



**DERİN TEKRARLAYAN SİNİR AĞI YÖNTEMLERİ
İLE ERZURUM BİYOLOJİK ATIKSU ARITMA
TESİSİNİN PERFORMANS TAHMİNİ**

Sırrı BALLI

Danışman: Dr. Öğr. Üyesi Ahmet COŞKUNÇAY

Yüksek Lisans Tezi

Bilgisayar Mühendisliği Ana Bilim Dalı

2022

(Her hakkı saklıdır.)

T.C.
ATATÜRK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI

**DERİN TEKRARLAYAN SİNİR AĞI YÖNTEMLERİ İLE ERZURUM BİYOLOJİK
ATIKSU ARITMA TESİSİNİN PERFORMANS TAHMİNİ**

(Performance Forecasting of Biological Wastewater Treatment Plant using Deep Recurrent
Neural Network Methods)

YÜKSEK LİSANS TEZİ

Sırrı BALLI

Danışman: Dr. Öğr. Üyesi Ahmet COŞKUNÇAY

Erzurum
Haziran, 2022

KABUL VE ONAY TUTANAĞI

Sırrı BALLI tarafından hazırlanan “Derin Tekrarlayan Sinir Ağı Yöntemleri ile Erzurum Biyolojik Atıksu Arıtma Tesisinin Performans Tahmini” başlıklı çalışması .. / .. / 2022 tarihinde yapılan tez savunma sınavı sonucunda başarılı bulunarak jürimiz tarafından Bilgisayar Mühendisliği Ana Bilim Dalı, Bilgisayar Mühendisliği Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Jüri Başkanı:	Dr. Öğr. Üyesi Bilal USANMAZ <i>Atatürk Üniversitesi</i>	<i>Aslı Islak İmzalıdır</i>
Danışman:	Dr. Öğr. Üyesi Ahmet COŞKUNÇAY <i>Atatürk Üniversitesi</i>	<i>Aslı Islak İmzalıdır</i>
Jüri Üyesi:	Prof. Dr. Alper NUHOĞLU <i>Atatürk Üniversitesi</i>	<i>Aslı Islak İmzalıdır</i>
Jüri Üyesi:	Dr. Öğr. Üyesi Gülşah TÜMÜKLÜ ÖZYER <i>Atatürk Üniversitesi</i>	<i>Aslı Islak İmzalıdır</i>
Jüri Üyesi:	Dr. Öğr. Üyesi Serhat PEKER <i>İzmir Bakırçay Üniversitesi</i>	<i>Aslı Islak İmzalıdır</i>

Enstitü Yönetim Kurulunun
.../.../.... tarih ve sayılı
kararı.

Bu tezin Atatürk Üniversitesi Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili maddelerinde belirtilen şartları yerine getirdiğini onaylarım.

Prof. Dr. Saltuk Buğrahan CEYHUN
Enstitü Müdürü

Not: Bu tezde kullanılan özgün ve başka kaynaklardan yapılan bildiriş, çizelge, şekil ve fotoğrafların kaynak olarak kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU

Yüksek Lisans Tezi olarak Dr. Öğr. Üyesi Ahmet COŞKUNÇAY danışmanlığında sunulan “Derin Tekrarlayan Sinir Ağı Yöntemleri ile Erzurum Biyolojik Atıksu Arıtma Tesisinin Performans Tahmini” başlıklı çalışmanın tarafımızdan bilimsel etik ilkelere uyularak yazıldığını, yararlanılan eserlerin kaynakçada gösterildiğini, Fen Bilimleri Enstitüsü tarafından belirlenmiş olan Turnitin Programı benzerlik oranlarının aşılmadığını ve aşağıdaki oranlarda olduğunu beyan ederiz.

Tez Bölümleri	Tezin Benzerlik Oranı (%)	Maksimum Oran (%)
Giriş	1	30
Kuramsal Temeller	1	30
Materyal ve Yöntem	3	35
Araştırma Bulguları	1	20
Sonuçlar	5	20
Tezin Geneli	3	25

Not: Yedi kelimeye kadar benzerlikler ile Başlık, Kaynakça, İçindekiler, Teşekkür, Dizin ve Ekler kısımları tarama dışı bırakılabilir. Yukarıdaki azami benzerlik oranları yanında tek bir kaynaktan olan benzerlik oranlarının %5'den büyük olmaması gerekir.

Beyan edilen bilgilerin doğru olduğunu, aksi halde doğacak hukuki sorumlulukları kabul ve beyan ederiz.

