serilerine uygulanmıştır. Bu kapsamda, Şekil 3.20'de yer alan alt adımlar ile Şekil 3.18-3.19'da verilen topluluk algoritmaları kullanılmıştır. Bu algoritmaların girdileri, çok değişkenli kaynak ve hedef veri kümeleridir.

Adım 4: Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin TL modelinin oluşturulması

Bu adımda, Bölüm 3.5.3'te tek değişkenli zaman serileri için verilen Adım 7 çok değişkenli zaman serilerine uygulanmıştır. Bu kapsamda, Şekil 3.21'de yer alan alt adımlar takip edilmiştir.

4. BULGULAR ve TARTIŞMA

Tez çalışmaları kapsamında yapılan deneysel çalışmalar ve sonuçları bu bölümde verilmektedir. Bölüm 4.1'de deneysel koşullar ve parametre optimizasyonu açıklanmıştır. Bölüm 4.2'de derin öğrenme yöntemlerinin karşılaştırılmasına dair deneysel çalışmalar ve sonuçları verilmiştir. Bölüm 4.3'te tek değişkenli zaman serileri için deneysel çalışmalar, Bölüm 4.4'te çok değişkenli zaman serileri için deneysel çalışmalar, Bölüm 4.5'te bu çalışmaların karşılaştırmaları ve Bölüm 4.6'da ise tartışma verilmiştir.

4.1. Deneysel Koşullar ve Parametre Optimizasyonu

Tez kapsamında geliştirilen yöntemler ile karşılaştırma amacıyla uygulanan istatistiksel, derin öğrenme ve makine öğrenmesi yöntemleri Google Colab ile Anaconda Spyder ortamında Python programlama dilinde kodlanmıştır. Zaman serileri için yapılan benzerlik analizleri ve öznitelik çıkarma işlemleri R programında R programlama dilinde kodlanmıştır. Deneyler Intel (R) Core (TM) i7-11800HQ CPU 4.60 GHz 64,0 GB RAM özelliklerine sahip bir bilgisayarda yapılmıştır.

Derin öğrenme ve transfer öğrenme yöntemleri için *tensorflow* ve *keras*, makine öğrenme yöntemleri için *scikit-learn*, istatistiksel yöntemler için *statsmodels* ve benzerlik hesapları için *tslearn* ve *dtaidistance* Python kütüphaneleri kullanılmıştır. Ek olarak, Python'da *math, numpy, pandas, matplotlib* ve *seaborn* kütüphaneleri kullanılmıştır. R programında öznitelik çıkarımı için *tsfeatures*, benzerlik hesapları için ise *factoextra* kütüphaneleri kullanılmıştır.

Deneysel çalışmalarda kullanılan ve Çizelge 3.1'de verilen tüm satış verilerine [0, 1] normalizasyonu uygulanmıştır. Her satış verisinin %64'ü eğitim, %16'sı doğrulama ve %20'si test verisi olarak ayrılmıştır.

Tez kapsamında gerçekleştirilen ilk deneysel çalışmada derin öğrenme yöntemleri kıyaslanmıştır. Bu kapsamda Bölüm 2.1'de anlatılan dokuz adet yöntem kullanılmıştır. Derin öğrenme yöntemlerinde kayıp fonksiyonu olarak ortalama karesel hata (Mean

Squarred Error – MSE) ve optimizasyon algoritması olarak Adam (Kingma ve Ba, 2021) kullanılmıştır.

Hiperparametre optimizasyonu ile her satış verisi ve her yöntem için en uygun hiperparametre değerleri belirlenmiştir. Bu kapsamda, Çizelge 4.1'de yer alan hiperparametreler için ızgara arama yöntemi kullanılarak hiperparametre optimizasyonu yapılmıştır. Derin öğrenme yöntemlerinde başlangıç ağırlıkları ve sapma değerleri rastgele seçilmiştir. Farklı başlangıç ağırlıkları ve sapma değerlerinden kaynaklanan rassallık nedeniyle her veri kümesi için derin öğrenme yöntemleri ile otuzar tekrar yapılmıştır. Tekrarların ortalaması alınarak nihai tahmin sonucu elde edilmiştir.

Tez kapsamında gerçekleştirilen ikinci deneysel çalışma, tek değişkenli satış verileri için uzaklık tabanlı derin transfer öğrenme yöntemi (ERP-LSTM-TL) ile yapılan karşılaştırmaları içermektedir. Bu kapsamda, ilk olarak transfer öğrenmenin ve uzaklık tabanlı kaynak seçiminin etkisini görmek amacıyla LSTM, LSTM-TL ve ERP-LSTM-TL modelleri karşılaştırılmıştır. Bu üç modelde hiperparametre optimizasyonu için Çizelge 4.1'de verilen değerler kullanılmıştır. Modellerde kayıp fonksiyon olarak MSE ve optimize edici olarak Adam (Kingma ve Ba, 2014) kullanılmıştır. Hiperparametre optimizasyonu nedeniyle her veri kümesi için elde edilen LSTM mimarisi farklıdır. LSTM-TL ve ERP-LSTM-TL modellerindeki dönem sayısı (epoch), LSTM modelinin dönem sayısından %50 daha azdır. Ayrıca, ERP-LSTM-TL yöntemi istatistiksel tahmin yöntemleri ve makine öğrenmesi yöntemleri ile kıyaslanmıştır. İstatistiksel tahmin yöntemlerinden basit hareketli ortalama (Simple Moving Average – SMA), basit üstel düzeltme (Simple Exponential Smoothing – SES), lineer (doğrusal) regresyon (Linear Regression – LR) ve otoregresif bütünleşik hareketli ortalama (Autoregressive Integrated Moving Average – ARIMA) yöntemleri kullanılmıştır. Makine öğrenmesi yöntemlerinden ise k-en yakın komşuluk regresyonu (k-Nearest Neighbor Regression – KNN-R), destek vektör regresyonu (Support Vector Regression – SVR), regresyon ağacı (Regression Tree – RT) ve yapay sinir ağı (Artificial Neural Network – ANN) yöntemleri ele alınmıştır. İstatistiksel tahmin ve makine öğrenmesi yöntemleri için Çizelge 4.2'de verilen parametreler ve değer aralıkları ile ızgara arama yöntemi uygulanarak en uygun parametre değerleri belirlenmiştir.

Çizelge 4.1. Derin öğrenme modellerindeki hiperparametreler

Yöntem(ler)	Hiperparametreler	Değer Aralığı
DNN, RNN, LSTM, GRU, Bi-LSTM	Girdi boyutu	[1-370]
	Gizli katman sayısı	{2, 3, 4, 5}
	Gizli katman nöron sayısı	{4, 8, 16, 32, 64, 128, 256}
	Öğrenme oranı	{0,001; 0,01; 0,1}
	Parti (Batch) büyüklüğü	{1, 2, 4, 8, 16, 32, 64}
	Dönem (Epoch) sayısı	{25, 30, 40, 50, 75, 100, 150, 250}
	Aktivasyon Fonksiyonu	{tanh, ReLu}
DAE	Gizli katman sayısı (Kodlayıcı-Kod çözücü katman sayıları)	{1, 2, 3, 4, 5}
	Gizli katman nöron (kod) sayısı	{4, 8, 16, 32, 64, 128, 256}
	Öğrenme oranı	{0,001; 0,01; 0,1}
	Parti büyüklüğü	{1, 2, 4, 8, 16, 32, 64}
	Dönem sayısı	{25, 30, 40, 50, 75, 100, 150, 250}
	Aktivasyon Fonksiyonu	{tanh, ReLu}
CNN	Evrişim katmanı sayısı	{1, 2, 3}
	Kernel boyutu	{3, 5, 7}
	Filtre boyutu	{3, 4, 5, 7, 8, 16, 32, 64, 128, 256}
	Havuzlama	{Maksimum, Ortalama}
	Havuzlama boyutu	{2, 3, 4, 5, 6}
	Öğrenme oranı	{0,001; 0,01; 0,1}
	Parti büyüklüğü	{1, 2, 4, 8, 16, 32, 64}
	Dönem sayısı	{25, 30, 40, 50, 75, 100, 150, 250}
	Aktivasyon Fonksiyonu	{tanh, ReLu}
CNN-LSTM, ConvLSTM	Girdi boyutu	[1-370]
	Gizli katman sayısı	{2, 3, 4, 5}
	Gizli katman nöron sayısı	{4, 8, 16, 32, 64, 128, 256}
	Evrişim katmanı sayısı	{1, 2, 3}
	Kernel boyutu	{3, 5, 7}
	Filtre boyutu	{3, 4, 5, 7, 8, 16, 32, 64, 128, 256}
	Havuzlama	{Maksimum, Ortalama}
	Havuzlama boyutu	{2, 3, 4, 5, 6}
	Alt dizilerin sayısı	{2, 3, 4, 5, 6}
	Alt dizideki zaman adımı sayısı	{2, 3, 4, 5, 7, 12, 52}
	Öğrenme oranı	{0,001; 0,01; 0,1}
	Parti büyüklüğü	{1, 2, 4, 8, 16, 32, 64}
	Dönem sayısı	{25, 30, 40, 50, 75, 100, 150, 250}
	Aktivasyon Fonksiyonu	{tanh, ReLu}

Çizelge 4.2. Karşılaştırmalı değerlendirme yöntemlerinin parametreleri

Yöntem	Parametreler ve Değer Aralıkları
SMA	Geçmiş [1-10] dönem ortalaması
SES	$0 \leq \alpha \leq 1$ (α: üstel düzeltme katsayısı)
ARIMA	$p \in \{0, 1, 2, 3, 4\}$, $q \in \{0, 1, 2, 3, 4\}$, $d \in \{0, 1, 2, 3\}$ (p: otoregresif model derecesi, q: hareketli ortalama model derecesi, d: fark alma derecesi).
KNN-R	$1 \leq k \leq 20$ (k: komşu sayısı)
RA	Maksimum derinlik ∈ {3, 5, 10, 15}, Maksimum yaprak düğümü ∈ {3, 5, 7, 10, 15}, Yapraktaki minimum örnek sayısı = 1, Bölünme için minimum örnek sayısı = 2.
SVR	$C \in \{0{,}01, 0{,}05, 0{,}1, \ldots, 1, \ldots, 200, 500, 1000\}$, $\varepsilon \in \{0{,}0001, 0{,}0005, \ldots, 0{,}01, \ldots, 8, 10\}$, $\gamma \in \{0{,}0001, 0{,}001, 0{,}005, \ldots, 0{,}1, \ldots, 7, 10\}$, Kernel çekirdek fonksiyonu ∈ {radyal tabanlı fonksiyon, polinom, sigmoid, doğrusal}, (C: düzenlileştirme katsayısı, γ: dağılım genişliği kontrol parametresi, ε: duyarsız Laplace kayıp fonksiyonu parametresi).
ANN	Girdi boyutu ∈ {1, 10, 25, 50, 75, 100}, Gizli katman sayısı (ANN) = 1, Gizli katman nöron sayıları ∈ {1, 10, 25, 50, 75, 100, 150, 200}, Öğrenme oranı ∈ {0,1; 0,01; 0,001}, Parti (Batch) büyüklüğü = 1, Dönem (Epoch) sayısı ∈ {25, 50, 100, 150, 200}, Aktivasyon Fonksiyonu ∈ {tanh, ReLU}, Kayıp fonksiyonu: MSE ve optimizasyon algoritması: Adam.

Tez kapsamında gerçekleştirilen üçüncü ve dördüncü deneysel çalışma, sırasıyla tek ve çok değişkenli zaman serileri için geliştirilen çözüm yaklaşımlarına ait deneysel çalışmaları içermektedir. Bu kapsamda tek değişkenli zaman serileri için geliştirilen metodolojide öznitelik seçimi için rastgele orman yöntemi kullanılmıştır. Rastgele orman yönteminde Çizelge 4.3'te belirtilen parametreler ve değer aralıkları ile ızgara arama yöntemi uygulanarak en uygun parametre değerleri belirlenmiştir. Rastgele orman yönteminin performansını değerlendirmek için Bölüm 3.5.3'te açıklanan torba dışı hata ve R^2 ile Bölüm 3.4'te açıklanan RMSE ile MAE ölçütleri kullanılmıştır. Derin kümeleme yönteminde DAE ve k-ortalamalar yöntemleri için Çizelge 4.4'te verilen hiperparametreler ve değer aralıkları kullanılarak ızgara arama yöntemi ile parametre optimizasyonu yapılmıştır. Derin kümeleme yönteminde küme kalitesini ölçmek için Bölüm 2.3.2'de anlatılan Silhouette, Calinski-Harabasz ve Dunn indeksleri kullanılmıştır.

Çizelge 4.3. Rastgele orman hiperparametreleri

Parametre	Değer Aralığı
Modeldeki ağaç sayısı (n_estimators)	[100, 300, 500, 700, 1000]
Ağacın maksimum derinliği (max_depth)	[2, 4, 5, 8, 10, 12, None]
Maksimum öznitelik sayısı (max_features)	["sqrt", "log2", None]
Yaprak düğümündeki minimum örnek sayısı (min_samples_leaf)	[1, 2, 3, 4, 5, 6, 7, 8]
Bölme için minimum örnek sayısı (min_samples_split)	[2, 3, 4, 5, 6, 7, 8]
Bölme kriteri (criterion)	[‘friedman_mse’, ‘absolute_error’, ‘poisson’, ‘squared_error’"]

Çizelge 4.4. DAE tabanlı derin kümeleme hiperparametreleri

Yöntem	Hiperparametreler	Değer Aralığı
DAE	Temsil alan boyutu	{2, 3, 4, 5, 6}
	Gizli katman sayısı (Kodlayıcı-Kod çözücü katman sayıları)	{1, 2, 3, 4, 5}
	Gizli katman nöron (kod) sayısı	{4, 8, 16, 32, 64, 128, 256}
	Öğrenme oranı	{0,001; 0,01; 0,1}
	Parti büyüklüğü	{1, 2, 4, 8, 16, 32, 64}
	Dönem sayısı	{50, 75, 100, 150, 250, 500}
	Aktivasyon fonksiyonu	{tanh, ReLu}
	Kayıp fonksiyonu	MSE
	Optimize edici	ADAM
k-ortalamalar	Küme sayısı	{2, 3, 4, 5, 6, 7}
	Başlangıç küme merkezlerini belirleme	Açgözlü k-means++
	Tekrar sayısı	10

Topluluk yöntemlerini içeren deneylerde, LSTM modellerinin eğitimi farklı başlangıç ağırlıkları ve sapma (bias) değerleri ile gerçekleştirilmiştir ve böylece ön eğitilmiş LSTM-TL modelleri elde edilmiştir. Elde edilen ön eğitilmiş LSTM-TL modelleri ile hedef veriye transfer öğrenme gerçekleştirilerek topluluk modelleri oluşturulmuştur. Hedef verisi için seçilen kaynak veri sayısının (p) 1 ile 5 arasında değiştiği gözlenmiştir. Kaynak veri sayısının 2 ile 4 arasında olduğu durumda, topluluktaki tekrar sayısı (N) bu değerlerin en küçük ortak katı alınarak ($N = 12$) belirlenmiştir. Kaynak veri sayısının $p = 5$ olduğu durumda ise topluluktaki tekrar sayısı ($N = 5$) olarak seçilmiştir. Topluluktaki toplam model sayısı maksimum 50 olarak sınırlandırılmıştır.

LSTM yönteminde girdi (parti büyüklüğü, zaman adımı, öznitelik sayısı) bilgilerini içeren 3-boyutlu tensör şeklinde ifade edilir (Chollet, 2021). Bu tezde, satış verileri için Çizelge 3.2'teki öznitelikler kullanılmıştır. Bu nedenle, tek değişkenli LSTM yönteminde öznitelik sayısı 1 olarak alınmıştır, çok değişkenli LSTM yönteminde ise öznitelik sayısı 5 olarak alınmıştır.

Bu tez kapsamında, tek değişkenli ve çok değişkenli zaman serileri için aşağıda belirtilen dört yaklaşım kıyaslanmıştır:

1. Topluluk tabanlı tek kaynaklı derin transfer öğrenme yaklaşımı: Bu yaklaşımda bir hedef veri kümesi için benzemezlik matrisine göre en yakın zaman serisi kaynak olarak seçilmektedir. Burada hem torbalama hem de yığınlama topluluk yöntemleri kullanılmıştır.

2. K-En Yakın Komşuluk (k-Nearest Neighbor – KNN) ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Bir hedef veri kümesinin benzemezlik matrisine göre k-en yakın komşuları (k=2 ve k=3) kaynak veri olarak seçilmektedir. Bu yaklaşımda hem torbalama hem de yığınlama topluluk yöntemleri kullanılmıştır.

3. Kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Bir hedef veri kümesi için aynı kümede yer alan tüm verilerin kaynak olarak kullanılmasına dayanan yaklaşımdır. Bu yaklaşımda hem torbalama hem de yığınlama topluluk yöntemleri kullanılmıştır. Bu yaklaşım, Bölüm 3.5.3'te Adım 6'da açıklanmıştır.

4. Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Bir hedef veri kümesi için aynı kümede yer alan tüm verilerin eğitim hatalarına göre ağırlıklandırılarak kaynak olarak kullanılmasına dayanan yaklaşımdır. Bu yaklaşımda sadece torbalama topluluk yöntemi kullanılmıştır. Bu yaklaşım, Bölüm 3.5.3'te Adım 7'de açıklanmıştır.

Transfer öğrenmenin etkisini görebilmek amacıyla yukarıdaki yaklaşımlara ek olarak her hedef verisine transfer öğrenme uygulamadan LSTM yöntemi kullanılarak torbalama topluluk yöntemi uygulanmıştır ve sonuçlar raporlanmıştır.

Tahmin yöntemleri için performansların karşılaştırılmasında Bölüm 3.4'te yer alan RMSE ile MAE ölçütleri kullanılmıştır. Ayrıca, yöntemlerin eğitim süreleri de karşılaştırılmıştır. Modellerin eğitim süresi Denklem (4.1) ile hesaplanmaktadır.

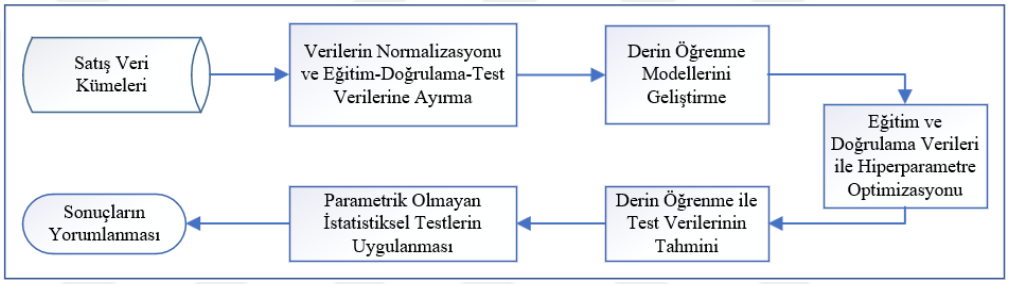
$$t_{ij} = \left(\sum_{k=1}^{p} t_k * N\right) + (t_{hk} * N * p) + t_j \; ; \; \forall i, j \tag{4.1}$$

Burada t_{ij} *i.* modelin *j.* topluluk yöntemi için toplam eğitim süresini, t_k *k.* kaynak verisi ile eğitilen LSTM modelinin eğitim süresini, t_{hk} *h.* hedef verisi için *k.* kaynak verisi ile ön eğitilmiş LSTM-TL modelinin eğitim süresini ve $N * p$ topluluktaki toplam kaynak model sayısını ve t_j *j.* topluluk yönteminin eğitim süresini ifade etmektedir. Torbalama topluluk yönteminde herhangi bir eğitim süreci olmadığı için bu süre sıfır olarak alınmıştır. Yığınlama topluluk yönteminde meta-öğrenici olan SVR yönteminin eğitim süresi dikkate alınmıştır. Ağırlıklandırılmış kümeleme ve topluluk tabanlı derin transfer öğrenmede torbalama topluluk yöntemi kullanılmıştır ve ağırlıklandırma sürecinin süresi dikkate alınmıştır. Transfer öğrenmenin uygulanmadığı LSTM yöntemi için toplam eğitim süresi $t_{ij} = \left(\sum_{k=1}^{p} t_h * N\right)$ ile hesaplanmaktadır.

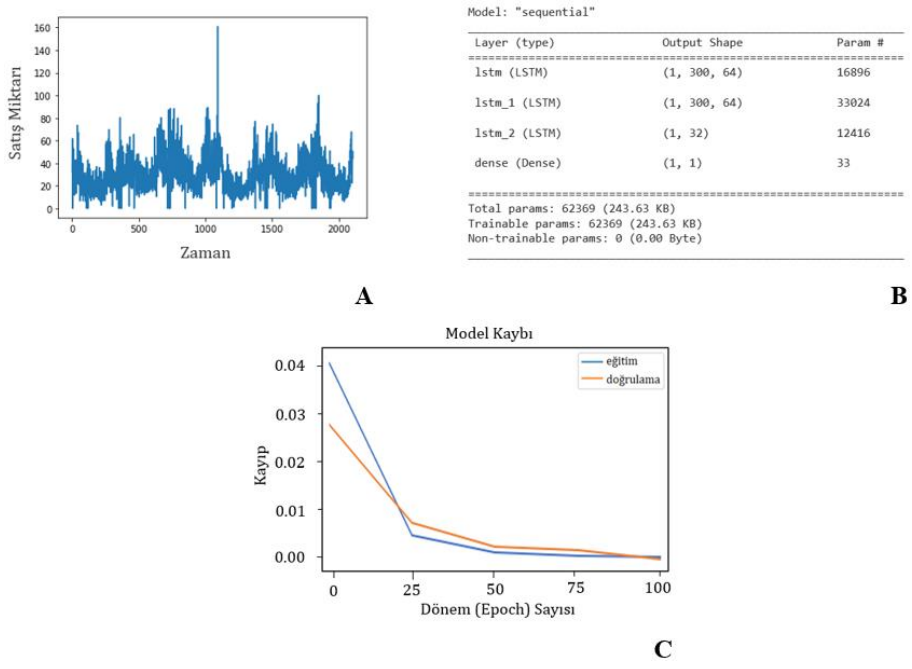
4.2. Derin Öğrenme Yöntemlerinin Karşılaştırılması

Derin öğrenme yöntemlerinin karşılaştırılması için yapılan çalışmada Şekil 4.1'de verilen akış takip edilmiştir. Bu bölümde, Çizelge 3.1'deki Supermarket verisi hariç diğer veriler kullanılmıştır. İlk adımda, her bir satış verisi normalize edildikten sonra eğitim, doğrulama ve test verilerine ayrılmıştır. İkinci adımda, her satış verisi için dokuz adet derin öğrenme modeli geliştirilmiştir. Üçüncü adımda, eğitim ve doğrulama verileri kullanılarak ızgara arama yöntemiyle hiperparametre optimizasyonu gerçekleştirilmiştir. Dördüncü adımda, her satış verisine ait model mimarileri ile test verileri üzerinde tahminler gerçekleştirilmiştir. Son adımda, her bir satış verisindeki derin öğrenme yöntemlerinin performansları arasında istatistiksel olarak anlamlı bir farklılık olup olmadığını belirlemek adına parametrik olmayan istatistiksel testler yapılmıştır ve test sonuçları yorumlanmıştır.

Hiperparametre optimizasyonu yapılması nedeniyle her derin öğrenme yönteminde her satış verisi için farklı ağ mimarileri oluşturulmuştur. Örneğin, Şekil 4.2 A'da grafiği verilen tek değişkenli Pharma veri kümesinde hiperparametre optimizasyonu sonucunda elde edilen tek değişkenli LSTM ağı mimarisi Şekil 4.2 B'de ve kayıp fonksiyonunun dönem sayısına göre değişimi Şekil 4.2 C'de gösterilmiştir. Bu durumda Pharma veri kümesi için Şekil 4.2 B'de görüldüğü üzere iki gizli ve bir çıktı katmanında oluşan bir LSTM modeli oluşturulmuştur. Bu modelde girdi (1, 300, 1) şeklindedir, parti büyüklüğü 1, zaman adımı 300 ve öznitelik sayısı 1'dir. Bu LSTM modeline ait Şekil 4.2 C'de verilen kayıp fonksiyonuna göre eğitim ve doğrulama verileri için model kaybının sıfıra yakınsaması eğitimin başarılı bir şekilde gerçekleştiğini ifade etmektedir.



Şekil 4.1. DL yöntemlerinin performanslarının karşılaştırılması için akış şeması



Şekil 4.2: Pharma veri kümesine ait grafikler
A) Veri setinin zamana bağlı grafiği **B)** Pharma verisi için oluşturulan LSTM model mimarisi **C)** Dönem sayısına göre kayıp fonksiyonundaki değişim

4.2.1. Deney sonuçları

Her satış verisine derin öğrenme yöntemlerinin uygulanması sonucunda elde edilen deneysel sonuçlar Çizelge 4.5'te verilmiştir. Çizelge 4.5'te test verisi üzerindeki ortalama RMSE ve MAE değerleri cinsinden tahmin performansları ve eğitim süreleri verilmiştir. Test süresi her bir yöntemde ortalama 0,4 saniyedir. Çizelge 4.5'te her veri kümesindeki en iyi performans ölçütleri koyu olarak işaretlenmiştir. Buna göre yedi veri kümesi içinden Walmart, Rossmann, Tv_Sales ve Pharma olmak üzere dört veri kümesinde hem RMSE hem MAE performans ölçütlerine göre en iyi sonuçlar LSTM yöntemi ile elde edilmiştir. Forklift ve LearnX olmak üzere iki veri kümesinde en iyi sonuçları GRU yöntemi, Favorita veri kümesinde ise en iyi sonucu ConvLSTM yöntemi vermiştir.

Eğitim süresi yönünden CNN ve DAE yöntemleri tüm veri kümelerinde en düşük eğitim sürelerine sahipken Bi-LSTM ve LSTM yöntemleri en uzun eğitim sürelerine sahiptir.

Derin öğrenme yöntemlerinin RMSE ve MAE açısından tahmin performansları Çizelge 3.1'de yer alan veri özellikleri ile ele alındığında aşağıda yer alan çıkarımlar yapılmıştır:

- Mevsimselliğe sahip veri kümelerinden Walmart verisinde LSTM, DNN ve DAE; LearnX verisinde GRU, Bi-LSTM ve DAE; Rossmann verisinde LSTM, CNN-LSTM ve CNN; Favorita verisinde ise ConvLSTM, CNN-LSTM ve CNN en yüksek performans sıralamasında ilk üç sırada yer alan yöntemlerdir.
- Sadece trend içeren Tv_Sales verisinde en iyi üç performansa sahip yöntemler LSTM, DNN ve GRU yöntemleridir.
- Hem mevsimsellik hem de trend içeren Pharma verisinde LSTM, CNN-LSTM ve GRU yöntemleri en iyi üç performansa sahip yöntemlerdir.
- Hem trend içeren hem de seyrek bir veri kümesi olan Forklift verisinde en iyi üç performansa sırasıyla GRU, ConvLSTM ve RNN yöntemleri ile elde edilmiştir.

Sonuçlara göre, mevsimsellik içeren verilerde LSTM ile LSTM içeren hibrit yöntemlerin daha yüksek performansa sahip olduğu görülmektedir. Trend içeren verilerde RNN, LSTM ve GRU yöntemleri gibi tekrarlayan sinir ağı modelleri daha başarılı sonuçlar

vermiştir. Hem mevsimsellik hem trend içeren verilerde ise LSTM ve türevlerinin daha yüksek performansa sahip olduğu görülmektedir. Rossmann, Favorita ve Pharma gibi mevsimsellik içeren büyük verilerde CNN ile LSTM ağlarından oluşan hibrit ağların tahmin doğruluğunu iyileştirdiği görülmektedir. Buna göre, LSTM yönteminin uzun dönemli zamansal bağımlılıkları öğrenebilme özelliği sayesinde verideki mevsimsellik ve trendi keşfedebildiği görülmektedir.

Örnek olarak, Pharma test verisindeki en iyi üç tahmin performansına sahip derin öğrenme yöntemi (sırasıyla LSTM, CNN-LSTM ve GRU) ele alınmıştır. Bu yöntemlerin tahmin değerleri ile gerçek satış değerleri Şekil 4.3'te verilmiştir. Bu grafiğe göre, gerçek satış miktarlarına en yakın tahmin değerlerinin LSTM ile elde edildiği görülmektedir. Ek olarak, Pharma verisindeki mevsimsellik ve trend içeren örüntünün LSTM, CNN-LSTM ve GRU modelleri ile başarılı bir şekilde yakalandığı görülmektedir.

4.2.2. Parametrik olmayan istatistiksel test sonuçları

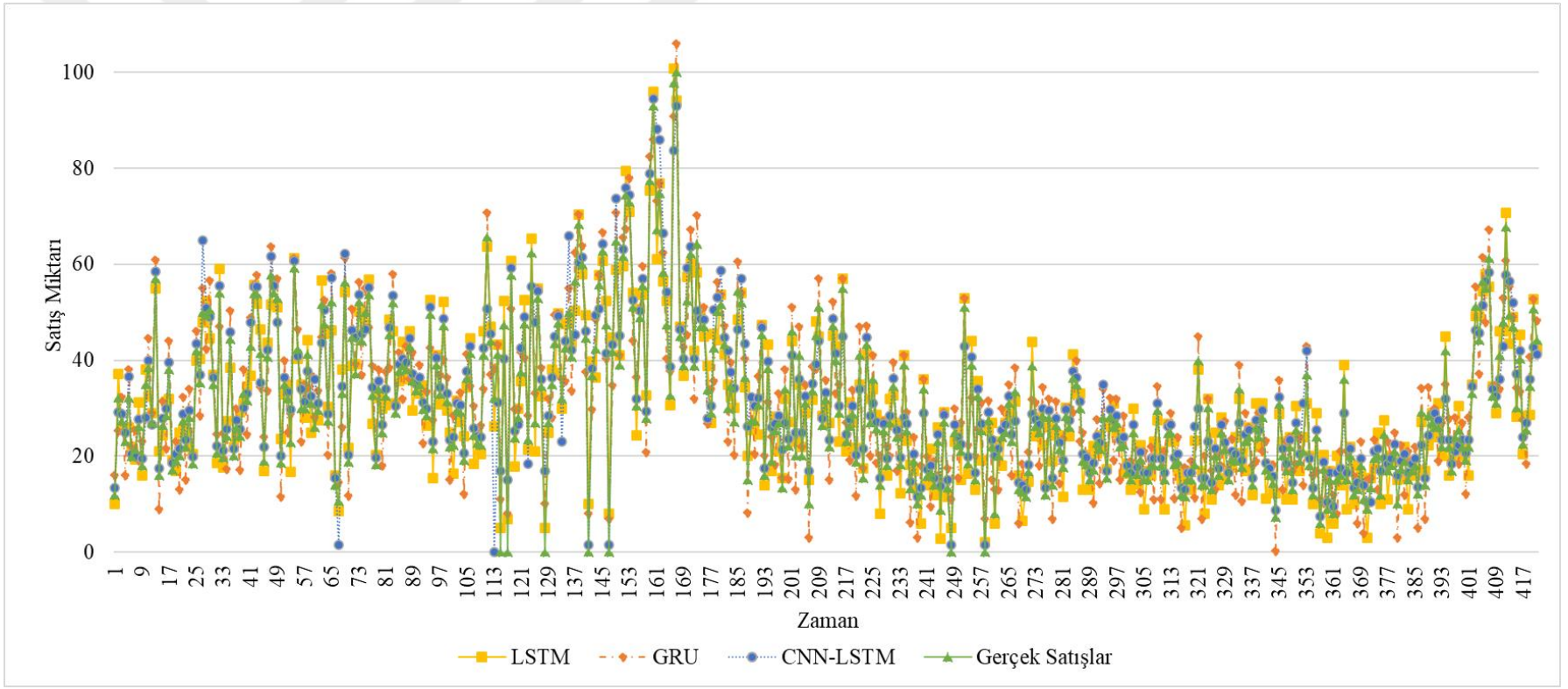
Bu çalışmada, derin öğrenme yöntemlerinin performansları arasındaki farkların istatistiksel olarak anlamlılığını değerlendirmek için parametrik olmayan istatistiksel testler kullanılmıştır. Parametrik olmayan istatistiksel testlerde varyans homojenliği ve normallik gibi varsayımlar bulunmamaktadır. Ayrıca yöntemlerin birden fazla veri kümesindeki performansları karşılaştırılabilmektedir (Demsar, 2006).

Bu çalışmada ele alınan derin öğrenme yöntemlerinin hem ikili hem de çoklu karşılaştırmaları gerçekleştirilmiştir (Demsar, 2006). Çoklu karşılaştırmalarda Friedman post-hoc testi ve Nemenyi post-hoc testi uygulanmıştır (Demsar, 2006). İkili karşılaştırmalarda Wilcoxon eşleştirilmiş işaretli sıra testi ile kazanç, kayıp ve eşitlik sayıları (counts of wins, losses and ties) işaret testi uygulanmıştır (Demsar, 2006).

Çizelge 4.5. Derin öğrenme yöntemlerinin test verilerindeki performansları

Veri Kümesi	Ölçüt	Yöntemler								
		DNN	DAE	CNN	RNN	LSTM	GRU	BiLSTM	CNN-LSTM	ConvLSTM
Walmart-Seri_1	RMSE	90 493,18	96 416,28	96 436,58	97 770,50	**90 000,71***	96 745,35	111 716,18	109 141,18	111 478,22
	MAE	74 722,67	76 422,03	76 644,88	77 141,59	**71 829,10***	76 689,38	91 796,85	90 151,42	90 988,57
	Süre	79,95	71,23	**62,12***	82,51	91,24	84,77	96,22	88,26	87,97
Forklift-Seri_26	RMSE	0,88	0,75	0,72	0,52	0,69	**0,44***	0,54	0,75	0,50
	MAE	0,82	0,73	0,71	0,50	0,65	**0,42***	0,53	0,72	0,48
	Süre	64,02	55,03	**50,11***	71,25	75,14	67,98	74,18	72,77	73,15
LearnX-Seri_4	RMSE	25,12	23,88	28,89	27,16	28,18	**23,05***	23,48	24,03	23,74
	MAE	20,44	19,15	23,61	22,49	23,07	**18,88***	18,95	19,02	18,95
	Süre	255,84	210,45	**187,52***	298,12	312,55	287,14	315,48	259,74	241,77
Rossmann-Seri_15	RMSE	1 868 434,87	2 234 172,35	1 816 163,10	1 831 218,14	**1 758 434,87***	1 981 191,27	2 220 472,35	1 766 852,71	1 852 350,90
	MAE	1 372 280,57	1 774 952,23	1 299 796,43	1 352 312,45	**1 282 280,57***	1 421 701,77	1 748 692,23	1 284 538,54	1 361 196,98
	Süre	579,10	476,23	**399,28***	522,04	715,23	602,48	752,89	571,77	501,29
Tv_Sales-Seri_14	RMSE	1438,72	1556,46	2463,75	1590,08	**1 379 42***	1442,65	1849,25	2001,14	1584,55
	MAE	790,83	811,77	1848,47	907,21	**715,58***	793,41	1225,41	1402,23	902,88
	Süre	776,41	**654,23***	683,79	975,65	986,79	874,50	1003,25	746,87	757,57
Favorita-Seri_16	RMSE	12 093,45	11 487,66	11 065,27	12 138,09	13 771,48	12 969,86	13 648,80	10 547,63	**9 832,65***
	MAE	9 012,31	7 896,36	7 785,25	9 102,45	9 124,39	9 114,23	9 174,13	6 578,93	**5 140,25***
	Süre	745,22	685,33	**601,45***	786,55	855,49	802,25	897,63	798,87	1150,26
Pharma-Seri_17	RMSE	14,01	13,54	16,41	13,42	**11,47***	13,15	13,76	12,82	14,79
	MAE	12,44	11,25	13,87	11,21	**9,61***	11,20	11,28	10,11	12,74
	Süre	633,08	558,25	**491,44***	697,55	715,25	657,63	714,87	685,22	675,43

* En başarılı performansları göstermektedir.



Şekil 4.3: Pharma-Seri_17 test verisinde en başarılı ilk üç derin öğrenme yönteminin tahmin performanslarının karşılaştırılması

Friedman testi için kurulan hipotezler şu şekildedir: (i) H_0: Derin öğrenme yöntemlerinin performansları arasında farklılık bulunmamaktadır. (ii) H_1: En az bir derin öğrenme yönteminin performansında farklılık bulunmaktadır. Friedman testi için Demsar'ın (2006) çalışmasında önerilen Iman ve Davenport istatistiği (1980) uygulanmıştır. Bu istatistiğe ait sonuçlar Çizelge 4.6'da verilmektedir. Ele alınan ölçütler için F tablo değeri hesaplanan F değerinden (F_F) küçük olduğu için H_0 hipotezleri reddedilmektedir. Bu durumda RMSE, MAE performans metriklerine ve eğitim süresine göre derin öğrenme yöntemleri arasında istatistiksel olarak anlamlı bir farklılık olduğu görülmektedir.

Çizelge 4.6. Friedman testleri sonuçları

Performans Kriteri	Hesaplanan F-Değeri (F_F)	F-Tablo Değeri ($F_{\alpha=0,05,\ k-1=8,\ (k-1)*(N-1)=48}$)	Sonuç
RMSE	162,00	2,14	$F_F > F_{0,05;\,8;\,48}$, H_0 hipotezi reddedilir.
MAE	162,00	2,14	
Eğitim Süresi	97,70	2,14	

Friedman testleri sonrasında istatistiksel olarak anlamlı farka sahip olan derin öğrenme yöntemlerinin tespit edilmesi amacıyla Nemenyi testi uygulanmıştır. Bu testte her yöntemin ortalama sıralama değeri (OSD) belirlenir. İki yöntemin karşılaştırılması için OSD'leri arasındaki fark hesaplanır ve bu fark kritik fark değeriyle kıyaslanır. Eğer iki yöntem arasındaki OSD farkı, kritik farktan büyük olursa bu yöntemler arasında istatistiksel olarak anlamlı bir fark bulunduğu sonucu çıkarılır.

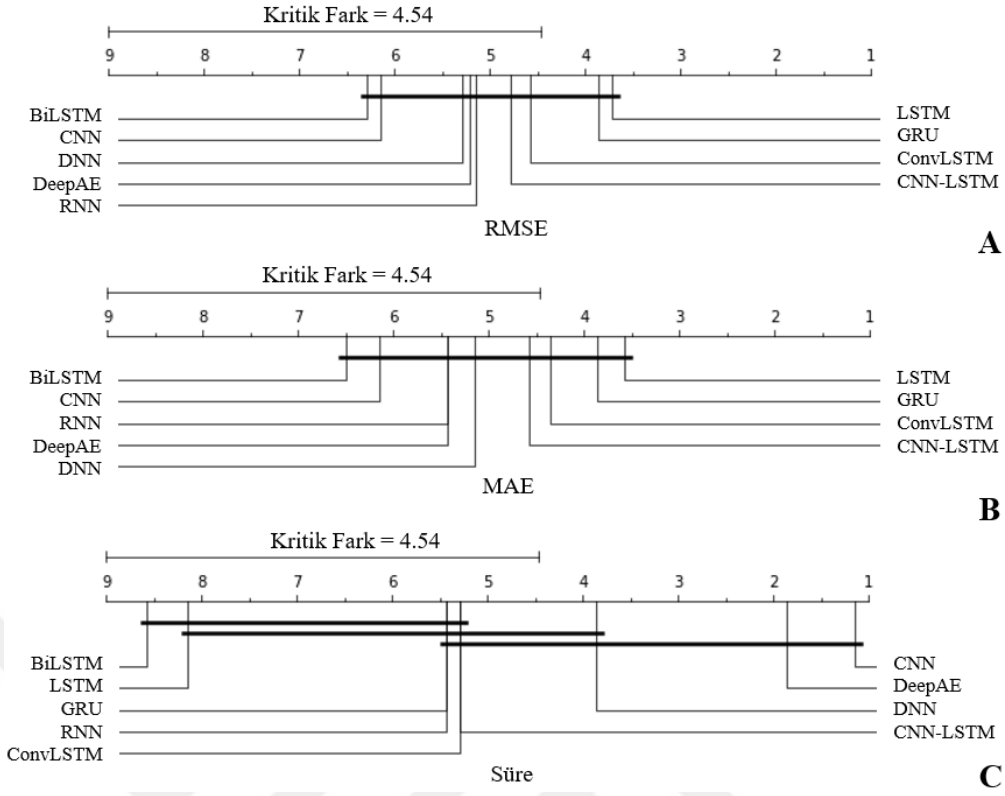
Derin öğrenme yöntemlerinin RMSE, MAE ve eğitim süresine göre OSD'leri sırasıyla Şekil 4.4 A, B ve C'de yer almaktadır. Kritik fark %5 anlamlılık düzeyinde 4,54 olarak hesaplanmıştır. Derin öğrenme yöntemleri Şekil 4.4'te aynı doğru parçası üzerinde OSD'ye göre sıralanarak gösterilmiştir. Buna göre, küçük OSD'ye sahip yöntemler diyagramın sağında, büyük OSD'ye sahip yöntemler ise diyagramın solunda yer almaktadır.

Şekil 4.4 A ve B'de incelenen derin öğrenme yöntemleri arasında RMSE ve MAE'ye göre anlamlı bir fark olmadığı görülmektedir. OSD büyüklüklerine göre değerlendirme yapıldığında ise Şekil 4.4 A ve B'de diyagramların sağında yer alan LSTM, GRU, ConvLSTM ve CNN-LSTM yöntemlerine ait OSD'ler, diyagramın solunda yer alan Bi-

LSTM, CNN, DNN, DeepAE ve RNN yöntemlerinin OSD'lerinden daha küçüktür. Bu nedenle, diyagramın sağındaki yöntemler diyagramın solundaki yöntemlerden daha iyi performansa sahiptir. Örnek olarak, LSTM yöntemi RMSE ve MAE ölçütlerine göre en küçük OSD'lere (sırasıyla 3,72 ve 3,58) sahiptir, ikinci sırada ise GRU yöntemi (sırasıyla 3,86 ve 3,77) gelmektedir. Bu sıralama, LSTM ve GRU gibi uzun dönemli bağımlılıkları öğrenebilen ve tekrarlayan sinir ağlarının tahmin doğruluğunda öne çıktığını göstermektedir. RMSE ve MAE ölçütlerine göre en büyük OSD'lere sahip iki yöntem olan Bi-LSTM (sırasıyla 6,36 ve 6,50) ve CNN (sırasıyla 6,20 ve 6,22) diğer yöntemlere kıyasla daha düşük performansa sahiptir.