Tez Yazarı (Öğrenci)	Tez Danışmanı
Sırrı BALLI	Dr. Öğr. Üyesi Ahmet COŞKUNÇAY
04.07.2022	04. 07.2022
İmza Aslı Islak İmzalıdır.	İmza Aslı Islak İmzalıdır.

* Tez ile ilgili YÖKTEZ’de yayınlamasına ilişkin bir engelleme var ise aşağıdaki alanı doldurunuz.

☐ Tezle ilgili patent başvurusu yapılması / patent alma sürecinin devam etmesi sebebiyle Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 2 (iki) yıl süreyle engellenmiştir.

☐ Enstitü Yönetim Kurulunun/....../.... tarih ve sayılı kararı ile teze erişim 6 (altı) ay süreyle engellenmiştir.

TEŞEKKÜR

Tez konumun belirlenmesinden, tezimin tamamlanmasına kadar yaptığım çalışmalarında kıymetli görüşleri ve yardımlarıyla bana yol gösteren, bilgi, birikim ve tecrübeleri ile bana destek olan, danışman hocam Dr. Öğr. Üyesi Sayın Ahmet COŞKUNÇAY'a teşekkürlerimi ve saygılarımı sunarım.

Tez çalışmam sırasında sağladıkları değerli katkıları ve görüşleri için Prof. Dr. Sayın Alper NUHOĞLU ve Dr. Öğr. Üyesi Sayın Bilal USANMAZ hocalarıma teşekkür ve minnetlerimi sunarım.

Tezimde kullandığım verilere ulaşmamda yardımcı olan ESKİ Genel Müdürü Sayın Remzi ERTEK'e, Arıtma Tesisleri Dairesi Başkanı Sayın M. Tayyip ÇUBUKCU'ya; değerli fikirlerini paylaşan, araştırmam için kıymetli katkılar sağlayan, mesai arkadaşlarım Sayın Selim AYDIN'a ve Sayın Yusuf Yücel ÖZONUR'a içten teşekkürlerimi sunarım.

Değerli destekleri ve motive edici katkılarından dolayı Daire Başkanım Sayın Mesut YAVİLIOĞLU ve Sayın mesai arkadaşlarıma; özveriyle destek olup, güzel niyetini, sabrını ve yakınlığını üzerimden eksik etmeyen tüm sevenlerime müteşekkirim.

Son olarak, bütün eğitim hayatım boyunca her daim yanımda olan ve tez yazım sürecimde bana sabırla destek veren ve dualarını esirgemeyen aileme sonsuz teşekkürler ederim.

Sırrı BALLI

ÖZET

YÜKSEK LİSANS TEZİ

DERİN TEKRARLAYAN SİNİR AĞI YÖNTEMLERİ İLE ERZURUM BİYOLOJİK ATIKSU ARITMA TESİSİNİN PERFORMANS TAHMİNİ

Sırrı BALLI

Danışman Dr. Öğr. Üyesi Ahmet COŞKUNÇAY

Amaç: Atıksu arıtma tesislerinde arıtılmış suda bulunan KOİ ve BOİ miktarları, üretilen suyun kalitesinin ve tesisin işletim performansının kontrolü için kullanılabilecek önemli göstergelerdendir. Deneysel yollarla KOİ ve BOİ'nin ölçümü hem zahmetli hem de zaman alan bir işlemdir. Bu çalışmada Erzurum Biyolojik Atıksu Arıtma Tesisinin (ERBASAT) günlük laboratuvar ölçüm sonuçlarını kullanarak, tesiste arıtılmış sudaki BOİ ve KOİ miktarlarının makine öğrenme yöntemleri ile tahminlemesi amaçlanmıştır. Arıtılmış sudaki BOİ ve KOİ parametrelerinin önceden tahmin edilmesi arıtma tesisinin yönetimi ve süreçlerin planlaması bakımından karar vericilere önemli faydalar kazandırıp, atık suyun etkin şekilde arıtılmasını sağlayacak ve halk sağlığı, etkin çevre yönetimi ve tesisin işletim maliyetinin düşürülmesi konularında kazanımlar sunacaktır.

Yöntem: Çalışmada, BOİ ve KOİ miktarlarının tahmini için derin öğrenme yönteminin zaman serilerinin tahminlemesi için kullanılan modellerinden RNN LSTM ve GRU modelleri kullanılmıştır. Çalışmada kullanılan modeller 2 katmanlı olarak tasarlanmış ve modellere giriş parametresi olarak atıksu arıtma tesisinin atıksuyun arıtılması sırasında tesis akredite laboratuvarında yapılan 943 günlük ölçüm sonuçları kullanılmıştır. Oluşturulan modellerin tahmin değerleri ile gerçek değerlerin uyumluluğunun olduğunun kontrol edilmesi için MAE MSE, RMSE, MAPE ve R^2 metrikleri kullanılmıştır.