Şekil 4.4 C'de eğitim süresine göre değerlendirme yer almaktadır. Buna göre, en küçük üç OSD'nin sırasıyla 1,17; 1,85 ve 3,87 olduğu ve bu değerlerin CNN, DAE ve DNN yöntemlerine ait olduğu görülmektedir. CNN (OSD = 1,17) yönteminin eğitim süresi, Bi-LSTM (OSD = 8,58) ile LSTM (OSD = 8,21) yöntemlerinin eğitim sürelerinden; DAE (OSD = 1,85) yönteminin eğitim süresi ise Bi-LSTM ile LSTM yöntemlerinin eğitim sürelerinden anlamlı olarak daha kısadır. Ayrıca DNN (OSD = 3,87) yöntemi, Bi-LSTM yöntemine göre istatistiksel olarak daha kısa sürede eğitilmiştir. Hesaplamalara örnek vermek gerekirse CNN (OSD = 1,17) ile Bi-LSTM (OSD = 8,58) yöntemlerinin OSD'leri arasındaki fark 8,58 – 1,17 = 7,41'dir ve bu fark kritik fark değerinden (7,41 > 4,54) büyüktür. Bu sebeple, süre açısından CNN yönteminin performansı Bi-LSTM yönteminin performansından istatistiksel olarak üstündür. Diğer tüm yorumlar benzer hesaplamalara göre yapılmıştır. Diğer yöntemlerin arasında süre açısından istatistiksel olarak anlamlı fark bulunmamaktadır.

Derin öğrenme yöntemlerinin ikili karşılaştırmalarında Wilcoxon eşleştirilmiş işaretli sıra testi kullanılmıştır ve bu amaçla kurulan tek taraflı hipotezler şu şekildedir: (i) H_0: İki derin öğrenme yönteminin performans ölçütü eşit veya birinci derin öğrenme yönteminin performans ölçütü, ikinci derin öğrenme yönteminin performans ölçütünden daha büyüktür. (ii) H_1: Birinci derin öğrenme yönteminin performans ölçütü, ikinci derin öğrenme yönteminin performans ölçütünden daha küçüktür. Burada, performans ölçütü olarak RMSE, MAE ve eğitim süresi kullanılmıştır.



Şekil 4.4: Nemenyi testi sonuç diyagramları
A) RMSE diyagramı **B)** MAE diyagramı **C)** Eğitim süresi diyagramı

Ele alınan derin öğrenme yöntemlerinin ikili karşılaştırma sonuçları Çizelge 4.7'de verilmektedir. Çizelge 4.7'deki satırlar hipotezdeki birinci derin öğrenme yöntemini, sütunlar ise hipotezdeki ikinci derin öğrenme yöntemini ifade etmektedir. Buna göre, H_0 hipotezi reddedilen ikili karşılaştırmalar %5 anlamlılık düzeyinde ($\alpha = 0{,}05$) ** işareti ile %10 anlamlılık düzeyinde ($\alpha = 0{,}10$) ise * işareti ile gösterilmiştir. Bu tabloya göre aşağıdaki gözlemler yapılmıştır:

- Tekrarlayan sinir ağlarından RNN ve GRU yöntemleri tüm ölçütlere göre Bi-LSTM'den istatistiksel olarak üstün performansa sahiptir. LSTM ağı, MAE ve eğitim süresine göre, ConvLSTM yöntemi ise RMSE ve MAE ölçütlerine göre Bi-LSTM'den daha iyi sonuç vermiştir. DNN yöntemi ise MAE ölçütüne göre RNN'den daha iyi performansa sahiptir.
- CNN, diğer tüm derin öğrenme yöntemlerinden; DAE ise CNN dışındaki tüm derin öğrenme yöntemlerinden istatistiksel olarak daha kısa eğitim süresine sahiptir.

- Eğitim süresi açısından DNN yöntemi tekrarlayan sinir ağlarından (RNN, LSTM, GRU ve Bi-LSTM); RNN, GRU ve CNN-LSTM yöntemleri ise LSTM ve Bi-LSTM yöntemlerinden istatistiksel olarak daha iyi performans göstermiştir.
- Diğer ikili karşılaştırmalarda istatistiksel olarak anlamlı bir farklılık bulunmamaktadır.

Çizelge 4.7. Wilcoxon testlerine ait p-değerleri

Yöntemler	Ölçüt	DNN	DAE	CNN	RNN	LSTM	GRU	BiLSTM	CNN-LSTM	ConvLSTM
DNN	RMSE		0,31	0,43	0,25	0,80	**0,09***	**0,09***	0,75	0,57
	MAE		0,31	0,37	**0,03****	0,88	**0,09***	**0,09***	0,75	0,63
	Süre		0,99	0,99	**0,06***	**0,01****	**0,01****	**0,01****	0,12	0,43
DAE	RMSE	0,69		0,43	0,25	0,84	0,69	0,25	0,60	0,57
	MAE	0,69		0,37	0,25	0,84	0,69	0,37	0,75	0,57
	Süre	**0,01****		0,95	**0,01****	**0,01****	**0,01****	**0,01****	**0,01****	**0,01****
CNN	RMSE	0,57	0,57		0,25	0,94	0,31	0,25	0,88	0,57
	MAE	0,63	0,63		0,31	0,94	0,31	0,25	0,84	0,57
	Süre	**0,01****	**0,05****		**0,01****	**0,01****	**0,01****	**0,01****	**0,01****	**0,01****
RNN	RMSE	0,75	0,75	0,75		0,84	0,63	**0,03****	0,69	0,43
	MAE	0,97	0,75	0,69		0,88	0,69	**0,03****	0,69	0,80
	Süre	0,94	0,99	0,99		**0,01****	0,69	**0,01****	0,63	0,75
LSTM	RMSE	0,20	0,16	**0,06***	0,16		0,20	0,16	0,16	0,20
	MAE	0,12	0,16	**0,06***	0,12		0,16	**0,05****	0,16	0,20
	Süre	0,99	0,99	0,99	0,99		0,99	**0,03****	0,99	0,88
GRU	RMSE	0,91	0,31	0,69	0,37	0,80		**0,01****	0,50	0,37
	MAE	0,91	0,31	0,69	0,31	0,84		**0,01****	0,57	0,37
	Süre	0,99	0,99	0,99	0,31	**0,01****		**0,01****	0,75	0,57
BiLSTM	RMSE	0,91	0,75	0,75	0,97	0,85	0,99		0,88	0,94
	MAE	0,91	0,63	0,75	0,97	0,95	0,99		0,88	0,96
	Süre	0,99	0,99	0,99	0,99	0,97	0,99		0,99	0,88
CNN-LSTM	RMSE	0,25	0,40	0,12	0,31	0,84	0,50	0,12		0,37
	MAE	0,25	0,25	0,16	0,31	0,84	0,43	0,12		0,43
	Süre	0,88	0,99	0,99	0,37	**0,01****	0,25	**0,01****		0,57
ConvLSTM	RMSE	0,43	0,43	0,43	0,57	0,80	0,63	**0,06***	0,63	
	MAE	0,37	0,43	0,43	0,20	0,80	0,63	**0,04****	0,57	
	Süre	0,57	0,99	0,99	0,25	0,12	0,43	0,12	0,43	

* %90 güven aralığında H_0 hipotezi reddedilen testler koyu olarak * ile işaretlenmiştir ($\alpha = 0{,}10$).
** %95 güven aralığında H_0 hipotezi reddedilen testler koyu olarak ** ile işaretlenmiştir ($\alpha = 0{,}05$).

Yukarıdaki istatistiksel testlere ilave olarak derin öğrenme yöntemlerinin karşılaştırılması amacıyla kazanç, kayıp ve eşitlik sayıları (counts of wins, losses and ties) için işaret testi uygulanmıştır (Demsar, 2006). Bu test, bir yöntemin diğer bir yönteme göre üstün performans gösterdiği veri kümelerinin sayısını dikkate alır (Demsar, 2006). Kazanç, kayıp ve eşitlik sayıları için kurulan hipotezler şu şekildedir: (i) H_0: İki derin öğrenme yöntemi eşit sayıda veri kümesinde iyi performansa sahiptir (performansları eşdeğerdir). (ii) H_1: Bir derin öğrenme yöntemi diğer derin öğrenme yönteminden en az w_α veri

kümesinde daha iyi performansa sahiptir. H_0 hipotezinde karşılaştırılan iki yöntem eşdeğer ise her biri N veri kümesinden N/2 kadarını kazanmalıdır. Bir veri kümesinde berabere kalan eşleşmeler (eşitlik durumu) iki yöntem arasında eşit olarak bölünür (Demsar, 2006). Kazanç sayısının dağılımı binom dağılımı olduğunda yedi adet veri kümesi için %95 ($\alpha = 0,05$) ve %90 ($\alpha = 0,10$) güven aralığında kritik kazanç değerleri sırasıyla yedi ve altıdır (Demsar, 2006).

Derin öğrenme yöntemleri için kazanç, kayıp ve eşitlik sayıları Çizelge 4.8'de verilmektedir. Çizelge 4.8'de satırdaki ve sütundaki yöntemler karşılaştırılan yöntemleri göstermektedir. Buna göre, %95 güven aralığında aralarında istatistiksel olarak anlamlı fark olan derin öğrenme yöntemleri (kazanç sayısı 7) ** işareti ile, %90 güven aralığında aralarında istatistiksel olarak anlamlı fark olan derin öğrenme yöntemleri ise (kazanç sayısı 6) * işareti ile belirtilmiştir.

Çizelge 4.8. Kazanç, kayıp ve eşitlik sayıları (kazanç/kayıp/eşitlik)

Yöntem	Ölçüt	DNN	DAE	CNN	RNN	LSTM	GRU	BiLSTM	CNN-LSTM	ConvLSTM
DNN	RMSE	0/0/7	3/4/0	4/3/0	4/3/0	2/5/0	4/3/0	4/3/0	2/5/0	3/4/0
	MAE	0/0/7	3/4/0	4/3/0	5/2/0	2/5/0	4/3/0	4/3/0	2/5/0	3/4/0
	Süre	0/0/7	0/7/0	0/7/0	**6/1/0***	**7/0/0****	**7/0/0****	**7/0/0****	5/2/0	4/3/0
DAE	RMSE	4/3/0	0/0/7	4/3/0	4/3/0	2/5/0	2/5/0	4/3/0	3/4/0	3/4/0
	MAE	4/3/0	0/0/7	4/3/0	4/3/0	2/5/0	2/5/0	4/3/0	2/5/0	3/4/0
	Süre	**7/0/0****	0/0/7	1/6/0	**7/0/0****	**7/0/0****	**7/0/0****	**7/0/0****	**7/0/0****	**7/0/0****
CNN	RMSE	3/4/0	3/4/0	0/0/7	3/4/0	1/6/0	3/4/0	3/4/0	2/5/0	2/5/0
	MAE	3/4/0	3/4/0	0/0/7	3/4/0	1/6/0	3/4/0	3/4/0	2/5/0	2/5/0
	Süre	**7/0/0****	**6/1/0***	0/0/7	**7/0/0****	**7/0/0****	**7/0/0****	**7/0/0****	**7/0/0****	**7/0/0****
RNN	RMSE	3/4/0	3/4/0	4/3/0	0/0/7	3/4/0	2/5/0	**6/1/0***	3/4/0	3/4/0
	MAE	2/5/0	3/4/0	4/3/0	0/0/7	3/4/0	2/5/0	**6/1/0***	3/4/0	2/5/0
	Süre	1/6/0	0/7/0	0/7/0	0/0/7	**7/0/0****	3/4/0	**7/0/0****	3/4/0	3/4/0
LSTM	RMSE	5/2/0	5/2/0	**6/1/0***	4/3/0	0/0/7	4/3/0	4/3/0	5/2/0	4/3/0
	MAE	5/2/0	5/2/0	**6/1/0***	4/3/0	0/0/7	4/3/0	5/2/0	5/2/0	4/3/0
	Süre	0/7/0	0/7/0	0/7/0	0/7/0	0/0/7	0/7/0	5/2/0	0/7/0	1/6/0
GRU	RMSE	3/4/0	5/2/0	4/3/0	5/2/0	3/4/0	0/0/7	**7/0/0****	4/3/0	5/2/0
	MAE	3/4/0	5/2/0	4/3/0	5/2/0	3/4/0	0/0/7	**7/0/0****	4/3/0	5/2/0
	Süre	0/7/0	0/7/0	0/7/0	4/3/0	**7/0/0****	0/0/7	**7/0/0****	3/4/0	4/3/0
BiLSTM	RMSE	3/4/0	3/4/0	4/3/0	1/6/0	3/4/0	0/7/0	0/0/7	3/4/0	2/5/0
	MAE	3/4/0	3/4/0	4/3/0	1/6/0	2/5/0	0/7/0	0/0/7	3/4/0	1/6/0
	Süre	0/7/0	0/7/0	0/7/0	0/7/0	2/5/0	0/7/0	0/0/7	0/7/0	1/6/0
CNN-LSTM	RMSE	5/2/0	3/4/0	5/2/0	4/3/0	2/5/0	3/4/0	4/3/0	0/0/7	3/4/0
	MAE	5/2/0	5/2/0	5/2/0	4/3/0	2/5/0	3/4/0	4/3/0	0/0/7	3/4/0
	Süre	2/5/0	0/7/0	0/7/0	3/4/0	**7/0/0****	4/3/0	**7/0/0****	0/0/7	3/4/0
ConvLSTM	RMSE	4/3/0	4/3/0	5/2/0	4/3/0	3/4/0	2/5/0	5/2/0	4/3/0	0/0/7
	MAE	4/3/0	4/3/0	5/2/0	5/2/0	3/4/0	2/5/0	5/2/0	4/3/0	0/0/7
	Süre	3/4/0	0/7/0	0/7/0	4/3/0	**6/1/0***	3/4/0	**6/1/0***	4/3/0	0/0/7

* %90 güven aralığında istatistiksel olarak anlamlı farka sahip yöntemler koyu olarak * ile işaretlenmiştir ($\alpha = 0,10$).
** %95 güven aralığında istatistiksel olarak anlamlı farka sahip yöntemleri koyu olarak ** ile işaretlenmiştir ($\alpha = 0,05$).

Çizelge 4.8'e göre, aşağıdaki gözlemler yapılmıştır:

- %95 güven aralığında GRU yöntemi tüm ölçütlere göre Bi-LSTM yönteminden, eğitim süresine göre ise LSTM yönteminden istatistiksel olarak üstündür.
- %95 güven aralığında DNN yöntemi LSTM, GRU ve Bi-LSTM yöntemlerinden; DAE yöntemi ise CNN dışındaki tüm yöntemlerden istatistiksel olarak daha kısa eğitim süresine sahiptir. Ayrıca CNN yöntemi DAE dışındaki tüm yöntemlerden; RNN ve CNN-LSTM yöntemleri ise LSTM ve Bi-LSTM'den istatistiksel olarak daha kısa sürede eğitilmiştir.
- %95 güven aralığında diğer ikili karşılaştırmalarda istatistiksel olarak anlamlı bir farklılık bulunmamaktadır.

4.2.3. Tartışma

Bu bölümde, dokuz farklı derin öğrenme yönteminin yedi adet satış veri kümesindeki performansları tartışılmıştır.

Tahmin performansına göre ele alınan yedi veri kümesinden dördünde LSTM yönteminin diğer derin öğrenme yöntemlerine kıyasla en yüksek performansa sahip olduğu gözlemlenmektedir. Öte yandan tüm veri kümelerinde en yüksek performansa sahip ilk üç yönteme bakıldığında LSTM'ye ek olarak GRU ve CNN-LSTM yöntemlerinin öne çıktığı görülmektedir. Bu durum, tahmin performansı dikkate alındığında tekrarlayan sinir ağı varyantlarının (LSTM, GRU, hibrit CNN-LSTM) diğer yöntemlere üstünlük sağladığını göstermektedir. Bu yöntemler sıralı verileri işleyerek belirli aralıklar ile geçmiş verilerin hatırlanmasını sağlamaktadır. Bu şekilde, verideki zamansal bağımlılıkların keşfedilmesi ile başarılı tahmin sonuçları elde edilebilmektedir. Diğer yandan, tahmin performansına göre her veri kümesinde en düşük performansa sahip üç yöntem arasında Bi-LSTM yöntemi yer almaktadır. Şöyle ki, Bi-LSTM Walmart-Seri_1 veri kümesinde en kötü performansa, Rossmann-Seri_15 ile Favorita-Seri_16 verilerinde en kötü ikinci performansa, Tv_Sales-Seri_14 ile Pharma verilerinde ise en kötü üçüncü performansa sahiptir. Bi-LSTM'deki karmaşık ağ yapısı modelin eğitimini

zorlaştırmaktadır. Bir diğer kritik bulgu ise derin öğrenme yöntemleri arasında tahmin performansına göre istatistiksel olarak öne çıkan bir yöntemin bulunmamasıdır.

Eğitim süresine göre değerlendirme yapıldığında CNN yönteminin diğer yöntemlerden istatistiksel olarak daha kısa eğitim süresine sahip olduğu görülmektedir. Bunun yanı sıra, DAE ve DNN yöntemlerinin de eğitim sürelerinin diğer derin öğrenme yöntemlerinden daha kısa olduğu gözlenmektedir. Bu durumda, CNN yöntemindeki ortaklama katmanı, DAE yöntemindeki otokodlayıcı özelliği boyut azaltmaya katkı sağlamaktadır. DNN yönteminde ise ağ yapısı diğer derin öğrenme yöntemlerinden daha basit niteliktedir. Tahmin performansı açısından üstünlük gösteren LSTM'nin diğer yöntemlere kıyasla daha uzun eğitim sürelerine sahip olduğu gözlenmiştir. Tahmin performansı iyi olan bir diğer yöntem olan GRU eğitim süreleri açısından LSTM'ye avantaj sağlamaktadır ancak CNN, DAE ve DNN yöntemlerine göre bu yöntemin eğitim süresi daha uzundur. Ayrıca CNN, DAE ve DNN dışında kalan yöntemler arasında eğitim süresine göre istatistiksel olarak anlamlı bir fark bulunamamıştır.

4.3. Tek Değişkenli Zaman Serileri için Deneysel Çalışmalar

Bu bölümde, tek değişkenli zaman serileri için geliştirilen çözüm yaklaşımlarının deneysel çalışmaları verilmektedir.

4.3.1. Uzaklık tabanlı derin transfer öğrenme yönteminin karşılaştırılması

Tek değişkenli zaman serileri için geliştirilen uzaklık tabanlı derin transfer öğrenme yaklaşımı (ERP-LSTM-TL) için iki farklı deneysel çalışma gerçekleştirilmiştir. İlk deneysel çalışmada, transfer öğrenmenin ve uzaklık tabanlı kaynak seçiminin etkisini görmek amacıyla LSTM, LSTM-TL ve ERP-LSTM-TL modelleri karşılaştırılmıştır. İkinci deneysel çalışmada ise ERP-LSTM-TL yönteminin performansını değerlendirmek için istatistiksel tahmin yöntemleri ve makine öğrenmesi yöntemleri ile karşılaştırma yapılmıştır.

Bu deneysel çalışmalarda Çizelge 3.1'de verilen Supermarket verisi hariç diğer tüm veriler kullanılmıştır. Bu çizelgedeki Walmart veri kümesi birden fazla bölge ve mağazaya ait farklı satış verileri içermektedir. Forklift, LearnX ve Pharma veri kümeleri ise birden fazla farklı ürünlere ait satış verileri içermektedir. Bu nedenle, bu çalışma kapsamında bu veri kümelerinden bir kaynak veri ve bir hedef veri olmak üzere iki adet satış verisi (seri) seçilmiştir. Rossmann, Tv_Sales ve Favorita veri kümeleri ise tek bir seriden oluşmaktadır. Her bir veri kümesinde kaynak ve hedef veri olarak kullanılan seriler Çizelge 4.9'da verilmiştir. Walmart, Forklift, LearnX ve Pharma veri kümelerindeki her bir hedef veriye karşılık yedi adet kaynak veri bulunmaktadır. Örneğin, Walmart-Seri_1 hedef verisi için Walmart-Seri_2, Forklift-Seri_25, LearnX-Seri_5, Tv_Sales-Seri_14, Rossmann-Seri_15, Favorita-Seri_16 ve Pharma-Seri_18 olmak üzere yedi farklı kaynak veri mevcuttur. Rossmann, Tv_Sales ve Favorita veri kümelerindeki serilerden biri hedef veri olarak alındığında ise altı adet kaynak veri vardır. Örneğin, Rossmann veri kümesindeki tek seri olan Seri_15 hedef veri olarak alındığında bu veri kümesinde başka bir seri olmadığı için aynı veri kümesinden kaynak veri seçimi tanımlı değildir. Bu nedenle, Rossmann-Seri_15 hedef verisi için Walmart-Seri_2, Forklift-Seri_25, LearnX-Seri_5, Tv_Sales-Seri_14, Rossmann-Seri_15, Favorita-Seri_16 ve Pharma-Seri_18 olmak üzere altı adet kaynak veri kullanılmıştır. Bu kapsamda, deneysel çalışmalarda toplam 46 adet kaynak ve hedef veri kombinasyonu incelenmiştir.

Çizelge 4.9. Veri kümeleri ve kullanılan seriler

Veri Kümesi	Kaynak Veri Kümesi	Hedef Veri Kümesi	Kaynak-Hedef Eşleşmesi
Walmart	Seri_2	Seri_1	7
Forklift	Seri_25	Seri_26	7
LearnX	Seri_5	Seri_4	7
Tv_Sales	Seri_14		6
Rossmann	Seri_15		6
Favorita	Seri_16		6
Pharma	Seri_18	Seri_17	7
Toplam kaynak-hedef eşleşmesi			46

Transfer Öğrenmenin Etkisi

Transfer öğrenmenin ve uzaklık tabanlı kaynak seçiminin etkisini görmek amacıyla LSTM, LSTM-TL ve ERP-LSTM-TL modelleri karşılaştırılmıştır. LSTM modelinde her bir hedef veri LSTM ile eğitilir ve transfer öğrenme yapılmamaktadır. LSTM-TL

modelinde, LSTM önce bir kaynak veri ile eğitilir, ardından bu ön eğitilmiş LSTM modeli hedef veri ile tekrar eğitilir. Bu modelde, kaynak seçimi yapılmamaktadır. ERP-LSTM-TL modeli ise uzaklık tabanlı kaynak seçiminin yapıldığı transfer öğrenme modelidir.
Çizelge 4.9'da verilen her hedef verisi için LSTM ve LSTM-TL modellerine ait RMSE ve MAE değerleri arasındaki farkların LSTM modeline göre yüzde değişimi sırasıyla Şekil 4.5 A ve B'de yer alan ısı haritalarında verilmektedir. Isı haritalarındaki koyu renkler LSTM-TL'nin LSTM'ye göre bu performans ölçütlerinde iyileşme sağladığı durumları, açık renkler ise bu performans ölçütlerinde yüzde değişimin sıfıra yakın ya da negatif olduğu durumları belirtmektedir. Şekil 4.5'teki taralı alanlar ise tek seri içeren veri kümeleri için kaynak-hedef veri eşleşmesinin tanımlı olmadığı kombinasyonları göstermektedir. Toplamda 46 adet kaynak ve hedef veri kombinasyonu için yapılan deneylerin 38 adedinde LSTM-TL modeli LSTM modeline göre her iki performans ölçütünde iyileşme sağlamıştır. Diğer sekiz adet kaynak-hedef veri eşleşmesinde ise negatif transfer öğrenme gözlenmiştir ve Şekil 4.5'teki ısı haritalarında kırmızı renk ile altı çizili olarak gösterilmiştir. RMSE'ye göre Forklift-Rossmann, Forklift-Tv_Sales, Rossmann-Forklift, Rossmann-LearnX, Rossmann-Pharma, Favorita-Forklift, Favorita-Pharma ve Pharma-Walmart olmak üzere sekiz adet kaynak-hedef veri kombinasyonunda; MAE'ye göre ise Forklift-Tv_Sales, Rossmann-Forklift, Rossmann-LearnX ve Rossmann-Pharma olmak üzere dört adet kaynak-hedef veri kombinasyonunda negatif transfer öğrenme gözlenmiştir. Her iki performans ölçütüne göre ise sekiz adet kaynak-hedef veri eşleşmesinde negatif transfer öğrenme meydana gelmiştir.

ERP-LSTM-TL yöntemi uygulandığında elde edilen değişim yüzdeleri ise Şekil 4.5'teki ısı haritalarında yuvarlak ile işaretlenmiştir. ERP-LSTM-TL yöntemi ile seçilen kaynak-hedef eşleşmeleriyle tüm hedef verilerde LSTM'ye göre iyileşme sağlanmıştır ve negatif transfer öğrenme meydana gelmemiştir. Örneğin, ERP-LSTM-TL ile LearnX-Seri_4 hedef verisi için LearnX-Seri_5 kaynak verisi olarak seçilmiştir. Bu durumda, LSTM modeline göre her iki performans ölçütünde sırasıyla %18,81 ile %18,10 oranında iyileşme elde edilmiştir. LearnX-Seri_4 hedef verisinde LSTM ve ERP-LSTM-TL yöntemleriyle elde edilen RMSE değerleri sırasıyla 28,18 ve 22,88'dir. LSTM'ye kıyasla ERP-LSTM-TL yöntemi ile tahmin hatasında sağlanan iyileşme oranı ((28,18 −

$22{,}88) / 28{,}18) * 100 = \% 18{,}81$ olarak hesaplanmıştır. Şekil 4.5'te verilen ısı haritalarındaki tüm iyileşme yüzdeleri benzer şekilde hesaplanmıştır. LearnX-Seri_4 hedef verisi için LSTM ve ERP-LSTM-TL ile elde edilen tahmin sonuçları sırasıyla Şekil 4.6 A ve B'de görselleştirilmiştir. Bu hedef veride, Rossmann kaynak verisi ile transfer öğrenme uygulandığında ise Şekil 4.5'te görüldüğü gibi negatif öğrenme gözlenmektedir.

Özetlemek gerekirse, ERP-LSTM-TL'deki benzerlik tabanlı kaynak seçimi negatif transfer öğrenmenin engellenmesinde etkilidir. ERP-LSTM-TL yöntemi hedef veriyle eğitilen LSTM'ye kıyasla RMSE ölçütüne göre ortalama %5,77 ve MAE ölçütüne göre ortalama %7,42 iyileşme sağlamıştır. Bu amaçla, ERP-LSTM-TL yöntemi ile seçilen kaynak veriler için Şekil 4.5'te yer alan iyileşme yüzdelerinin ortalaması alınmıştır. Örneğin, ele alınan yedi hedef verisinde RMSE ölçütüne göre ortalama iyileşme $(10{,}92 + 2{,}90 + 18{,}81 + 0{,}07 + 1{,}42 + 0{,}01 + 6{,}26) / 7 = \% 5{,}77$ olarak hesaplanmıştır. Ayrıca her bir hedef verisi için eğitim süresindeki iyileşme, LSTM ağı ile LSTM-TL'nin eğitim süreleri arasındaki farkın LSTM'nin eğitim süresine oranı olarak hesaplanmıştır. Böylece, LSTM-TL ve LSTM'nin eğitim süreleri karşılaştırıldığında LSTM-TL ile eğitim sürelerinde %40,28 ile %53,19 arasında azalma elde edilmiştir. Eğitim sürelerindeki iyileşme değerleri benzer şekilde hesaplandığında ERP-LSTM-TL yönteminin eğitim sürelerinde ortalama %47,90 azalma sağladığı görülmüştür.

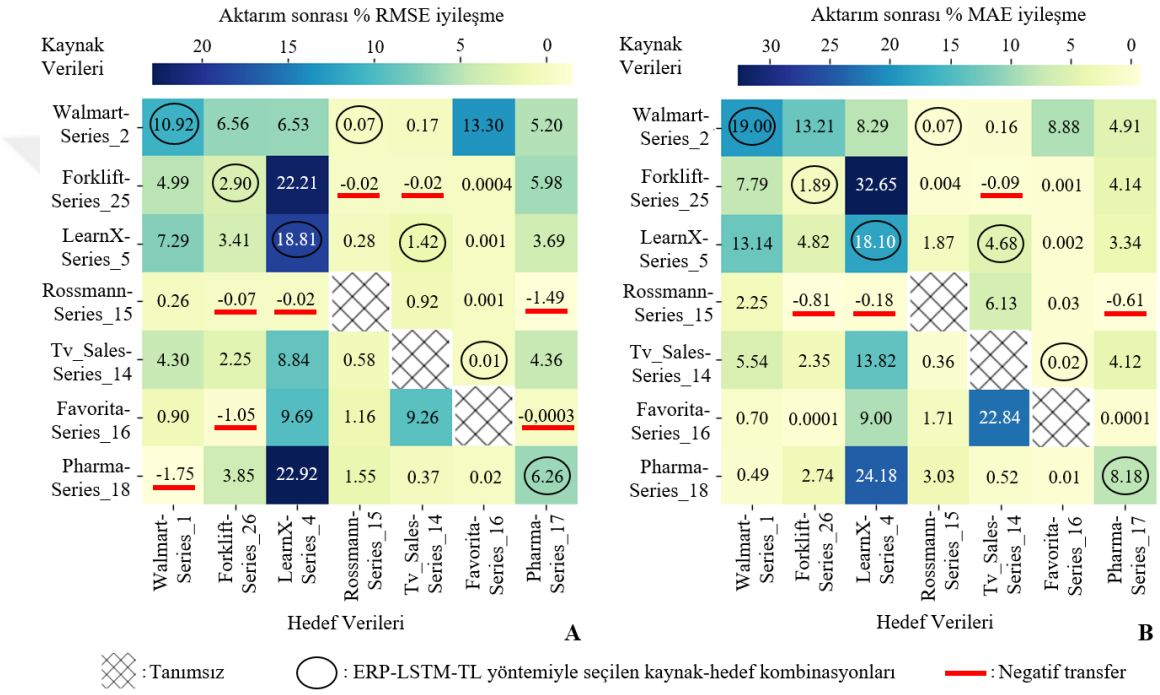
Tahmin yöntemlerinin karşılaştırılması ve parametrik olmayan istatistiksel testler

ERP-LSTM-TL yöntemi, istatistiksel tahmin yöntemleri ve makine öğrenmesi yöntemleri ile kıyaslanmıştır. Her bir hedef veri için tahmin yöntemlerinin sonuçları Çizelge 4.10'da verilmiştir. Buna göre, ERP-LSTM-TL yedi adet veri kümesinin beşinde her iki performans kriterine göre en iyi sonuçları vermiştir. Forklift verisinde en iyi sonucu LR, Favorita verisinde ise ANN vermiştir.

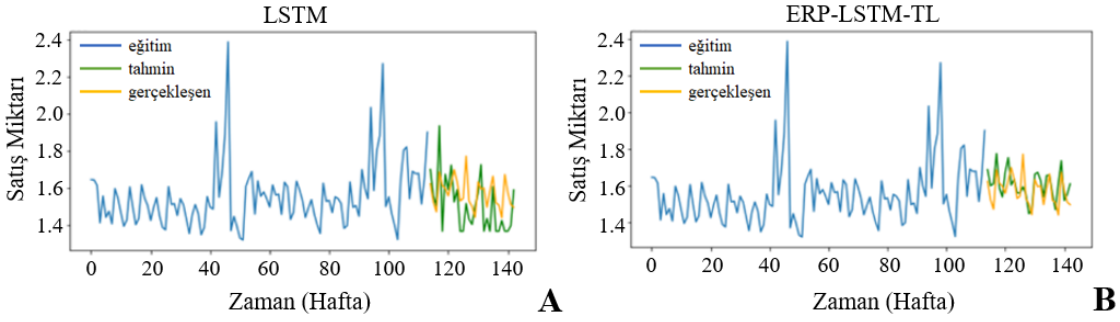
Tahmin yöntemlerinin performanslarının karşılaştırılması için Demsar (2006)'da önerilen parametrik olmayan istatistiksel testler kullanılmıştır. Bu testlerin normallik veya varyans homojenliği gibi varsayımları yoktur. Bu kapsamda, tahmin yöntemleri için ikili ve çoklu karşılaştırmalar yapılmıştır (Demsar, 2006). Çoklu karşılaştırmalar için Friedman ile

Nemenyi post-hoc testleri, ikili karşılaştırmalar için Wilcoxon eşleştirilmiş işaretli sıra testi uygulanmıştır (Demsar, 2006).

Friedman testinde hipotezler şöyledir: (i) H_0: Tahmin yöntemlerinin performanslarında farklılık yoktur. (ii) H_1: Tahmin yöntemlerinden en az birinin performansında farklılık vardır. Friedman testinde Demsar (2006)'da önerildiği üzere Iman ve Davenport (1980) istatistiği kullanılmıştır.



Şekil 4.5. Isı haritaları
A) LSTM ve LSTM-TL'nin RMSE değerleri arasındaki farkın LSTM'ye göre yüzde değişimine ait ısı haritası **B)** LSTM ve LSTM-TL'nin MAE değerleri arasındaki farkın LSTM'ye göre yüzde değişimine ait ısı haritası



Şekil 4.6. LearnX-Seri_4 hedef verisi için sonuç grafikleri
A) LSTM sonuç grafiği **B)** ERP-LSTM-TL sonuç grafiği

Çizelge 4.10. Tahmin yöntemlerinin karşılaştırılması

Yöntem	Ölçüt	Hedef Veri Kümesi						
		Walmart-Seri_1	Forklift-Seri_26	LearnX-Seri_4	Rossmann-Seri_15	Tv_Sales-Seri_14	Favorita-Seri_16	Pharma-Seri_17
SMA	RMSE	115 770,64	0,75	45,68	3 329 590,09	2 908,25	14 670,57	19,67
	MAE	98 716,04	0,71	38,44	2 711 614,43	1 920,06	9 191,51	14,25
SES	RMSE	116 166,98	0,68	32,46	3 263 554,06	2 897,19	14 514,52	19,10
	MAE	98 550,29	0,65	25,38	2 502 639,81	1 915,83	9 095,75	13,80
ARIMA	RMSE	122 601,63	0,66	28,04	3 264 280,31	2 897,20	14 421,66	19,10
	MAE	104 878,25	0,64	21,43	2 506 686,66	1 915,83	9 047,29	13,80
LR	RMSE	112 991,92	0,46*	32,48	3 310 276,02	2 271,73	13 790,60	15,79
	MAE	93 347,53	0,39**	25,89	2 656 003,10	1 829,05	9 130,77	12,32
KNN-R	RMSE	98 477,90	2,74	29,65	2 902 171,84	1 495,86	13 721,35	14,57
	MAE	78 923,85	2,67	22,36	2 262 102,26	895,17	8 938,76	11,12
RT	RMSE	206 813,33	1,29	31,57	3 704 580,63	1 807,19	19 971,05	16,62
	MAE	141 551,82	1,24	24,27	2 496 191,80	1 214,33	10 967,13	12,37
SVR	RMSE	85 313,99	1,35	29,30	3 256 237,12	2 628,90	14 058,08	13,94
	MAE	68 845,31	1,30	21,35	2 491 571,59	1 478,29	8 950,41	10,65
ANN	RMSE	97 200,29	0,54	24,41	1 760 873,74	1 485,04	12 141,89*	13,10
	MAE	79 663,94	0,46	18,94	1 282 503,70	789,19	7 147,44**	9,85
LSTM	RMSE	90 000,71	0,69	28,18	1 758 434,87	1 379,42	13 771,48	11,47
	MAE	74 722,67	0,65	23,07	1 282 280,57	715,58	9 124,39	9,61
ERP-LSTM-TL	RMSE	80 169,47*	0,67	22,88*	1 757 254,14*	1 359,90*	13 770,33	10,75*
	MAE	60 526,93**	0,64	18,89**	1 281 410,33**	682,13**	9 122,89	8,82**

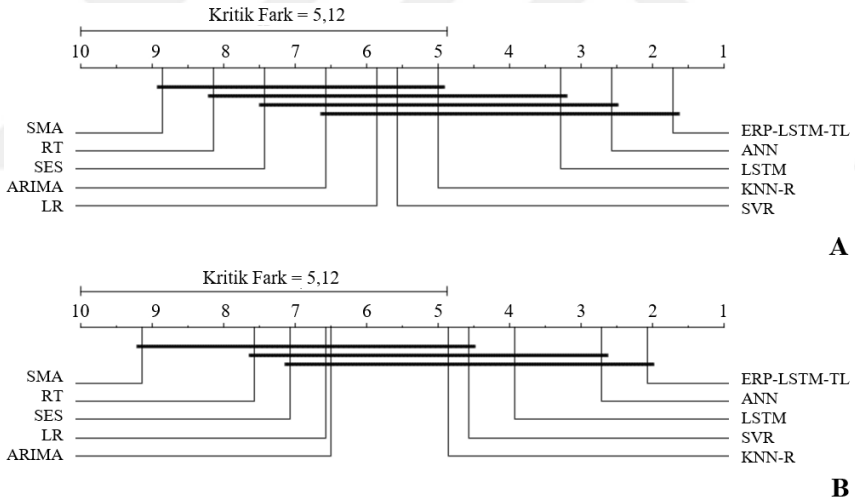
* Hedef verideki en küçük RMSE değeridir.
** Hedef verideki en küçük MAE değeridir.

RMSE ve MAE performans kriterleri için %5 anlamlılık düzeyinde Friedman testlerinin sonuçları Çizelge 4.11'de verilmektedir. Buna göre, tahmin yöntemlerinden en az birinde RMSE ve MAE değerlerinde istatistiksel olarak anlamlı bir fark vardır. İstatistiksel olarak farklı performansa sahip tahmin yöntemlerinin belirlenmesi için Nemenyi post-hoc testi yapılmıştır. Nemenyi post-hoc testinde yöntemlerin ortalama sıralama değerleri (OSD) arasındaki farklar kritik uzaklık ile karşılaştırılır. Eğer iki yöntemin ortalama sıralama değerleri arasındaki fark kritik uzaklıktan küçük ise bu iki yöntem arasında istatistiksel olarak anlamlı fark olmadığına kanaat getirilir. Bu nedenle, Nemenyi test sonuçlarında bu iki yöntem aynı doğru üzerinde ve birbiriyle bağlantılı olarak gösterilmektedir. Tahmin yöntemlerinin ortalama sıralama değerleri Şekil 4.7 A ve B'de yer almaktadır ve kritik uzaklık (Kritik Fark - KF) %5 anlamlılık düzeyinde 5,12 olarak hesaplanmıştır. Bir yöntemin ortalama sıralama değerinin küçük olması yöntemin performansının diğer yöntemlerden üstün olduğunu göstermektedir. Buna göre, Şekil 4.7 A ve B'de ERP-LSTM-TL yöntemi ile RMSE ve MAE performans kriterlerine göre en iyi ortalama sıralama değerleri (sırasıyla 1,75 ve 2,09) elde edilmiştir. ERP-LSTM-TL yöntemi

RMSE'ye göre SMA (OSD = 8,84), SES (OSD = 7,42) ve RA (OSD = 8,21)'dan; MAE'ye göre ise SMA (OSD = 9,27) ve RA (OSD = 7,58)'dan istatistiksel olarak daha iyi sonuç vermiştir. Örneğin, RMSE'ye göre ERP-LSTM-TL (OSD = 1,75) ile SMA (OSD = 8,84) yöntemlerinin ortalama sıralama değerleri arasındaki fark $8{,}84 - 1{,}75 = 7{,}09$'dur ve bu değer kritik farktan (7,09 > 5,12) büyüktür. Bu nedenle, RMSE'ya göre ERP-LSTM-TL yönteminin performansı SMA yönteminin performansından istatistiksel olarak üstündür. Diğer yöntemler ile ERP-LSTM-TL arasında istatistiksel olarak anlamlı fark bulunmadığı gözlenmiştir.

Çizelge 4.11. Friedman test sonuçları

Performans Kriteri	Hesaplanan F-Değeri (F_F)	F-Tablo Değeri ($F_{\alpha=0{,}05,\ k-1=9,\ (k-1)*(N-1)=54}$)	Sonuç
RMSE	9,81	2,06	$F_F > F_{0{,}05;\,9;\,54}$, H_0 hipotezi reddedilir.
MAE	7,39	2,06	



Şekil 4.7. Nemenyi testi sonuç diyagramları
A) RMSE diyagramı **B)** MAE diyagramı

Tahmin yöntemlerinin ikili karşılaştırmaları için Wilcoxon eşleştirilmiş işaretli sıra testi uygulanmıştır. Tek taraflı kurulan hipotez testlerinde, H_0 ve H_1 hipotezleri şöyledir: (i) H_0: İki tahmin yönteminin hata miktarları eşit veya birinci yöntemin hata miktarı, ikinci yöntemin hata miktarından daha fazladır. (ii) H_1: Birinci tahmin yönteminin hata miktarı, ikinci tahmin yönteminin hata miktarından azdır. Çizelge 4.12'de tahmin yöntemlerinin ikili karşılaştırmalarına ilişkin Wilcoxon eşleştirilmiş işaretli sıra testi sonuçları

verilmektedir. Bu tabloda satırlar birinci tahmin yöntemini, sütunlar ise ikinci tahmin yöntemini göstermektedir ve %5 anlamlılık düzeyinde ($\alpha = 0,05$) H_0 hipotezi reddedilen ikili karşılaştırmalar * işareti ile belirtilmiştir.

Çizelge 4.12'de verilen Wilcoxon eşleştirilmiş işaretli sıra testi sonuçlarına göre, aşağıdaki bulgular gözlenmiştir:

- Klasik tahmin yöntemleri analiz edildiğinde SES yöntemi, MAE performans kriterine göre SMA yönteminden; LR ise her iki performans kriterine göre SMA yönteminden istatistiksel olarak daha iyi performans göstermektedir (p-değeri ≤ 0,05). SMA ve ARIMA'nın performansı diğer tahmin yöntemlerinin performansından düşüktür (p-değeri > 0,05).
- Makine öğrenmesi yöntemleri incelendiğinde KNN-R yöntemi; klasik tahmin yöntemleri ile RT'den istatistiksel olarak daha iyi performans göstermektedir (p-değeri ≤ 0,05). SVR yönteminin performansı SMA, SES ve ARIMA yöntemlerinin performansından istatistiksel olarak üstündür (p-değeri ≤ 0,05). ANN ile SMA, SES, ARIMA, LR ve RT yöntemlerinin aralarındaki farklar istatistiksel olarak anlamlı olup ANN'nin performansı bu yöntemlerden daha iyidir (p-değeri ≤ 0,05). Ayrıca, KNN-R ile ANN'nin RMSE değerleri arasındaki fark istatistiksel olarak anlamlıdır ve ANN yöntemi KNN-R'ye göre daha iyi performans göstermektedir (p-değeri ≤ 0,05). RT yöntemi ise diğer tahmin yöntemlerinden istatistiksel olarak daha düşük performans göstermektedir (p-değeri > 0,05). Özetle, ANN yöntemi makine öğrenmesi yöntemleri içinde en iyi performansa sahip yöntemdir.
- Derin öğrenme yöntemleri içerisinde yer alan LSTM ise SMA, SES, LR ve RA yöntemlerinden her iki performans kriterine göre; ARIMA ve KNN-R yöntemlerinden ise RMSE performans kriterine göre istatistiksel olarak performans üstünlüğüne sahiptir (p-değeri ≤ 0,05). Önerilen ERP-LSTM-TL yöntemi, ANN dışındaki tüm tahmin yöntemlerine göre istatistiksel olarak daha düşük hata değerleri vermektedir. Ayrıca anlamlılık düzeyi %10 (α =0,10) olduğunda, ERP-LSTM-TL yöntemi RMSE'ye göre diğer tüm tahmin yöntemlerine istatistiksel olarak performans üstünlüğü sağlamaktadır.