Bulgular: Çalışılan modeller içerisinde BOİ parametresini MAPE metriğine göre en yüksek doğrulukta GRU modeli %79,16 başarı ile; KOİ parametresini ise MAPE metriğine göre en yüksek doğrulukta LSTM modeli %79,04 başarı ile tahmin edebilmiştir.

Sonuç: Çalışma sonunda, çalışılan RNN, LSTM ve GRU modellerin tahmin sonuçlarının tesis yönetimine yardımcı olacak doğrulukta olduğu, modellerin yüksek öğrenim hızına sahip olduğunu ve derin öğrenme modellerinin atıksu arıtma tesislerinin BOİ ve KOİ performans parametrelerinin tahmin edilmesinde kullanılabilecek alternatif bir yöntem olduğu gözlenmiştir.

Anahtar Kelimeler: Derin Öğrenme, RNN, LSTM, GRU, BOİ ve KOİ tahmini

Haziran 2022, 105 sayfa

ABSTRACT

MASTER THESIS

PERFORMANCE FORECASTING OF BIOLOGICAL WASTEWATER TREATMENT PLANT USING DEEP RECURRENT NEURAL NETWORK METHODS

SİRRİ BALLI

Supervisor: Asst. Prof. Dr. Ahmet COŞKUNÇAY

Purpose: The amounts of COD and BOD in the treated water in wastewater treatment plants are important indicators that can be used to control the quality of the water produced and the operating performance of the plant. The measurement of COD and BOD with experimental procedures is a laborious and time-consuming process. In this study, it is aimed to predict the amounts of BOD and COD in the treated water in WWTP with the help of machine learning methods by using daily laboratory measurement results of Erzurum Biological Wastewater Treatment Plant (ERBASAT). Predicting the BOD and COD parameters in the treated water in the plant will provide significant benefits to the decision makers in terms of the management of the treatment plant and the planning of the processes, and will provide effective treatment of wastewater and will offer gains in public health, effective environmental management and reducing the operating cost of the plant.

Method: In the study, RNN LSTM and GRU models, which are time series models of the deep learning method, were used to estimate the amounts of BOD and COD. The models used in the study were designed as two layers and the 943-day measurement results obtained in the accredited plant laboratory during the treatment of WWTP's wastewater were used as the input parameter to the models. MAE MSE, RMSE, MAPE and R^2 metrics were used to check the compatibility of the estimated values of the models with the actual values.

Findings: Among the studied models, the GRU model was able to predict the BOD parameter with the highest accuracy of 79,16%, and the LSTM model was able to predict the COD parameter with the highest accuracy of 79,04% according to the MAPE metric.

Results: At the end of the study, it has been observed that the estimation results of the studied RNN, LSTM and GRU models have higher learning speed, and accurate to help the plant management and those deep learning models are an alternative method that can be used to predict the BOD and COD performance parameters of wastewater treatment plants.

Keywords: Deep Learning, RNN, LSTM, GRU, BOD and COD prediction

June 2022, 105 pages

İÇİNDEKİLER

KABUL VE ONAY TUTANAĞI.....	i
ETİK BİLDİRİM VE İNTİHAL BEYAN FORMU	ii
TEŞEKKÜR	iii
ÖZET	iv
ABSTRACT	v
İÇİNDEKİLER.....	vi
SİMGELER ve KISALTMALAR DİZİNİ	viii
ŞEKİLLER DİZİNİ	xi
TABLolar DİZİNİ.....	xiv
GİRİŞ.....	1
KURAMSAL TEMELLER.....	4
Yapay Zekâ	4
Makine Öğrenmesi	4
Gözetimli öğrenim	5
Gözetimsiz öğrenim	6
Pekiştirmeli öğrenim	6
Yapay Sinir Ağları	6
YSA mimarisi	7
YSA gelişim süreci	9
Kaynak Özetleri	10
ML tahminleme çalışmaları	10
ML ile su parametrelerinin tahminlemesi çalışmaları.....	11
DL (RNN) zaman serilerinin tahminlemesi çalışmaları.....	13
DL (RNN) ile su parametrelerinin zaman serisi tahminlemesi çalışmaları	14
MATERYAL ve YÖNTEM	16
Materyal	16
Erzurum atıksu arıtma tesisi.....	16
Tesis hatlarının işleyiş özeti	16
Veri seti parametreleri.....	18
Derin öğrenme kütüphaneleri.....	18
Yöntem.....	20
Derin öğrenme.....	20