Önerilen yaklaşımın performansı Çizelge 3.1'de yer alan veri kümelerinin özelliklerine göre incelendiğinde mevsimsellik içeren beş veri kümesinden dördünde (Walmart, LearnX, Rossmann ve Pharma) önerilen yaklaşım ile en doğru tahmin sonuçlarının elde edildiği görülmüştür. Diğer dört veri kümesinden daha uzun mevsimsel periyoda sahip Favorita'da ise 10 adet yöntem içerisinde, önerilen yaklaşım üçüncü en iyi performansa sahip yöntemdir. Ayrıca artma eğilimi olan Tv_Sales ile hem artma hem de azalma eğilimi olan Pharma veri kümelerinde en iyi performans önerilen yaklaşıma aittir. Ancak çok sayıda sıfır içeren, düzensiz ve aralıklı (intermittent) bir veri kümesi olan Forklift için önerilen yaklaşım RMSE'ye göre dördüncü, MAE'ye göre üçüncü en iyi performansa sahiptir.

Sonuç olarak, yapılan deneysel çalışmalarda farklı sektörler ve farklı ürünlerdeki satış verilerine benzerlik tabanlı kaynak seçimi ve derin transfer öğrenme uygulandığında tahmin doğruluğu ve eğitim süresinde iyileşme sağlanmıştır. Özellikle hedef veriye benzer bir kaynak verinin seçilmesinin derin transfer öğrenmede aktarılabilirlik açısından fayda sağladığı gözlenmiştir.

Çizelge 4.12. Wilcoxon eşleştirilmiş işaretli sıra testi testlerinin p-değerleri

Yöntem	Ölçüt	SMA	SES	ARIMA	LR	KNN-R	RT	SVR	ANN	LSTM	ERP-LSTM-TL
SMA	RMSE		0,91	0,91	0,99	0,99	0,20	0,99	0,99	0,99	0,99
	MAE		0,99	0,91	0,99	0,99	0,63	0,99	0,99	0,99	0,99
SES	RMSE	0,09		0,43	0,84	0,99	0,20	0,99	0,99	0,99	0,99
	MAE	0,01*		0,43	0,57	0,99	0,57	0,99	0,99	0,95	0,95
ARIMA	RMSE	0,09	0,57		0,75	0,97	0,09	0,97	0,99	0,97	0,99
	MAE	0,09	0,57		0,50	0,97	0,37	0,98	0,99	0,88	0,95
LR	RMSE	0,01*	0,16	0,25		0,98	0,12	0,75	0,99	0,99	0,99
	MAE	0,01*	0,43	0,50		0,98	0,50	0,99	0,99	0,99	0,99
KNN-R	RMSE	0,01*	0,01*	0,03*	0,02*		0,01*	0,37	0,99	0,95	0,95
	MAE	0,01*	0,01*	0,03*	0,02*		0,02*	0,37	0,94	0,91	0,95
RT	RMSE	0,80	0,80	0,91	0,88	0,99		0,94	0,99	0,99	0,99
	MAE	0,37	0,43	0,63	0,50	0,98		0,94	0,99	0,99	0,99
SVR	RMSE	0,01*	0,01*	0,03*	0,25	0,63	0,06		0,91	0,91	0,99
	MAE	0,01*	0,01*	0,02*	0,01*	0,63	0,06		0,91	0,57	0,95
ANN	RMSE	0,01*	0,01*	0,01*	0,01*	0,01*	0,01*	0,09		0,80	0,91
	MAE	0,01*	0,01*	0,01*	0,01*	0,06	0,01*	0,09		0,75	0,84
LSTM	RMSE	0,01*	0,01*	0,03*	0,01*	0,05*	0,01*	0,09	0,20		0,99
	MAE	0,01*	0,05*	0,12	0,01*	0,09	0,01*	0,43	0,25		0,99
ERP-LSTM-TL	RMSE	0,01*	0,01*	0,01*	0,01*	0,05*	0,01*	0,01*	0,09	0,01*	
	MAE	0,01*	0,05*	0,05*	0,01*	0,05*	0,01*	0,05*	0,16	0,01*	

* %5 anlamlılık düzeyinde H_0 hipotezi reddedilmiştir ($\alpha = 0,05$).

4.3.2. Önerilen çözüm yaklaşımlarının karşılaştırılması

Bu bölümde tek değişkenli satış verileri için geliştirilen kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımına ait deneysel çalışmalar sunulmaktadır. Bu kapsamda her bir satış verisinden Çizelge 4.7'de kaynak ve hedef kümeler olarak belirtilen seriler ele alınmıştır. Böylece toplam 11 adet zaman serisi kullanılmıştır. Bu serilerin her biri sırasıyla kaynak veri olarak alınmıştır ve geriye kalan seriler hedef olarak kullanılarak transfer öğrenme yapılmıştır. Örneğin, Walmart-Seri_1 kaynak veri olarak alındığında geriye kalan 10 seri hedef veri olarak kullanılmıştır ve Walmart-Seri_1 kaynak verisinden bu hedef verilere transfer öğrenme gerçekleştirilmiştir. Buradaki amaç mevcut serilerin birbiri için kaynak ve hedef olarak kullanılması durumunda elde edilen transfer öğrenme sonuçlarını analiz etmektir. Başka bir ifadeyle, tüm kaynak-hedef eşleşmelerinin transfer öğrenme sonuçlarını incelemek ve yorumlamak hedeflenmiştir. Kaynak verilerle LSTM modelleri eğitilmiştir ve ön eğitilmiş LSTM modelleri kullanılarak hedef verilere parametre aktarımı yoluyla TL (LSTM-TL) uygulanmıştır. Hedef veri kümelerinde LSTM'ye kıyasla LSTM-TL modeli ile elde edilen RMSE ve MAE iyileşme yüzdeleri hesaplanmıştır.

Tek değişkenli satış verileri için geliştirilen kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımına ait deneysel çalışmaların adımları ve elde edilen sonuçlar aşağıda açıklanmıştır.

Adım 1: Özniteliklerin çıkarımı

Bu adımda, Şekil 3.14'te yer alan zaman serilerine ait istatistiksel öznitelikler, satış ile ilgili öznitelikler ve uzaklık tabanlı öznitelikler çıkarılmıştır. Buna göre, 30 öznitelik elde edilmiştir ve bu öznitelikler Ek 4 ve Ek 5'te sunulmaktadır.

Adım 2: Transfer öğrenme için regresyon modelinin oluşturulması

Öznitelik çıkarımı sonrası, kaynak-hedef eşleşmeleri ve bu eşleşmelerdeki transfer öğrenme sonuçlarının RMSE/MAE iyileşme oranları alınarak tahmin problemi

tanımlanmıştır. Transfer öğrenme aktarım başarısının tahmininde kullanılan veri kümesi Şekil 4.8'de gösterilmektedir. Bu veri kümesinde toplam 110 satır veri ve 56 öznitelik bulunmaktadır. TL'nin başarılı olduğu (pozitif TL) veri sayısı 78 (%71), TL'nin başarılı olmadığı (negatif TL) veri sayısı 32 (%29)'dir. Çıktı değişkenine bağlı olarak iki farklı tahmin problemi olarak ele alınmıştır: Bir problemde çıktı değişkeni RMSE'deki iyileşme oranı iken, diğer problemde çıktı değişkeni MAE'deki iyileşme oranıdır.

Adım 3: Öznitelik seçimi

Bu adımda, rastgele orman yöntemi ile tahmin modelleri geliştirilmiştir ve öznitelik seçimi yapılmıştır. Çizelge 4.3'te verilen parametreler kullanılarak ızgara arama yöntemi ile parametre optimizasyonu yapılmıştır. Izgara arama sırasında birini dışarıda bırak çapraz doğrulama (Leave-one-out cross validation – LOOCV) uygulanmıştır.

Yapılan deneyler sonucunda her iki problemden toplam kırk sekiz adet rastgele orman elde edilmiştir ve bu modellere ilişkin bilgiler Ek 6'da verilmiştir. Transfer öğrenme aktarım başarısının RMSE ile tahmininde 14 numaralı rasgele orman yüksek R^2 değerine ve en düşük torba dışı (out-of-bag) hataya sahiptir. Çıktı değişkeni MAE'deki iyileşme oranı olan problemde 38 numaralı orman yüksek R^2 değerine ve en düşük torba dışı hataya sahiptir. Ayrıca bu modellerin negatif iyileşme durumlarını (negatif TL durumu) doğru tahmin edebildikleri tespit edilmiştir. Seçilen rastgele orman modellerindeki öznitelikler önem derecesine göre büyükten küçüğe sıralanmıştır ve en büyük önem derecesine sahip 20 öznitelik seçilmiştir. Bu özniteliklerin %82 kapsayıcılığı bulunmaktadır. Her iki ağaçtan gelen 20'şer öznitelik ve önem değerleri Çizelge 4.13'te verilmektedir. Buna göre, 22 öznitelik seçilerek önem değerleri [0-1] aralığına getirilmiştir. Bu 22 özniteliğin iki tanesi uzaklık tabanlı öznitelik, iki tanesi satış ile ilgili öznitelik ve kalanı zaman serisi öznitelikleridir. En yüksek öneme sahip ilk iki öznitelik DTW ve ERP uzaklık tabanlı özniteliklerdir. En düşük öneme sahip öznitelik ise kaynak verinin birinci dereceden kısmi otokorelasyon (Pacf1x) özniteliğidir.

Deney No	Kaynak Veri	Hedef Veri	Kaynak Veriye Ait Öznitelikler			Hedef Veriye Ait Öznitelikler			Kaynak-Hedef Eşleşmesine Yönelik Öznitelikler				TL İyileşme Yüzdesi	
			X1	...	X26	X27	...	X52	X53	X54	X55	X56	RMSE	MAE
			Mean_S	...	ZeroSales Per_S	Mean_T	...	ZeroSales Per_T	Sector_ Info	Similar Product	ERP	DTW		
1	Seri_1	Seri_2	1555264,40	...	0,00	1925751,34	...	0,00	1,00	1,00	52979632,00	3432420,91	2,12	1,98
2	Seri_1	Seri_4	1555264,40	...	0,00	85,25	...	0,00	0,00	0,00	222440023,00	40382394,00	8,11	9,27
3	Seri_1	Seri_5	1555264,40	...	0,00	120,60	...	0,00	0,00	0,00	222456330,00	40381347,80	1,02	1,08
4	Seri_1	Seri_14	1555264,40	...	0,00	4573,55	...	0,00	0,00	1,00	223554970,00	42169304,20	0,08	0,07
5	Seri_1	Seri_15	1555264,40	...	0,00	6234798,96	...	0,00	1,00	1,00	5650777814,00	153147259,00	0,11	0,12
6	Seri_1	Seri_16	1555264,40	...	0,00	83814,54	...	0,00	1,00	1,00	332568090,00	51598592,50	11,70	9,03
7	Seri_1	Seri_17	1555264,40	...	0,00	29,92	...	0,01	0,00	0,00	222446626,00	61265516,70	7,12	6,65
8	Seri_1	Seri_18	1555264,40	...	0,00	3,90	...	0,02	0,00	0,00	222408523,00	61266707,80	3,13	1,98
9	Seri_1	Seri_25	1555264,40	...	0,00	1,46	...	0,37	0,00	0,00	222402477,00	22288070,80	4,87	5,16
10	Seri_1	Seri_26	1555264,40	...	0,00	1,04	...	0,42	0,00	0,00	222402573,00	22288077,20	-3,77	-4,12
11	Seri_2	Seri_1	1925751,34	...	0,00	1555264,40	...	0,00	1,00	1,00	52979632,00	3432420,91	10,92	19,00
12	Seri_2	Seri_4	1925751,34	...	0,00	85,25	...	0,00	0,00	0,00	275419655,00	50507180,90	6,53	8,29
	.	.	.		.	.		.	.	.	.	.	.	.
	.	.	.	⋱	.	.	⋱	.	.	.	.	.	.	.
	.	.	.		.	.		.	.	.	.	.	.	.
101	Seri_26	Seri_1	1,04	...	0,42	1555264,40	...	0,00	0,00	0,00	222402573,00	22288077,20	4,99	7,79
102	Seri_26	Seri_2	1,04	...	0,42	1925751,34	...	0,00	0,00	0,00	275382205,00	27746396,10	0,07	0,09
103	Seri_26	Seri_4	1,04	...	0,42	85,25	...	0,00	0,00	0,00	74956,00	2601,55	22,21	32,65
104	Seri_26	Seri_5	1,04	...	0,42	120,60	...	0,00	0,00	0,00	106129,00	3697,39	14,07	17,81
105	Seri_26	Seri_14	1,04	...	0,42	4573,55	...	0,00	0,00	0,00	4458725,00	191500,92	-0,02	-0,09
106	Seri_26	Seri_15	1,04	...	0,42	6234798,96	...	0,00	0,00	0,00	5873180387,00	214103035,00	-0,02	0,04
107	Seri_26	Seri_16	1,04	...	0,42	83814,54	...	0,00	0,00	0,00	141478709,00	3486068,76	0,00	0,00
108	Seri_26	Seri_17	1,04	...	0,42	29,92	...	0,01	0,00	0,00	62769,40	1430,77	5,98	4,14
109	Seri_26	Seri_18	1,04	...	0,42	3,90	...	0,02	0,00	0,00	7968,62	105,89	-0,22	-0,31
110	Seri_26	Seri_25	1,04	...	0,42	1,46	...	0,37	0,00	0,00	128,00	12,81	2,90	1,89

Şekil 4.8. Transfer öğrenme aktarım başarısının tahmininde kullanılan veri kümesi

Adım 4: Benzemezlik matrisinin oluşturulması

Rastgele orman modelleri ile transfer öğrenme başarısını etkileyen öznitelikler belirlenmiştir ve Çizelge 4.13'te yer alan bu öznitelikler ile benzemezlik matrisi oluşturulmuştur. DTW ve ERP dışında kalan öznitelikler Çizelge 4.14'teki gibi alınmıştır. Çizelge 4.14'teki veri kullanılarak Denklem (3.13) ile satış verileri arasındaki ağırlıklı Öklid uzaklığı hesaplanmıştır. Hesaplanan Öklid uzaklıkları Ek 7'de verilmiştir.

Satış verileri arasındaki DTW ve ERP uzaklıkları sırasıyla Ek 8 ve Ek 9'da verilmiştir. DTW ve ERP uzaklıklarına [0, 1] normalizasyonu uygulanmıştır. Normalize edilmiş DTW ve ERP uzaklıkları sırasıyla Ek 10 ve Ek 12'de verilmiştir. Ardından bu normalize DTW ve ERP uzaklıkları ilgili özniteliğin önem derecesi ile ağırlıklandırılmıştır. Önem derecesi ile ağırlıklandırılmış ve normalize edilmiş DTW ve ERP uzaklıkları sırasıyla Ek 11 ve Ek 13'te verilmiştir. Tüm bu uzaklıklar Denklem (3.14)'teki formülde kullanılarak benzemezlik matrisi Çizelge 4.15'teki gibi elde edilmiştir.

Çizelge 4.13. Rastgele orman modellerinde seçilen öznitelikler ve önem dereceleri

No	Öznitelik	Çıktı değişkeni RMSE'deki iyileşme oranı olan modeldeki önem derecesi	Çıktı değişkeni MAE'deki iyileşme oranı olan modeldeki önem derecesi	Ortalama Önem Derecesi	Normalize Edilmiş Önem Dereceleri [0-1]
1	DTW	0,22	0,21	0,21	0,26
2	ERP	0,20	0,18	0,19	0,23
3	Trough_T	0,06	0,06	0,06	0,07
4	MaxKlShift_T	0,05	0,06	0,06	0,07
5	Peak_T	0,04	0,02	0,03	0,03
6	Acf1e_T	0,04	0,07	0,05	0,06
7	Hurst_T	0,02	0,03	0,02	0,03
8	FlatSpots_T	0,02	0,01	0,01	0,02
9	Trend_T	0,02	0,02	0,02	0,02
10	Acf1x_T	0,02	0,01	0,01	0,02
11	MaxKlShift_S	0,01	0,02	0,02	0,02
12	Trough_S	0,01	0,02	0,02	0,02
13	Stability_T	0,01	0,01	0,01	0,02
14	Peak_S	0,01	0,01	0,01	0,02
15	SeasonalStrength_S	0,01	0,01	0,01	0,01
16	Trend_S	0,01	0,01	0,01	0,01
17	Sector_Info*	0,01	0,00	0,01	0,01
18	Acf1e_S	0,01	0,02	0,02	0,02
19	ZeroSalesPer_T*	0,01	0,01	0,01	0,01
20	SeasonalStrength_T	0,01	0,01	0,01	0,01
21	Linearity_S**	0,01	0,02	0,01	0,02
22	Pacf1x_T**	0,01	0,01	0,01	0,01
Toplam Önem Derecesi		0,82	0,82	0,82	1,00

* Sadece 14 numaralı modelden gelen öznitelikleri ifade etmektedir.
** Sadece 38 numaralı modelden gelen öznitelikleri ifade etmektedir.

Çizelge 4.14. Seçilen öznitelikler

Seri/ Öznitelik	Acf1e_S	Trend_S	...	Hurst_T	ZeroSales Per_T	Sector_Info				
						Perakende	Online Satış	Tv_Satışı	Eczane	Forklift Satışı
Seri_1	0,02	0,62	...	0,72	0,00	1,00	0,00	0,00	0,00	0,00
Seri_2	0,05	0,40	...	0,81	0,00	1,00	0,00	0,00	0,00	0,00
Seri_4	0,34	0,03	...	0,77	0,00	0,00	1,00	0,00	0,00	0,00
Seri_5	0.,29	0,04	...	0,75	0,00	0,00	1,00	0,00	0,00	0,00
Seri_14	0,71	0,94	...	1,00	0,00	0,00	0,00	1,00	0,00	0,00
Seri_15	0,06	0,02	...	0,50	0,00	1,00	0,00	0,00	0,00	0,00
Seri_16	0,23	0,14	...	0,71	0,00	1,00	0,00	0,00	0,00	0,00
Seri_17	0,11	0,44	...	0,79	0,01	0,00	0,00	0,00	1.00	0,00
Seri_18	0,02	0,10	...	0,61	0,02	0,00	0,00	0,00	1.00	0,00
Seri_25	0,00	0,46	...	0,77	0,37	0,00	0,00	0,00	0,00	1,00
Seri_26	-0,09	0,30	...	0,70	0,42	0,00	0,00	0,00	0,00	1,00
Önem Derecesi	0,02	0,01	...	0,03	0,01	0,01	0,01	0,01	0,01	0,01

Çizelge 4.15. Benzemezlik matrisi

Seri	Seri_1	Seri_2	Seri_4	Seri_5	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26
Seri_1	0,00	0,06	0,35	0,37	0,49	0,78	0,55	0,52	0,44	0,31	0,43
Seri_2	0,06	0,00	0,35	0,37	0,50	0,75	0,56	0,54	0,45	0,32	0,43
Seri_4	0,35	0,35	0,00	0,13	0,37	0,83	0,45	0,35	0,30	0,25	0,38
Seri_5	0,37	0,37	0,13	0,00	0,40	0,86	0,47	0,38	0,32	0,27	0,39
Seri_14	0,49	0,50	0,37	0,40	0,00	0,98	0,45	0,41	0,45	0,40	0,48
Seri_15	0,78	0,75	0,83	0,86	0,98	0,00	0,80	0,78	0,66	0,82	0,70
Seri_16	0,55	0,56	0,45	0,47	0,45	0,80	0,00	0,37	0,32	0,45	0,38
Seri_17	0,52	0,54	0,35	0,38	0,41	0,78	0,37	0,00	0,20	0,30	0,28
Seri_18	0,44	0,45	0,30	0,32	0,45	0,66	0,32	0,20	0,00	0,27	0,21
Seri_25	0,31	0,32	0,25	0,27	0,40	0,82	0,45	0,30	0,27	0,00	0,24
Seri_26	0,43	0,43	0,38	0,39	0,48	0,70	0,38	0,28	0,21	0,24	0,00

Adım 5: Zaman serilerinin kümelenmesi

Bu adımda, DAE tabanlı derin kümelemede Çizelge 4.4'te verilen hiperparametreler ve değer aralıkları kullanılarak ızgara arama yöntemi ile hiperparametre optimizasyonu yapılmıştır. Buna göre, iki farklı temsil alan boyutuna (latent_space) ve üç farklı küme sayısına göre derin kümeleme sonuçları Çizelge 4.16'da verilmiştir. Silhouette katsayısına göre en iyi kümeleme sonucu temsil alan boyutu 3 ve küme sayısı 2 iken elde edilmiştir. Dunn ve Calinski-Harabasz indekslerine göre ise en iyi kümeleme sonucu temsil alan boyutu 3 ve küme sayısı 4 iken elde edilmektedir.

Temsil alan boyutu 3 iken aynı kümedeki satış verilerinin kaynak olarak kullanıldığı durumlarda negatif transfer öğrenme olup olmadığı araştırılmıştır. Bu kapsamda, Ek 14'te verilen tek kaynaklı derin transfer öğrenmedeki RMSE başarı yüzdeleri kullanılmıştır. Buna göre, Çizelge 4.17'de gösterildiği gibi temsil alan boyutunun 3, küme sayısının 3 ve 4 olduğu kümeleme sonuçlarında negatif transfer öğrenme durumunun olmadığı tespit edilmiştir. Küme sayısı 2 iken ise satış verileri homojen bir şekilde ayrışmadığı için negatif transfer öğrenme gözlenmiştir. Bu nedenle, bu çalışmada küme sayısının 3 ve 4 olduğu durumlar ele alınmıştır.

Çizelge 4.16. Derin kümeleme sonucu küme atamaları

Seri	Temsil Alan Boyutu: 2			Temsil Alan Boyutu: 3		
	k=2	k=3	k=4	k=2	k=3	k=4
Seri_1	1	1	1	2	1	1
Seri_2	1	3	4	2	1	1
Seri_4	1	3	3	1	2	2
Seri_5	1	3	3	1	2	2
Seri_14	1	3	3	1	1	1
Seri_15	2	2	2	1	1	3
Seri_16	1	1	1	1	1	1
Seri_17	1	3	2	1	2	2
Seri_18	1	3	4	1	2	2
Seri_25	1	3	3	1	3	4
Seri_26	1	1	1	1	3	4
Silhouette Katsayısı:	0,80	0,69	0,75	0,83	**0,79**	0,73
Dunn İndeksi:	0,32	0,12	0,12	0,32	**0,56**	1,19
Calinski-Harabasz İndeksi:	82,39	121,78	157,30	30,68	**143,13**	247,61

Çizelge 4.17. Temsil alan boyutu 3, küme sayısı 3 ve 4 iken küme atamaları

Küme No/Küme Sayısı	Temsil Alan Boyutu = 3					
	k = 3	Negatif TL	Pozitif TL	k = 4	Negatif TL	Pozitif TL
1.Küme	Walmart-Seri_1	0	4	Walmart-Seri_1	0	3
	Walmart-Seri_2	0	4	Walmart-Seri_2	0	3
	Tv_Sales-Seri_14	0	4	Tv_Sales-Seri_14	0	3
	Favorita-Seri_16	0	4	Favorita-Seri_16	0	3
	Rossmann-Seri_15	0	4			
2.Küme	LearnX-Seri_4	0	3	LearnX-Seri_4	0	3
	LearnX-Seri_5	0	3	LearnX-Seri_5	0	3
	Pharma-Seri_17	0	3	Pharma-Seri_17	0	3
	Pharma-Seri_18	0	3	Pharma-Seri_18	0	3
3.Küme	Forklift-Seri_25	0	1	Rossmann-Seri_15		
	Forklift-Seri_26	0	1			
4.Küme				Forklift-Seri_25	0	1
				Forklift-Seri_26	0	1
Toplam Deney Sayısı		0	34	Toplam Deney Sayısı	0	26
TL Aktarım Yüzdesi		%0	%100	TL Aktarım Yüzdesi	%0	%100

Adım 6-7: Kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımlarının deneysel sonuçları

Temsil alan boyutu 3, küme sayısının 3 ve 4 olduğu durumlarda tek değişkenli zaman serileri için geliştirilen dört farklı yaklaşım karşılaştırılmıştır. Çizelge 4.17'de görüldüğü üzere küme sayısı 3 ve 4 iken oluşan 2. kümeler aynı serileri içermektedir. 2. küme içinden Pharma-Seri_18 yaklaşımların uygulanması için hedef verisi olarak seçilmiştir.

Ayrıca, küme sayısı 3 ve 4 iken oluşan 1. Kümeler Rossmann-Seri_15 haricinde aynı serileri içermektedir.

Küme sayısı 3 iken oluşan 3. küme ile küme sayısı 4 iken oluşan 4. küme aynı iki seriyi (Forklift-Seri_25 ile Forklift-Seri_26) içermektedir. Küme sayısı 4 iken oluşan 3. küme sadece Rossmann-Seri_15 satış verisinden oluşmaktadır. Bu nedenle, küme atamaları arasındaki farkın çok kaynaklı transfer öğrenme sonuçlarına etkisini inceleyebilmek için 1. kümeden Favorita-Seri_16 seçilmiştir.

Satış verilerinin benzemezlik matrisine göre en yakın komşuları (k=1, k=2 ve k=3) Çizelge 4.18'de verilmiştir:

Çizelge 4.18. Serilerin benzemezlik matrisine göre en yakın komşuları

Seri	1-NN	2-NN	3-NN
Walmart-Seri_1	Seri_2	Seri_2-25	Seri_2-25-4
Walmart-Seri_2	Seri_1	Seri_1-25	Seri_1-25-4
LearnX-Seri_4	Seri_5	Seri_5-25	Seri_5-25-18
LearnX-Seri_5	Seri_5	Seri_5-25	Seri_5-25-18
Tv_Sales-Seri_14	Seri_4	Seri_4-25	Seri_4-25-5
Rossmann-Seri_15	Seri_18	Seri_18-26	Seri_18-26-2
Favorita-Seri_16	Seri_18	Seri_18-17	Seri_18-17-26
Pharma-Seri_17	Seri_18	Seri_18-26	Seri_18-26-25
Pharma-Seri_18	Seri_17	Seri_17-26	Seri_17-26-25
Forklift-Seri_25	Seri_26	Seri_26-4	Seri_26-4-18
Forklift-Seri_26	Seri_18	Seri_18-25	Seri_18-25-17

Pharma-Seri_18 ve Favorita-Seri_16 hedef verileri için geliştirilen çözüm yaklaşımları uygulama detayları aşağıda açıklanmaktadır;

1. Topluluk tabanlı tek kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.15'teki benzemezlik matrisine göre Pharma-Seri_18'e en yakın seri olan Pharma-Seri_17, Favorita-Seri_16'ya en yakın seri olan Pharma-Seri_18 kaynak veri olarak seçilmiştir.

2. KNN ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.18'deki 2-NN komşuluğunda Pharma-Seri_18 için Pharma-Seri_17 ile Forklift-Seri_26, Favorita-Seri_16 için Pharma-Seri_17 ile Pharma-Seri_18 kaynak veri olarak alınmıştır. 3-NN komşuluğunda ise Pharma-Seri_18 için Pharma-Seri_17, Forklift-Seri_26 ve Forklift-Seri_25, Favorita-Seri_16 için Pharma-Seri_17, Pharma-Seri_18 ve Forklift-Seri_25 kaynak veri olarak alınmıştır.

3. Kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.17'ye göre küme sayısı 3 ve 4 iken Pharma-Seri_18 için LearnX-Seri_4, LearnX-Seri_5 ve Pharma-Seri_17 olmak üzere üç adet seri kaynak veri olarak alınmıştır. Küme sayısı 3 iken Favorita-Seri_16 için Walmart-Seri_1, Walmart-Seri_2, Tv_Sales-Seri_14 ve Rossmann-Seri_15 olmak üzere dört adet seri kaynak veri olarak alınmıştır. Küme sayısı 4 iken Favorita-Seri_16 için Walmart-Seri_1, Walmart-Seri_2 ve Tv_Sales-Seri_14 olmak üzere üç adet seri kaynak veri olarak alınmıştır.

4. Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Pharma-Seri_18 ve Favorita-Seri_16 için üçüncü yaklaşımda belirtilen kaynak veriler kullanılmıştır. Üçüncü yaklaşımdan farklı olarak kaynak verilerin hedef veriler üzerindeki eğitim hatalarına bağlı olarak her bir kaynak modelin nihai tahmine olan katkısı (ağırlığı) Denklem (3.16) ile hesaplanmıştır. Hedef veri üzerindeki kaynak verilerin eğitim hataları ve bu hatalara bağlı olarak her bir kaynak modelin nihai tahmine olan katkısı (ağırlığı) hesaplanmıştır. Kullanılan model sayısına bağlı olarak ağırlıklar [0-1] aralığında olacak şekilde her iki hedef verisi için hesaplanmıştır. Pharma-Seri_18 hedef verisi için bu model ağırlıkları kaynak bazında sırasıyla Ek 15, Ek 16 ve Ek 17'de verilmiştir.

Transfer öğrenmenin etkisini görebilmek amacıyla yukarıdaki yaklaşımlara ek olarak her iki hedef verisinde LSTM yöntemi ile torbalama topluluk yöntemi uygulanmıştır.

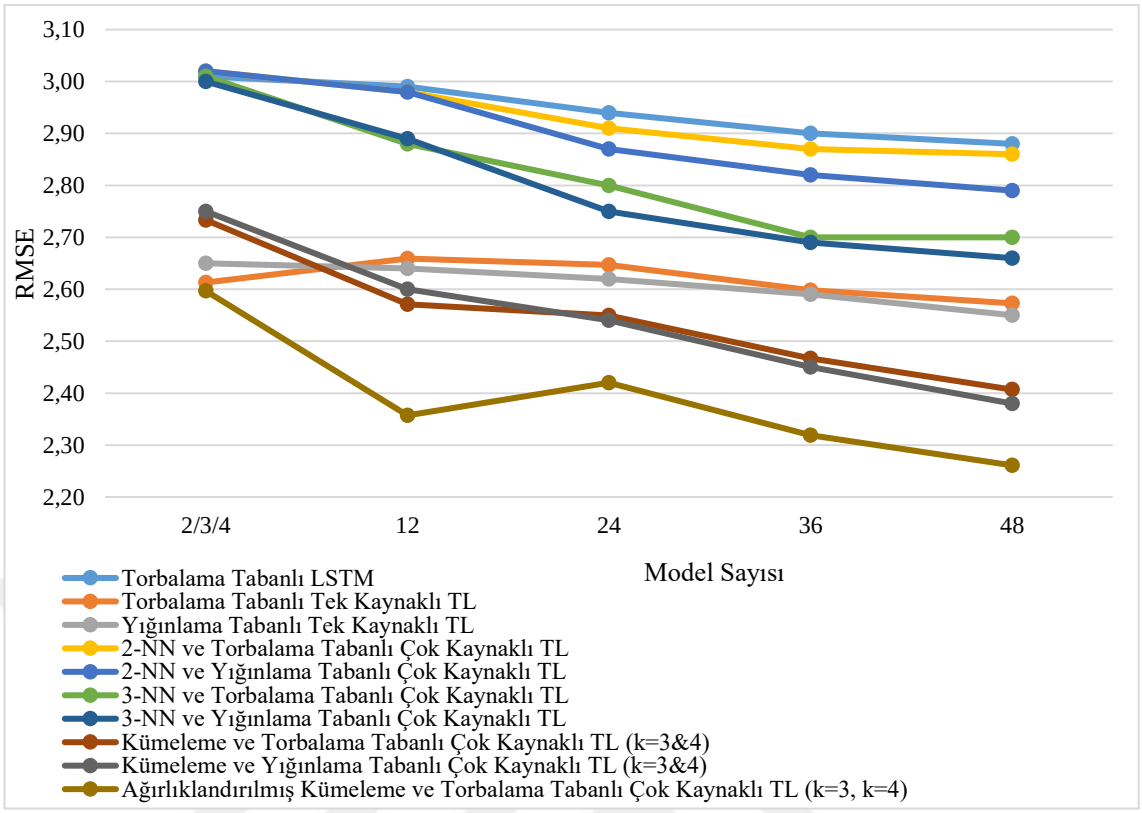
Pharma-Seri_18 için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dakika) Ek 18'de verilmiştir. Ayrıca, bu sonuçlar sırasıyla Şekil 4.9, Şekil 4.10 ve Şekil 4.11'de gösterilmiştir. Ek 18'de yer alan sonuçlar içerisinden toplam 48 veya 50

modelin yer aldığı topluluk yaklaşımlarının sonuçları üzerinden yorumlar yapılmıştır. Sonuçlar incelendiğinde hedef veri kümesi için en iyi sonuçların ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme ile elde edildiği görülmektedir. Bu durum eğitim hatasına göre ağırlıklandırma işleminin tahmin sonuçlarını iyileştirdiğini göstermektedir. İkinci en iyi sonuç kümeleme ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Üçüncü en iyi sonuç ise kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir.

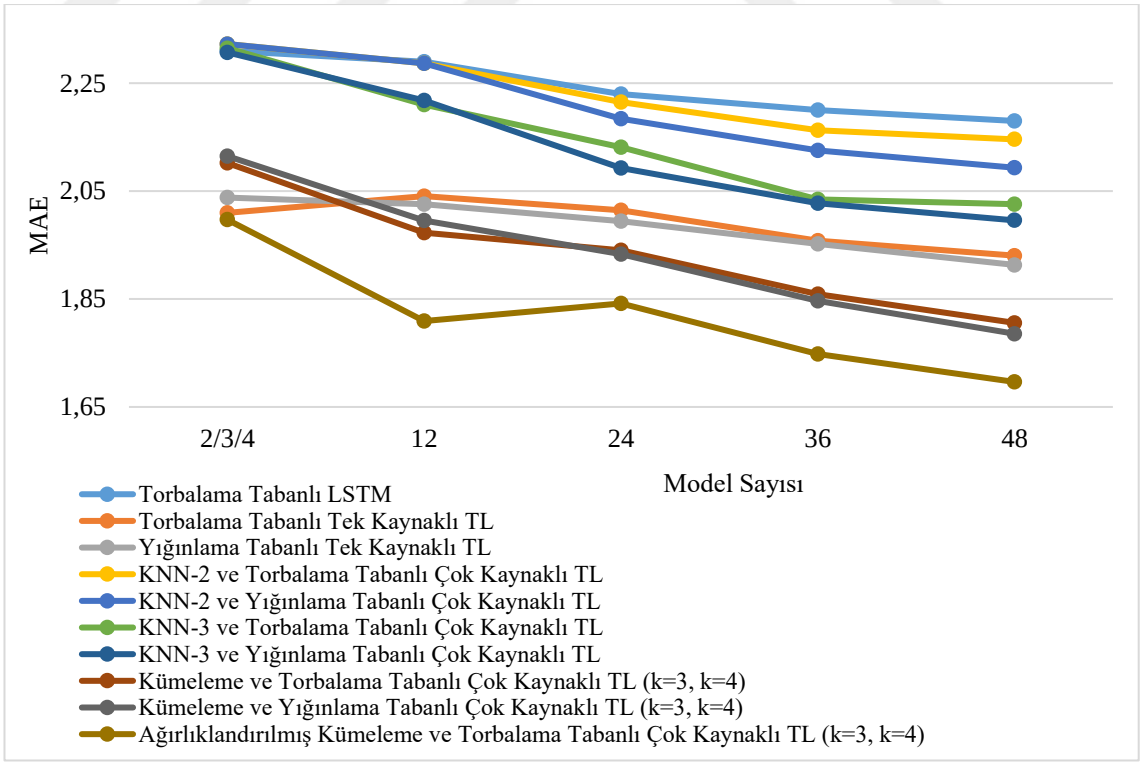
Çok kaynaklı transfer öğrenme yaklaşımlarının tek kaynaklı transfer öğrenme yaklaşımlarına göre performans açısından üstün olduğu görülmektedir. Diğer yandan, k-en yakın komşuluğun uygulandığı uzaklık tabanlı çoklu kaynak seçiminin tek kaynaklı derin transfer öğrenmeye göre tahmin performansını iyileştiremediği görülmektedir.

Her yaklaşım için eğitim süresi Denklem (4.1) ile hesaplanmıştır. Eğitim süreleri açısından incelendiğinde en iyi sonuçların KNN ve topluluk tabanlı derin transfer öğrenme ile elde edildiği görülmektedir. İkinci ve üçüncü en kısa eğitim sürelerini sırasıyla kümeleme ve ağırlıklandırılmış kümeleme tabanlı çok kaynaklı derin transfer öğrenme yöntemleri vermiştir.

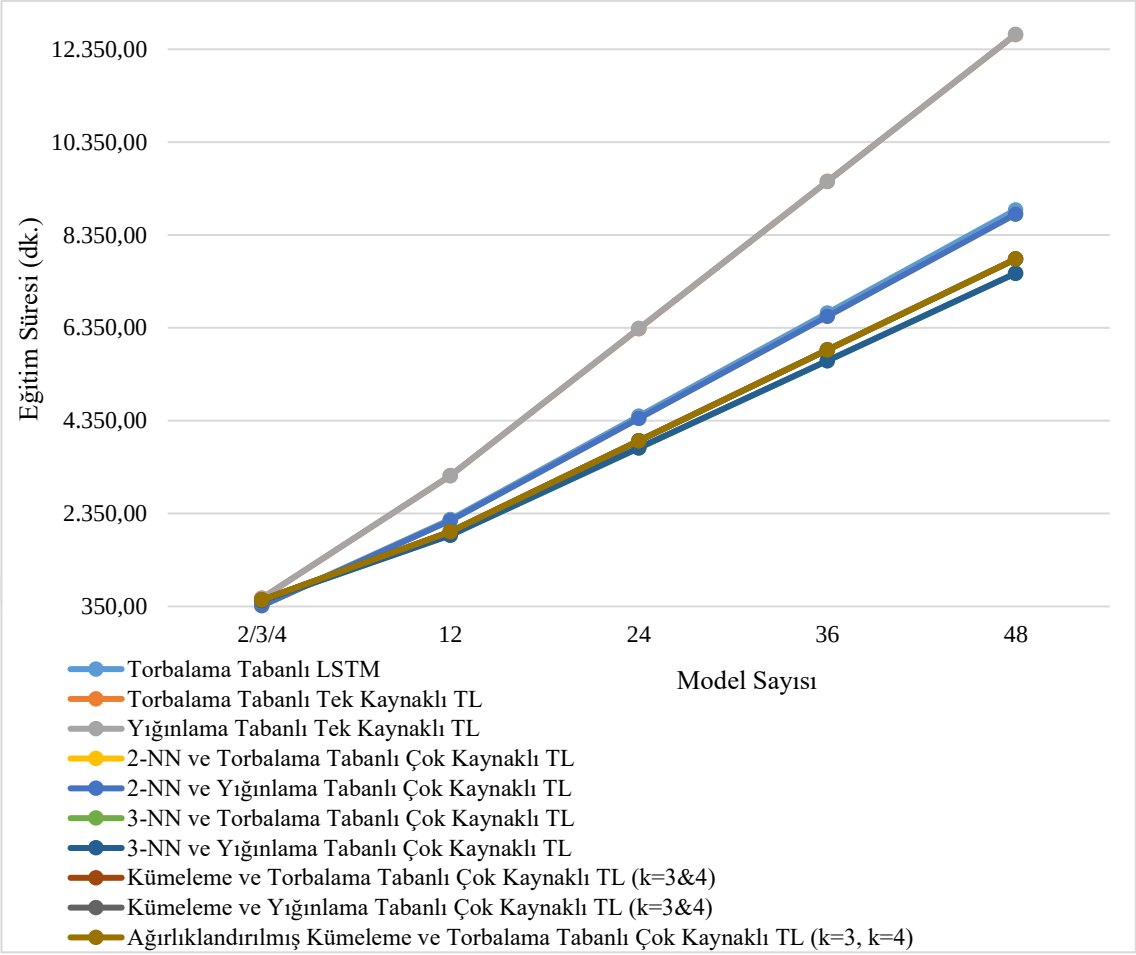
Veri özellikleri açısından sonuçlar değerlendirildiğinde şu gözlemler yapılmıştır. Küme sayısı 3 ve 4 iken Pharma-Seri_18 hedef verisi LearnX-Seri_4, LearnX-Seri_5 ve Pharma-Seri_17 ile aynı kümededir. Pharma hem mevsimsellik hem de trende sahipken LearnX mevsimsellik içermektedir. Ayrıca, hem Pharma hem de LearnX veri kümeleri tek bir ürüne ait satışları içeren zaman serilerinden oluşmaktadır. Diğer bir ifadeyle, Pharma-Seri_17 bir ilacın satış verileri iken Pharma-Seri_18 başka bir ilacın satış verileridir. Bu seriler hem mevsimsellik içermeleri hem de tek bir ürüne ait satışlardan oluşmaları nedeniyle ortak özelliklere sahiptir. Bu nedenle verilerin aynı kümeye atanması ve birbiri için kaynak olarak kullanılmaları transfer öğrenmedeki aktarım başarısını arttırmıştır. Pharma-Seri_18 hedef verisi için hem aynı veri kümesinden farklı bir serinin (Pharma-Seri_17) hem de benzer özelliğe sahip LearnX verisinin kaynak olarak kullanılması hedef verideki tahmin performansını iyileştirmiştir.



Şekil 4.9. Pharma-Seri_18 verisi için topluluk yöntemlerin RMSE değerleri



Şekil 4.10. Pharma-Seri_18 verisi için topluluk yöntemlerin MAE değerleri



Şekil 4.11. Pharma-Seri_18 verisi için topluluk yöntemlerin eğitim süreleri

Transfer öğrenmenin etkisi açısından sonuçlar incelenmiştir. Transfer öğrenmenin uygulandığı yaklaşımlarda (LSTM-TL) transfer öğrenmenin yapılmadığı torbalama tabanlı LSTM yöntemine göre tahmin başarısının iyileştiği gözlenmiştir. Ek 18'de yer alan sonuçlar incelendiğinde en başarısız sonuçların torbalama tabanlı LSTM yöntemi ile elde edildiği görülmektedir. Transfer öğrenmenin uygulandığı tüm yaklaşımlarda LSTM yöntemine kıyasla daha başarılı sonuçlar elde edilmiştir. Transfer öğrenmenin etkisini gösterebilmek için transfer öğrenmenin uygulandığı yaklaşımlar içerisinde en başarılı sonuca sahip yöntem ile transfer öğrenmenin uygulanmadığı LSTM yönteminin sonucu karşılaştırılmıştır. Transfer öğrenmenin uygulandığı yaklaşımlar içerisinde en başarılı sonuç ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yöntemidir. Bu yöntemde RMSE ve MAE değerleri sırasıyla 2,26 ile 1,70'tir. Transfer öğrenmenin uygulanmadığı torbalama tabanlı LSTM yönteminde ise RMSE ve MAE değerleri sırasıyla 2,88 ile 2,18'dir. Bu durumda, transfer öğrenme ile RMSE ve

MAE performans metriklerine göre sırasıyla %21,53 ve %22,02 iyileşme sağlanmıştır. RMSE performans metriği için bu yüzde iyileşme $((2{,}88 - 2{,}26) / 2{,}88) * 100 = \%\,21{,}53$ denklemi ile hesaplanmıştır. Benzer şekilde, MAE performans metriği için yüzde iyileşme hesaplanmıştır. Öte yandan, transfer öğrenmenin uygulandığı yaklaşımlarda eğitim süreleri daha uzundur.