RNN(Tekrarlayan sinir ağıları)	42
Uzun kısa süreli bellek (UKSB) modeli	48
GRU	52
Veri setinin hazırlaması.....	56
ARAŞTIRMA BULGULARI	59
Veri Seti Hazırlık Aşaması	59
Eksik verilerin tamamlanması.....	59
Aykırı değerlerin düzeltilmesi.....	62
Veri normleştirme	63
Veri setinin bölünmesi	64
Veri seti çerçeve dizisinin oluşturulması	64
Modellerin Oluşturulması	65
Kullanılan metrikler	66
Tahmin Aşaması.....	66
RNN modelinin uygulanması.....	66
LSTM modelinin uygulanması	70
GRU modelinin uygulanması.....	74
Modellerin Performans özeti.....	79
Modellerin Eğitim Süreleri	80
SONUÇLAR.....	82
KAYNAKLAR.....	84
ÖZGEÇMİŞ.....	89

SİMGELER ve KISALTMALAR DİZİNİ

AdaGrad	: Adaptive Gradient
AI	: Artificial Intelligence
AKM	: Askıda Katı Madde
ALR	: Adaptive Learning Rate
ANFIS	: Adaptive Neuro-Fuzzy Inference System
ANN	: Artificial Neural Network
ANN-Q	: Artificial Neural Networks-Quantitative
ARIMA	: Autoregressive Integrated Moving Average (Bütünleşik Otoregresif Hareketli Ortalama)
ARM	: Association Rule Mining
BD	: Big Data
BDTR	: Boosted Decision Tree Regression
BLR	: Bayesian Linear Regression
BOİ	: Biyokimyasal Oksijen İhtiyacı
CEL	: Cross Entropy Loss
CNN	: Convolutional Neural Network
CRBM	: Conditional Restricted Boltzmann Machines
DFR	: Decision Forest Regression
DL	: Deep Learning
DO	: Dissolved Oxygen (Çözülmüş Oksijen)
DÖ	: DERİN Öğrenme
DQN	: Deep Q Network
DR	: Dimensionality Reduction
Dstat	: Directional Change Statistic
DT	: Decision Trees
DVM	: Destek Vektör Makinesi
ELM	: Extreme Learning Machine
ERBASAT	: Erzurum Biyolojik Atıksu Arıtma Tesisi
ESKİ	: Erzurum Su ve Kanalizasyon İdaresi
FCRBM	: Factored Conditional Restricted Boltzmann Machines
FFNN	: Feed Forward Neural Network (İleri Beslemeli Yapay Sinir Ağları)
FL	: Fuzzy Logic (Bulanık Mantık)
GD	: Gradient Descent (Gradyan İnişi)
GM	: Grey Model

GRU	: Gated Recurrent Unit
HC	: Hierarchical Conceptual
KNN	: k-Nearest Neighbors
KOİ	: Kimyasal oksijen ihtiyacı
KPCA	: Kernel Principal Component Analysis
l	: Litre
LSTM	: Long Short-Term Memory
MAE	: Mean Absolute Error(Ortalama Mutlak Hata)
MAPE	: Mean Absolute Percentage Error(Ortalama Mutlak Yüzde Hatası)
MBGD	: Mini Batch Gradient Descent (Mini Toplulaştırılmış Gradyan İnişi)
mg	: Miligram
ML	: Machine Learning (Makine Öğrenmesi)
MÖ	: Makine Öğrenmesi
MSE	: Mean Squared Error (Ortalama Kare Hatası)
NB	: Naive Bayes
Nm³/sa	: Normal m ³ /saat (hava debisi)
NMSE	: Normalized Mean Square Error
NN	: Neural Network (Yapay Sinir Ağları)
NNR	: Neural Network Regression
PCA	: Principal Component Analysis
R²	: Coefficient of Determination (Determinasyon, Korelasyon katsayısı)
RBFNN	: Radial Basis Function Neural Network
ReLU	: Rectified Linear Unit
RF	: Random Forests
RL	: Reinforcement Learning
RMSE	: Root Mean Squared Error
RMSProp	: Root Mean Square Propagation
RNN	: Recurrent Neural Network
RSE	: Residual Standard Error
SARSA	: State–Action–Reward–State–Action
SGD	: Stochastic Gradient Descent
SL	: Supervised Learning
SVD	: Singular Value Decomposition
SVM	: Support Vector Machine
SVR	: Support Vector Regression