Favorita-Seri_16 için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dakika) Çizelge 4.19'da verilmektedir. En iyi ilk iki sonuç sırasıyla küme sayısı 4 ve 3 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Üçüncü en iyi sonuç küme sayısı 4 iken kümeleme ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme ile elde edilmiştir.

Bu hedef veride, çok kaynaklı transfer öğrenme, tek kaynaklı transfer öğrenmeye göre daha başarılı sonuçlar vermiştir. Çok kaynaklı kümeleme tabanlı yaklaşımlarda, küme sayısının 4 olduğu durumda küme sayısının 3 olduğu duruma göre daha başarılı sonuçlar elde edilmiştir. Burada, küme sayısı 3'ten 4'e çıkarıldığında Favorita-Seri_16'nın yer aldığı 1. kümeden sadece Rossmann-Seri_15 ayrılmaktadır. Buna göre, iyileşme performansının küme sayısına ve kümedeki kaynak verilere bağlı olduğu ve ağırlıklandırılmış kümeleme yaklaşımının performans açısından öne çıktığı gözlenmiştir.

Eğitim süreleri açısından sonuçlar incelendiğinde, küme sayısının 3 olduğu çok kaynaklı derin transfer öğrenme yaklaşımları öne çıkmıştır. Buna göre; sırasıyla kümeleme ve torbalama tabanlı, ağırlıklandırılmış kümeleme tabanlı ile kümeleme ve yığınlama tabanlı derin transfer öğrenme yaklaşımları sırasıyla en kısa eğitim sürelerine sahiptir.

Veri özellikleri açısından sonuçlar incelendiğinde en başarılı sonucun elde edildiği küme sayısının 4 olduğu durumda Favorita-Seri_16 hedef verisi için Walmart-Seri_1, Walmart-Seri_2 ve Tv_Sales-Seri_14 kaynak veri olarak kullanılmıştır. Favorita ve Walmart perakende satış verileridir ve mevsimsellik içermektedir. Tv_Sales ise bir ürünün (TV) farklı modellerine ait toplam satışları içeren satış verisidir ve trende sahiptir. Küme sayısı 3 iken bu serilere ek olarak Rossmann-Seri_15 kaynak olarak kullanılmıştır. Rossmann perakende satış verisidir ve mevsimsellik içermektedir. Bu serinin kaynak olarak

kullanılması başarıyı küçük bir oranda düşürmüştür. Diğer yandan, küme sayısının 3 olduğu durumda ağırlıklandırılmış kümeleme yaklaşımı, kümeleme tabanlı yaklaşıma göre daha başarılıdır. Bu durum Rossmann-Seri_15'in kaynak olarak dahil edilmesi durumunda ağırlıklandırma yoluyla başarının daha fazla düşmesi engellenmiştir.

Transfer öğrenmenin etkisi açısından sonuçlar incelenmiştir. Transfer öğrenmenin uygulandığı tüm yaklaşımlarda transfer öğrenmenin yapılmadığı torbalama tabanlı LSTM yöntemine göre tahmin başarısının iyileştiği görülmektedir. Pharma-Seri_18'in kaynak olarak kullanıldığı torbalama tabanlı tek kaynaklı derin transfer öğrenme yaklaşımı ile torbalama tabanlı LSTM yönteminin tahmin hataları birbirine oldukça yakındır. Bu iki yaklaşıma ait sonuçlar en başarısız sonuçlardır. Burada benzemezlik matrisine göre en yakın serinin kaynak olarak seçilmesinin her zaman başarıyı arttıramayacağını göstermektedir. Bu durum, aslında kümelemeye dayalı kaynak seçiminin önemini de göstermektedir. Transfer öğrenmenin uygulandığı yaklaşımlar içerisinde en başarılı sonuç ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yöntemidir. Bu yöntemde RMSE ve MAE değerleri sırasıyla 13 897,86 ile 9 039,41'dir. Torbalama tabanlı LSTM yönteminde ise RMSE ve MAE değerleri sırasıyla 14 371,89 ile 9 448,76'tır. Bu durumda, transfer öğrenme ile RMSE ve MAE performans metriklerine göre sırasıyla %3,30 ve %4,33 iyileşme sağlanmıştır. Öte yandan, kümeleme tabanlı transfer öğrenme yaklaşımlarının eğitim süreleri torbalama tabanlı LSTM yöntemine kıyasla daha kısadır. Bu durum, transfer öğrenmede kaynak kümelerin eğitim süresi açısından belirleyici olduğunu göstermektedir.

Her iki hedef verisi için model sayısı arttıkça tahmin performansının iyileştiği gözlenmiştir. Model sayısının küçük değerlerinde daha yüksek oranlarda iyileşme olduğu, model sayısı 40'tan fazla olduğunda ise marjinal faydanın azaldığı tespit edilmiştir. Ayrıca, model sayısının artması ile eğitim süreleri artmaktadır.

Özet olarak, kümeleme ve topluluk tabanlı çok kaynaklı transfer öğrenme tahmin sonuçlarını iyileştirme potansiyeline sahiptir. Ancak hem tek kaynaklı hem de çok kaynaklı transfer öğrenmede uzaklığa dayalı seçim stratejileri tahmin sonuçlarını iyileştirmeyi garanti edememektedir.

Çizelge 4.19. Favorita-Seri_16 hedef verisi için yöntemlerin performanslarının karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL											
			Tek Kaynaklı TL		Çok Kaynaklı TL									
					KNN Tabanlı				Kümeleme Tabanlı				Ağırlıklandırılmış Kümeleme Tabanlı	
					k=2		k=3		k=3		k=4		k=3	k=4
			Seri_18		Seri_18-17		Seri_18-17-26		Seri_1-2-14-15		Seri_1-2-14		Seri_1-2-14-15	Seri_1-2-14
		Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Torbalama
RMSE	2/3/4	14 690,17	14 700,24	14 702,88	14 670,12	14 671,89	14 682,47	14 682,69	14 660,85	14 661,07	14 640,25	14 638,47	14 628,39	14 603,44
MAE		9 732,53	9 739,75	9 741,50	9 719,00	9 720,17	9 727,97	9 728,12	9 712,86	9 713,01	9 693,32	9 692,15	9 691,36	9 668,95
Süre		166,84	464,18	464,27	442,05	442,14	490,82	490,92	193,51	193,63	170,60	170,70	193,54	170,63
RMSE	12	14 613,75	14 613,26	14 611,58	14 590,98	14 588,63	14 601,56	14 599,21	14 541,42	14 540,08	14 517,42	14 519,08	14 471,42	14 421,42
MAE		9 681,26	9 681,33	9 678,22	9 660,70	9 657,99	9 673,58	9 672,02	9 627,89	9 627,00	9 582,21	9 583,30	9 581,54	9 518,84
Süre		1 001,03	2 785,08	2 785,34	2 652,30	2 652,56	1 963,27	1 963,53	721,06	721,32	682,39	682,65	721,21	682,53
RMSE	24	14 522,98	14 522,37	14 510,03	14 498,87	14 486,55	14 521,28	14 508,94	14 456,86	14 444,58	14 400,18	14 390,94	14 357,52	14 301,25
MAE		9 614,09	9 615,28	9 607,11	9 569,96	9 561,83	9 614,55	9 606,38	9 542,23	9 534,12	9 467,16	9 461,09	9 476,66	9 402,12
Süre		2 042,07	5 570,17	5 570,90	3 104,73	3 105,46	3 926,54	3 927,27	1 442,13	1 442,86	1 364,77	1 365,50	1 442,30	1 364,94
RMSE	36	14 443,63	14 442,33	14 426,86	14 402,99	14 387,56	14 452,99	14 437,51	14 322,88	14 307,54	14 301,85	14 286,53	14 223,98	14 092,54
MAE		9 532,57	9 532,64	9 522,43	9 469,01	9 458,87	9 539,68	9 529,46	9 416,34	9 406,26	9 323,05	9 313,07	9 351,32	9 186,61
Süre		3 009,70	8 355,25	8 356,35	4 657,10	4 658,19	5 889,82	5 890,91	2 163,19	2 164,28	2 047,16	2 048,25	2 163,38	2 047,35
RMSE	48	14 371,89	14 371,26	14 355,03	14 312,25	14 296,09	14 327,88	14 313,70	14 221,57	14 203,49	14 145,88	14 127,90 **(3)**	13 987,25 **(2)**	13 897,86 **(1)**
MAE		9 448,76	9 448,15	9 437,48	9 329,83	9 319,29	9 419,63	9 410,31	9 270,72	9 258,94	9 200,72	9 189,03 **(3)**	9 117,97 **(2)**	9 039,41 **(1)**
Süre		4 101,14	11 140,34	11 141,79	6 209,46	6 210,92	7 853,09	7 854,54	2 884,26 **(1)**	2 885,71 **(3)**	3 096,10	3 097,56	2 884,47 **(2)**	3 096,10

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

4.3.3. Yeni veri setleri ile önerilen çözüm yaklaşımlarının performanslarının değerlendirilmesi

Tek değişkenli zaman serileri için geliştirilen kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımlarının performansı yeni veri kümeleri ile değerlendirilmiştir. Bu kapsamda, Çizelge 4.7'de yer alan veri kümelerine ek olarak Çizelge 3.1'de yer alan Supermarket veri kümesi kullanılmıştır. Ayrıca, farklı bir Walmart mağazasına ait satışları içeren Seri_3 ile LearnX verisinden farklı bir kursun satışlarını içeren Seri_6 veri kümesi ele alınmıştır. Böylece, toplam üç veri çalışmaya dahil edilmiştir ve 14 satış verisine Bölüm 3.5.3'te açıklanan yöntem Şekil 3.16'daki akış şemasına göre uygulanmıştır. Bu kapsamda, her adım için elde edilen sonuçlar aşağıda verilmiştir:

Adım 1-3: Bölüm 4.3.2 ile aynıdır.

Adım 4: Benzemezlik matrisinin oluşturulması

Bu adımda, yeni veri kümelerinin Çizelge 4.13'te yer alan seçilen öznitelik değerleri belirlenmiştir. Ardından benzemezlik matrisi oluşturulmuştur. Bu kapsamda elde edilen sonuçlar Ek19-23'te verilmiştir. Sonuç olarak, tüm bu uzaklıklar Denklem (3.14)'teki formülde kullanılarak benzemezlik matrisi Çizelge 4.20'deki gibi elde edilmiştir.

Çizelge 4.20. Benzemezlik matrisi

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0,00	0,06	0,11	0,35	0,37	0,31	0,48	0,77	0,53	0,50	0,43	0,31	0,41	0,36
Seri_2	0,06	0,00	0,12	0,34	0,37	0,32	0,50	0,74	0,55	0,52	0,44	0,31	0,42	0,36
Seri_3	0,11	0,12	0,00	0,30	0,32	0,27	0,39	0,86	0,48	0,44	0,39	0,29	0,40	0,33
Seri_4	0,35	0,16	0,30	0,00	0,13	0,07	0,37	0,82	0,44	0,34	0,29	0,25	0,36	0,40
Seri_5	0,37	0,34	0,32	0,13	0,00	0,14	0,40	0,84	0,46	0,36	0,31	0,27	0,38	0,42
Seri_6	0,31	0,32	0,27	0,07	0,14	0,00	0,37	0,80	0,41	0,34	0,26	0,23	0,33	0,37
Seri_14	0,48	0,37	0,39	0,37	0,40	0,37	0,00	0,98	0,45	0,40	0,45	0,40	0,48	0,44
Seri_15	0,77	0,62	0,86	0,82	0,84	0,80	0,98	0,00	0,80	0,77	0,65	0,81	0,70	0,76
Seri_16	0,53	0,52	0,48	0,44	0,46	0,41	0,45	0,80	0,00	0,36	0,32	0,44	0,38	0,38
Seri_17	0,50	0,46	0,44	0,34	0,36	0,34	0,40	0,77	0,36	0,00	0,19	0,29	0,28	0,32
Seri_18	0,43	0,55	0,39	0,29	0,31	0,26	0,45	0,65	0,32	0,19	0,00	0,26	0,21	0,29
Seri_25	0,31	0,46	0,29	0,25	0,27	0,23	0,40	0,81	0,44	0,29	0,26	0,00	0,23	0,34
Seri_26	0,41	0,38	0,40	0,36	0,38	0,33	0,48	0,70	0,38	0,28	0,21	0,23	0,00	0,26
Seri_27	0,36	0,36	0,33	0,40	0,42	0,37	0,44	0,76	0,38	0,32	0,29	0,34	0,26	0,00

Adım 5: Zaman serilerinin kümelenmesi

Bu adımda, benzemezlik matrisi girdi olarak alınmıştır ve DAE tabanlı derin kümeleme uygulanmıştır. DAE tabanlı derin kümelemede Çizelge 4.4'te verilen hiperparametreler ve değer aralıkları kullanılarak ızgara arama yöntemi ile parametre optimizasyonu yapılmıştır. Parametre optimizasyonu sonucu elde edilen parametrelerle en iyi sonuçların elde edildiği üç farklı temsil alan boyutuna ve üç farklı küme sayısına göre derin kümeleme sonuçları Çizelge 4.21'de verilmektedir. Silhouette katsayısına göre en iyi kümeleme sonucu temsil alan boyutu 3 ve küme sayısı 3 ile temsil alan boyutu 4 ve küme sayısı 3 olduğu durumlarda elde edilmiştir. Dunn indeksine göre en iyi kümeleme sonucu temsil alan boyutu 4 ve küme sayısı 3 ile 4 olduğu durumlarda, Calinski-Harabasz indeksine göre en iyi kümeleme sonucu temsil alan boyutu 3 ve küme sayısı 4 olduğu durumda elde edilmiştir. Tüm indekslere göre değerlendirme yapıldığında nispeten daha iyi değerlere sahip olan temsil alan boyutu 3 iken oluşan kümelenmeler ele alınmıştır. Bu nedenle, temsil alan boyutu 3 iken aynı kümedeki satış verilerinin kaynak olarak kullanıldığı durumlarda negatif transfer öğrenme yaşanıp yaşanmayacağı araştırılmıştır. Bu araştırma için Ek 14'te verilen tüm kaynak-hedef eşleşmelerine ait tek değişkenli zaman serileri için tek kaynaklı derin transfer öğrenmedeki RMSE başarı yüzdeleri kullanılmıştır. Bu araştırma sonucunda Çizelge 4.22'de gösterildiği gibi temsil alan boyutunun 3 ve küme sayısının 3 ya da 4 olduğu kümeleme sonuçlarında negatif transfer öğrenme durumunun yaşanmayacağı tespit edilmiştir. Bu nedenle, bu çalışmada küme sayısının 3 ile 4 olduğu durumlar ele alınmıştır. Küme sayısı 2 iken satış verileri homojen bir şekilde ayrışmadığı için negatif transfer öğrenme durumu yaşandığı gözlenmiştir.

Çizelge 4.21. Derin kümeleme sonucunda küme atamaları

Seri	Temsil Alan Boyutu: 2			Temsil Alan Boyutu: 3			Temsil Alan Boyutu: 4		
	k=2	k=3	k=4	k=2	k=3	k=4	k=2	k=3	k=4
Seri_1	1	1	1	1	1	1	1	1	1
Seri_2	1	1	1	1	1	1	1	1	1
Seri_3	1	1	1	1	1	1	1	1	1
Seri_4	2	2	2	2	2	2	2	2	2
Seri_5	2	2	2	2	2	2	2	2	2
Seri_6	2	2	2	2	2	2	2	2	2
Seri_14	2	2	2	1	1	1	1	1	3
Seri_15	2	1	3	1	1	3	1	1	3
Seri_16	2	1	1	1	1	1	1	1	1
Seri_17	1	3	4	1	2	2	2	2	2
Seri_18	1	3	4	1	2	2	2	2	2
Seri_25	1	3	4	1	3	4	1	3	4
Seri_26	1	3	4	1	3	4	1	3	4
Seri_27	1	1	1	2	2	4	1	2	3
Silhouette Katsayısı:	0.50	0.63	0.69	0.61	**0.83**	0.79	0.69	0.82	0.83
Dunn İndeksi:	0.16	0.14	0.16	0.45	**1.12**	0.93	0.55	1.18	1.18
Calinski-Harabasz İndeksi:	20.95	36.00	72.54	21.60	**302.43**	315.70	21.72	126.21	186.18

Çizelge 4.22. Temsil alan boyutu 3, küme sayısı 3 ve 4 iken oluşan küme atamaları

Küme No/Küme Sayısı	Temsil Alan Boyutu: 3					
	k = 3	Negatif TL	Pozitif TL	k = 4	Negatif TL	Pozitif TL
1.Küme	Walmart-Seri_1	0	5	Walmart-Seri_1	0	4
	Walmart-Seri_2	0	5	Walmart-Seri_2	0	4
	Walmart-Seri_3	0	5	Walmart-Seri_3	0	4
	Tv_Sales-Seri_14	0	5	Tv_Sales-Seri_14	0	4
	Favorita-Seri_16	0	5	Favorita-Seri_16	0	4
	Rossmann-Seri_15	0	5			
2.Küme	LearnX-Seri_4	0	5	LearnX-Seri_4	0	4
	LearnX-Seri_5	0	5	LearnX-Seri_5	0	4
	LearnX-Seri_6	0	5	LearnX-Seri_6	0	4
	Pharma-Seri_17	0	5	Pharma-Seri_17	0	4
	Pharma-Seri_18	0	5	Pharma-Seri_18	0	4
	Supermarket-Seri_27	0	5			
3.Küme	Forklift-Seri_25	0	1	Rossmann-Seri_15		
	Forklift-Seri_26	0	1			
4.Küme				Forklift-Seri_25	0	1
				Forklift-Seri_26	0	1
				Supermarket-Seri_27	0	1
Toplam Deney Sayısı		0	62	Toplam Deney Sayısı	0	43
TL Aktarım Yüzdesi		%0	%100	TL Aktarım Yüzdesi	%0	%100

Adım 6-7: Kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımlarının deneysel sonuçları

Bu adımda, temsil alan boyutu 3 ve küme sayısının 3 ile 4 olduğu kümeler ele alınarak yeni eklenen veri setleri hedef verileri olarak kullanılmıştır. Bu hedef verilere tek değişkenli zaman serileri için geliştirilen dört farklı yaklaşım uygulanmıştır ve sonuçlar karşılaştırılmıştır.

Yaklaşımlarda kullanılmak üzere serilerin benzemezlik matrisine göre en yakın komşuları (k=1, k=2 ve k=3) belirlenmiştir ve Çizelge 4.23'te verilmiştir.

Çizelge 4.23. Serilerin benzemezlik matrisine göre en yakın komşuları

Seri	1-NN	2-NN	3-NN
Walmart-Seri_1	Seri_2	Seri_2-3	Seri_2-3-25
Walmart-Seri_2	Seri_2	Seri_2-3	Seri_2-3-26
Walmart-Seri_3	Seri_1	Seri_1-2	Seri_1-2-6
LearnX-Seri_4	Seri_6	Seri_6-5	Seri_6-5-2
LearnX-Seri_5	Seri_4	Seri_4-6	Seri_4-6-25
LearnX-Seri_6	Seri_4	Seri_4-5	Seri_4-5-25
Tv_Sales-Seri_14	Seri_4	Seri_4-6	Seri_4-6-2
Rossmann-Seri_15	Seri_2	Seri_2-18	Seri_2-18-26
Favorita-Seri_16	Seri_18	Seri_18-17	Seri_18-17-26
Pharma-Seri_17	Seri_18	Seri_18-26	Seri_18-26-25
Pharma-Seri_18	Seri_17	Seri_17-26	Seri_17-26-6
Forklift-Seri_25	Seri_26	Seri_26-6	Seri_26-6-4
Forklift-Seri_26	Seri_18	Seri_18-25	Seri_18-25-27
Supermarket-Seri_27	Seri_26	Seri_26-18	Seri_26-18-17

Walmart-Seri_3, LearnX-Seri_6 ve Supermarket-Seri_27 hedef verileri için önerilen çözüm yaklaşımlarının uygulama detayları aşağıda açıklanmaktadır:

1. Topluluk tabanlı tek kaynaklı derin transfer öğrenme yaklaşımı: Her hedef verisi için Çizelge 4.21'deki benzemezlik matrisine göre en yakın olan seri kaynak veri olarak seçilmiştir. Buna göre, Walmart-Seri_3 için Walmart-Seri_1, LearnX-Seri_6 için LearnX-Seri_4 ve Supermarket-Seri_27 için Forklift-Seri_26 kaynak veri olarak seçilmiştir.

2. KNN ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.23'e göre her hedef verisi için 2-NN ve 3-NN komşuları kaynak olarak kullanılmıştır. Walmart-Seri_3 hedef verisi için 2-NN olduğu durumda Walmart-Seri_1 ile Walmart-Seri_2, 3-NN olduğu durumda ise Walmart-Seri_1, Walmart-Seri_2 ve LearnX-Seri_6 kaynak veri olarak alınmıştır.

LearnX-Seri_6 hedef verisi için 2-NN'de LearnX-Seri_4 ile LearnX-Seri_5, 3-NN'de ise LearnX-Seri_4, LearnX-Seri_5 ve Forklift-Seri_25 kaynak veri olarak alınmıştır.

Supermarket-Seri_27 hedef verisi için 2-NN'de Forklift-Seri_26 ile Pharma-Seri_18, 3-NN'de ise Forklift-Seri_26, Pharma-Seri_18 ve Pharma-Seri_17 kaynak veri olarak alınmıştır.

3. Kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.22'ye göre Walmart-Seri_3 hedef verisi için küme sayısı 3 iken aynı kümede yer alan Walmart-Seri_1, Walmart-Seri_2, Tv_Sales-Seri_14, Rossmann-Seri_15 ve Favorita-Seri_16 olmak üzere beş seri kaynak veri olarak kullanılmıştır. Küme sayısı 4 olduğu durumda ise Walmart-Seri_3 hedef verisinin olduğu kümede artık Rossmann-Seri_15 yer almamaktadır, bu nedenle bu seri dışında kalan diğer dört seri kaynak veri olarak kullanılmıştır.

LearnX-Seri_6 hedef verisi için küme sayısı 3 iken aynı kümede yer alan LearnX-Seri_4, LearnX-Seri_5, Pharma-Seri_17, Pharma-Seri_18 ve Supermarket-Seri_27 olmak üzere beş seri kaynak veri olarak kullanılmıştır. Küme sayısı 4 olduğu durumda ise LearnX-Seri_6 hedef verisinin olduğu kümede artık Supermarket-Seri_27 yer almamaktadır, bu nedenle bu seri dışında kalan diğer dört seri kaynak veri olarak kullanılmıştır.

Supermarket-Seri_27 hedef verisi için küme sayısı 3 iken aynı kümede yer alan LearnX-Seri_4, LearnX-Seri_5, LearnX-Seri_6, Pharma-Seri_17 ve Pharma-Seri_18 olmak üzere beş seri kaynak veri olarak kullanılmıştır. Küme sayısı 4 olduğu durumda ise Supermarket-Seri_27, Forklift-Seri_25 ve Forklift-Seri_26 ile farklı bir kümede kümelenmiştir. Bu nedenle bu iki seri kaynak veri olarak kullanılmıştır.

4. Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Walmart-Seri_3, LearnX-Seri_6 ve Supermarket-Seri_27 hedef verileri için üçüncü yaklaşımda belirtilen kaynak veriler ağırlıklandırılarak kullanılmıştır.

Transfer öğrenmenin etkisini görebilmek amacıyla yukarıdaki yaklaşımlara ek olarak her üç hedef verisinde LSTM yöntemi ile torbalama topluluk yöntemi uygulanmıştır.

Walmart-Seri_3 hedef verisi için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dakika) Çizelge 4.24'te verilmiştir. En başarılı sonuç 2-NN ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme ile elde edilmiştir. İkinci en başarılı sonuç 2-NN ve torbalama tabanlı çok kaynaklı derin transfer öğrenme ile elde edilmiştir. Üçüncü en başarılı sonuç ise küme sayısı 4 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir.

Bu hedef veride, çok kaynaklı transfer öğrenme, tek kaynaklı transfer öğrenmeye göre daha başarılı sonuçlar vermiştir. Çok kaynaklı durumlar içerisinde ise kümeleme ve ağırlıklandırılmış kümeleme tabanlı yöntemlerde küme sayısının 4 olduğu durumda küme sayısının 3 olduğu duruma göre daha başarılı sonuçlar elde edilmiştir. Burada küme sayısı 3'ten 4'e çıkarıldığında Walmart-Seri_3'ün yer aldığı kümeden sadece Rossmann-Seri_15 ayrılmaktadır ve bu durumda sonuçlar iyileşmektedir. Buradaki sonuç küme sayısına ve kümedeki kaynak verilere bağlı olarak sonucun iyileşebileceği ve her durumda ağırlıklandırılmış kümeleme yaklaşımının daha iyi performans gösterdiği belirtilebilir.

Walmart-Seri_3 hedef verisinde en başarısız iki sonuç 3-NN tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile elde edilmiştir. Halbuki, bu hedef verisinde en iyi iki sonuç ise 2-NN tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile alınmıştır. 2-NN ile 3-NN arasındaki tek fark 3-NN durumunda LearnX-Seri_6 verisinin üçüncü kaynak veri olarak kullanılmasıdır. LearnX-Seri_6'nın kaynak veri olarak kullanılması ise Walmart-Seri_3 hedef verisi için transfer öğrenme başarısını düşürmüştür. Bu durum, uzaklık tabanlı kaynak seçiminin transfer öğrenmedeki başarıyı sağlamada yeterli

olmayabileceğini göstermektedir. Buna karşın, kümeleme tabanlı yaklaşımların daha etkin olduğu gözlenmiştir.

Her yaklaşım için eğitim süresi Denklem (4.1) ile hesaplanmıştır. Eğitim süreleri açısından incelendiğinde Walmart-Seri_3 hedef verisi için transfer öğrenme yaklaşımları içerisinden 2-NN tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları en kısa iki süreye sahiptir. Üçüncü ve dördüncü en kısa eğitim süreleri ise tek kaynaklı derin transfer öğrenme yaklaşımları ile elde edilmiştir. Küme sayısının 3 olduğu durumda kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları en uzun eğitim sürelerine sahiptir.

Ayrıca, veri özellikleri açısından sonuçlar değerlendirilmiştir. Walmart-Seri_3 hedef verisinde en iyi iki sonuç 2-NN tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile Walmart-Seri_1 ve Walmart-Seri_2 kaynak olarak kullanıldığı durumda elde edilmiştir. Bu seriler Walmart veri kümesine ait farklı mağazaların satış verileridir ve birbiri için kaynak veri olarak kullanılmaları transfer öğrenmedeki başarıyı arttırmaktadır. Öte yandan, üçüncü en iyi sonucun elde edildiği ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yönteminde Walmart-Seri_1, Walmart-Seri_2, Tv_Sales-Seri_14 ve Favorita-Seri_16 kaynak veri olarak kullanılmaktadır. Tv_Sales-Seri_14 dışındaki tüm seriler mevsimsellik içerirken Tv_Sales-Seri_14 trend içermektedir. Mevsimsellik içeren veri kümelerinin birbiri için kaynak veri olarak kullanılması sonucu iyileştirmektedir. Aynı zamanda Favorita verisi perakende satış verisidir ve benzer şekilde perakende satış verisi olan Walmart-Seri_3 için kaynak olarak kullanılması performansı iyileştirmiştir.

Transfer öğrenmenin etkisi açısından sonuçlar incelendiğinde transfer öğrenmenin uygulandığı tüm yaklaşımlarda transfer öğrenmenin yapılmadığı torbalama tabanlı LSTM yöntemine göre iyileşme sağlanmıştır. Transfer öğrenmenin uygulandığı yaklaşımlar içinde en başarılı sonuç 2-NN ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme ile elde edilmiştir. Bu yöntemle torbalama tabanlı LSTM yöntemine kıyasla RMSE ve MAE performans metriklerine göre sırasıyla %8,14 ve %8,06 iyileşme sağlanmıştır. Öte yandan, transfer öğrenmenin uygulandığı yaklaşımlarda eğitim süreleri daha uzundur.

Çizelge 4.24. Tek değişkenli Walmart-Seri_3 hedef verisi için yöntemlerin performanslarının karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL																
			Tek Kaynaklı TL	Çok Kaynaklı TL															
				KNN Tabanlı					Kümeleme Tabanlı							Ağırlıklandırılmış Kümeleme Tabanlı			
				k=2		k=3			k=3			k=4				k=3		k=4	
			Seri_1	Seri_1-2		Seri_1-2-6		MS	Seri_1-2-14-15-16		MS	Seri_1-2-14-16		MS	Seri_1-2-14-15-16	MS	Seri_1-2-14-16	
		Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama		Torbalama	
RMSE	2/3	20 939,88	20 095,85	20 087,23	19 795,85	19 794,57	20 349,01	20 350,87	5	19 110,78	19 108,81	4	19 002,63	18 987,03	5	18 802,52	4	18 780,85	
MAE		18 042,93	17 315,67	17 308,24	16 528,70	16 527,63	17 116,46	17 118,03		16 466,88	16 465,18		15 866,39	15 853,37		16 201,27		15 681,22	
Süre		5,58	8,70	8,79	7,91	8,00	18,11	18,20		144,19	144,28		119,72	119,81		144,23		119,75	
RMSE	12	20 051,34	19 280,13	19 278,58	18 502,36	18 483,27	19 470,26	19 460,12	15	18 520,26	18 489,77	12	18 442,52	18 339,78	15	18 098,85	12	18 002,36	
MAE		16 922,42	16 271,56	16 270,25	15 421,17	15 405,26	16 256,85	16 248,38		15 630,27	15 604,53		15 371,29	15 285,66		15 274,61		15 004,43	
Süre		33,46	52,19	52,45	47,47	47,73	72,45	72,71		432,56	432,82		359,15	359,41		432,71		359,29	
RMSE	24	19 374,16	18 682,89	18 651,44	17 557,11	17 517,95	18 752,56	18 698,87	25	18 011,26	17 900,62	24	17 998,33	17 775,26	25	17 617,29	24	17 517,97	
MAE		16 296,47	15 715,02	15 688,56	14 569,34	14 536,85	15 629,70	15 584,95		15 150,08	15 057,01		14 935,48	14 750,37		14 818,69		14 536,86	
Süre		66,92	104,38	105,11	94,93	95,66	144,89	145,62		720,93	721,66		718,29	719,02		721,10		718,46	
RMSE	36	18 933,51	18 275,59	18 205,56	17 288,95	17 211,22	18 355,23	18 290,03	35	17 648,59	17 550,84	36	17 619,99	17 475,17	35	17 420,15	36	17 298,65	
MAE		15 808,68	15 259,35	15 200,88	13 846,86	13 784,60	15 231,64	15 177,54		14 735,83	14 654,21		14 111,99	13 996,00		14 545,09		13 854,62	
Süre		100,37	156,57	157,66	142,40	143,49	217,34	218,43		1 009,30	1 010,39		1 077,44	1 078,53		1 009,49		1 077,63	
RMSE	48	18 610,83	17 998,87	17 750,78	17 100,07 **(2)**	17 096,45 **(1)**	18 003,45	17 940,81	50	17 450,12	17 337,07	48	17 428,87	17 281,07	50	17 250,74	48	17 111,57 **(3)**	
MAE		15 526,58	15 001,53	14 794,75	14 277,84 **(2)**	14 274,81 **(1)**	15 032,12	14 979,82		14 544,16	14 449,93		14 552,37	14 428,96		14 377,98		14 287,44 **(3)**	
Süre		133,83*****	208,76	210,22 **(3)**	189,87 **(1)**	191,33 **(2)**	289,79	291,24		1 441,86	1 443,31		1 436,58	1 438,04		1 442,07		1 436,79	

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

LearnX-Seri_6 hedef verisi için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dk.) Çizelge 4.25'te verilmiştir. En iyi sonuç küme sayısı 3 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. İkinci ve üçüncü en iyi sonuçlar sırasıyla küme sayısı 3 ve 4 iken kümeleme ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme yöntemleri ile elde edilmiştir. Ayrıca, sadece MAE performans ölçütüne göre üçüncü en iyi sonuç ise küme sayısı 4 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir.

LearnX-Seri_6 hedef verisinde en başarısız iki sonuç 3-NN tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile alınmıştır. Bu durum uzaklık tabanlı kaynak seçimi yaklaşımlarının diğer yaklaşımlara göre performans üstünlüğü sağlayamadığını göstermektedir. Öte yandan, kümeleme ve ağırlıklandırılmış kümeleme tabanlı çok kaynaklı derin transfer öğrenme yaklaşımlarının tek kaynaklı derin transfer öğrenme yaklaşımlarına göre daha iyi performansa sahip oldukları görülmektedir.

Eğitim süreleri açısından incelendiğinde LearnX-Seri_6 hedef verisi için KNN-3 tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları en kısa iki süreye sahiptir. Üçüncü ve dördüncü en kısa eğitim süreleri ise 2-NN tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile elde edilmiştir. Küme sayısının 4 olduğu durumda ise kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları en uzun eğitim sürelerine sahiptir.

Sonuçlar, veri özellikleri açısından değerlendirilmiştir. En iyi sonuç küme sayısı 3 iken LearnX-Seri_4, LearnX-Seri_5, Pharma-Seri_17, Pharma-Seri_18 ve Supermarket-Seri_27'nin kaynak olarak kullanıldığı ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Tüm bu seriler mevsimsellik içermektedir. Aynı zamanda LearnX ile Pharma verilerinin ortak özelliği tek bir ürünün satışına ait seriler olmasıdır. Diğer bir ifadeyle, LearnX ile Pharma'daki her bir seri farklı bir ürünün satışlarıdır. Bu nedenle, birbiri için kaynak olarak kullanılmaları sonucu iyileştirmiştir. Tek kaynaklı transfer öğrenmede ise sadece LearnX-Seri_4'ün kaynak olarak kullanılması ve 2-NN çok kaynaklı transfer öğrenmede sadece LearnX-Seri_4 ile LearnX-Seri_5'in kaynak olarak kullanılması sonuçları kümeleme tabanlı çok kaynaklı yaklaşımda olduğu kadar iyileştirmemiştir. Burada kümeleme ile

Pharma verilerinin de kaynak olarak kullanılması ve kaynak verilerinin ağırlıklandırılması performansı iyileştirmiştir. Ayrıca küme sayısı 3'ten 4'e çıkarıldığında Supermarket-Seri_27 verisi artık kaynak olarak kullanılmamaktadır. Bu durumda performansta küçük bir azalma olmuştur. Supermarket-Seri_27 serisi de mevsimsellik içermektedir ve LearnX-Seri_6 için TL başarısını arttırmıştır. 3-NN tabanlı çok kaynaklı yaklaşımlar ile en başarısız sonuçlar elde edilmiştir. Bu durumda LearnX-Seri_4 ile LearnX-Seri_5 serilerinin yanında Forklift-Seri_25 verisi de kaynak olarak kullanılmıştır ve 2-NN durumuna göre performansta düşüş yaşanmıştır. Forklift verisi de tek bir ürüne ait satışları içermektedir ancak trende sahip seyrek bir veridir. Bu nedenle LearnX-Seri_6 hedef verisine uzaklık açısından yakın bir seri olabilir ancak transfer öğrenme açısından fayda sağlamamaktadır.

Transfer öğrenmenin etkisi açısından sonuçlar incelendiğinde transfer öğrenmenin uygulandığı tüm yaklaşımlarda transfer öğrenmenin yapılmadığı torbalama tabanlı LSTM yöntemine göre iyileşme elde edilmiştir. Transfer öğrenmenin uygulandığı yaklaşımlar içinde en başarılı tahmin sonucu küme sayısı 3 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Bu yöntemle torbalama tabanlı LSTM yöntemine kıyasla RMSE ve MAE performans metriklerine göre sırasıyla %29,57 ve %29,59 oranlarında iyileşme sağlanmıştır. Transfer öğrenme ile tahmin sonuçlarında yaklaşık %30'a yakın iyileşme sağlanırken diğer taraftan transfer öğrenmenin uygulandığı yaklaşımlarda daha uzun eğitim sürelerine ihtiyaç duyulabilmektedir.

Supermarket-Seri_27 hedef verisi için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dk.) Çizelge 4.26'da verilmiştir. En başarılı tahmin sonuçları küme sayısı 3 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Küme sayısı 3 iken kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi en iyi ikinci performansa sahiptir. Küme sayısı 4 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ise en iyi üçüncü performansa sahiptir. Bu serideki sonuçlar aslında küme sayısı 3 olduğu durumdaki gibi kaynak veri sayısı ve çeşitliliğinin artması durumunda sonuçların daha iyi olabileceğini göstermektedir.

Çizelge 4.25. Tek değişkenli LearnX-Seri_6 hedef verisi için yöntemlerin performanslarının karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL															
			Tek Kaynaklı TL		Çok Kaynaklı TL													
					KNN Tabanlı					Kümeleme Tabanlı						Ağırlıklandırılmış Kümeleme Tabanlı		
					k=2		k=3			k=3			k=4			k=3		k=4
			Seri_4		Seri_4-5		Seri_4-5-25		MS	Seri_4-5-17-18-27		MS	Seri_4-5-17-18		MS	Seri_4-5-17-18-27	MS	Seri_4-5-17-18
		Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama		Torbalama
RMSE	2/3	87,48	71,12	71,71	72,30	72,37	77,30	77,19	5	70,07	69,98	4	72,27	70,07	5	66,98	4	70,01
MAE		72,73	59,13	59,62	60,11	60,17	64,27	64,18		58,26	58,19		59,62	57,81		55,69		57,76
Süre		17,28	37,08	37,17	35,09	35,18	43,53	43,63		403,67	403,83		424,80	424,93		403,71		394,64
RMSE	12	76,21	62,21	61,17	63,25	64,15	69,65	70,07	15	63,13	62,04	12	65,63	65,45	15	57,36	12	62,96
MAE		62,87	51,32	50,46	52,18	52,92	57,46	57,81		52,08	51,18		53,45	53,30		47,32		51,27
Süre		103,67	222,48	222,74	210,51	210,77	174,11	174,37		1 211,02	1 211,48		1 183,80	1 184,06		1 211,17		1 183,94
RMSE	24	68,53	56,17	55,01	57,78	56,16	64,89	63,13	25	55,56	54,76	24	57,49	57,11	25	50,84	24	56,03
MAE		55,81	45,74	44,80	47,05	45,74	52,85	51,41		45,25	44,60		46,80	46,49		41,40		45,61
Süre		207,34	444,96	445,69	421,03	421,76	348,22	348,95		2 018,37	2 019,13		2 367,60	2 368,33		2 018,54		2 367,77
RMSE	36	62,11	50,99	49,16	53,23	50,95	60,27	58,61	35	49,08	47,67	36	51,89	51,03	35	45,27	36	50,10
MAE		50,55	41,51	40,02	43,33	41,47	49,06	47,71		39,95	38,80		41,98	41,28		36,85		40,53
Süre		311,02	667,44	668,53	631,54	632,63	522,34	523,43		2 825,72	2 826,79		3 551,40	3 552,49		2 825,91		3 551,59
RMSE	48	59,09	48,63	46,77	50,27	48,56	57,66	55,55	50	44,38	42,88 **(2)**	48	46,41	43,97 **(3)**	50	41,62 **(1)**	48	44,38
MAE		47,82	39,10	37,84	40,67	39,28	46,65	44,94		35,90	34,69 **(2)**		36,98	35,36 **(3)**		33,67 **(1)**		35,36 **(3)**
Süre		414,69*****	889,92	891,38	842,06 **(3)**	843,51	696,45 **(1)**	697,90 **(2)**		4 036,75	4 038,27		4 735,20	4 736,66		4 036,96		4 735,41

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

Supermarket-Seri_27 hedef verisinde en başarısız sonuç veren yaklaşım torbalama tabanlı tek kaynaklı derin transfer öğrenme yöntemidir. 2-NN ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı ise ikinci en başarısız performansa sahiptir. Bu durum uzaklık tabanlı kaynak seçimi yaklaşımlarının diğer yaklaşımlara göre performans üstünlüğü sağlayamadığını göstermektedir. Öte yandan, kümeleme ve ağırlıklandırılmış kümeleme tabanlı çok kaynaklı derin transfer öğrenme yaklaşımlarının tek kaynaklı derin transfer öğrenme yaklaşımlarına göre daha üstün performansa sahip oldukları görülmektedir.

Eğitim süreleri açısından incelendiğinde Supermarket-Seri_27 hedef verisi için tek kaynaklı derin transfer öğrenme yaklaşımları en kısa iki süreye sahiptir. Küçük bir farkla küme sayısı 4 iken kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları da kısa eğitim sürelerine sahiptir. 3-NN tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ise en uzun eğitim sürelerine sahiptir.

Veri özellikleri açısından sonuçlar incelendiğinde Supermarket-Seri_27 hedef verisinde en başarılı sonuçlar küme sayısı 3 iken LearnX-Seri_4, LearnX-Seri_5, LearnX-Seri_6, Pharma-Seri_17 ve Pharma-Seri_18 verileri kaynak olarak kullanıldığında elde edilmiştir. Tüm bu seriler mevsimsellik içermektedir ve böylece Supermarket-Seri_27 hedef verisine olan transfer öğrenmede başarılı olmuşlardır. Küme sayısı 4 iken kümelenme değişerek Forklift-Seri_25 ile Forklift-Seri_26 serileri kaynak olarak kullanılmıştır ve ağırlıklandırma stratejisi ile en başarılı üçüncü sonuç elde edilmiştir. Hem Supermarket-Seri_27 hem de Forklift verisi azalma trendine sahiptir. En başarısız sonuç ise Forklift-Seri_26'nın kaynak olarak kullanıldığı tek kaynaklı transfer öğrenme ile alınmıştır. Benzer kaynak verilerle transfer öğrenme başarısında elde edilen bu iki farklı sonuç çözüm yaklaşımı farklılığından kaynaklanmaktadır. Bu örnek ağırlıklandırma stratejisi ile sonuçların iyileştirilebileceğini göstermiştir.

Transfer öğrenmenin etkisi açısından sonuçlar incelendiğinde transfer öğrenmenin uygulandığı tüm yaklaşımlarda transfer öğrenmenin yapılmadığı torbalama tabanlı LSTM yöntemine göre iyileşme sağlanmıştır. Transfer öğrenmenin uygulandığı yaklaşımlar içinde en başarılı sonuç küme sayısı 3 iken ağırlıklandırılmış kümeleme ve

torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Bu yöntemle torbalama tabanlı LSTM yöntemine kıyasla RMSE ve MAE performans metriklerine göre sırasıyla %15,29 ve %15,40 oranlarında iyileşme sağlanmıştır. Öte yandan, transfer öğrenmenin uygulandığı yaklaşımlarda daha uzun eğitim süreleri gözlenmiştir.