UL : Unsupervised Learning
YSA : Yapay Sinir Ağları
 θ_c : Çamur yaşı, zaman⁻¹ (T⁻¹)



ŞEKİLLER DİZİNİ

Şekil 1. Geleneksel ve makine öğrenmesi uygulamaları girdi ve çıktıları.....	5
Şekil 2. Makine öğrenimi türleri.	5
Şekil 3. Biyolojik ve yapay sinir hücresinin (nöron) yapısı.....	7
Şekil 4. Örnek YSA mimarisi	8
Şekil 5. Yapay sinir ağlarının kilometre taşları.....	9
Şekil 6. Derin öğrenmenin tarihi ve gelişimi.	10
Şekil 7. ERBASAT’ın üstten görünümü	16
Şekil 8. Tesis hatlarının işleyiş şeması.....	17
Şekil 9. ERBASAT havalandırma ve son çökertme havuzları süreçler.....	17
Şekil 10. Örnek derin öğrenme mimarisi.	22
Şekil 11. Yapay nöron önyargı ilişkisi.	22
Şekil 12. Tek katmanlı ve tek nöronlu sinir ağı.	23
Şekil 13. Örnek veri setinin doğrusal ve doğrusal olmayan fonksiyonlarla temsil edilmesi. ..	24
Şekil 14. Aktivasyon fonksiyonları.	24
Şekil 15. İki nöronlu yapay sinir ağı.	25
Şekil 16. Tek katmanlı yapay sinir ağı.	25
Şekil 17. Çok katmanlı YSA (Derin öğrenme modeli).	26
Şekil 18. Geri yayılım için örnek yapay sinir ağı.....	29
Şekil 19. Gradyan inişi grafiği.	30
Şekil 20. Öğrenme oranının modelin eğitimine etkisi.....	31
Şekil 21. Global minimum ve lokal minimum.	31
Şekil 22. Gradyan inişine göre ağırlıkların güncellenmesi.	33
Şekil 23. Stokastik gradyan inişine göre ağırlıkların güncellenmesi.	34
Şekil 24. Mini-gruplandırılmış stokastik gradyan inişine göre ağırlıkların güncellenmesi.	34
Şekil 25. Gradyan inişi ve momentum.	35
Şekil 26. Gradyan inişi ve Nesterov momentum.....	35
Şekil 27. Örnek seyreltme işlemi.	40
Şekil 28. Erken durdurma doğruluk oranı epok ilişkisi.....	41
Şekil 29. Erken durdurma hata epok ilişkisi.....	41
Şekil 30. Tekrarlayan sinir ağlarında veri giriş ve çıkış çeşitlilikleri.....	43
Şekil 31. Tekrarlayan sinir ağlarının yapısı.....	44
Şekil 32. Derin tekrarlayan sinir ağı (Deep RNN).	44
Şekil 33. Tekrarlayan sinir ağı.	45

Şekil 34. Tekrarlayan sinir ağı ileri ve geri yayılım.....	47
Şekil 35. Tekrarlayan sinir ağlarında gizli durum ağırlıklarının güncellenmesi.....	47
Şekil 36. Uzun kısa süreli bellek modeli.....	49
Şekil 37. Uzun kısa süreli bellek hücre durumu ve kapıları.....	49
Şekil 38. Uzun kısa süreli bellek modeli ağırlıkları.....	50
Şekil 39. Kapılı tekrarlayan ünite (GRU) modeli.....	53
Şekil 40. Kapılı tekrarlayan ünite (GRU) modeli gizli durum ve kapıları.....	53
Şekil 41. Kapılı tekrarlayan ünite (GRU) modeli ağırlıkları.....	54
Şekil 42. Veri Setinin çerçeve dizisine dönüştürülmesi.....	58
Şekil 43. Ham veri seti özniteliklerinin zaman serisi grafiği.....	60
Şekil 44. Eksik öznitelikleri tamamlanan veri seti özniteliklerinin zaman serisi grafiği.....	61
Şekil 45. Ham ve eksik verileri tamamlanan özniteliklerinin özellik ilişkisi ısı haritası.....	62
Şekil 46. BOİ ve KOİ özniteliklerinin zaman serisi grafiği.....	62
Şekil 47. Aykırı değerleri düzenlenmiş BOİ ve KOİ özniteliklerinin zaman serisi grafiği.....	62
Şekil 48. BOİ ve KOİ özniteliklerinin ham ve aykırı değerleri düzenlenmiş dağılım grafiği.....	63
Şekil 49. Eğitim, doğrulama ve test veri setleri.....	64
Şekil 50. BOİ ve KOİ parametrelerinin eğitim ve test değerlerinin zaman serisi grafiği.....	64
Şekil 51. Veri setinin pencere dizisine dönüştürülmesi.....	65
Şekil 52. BOİ tahmini için 256 ünite ile oluşturulan stateless RNN modelinin tahmin ve hata grafiği.....	67
Şekil 53. BOİ tahmini için 32 ünite ile oluşturulan stateless RNN modelinin tahmin ve hata grafiği.....	67
Şekil 54. BOİ tahmini için 128 ünite ile oluşturulan stateful RNN modelinin tahmin ve hata grafiği.....	68
Şekil 55. KOİ tahmini için 256 ünite ile oluşturulan stateless RNN modelinin tahmin ve hata grafiği.....	69
Şekil 56. KOİ tahmini için 32 ünite ile oluşturulan stateful RNN modelinin tahmin ve hata grafiği.....	69
Şekil 57. KOİ tahmini için 64 ünite ile oluşturulan stateful RNN modelinin tahmin ve hata grafiği.....	70
Şekil 58. BOİ tahmini için 2048 ünite ile oluşan stateless LSTM modelinin tahmin ve hata grafiği.....	71
Şekil 59. BOİ tahmini için 2048 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.....	71