Özetlemek gerekirse, yeni veri setleri ile kümeleme ve ağırlıklandırılmış kümeleme yaklaşımlarının birçok deneyde performans üstünlüğü sağladığı gösterilmiştir. Deneylerin büyük bir bölümünde çok kaynaklı derin transfer öğrenme yaklaşımları tek kaynaklı derin transfer öğrenme yaklaşımlarına göre daha başarılı sonuçlar vermiştir. Çok kaynaklı derin transfer öğrenmede, ağırlıklandırılmış kümeleme tabanlı yaklaşımın diğer yaklaşımlara göre daha üstün performansa sahip olduğu gözlenmiştir. Çok kaynaklı derin transfer öğrenme yaklaşımlarından kümeleme tabanlı yaklaşım da başarılı sonuçlar vermiştir. Diğer yandan, KNN tabanlı yaklaşımın her zaman başarılı sonuçlar elde edemediği görülmüştür. Ayrıca, kümeleme tabanlı ve KNN tabanlı yaklaşımlarda genellikle yığınlama topluluk yönteminin torbalama topluluk yöntemine göre daha başarılı sonuçlar verdiği gözlemlenmiştir. Öte yandan, ağırlıklandırılmış kümeleme ve torbalama tabanlı yaklaşımın ise genellikle kümeleme ve yığınlama tabanlı yaklaşımdan daha iyi performansa sahip olduğu görülmüştür. Bu durum geliştirilen ağırlıklandırma mekanizmasının başarılı sonuçlar üretebildiğini göstermektedir.

Eğitim süreleri ise hedef veriye, kullanılan kaynak verilere ve yaklaşımlara bağlı olarak değişkenlik göstermektedir. Özellikle eğitim süresi kullanılan kaynak verilerin uzunluğuna (boyutuna) bağlıdır ve kaynak verilerin uzunluğu arttıkça eğitim süreleri de artmaktadır. Bu nedenle eğitim süresi açısından herhangi bir yöntem ön plana çıkmamaktadır.

Bu tez kapsamında geliştirilen çözüm yaklaşımları ile negatif transfer öğrenmeye neden olan kaynak-hedef eşleşmelerinin seçimi önlenmiştir. Negatif transfer öğrenmenin elimine edildiği durumlarda transfer öğrenme ile tahmin sonuçlarının iyileştirilebileceği gösterilmiştir.

Çizelge 4.26. Tek değişkenli Supermarket-Seri_27 hedef verisi için yöntemlerin performanslarının karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL																
			Tek Kaynaklı TL		Çok Kaynaklı TL														
					KNN Tabanlı					Kümeleme Tabanlı						Ağırlıklandırılmış Kümeleme Tabanlı			
					k=2		k=3			k=3			k=4			k=3		k=4	
			Seri_26		Seri_26-18		Seri_26-18-17		MS	Seri_4-5-6-17-18		MS	Seri_25-26		MS	Seri_4-5-6-17-18	MS	Seri_25-26	
		Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama		Torbalama	
RMSE	2/3	179,51	170,15	170,70	166,71	167,11	160,97	161,11	5	155,98	156,06	2	162,84	164,36	5	152,18	2	158,39	
MAE		138,88	131,64	132,07	123,84	124,13	119,57	119,68		111,96	112,02		125,99	127,16		109,23		122,54	
Süre		6,99	8,35	8,45	191,68	191,77	357,04	357,14		387,69	387,84		9,30	9,39		387,73		9,33	
RMSE	12	158,07	150,26	148,68	147,06	146,98	142,85	142,08	15	137,16	136,56	12	142,72	143,83	15	133,20	12	139,77	
MAE		117,42	111,62	110,44	105,56	105,50	102,53	101,98		96,15	95,73		106,02	106,84		93,38		103,83	
Süre		41,92	50,13	50,39	1 150,06	1 150,32	1 428,17	1 428,43		1 176,60	1 177,06		55,77	56,03		1 176,75		55,92	
RMSE	24	143,60	136,89	134,95	133,26	131,56	133,23	131,13	25	125,13	123,71	24	129,29	128,87	25	120,08	24	127,15	
MAE		103,07	98,26	96,86	93,42	92,23	93,40	91,93		87,32	86,33		92,80	92,50		83,79		91,27	
Süre		83,85	100,25	100,98	2 300,11	2 300,84	2 856,35	2 857,08		2 412,38	2 413,14		111,55	112,27		2 412,55		111,72	
RMSE	36	131,50	125,48	122,96	123,89	122,29	121,25	119,91	35	116,26	115,67	36	119,86	118,12	35	111,87	36	117,20	
MAE		92,19	87,97	86,20	86,45	85,34	84,61	83,68		81,08	80,67		84,03	82,81		78,02		82,16	
Süre		125,77	150,38	151,47	3 450,17	3 451,26	4 284,52	4 285,62		2 745,40	2 746,46		167,32	168,41		2 745,59		167,51	
RMSE	48	126,53	120,85	118,37	119,19	118,01	117,55	116,28	50	110,30 **(2)**	112,56	48	114,26	113,78	50	107,18 **(1)**	48	112,41 **(3)**	
MAE		88,30	84,33	82,60	83,12	83,30	81,98	81,09		76,87 **(2)**	78,45		79,73	79,40		74,70 **(1)**		78,44 **(3)**	
Süre		167,70*****	200,51 **(1)**	201,96 **(2)**	4 600,23	4 601,68	5 712,70	5 714,15		3 922,00	3 923,52		223,09 **(3)**	224,55		3 922,21		223,30	

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

4.4. Çok Değişkenli Zaman Serileri için Deneysel Çalışmalar

Bu bölümde çok değişkenli zaman serileri için geliştirilen kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımına ait deneysel çalışmalar sunulmuştur. Bu kapsamda, Bölüm 4.3.3'teki 14 satış verisi, Çizelge 3.2'de yer alan öznitelikler ile çok değişkenli satış verisi olarak ele alınmıştır. Bu veriler çok değişkenli LSTM uygulanmıştır.

Çok değişkenli satış verileri için geliştirilen yaklaşımların deneysel sonuçları aşağıda açıklanmıştır.

Adım 1: Uzaklık tabanlı benzemezlik matrisinin oluşturulması

Çok değişkenli satış verileri arasındaki DTW_D uzaklığı hesaplanmıştır ve DTW_D uzaklıkları Ek 24'te verilmiştir. DTW_D uzaklıklarına [0, 1] normalizasyonu uygulanmıştır. Normalize edilmiş DTW_D uzaklıkları Çizelge 4.27'de verilmiştir. Bu çizelgedeki DTW_D uzaklıkları kullanılarak kaynak seçimi yapılmıştır.

Çizelge 4.27. Benzemezlik matrisi (Normalize edilmiş DTW_D uzaklıkları)

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0,00	0,02	0,06	0,19	0,19	0,19	0,20	0,72	0,24	0,29	0,29	0,10	0,10	0,10
Seri_2	0,02	0,00	0,09	0,24	0,24	0,24	0,25	0,61	0,31	0,36	0,36	0,13	0,13	0,13
Seri_3	0,06	0,09	0,00	0,05	0,05	0,05	0,05	0,92	0,05	0,07	0,07	0,03	0,03	0,03
Seri_4	0,19	0,24	0,05	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_5	0,19	0,24	0,05	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_6	0,19	0,24	0,05	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_14	0,20	0,25	0,05	0,00	0,00	0,00	0,00	1,00	0,01	0,00	0,00	0,00	0,00	0,00
Seri_15	0,72	0,61	0,92	1,00	1,00	1,00	1,00	0,00	0,98	1,00	1,00	1,00	1,00	1,00
Seri_16	0,24	0,31	0,05	0,02	0,02	0,02	0,01	0,98	0,00	0,02	0,02	0,02	0,02	0,02
Seri_17	0,29	0,36	0,07	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_18	0,29	0,36	0,07	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_25	0,10	0,13	0,03	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_26	0,10	0,13	0,03	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_27	0,10	0,13	0,03	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00

Adım 2: Zaman serilerinin kümelenmesi

Benzemezlik matrisi girdi olarak alınmıştır ve DAE tabanlı derin kümeleme uygulanmıştır. DAE tabanlı derin kümelemede Çizelge 4.19'da verilen hiperparametreler ve değer aralıkları kullanılarak ızgara arama yöntemi ile parametre optimizasyonu

yapılmıştır. Buna göre, en iyi sonuçların elde edildiği iki farklı temsil alan boyutu ve üç farklı küme sayısına göre derin kümeleme sonuçları Çizelge 4.28'da verilmiştir. Silhouette katsayısı, Dunn indeksi ve Calinski-Harabasz indeksi olmak üzere her üç kümeleme doğrulama yöntemine göre en iyi kümeleme sonucu temsil alan boyutu 3 ve küme sayısı 4 iken elde edilmiştir. Her üç kümeleme doğrulama yöntemine göre en iyi ikinci kümeleme sonucu temsil alan boyutu 3 ve küme sayısı 3 iken elde edilmiştir.

Çizelge 4.28. Derin kümeleme sonucunda küme atamaları

Seri	Temsil Alan Boyutu: 3			Temsil Alan Boyutu: 4		
	k=2	k=3	k=4	k=2	k=3	k=4
Seri_1	1	1	1	1	2	2
Seri_2	1	1	1	1	2	2
Seri_3	1	2	2	1	1	1
Seri_4	2	3	3	1	1	3
Seri_5	2	3	3	1	1	3
Seri_6	2	3	3	1	1	3
Seri_14	2	1	1	1	1	3
Seri_15	1	1	4	2	3	4
Seri_16	2	3	4	1	1	3
Seri_17	1	3	3	1	1	3
Seri_18	1	3	3	1	1	3
Seri_25	1	2	2	1	1	4
Seri_26	1	2	2	1	1	4
Seri_27	1	2	4	1	1	4
Silhouette Katsayısı:	0.61	*0.84*	**0.89**	0.77	0.75	0.76
Dunn İndeksi:	0.45	*1.23*	**1.70**	0.70	0.88	0.88
Calinski-Harabasz İndeksi:	21.60	*619.83*	**643.62**	41.24	96.00	126.94

Temsil alan boyutu 3 iken aynı kümedeki satış verilerinin kaynak olarak kullanıldığı durumlarda negatif transfer öğrenme olup olmadığı incelenmiştir. Bu kapsamda, Ek 25'te verilen çok değişkenli satış verileri için tek kaynaklı derin transfer öğrenmedeki RMSE başarı yüzdeleri kullanılmıştır. Buna göre, Çizelge 4.29'da gösterildiği gibi temsil alan boyutunun 3 ve küme sayısının 4 olduğu kümeleme sonuçlarında negatif transfer öğrenme durumunun yaşanmayacağı tespit edilmiştir. Temsil alan boyutunun 3 ve küme sayısının 3 olduğu kümeleme sonuçlarında ise sadece 2 deneyde negatif transfer öğrenme gözlenmiştir. Deneysel çalışmalarda, küme sayısının 3 ve 4 olduğu durumlar ele alınmıştır. Küme sayısı 2 iken elde edilen kümeleme sonuçlarında ise daha fazla sayıda negatif transfer öğrenme gözlenmiştir.

Çizelge 4.29. Temsil alan boyutu 3, küme sayısı 3 ve 4 iken oluşan küme atamaları

Küme No/Küme Sayısı	Temsil Alan Boyutu: 3					
	k=3	Negatif TL	Pozitif TL	k=4	Negatif TL	Pozitif TL
1.Küme	Walmart-Seri_1	0	3	Walmart-Seri_1	0	2
	Walmart-Seri_2	0	3	Walmart-Seri_2	0	2
	Tv_Sales-Seri_14	0	3	Tv_Sales-Seri_14	0	2
	Rossmann-Seri_15	0	3			
2.Küme	LearnX-Seri_4	0	5	LearnX-Seri_4	0	4
	LearnX-Seri_5	0	5	LearnX-Seri_5	0	4
	LearnX-Seri_6	0	5	LearnX-Seri_6	0	4
	Favorita-Seri_16	1	4	Pharma-Seri_17	0	4
	Pharma-Seri_17	0	5	Pharma-Seri_18	0	4
	Pharma-Seri_18	1	5			
3.Küme	Walmart-Seri_3	0	3	Forklift-Seri_25	0	2
	Forklift-Seri_25	0	3	Forklift-Seri_26	0	2
	Forklift-Seri_26	0	3	Supermarket-Seri_27	0	2
	Supermarket-Seri_27	0	3			
4.Küme				Walmart-Seri_3	0	2
				Rossmann-Seri_15	0	2
				Favorita-Seri_16	0	2
Toplam Deney Sayısı		2	53	Toplam Deney Sayısı	0	38
TL Aktarım Yüzdesi		%0,04	%99,96	TL Aktarım Yüzdesi	%0	%100

Adım 3-4: Kümeleme ve topluluk tabanlı derin transfer öğrenme yaklaşımlarının deneysel sonuçları

Temsil alan boyutu 3, küme sayısının 3 ve 4 olduğu kümeler ele alınarak Bölüm 4.3.3'te yer alan yeni veri setleri, hedef veriler olarak kullanılmıştır. Serilerin benzemezlik matrisine göre en yakın komşuları (k=1, k=2 ve k=3) belirlenmiştir ve Çizelge 4.30'da verilmiştir.

Walmart-Seri_3, LearnX-Seri_6 ve Supermarket-Seri_27 hedef verileri için önerilen çözüm yaklaşımlarının uygulama detayları aşağıda açıklanmaktadır:

1. Topluluk tabanlı tek kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.27'ye göre, Walmart-Seri_3 için Supermarket-Seri_27, LearnX-Seri_6 için LearnX-Seri_5 ve Supermarket-Seri_27 için LearnX-Seri_6 kaynak veri olarak seçilmiştir.

Çizelge 4.30. Serilerin benzemezlik matrisine göre en yakın komşuları

Seri	1-NN	2-NN	3-NN
Walmart-Seri_1	Seri_2	Seri_2-3	Seri_2-3-27
Walmart-Seri_2	Seri_2	Seri_2-3	Seri_2-3-27
Walmart-Seri_3	Seri_27	Seri_27-25	Seri_27-25-26
LearnX-Seri_4	Seri_5	Seri_5-6	Seri_5-6-27
LearnX-Seri_5	Seri_6	Seri_6-4	Seri_6-4-27
LearnX-Seri_6	Seri_5	Seri_5-27	Seri_5-27-4
Tv_Sales-Seri_14	Seri_27	Seri_27-6	Seri_27-6-5
Rossmann-Seri_15	Seri_2	Seri_2-1	Seri_2-1-3
Favorita-Seri_16	Seri_14	Seri_14-27	Seri_14-27-6
Pharma-Seri_17	Seri_18	Seri_18-25	Seri_18-25-26
Pharma-Seri_18	Seri_25	Seri_25-26	Seri_25-26-17
Forklift-Seri_25	Seri_26	Seri_26-18	Seri_26-18-17
Forklift-Seri_26	Seri_25	Seri_25-18	Seri_25-18-17
Supermarket-Seri_27	Seri_6	Seri_6-5	Seri_6-5-4

2. KNN ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.30'a göre Walmart-Seri_3 hedef verisi için 2-NN'de Supermarket-Seri_27 ile Forklift-Seri_25, 3-NN'de ise Supermarket-Seri_27, Forklift-Seri_25 ve Forklift-Seri_26 kaynak veri olarak alınmıştır.

LearnX-Seri_6 hedef verisi için 2-NN'de LearnX-Seri_5 ile Supermarket-Seri_27, 3-NN'de ise LearnX-Seri_5, Supermarket-Seri_27 ve LearnX-Seri_4 kaynak veri olarak alınmıştır.

Supermarket-Seri_27 hedef verisi için 2-NN'de LearnX-Seri_6 ile LearnX-Seri_5, 3-NN'de ise LearnX-Seri_6, LearnX-Seri_5 ve LearnX-Seri_4 kaynak veri olarak alınmıştır.

3. Kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Çizelge 4.29'a göre Walmart-Seri_3 hedef verisi için küme sayısı 3 iken aynı kümede yer alan Forklift-Seri_25, Forklift-Seri_26 ve Supermarket-Seri_27 olmak üzere üç seri kaynak veri olarak kullanılmıştır. Küme sayısının 4 olduğu durumda ise Walmart-Seri_3, Rossmann-Seri_15 ve Favorita-Seri_16 olmak üzere farklı serilerle kümelenmiştir ve bu iki seri kaynak veri olarak kullanılmıştır.

LearnX-Seri_6 hedef verisi için küme sayısı 3 iken aynı kümede yer alan LearnX-Seri_4, LearnX-Seri_5, Favorita-Seri_16, Pharma-Seri_17 ve Pharma-Seri_18 olmak üzere beş seri kaynak veri olarak kullanılmıştır. Küme sayısının 4 olduğu durumda ise LearnX-Seri_6 hedef verisinin olduğu kümede artık Favorita-Seri_16 yer almamaktadır, bu nedenle bu seri dışında kalan diğer dört seri kaynak veri olarak kullanılmıştır.

Supermarket-Seri_27 hedef verisi için küme sayısı 3 iken aynı kümede yer alan Walmart-Seri_3, Forklift-Seri_25 ve Forklift-Seri_26 olmak üzere üç seri kaynak veri olarak kullanılmıştır. Küme sayısı 4 olduğu durumda ise Supermarket-Seri_27 hedef verisinin olduğu kümede Walmart-Seri_3 yer almamaktadır, bu nedenle bu seri dışında kalan diğer iki seri kaynak veri olarak kullanılmıştır.

4. Ağırlıklandırılmış kümeleme ve topluluk tabanlı çok kaynaklı derin transfer öğrenme yaklaşımı: Walmart-Seri_3, LearnX-Seri_6 ve Supermarket-Seri_27 hedef verileri için üçüncü yaklaşımda belirtilen kaynak veriler ağırlıklandırılarak kullanılmıştır.

Transfer öğrenmenin etkisini görebilmek amacıyla yukarıdaki yaklaşımlara ek olarak her üç hedef verisine transfer öğrenme yapılmadan LSTM yöntemi kullanılarak torbalama topluluk yöntemi uygulanmıştır.

Walmart-Seri_3 hedef verisi için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dakika) Çizelge 4.31'de verilmiştir. En başarılı sonuç yığınlama tabanlı tek kaynaklı derin transfer öğrenme ile elde edilmiştir. İkinci en başarılı sonuç torbalama tabanlı tek kaynaklı derin transfer öğrenme ile elde edilmiştir. Üçüncü en başarılı sonuç ise küme sayısı 4 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir.

Walmart-Seri_3 hedef verisinde tek kaynaklı transfer öğrenme, çok kaynaklı transfer öğrenmeye göre daha başarılı sonuçlar vermiştir. Çok kaynaklı durumlar içerisinde ise kümeleme ve ağırlıklandırılmış kümeleme tabanlı yöntemlerde küme sayısının 4 olduğu durumda küme sayısının 3 olduğu duruma göre daha başarılı sonuçlar elde edilmiştir. Burada, küme sayısı 3'ten 4'e çıkarıldığında Walmart-Seri_3'ün yer aldığı kümeler

değişmektedir ve sonuçlar iyileşmektedir. Bu durum, kümedeki kaynak verilere bağlı olarak sonucun iyileşebileceğini ve bu kaynak verilerin ağırlıklandırılması ile performansın arttırılabileceğini göstermektedir.

Öte yandan, Walmart-Seri_3 hedef verisinde en başarısız sonuç KNN-3 ve torbalama tabanlı çok kaynaklı derin transfer öğrenme ile kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile elde edilmiştir. KNN-3 ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme ile kümeleme ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ise en düşük ikinci performansa sahip yöntemlerdir. Bu hedef verisinde, KNN-3 tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları aynı sonuçları vermiştir. Burada, KNN-3 ile belirlenen kaynak veriler ile kümeleme ile elde edilen kümede yer alan veriler aynıdır.

Veri özellikleri açısından sonuçlar değerlendirildiğinde şu gözlemler yapılmıştır. Walmart-Seri_3 için en iyi sonuç tek kaynaklı transfer öğrenme yaklaşımlarında Supermarket-Seri_27 kaynak veri olarak kullanıldığında sağlanmıştır. Walmart ve Supermarket verileri Amerika'ya ait perakende satış verileridir ve her ikisi de mevsimsellik içermektedir. Bu ortak özellikler transfer öğrenmede başarı sağlamaktadır. Öte yandan, üçüncü en başarılı sonuç küme sayısı 4 iken Rossmann-Seri_15 ve Favorita-Seri_16'nın kaynak olarak kullanıldığı ağırlıklandırılmış kümeleme ve torbalama tabanlı yöntem ile elde edilmiştir. Rossmann, Favorita ve Walmart verileri perakende satış verileridir ve mevsimsellik içermektedirler. En başarısız sonuçlar ise Supermarket-Seri_27, Forklift-Seri_25 ve Forklift-Seri_26'nın kaynak olarak kullanıldığı çok kaynaklı yaklaşımlarda elde edilmiştir. Supermarket-Seri_27 tek başına iyileşme sağlarken seri 25 ile 26 kaynak olarak dahil edildiğinde başarı düşmüştür. Bu seriler trend içeren seyrek veriler olduğu için bu serilerden Walmart'a yapılan transfer öğrenmede başarı oranı daha düşüktür.

Çizelge 4.31. Çok değişkenli Walmart-Seri_3 hedef verisi için yöntemlerin performanslarının karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL											
			Tek Kaynaklı TL		Çok Kaynaklı TL									
					KNN Tabanlı				Kümeleme Tabanlı				Ağırlıklandırılmış Kümeleme Tabanlı	
					k=2		k=3		k=3		k=4		k=3	k=4
			Seri_27		Seri_27-25		Seri_27-25-26		Seri_27-25-26		Seri_15-16		Seri_27-25-26	Seri_15-16
		Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Torbalama
RMSE	2/3	21 091,91	19 565,78	19 557,39	19 703,15	19 701,88	19 832,56	19 834,37	19 832,56	19 834,37	19 811,27	19 813,08	19 790,25	19 721,30
MAE		18 173,93	16 858,93	16 851,70	16 977,30	16 976,20	17 088,80	17 090,37	16 737,79	16 739,32	16 719,82	16 721,35	16 702,08	16 643,89
Süre		5,77	13,33	13,42	12,05	12,14	16,45	16,55	16,45	16,55	117,30	117,39	16,48	117,34
RMSE	12	19 517,57	18 122,16	18 120,70	18 359,86	18 340,92	18 602,23	18 592,54	18 602,23	18 592,54	18 526,30	18 423,09	18 400,20	18 360,12
MAE		16 468,89	15 294,29	15 293,06	15 494,89	15 478,91	15 699,44	15 691,27	15 647,17	15 639,02	15 583,30	15 496,49	15 477,23	15 443,52
Süre		34,63	79,95	80,21	72,30	72,56	65,78	66,04	65,78	66,04	681,57	681,83	65,93	681,71
RMSE	24	18 751,57	17 427,11	17 397,77	17 912,26	17 872,31	18 018,59	17 947,03	18 018,59	17 947,03	18 002,41	17 779,29	17 876,52	17 818,25
MAE		15 772,79	14 658,72	14 634,05	15 066,80	15 033,20	15 156,24	15 096,05	15 044,76	14 985,01	15 031,25	14 844,96	14 926,14	14 877,49
Süre		69,26	159,90	160,63	144,60	145,32	131,57	132,30	131,57	132,30	1.363,13	1.363,86	131,74	1.363,30
RMSE	36	18 495,55	17 205,16	17 139,23	17 623,52	17 544,29	17 703,59	17 650,70	17 703,59	17 650,70	17 700,19	17 554,71	17 623,56	17 598,97
MAE		15 444,44	14 365,58	14 310,54	14 714,90	14 648,74	14 781,75	14 737,59	14 755,42	14 711,34	14 752,58	14 631,33	14 688,72	14 668,22
Süre		103,89	239,85	240,95	216,89	217,99	197,35	198,45	197,35	198,45	2.044,70	2.045,79	197,54	2.044,89
RMSE	48	18 368,52	17 107,68 **(2)**	17 030,76 **(1)**	17 450,59	17 446,90	17 556,74	17 443,00	17 556,74	17 443,00	17 542,84	17 404,04	17 398,36	17 357,65 **(3)**
MAE		15 306,76	14 258,75 **(2)**	14 194,64 **(1)**	14 544,55	14 541,47	14 633,02	14 538,22	14 569,03	14 474,65	14 557,50	14 442,32	14 437,61	14 403,82 **(3)**
Süre		138,52*****	319,80	321,26	289,19	290,65	263,14 **(1)**	264,59 **(3)**	263,14 **(1)**	264,59 **(3)**	2 726,27	2 727,72	263,35 **(2)**	2 726,48

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

Eğitim sürelerine göre karşılaştırma yapıldığında Walmart-Seri_3 hedef verisi için KNN-2 tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları (torbalama ve yığınlama tabanlı) en kısa iki süreye sahiptir. Üçüncü ve dördüncü en kısa eğitim süreleri ise tek kaynaklı derin transfer öğrenme yaklaşımları (torbalama ve yığınlama tabanlı) ile elde edilmiştir. Küme sayısının 3 olduğu durumda kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları en uzun eğitim sürelerine sahiptir.

LearnX-Seri_6 hedef verisi için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dakika) Çizelge 4.32'de verilmiştir. En iyi sonuç, küme sayısı 4 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. İkinci en iyi sonuç, küme sayısı 4 iken kümeleme ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Üçüncü en iyi sonuç ise küme sayısı 4 iken kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir.

LearnX-Seri_6 hedef verisinde en düşük iki performansa sahip yöntem KNN-2 tabanlı çok kaynaklı derin transfer öğrenme yaklaşımlarıdır (sırasıyla torbalama ve yığınlama). Bu durum, uzaklık tabanlı kaynak seçimi yaklaşımlarının diğer yaklaşımlara göre performans üstünlüğü sağlayamadığını göstermektedir. Öte yandan, kümeleme ve ağırlıklandırılmış kümeleme tabanlı çok kaynaklı derin transfer öğrenme yaklaşımlarının tek kaynaklı derin transfer öğrenme yaklaşımlarına göre daha iyi performansa sahip oldukları görülmektedir.

Eğitim süreleri incelendiğinde LearnX-Seri_6 hedef verisi için KNN-2 tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları (torbalama ve yığınlama) en kısa iki süreye sahiptir. Üçüncü ve dördüncü en kısa eğitim süreleri ise, KNN-3 tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ile elde edilmiştir. Küme sayısının 3 olduğu durumda ise kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları en uzun eğitim sürelerine sahiptir.

Çizelge 4.32. Çok değişkenli LearnX-Seri_6 hedef verisi için yöntemlerin performanslarının karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL																
			Tek Kaynaklı TL	Çok Kaynaklı TL															
				KNN Tabanlı					Kümeleme Tabanlı						Ağırlıklandırılmış Kümeleme Tabanlı				
				k=2		k=3		MS	k=3		MS	k=4		MS	k=3		MS	k=4	
			Seri_5	Seri_5-27		Seri_5-27-4			Seri_4-5-16-17-18			Seri_4-5-17-18			Seri_4-5-16-17-18			Seri_4-5-17-18	
		Torbalama	Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama	Yığınlama		Torbalama			Torbalama	
RMSE	2/3	85,39	69,99	72,95	73,02	70,75	70,65	5	68,28	68,35	4	67,41	65,36	5	65,62		4	63,03	
MAE		70,70	57,95	60,40	60,46	58,58	58,50		56,54	56,59		55,82	54,12		54,33			52,19	
Süre		18,30	35,27	28,39	28,48	47,81	47,91		480,28	480,43		389,29	389,42		480,32			389,33	
RMSE	12	74,60	61,30	62,55	63,44	61,84	62,21	15	60,14	59,10	12	59,13	58,97	15	57,17		12	55,93	
MAE		61,54	50,57	51,60	52,34	51,02	51,32		49,61	48,76		48,78	48,65		47,16			46,14	
Süre		109,77	211,63	170,35	170,61	191,25	191,51		1 440,83	1 441,28		1 167,88	1 168,14		1 440,98			1 168,03	
RMSE	24	67,74	55,98	57,23	55,63	56,22	54,70	25	53,84	53,06	24	53,85	53,49	25	52,10		24	51,26	
MAE		55,07	45,59	46,61	45,30	45,78	44,54		43,85	43,21		43,85	43,56		42,43			41,75	
Süre		219,55	423,25	340,71	341,43	382,49	383,22		2 401,38	2 402,14		2 335,77	2 336,50		2 401,55			2 335,94	
RMSE	36	60,17	50,10	52,45	50,20	51,16	49,75	35	49,56	48,14	36	46,52	45,75	35	47,15		36	44,96	
MAE		48,94	40,78	42,69	40,87	41,64	40,50		40,34	39,18		37,87	37,24		38,38			36,60	
Süre		329,32	634,88	511,06	512,15	573,74	574,83		3 361,93	3 362,99		3 503,65	3 504,74		3 362,12			3 503,84	
RMSE	48	56,37	47,77	50,51	48,79	48,28	46,51	50	46,01	44,00	48	42,77 **(3)**	40,90 **(2)**	50	43,27		48	40,02 **(1)**	
MAE		45,60	38,65	40,86	39,47	39,06	37,63		37,22	35,59		34,60 **(3)**	33,19 **(2)**		34,60			32,38 **(1)**	
Süre		439,10*****	846,51	681,41 **(1)**	682,87 **(2)**	764,98 **(3)**	766,44		4 802,76	4 804,28		4 671,54	4 672,99		4 802,97			4 671,74	

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

Veri özellikleri açısından sonuçlar değerlendirildiğinde; LearnX-Seri_6 için en iyi sonuç küme sayısı 4 iken LearnX-Seri_4, LearnX-Seri_5, Pharma-Seri_17 ve Pharma-Seri_18'in kaynak veri olarak kullanıldığı ağırlıklandırılmış kümeleme ve torbalama tabanlı yöntemde elde edilmiştir. Hem hedef veri hem de kaynak olarak kullanılan veriler mevsimsellik ile tek bir ürüne ait satışları içermektedir. İkinci ve üçüncü en iyi sonuçlar küme sayısı 4 iken kümeleme tabanlı çok kaynaklı yaklaşımlarda, aynı kaynak verilerle elde edilmiştir. En başarısız sonuçlar ise KNN-2 tabanlı çok kaynaklı transfer öğrenme yaklaşımlarında LearnX-Seri_5 ve Supermarket-Seri_27'nin kaynak veri olarak kullanılması sonucu elde edilmiştir. Sadece LearnX-Seri_5'in kaynak olarak kullanıldığı tek kaynaklı transfer öğrenme yaklaşımı ile kıyaslandığında Supermarket-Seri_27'nin ikinci bir kaynak olarak kullanılması transfer öğrenmedeki başarıyı düşürmüştür. Supermarket-Seri_27 çok sayıdaki ürüne ait toplam satışları içeren perakende satışları içermektedir.

Supermarket-Seri_27 hedef verisi için dört yaklaşıma ait tahmin hataları (RMSE ve MAE) ile eğitim süreleri (dakika) Çizelge 4.33'te verilmiştir. En başarılı tahmin sonucu küme sayısı 3 iken ağırlıklandırılmış kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ile elde edilmiştir. Küme sayısı 3 iken kümeleme ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme yöntemi en iyi ikinci performansa sahiptir. Küme sayısı 4 iken kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ise en iyi üçüncü performansa sahiptir.

Supermarket-Seri_27 hedef verisinde en başarısız tahmin sonucunu veren yaklaşım küme sayısı 4 iken kümeleme ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yöntemidir. Küme sayısı 4 iken kümeleme ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme yöntemi ise ikinci en başarısız performansa sahiptir. Bu hedef verisinde küme sayısı 3 iken kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları ile en başarılı tahmin sonuçları elde edilirken küme sayısı 4 olduğu durumda ise en başarısız sonuçlar elde edilmiştir. Küme sayısı 3'ten 4'e çıkarıldığında bu iki küme arasındaki fark, Walmart-Seri_3 verisinin kaynak veri olarak kullanılmamasıdır. Bu durum, kümelemenin ve küme sayısının önemini göstermektedir. Ayrıca, performans açısından çok kaynaklı derin transfer öğrenme yaklaşımlarının tek kaynaklı derin transfer öğrenme yaklaşımlarından öne çıktığı görülmektedir.

Çizelge 4.33. Çok değişkenli Supermarket-Seri_27 hedef verisi için yöntemlerin performanslarının karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL											
			Tek Kaynaklı TL		Çok Kaynaklı TL									
					KNN Tabanlı				Kümeleme Tabanlı				Ağırlıklandırılmış Kümeleme Tabanlı	
					k=2		k=3		k=3		k=4		k=3	k=4
			Seri_6		Seri_6-5		Seri_6-5-4		Seri_3-25-26		Seri_25-26		Seri_3-25-26	Seri_25-26
		Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Torbalama
RMSE	2/3	192,93	172,26	172,82	171,56	171,97	171,12	171,27	163,13	163,21	173,88	175,50	156,85	170,20
MAE		149,27	133,27	133,70	132,73	133,05	132,39	132,51	126,21	126,27	134,53	135,78	121,35	131,68
Süre		9,62	24,37	24,46	26,91	27,00	43,42	43,51	18,07	18,17	12,15	12,24	18,10	12,18
RMSE	12	167,71	150,28	148,70	149,98	149,90	148,18	147,38	144,52	143,89	151,99	153,17	135,62	149,52
MAE		124,47	111,63	110,46	111,41	111,35	110,07	109,48	107,35	106,88	112,90	113,78	100,74	111,07
Süre		57,70	146,21	146,47	161,45	161,71	173,66	173,92	72,28	72,54	72,88	73,14	72,42	73,03
RMSE	24	149,75	135,52	133,60	134,26	132,55	133,44	131,34	131,13	129,64	136,74	136,30	123,95	134,59
MAE		107,29	97,27	95,89	96,37	95,14	95,78	94,27	94,12	93,05	98,15	97,83	88,97	96,61
Süre		115,40	292,42	293,15	322,89	323,62	347,32	348,05	144,56	145,28	145,77	146,50	144,73	145,94
RMSE	36	137,79	125,26	122,74	123,84	122,24	122,67	121,31	120,77	120,16	126,16	124,33	115,86	124,68
MAE		97,27	88,43	86,66	87,43	86,30	86,60	85,65	85,26	84,83	89,07	87,77	81,79	88,02
Süre		173,09	438,63	439,72	484,34	485,43	520,98	522,07	216,83	217,93	218,65	219,74	217,02	218,84
RMSE	48	133,90	121,95	119,45	119,97	118,78	118,19	116,91	114,88 **(3)**	112,33 **(2)**	123,23	122,71	110,21 **(1)**	121,32
MAE		94,08	85,70	83,94	84,31	83,47	83,06	82,16	80,73 **(3)**	78,94 **(2)**	86,54	86,18	77,45 **(1)**	85,20
Süre		230,79*****	584,84	586,30	645,79	647,24	694,64	696,10	289,11 **(1)**	290,57 **(3)**	291,54	292,99	289,32 **(2)**	291,75

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

Eğitim süreleri incelendiğinde Supermarket-Seri_27 hedef verisi için küme sayısı 3 iken kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları en kısa sürelere sahiptir. Küçük bir farkla küme sayısı 4 iken kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları da kısa eğitim sürelerine sahiptir. KNN-3 tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları ise en uzun eğitim sürelerine sahiptir.

Veri özellikleri açısından sonuçlar değerlendirildiğinde; Supermarket-Seri_27 için en iyi sonuç küme sayısı 3 iken Walmart-Seri_3, Forklift-Seri_25 ve Forklift-Seri_26'nın kaynak veri olarak kullanıldığı ağırlıklandırılmış kümeleme ve torbalama tabanlı yöntemde elde edilmiştir. Walmart ve Supermarket verileri Amerika'ya ait perakende satış verileridir, her iki veri de mevsimsellik içermektedir. Forklift-Seri_25 ve Forklift-Seri_26 ise tek bir ürüne ait satışları içeren, trende sahip seyrek verilerdir. Ağırlıklandırma stratejisi ile Walmart-Seri_3'ün katkısının Forklift-Seri_25 ve Forklift-Seri_26 verilerine göre arttırılması ile başarının arttırıldığı görülmektedir. Bu durum, ağırlıklandırma stratejisinin etkinliğini göstermektedir. Diğer yandan, küme sayısı 3 ile 4 arasındaki tek fark Walmart-Seri_3'ün kaynak olarak kullanılmasıdır. Küme sayısı 4 iken en başarısız sonuçlar elde edilmiştir.

Transfer öğrenmenin etkisi açısından sonuçlar incelendiğinde her üç hedef verisi için transfer öğrenmenin uygulandığı tüm yaklaşımlarda transfer öğrenmenin yapılmadığı torbalama tabanlı LSTM yöntemine göre iyileşme elde edilmiştir. Transfer öğrenme ile LSTM yöntemine kıyasla RMSE ve MAE performans metriklerine göre Walmart-Seri_3 hedef verisinde sırasıyla %7,28 ve %7,27; LearnX-Seri_6 hedef verisinde sırasıyla %29,00 ve %29,00 ve Supermarket-Seri_27 hedef verisinde sırasıyla %17,69 ve %17,68 oranlarında iyileşme sağlanmıştır. Öte yandan, transfer öğrenmenin uygulandığı yaklaşımlarda daha uzun eğitim süreleri ihtiyaç duyulabilmektedir.

Özetle, çok değişkenli satış verilerinde tek değişkenli satış verilerine benzer şekilde kümeleme ve ağırlıklandırılmış kümeleme tabanlı çok kaynaklı derin transfer öğrenme yaklaşımları birçok deneyde performans üstünlüğü sağlamıştır. Sadece Walmart-Seri_3 hedef verisinde tek kaynaklı transfer öğrenme, çok kaynaklı transfer öğrenme yaklaşımlarına göre daha başarılı sonuçlar vermiştir. Diğer hedef verilerinde, çok

kaynaklı derin transfer öğrenme yaklaşımları tek kaynaklı derin transfer öğrenme yaklaşımlarına göre daha başarılı sonuçlar vermiştir.

Çok kaynaklı derin transfer öğrenme yaklaşımlarından ise ağırlıklandırılmış kümeleme tabanlı yaklaşımın diğer yaklaşımlara göre daha üstün performansa sahip olduğu gözlenmiştir. Kümeleme tabanlı yaklaşım ile başarılı sonuçlar elde edilmiştir. Bu yaklaşımlara göre, KNN tabanlı yaklaşımların performansı geride kalmıştır. Ek olarak, kümeleme tabanlı ve KNN tabanlı yaklaşımlarda yığınlama topluluk yönteminin torbalama topluluk yöntemine göre daha başarılı sonuçlar verdiği gözlenmiştir. Diğer taraftan, ağırlıklandırılmış kümeleme ve torbalama tabanlı yaklaşımın ise çoğunlukla kümeleme ve yığınlama tabanlı yaklaşımdan daha iyi performansa sahip olduğu görülmüştür. Bu durum geliştirilen ağırlıklandırma mekanizmasının çok değişkenli satış verileri için başarılı sonuçlar üretebildiğini göstermektedir.

4.5. Tek ve Çok Değişkenli Satış Verilerinin Deneysel Sonuçlarının Karşılaştırılması

Bu bölümde, Walmart-Seri_3, LearnX-Seri_6 ve Supermarket-Seri_27 satış verilerinin tek değişkenli ve çok değişkenli zaman serileri olarak kullanıldığı durumlar karşılaştırılmıştır. Bu satış verilerinde tek ve çok kaynaklı topluluk yaklaşımları ile elde edilen sonuçların en iyi (minimum – min), ortalama (ort.) ve en kötü (maksimum – max) performans ölçüt değerleri Çizelge 4.34'te sunulmaktadır. Bu çizelgedeki tek değişkenli satış verilerine ait değerler Çizelge 24-26 ve çok değişkenli satış verilerine ait değerler ise Çizelge 31-33'te yer alan 48 veya 50 model içeren tüm topluluk tabanlı derin transfer öğrenme yaklaşımlarının özetlenmesi ile elde edilmiştir. Örneğin, Çizelge 24'te verilen tek değişkenli Walmart-Seri_3 için RMSE ve MAE ölçütlerine göre en iyi sonuçlar sırasıyla 17 096,45 ve 14 274,81 değerleridir. Bu sonuçlar, toplamda 48 model içeren 2-NN ve yığınlama tabanlı çok kaynaklı derin transfer öğrenme yaklaşımıyla elde edilmiştir. RMSE ve MAE ölçütlerine göre en kötü sonuçlar sırasıyla 18 003,45 ve 15 032,12'dir. Bu sonuç toplamda 48 model içeren 3-NN ve torbalama tabanlı çok kaynaklı derin transfer öğrenme yaklaşımıyla elde edilmiştir. Bu hedef verisinde toplamda 48 veya 50 model içeren tüm yaklaşımlarda elde edilen RMSE ve MAE ölçütlerinin ortalaması sırasıyla 17 479,16 ve 14 583,48 olarak hesaplanmıştır.

Çizelge 4.34'e göre, RMSE ve MAE performans ölçütlerine göre Supermarket-Seri_27 verisi dışında diğer iki veride (Walmat-Seri_3 ve LearnX-Seri_6) çok değişkenli satış verileri ile yapılan deneylerde tek değişkenli versiyona göre iyileşme sağlanmıştır. Supermarket-Seri_27 verisinde ise tek değişkenli zaman serileri olarak yapılan deneylerde daha iyi sonuçlar elde edilmiştir. Hafta, ay ve yıl bilgisi olmak üzere zaman bilgisi ile tatil olup olmadığına dair bilginin öznitelik olarak eklenmesi ile tahmin performansları iyileşmiştir.

Çizelge 4.34. Tek ve çok değişkenli satış verileri için karşılaştırma

Hedef Veri	Ölçüt	Tek Değişkenli			Çok Değişkenli		
		Min	Ort.	Max	Min	Ort.	Max
Walmart-Seri_3	RMSE	17 096,45	17 479,16	18 003,45	**17 030,76**	17 394,86	17 556,74
	MAE	14 274,81	14 583,48	15 032,12	**14 194,64**	14 460,97	14 569,03
	Süre	**189,87**	834,99	1 443,31	263,14	893,35	2 727,72
LearnX-Seri_6	RMSE	41,62	47,59	57,66	**40,02**	45,40	50,51
	MAE	33,67	38,37	46,65	**32,38**	36,70	40,86
	Süre	**696,45**	2 598,37	4 736,66	764,98	2 751,37	4 804,28
Supermarket-Seri_27	RMSE	**107,18**	115,06	120,85	110,21	118,33	122,71
	MAE	**74,70**	80,34	84,33	77,45	83,14	86,18
	Süre	**200,51**	2 789,16	5 714,15	289,11	466,68	696,10

Tek değişkenli satış verileri için geliştirilen metodoloji, çok değişkenli satış verileri ile geliştirilen metodolojiye göre daha kapsamlıdır. Şöyle ki tek değişkenli satış verileri için öznitelik çıkarımı ve öznitelik seçimine dayalı bir benzemezlik matrisi oluşturulmaktadır. Bu benzemezlik matrisi kullanılarak derin kümeleme yapılmıştır. Elde edilen kümelere göre kümeleme ve ağırlıklandırılmış kümeleme yaklaşımları uygulanmıştır. Bu yaklaşımlarda zaman serilerine ait öznitelikler, satışla ilgili öznitelikler ve uzaklık tabanlı öznitelikler kullanılmıştır. Bu özniteliklerden gelen bilgilerin kullanılması ile ortak özelliklere sahip satış verileri aynı kümeye atanmışlardır. Bu sayede tek değişkenli satış verilerindeki kaynak seçimi çok değişkenli satış verilerindeki seçime göre yaklaşımlarda daha yüksek başarı sağlamıştır. Bu duruma iki satış verisi ve farklı öznitelikler üzerinden örnek verilebilir. Örneğin, Walmart-Seri_3 hedef verisi için tek değişkenli satış verileri ile yapılan deneysel çalışmalarda küme sayısı 4 iken Walmart-Seri_1, Walmart-Seri_2, Tv_Sales-Seri_14, Rossmann-Seri_15 ve Favorita-Seri_16 verileri ile kümelenmiştir. Küme sayısı 3 iken ise Rossmann-Seri_15 haricindekilerle kümelenmiştir. Bu kümelerde Walmart serileri aynı kümeye atanmışlardır. Ayrıca aynı kümedeki serilerden Tv_Sales-Seri_14 dışındaki tüm seriler perakende satış verileridir ve mevsimsellik içermektedirler.

Satış verileri arasındaki bu ortak özellikler özniteliklerden gelen bilgiler sayesinde yakalanmıştır. Şöyle ki, satış ile ilgili özniteliklerden biri olan sektör bilgisi (Sector_Info) ile aynı sektöre ait satış verileri ve zaman serisi özniteliklerinden mevsimsellik (Seasonal_Strength) özniteliği ile mevsimsellik içeren satış verileri aynı kümelere atanmıştır. Bu sayede tek değişkenli zaman serilerinde kümeleme ve ağırlıklandırılmış kümeleme tabanlı derin transfer öğrenme yaklaşımları daha başarılı sonuçlar vermiştir. Bir diğer örnekte ise LearnX-Seri_6 hedef verisi verilebilir. LearnX-Seri_6 hedef verisi genellikle LearnX ve Pharma satış verileri ile kümelenmektedir. LearnX ve Pharma verileri daha önce de bahsedildiği gibi tek bir ürüne ait satışları içeren serilerden oluşmaktadır ve mevsimselliğe sahiptirler. Mevsimsellik özniteliğinin yanı sıra satışa ait özniteliklerden biri olan ürün grubu (ProductGroup) özniteliği bu kümelenmede etkili olmuştur. Bunun nedeni, tek değişkenli zaman serilerinin kümelenmesinde LearnX ve Pharma verileri gibi tek ürün grubu satışlarının olduğu verilerin genellikle birlikte kümelenmesidir. Bu iki örnekte görüldüğü üzere özniteliklerden gelen bilgilerin benzemezlik matrisi yoluyla derin kümelemede girdi olarak kullanılması transfer öğrenmedeki bilgi aktarım başarısını arttırmıştır.

Çok değişkenli satış verilerinde sadece veriler arasındaki şekilsel uzaklıklar dikkate alınarak benzemezlik matrisi oluşturulmuştur ve bu matrise derin kümeleme uygulanmıştır. Tek değişkenli satış verilerinde olduğu gibi özniteliklerden gelen bilginin kullanılmamasına rağmen iki hedef veride (Walmat-Seri_3 ve LearnX-Seri_6) tek değişkenli duruma göre iyileşme sağlanmıştır. Bu iyileşme, satış verilerine eklenen zaman ve tatil bilgilerine ait öznitelikler ile ilişkilendirilebilir. Diğer yandan, Supermarket-Seri_27 hedef verisinde ise tek değişkenli yaklaşımın öne çıkması özniteliklerden gelen bilginin önemini işaret etmektedir. Şöyle ki, küme sayısı 3 iken Supermarket-Seri_27 hedef verisi Walmart-Seri_3, Forklift-Seri_25 ve Forklift-Seri_26 verileriyle, küme sayısı 4 iken ise Forklift-Seri_25 ve Forklift-Seri_26 verileriyle kümelenmiştir. Küme sayısı 3'ten 4'e çıkarıldığında bu hedef verisi için Walmart-Seri_3 serisi kaynak olarak kullanılmamaktadır. Ancak Walmart-Seri_3 ile Supermarket-Seri_27 verileri Amerika'ya ait perakende satış verileridir, çoklu ürün grubu satışlarına sahiptirler ve mevsimsellik içermektedirler. Sadece şekilsel uzaklıklar ele alındığı için bu ortak özellikler geri planda kalmıştır. Bu nedenle, küme sayısı 4 iken şekilsel uzaklık

açısından daha yakın olan Forklift-Seri_25 ve Forklift-Seri_26 verileri ile kümelenmiştir ve küme sayısının 3 olduğu duruma göre tahmin performansında düşüş yaşanmıştır.

Çok değişkenli versiyonda Walmart-Seri_3 hedef verisi küme sayısı 4 iken Supermarket-Seri_27, Forklift-Seri_25 ve Forklift-Seri_26 verileriyle, küme sayısı 3 iken ise Rossmann-Seri_15 ve Favorita-Seri_16 verileriyle kümelenmiştir. Tek değişkenli versiyonda Walmart-Seri_3 hedef verisi küme sayısı 3 iken Walmart serileri veya Supermarket-Seri_27 dışında başka perakende satışlarını içeren seriler ile birlikte kümelenmemiştir. Küme sayısı 4 iken ise iki farklı perakende satış verisi ile kümelenmiştir. Bu nedenle çok değişkenli kümeleme ve ağırlıklandırılmış kümeleme tabanlı yaklaşımlarda tahmin performansı tek değişkenli versiyona göre daha düşüktür. Bu performans düşüklüğü özniteliklerin kullanılmamasından kaynaklanmaktadır. Öte yandan, çok değişkenli versiyonda Walmart-Seri_3 hedef verisi için Supermarket-Seri_27'nin kaynak olarak kullanıldığı tek kaynaklı yaklaşımda ise tek değişkenli versiyonda elde edilen en yüksek tahmin performansından daha iyi performans elde edilmiştir. Bu performans artışında ise satış verilerinin ek bilgilerle çok değişkenli hale getirilmesi rol oynamaktadır.

Tek değişkenli ve çok değişkenli satış verileri için geliştirilen metodolojiler arasında en önemli farklılık negatif transfer öğrenmenin önlenebilme durumudur. Tek değişkenli versiyonda geliştirilen metodoloji ile elde edilen tüm kümelerde kaynak-hedef eşleşmelerinde negatif transfer öğrenme durumu tamamen önlenmiştir. Ancak, çok değişkenli versiyonda Bölüm 4.4'te verilen Çizelge 4.29'da görüldüğü üzere küme sayısı 3 iken elde edilen ikinci kümede negatif transfer öğrenme yaşanmaktadır. Deneysel çalışmalarda negatif transfer öğrenmenin yaşandığı veri kümeleri (Favorita-Seri_16 ile Pharma_Seri_18) hedef veri olarak kullanılmadığı için bu durumun negatif etkisi görülmemiştir. Eğer bu veriler üzerinde deneysel çalışmalar gerçekleştirilseydi transfer öğrenme başarısının düştüğü gözlemlenebilirdi. Bu durum, transfer öğrenme başarısı için negatif transfer öğrenmenin önlenmesinin önemini ve gerekliliğini göstermektedir.

Çizelge 4.50'deki özet sonuçlar eğitim süreleri açısından incelendiğinde tüm çözüm yaklaşımlarında çok değişkenli zaman serileri ile yapılan deneysel çalışmalar tek değişkenli zaman serileri ile yapılan deneysel çalışmalara göre daha uzun eğitim sürelerine sahiptir. Eğitim süreleri açısından oluşan bu fark temelde LSTM yönteminden kaynaklanmaktadır. Tek değişkenli versiyonda tek değişkenli LSTM modeli kullanılırken çok değişkenli versiyonda çok değişkenli LSTM modeli kullanılmaktadır. Çok değişkenli LSTM modelinde tek değişkenli LSTM modeline göre girdi boyutu artmaktadır. Bu durum LSTM eğitimindeki parametre sayısını arttırmaktadır ve buna bağlı eğitim süreleri de artmaktadır. Öte yandan, her veri kümesinde hem tek hem de çok değişkenli zaman serileri için LSTM modeline hiperparametre optimizasyonu yapılmıştır. Bu nedenle, oluşan ağ mimarileri farklı olabilmektedir ve bu durum eğitim sürelerine yansımaktadır. Ek olarak, tek ve çok değişkenli durumda kaynak olarak kullanılan verilere bağlı olarak da eğitim süreleri değişmektedir. Kaynak olarak kullanılan satış verilerinin uzunluğu (boyutu) arttıkça eğitim süreleri de artmaktadır.

4.6. Tartışma

Tek değişkenli satış verilerinde, transfer öğrenmedeki başarı durumunu açıklayabilmek için zaman serisi, satış ve uzaklık tabanlı öznitelikler belirlenmiştir ve satış verileri daha fazla bilgi içerecek şekilde kullanılmıştır. Seçilen öznitelikler ile kümelemenin yapılması benzer özelliklere sahip verilerin aynı kümeye atanmasını sağlamıştır. Bu sayede, hem transfer öğrenmedeki tahmin performansı artmıştır, hem de kaynak seçiminde açıklanabilirlik sağlanmıştır. Ayrıca, bu yaklaşımla negatif transfer öğrenmeye neden olan kaynak-hedef eşleşmeleri önlenmiştir. Yeni satış verileri ile geliştirilen metodolojinin etkinliği gösterilmiştir. Geliştirilen metodoloji kaynak veri seçiminde açıklanabilirliği sağlamaktadır.

Çok değişkenli satış verilerinde ise transfer öğrenmede başarı durumu satış verileri arasındaki şekilsel uzaklıklar ile ilişkilendirilmiştir. Önerilen metodoloji, sadece şekilsel olarak benzer satış verileri arasında transfer öğrenmeyi esas aldığı için bazı veri kümelerinde etkili performans göstermiştir. Ayrıca, satış verilerine zaman ve tatil bilgilerinin eklenmesi ile LSTM yönteminde elde edilen başarı artmıştır.

Özetle, tek değişkenli satış verileri için geliştirilen metodoloji aşağıdaki durumlarda tercih edilebilir:

- Farklı sektörlere ait farklı ürün/hizmet satış verileri mevcut ise,
- Satış verisi miktarı yeterli ise,
- Satış verilerine ait ek bilgiler (tatil, hava durumu, vb.) bulunmuyor ise kullanılabilir.

Çok değişkenli satış verileri için geliştirilen metodoloji ise aşağıdaki durumlarda tercih edilebilir:

- Aynı sektöre veya aynı ürün/hizmete ait satış verileri mevcut ise,
- Satış verisi miktarı az ise,
- Satış verilerine ait ek bilgiler (tatil, hava durumu, vb.) bulunuyor ise kullanılabilir.

Bu tezde, geliştirilen metodolojiler üretim ve hizmet sektörlerinde farklı ürün/hizmet satış tahminlerinde kullanılabilir. Ayrıca, önerilen metodolojiler yeni ürünlerin satış tahmininde ve yetersiz veri içeren durumlarda da katkı sağlayabilir.

5. SONUÇ

Tedarik zincirinin farklı aşamalarında yer alan üretici, perakendeci ve distribütör firmalar satış tahmini yapmaktadır. Tedarik zinciri ağı boyunca tecrübenin ve bilginin aktarımı tahmin kaynaklı zaman ve maliyet kayıplarını önleyebilir. Bu doğrultuda, bu tez çalışmasında satış ve talep tahmini için bilgi aktarımını sağlayan transfer öğrenme ele alınmıştır.

Bu tezde, hem tek değişkenli hem de çok değişkenli satış verileri için derin transfer öğrenmeye dayalı çözüm metodolojileri geliştirilmiştir. Geliştirilen metodolojilerde farklı uzunluklarda satış ve talep verileri kullanılmıştır. Satış ve talep tahmini için LSTM yöntemi en uygun derin öğrenme yöntemi olarak seçilmiştir. Derin transfer öğrenme başarısının artması için benzerlik analizi, derin kümeleme ve topluluk yöntemleri kullanılmıştır. Ayrıca, derin kümelemede küme sayısının bilinmediği durumlar ele alınmıştır. Özellikle, tek değişkenli satış verileri için transfer öğrenmede kaynak seçimi, bilginin aktarılabilirliği ve açıklanabilirliğini sağlayan bütünleşik bir çözüm yaklaşımı sunulmuştur. Çok değişkenli satış verileri için ise şekilsel uzaklık tabanlı derin transfer öğrenme çözüm yaklaşımları önerilmiştir. Geliştirilen ağırlıklandırma mekanizmaları ile topluluktaki modellere eğitim başarılarına göre farklı önem dereceleri verilmiştir.

Yapılan deneysel çalışmalar, derin kümeleme ve topluluk yöntemleri ile birlikte ağırlıklandırma mekanizmasının transfer öğrenme başarısını arttırdığını göstermektedir. Ayrıca, çok kaynaklı derin transfer öğrenme yaklaşımlarının tek kaynaklı derin transfer öğrenme yaklaşımlarına göre tahmin sonuçlarını iyileştirdiği gözlenmiştir. Diğer taraftan, çok kaynaklı derin transfer öğrenmede KNN tabanlı yaklaşımların daha düşük performansa sahip olduğu tespit edilmiştir. Tek değişkenli satış verilerinde, kaynak seçimi için açıklanabilirliği sağlayan özniteliklerin çıkarılması ve ardından kümeleme ile benzer satış verilerinin belirlenmesi transfer öğrenmede tahmin performansını arttırmıştır. Çok değişkenli satış verilerinde ise, kaynak seçimi için sadece şekilsel uzaklık tabanlı yaklaşımın kullanımı transfer öğrenmede elde edilen faydayı sınırlamıştır. Bu durum, açıklanabilirlik ve öznitelik mühendisliğinin önemine işaret etmektedir.

Gelecek çalışmalarda, transfer öğrenme kararlarında ağ benzerliği ve veri dağılımlarına yönelik çalışmalar yapılabilir. Ayrıca, akıllı kaynak seçimi için farklı topluluk stratejileri kullanılabilir. Geliştirilen metodolojiler enerji, finans, çevre ve tedarik zinciri (müşteri kaybı) alanlarında farklı zaman serisi tahmin problemlerinde uygulanabilir. Diğer gelecek çalışma konuları arasında, çok değişkenli zaman serilerinde açıklanabilirlik konusu ile ilave değişkenlerin (ürünün bozulabilirlik durumu, hava durumu ve ekonomik göstergeler gibi çevresel faktörler) ele alınması yer alabilir. İlave olarak, önerilen metodolojilerin pratikte etkin kullanımını sağlamak amacıyla bir karar destek sistemi ve ara yüz geliştirilebilir.

KAYNAKLAR

Abanda, A., Mori U., Lozano J. A. (2019). A review on distance based time series classification, *Data Mining and Knowledge Discovery*, *33*(2), 378-412. https://doi.org/10.1007/s10618-018-0596-4

Acı, M., Doğansoy, G. A. (2022). Demand forecasting for e-retail sector using machine learning and deep learning methods. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *37*(3), 1325-1339. https://doi.org/10.17341/gazimmfd.944081

Aggarwal, C. C., Reddy, C. K. (2014). Data clustering. Algorithms and applications. *Chapman&Hall CRC Data mining and Knowledge Discovery series*, Londra.

Ali, S., Abuhmed, T., El-Sappagh, S., Muhammad, K., Alonso-Moral, J. M., Confalonieri, R., ... & Herrera, F. (2023). Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information fusion*, *99*, 101805. https://doi.org/10.1016/j.inffus.2023.101805

Alqahtani, A., Ali, M., Xie, X., & Jones, M. W. (2021). Deep time-series clustering: A review. *Electronics*, *10*(23), 3001. https://doi.org/10.3390/electronics10233001

Ansari, Z., Azeem, M. F., Ahmed, W., & Babu, A. V. (2015). Quantitative evaluation of performance and validity indices for clustering the web navigational sessions. *arXiv preprint arXiv:1507.03340*. https://doi.org/10.48550/arXiv.1507.03340

Arora, T., Chandna, R., Conant, S., Sadler, B., & Slater, R. (2020). Demand forecasting in wholesale alcohol distribution: an ensemble approach. *SMU Data Science Review*, *3*(1), 7.

Bai, Y., Xie, J., Wang, D., Zhang, W., & Li, C. (2021). A manufacturing quality prediction model based on AdaBoost-LSTM with rough knowledge. *Computers & Industrial Engineering*, *155*, 107227. https://doi.org/10.1016/j.cie.2021.107227

Bandara, K., Bergmeir, C., & Smyl, S. (2020). Forecasting across time series databases using recurrent neural networks on groups of similar series: A clustering approach. *Expert systems with applications*, *140*, 112896. https://doi.org/10.1016/j.patcog.2021.108148

Bandara, K., Shi, P., Bergmeir, C., Hewamalage, H., Tran, Q., & Seaman, B. (2019). Sales demand forecast in e-commerce using a long short-term memory neural network methodology. In *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part III 26* (pp. 462-474). Springer International Publishing. https://doi.org/10.1007/978-3-030-36718-3_39

Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., ... & Gamboa, H. (2020). TSFEL: Time series feature extraction library. *SoftwareX*, *11*, 100456. https://doi.org/10.1016/j.softx.2020.100456

Bengio, Y. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning*, *2*(1), 1-127. http://dx.doi.org/10.1561/2200000006

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, *35*(8), 1798-1828. https://doi.org/10.1109/TPAMI.2013.50

Bengio Y., Simard P., Frasconi P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166, 1994. https://doi.org/10.1109/72.279181

Bonifati, A., Del Buono, F., Guerra, F., & Tiano, D. (2022). Time2Feat: learning interpretable representations for multivariate time series clustering. *Proceedings of the VLDB Endowment (PVLDB)*, *16*(2), 193-201. https://dx.doi.org/10.14778/3565816.3565822

Bramer, M. (2007). *Principles of data mining*. Springer.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32. https://doi.org/10.1023/A:1010933404324

Brownlee, J. (2017). A gentle introduction to transfer learning for deep learning. *Machine Learning Mastery*, *20*. https://machinelearningmastery.com/transfer-learning-for-deep-learning/ (Erişim tarihi: 20.12.2020).

Busari, G. A., Kwak, N. W., & Lim, D. H. (2021). An application of AdaBoost-GRU ensemble model to economic time series prediction. *Indian Journal of Science and Technology*, *14*(31), 2557-2566. https://doi.org/10.17485/IJST/v14i31.1204

Busari, G. A., & Lim, D. H. (2021). Crude oil price prediction: A comparison between AdaBoost-LSTM and AdaBoost-GRU for improving forecasting performance. *Computers & Chemical Engineering*, *155*, 107513. https://doi.org/10.1016/j.compchemeng.2021.107513

Calinski, T., & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*(1), 1-27. https://doi.org/10.1080/03610927408827101

Cameron, A. C., & Windmeijer, F. A. (1996). R-squared measures for count data regression models with applications to health-care utilization. *Journal of Business & Economic Statistics*, *14*(2), 209-220. https://doi.org/10.1080/07350015.1996.10524648

Canchumuni S. W., Emerick A. A., & Pacheco M. A. C. (2019) Towards a robust parameterization for conditioning facies models using deep variational autoencoders and ensemble smoother. *Computers & Geosciences*, *128*, 87-102. https://doi.org/10.1016/j.cageo.2019.04.006

Chen, L., & Ng, R. (2004, August). On the marriage of lp-norms and edit distance. In *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30* (pp. 792-803).

Chen, T., Yin, H., Chen, H., Wu, L., Wang, H., Zhou, X., & Li, X. (2018, November). Tada: trend alignment with dual-attention multi-task recurrent neural networks for sales prediction. In *2018 IEEE international conference on data mining (ICDM)* (pp. 49-58). IEEE. https://doi.org/10.1109/ICDM.2018.00020

Chen, Y., Kang, Y., Chen, Y., & Wang, Z. (2020). Probabilistic forecasting with temporal convolutional neural network. *Neurocomputing*, *399*, 491-501. https://doi.org/10.1016/j.neucom.2020.03.011

Christ, M., Braun, N., Neuffer, J., & Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, *307*, 72-77. https://doi.org/10.1016/j.neucom.2018.03.067

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*. https://doi.org/10.48550/arXiv.1406.1078

Chollet, F. (2021). Deep learning with Python. *Simon and Schuster, Manning Publication*, USA. https://livebook.manning.com/book/deep-learning-with-python-second-edition/chapter-1/

Chopra, S., Meindl, P., & Kalra, D. V. (2013). Supply chain management: strategy, planning, and operation (Vol. 232). *MA: Pearson Boston*.

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*. https://doi.org/10.48550/arXiv.1412.3555

Dang, Y., Chen, Z., Li, H., & Shu, H. (2022). A comparative study of non-deep learning, deep learning, and ensemble learning methods for sunspot number prediction. *Applied Artificial Intelligence*, *36*(1), 2074129. https://doi.org/10.1080/08839514.2022.2074129

Dharshini M. P. A., & Vijila S. A. (2021) Survey of machine learning and deep learning approaches on sales forecasting, In *2021 4th International Conference on Computing and Communications Technologies (ICCCT),* 59-64. https://doi:10.1109/ICCCT53315.2021.9711878

Demsar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine learning research*, *7*, 1-30.

Deng, X., Tian, X., Chen, S., & Harris, C. J. (2017, May). Deep learning based nonlinear principal component analysis for industrial process fault detection. In *2017 International Joint Conference on Neural Networks (IJCNN)* (pp. 1237-1243). IEEE. https://doi.org/10.1109/IJCNN.2017.7965994

Doran, D., Schulz, S., & Besold, T. R. (2017). What does explainable AI really mean? A new conceptualization of perspectives. *arXiv preprint arXiv:1710.00794*. https://doi.org/10.48550/arXiv.1710.00794

Dunn, J. C. (1974). Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, *4*(1), 95-104. https://doi.org/10.1080/01969727408546059

Eglite, L., and Birzniece, I. (2022) Retail sales forecasting using deep learning: systematic literature review. *Complex Systems Informatics And Modeling Quarterly,* (30), 53-62. https://doi:10.7250/csimq.2022-30.03

Erol, B., & İnkaya, T. (2024). Satış tahmini için uzun kısa-süreli bellek ağı tabanlı derin transfer öğrenme yaklaşımı. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, *39*(1), 191-202. https://doi.org/10.17341/gazimmfd.1089173

Facebook Infrastructure Data Science, (2021). "Kats," https://facebookresearch.github.io/Kats/

Friedman, J. H. (1984). A variable span scatterplot smoother. *Technical Report 5. Laboratory for Computational Statistics*, Stanford University.

Fukuoka, R., Suzuki, H., Kitajima, T., Kuwahara, A., & Yasuno, T. (2018). Wind speed prediction model using LSTM and 1D-CNN. *Journal of Signal Processing*, *22*(4), 207-210. https://doi.org/10.2299/jsp.22.207

Fulcher, B. D., & Jones, N. S. (2017). hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. *Cell systems*, *5*(5), 527-531. https://doi.org/10.1016/j.cels.2017.10.001

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M., & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, *115*, 105151. https://doi.org/10.1016/j.engappai.2022.105151

Garland, J., James, R., & Bradley, E. (2014). Model-free quantification of time-series predictability. *Physical Review E, 90*(5), 052910. https://doi.org/10.1103/PhysRevE.90.052910

Garza, A., & Mergenthaler-Canseco, M. (2023). TimeGPT-1. *arXiv preprint arXiv:2310.03589*. https://doi.org/10.48550/arXiv.2310.03589

Gashler, M. S., & Ashmore, S. C. (2016). Modeling time series data with deep Fourier neural networks. *Neurocomputing*, *188*, 3-11. https://doi.org/10.1016/j.neucom.2015.01.108

Géron, A. (2017). Hands-on machine learning with Scikit-Learn, Keras and TensorFlow. First edition, *O'Reilly Media, Inc.*

Ghahramani, Z. (2003). Unsupervised learning. In *Summer school on machine learning* (pp. 72-112). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-28650-9_5

Github. (2021). https://github.com/prathameshkurunkar7/Supermarket-Sales-Forecasting-And-Analysis

Goerg, G. M. (2013, June). Forecastable component analysis. In *Proceedings of the 30th International Conference on International Conference on Machine Learning-Volume 28* (pp. 64-72).

Goerg, G. M. (2014). ForeCA: An R package for Forecastable Component Analysis. R package version 0.1. URL: http://cran.rproject.org/package=ForeCA.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goyal, A., Krishnamurthy, S., Kulkarni, S., Kumar, R., Vartak, M., & Lanham, M. A. (2018, April). A solution to forecast demand using long short-term memory recurrent neural networks for time series forecasting. *In Proceedings of the Midwest Decision Sciences Institute Conference*, Indianapolis, IN, USA (pp. 12-14).

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural networks*, *18*(5-6), 602-610. https://doi.org/10.1016/j.neunet.2005.06.042

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., & Schmidhuber, J. (2016). LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*, *28*(10), 2222-2232. https://doi.org/10.1109/TNNLS.2016.2582924

Gregorutti, B., Michel, B., & Saint-Pierre, P. (2017). Correlation and variable importance in random forests. *Statistics and Computing*, *27*, 659-678. https://doi.org/10.1007/s11222-016-9646-1

Gu, Q., & Dai, Q. (2021). A novel active multi-source transfer learning algorithm for time series forecasting. *Applied Intelligence*, *51*, 1326-1350. https://doi.org/10.1007/s10489-020-01871-5

Gu, Q., Dai, Q., Yu, H., & Ye, R. (2021). Integrating multi-source transfer learning, active learning and metric learning paradigms for time series prediction. *Applied Soft Computing*, *109*, 107583. https://doi.org/10.1016/j.asoc.2021.107583

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of machine learning research*, *3*(Mar), 1157-1182.

Han, J., Kamber, M., Pei, J. (2012). Data Mining Concepts and Techniques, *Morgan Kaufmann Publishers Inc.*

Hansen, L. K., & Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, *12*(10), 993-1001. https://doi.org/10.1109/34.58871

Harris, B. (2021). "brendanjohnharris/Catch22.jl: v0.2.1," Zenodo, https://doi.org/10.5281/zenodo.5030712.

Helmini, S., Jihan, N., Jayasinghe, M., & Perera, S. (2019). Sales forecasting using multivariate long short term memory network models. *PeerJ PrePrints*, *7*, e27712v1. https://doi.org/10.7287/peerj.preprints.27712v1

Henderson, T. (2021). Rcatch22: Calculation of 22 Canonical time-series characteristics. *R package version 0.1*, *13*.

Henderson, T., & Fulcher, B. D. (2021, December). An empirical evaluation of time-series feature sets. In *2021 International Conference on Data Mining Workshops (ICDMW)* (pp. 1032-1038). IEEE. https://doi.org/10.1109/ICDMW53433.2021.00134

Herff, C., Krusienski, D.J. (2019). Extracting Features from Time Series. In: Kubben, P., Dumontier, M., Dekker, A. (eds) Fundamentals of Clinical Data Science. Springer, Cham. https://doi.org/10.1007/978-3-319-99713-17

Hirt, R., Srivastava, A., Berg, C., & Kühl, N. (2020). Sequential transfer machine learning in networks: Measuring the impact of data and neural net similarity on transferability. *arXiv preprint arXiv:2003.13070*. https://doi.org/10.48550/arXiv.2003.13070

Hochreiter, S., & Schmidhuber, J. (1996). LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, *9*.

Hossain, M., Rekabdar, B., Louis, S. J., & Dascalu, S. (2015, July). Forecasting the weather of Nevada: A deep learning approach. In *2015 international joint conference on neural networks (IJCNN)* (pp. 1-6). IEEE. https://doi.org/10.1109/IJCNN.2015.7280812

Hyndman R. J., Kang Y, Montero-Manso P, Talagala T, Wang E, Yang Y, O'Hara-Wild M (2020). tsfeatures: Time Series Feature Extraction. *R package version 1.0.2*, URL https: //CRAN.R-project.org/package=tsfeatures.

Hyndman, R. J., Wang, E., & Laptev, N. (2015, November). Large-scale unusual time series detection. In *2015 IEEE international conference on data mining workshop (ICDMW)* (pp. 1616-1619). IEEE. https://doi.org/10.1109/ICDMW.2015.104

Hu, Q., Zhang, R., & Zhou, Y. (2016). Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy*, *85*, 83-95. https://doi.org/10.1016/j.renene.2015.06.034

Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A review of deep transfer learning and recent advancements. *Technologies, 11*(2), 40.

Iman, R. L., & Davenport, J. M. (1980). Approximations of the critical region of the fbietkan statistic. *Communications in Statistics-Theory and Methods*, *9*(6), 571-595. https://doi.org/10.1080/03610928008827904

Ingle, C., Bakliwal, D., Jain, J., Singh, P., Kale, P., and Chhajed, V. (2021) Demand forecasting: Literature review on various methodologies, In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1-7. https://doi:10.1109/ICCCNT51525.2021.9580139

Jiang, W. (2020). Time series classification: Nearest neighbor versus deep learning models. *SN Applied Sciences*, *2*(4), 721. https://doi.org/10.1007/s42452-020-2506-9

Kaggle: Makine Öğrenimi ve Veri Bilimi Topluluğunuz, https://www.kaggle.com/datasets. Erişim tarihi Ekim 5, 2020.

Kaneko, Y., & Yada, K. (2016, December). A deep learning approach for the prediction of retail store sales. *In 2016 IEEE 16th International conference on data mining workshops (ICDMW)* (pp. 531-537). IEEE. https://doi.org/10.1109/ICDMW.2016.0082

Karb, T., Kühl, N., Hirt, R., & Glivici-Cotruta, V. (2020). A network-based transfer learning approach to improve sales forecasting of new products. *arXiv preprint arXiv:2005.06978*. https://doi.org/10.48550/arXiv.2005.06978

Kaya, K., Yılmaz, Y., Yaslan, Y., Öğüdücü, Ş. G., & Çıngı, F. (2022). Demand forecasting model using hotel clustering findings for hospitality industry. *Information Processing & Management*, *59*(1), 102816. https://doi.org/10.1016/j.ipm.2021.102816

Kaya S. K., & Yıldırım Ö. (2020). A prediction model for automobile sales in Turkey using deep neural networks. *Endüstri Mühendisliği*, *31*(1), 57-74.

Kaufman, L., & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Kilimci, Z. H., Akyuz, A. O., Uysal, M., Akyokus, S., Uysal, M. O., Atak Bulbul, B., & Ekmis, M. A. (2019). An improved demand forecasting model using deep learning approach and proposed decision integration strategy for supply chain. *Complexity, 2019*. https://doi.org/10.1155/2019/9067367

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://doi.org/10.48550/arXiv.1412.6980

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., & Inman, D. J. (2021). 1D convolutional neural networks and applications: A survey. *Mechanical systems and signal processing*, *151*, 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kochak, A., & Sharma, S. (2015). Demand forecasting using neural network for supply chain management. *International Journal of Mechanical Engineering and Robotics Research*, *4*(1), 96-104.

Kraus, M., Feuerriegel, S., & Oztekin, A. (2020). Deep learning in business analytics and operations research: Models, applications and managerial implications. *European Journal of Operational Research*, *281*(3), 628-641. https://doi.org/10.1016/j.ejor.2019.09.018

Lai, C. S., Zhong, C., Pan, K., Ng, W. W., & Lai, L. L. (2021). A deep learning based hybrid method for hourly solar radiation forecasting. *Expert Systems with Applications*, *177*, 114941. https://doi.org/10.1016/j.eswa.2021.114941

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

Levenshtein, V. I. (1966, February). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, No. 8, pp. 707-710).

Linardatos, P., Papastefanopoulos, V., & Kotsiantis, S. (2020). Explainable ai: A review of machine learning interpretability methods. *Entropy*, *23*(1), 18. https://doi.org/10.3390/e23010018

Livieris, I. E., Pintelas, E., Stavroyiannis, S., & Pintelas, P. (2020). Ensemble deep learning models for forecasting cryptocurrency time-series. *Algorithms*, *13*(5), 121. https://www.mdpi.com/1999-4893/13/5/121

Loureiro, A. L., Miguéis, V. L., & Da Silva, L. F. (2018). Exploring the use of deep neural networks for sales forecasting in fashion retail. *Decision Support Systems*, 114, 81-93. https://doi.org/10.1016/j.dss.2018.08.010

Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., & Jones, N. S. (2019). catch22: CAnonical Time-series CHaracteristics: Selected through highly comparative time-series analysis. *Data Mining and Knowledge Discovery*, *33*(6), 1821-1852. https://doi.org/10.1007/s10618-019-00647-x

Ma, S., & Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, *288*(1), 111-128. https://doi.org/10.1016/j.ejor.2020.05.038

Maasoumi, E., & Racine, J. (2002). Entropy and predictability of stock market returns. *Journal of Econometrics*, *107*(1–2), 291–312. https://doi.org/10.1016/S0304-4076(01)00125-7

Malhotra, P., TV, V., Vig, L., Agarwal, P., & Shroff, G. (2017). TimeNet: Pre-trained deep recurrent neural network for time series classification. *arXiv preprint arXiv:1706.08838*. https://doi.org/10.48550/arXiv.1706.08838

Martinez, W. L., & Martinez, A. R. (2001). *Computational statistics handbook with MATLAB*. Chapman and Hall/CRC.

Maulik, U., & Bandyopadhyay, S. (2002). Performance evaluation of some clustering algorithms and validity indices. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(12), 1650-1654. https://doi.org/10.1109/TPAMI.2002.1114856

Meiseles, A., & Rokach, L. (2020). Source model selection for deep learning in the time series domain. *IEEE Access*, *8*, 6190-6200. https://doi.org/10.1109/ACCESS.2019.2963742

Merigo, J. M., & Casanovas, M. (2011). Induced aggregation operators in the Euclidean distance and its application in financial decision making. *Expert systems with applications*, *38*(6), 7603-7608. https://doi.org/10.1016/j.eswa.2010.12.103

Minh, D., Wang, H. X., Li, Y. F., & Nguyen, T. N. (2022). Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, 1-66. https://doi.org/10.1007/s10462-021-10088-y

Munkhdalai, L., Park, K. H., Batbaatar, E., Theera-Umpon, N., & Ryu, K. H. (2020). Deep learning-based demand forecasting for Korean postal delivery service. *IEEE Access*, *8*, 188135-188145. https://doi.org/10.1109/ACCESS.2020.3030938

Müller, M. (2007). Dtw-based motion comparison and retrieval. *Information Retrieval for Music and Motion*, 211-226. https://doi.org/10.1007/978-3-540-74048-3_10

Nassar, L., Okwuchi, I. E., Saad, M., Karray, F., Ponnambalam, K., & Agrawal, P. (2020, July). Prediction of strawberry yield and farm price utilizing deep learning. In *2020 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-7). IEEE. https://doi.org/10.1109/IJCNN48605.2020.9206998

Ncir, C. E. B., Hamza, A., & Bouaguel, W. (2021). Parallel and scalable Dunn Index for the validation of big data clusters. *Parallel Computing*, *102*, 102751. https://doi.org/10.1016/j.parco.2021.102751

Niu, S., Liu, Y., Wang, J., & Song, H. (2020). A decade survey of transfer learning (2010–2020). *IEEE Transactions on Artificial Intelligence*, *1*(2), 151-166. https://doi.org/10.1109/TAI.2021.3054609

Nutakki, G. C., Abdollahi, B., Sun, W., & Nasraoui, O. (2019). An introduction to deep clustering. *Clustering Methods for Big Data Analytics: Techniques, Toolboxes and Applications*, 73-89. https://doi.org/10.1007/978-3-319-97864-2_4

O'Hara-Wild, M., Hyndman, R., Wang, E., Cook, D., Talagala, T. S., & Chhay, L. (2021). feasts: Feature extraction and statistics for time series. R package version 0.2. 2.

Okwuchi, I., Nassar, L., Karray, F., & Ponnambalam, K. (2020, October). Deep learning ensemble based model for time series forecasting across multiple applications. In *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 3077-3083). IEEE. https://doi.org/10.1109/SMC42975.2020.9282948

Ongsulee, P. (2017, November). Artificial intelligence, machine learning and deep learning. In *2017 15th international conference on ICT and knowledge engineering (ICT&KE)* (pp. 1-6). IEEE.

Oroojlooyjadid, A., Snyder, L. V., & Takáč, M. (2020). Applying deep learning to the newsvendor problem. *IISE Transactions*, *52*(4), 444-463. https://doi.org/10.1080/24725854.2019.1632502

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, *22*(10), 1345-1359. https://doi.org/10.1109/TKDE.2009.191

Pan, H., & Zhou, H. (2020). Study on convolutional neural network and its application in data mining and sales forecasting for E-commerce. *Electronic Commerce Research*, *20*(2), 297-320. https://doi.org/10.1007/s10660-020-09409-0

Polikar, R. (2012). Ensemble learning. *Ensemble machine learning: Methods and applications*, 1-34. https://doi.org/10.1007/978-1-4419-9326-7_1

Punia, S., Nikolopoulos, K., Singh, S. P., Madaan, J. K., & Litsiou, K. (2020). Deep learning with long short-term memory networks and random forests for demand forecasting in multi-channel retail. *International journal of production research*, *58*(16), 4964-4979. https://doi.org/10.1080/00207543.2020.1735666

Punia, S., Singh, S. P., & Madaan, J. K. (2020). From predictive to prescriptive analytics: A data-driven multi-item newsvendor model. *Decision Support Systems*, *136*, 113340. https://doi.org/10.1016/j.dss.2020.113340

Puspita, P. E., İnkaya, T., & Akansel, M. (2019). Clustering-based sales forecasting in a forklift distributor. *International Journal of Engineering Research and Development*, *11*(1), 25-40. https://doi.org/10.29137/umagd.473977

Ragab, M. G., Abdulkadir, S. J., Aziz, N., Al-Tashi, Q., Alyousifi, Y., Alhussian, H., & Alqushaibi, A. (2020). A novel one-dimensional cnn with exponential adaptive gradients for air pollution index prediction. *Sustainability*, *12*(23), 10090. https://doi.org/10.3390/su122310090

Ren, S., Choi, T. M., Lee, K. M., & Lin, L. (2020). Intelligent service capacity allocation for cross-border-E-commerce related third-party-forwarding logistics operations: A deep learning approach. *Transportation Research Part E: Logistics and Transportation Review*, *134*, 101834. https://doi.org/10.1016/j.tre.2019.101834

Ren, Y., Pu, J., Yang, Z., Xu, J., Li, G., Pu, X., ... & He, L. (2022). Deep clustering: A comprehensive survey. *arXiv preprint arXiv:2210.04142*. https://doi.org/10.48550/arXiv.2210.04142

Ribeiro, M. H. D. M., da Silva, R. G., Moreno, S. R., Mariani, V. C., & dos Santos Coelho, L. (2022). Efficient bootstrap stacking ensemble learning model applied to wind power generation forecasting. *International Journal of Electrical Power & Energy Systems*, *136*, 107712. https://doi.org/10.1016/j.ijepes.2021.107712

Ribeiro, M. H. D. M., & dos Santos Coelho, L. (2020). Ensemble approach based on bagging, boosting and stacking for short-term prediction in agribusiness time series. *Applied soft computing*, *86*, 105837. https://doi.org/10.1016/j.asoc.2019.105837

Rose, O. (1996). Estimation of the Hurst parameter of long-range dependent time series. *University of Würzburg Institute of Computer Science Research Report Series*, (Vol. 137). Report.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53-65. https://doi.org/10.1016/0377-0427(87)90125-7

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, *26*(1), 43–49.

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International journal of forecasting*, *36*(3), 1181-1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Shen, S., Sadoughi, M., Li, M., Wang, Z., & Hu, C. (2020). Deep convolutional neural networks with ensemble learning and transfer learning for capacity estimation of lithium-ion batteries. *Applied Energy*, *260*, 114296. https://doi.org/10.1016/j.apenergy.2019.114296

Sherstinsky, A. (2020). Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena*, *404*, 132306. https://doi.org/10.1016/j.physd.2019.132306

Shi, X., Chen, Z., Wang, H., Yeung, D. Y., Wong, W. K., & Woo, W. C. (2015). Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, *28*.

Shokoohi-Yekta, M., Hu, B., Jin, H., Wang, J., & Keogh, E. (2017). Generalizing DTW to the multi-dimensional case requires an adaptive approach. *Data mining and knowledge discovery*, *31*, 1-31. https://doi.org/10.1007/s10618-016-0455-0

Sun, S., Wei, Y., & Wang, S. (2018). AdaBoost-LSTM ensemble learning for financial time series forecasting. In *Computational Science–ICCS 2018: 18th International Conference, Wuxi, China, June 11–13, 2018 Proceedings, Part III 18* (pp. 590-597). Springer International Publishing. https://doi.org/10.1007/978-3-319-93713-7_55

Swapna, G., Kp, S., & Vinayakumar, R. (2018). Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals. *Procedia computer science*, *132*, 1253-1262. https://doi.org/10.1016/j.procs.2018.05.041

Şeker, A., Diri, B., & Balık, H. H. (2017). Derin öğrenme yöntemleri ve uygulamaları hakkında bir inceleme. *Gazi Mühendislik Bilimleri Dergisi*, *3*(3), 47-64.

Tan, P. N., Steinbach, M., & Kumar, V. (2016). *Introduction to data mining*. Pearson Education India.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242-264). IGI global. https://doi.org/10.4018/978-1-60566-766-9.ch011

Vangala, S., & Vadlamani, R. (2020). ATM Cash demand forecasting in an Indian Bank with chaos and deep learning. *arXiv preprint arXiv:2008.10365*. https://doi.org/10.48550/arXiv.2008.10365

Wang, J., Liu, G. Q., & Liu, L. (2019, March). A selection of advanced technologies for demand forecasting in the retail industry. In *2019 IEEE 4th International Conference on Big Data Analytics (ICBDA)* (pp. 317-320). IEEE. https://doi.org/10.1109/ICBDA.2019.8713196

Wang, J., Yu, L. C., Lai, K. R., & Zhang, X. (2016, August). Dimensional sentiment analysis using a regional CNN-LSTM model. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)* (pp. 225-230).

Wang, L., Peng, H., Tan, M., & Pan, R. (2021). A multistep prediction of hydropower station inflow based on bagging-LSTM model. *Discrete Dynamics in Nature and Society*, *2021*, 1-12. https://doi.org/10.1155/2021/1031442

Wang, S., Jiang, Y., Hou, X., Cheng, H., & Du, S. (2017). Cerebral micro-bleed detection based on the convolution neural network with rank based average pooling. *IEEE Access*, *5*, 16576-16583. https://doi.org/10.1109/ACCESS.2017.2736558

Wang, T. (2021). A K-means group division and lstm based method for hotel demand forecasting. *Tehnicki Vjesnik-Technical Gazette*, *28*(4), 1345-1353. https://doi.org/10.17559/TV-20210507172841

Wang, T., Li, L., & Huang, W. (2020). Research on the Construction of Sales Forecasting Model of Fashion Products Based on Feature Representation of Multimodal and Deep Learning. *WHICEB 2020 Proceedings*. 33. https://aisel.aisnet.org/whiceb2020/33

Wang, X., Smith, K., & Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data mining and knowledge Discovery*, *13*, 335-364. https://doi.org/10.1007/s10618-005-0039-x

Weng, T., Liu, W., & Xiao, J. (2020). Supply chain sales forecasting based on lightGBM and LSTM combination model. *Industrial Management & Data Systems*, *120*(2), 265-279. https://doi.org/10.1108/IMDS-03-2019-0170

Willinger, W., Paxson, V., & Taqqu, M. S. (1998). Self-similarity and heavy tails: Structural modeling of network traffic. *A practical guide to heavy tails: statistical techniques and applications*, *23*(1), 27-53.