Şekil 60. BOİ tahmini için 32 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.	72
Şekil 61. KOİ tahmini için 16 ünite ile oluşan stateless LSTM modelinin tahmin ve hata grafiği.	73
Şekil 62. KOİ tahmini için 1024 ünite ile oluşan stateless LSTM modelinin tahmin ve hata grafiği.	73
Şekil 63. KOİ tahmini için 16 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.	74
Şekil 64. KOİ tahmini için 32 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.	74
Şekil 65. BOİ tahmini için 32 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.	75
Şekil 66. BOİ tahmini için 256 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.	75
Şekil 67. BOİ tahmini için 128 ünite ile oluşan stateful GRU modelinin tahmin ve hata grafiği.	76
Şekil 68. KOİ tahmini için 16 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.	77
Şekil 69. KOİ tahmini için 32 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.	77
Şekil 70. KOİ tahmini için 32 ünite ile oluşan stateful GRU modelinin tahmin ve hata grafiği.	78
Şekil 71. KOİ tahmini için 256 ünite ile oluşan stateful GRU modelinin tahmin ve hata grafiği.	78

TABLÖLAR DİZİNİ

Tablo 1. Biyolojik Sinir Yapısı ile YSA Karşılıkları.....	7
Tablo 2. Geleneksel Algoritmalar ve YSA Modeli Karşılaştırılması	8
Tablo 3. Çalışmada Kullanılan Modellerin Girdi Parametreleri.....	18
Tablo 4. Önyargı - Varyans İlişki Durumları.....	38
Tablo 5. Öğrenmenin Genelleştirilmesi.....	39
Tablo 6. Veri Setinde Bulunan Eksik Öznitelik Bilgileri.	59
Tablo 7. Veri Seti Örnek Girdi Parametreleri.	63
Tablo 8. Veri Seti Normalleştirilmiş Örnek Girdi Parametreleri.....	63
Tablo 9. Stateless RNN Modelinin BOİ Tahmin Sonuçları.	66
Tablo 10. Stateful RNN Modelinin BOİ Tahmin Sonuçları.	67
Tablo 11. Stateless RNN Modelinin KOİ Tahmin Sonuçları.	68
Tablo 12. Stateful RNN Modelinin KOİ Tahmin Sonuçları.....	69
Tablo 13. Stateless LSTM Modelinin BOİ Tahmin Sonuçları.	70
Tablo 14. Stateful LSTM Modelinin BOİ Tahmin Sonuçları.....	71
Tablo 15. Stateless LSTM Modelinin KOİ Tahmin Sonuçları.	72
Tablo 16. Stateful LSTM Modelinin KOİ Tahmin Sonuçları.....	73
Tablo 17. Stateless GRU Modelinin BOİ Tahmin Sonuçları.	75
Tablo 18. Stateful GRU Modelinin BOİ Tahmin Sonuçları.	76
Tablo 19. Stateless GRU Modelinin KOİ Tahmin Sonuçları.	76
Tablo 20. Stateful GRU Modelinin KOİ Tahmin Sonuçları.....	78
Tablo 21. Çalışılan Modellerin En İyi BOİ Tahmin Sonuçları.....	79
Tablo 22. Çalışılan Modellerin En İyi KOİ Tahmin Sonuçları.	79
Tablo 23. Literatürde Yapılan Çalışmaların BOİ Tahmin Sonuçları.....	79
Tablo 24. Literatürde Yapılan Çalışmaların KOİ Tahmin Sonuçları.	80
Tablo 25. RNN Modelinin Eğitim Süreleri	80
Tablo 26. LSTM Modelinin Eğitim Süreleri	81
Tablo 27. GRU Modelinin Eğitim Süreleri	81