Wolpert, D. H. (1992). Stacked generalization. *Neural networks*, *5*(2), 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1

Xiao, C., Chen, N., Hu, C., Wang, K., Gong, J., & Chen, Z. (2019). Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach. *Remote Sensing of Environment*, *233*, 111358. https://doi.org/10.1016/j.rse.2019.111358

Yang, M., Zhao, M., Huang, D., & Su, X. (2022). A composite framework for photovoltaic day-ahead power prediction based on dual clustering of dynamic time warping distance and deep autoencoder. *Renewable Energy*, *194*, 659-673. https://doi.org/10.1016/j.renene.2022.05.141

Ye, R., & Dai, Q. (2018). A novel transfer learning framework for time series forecasting. *Knowledge-Based Systems*, *156*, 74-99. https://doi.org/10.1016/j.knosys.2018.05.021

Ye, R., & Dai, Q. (2021). Implementing transfer learning across different datasets for time series forecasting. *Pattern Recognition*, *109*, 107617. https://doi.org/10.1016/j.patcog.2020.107617

Zhang, D., Lin, R., Wei, T., Ling, L., & Huang, J. (2023). A novel deep transfer learning framework with adversarial domain adaptation: application to financial time-series forecasting. *Neural Computing and Applications*, *35*(34), 24037-24054. https://doi.org/10.1007/s00521-023-09047-1

Zhang, W., Deng, L., Zhang, L., & Wu, D. (2022). A survey on negative transfer. *IEEE/CAA Journal of Automatica Sinica*, *10*(2), 305-329. https://doi.org/10.1109/JAS.2022.106004

Zhang, K., Zhang, Y., Zhang, G., He, X., & Yang, J. (2021). A novel hybrid model based on ESMD-PE and mRMR-LSTM-Adaboost for short-term wind power prediction. *AIP Advances*, *11*(12). https://doi.org/10.1063/5.0060920

Zhao, K., & Wang, C. (2017). Sales forecast in e-commerce using convolutional neural network. *arXiv preprint arXiv:1708.07946*. https://doi.org/10.48550/arXiv.1708.07946

Zhou, P., Chen, L., Dai, X., Li, B., & Chai, T. (2020). Intelligent prediction of train delay changes and propagation using RVFLNs with improved transfer learning and ensemble learning. *IEEE Transactions on Intelligent Transportation Systems*, *22*(12), 7432-7444. https://doi.org/10.1109/TITS.2020.3002785

Zhou, S., Xu, H., Zheng, Z., Chen, J., Bu, J., Wu, J., ... & Ester, M. (2022). A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. arXiv preprint arXiv:2206.07579. https://doi.org/10.48550/arXiv.2206.07579

Zhou, Z. H. (2012). *Ensemble methods: foundations and algorithms*. CRC press.

Zhou, Z. H., & Zhou, Z. H. (2021). Ensemble learning (pp. 181-210). Springer Singapore.

EKLER

EK 1 Temel DL Yöntemleri Kullanarak Satış ve Talep Tahmini Yapan Çalışmalar

EK 2 DL Yöntemlerde Teknik Bir İyileştirme veya Geliştirme Sağlayarak Satış ve Talep Tahmini Yapan Çalışmalar

EK 3 Birleşik DL Yöntemleri veya Topluluk Tabanlı DL Yöntemleri Kullanarak Satış ve Talep Tahmini Yapan Çalışmalar

EK 4 Zaman Serisi Öznitelikleri ve Açıklamaları

EK 5 Öznitelikler ve Açıklamaları

EK 6 Rastgele Orman için Deneysel Sonuçlar

EK 7 Seçilen Öznitelikler ve Önem Dereceleri ile Oluşturulmuş Ağırlıklı Öklid Uzaklıkları Matrisi

EK 8 Tek Değişkenli Satış Zaman Serileri Arasındaki DTW Uzaklıkları

EK 9 Tek Değişkenli Satış Zaman Serileri Arasındaki ERP Uzaklıkları

EK 10 Normalize Edilmiş DTW Uzaklıkları

EK 11 Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş DTW Uzaklıkları ($w_{DTW} = 0{,}26$)

EK 12 Normalize Edilmiş ERP Uzaklıkları

EK 13 Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş ERP Uzaklıkları ($w_{ERP} = 0{,}23$)

EK 14 Tek Değişkenli Zaman Serileri için Tek Kaynaklı TL'deki RMSE İyileşme Oranları

EK 15 Pharma-Seri_18 Hedef Verisi için Learnx-Seri_4 Kaynak Veri Ağırlıkları

EK 16 Pharma-Seri_18 Hedef Verisi için Learnx-Seri_5 Kaynak Veri Ağırlıkları

EK 17 Pharma-Seri_18 Hedef Verisi için Pharma-Seri_17 Kaynak Veri Ağırlıkları

EK 18 Pharma-Seri_18 Hedef Verisi için Yöntemlerin Performanslarının Karşılaştırılması

EK 19 Tüm Satış Verileri için Seçilen Öznitelikler ve Önem Dereceleri ile Oluşturulmuş Ağırlıklı Öklid Uzaklıkları Matrisi

EK 20 Tüm Satış Verileri için Normalize Edilmiş DTW Uzaklıkları

EK 21 Tüm Satış Verileri için Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş DTW Uzaklıkları ($w_{DTW} = 0{,}26$)

EK 22 Tüm Satış Verileri için Normalize Edilmiş ERP Uzaklıkları

EK 23 Tüm Satış Verileri için Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş ERP Uzaklıkları ($w_{ERP} = 0{,}23$)

EK 24 Çok Değişkenli Satış Zaman Serileri Arasındaki DTW_D Uzaklıkları

EK 25 Çok Değişkenli Zaman Serileri için Tek Kaynaklı TL'deki RMSE İyileşme Oranları

EK 1 Temel DL Yöntemleri Kullanarak Satış ve Talep Tahmini Yapan Çalışmalar

Makale	Amaç	Yöntem	Veri Tipi ve Kaynağı	Girdi (Öznitelik Kümesi)	Çıktı*	Sonuçlar
Kaneko vd. (2016)	DL yaklaşımını kullanarak perakende mağazalar için bir satış tahmini modeli oluşturmak	DNN	Süpermarketlerin üç yıllık POS verileri	Ürün özellikleri (genel, Japon ve yabancı yiyecekler, sebze ve meyveler, vb.)	TAT	Tahmin doğruluğunun hesaba katılan ürün özniteliklerinin sayısına göre değiştiği, bu nedenle özniteliklerin önemli olduğu belirtilmiştir.
Loureiro vd. (2018)	Moda endüstrisindeki satışları tahmin etmek ve gelecek sezonlarda yeni bireysel ürünlerin satışlarını öngörmek için DL yaklaşımını kullanmak	DNN, DT, RF, SVM, NN ve Regresyon	Bir şirketin geçmiş gerçek satış verileri	Renk, beden, moda, ürün grubu, müşteri segmenti, fiyat, satış beklenti seviyesi, vb.	TAT	DL ve RF moda satış tahmininde iyi bir performansa sahip olduğu ve tahmini iyileştirmek için modaya özgü başka özniteliklerin de dikkate alınması gerektiği belirtilmiştir.
Goyal vd. (2018)	Geleneksel talep tahmin modellerinin dezavantajlarını dengeleyen RNN ve LSTM ile talebi etkili bir şekilde tahmin etmek	RNN, LSTM, NN, ETS	M3 yarışma tahmin veri kümesi	Zaman serisi (aylık talep miktarları), ürün kategorileri, mevsimsellik ve trend bileşenleri	ÇAT	LSTM, temel modellere kıyasla üstün performansı ile TZ, Lojistik ve diğer endüstrilerde talep tahmininde başarılı bir karar destek aracı olarak düşünülebilir.
Wang vd. (2019)	Bir perakende zincirinde bozulabilir ve dayanıklı ürünlere olan farklı yöntemlerle tahmin etmek ve sonuçları karşılaştırmak	ARIMA, SVM, RNN ve LSTM	Yüzlerce süpermarketin satış veri seti	Haftanın günleri, petrol fiyatı, tatiller ve doğal afetler gibi dış faktörler, vb.	TAT	SVM ve LSTM, tahmin görevlerinde daha iyi bir performansa sahiptir.
Helmini vd. (2019)	Satış tahmini için gözetleme deliği bağlantılarına sahip LSTM modelini kullanmak	LSTM, RF	Rossmann süpermarketlerinin satış verileri	Zaman serisi, haftanın günleri, satışlar, promosyon, okul tatil günleri, vb.	TAT	Gözetleme bağlantılarına sahip LSTM'in satış tahmininde üstün olduğu vurgulanmıştır.
Punia vd. (2020)	Sipariş miktarlarını bulmak için DL yöntemi kullanmak ve tahmin sonuçları ile kapasite kısıtlamalı çok öğeli envanter optimizasyonu gerçekleştirmek	DNN, ARIMA, NN, RF, ETS ve Regresyon	Çabuk bozulan gıda ürünlerinin geçmiş haftalık satış verileri	Ürün türü ve satış miktarı, satış noktaları, promosyonlar, mağaza, tatiller, hava durumu, petrol fiyatı, ekonomik indeks vb.	TAT	DL tabanlı talep tahmini riskli ürünlerin satış tahmininde üstün performans göstermektedir.

* TAT: Tek Adımlı Tahmin; ÇAT: Çok Adımlı Tahmin

EK 2 DL Yöntemlerde Teknik Bir İyileştirme veya Geliştirme Sağlayarak Satış ve Talep Tahmini Yapan Çalışmalar

Makale	Amaç	DL Yöntem ve Teknik İyileştirme Türü	Veri Tipi ve Kaynağı	Girdi (Öznitelik Kümesi)	Çıktı*	Sonuçlar
Oroojlooyjadid vd. (2019)	DL algoritmasını haber satıcısı problemine veri özellikleriyle uyarlamak için, envanter yokluğu ve elde tutma maliyetlerini dikkate alan revize edilmiş bir kayıp fonksiyonu ile kullanarak sipariş miktarlarını optimize etmek	Aktivasyon fonksiyonunu değiştirme - DNN	Bir perakendecinin geçmiş satış veri kümesi	Haftanın günleri, sipariş miktarı, gerçekleşen talep, stok maliyeti, hurda maliyeti, vb.	ÇAT	Aktivasyon fonksiyonunu değiştirerek envanter optimizasyonunu sağlayan DL algoritması özellikle yüksek volatiliteli veya seyrek talepler için daha iyi performansa sahiptir.
Chen vd. (2020)	Birden çok ilgili zaman serisi tahmini için CNN'ye dayalı olasılıklı bir tahmin çerçevesi önermek ve farklı aktivasyon fonksiyonları ile hem parametrik hem de parametrik olmayan ayarlar altında olasılık yoğunluğunu tahmin etmek	Aktivasyon fonksiyonunu değiştirme - DeepTCN-Quantile ve DeepTCN-Gaussian	Bir perakendecinin günlük talep ve sevkiyat veri kümeleri. Bir otomobil şirketinin parça satışı veri kümesi.	Zamana bağlı (ürün fiyatı, haftanın günü, tatil vb.) ve zamandan bağımsız (ürün markası, depo kategorisi (yiyecek, moda), coğrafi bilgiler vb.) özellikler	TAT	DeepTCN, mevsimsellik, seri içinde ve arasında tatil etkileri gibi karmaşık kalıpları öğrenebilir ve özellikle geçmiş veriler seyrek olduğunda veya mevcut olmadığında, daha doğru tahminler için bu kalıplardan faydalanabilir.
Kraus vd. (2019)	Yeni bir derin gömülü ağ (Deep embedded LSTM) önererek özelleştirilmiş mimarilerin değerini vurgulamak ve bu amaçla eczanelerin gün öncesi satış hacmini tahmin etmek	Katman bazında iyileşme - Deep embedded LSTM	1113 farklı eczaneden alınmış geçmiş günlük satış verisi	Mağaza türü, resmi veya okul tatili, ürün çeşitliliği, bir sonraki rakip mağazaya uzaklık, haftanın günü, vb.	TAT	LSTM'e gömme katmanları dahil edilmesi ile tahmin sonuçları iyileştirilmiştir. Farklı ağ mimarileri, eğitim rutinleri ve parametre ayarları ile tahmin performansı iyileştirilebilir.

* TAT: Tek Adımlı Tahmin; ÇAT: Çok Adımlı Tahmin

EK 2 DL Yöntemlerde Teknik Bir İyileştirme veya Geliştirme Sağlayarak Satış ve Talep Tahmini Yapan Çalışmalar (devam)

Makale	Amaç	DL Yöntem ve Teknik İyileştirme Türü	Veri Tipi ve Kaynağı	Girdi (Öznitelik Kümesi)	Çıktı*	Sonuçlar
Munkhdalai vd. (2020)	Posta merkezinde talebi tahmin etmek için seçim ve güncelleme katmanlarıyla (MLP-SUL) donatılmış yeni bir DL model önermek.	Katman bazında iyileşme - MLP-SUL	24 posta merkezinin posta zaman serisi verileri	Zamansal değişkenler (tatil, önceki_tatil, 2 günlük tatilden sonra, hafta sonu vb.), olaylar (ay yeni yılı, Chuseok)	ÇAT	MLP-SUL modeli seçim ve güncelleme katmanları sayesinde uzun süreli tatiller, özel günler, vb. nedeniyle büyük talep değişkenliği yaşanan çeşitli olaylara dayalı talep ve satış tahmininde uygulanabilir.
Chen vd. (2018)	Satış tahmini için ikili dikkat içeren çok görevli RNN'ler ile eğilim hizalaması yapan TADA adlı yeni bir çerçeve önermek	Yeni bir yöntem geliştirme - TADA	Favorita ve bir güneş enerjisi cihazı (güneş panelleri, piller, vb.) tedarikçisinin satış verileri	Dahili (mağaza türü, ürün ailesi ve indeksi, pil türü vb.) ve harici (promosyon, petrol fiyatı, tatil, ödeme günü, indirim, güneşe maruz kalma vb.) özellikler	TAT	TADA satış zaman serilerini etkileyen faktörleri dahili ve harici özellikler olarak sınıflandıran ve bu iki yönü çok görevli LSTM kodlayıcı ile yenilikçi bir şekilde modelleyen ilk çalışmadır.
Salinas vd. (2020)	Çok sayıda ilgili zaman serisinde doğru olasılıklı tahminler üretmek için DeepAR adlı yeni bir metodoloji önermek	Yeni bir yöntem geliştirme - DeepAR	Aylık otomobil ve parça satışlarını içeren veri kümesi ve Amazon haftalık ürün satış verileri	Zaman serisi (zaman ve satış miktarları)	TAT	DeepAR, ilgili zaman serilerinden küresel bir model öğrenmede etkilidir ve mevsimsellik ve belirsizlik gibi karmaşık kalıpları öğrenebilir. Minimum düzeyde hiper parametre ayarı gibi manuel çalışma gerektirir.

***** TAT: Tek Adımlı Tahmin; ÇAT: Çok Adımlı Tahmin

EK 3 Birleşik DL Yöntemleri veya Topluluk Tabanlı DL Yöntemleri Kullanarak Satış ve Talep Tahmini Yapan Çalışmalar

Makale	Amaç	Birleşik Yöntem / Topluluk Öğrenme	Veri Tipi ve Kaynağı	Girdi (Öznitelik Kümesi)	Çıktı*	Sonuçlar
Kilimci vd. (2019)	Stokları optimize etmek ve maliyetleri düşürmek için akıllı bir talep tahmin sistemi geliştirmek	Topluluk öğrenme modeli: DL model, zaman serisi modelleri ve SVR	Süpermarket satış ve stok verileri	Ürün, mağaza, haftalık satış ve stok miktarı, en çok ilişkili ürünün satış miktarı, hava koşulları, vb.	TAT	Talep tahmin sistemleri için derin öğrenme metodolojisi, SVR algoritması ve farklı zaman serisi yöntemlerini birleştiren bilinen ilk çalışmadır.
Arora vd. (2020)	Satış talebini tahmin etmek için DL tabanlı bir topluluk yaklaşımı önermek	Topluluk tahmin modeli: tek değişkenli, çok değişkenli zaman serisi modelleri ve LSTM	Bir toptan alkol dağıtım şirketinden alınan satış ve fiyatlandırma veri kümesi (bira, şarap ve alkollü içkiler)	Hava durumu, promosyonlar, kampanyalar, tatiller ve özel etkinlikler (festivaller, konserler vb.)	ÇAT	En çok satılan ve en yüksek hasılatlı ürünler için topluluk modellerin hatayı azalttığı ve tüm ürünler için tek bir talep modeli yerine, her üründeki hatayı önemli ölçüde azaltmak için ayrı ayrı ayarlanmış bir modele ihtiyaç olduğu belirtilmiştir.
Punia vd. (2020)	LSTM ve RF'yi birleştiren yeni bir tahmin yöntemi önermek	Birleşik model: LSTM ve RF modelleri	Tek bir çevrimiçi ve on bir çevrimdışı mağazadan oluşan çok kanallı bir perakendecinin paketlenmiş gıda satış veri kümesi	Ürüne ait (kategori, fiyat, marka vb.), işleme ait (satış, indirim, tatil), mağazaya ait (ziyaretler, hane sayısı, park yeri vb.) öznitelikler	ÇAT	Çevrimiçi kanal için LSTM, yeterli bağımsız değişkenlerle çevrimdışı kanal için RF ve her iki kanal için birleşik yöntem talep tahmini için en uygun olarak bulunmuştur. Tahminlerdeki ani değişiklikler (tatiller, promosyonlar vb.) yakalanmıştır. Değişkenlerin önem sırası ve göreceli önemi ile çok kanallı bir perakendecinin promosyon etkinliklerini yönetmesi ve çeşitlilik planlama kararları vermesi sağlanır.

***** TAT: Tek Adımlı Tahmin; ÇAT: Çok Adımlı Tahmin

EK 3 Birleşik DL Yöntemleri veya Topluluk Tabanlı DL Yöntemleri Kullanarak Satış ve Talep Tahmini Yapan Çalışmalar (devam)

Makale	Amaç	Birleşik Yöntem / Topluluk Öğrenme	Veri Tipi ve Kaynağı	Girdi (Öznitelik Kümesi)	Çıktı*	Sonuçlar
Weng vd. (2019)	Tedarik zinciri satışlarını tahmin etmek için DL tabanlı birleşik yöntem önermek	Birleşik model: lightGBM ve LSTM	Üç TZ satış veri kümesi: Rossmann, Favorita ve Jollychic E-ticaret satış veri kümesi	Zaman serisi ve istatistiksel özellikler, detay özellikler (ürün türü, bozulabilirliği, geçmiş promosyon günleri vb.)	TAT	Sonuçlar, birleşik modelin TZ satışlarını doğru bir şekilde tahmin edebildiğini ve tahmin sonuçlarının yorumlanabilir olduğunu göstermektedir. Kurumsal TZ'nin iş ve üretim modu, günlük yönetimi, fiyat yönetimi ve pazarlamasını iyileştirilebilecek önemli sonuçlardır.
Ren vd. (2020)	Envanter optimizasyonu ve talep tahmin sürecini entegre ederek DL tabanlı tek adımlı optimum karar verme yaklaşımı sunmak	Birleşik Model: Seq2Seq tabanlı CNN-LSTM	Lojistik hizmetine ait gerçek dünya talep veri kümesi	Zaman serisi (zaman ve talep miktarı)	TAT	Önerilen yaklaşım, nokta tahmin sonuçları üretebilir ve dinamik bir dağıtım yoluyla talep belirsizliğini ölçerek lojistik hizmet kapasitesi tahsisi konusunda en uygun kararı verebilir.
Nassar vd. (2020)	Çilek verimini ve ardından çilek fiyatını tahmin etmek için birleşik DL modelleri önermek	Birleşik Modeller; CNN-LSTM, CNN-LSTM-GRU ve ConvLSTM	Hava durumu verileri, çilek verimi ve halka açık iki web sitesinden fiyat veri kümesi	13 hava durumu özelliği (güneş radyasyonu, hava sıcaklığı, vb.), fiyat ve verim	ÇAT	CNN-LSTM modeli, diğer birleşik ve basit DL modellerine kıyasla en iyi performansa sahiptir ve dikkatin ağının eklenmesi iyileşmeyi artırmıştır.
Wang vd. (2020)	Ürünlerin geçmiş satış özellikleri, resim özellikleri ve temel özellikleri gibi farklı bilgi modlarının farklı DL yöntemleriyle temsil edildiği bir moda ürün satış tahmin modeli önermek	Multimodal DL Model: DNN, CNN ve LSTM	Bir giyim e-ticaret şirketinin satış kayıtları veri kümesi	Ürün kimliği ve satışları, ürün resmi özellikleri (renk, doku, vb.), temel ürün özellikleri (orijinal fiyatı, pazara giriş süresi, indirimi)	TAT	Çok modlu özelliklerin kullanımı DL yöntemin tahmin doğruluğunu artırmaktadır. İşletmeler, doğru ürünleri doğru zamanda hazırlamak, dönemsel operasyon maliyetlerini düşürmek ve satış karlılıklarını artırmak için ürün satışlarını tahmin etmek için önerilen modeli kullanabilir.

***** TAT: Tek Adımlı Tahmin; ÇAT: Çok Adımlı Tahmin

EK 4 Zaman Serisi Öznitelikleri ve Açıklamaları

Mean: Ortalamayı ifade eden özniteliktir. Tek değişkenli bir zaman serisinin (*X*) ortalaması Denklem (1) ile hesaplanır (Hyndman vd., 2015).

$$\bar{X} = \frac{1}{n} * \sum_{t=1}^{n} X_t \quad (1)$$

Burada, X_t *X* zaman serisindeki *t.* zaman adımındaki gözlemin değerini, *n X* zaman serisindeki toplam gözlem sayısını ve $\bar{X}$ zaman serisindeki gözlemlerin ortalamasını ifade eder (Hyndman vd., 2015).

Variance: Varyansı ifade eden özniteliktir. Tek değişkenli bir zaman serisinin (*X*) varyansı Denklem (2) ile hesaplanır (Hyndman vd., 2015).

$$\sigma^2 = \frac{1}{n} * \sum_{t=1}^{n} (X_t - \bar{X})^2 \quad (2)$$

Burada, X_t *X* zaman serisindeki *t.* zaman adımındaki gözlemin değerini, *n X* zaman serisindeki toplam gözlem sayısını, $\bar{X}$ zaman serisindeki gözlemlerin ortalamasını ve σ^2 zaman serisinin varyansını ifade eder (Hyndman vd., 2015).

Acf1x: Birinci dereceden otokorelasyon katsayısını ifade eden özniteliktir. Birinci dereceden otokorelasyon (Acf1), bir zaman serisinin bir önceki zaman noktası ile olan otokorelasyonunu ifade eder. Tek değişkenli bir zaman serisinin (*X*) birinci dereceden otokorelasyon katsayısı (Acf1x) Denklem (3) ile hesaplanır (Hyndman vd., 2015; Kang vd., 2017).

$$Acf1x = \frac{cov(X_t,\ X_{t-1})}{\sqrt{\sigma^2(X_t) * \sigma^2(X_{t-1})}} \quad (3)$$

Burada, X_t ve X_{t-1} sırasıyla *t.* ve *(t-1).* zaman noktalarındaki zaman serisinin değerlerini, $cov(X_t,\ X_{t-1})$ X_t ve X_{t-1}arasındaki kovaryansı, $\sigma^2(X_t)$ ve $\sigma^2(X_{t-1})$ sırasıyla X_t ve X_{t-1} zaman noktalarındaki varyansları ifade eder. Acf1x değeri, -1 ile 1 arasında bir değer alır.

Pozitif bir değer, bir önceki zaman noktası ile güçlü pozitif bir ilişki olduğunu gösterirken, negatif bir değer, güçlü negatif bir ilişki olduğunu gösterir. Acf1x'in mutlak değeri bire yakınsa, güçlü bir otokorelasyon ilişkisi olduğunu gösterir. Acf1x'in sıfıra yakın olması ise otokorelasyonun zayıf olduğunu gösterir (Hyndman vd., 2015; Kang vd., 2017).

Acf1e: Hatanın birinci dereceden otokorelasyonu ifade eden özniteliktir. Hatanın birinci dereceden otokorelasyonu (ACF1e), bir modelin tahmin edilen hatalarının bir önceki tahmin edilen hatayla olan otokorelasyonunu ifade eder. Tek değişkenli bir zaman serisinin (X) hatanın birinci dereceden otokorelasyon katsayısı (Acf1) Denklem (4) ile hesaplanır (Hyndman vd., 2015).

$$Acf1e = \frac{cov(\varepsilon_t,\ \varepsilon_{t-1})}{\sqrt{\sigma^2(\varepsilon_t) * \sigma^2(\varepsilon_{t-1})}} \tag{4}$$

Burada, ε_t ve ε_{t-1} sırasıyla *t.* ve *(t-1).* zaman noktalarındaki modelin tahmin ettiği hataları, $cov(\varepsilon_t,\ \varepsilon_{t-1})$ ε_t ve ε_{t-1}arasındaki kovaryansı, $\sigma^2(\varepsilon_t)$ ve $\sigma^2(\varepsilon_{t-1})$ sırasıyla ε_t ve ε_{t-1} hata terimlerinin varyansları ifade eder. Acf1e değeri, -1 ile 1 arasında bir değer alır. Pozitif bir değer, bir önceki tahmin edilen hata ile güçlü pozitif bir ilişki olduğunu gösterirken, negatif bir değer, güçlü negatif bir ilişki olduğunu gösterir. Acf1e'in mutlak değeri bire yakınsa, güçlü bir otokorelasyon ilişkisi olduğunu gösterir. Acf1e'in sıfıra yakın olması ise otokorelasyonun zayıf olduğunu gösterir (Hyndman vd., 2015).

Pacf1x: Birinci dereceden kısmi otokorelasyonu ifade eden özniteliktir. Birinci dereceden kısmi otokorelasyon (Pacf1x), bir zaman serisinin bir önceki zaman noktası ile olan kısmi otokorelasyonunu ifade eder. Kısmi otokorelasyon, bir değişkenin bir önceki zaman noktası ile olan ilişkisini kontrol etmek için diğer ara değişkenlerin etkisinin arındırılmasını içerir. Tek değişkenli bir zaman serisinin (X) hatanın birinci dereceden otokorelasyon katsayısı (Acf1) Denklem (5) ile hesaplanır (Hyndman vd., 2015).

$$Pacf1x = \frac{cov(X_t - \widehat{X_t},\ X_{t-1} - \widehat{X_{t-1}})}{\sqrt{\sigma^2(X_t - \widehat{X_t}) * \sigma^2(X_{t-1} - \widehat{X_{t-1}})}} \tag{5}$$

Burada, X_t ve X_{t-1} sırasıyla *t.* ve *(t-1).* zaman noktalarındaki zaman serisinin değerlerini, $\widehat{X_t}$ ve $\widehat{X_{t-1}}$ *t.* ve *(t-1).* zaman noktalarında zaman serisinin tahmin edilen değerlerini, $cov(X_t - \widehat{X_t},\ X_{t-1} - \widehat{X_{t-1}})$ X_t ve X_{t-1}arasındaki kısmi kovaryansı, $\sigma^2\left(X_t - \widehat{X_t}\right)$ ve $\sigma^2\left(X_{t-1} - \widehat{X_{t-1}}\right)$ sırasıyla X_t ve X_{t-1} zaman noktalarındaki varyansları ifade eder. Pacf1x değeri, -1 ile 1 arasında bir değer alır. Pozitif bir değer, bir önceki zaman noktası ile güçlü pozitif bir ilişki olduğunu gösterirken, negatif bir değer, güçlü negatif bir ilişki olduğunu gösterir. Pacf1x'in mutlak değeri bire yakınsa, güçlü bir kısmi otokorelasyon ilişkisi olduğunu gösterir. Pacf1x'in sıfıra yakın olması ise kısmi otokorelasyonun zayıf olduğunu gösterir (Hyndman vd., 2015).

Linearity ve Curvature: Linearity doğrusallığın gücünü ve curvature eğriliğin gücünü ifade eden özniteliktir. Doğrusallık ve eğrilik, ortogonal ikinci dereceden regresyonun katsayılarına dayalı olarak hesaplanan bir zaman serisinin doğrusallığını ve eğriliğini ölçer. Tek değişkenli bir zaman serisinin eğrilik gücünü matematiksel olarak ölçmek için genellikle ikinci dereceden polinom regresyonu kullanılır. İkinci dereceden polinom regresyonu, zaman serisinin doğrusal olmayan eğriliklerini yakalamak için kullanılır. Matematiksel olarak, ikinci dereceden polinom regresyonu Denklem (6) ile ifade edilir (Hyndman vd., 2015; Hyndman vd., 2020):

$$y_t = \beta_0 + \beta_1 * t + \beta_2 * t^2 + \varepsilon_t \tag{6}$$

Burada, *t* zaman serisinin zaman değişkenini, y_t zaman serisinin *t.* zaman noktasındaki değerini ifade eder. β_0, β_1 ve β_2 sırasıyla regresyon doğrusunun sabit terimi, doğrusal terimi ve karesel terimini ifade eder. ε_t ise hata terimidir. β_1 terimi doğrusallığın gücünü, β_2 eğriliğin gücünü ifade eder. İkinci dereceden polinom regresyonunda genellikle en küçük kareler yöntemi kullanılarak katsayılar tahmin edilir. Bu tahminler, veriye en iyi uyan eğriyi temsil eder ve zaman serisinin eğriliklerini belirler (Hyndman vd., 2015; Hyndman vd., 2020).

Trend ve SeasonalStrength: Trend, trendin gücünü ifade eden özniteliktir. Trend, genellikle zamana bağlı olarak artan veya azalan bir değişim desenini ifade eder.

SeasonalStrength ise mevsimsel zaman serileri için mevsimsel gücü ifade eden özniteliktir (Hyndman vd., 2015; Hyndman vd., 2020).

Bir zaman serisini trend, mevsimsellik ve kalan olmak üzere üç bileşene ayırmak için STL (Seasonal-Trend decomposition using Loess – STL) ayrıştırması kullanılır. Bu ayrıştırma Denklem (7)'de verildiği üzere bir trend bileşeni, *M* kadar mevsimsellik bileşeni ve kalan içermektedir (Cleveland vd.,1990; Kang vd., 2017).

$$x_t = f_t + s_{1,t} + \cdots + s_{M,t} + \varepsilon_t \tag{7}$$

f_t düzleştirilmiş trend bileşeni, $s_{i,t}$ *i.* mevsimsel bileşen ve ε_t bir kalan bileşenidir. Bileşenler yinelemeli olarak tahmin edilir (Hyndman vd., 2015; Hyndman vd., 2020). Başlangıç değerleri $s_{i,t}^{(0)}$ olarak verilmişken, $s_{i,t}^{(k)}$ $s_{i,t}$ teriminin *k.* yinelemedeki tahmini olsun. *k=1,2,...* için güncellenmiş $s_{i,t}^{(k)}$ tahminlerini elde etmek için $x_t - \sum_{j\neq 1}^{j=M} s_{j,t}^{(k-1)}$ bir STL ayrıştırması uygulanır. Pratikte bu hızla yakınsar ve yalnızca iki yineleme gerekir (Hyndman vd., 2020). Prosedürün otomatik olarak uygulanmasına izin vermek için STL'ye yönelik mevsimsel pencere aralığı her durumda 21 olarak ayarlanmıştır (Hyndman vd., 2020). Mevsimsel olmayan bir zaman serisi için, f_t Friedman'ın süper pürüzsüz yöntemi kullanılarak $x_t = f_t + \varepsilon_t$ ile basitçe tahmin edilir (Friedman, 1984). Trendin gücü (trend) ve mevsimselliğin gücü (seasonal.strength) sırasıyla Denklem (8) ve Denklem (9) tanımlanır (Hyndman vd., 2015; Kang vd., 2017; Hyndman vd., 2020).

$$trend = 1 - \frac{\sigma^2(\varepsilon_t)}{\sigma^2(f_t + \varepsilon_t)} \tag{8}$$

$$seasonal.strength = 1 - \frac{\sigma^2(\varepsilon_t)}{\sigma^2(s_{i,t} + \varepsilon_t)} \tag{9}$$

Trend ve mevsimselliğin gücü değerleri sıfırdan küçükse 0, birden büyük değerler ise 1 olarak ayarlanır. Mevsimsel olmayan zaman serileri için mevsimsel güç sıfırdır. Mevsimsel zaman serileri için mevsimsel güç bir *M*-vektörüdür, burada *M* periyotların sayısıdır. Periyot verilerdeki mevsimsel dönemlerin sayısıdır. Bu sayı gözlemlerin

kendisi tarafından değil, gözlem sıklığı tarafından belirlenir ve mevsimsel olmayan veriler için periyot 1'e ayarlanır (Wang vd., 2006; Hyndman vd., 2015; Kang vd., 2017).

Mevsimsel bileşendeki tepe ve çukurların boyutu ve konumu, tepe noktalarının (zirve) ve çukurun (çukur) gücünü hesaplamak için kullanılır. İlk adım, zaman serisinin mevsimsel bileşenindeki iniş ve çıkış noktalarını belirlemektir. Zirveler yerel maksimumları, çukurlar ise yerel minimumları temsil etmektedir. Bu noktaları belirlemek için yerel maksimum/minimum tespit algoritmaları veya basit eşikleme gibi çeşitli yöntemler kullanılabilir (Hyndman vd., 2015; Kang vd., 2017; Hyndman vd., 2020).

Peak: Mevsimsel bileşendeki zirvelerin boyutunu ifade eden özniteliktir. Tek değişkenli bir zaman serisinin en yüksek noktası tepeyi ifade eder ve genellikle bu değeri bulmak için zaman serisinin tüm gözlemleri arasından en büyük değer seçilir. Bir X zaman serisinin tepe değeri $peak = \max(X_t)$ ile hesaplanır (Hyndman vd., 2015; Kang vd., 2017).

Bir zaman serisindeki tüm mevsimsel bileşenin zirveleri belirlendikten sonra boyutlarına ve konumlarına göre güçleri hesaplanır. Zirvenin gücü, zirve değeri ile komşu çukurların ortalaması arasındaki farktır. Matematiksel olarak bir zirvenin gücünü Denklem (10) ile hesaplanabilir (Hyndman vd., 2015; Kang vd., 2017):

$$P = peak - \frac{1}{m}\sum_{t=1}^{m} T_i \tag{10}$$

Burada, *peak* değeri mevsimsel bir bileşenin maksimum değerini, m komşu çukurların sayısını ve T_i i. komşu çukurun değerini ifade etmektedir (Hyndman vd., 2015; Kang vd., 2017).

Trough: Mevsimsel bileşendeki çukurların boyutunu ifade eden özniteliktir. Tek değişkenli bir zaman serisinin en düşük noktası çukuru ifade eder. Bir X zaman serisinin çukur değeri $trough = \min(X_t)$ ile hesaplanır (Hyndman vd., 2015; Kang vd., 2017).

Bir zaman serisindeki tüm mevsimsel bileşenin çukurları belirlendikten sonra boyutlarına ve konumlarına göre güçleri hesaplanır. Çukurun gücü, çukur değeri ile komşu zirvelerin ortalaması arasındaki farktır. Matematiksel olarak bir çukurun gücünü Denklem (11) ile hesaplanabilir (Hyndman vd., 2015; Kang vd., 2017):

$$T = \frac{1}{m}\sum_{t=1}^{m} T_i - trough \tag{11}$$

Burada, *trough* değeri mevsimsel bir bileşenin minimum değerini, *m* komşu zirvelerin sayısını ve P_i *i.* komşu zirvenin değerini ifade etmektedir (Hyndman vd., 2015; Kang vd., 2017).

Entropy: Bir zaman serisinin tahmin edilebilirliğinin ölçüsünü ifade eden özniteliktir. Entropi temelli ölçümler bir zaman serisinin "öngörülebilirliğinin" ölçülmesinde yaygın olarak kullanılmaktadır (Maasoumi ve Racine, 2002; Goerg, 2013; Garland vd., 2014). Sabit bir x_t sürecinin $f_x(\lambda)$ spektral yoğunluğunun Shannon entropisinin bir tahmini olan spektral entropi ölçüsü Denklem (12) ile hesaplanır (Goerg, 2013; Goerg, 2014).

$$entropy = -\int_{-\pi}^{\pi} \widehat{f_x}(\lambda) \log \widehat{f_x}(\lambda)\, d\lambda \tag{12}$$

Burada $\widehat{f_x}(\lambda)$ zaman serisinin spektrumunun bir tahminidir. $f_x(\lambda)$ x_t'nin periyot bölgesi içindeki λ frekansının önemini tanımladığı için Shannon entropisi farklı frekansların göreceli katkılarını temsil eder (Kang vd., 2017). Göreceli olarak küçük entropi değeri, x_t'nin daha fazla sinyal içerdiğini ve daha tahmin edilebilir olduğunu göstermektedir. Öte yandan, entropinin nispeten büyük değeri geleceğe ilişkin daha fazla belirsizliğe işaret eder ve bu da zaman serisini tahmin etmenin daha zor olduğunu göstermektedir (Hyndman vd., 2015; Kang vd., 2017).

Stability ve Lumpiness: Stability (kararlılık) ve lumpiness (topaklık), örtüşmeyen zaman pencerelerine dayalı iki zaman serisi özelliğidir (Hyndman vd., 2015). Tüm örtüşmeyen zaman pencereleri için ortalamalar ve varyanslar üretilir. Bu durumda stability ortalamaların varyansı iken lumpiness ise varyansların varyansıdır (Hyndman vd., 2015). Stability zaman serisinin kararlılığının ölçüsüdür. Lumpiness ise varyansın değişimini

ifade eden özniteliktir. Zaman serisindeki farklı dönemleri (örneğin aylık, üç aylık veya yıllık bazda) ifade edecek şekilde oluşturulan pencereler ile pencerelerdeki ortalamaların ve varyansların değişimleri gözlemlenir. Böylece zaman içindeki davranış değerlendirilir (Hyndman vd., 2015; Hyndman vd., 2020).

Spike: Bir zaman serisinin sivriliğinin (ani artış) ölçüsünü ifade eden özniteliktir. Hata bileşeni ε_t'nin bir-dışarıda bırakılan varyanslarının (leave-one-out variances) varyansı olarak hesaplanır. Bu, bir zaman serisinin değişkenliğinin değişen değişkenlik derecesini belirleyerek, ani ve belirgin dalgalanmaların olup olmadığını belirlemeye yardımcı olabilir (Hyndman vd., 2015; Hyndman vd., 2020). Spike değerini hesaplamak için ilk adımda her bir veri tek tek hariç bırakılarak hata bileşeninin varyansı Denklem (13) ile hesaplanır. Diğer bir ifadeyle, her bir gözlem seriden tek tek çıkarılır ve geri kalan veri setiyle modeli yeniden tahmin ederek elde edilen hata bileşeninin varyansı hesaplanır. İkinci adımda, hata bileşenlerin varyansının (v_t) varyansı alınarak spike değeri Denklem (14) ile hesaplanır (Hyndman vd., 2015; Hyndman vd., 2020).

$$v_t = \frac{1}{n-1} * \sum_{t=1}^{n-1} (\varepsilon_t - \bar{\varepsilon})^2 \tag{13}$$

$$Spike = \sigma_{v_t}^2 = \frac{1}{n-1} * \sum_{t=1}^{n} (v_t - \bar{v})^2 \tag{14}$$

Burada ε_t *t.* zaman adımındaki hata terimini, *n X* zaman serisindeki toplam gözlem sayısını, $\bar{\varepsilon}$ hata terimlerinin ortalamasını, v_t hata terimlerinin varyansını ve $\bar{v}$ ise v_t değerlerinin ortalamasını ifade eder (Hyndman vd., 2015; Hyndman vd., 2020).

FlatSpots: Seri değerlerinin nispeten sabit kaldığı veya çok az değişiklik gösterdiği dönem sayısını ifade eden özniteliktir. Bir zaman serisinin örneklem uzayını on eşit boyutta aralıklara böler ve herhangi bir tek aralık içindeki maksimum çalışma uzunluğunu hesaplar. Bu, bir zaman serisindeki belirli bir dönem boyunca değerlerin sabit veya çok az değiştiği durumları belirlemeye yardımcı olur (Hyndman vd., 2015; Hyndman vd., 2020).

Flatspots değerinin hesaplanması için ilk olarak zaman serisi on eşit boyuttaki aralığa bölünür. N adet gözlem içeren bir X zaman serisi için her bir aralık $I_i = [a_i, b_i)$ olur. Burada $\Delta = (\max(X_t) - \min(X_t))/10$ olmak üzere $a_i = \min(X_t) + (i-1) * \Delta$ ve $b_i = \min(X_t) + i * \Delta$ formülleri ile hesaplanır. Her bir eşit boyuttaki aralık için, bu aralıktaki değerlerin ardışık olarak ne kadar süre sabit kaldığı araştırılır. Diğer bir ifadeyle, bir aralıktaki ardışık aynı değerlerin sayısı bulunur. Bu, maksimum çalışma uzunluğu olarak adlandırılabilir. On eşit boyuttaki aralıkların her biri için hesaplanan maksimum çalışma uzunluklarının en büyüğü bulunur. Bu değer flatspots değeridir (Hyndman vd., 2015; Hyndman vd., 2020).

CrossingPoints: Bir zaman serisinin ortanca çizgiyi geçme sayısını ifade eden özniteliktir. Ortanca (Medyan) çizgisi, veri setinin ortasındaki değerdir. Geçiş noktaları (Crossing_points) değerini hesaplamak için ilk adımda zaman serisinin medyanı hesaplanır. Bunun için, veri setinin değerleri küçükten büyüğe sıralanır. Eğer veri setinin boyutu tek ise, medyan doğrudan ortadaki değerdir. Eğer veri setinin boyutu çift ise, medyan ortadaki iki değerin ortalamasıdır (Hyndman vd., 2015; Hyndman vd., 2020).

Medyan çizgisini geçiş noktalarını bulmak için, her iki ardışık veri noktasının biri medyan çizgisinin altında ve diğeri üstünde olduğu noktalar sayılır. Tüm zaman serisini taranarak medyan çizgisini geçen her ardışık veri noktası çifti sayılır. Bu değer, zaman serisinin medyan çizgisini geçme sayısını verir. Matematiksel olarak, geçiş noktaları hesaplaması Denklem (15) ile ifade edilebilir (Hyndman vd., 2015; Hyndman vd., 2020):

$$Crossing_points = \sum_{t=1}^{n-1} sign(x_t - Median) * sign(x_{t+1} - Median) \quad (15)$$

Burada *n* zaman serisindeki toplam veri noktalarının sayısını, x_t *t.* zaman adımındaki veri noktasını, *median* zaman serisinin medyanını ve sign(*x*) *x* değerinin işaretini (pozitif, negatif veya sıfır) döndüren bir işaret fonksiyonudur. Bu formül, medyan çizgisini geçiş noktalarını bulmak için her ardışık veri noktası çiftinin medyan çizgisini altında veya üstünde olup olmadığını kontrol eder ve bu geçişlerin sayısını toplar. Bu şekilde, crossing_points hesaplanmış olur (Hyndman vd., 2015; Hyndman vd., 2020).

MaxLevelShift, **MaxVarShift ve MaxKlShift:** Bu üç özellik, kayan (örtüşen) pencerelere dayalı olarak bir zaman serisinin özelliklerini hesaplar. Bu üç özellik hesabı için pencere boyutu iki olarak alınmıştır. Bu üç özellik için kaydırmanın boyutu ve kaydırmanın zaman indeksi olmak üzere iki değerden oluşan bir vektör döndürülür (Hyndman vd., 2015). *MaxLevelShift*, ardışık iki pencere arasındaki en büyük ortalama kaymayı ifade eden özniteliktir. Kayan pencere zaman serisi boyunca örtüşen şekilde kaydırılarak oluşan her bir pencere için ortalama hesaplanır. Ardışık iki pencere arasındaki ortalama kayma, bu pencereler arasındaki ortalama değerlerin farkı alınarak hesaplanır. En büyük kayma ve kayma zaman indeksi kaydedilir (Hyndman vd., 2015; Hyndman vd., 2020). *MaxVarShift*, ardışık iki pencere arasındaki en büyük varyans kaymasını ifade eden özniteliktir. Her pencerenin varyansı hesaplanır. Ardışık iki pencere arasındaki varyans kayması, bu pencereler arasındaki varyans değerlerinin farkı alınarak hesaplanır. En büyük kayma ve kayma zaman indeksi kaydedilir (Hyndman vd., 2015; Hyndman vd., 2020). *MaxKlShift*, ardışık iki pencere arasındaki Kulback-Leibler sapmasındaki en büyük kaymayı ifade eden özniteliktir. Her bir pencere için olasılık dağılımı hesaplanır. Ardışık iki pencere arasındaki Kulback-Leibler ayrımı, bu pencereler arasındaki olasılık dağılımlarının farkı alınarak hesaplanır. En büyük kayma ve kayma zaman indeksi kaydedilir (Hyndman vd., 2015; Hyndman vd., 2020).

Nonlinearity: Doğrusal olmama katsayısını ifade eden özniteliktir. Doğrusal olmama katsayısı, Teräsvirta'nın doğrusal olmama testinde kullanılan istatistiğin bir modifikasyonu kullanılarak hesaplanır (Hyndman vd., 2020). Teräsvirta'nın testi $X^2 = T * \log(SSE_1/SSE_0)$ istatistiğini kullanır; burada SSE_1 ve SSE_0 sırasıyla doğrusal olmayan ve doğrusal bir otoregresyondan elde edilen hata karelerinin toplamıdır. Bu ergodik değildir, dolayısıyla bu formül $10 * X^2/T$ olarak tanımlanır ve bu, T→∞ şeklinde doğrusal olmamanın boyutunu gösteren bir değere yakınsar. Bu, seri doğrusal olmadığında büyük değerler alır ve seri doğrusal olduğunda 0 civarında değerler alır (Hyndman vd., 2020).

Hurst: Bir zaman serisinin uzun süreli hafızasının veya kalıcılığının bir ölçüsünü ifade eden özniteliktir. Adını Harold Edwin Hurst'tan alan Hurst üssü, bir zaman serisinin uzun vadeli hafızasının bir ölçüsüdür. Uzun vadeli bağımlılık sergileyen zaman serilerinin

analizinde özellikle yararlıdır. Hurst üssü (*H*), bir zaman serisinin ölçeklendirme davranışını karakterize eder ve öngörülebilirliği ve kalıcılığı hakkında fikir verir (Rose, 1996; Willinger, 1998; Wang vd., 2006).

Hurst üssünü tahmin etmek için en yaygın yöntemlerden biri ölçeklenmiş aralık analizidir. Bu yöntem, zaman serisinin çeşitli pencere boyutları için ölçeklenmiş aralığını (R/S) hesaplamayı ve bu değerleri bir güç yasası ilişkisine uydurmayı içerir. *N* uzunluğunda bir zaman serisi her biri *n* boyutlu *m* adet örtüşmeyen pencereye ayrılır ve her pencerenin varyansı Denklem (16) ile hesaplanır. Daha sonra, ölçeklenmiş aralık (*R/S*) Denklem (17) ile hesaplanır (Rose, 1996; Willinger, 1998; Wang vd., 2006):

$$Y_i = \sum_{j=1}^{n}(x_j - \bar{x}),\ i = 1, 2, \dots, m \tag{16}$$

$$\frac{R}{S} = \frac{\max(Y_i) - \min(Y_i)}{\sqrt{\sum_{i=1}^{m}(Y_j - \bar{x})}},\ i = 1, 2, \dots, m \tag{17}$$

Hurst üssünü belirlemek (*H*) için log(*R*/*S*) ile log(*n*)'nin karşılıklı grafikleri çizilir. Bu noktalara Denklem (18) olduğu gibi bir doğrusal regresyon çizgisi uydurulur ve çizginin eğimi Hurst üssünü temsil eder. Doğrusal regresyonun *K* eğimi kullanılarak Hurst üssü *H* Denklem (19) ile hesaplanır (Rose, 1996; Willinger, 1998):

$$\log(R/S) = K * \log(n) + 0{,}5 \tag{18}$$

$$H = K/2 \tag{19}$$

Hurst üssü *H*, zaman serisinin uzun vadeli belleğini aşağıdaki gibi karakterize eder (Rose, 1996; Willinger, 1998):

- $H<0.5$: Ters yönlü davranış (ortalama dönüş)
- $H=0.5$: Rastgele davranış (ilişkisiz)
- $0.5<H<1$: Sürekli davranış (eğilim veya bellek etkisi)

EK 5 Öznitelikler ve Açıklamaları

Öznitelik Grubu	No	Kaynak	No	Hedef	Açıklama
Zaman Serisi Öznitelikleri	X1	Mean_S	X27	Mean_T	Ortalama
	X2	Variance_S	X28	Variance_T	Varyans
	X3	Acf1x_S	X29	Acf1x_T	Birinci dereceden otokorelasyon
	X4	Acf1e_S	X30	Acf1e_T	Hatanın birinci dereceden otokorelasyonu
	X5	Pacf1x_S	X31	Pacf1x_T	Birinci dereceden kısmi otokorelasyon
	X6	Trend_S	X32	Trend_T	Trendin gücü
	X7	Linearity_S	X33	Linearity_T	Doğrusallığın gücü
	X8	Curvature_S	X34	Curvature_T	Eğrilik gücü
	X9	SeasonalStrength_S	X35	SeasonalStrength_T	Mevsimsel zaman serileri için mevsimsel güç
	X10	Peak_S	X36	Peak_T	Mevsimsel bileşendeki zirvelerin boyutu
	X11	Trough_S	X37	Trough_T	Mevsimsel bileşendeki çukurların boyutu
	X12	Entropy_S	X38	Entropy_T	Bir zaman serisinin tahmin edilebilirliğinin ölçüsü
	X13	Lumpiness_S	X39	Lumpiness_T	Hatada değişen varyans
	X14	Spike_S	X40	Spike_T	Bir zaman serisinin sivriliğinin (ani artış) ölçüsü
	X15	FlatSpots_S	X41	FlatSpots_T	Seri değerlerinin nispeten sabit kaldığı veya çok az değişiklik gösterdiği dönem sayısı
	X16	CrossingPoints_S	X42	CrossingPoints_T	Bir zaman serisinin ortanca çizgiyi geçme sayısı
	X17	MaxLevelShift_S	X43	MaxLevelShift_T	Ardışık iki pencere arasındaki en büyük ortalama kayma
	X18	MaxVarShift_S	X44	MaxVarShift_T	Ardışık iki pencere arasındaki en büyük varyans kayması
	X19	MaxKlShift_S	X45	MaxKlShift_T	Ardışık iki pencere arasındaki Kulback-Leibler sapmasındaki en büyük kayma
	X20	Nonlinearity_S	X46	Nonlinearity_T	Doğrusal olmama katsayısı
	X21	Stability_S	X47	Stability_T	Hatada ortalamaların varyansı (zaman serisi kararlılığının ölçüsü)
	X22	Hurst_S	X48	Hurst_T	Bir zaman serisinin uzun süreli hafızasının veya kalıcılığının bir ölçüsü
Satışa Ait Öznitelikler	X23	SeriesSize_S	X49	SeriesSize_T	Satış zaman serisinin büyüklüğü
	X24	SalesStartYear_S	X50	SalesStartYear_T	Satış zaman serisinin başlangıç yılı
	X25	ProductGroup_S	X51	ProductGroup_T	Satış verisinin tek ürün grubu (0) /çoklu ürün grubu (1) içermesi
	X26	ZeroSalesPer_S	X52	ZeroSalesPer_T	Satış seyrekliği/sıfır satış yüzdesi
	X53	Sector_Info			Kaynak ve hedef satış verilerinin aynı (1) veya farklı (0) sektöre ait olması
	X54	SimilarProduct			Kaynak ve hedef satış verilerinin benzer ürünlerin satışını içerip (1) içermemesi (0)
Uzaklık Tabanlı Öznitelikler	X55	ERP			Kaynak ve hedef satış zaman serisi arasındaki ERP uzaklığı
	X56	DTW			Kaynak ve hedef satış zaman serisi arasındaki DTW uzaklığı
Çıktı Değişkeni	RMSE ya da MAE cinsinden TL İyileşme Yüzdesi				

EK 6 Rastgele Orman için Deneysel Sonuçlar

Çıktı Değişkeni	Model No	Derinlik	Ağaç Sayısı	R^2	Torba Dışı Hata	RMSE	MAE
RMSE	1	4	100	0,79	0,51	3,34	2,35
	2	4	200	0,81	0,50	3,24	2,28
	3	4	300	0,81	0,50	3,24	2,24
	4	4	400	0,8	0,47	3,31	2,26
	5	4	500	0,8	0,44	3,26	2,23
	6	4	700	0,81	0,40	3,22	2,23
	7	4	900	0,8	0,38	3,26	2,25
	8	4	1000	0,81	0,38	3,22	2,26
	9	5	100	0,85	0,47	2,84	1,9
	10	5	200	0,85	0,43	2,82	1,9
	11	5	300	0,85	0,36	2,82	1,86
	12	5	400	0,85	0,35	2,84	1,88
	13	5	500	0,81	0,32	3,17	1,99
	14*	5	700	0,86	**0,30**	2,76	1,86
	15	5	900	0,86	0,31	2,79	1,87
	16	5	1000	0,85	0,32	2,83	1,86
	17	6	100	0,89	0,45	2,46	1,61
	18	6	200	0,87	0,46	2,65	1,66
	19	6	300	0,87	0,45	2,65	1,62
	20	6	400	0,89	0,45	2,5	1,61
	21	6	500	0,88	0,45	2,53	1,61
	22	6	700	0,88	0,44	2,59	1,61
	23	6	900	0,88	0,43	2,57	1,6
	24	6	1000	0,88	0,43	2,52	1,6
MAE	25	4	100	0,8	0,50	3,67	2,54
	26	4	200	0,8	0,49	3,66	2,61
	27	4	300	0,8	0,48	3,65	2,51
	28	4	400	0,8	0,44	3,64	2,52
	29	4	500	0,8	0,40	3,67	2,56
	30	4	700	0,79	0,38	3,74	2,57
	31	4	900	0,79	0,38	3,75	2,58
	32	4	1000	0,79	0,47	3,72	2,55
	33	5	100	0,85	0,43	3,12	2,09
	34	5	200	0,85	0,36	3,15	2,14
	35	5	300	0,85	0,35	3,12	2,09
	36	5	400	0,85	0,35	3,17	2,17
	37	5	500	0,85	0,33	3,19	2,13
	38*	5	700	0,85	**0,29**	3,14	2,14
	39	5	900	0,85	0,31	3,16	2,14
	40	5	1000	0,85	0,33	3,13	2,13
	41	6	100	0,87	0,47	2,88	1,89
	42	6	200	0,87	0,47	2,88	1,83
	43	6	300	0,87	0,44	2,92	1,84
	44	6	400	0,87	0,45	2,89	1,86
	45	6	500	0,87	0,42	2,96	1,87
	46	6	700	0,88	0,42	2,84	1,87
	47	6	900	0,87	0,42	2,93	1,84
	48	6	1000	0,88	0,43	2,86	1,85

* Yüksek R^2 değeri ve en düşük torba dışı hataya sahip seçilen rastgele orman modelleri

EK 7 Seçilen Öznitelikler ve Önem Dereceleri ile Oluşturulmuş Ağırlıklı Öklid Uzaklıkları Matrisi

Seri	Seri_1	Seri_2	Seri_4	Seri_5	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26
Seri_1	0,00	0,06	0,29	0,31	0,43	0,38	0,47	0,44	0,35	0,28	0,39
Seri_2	0,06	0,00	0,27	0,30	0,43	0,37	0,46	0,44	0,35	0,27	0,39
Seri_4	0,29	0,27	0,00	0,13	0,37	0,34	0,44	0,35	0,30	0,25	0,38
Seri_5	0,31	0,30	0,13	0,00	0,40	0,37	0,46	0,38	0,32	0,27	0,39
Seri_14	0,43	0,43	0,37	0,40	0,00	0,49	0,44	0,41	0,45	0,40	0,48
Seri_15	0,38	0,37	0,34	0,37	0,49	0,00	0,32	0,29	0,17	0,33	0,22
Seri_16	0,47	0,46	0,44	0,46	0,44	0,32	0,00	0,36	0,32	0,44	0,37
Seri_17	0,44	0,44	0,35	0,38	0,41	0,29	0,36	0,00	0,20	0,30	0,28
Seri_18	0,35	0,35	0,30	0,32	0,45	0,17	0,32	0,20	0,00	0,27	0,21
Seri_25	0,28	0,27	0,25	0,27	0,40	0,33	0,44	0,30	0,27	0,00	0,24
Seri_26	0,39	0,39	0,38	0,39	0,48	0,22	0,37	0,28	0,21	0,24	0,00

EK 8 Tek Değişkenli Satış Zaman Serileri Arasındaki DTW Uzaklıkları

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0	3432421	13853392	40382394	40381348	40380385	42169304	153147259	51598593	61265517	61266708	22288071	22288077	21533594
Seri_2	3432421	0	18372032	50507181	50506134	50505176	52789415	130664429	65461988	76713446	76714639	27746389	27746396	26795016
Seri_3	13853392	18372032	0	10424428	10423384	10422421	10778357	197709520	10777791	15806250	15807440	5770407	5770413	5574730
Seri_4	40382394	50507181	10424428	0	888	1314	187716	214100383	3478610	1041	2341	2552	2602	1448
Seri_5	40381348	50506134	10423384	888	0	883	186745	214099281	3475138	1583	3463	3649	3697	1275
Seri_6	40380385	50505176	10422421	1314	883	0	185545	214098347	3473149	1897	4438	4616	4663	1212
Seri_14	42169304	52789415	10778357	187716	186745	185545	0	213612592	2735031	189999	191286	191476	191501	178560
Seri_15	153147259	130664429	197709520	214100383	214099281	214098347	213612592	0	209486468	214128020	214128700	214102992	214103035	214089650
Seri_16	51598593	65461988	10777791	3478610	3475138	3473149	2735031	209486468	0	3480296	3485651	3485998	3486069	3462808
Seri_17	61265517	76713446	15806250	1041	1583	1897	189999	214128020	3480296	0	1026	1355	1431	1996
Seri_18	61266708	76714639	15807440	2341	3463	4438	191286	214128700	3485651	1026	0	93	106	2981
Seri_25	22288071	27746389	5770407	2552	3649	4616	191476	214102992	3485998	1355	93	0	13	2983
Seri_26	22288077	27746396	5770413	2602	3697	4663	191501	214103035	3486069	1431	106	13	0	2989
Seri_27	21533594	26795016	5574730	1448	1275	1212	178560	214089650	3462808	1996	2981	2983	2989	0

EK 9 Tek Değişkenli Satış Zaman Serileri Arasındaki ERP Uzaklıkları

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0	52979632	164816074	222440023	222456330	222472900	223554970	5650777814	332568090	222446626	222408523	222402477	222402573	222375742
Seri_2	52979632	0	217795706	275419655	275435962	275452532	276534602	5597798182	385547722	275426258	275388155	275382109	275382205	275355374
Seri_3	164816074	217795706	0	57623949	57640256	57656826	58738896	5815593888	167752016	57630552	57592449	57586403	57586499	57559668
Seri_4	222440023	275419655	57623949	0	31427	59939	4383769	5873105431	141403753	60971	73020	74858	74956	64791
Seri_5	222456330	275435962	57640256	31427	0	29782	4352596	5873074258	141372580	91494	104190	106031	106129	83738
Seri_6	222472900	275452532	57656826	59939	29782	0	4323832	5873045494	141343816	120129	132954	134795	134893	105442
Seri_14	223554970	276534602	58738896	4383769	4352596	4323832	0	5868730100	137019984	4439048	4456090	4458627	4458725	4421088
Seri_15	5650777814	5597798182	5815593888	5873105431	5873074258	5873045494	5868730100	0	5842576780	5873161965	5873177967	5873180289	5873180387	5873142750
Seri_16	332568090	385547722	167752016	141403753	141372580	141343816	137019984	5842576780	0	141426366	141471784	141478611	141478709	141441072
Seri_17	222446626	275426258	57630552	60971	91494	120129	4439048	5873161965	141426366	0	54801	62671	62769	74993
Seri_18	222408523	275388155	57592449	73020	104190	132954	4456090	5873177967	141471784	54801	0	7871	7969	42673
Seri_25	222402477	275382109	57586403	74858	106031	134795	4458627	5873180289	141478611	62671	7871	0	128	37539
Seri_26	222402573	275382205	57586499	74956	106129	134893	4458725	5873180387	141478709	62769	7969	128	0	37637
Seri_27	222375742	275355374	57559668	64791	83738	105442	4421088	5873142750	141441072	74993	42673	37539	37637	0

EK 10 Normalize Edilmiş DTW Uzaklıkları

Seri	Seri_1	Seri_2	Seri_4	Seri_5	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26
Seri_1	0,00	0,02	0,19	0,19	0,20	0,72	0,24	0,29	0,29	0,10	0,10
Seri_2	0,02	0,00	0,24	0,24	0,25	0,61	0,31	0,36	0,36	0,13	0,13
Seri_4	0,19	0,24	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_5	0,19	0,24	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_14	0,20	0,25	0,00	0,00	0,00	1,00	0,01	0,00	0,00	0,00	0,00
Seri_15	0,72	0,61	1,00	1,00	1,00	0,00	0,98	1,00	1,00	1,00	1,00
Seri_16	0,24	0,31	0,02	0,02	0,01	0,98	0,00	0,02	0,02	0,02	0,02
Seri_17	0,29	0,36	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_18	0,29	0,36	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_25	0,10	0,13	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_26	0,10	0,13	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00

EK 11 Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş DTW Uzaklıkları ($w_{DTW} = 0,26$)

Seri	Seri_1	Seri_2	Seri_4	Seri_5	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26
Seri_1	0,00	0,00	0,05	0,05	0,05	0,18	0,06	0,07	0,07	0,03	0,03
Seri_2	0,00	0,00	0,06	0,06	0,06	0,16	0,08	0,09	0,09	0,03	0,03
Seri_4	0,05	0,06	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00
Seri_5	0,05	0,06	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00
Seri_14	0,05	0,06	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00
Seri_15	0,18	0,16	0,26	0,26	0,26	0,00	0,25	0,26	0,26	0,26	0,26
Seri_16	0,06	0,08	0,00	0,00	0,00	0,25	0,00	0,00	0,00	0,00	0,00
Seri_17	0,07	0,09	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00
Seri_18	0,07	0,09	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00
Seri_25	0,03	0,03	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00
Seri_26	0,03	0,03	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00

EK 12 Normalize Edilmiş ERP Uzaklıkları

Seri	Seri_1	Seri_2	Seri_4	Seri_5	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26
Seri_1	0,00	0,01	0,04	0,04	0,04	0,96	0,06	0,04	0,04	0,04	0,04
Seri_2	0,01	0,00	0,05	0,05	0,05	0,95	0,07	0,05	0,05	0,05	0,05
Seri_4	0,04	0,05	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_5	0,04	0,05	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_14	0,04	0,05	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_15	0,96	0,95	1,00	1,00	1,00	0,00	0,99	1,00	1,00	1,00	1,00
Seri_16	0,06	0,07	0,02	0,02	0,02	0,99	0,00	0,02	0,02	0,02	0,02
Seri_17	0,04	0,05	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_18	0,04	0,05	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_25	0,04	0,05	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00
Seri_26	0,04	0,05	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00

EK 13 Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş ERP Uzaklıkları ($w_{ERP} = 0,23$)

Seri	Seri_1	Seri_2	Seri_4	Seri_5	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26
Seri_1	0,00	0,00	0,01	0,01	0,01	0,22	0,01	0,01	0,01	0,01	0,01
Seri_2	0,00	0,00	0,01	0,01	0,01	0,22	0,02	0,01	0,01	0,01	0,01
Seri_4	0,01	0,01	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00
Seri_5	0,01	0,01	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00
Seri_14	0,01	0,01	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00
Seri_15	0,22	0,22	0,23	0,23	0,23	0,00	0,23	0,23	0,23	0,23	0,23
Seri_16	0,01	0,02	0,01	0,01	0,01	0,23	0,00	0,01	0,01	0,01	0,01
Seri_17	0,01	0,01	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00
Seri_18	0,01	0,01	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00
Seri_25	0,01	0,01	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00
Seri_26	0,01	0,01	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00

EK 14 Tek Değişkenli Zaman Serileri için Tek Kaynaklı TL'deki RMSE Yüzde İyileşme Oranları

Kaynak/Hedef	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0	2,12	3,7	8,11	1,02	5,47	0,08	0,11	11,7	7,12	3,13	4,87	-3,77	0,21
Seri_2	10,92	0	4,7	6,53	0,28	4,13	0,17	0,07	13,30	5,20	0,06	6,56	-7,36	0,19
Seri_3	10,9	12,3	0	0,494	0,75	3,51	0,07	0,09	11,93	6,66	5,13	1,67	-0,087	1,5
Seri_4	18,45	15,26	12,13	0	22,07	27,86	2,59	0,03	-2,25	4,28	0,77	-0,03	-1,77	15,26
Seri_5	7,29	10,27	9,26	18,81	0	31,73	1,42	0,28	0,00	3,69	1,27	3,41	-2,31	12,48
Seri_6	5,32	4,49	3,8	22,15	23,13	0	1,67	0,33	0,2	2,27	1,35	-0,02	-1,23	16,3
Seri_14	4,30	2,85	4,59	8,84	12,32	7,45	0	0,58	0,01	4,36	-0,16	2,25	0,04	2,12
Seri_15	0,26	1,22	1,16	-0,02	-4,02	-0,001	0,92	0,00	0,00	-1,49	-3,40	-0,07	-2,63	0,27
Seri_16	0,90	2,30	2,1	9,69	7,75	8,11	9,26	1,16	0,00	0,00	-5,13	-1,05	-7,02	0,88
Seri_17	-4,33	-3,45	-2,3	11,65	17,25	15,26	0,45	0,08	0,10	0,00	7,83	0,55	-0,40	5,01
Seri_18	-1,75	-5,82	-5,11	22,92	13,12	15,96	0,37	1,55	0,02	6,26	0,00	3,85	-1,47	4,77
Seri_25	6,84	1,05	1,23	11,36	14,15	12,12	0,00	-4,33	0,00	2,71	0,00	0,00	44,55	5,33
Seri_26	4,99	0,07	2,15	22,21	14,07	15,23	-0,02	-0,02	0,00	5,98	-0,22	2,90	0,00	4,71
Seri_27	-0,12	-0,15	0,002	9,87	11,7	17,11	1,36	1,17	1,47	5,77	5,43	6,71	4,98	0,00
Negatif Transfer Öğrenme Yaşanan Deney Sayısı:														37
Pozitif Transfer Öğrenme Yaşanan Deney Sayısı:														145
Toplam Deney Sayısı:														182
Negatif Transfer Öğrenme Oranı:														0,20

EK 15 Pharma-Seri_18 Hedef Verisi için Learnx-Seri_4 Kaynak Veri Ağırlıkları

Kaynak Veri	Kaynak Model Sayısı	Eğitim Hatası	Oran	Model Sayısı Bazında Ağırlıklar															
				3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
LearnX-Seri_4	1	2,80	0,07	0,68	0,19	0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01
	2	2,50	0,17		0,47	0,17	0,12	0,10	0,08	0,07	0,06	0,05	0,04	0,04	0,03	0,03	0,03	0,02	0,02
	3	2,30	0,24			0,24	0,17	0,13	0,11	0,09	0,08	0,07	0,06	0,05	0,04	0,04	0,04	0,03	0,03
	4	2,78	0,08				0,06	0,04	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01
	5	2,63	0,13					0,07	0,06	0,05	0,04	0,04	0,03	0,03	0,02	0,02	0,02	0,02	0,02
	6	2,33	0,23						0,10	0,09	0,08	0,07	0,06	0,05	0,04	0,04	0,03	0,03	0,03
	7	2,55	0,15							0,06	0,05	0,04	0,04	0,03	0,03	0,03	0,02	0,02	0,02
	8	2,80	0,07								0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01
	9	2,30	0,24									0,07	0,06	0,05	0,04	0,04	0,04	0,03	0,03
	10	1,89	0,37										0,09	0,08	0,07	0,06	0,06	0,05	0,05
	11	1,96	0,35											0,07	0,07	0,06	0,05	0,05	0,04
	12	2,50	0,17												0,03	0,03	0,03	0,02	0,02
	13	2,40	0,20													0,03	0,03	0,03	0,03
	14	2,20	0,27														0,04	0,04	0,03
	15	2,10	0,30															0,04	0,04
	16	2,10	0,30																0,04

EK 16 Pharma-Seri_18 Hedef Verisi için Learnx-Seri_5 Kaynak Veri Ağırlıkları

Kaynak Veri	Kaynak Model Sayısı	Eğitim Hatası	Oran	Model Sayısı Bazında Ağırlıklar															
				3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
LearnX-Seri_5	1	2,91	0,03	0,32	0,09	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00
	2	2,86	0,05		0,14	0,05	0,04	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01
	3	2,80	0,07			0,07	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01
	4	2,79	0,07				0,05	0,04	0,03	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01
	5	2,77	0,08					0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01
	6	2,66	0,12						0,05	0,05	0,04	0,03	0,03	0,02	0,02	0,02	0,02	0,02	0,01
	7	2,59	0,14							0,05	0,05	0,04	0,03	0,03	0,03	0,02	0,02	0,02	0,02
	8	2,52	0,16								0,06	0,05	0,04	0,03	0,03	0,03	0,02	0,02	0,02
	9	2,47	0,18									0,05	0,04	0,04	0,03	0,03	0,03	0,02	0,02
	10	2,41	0,20										0,05	0,04	0,04	0,03	0,03	0,03	0,03
	11	2,35	0,22											0,05	0,04	0,04	0,03	0,03	0,03
	12	2,31	0,23												0,04	0,04	0,04	0,03	0,03
	13	2,38	0,21													0,04	0,03	0,03	0,03
	14	2,44	0,19														0,03	0,03	0,02
	15	2,50	0,17															0,02	0,02
	16	2,81	0,07																0,01

EK 17 Pharma-Seri_18 Hedef Verisi için Pharma-Seri_17 Kaynak Veri Ağırlıkları

Kaynak Veri	Kaynak Model Sayısı	Eğitim Hatası	Oran	Model Sayısı Bazında Ağırlıklar															
				3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48
Pharma-Seri_17	1	3,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	2	2,90	0,04		0,10	0,04	0,03	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00
	3	2,10	0,30			0,31	0,22	0,17	0,14	0,12	0,11	0,09	0,07	0,06	0,06	0,05	0,05	0,04	0,04
	4	2,20	0,27				0,19	0,15	0,12	0,10	0,09	0,08	0,07	0,06	0,05	0,05	0,04	0,04	0,03
	5	2,50	0,17					0,10	0,08	0,07	0,06	0,05	0,04	0,04	0,03	0,03	0,03	0,02	0,02
	6	2,80	0,07						0,03	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01
	7	2,70	0,10							0,04	0,04	0,03	0,03	0,02	0,02	0,02	0,02	0,01	0,01
	8	2,80	0,07								0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01
	9	2,60	0,14									0,04	0,03	0,03	0,03	0,02	0,02	0,02	0,02
	10	2,70	0,10										0,03	0,02	0,02	0,02	0,02	0,01	0,01
	11	2,80	0,07											0,01	0,01	0,01	0,01	0,01	0,01
	12	2,40	0,20												0,04	0,03	0,03	0,03	0,03
	13	2,60	0,14													0,02	0,02	0,02	0,02
	14	2,40	0,20														0,03	0,03	0,03
	15	2,30	0,24															0,03	0,03
	16	2,40	0,20																0,03

EK 18 Pharma-Seri_18 Hedef Verisi için Yöntemlerin Performanslarının Karşılaştırılması

Ölçüt	MS	LSTM	LSTM-TL								
			Tek Kaynaklı TL		Çok Kaynaklı TL						
					KNN Tabanlı				Kümeleme Tabanlı		Ağırlıklandırılmış Kümeleme Tabanlı
					k=2		k=3		k=3, k=4		k=3, k=4
			Seri_17		Seri_17-26		Seri_17-26-25		Seri_4-5-17		Seri_4-5-17
		Torbalama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama	Yığınlama	Torbalama
RMSE	2/3	3,01	2,61	2,65	3,02	3,02	3,01	3,00	2,73	2,75	2,60
MAE		2,31	2,01	2,04	2,32	2,32	2,31	2,31	2,10	2,11	2,00
Süre		370,48	527,78	527,87	366,59	366,68	470,23	470,33	489,38	489,48	489,41
RMSE	12	2,99	2,66	2,64	2,98	2,98	2,88	2,89	2,57	2,60	2,36
MAE		2,29	2,04	2,03	2,29	2,29	2,21	2,22	1,97	2,00	1,81
Süre		2 222,90	3 166,70	3 166,96	2 199,56	2 199,82	1 880,94	1 881,20	1 957,51	1 957,77	1 957,65
RMSE	24	2,94	2,65	2,62	2,91	2,87	2,80	2,75	2,55	2,54	2,42
MAE		2,23	2,01	1,99	2,21	2,18	2,13	2,09	1,94	1,93	1,84
Süre		4 445,81	6 333,39	6 334,12	4 399,11	4 399,84	3 761,88	3 762,61	3 915,02	3 915,74	3 915,19
RMSE	36	2,9	2,60	2,59	2,87	2,82	2,70	2,69	2,47	2,45	2,32
MAE		2,2	1,96	1,95	2,16	2,13	2,03	2,03	1,86	1,85	1,75
Süre		6 668,71	9 500,09	9 501,18	6 598,67	6 599,76	5 642,82	5 643,91	5 872,52	5 873,61	5 872,71
RMSE	48	2,88	2,57	2,55	2,86	2,79	2,70	2,66	2,41 **(3)**	2,38 **(2)**	2,26 **(1)**
MAE		2,18	1,93	1,91	2,15	2,09	2,03	2,00	1,81 **(3)**	1,79 **(2)**	1,70 **(1)**
Süre		8 891,62	12 666,79	12 668,24	8 798,22	8 799,68	7 523,76 **(1)**	7 525,21 **(2)**	7 830,03 **(3)**	7 831,49	7 830,24

- MS (Model Sayısı) topluluktaki toplam model sayısını (N*p adet) ifade etmektedir.
- İlgili yöntemin başarı sırası (.) ile gösterilmiştir.
- Eğitim süreleri dakika olarak verilmiştir.

EK 19 Tüm Satış Verileri için Seçilen Özniteliklerle Önem Dereceleri ile Oluşturulmuş Ağırlıklı Öklid Uzaklıkları Matrisi

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0,00	0,06	0,09	0,29	0,31	0,26	0,42	0,36	0,46	0,42	0,34	0,27	0,38	0,32
Seri_2	0,06	0,00	0,09	0,27	0,30	0,24	0,42	0,36	0,45	0,42	0,34	0,27	0,37	0,32
Seri_3	0,09	0,09	0,00	0,28	0,31	0,26	0,37	0,39	0,46	0,42	0,37	0,28	0,40	0,32
Seri_4	0,29	0,09	0,28	0,00	0,13	0,07	0,36	0,33	0,43	0,34	0,29	0,25	0,36	0,40
Seri_5	0,31	0,27	0,31	0,13	0,00	0,14	0,40	0,35	0,45	0,36	0,31	0,27	0,38	0,42
Seri_6	0,26	0,24	0,26	0,07	0,14	0,00	0,36	0,31	0,40	0,34	0,26	0,23	0,33	0,37
Seri_14	0,42	0,30	0,37	0,36	0,40	0,36	0,00	0,49	0,44	0,40	0,44	0,40	0,48	0,44
Seri_15	0,36	0,24	0,39	0,33	0,35	0,31	0,49	0,00	0,32	0,29	0,17	0,32	0,22	0,27
Seri_16	0,46	0,42	0,46	0,43	0,45	0,40	0,44	0,32	0,00	0,35	0,31	0,43	0,37	0,37
Seri_17	0,42	0,36	0,42	0,34	0,36	0,34	0,40	0,29	0,35	0,00	0,19	0,29	0,28	0,32
Seri_18	0,34	0,45	0,37	0,29	0,31	0,26	0,44	0,17	0,31	0,19	0,00	0,26	0,21	0,29
Seri_25	0,27	0,42	0,28	0,25	0,27	0,23	0,40	0,32	0,43	0,29	0,26	0,00	0,23	0,34
Seri_26	0,38	0,34	0,40	0,36	0,38	0,33	0,48	0,22	0,37	0,28	0,21	0,23	0,00	0,26
Seri_27	0,32	0,32	0,32	0,40	0,42	0,37	0,44	0,27	0,37	0,32	0,29	0,34	0,26	0,00

EK 20 Tüm Satış Verileri için Normalize Edilmiş DTW Uzaklıkları

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0,00	0,01	0,06	0,19	0,19	0,19	0,20	0,72	0,24	0,29	0,29	0,10	0,10	0,10
Seri_2	0,02	0,00	0,09	0,24	0,24	0,24	0,25	0,61	0,31	0,36	0,36	0,13	0,13	0,13
Seri_3	0,06	0,09	0,00	0,05	0,05	0,05	0,05	0,92	0,05	0,07	0,07	0,03	0,03	0,03
Seri_4	0,19	0,24	0,05	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_5	0,19	0,24	0,05	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_6	0,19	0,24	0,05	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_14	0,20	0,25	0,05	0,00	0,00	0,00	0,00	1,00	0,01	0,00	0,00	0,00	0,00	0,00
Seri_15	0,72	0,61	0,92	1,00	1,00	1,00	1,00	0,00	0,98	1,00	1,00	1,00	1,00	1,00
Seri_16	0,24	0,31	0,05	0,02	0,02	0,02	0,01	0,98	0,00	0,02	0,02	0,02	0,02	0,02
Seri_17	0,29	0,36	0,07	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_18	0,29	0,36	0,07	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_25	0,10	0,13	0,03	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_26	0,10	0,13	0,03	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_27	0,10	0,13	0,03	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00

EK 21 Tüm Satış Verileri için Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş DTW Uzaklıkları ($w_{DTW} = 0,26$)

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0,00	0,00	0,02	0,05	0,05	0,05	0,05	0,18	0,06	0,07	0,07	0,03	0,03	0,03
Seri_2	0,00	0,00	0,02	0,06	0,06	0,06	0,06	0,16	0,08	0,09	0,09	0,03	0,03	0,03
Seri_3	0,02	0,02	0,00	0,01	0,01	0,01	0,01	0,24	0,01	0,02	0,02	0,01	0,01	0,01
Seri_4	0,05	0,06	0,01	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_5	0,05	0,06	0,01	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_6	0,05	0,06	0,01	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_14	0,05	0,06	0,01	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_15	0,18	0,16	0,24	0,26	0,26	0,26	0,26	0,00	0,25	0,26	0,26	0,26	0,26	0,26
Seri_16	0,06	0,08	0,01	0,00	0,00	0,00	0,00	0,25	0,00	0,00	0,00	0,00	0,00	0,00
Seri_17	0,07	0,09	0,02	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_18	0,07	0,09	0,02	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_25	0,03	0,03	0,01	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_26	0,03	0,03	0,01	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00
Seri_27	0,03	0,03	0,01	0,00	0,00	0,00	0,00	0,26	0,00	0,00	0,00	0,00	0,00	0,00

EK 22 Tüm Satış Verileri için Normalize Edilmiş ERP Uzaklıkları

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0,00	0,01	0,03	0,04	0,04	0,04	0,04	0,96	0,06	0,04	0,04	0,04	0,04	0,04
Seri_2	0,01	0,00	0,04	0,05	0,05	0,05	0,05	0,95	0,07	0,05	0,05	0,05	0,05	0,05
Seri_3	0,03	0,04	0,00	0,01	0,01	0,01	0,01	0,99	0,03	0,01	0,01	0,01	0,01	0,01
Seri_4	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_5	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_6	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_14	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_15	0,96	0,95	0,99	1,00	1,00	1,00	1,00	0,00	0,99	1,00	1,00	1,00	1,00	1,00
Seri_16	0,06	0,07	0,03	0,02	0,02	0,02	0,02	0,99	0,00	0,02	0,02	0,02	0,02	0,02
Seri_17	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_18	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_25	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_26	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00
Seri_27	0,04	0,05	0,01	0,00	0,00	0,00	0,00	1,00	0,02	0,00	0,00	0,00	0,00	0,00

EK 23 Tüm Satış Verileri için Önem Derecesi ile Ağırlıklandırılmış ve Normalize Edilmiş ERP Uzaklıkları ($w_{ERP} = 0,23$)

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,22	0,01	0,01	0,01	0,01	0,01	0,01
Seri_2	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,22	0,02	0,01	0,01	0,01	0,01	0,01
Seri_3	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_4	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_5	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_6	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_14	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_15	0,22	0,22	0,23	0,23	0,23	0,23	0,23	0,00	0,23	0,23	0,23	0,23	0,23	0,23
Seri_16	0,01	0,02	0,01	0,01	0,01	0,01	0,01	0,23	0,00	0,01	0,01	0,01	0,01	0,01
Seri_17	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_18	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_25	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_26	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00
Seri_27	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,23	0,01	0,00	0,00	0,00	0,00	0,00

EK 24 Çok Değişkenli Satış Zaman Serileri Arasındaki DTW_D Uzaklıkları

Seri	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0	3432421	13853391	40382394	40381348	40380385	42169304	153147259	51598592	61265517	61266708	22288071	22288077	21533594
Seri_2	3432421	0	18372032	50507181	50506134	50505176	52789415	130664429	65461988	76713446	76714639	27746389	27746396	26795016
Seri_3	13853391	18372032	0	10424428	10423384	10422421	10778357	197709520	10777791	15806250	15807440	5770407	5770413	5574730
Seri_4	40382394	50507181	10424428	0	931	1480	187719	214100383	3478610	1550	2696	2605	2646	1595
Seri_5	40381348	50506134	10423384	931	0	924	186748	214099281	3475138	2052	3753	3686	3728	1458
Seri_6	40380385	50505176	10422421	1480	924	0	185547	214098347	3473149	2314	4717	4645	4688	1354
Seri_14	42169304	52789415	10778357	187719	186748	185547	0	213612592	2735031	190004	191292	191477	191502	178562
Seri_15	153147259	130664429	197709520	214100383	214099281	214098347	213612592	0	209486468	214128020	214128700	214102992	214103035	214089650
Seri_16	51598592	65461988	10777791	3478610	3475138	3473149	2735031	209486468	0	3480296	3485651	3485998	3486069	3462808
Seri_17	61265517	76713446	15806250	1550	2052	2314	190004	214128020	3480296	0	1335	1771	1827	2518
Seri_18	61266708	76714639	15807440	2696	3753	4717	191292	214128700	3485651	1335	0	1134	1135	3391
Seri_25	22288071	27746389	5770407	2605	3686	4645	191477	214102992	3485998	1771	1134	0	22	2995
Seri_26	22288077	27746396	5770413	2646	3728	4688	191502	214103035	3486069	1827	1135	22	0	2999
Seri_27	21533594	26795016	5574730	1595	1458	1354	178562	214089650	3462808	2518	3391	2995	2999	0

EK 25 Çok Değişkenli Zaman Serileri için Tek Kaynaklı TL'deki RMSE Yüzde İyileşme Oranları

Kaynak/Hedef	Seri_1	Seri_2	Seri_3	Seri_4	Seri_5	Seri_6	Seri_14	Seri_15	Seri_16	Seri_17	Seri_18	Seri_25	Seri_26	Seri_27
Seri_1	0	12,25	5,66	9,03	3,06	6,21	1,09	0,78	13,26	8,26	4,45	2,65	-4,25	3,96
Seri_2	13,26	0	4,7	5,85	0,45	3,36	1,26	0,59	14,25	4,85	0,28	4,25	-5,59	3,88
Seri_3	11,21	11,96	0	0,52	0,96	2,65	0,11	1,03	12,63	7,71	6,18	0,08	-0,06	14,95
Seri_4	10,02	10,23	9,01	0	23,01	25,85	3,03	0,88	-3,23	5,26	1,84	-0,12	-1,48	12,36
Seri_5	4,42	4,74	5,1	19,02	0	27,89	1,77	0,75	0,36	4,12	2,12	4,21	-2,11	10,75
Seri_6	3,69	4,49	4,11	21,03	21,74	0	2,05	0,45	0,87	2,96	2,04	-0,22	-1,23	10,1
Seri_14	4,41	3,36	4,65	5,65	8,85	6,59	0	1,23	0,23	5,12	1,88	3,26	-0,01	3,26
Seri_15	1,23	1,41	3,21	-0,03	-3,98	-0,01	1,18	0	0,19	-2,85	-2,85	-0,18	-2,15	0,05
Seri_16	1,12	2,13	5,36	8,55	6,63	7,53	10,27	2,59	0	-2,26	-4,98	-0,05	-6,23	1,25
Seri_17	-5,03	-2,56	-3,13	12,03	16,86	17,27	0,06	0,05	1,25	0	13,63	0,86	0,13	4,89
Seri_18	-2,06	4,78	-4,77	20,96	14,15	16,12	0,07	0,84	0,04	12,49	0	4,12	0,02	4,44
Seri_25	2,26	2,03	1,71	10,45	12,84	11,3	-0,01	-4,09	0,10	3,03	-0,01	0	46,17	6,41
Seri_26	2,88	1,02	0,86	11,52	10,26	10,52	-0,22	-0,18	0,08	4,46	-0,09	10,11	0	4,93
Seri_27	4,27	4,01	7,57	8,82	9,48	12,12	2,59	2,36	4,45	4,89	4,68	7,12	5,56	0
Negatif Transfer Öğrenme Yaşanan Deney Sayısı:														32
Pozitif Transfer Öğrenme Yaşanan Deney Sayısı:														150
Toplam Deney Sayısı:														182
Negatif Transfer Öğrenme Oranı:														0,18