GİRİŞ

Artan dünya nüfusu ve su talebi sebebi ile yaşamımız için hayati öneme sahip olan su kaynakları, içme ve kullanım alanlarında; miktar, kalite gibi birçok ciddi sorunla karşı karşıya kalmıştır (Aksungur & Firidin, 2008). Dünya Kaynakları Enstitüsü (WRI) tarafından yapılan bir araştırmaya göre; hızla artan nüfus, iklim değişikliği ve tarım ve sanayi üretiminde dünya kaynaklarına olan ihtiyacın artması gibi etkenlerin 2040 yılına kadar 33 ülkede su rezervlerinde %40-70'lik bir düşüşe sebep olması beklenmektedir (Maddocks ve diğerleri, 2015). Bununla birlikte, kullanılan su kaynaklarının arıtılmadan doğaya bırakılması, su kaynaklarının azalmasına ve çevrenin kirlenmesine sebep olmaktadır. Çevrenin temiz kalması ve su kaynaklarının daha verimli kullanılması için atık suların boşaltıldıkları ortamın ekolojik özelliklerini değiştirmeyecek hale getirilmesi gerekmektedir. Atıksu arıtma tesisleri, evsel ya da endüstriyel kullanım sonucu çıkan suların çevre kirliliğine yol açmaması ve yeniden kullanılabilmesi için, sulara kaybettikleri fiziksel kimyasal ve biyolojik özelliklerini geri kazandırmaktadır (Gücker ve diğerleri, 2006).

Atıksu arıtma tesisleri, işletim maliyetlerinin yüksek ve yapılarındaki karmaşıklıkla dolaylı olarak proses yönetimi de zor olan tesislerdir (Chen ve diğerleri, 2010). Atıksu arıtma tesislerinde kullanılan enerji ve işletim maliyetleri son yıllarda giderek yükseltmekte ve bu tesis kaynaklarının etkin kullanımının önemini artırmaktadır (Yang & Chen, 2021). Atıksu arıtma tesislerindeki işletim maliyetlerinin azaltılması ve üretim kalitesinin artırılması için tesis işletimindeki süreçlerin ve üretilen atıksuyun kontrol edilip izlenmesi gerekmektedir. Su ve atık su kalitesinin analizi için dünyada kullanılan 400'den fazla laboratuvar yöntemi bulunmaktadır (Rice ve diğerleri, 2012). Atık su kalitesinin analizinin sağlam ve doğru olması için, güvenilirliği kanıtlanmış bir yöntem ile yapılması gerekmektedir. BOİ testi, atık su arıtma tesislerinin işletim performansını kontrol etmek ve organik su kirliliğini tanımlamak için yaygın olarak kullanılan bir yöntemdir. Atıksuyun organik mukavemetini tespit edilmesi için 20 gün alınan nihai biyokimyasal oksijen ihtiyacının ve reaksiyon hız sabitinin belirlenmesi gereklidir. Ancak, genel olarak BOİ, oksijen tüketiminin 5 günde ölçülmesiyle (BOİ5) standardize edilir (Sibil ve diğerleri, 2014).

Verinin günümüzün en değerli kaynaklarından biri haline gelmesi ile verinin toplanması, analiz edilmesi, anlamlandırılması ve iş süreçlerine katılması alanlarında yapılan akademik ve sektörel çalışmalar büyük bir ivme kazanmış; son zamanda yaşanan teknolojik gelişmeler ise bu ivmeyi daha da hızlandırmıştır. Özellikle bilgisayarların artan işlem gücü, mobil cihazlar ve

nesnelerin interneti gibi teknolojik gelişmeler; makine öğrenimi ve derin öğrenmenin bu çalışmaların odağında olmasında önemli rol oynamıştır. Akademik ve sektörel alanda yapılan makine öğrenmesi çalışmaları ile teknoloji istikrarlı bir şekilde gelişmekte ve verinin karar vermede önemli olduğu akıllı uygulamalar, eğitim, sağlık, otomotiv, bankacılık, finans ve mühendislik gibi hayatın birçok alanında kullanılır hale gelmiştir.

İnsan beyin hücrelerinin öğrenme yeteneğinden ve biyolojik işleyişinden esinlenerek geliştirilen yapay sinir ağlarında daha fazla gizli katman kullanımına dayanan derin öğrenme, son yıllarda çeşitli uygulama alanlarında büyük bir başarı göstermiş ve görüntü işleme, bilgisayarla görme, tıbbi görüntüleme, makine çevirisi, konuşma tanıma, tıbbi bilgi işleme, robotik ve kontrol gibi alanlarda diğer makine öğrenimi yaklaşımlarından daha fazla kullanılır hale gelmiştir. Büyük verinin yaygınlaşması, açık kaynak kodlu kütüphanelerin yaygın hale gelmesi ve GPU gibi işlem gücü yüksek donanımların sürece katılması ise derin öğrenmeyi daha da popüler kılmıştır. Derin öğrenmenin ihtiyacı olan bu kaynakların sağlanması; modelin, finans, pazarlama, üretim ve risk tahminlemesi gibi alanlarda sıklıkla kullanılır hale gelmesini sağlamış ve model ile geliştirilen gelecek tahmini yapabilen uygulamalar analiz etme, karar verme konularında yol gösterici rol üstlenmiştir.

Derin öğrenme kullanım alanlarının artması, uygulanma yöntemlerinin geliştirilme çalışmalarını da beraberinde getirmiştir. Derin öğrenmenin zaman serileri uygulamaları için kullanılan RNN, LSTM ve GRU yöntemleri geleneksel makine öğrenme yöntemleri (Masood & Ahmad, 2021) ve istatistiksel yöntemlere (Siami-Namini ve diğerleri, 2018) göre başarılı sonuçlar vermektedir.

arıtma tesisinin performansını ve işletim kalitesini gösteren atıksu arıtma tesislerinde arıtılmış sudaki BOİ ve KOİ konsantrasyon (Yılmaz & Dogan, 2008) değerlerin tahmin edilmesi, hem doğal kaynakların korunması hem de tesis işletim maliyetlerini düşürmesi açısından (Osman Tuğrul & Egemen, 2018) önem taşımaktadır. Atıksuda bulunan BOİ ve KOİ miktarının tespiti deneysel olarak hem zor hem de zaman alan bir işlem olduğundan bu çalışmada, atıksu arıtma tesislerinde arıtılmış suyun kirlilik değerlerinin tespiti için derin öğrenmenin zaman serileri uygulamaları için kullanılan RNN, LSTM ve GRU yöntemleri kullanılarak BOİ ve KOİ miktarının tahmin edilmesi amaçlanmıştır. Çalışılan modeller stateless ve stateful olarak modellenmiştir. Tesiste arıtılmış olan suyun BOİ ve KOİ miktarının tahmin edilmesi ile arıtma tesisi proseslerinin daha iyi yönetimi sağlanacak; işletmeye zaman ve maliyet bakımından faydalar kazandıracaktır.

Bu tez çalışmasında yer alan bölümlerden,

- kaynak özetleri bölümünde; geleneksel makine öğrenmeleri ve derin öğrenme yöntemleri ile yapılan tahminleme çalışmaları hakkında yapılan literatür araştırmasına,
- kuramsal temeller bölümünde; derin öğrenme ve zaman serisi konularının temel bilgileri olan yapay zeka, makine öğrenmesi ve YSA konularına,
- materyal ve yöntem bölümünde; ERBASAT ve veri seti hakkında bilgilere ve çalışmaya konu olan ve zaman serisi derin öğrenme modelleri olan RNN, LSTM ve GRU'ya ait teknik detaylara,
- araştırma bulguları bölümünde, çalışılan modellerde kullanılan veri setinin hazırlanmasına ve modellerden elde edilen deney sonuçlarına,
- sonuç bölümünde; çalışmanın deney sonuçlarına ve özetine

yer verilmiştir.



KURAMSAL TEMELLER

Yapay Zekâ

Yapay Zekâ (AI), belirli bir görevin yürütülmesi için insan ve doğadaki diğer canlıların düşünme, karar verme ve hareket yöntemlerini analiz ederek, ihtiyaç duyulan bazı eylemleri yerine getirme kabiliyetini sağlayabilmek için geliştirilmiş, doğal zekayı taklit edebilen bilgisayar sistemleridir. Genellikle istatistiksel yöntemlerle eğitilen ve problemleri insanlar gibi çözmeye çalışan bu bilgisayar sistemleri ile zekâ gerektiren karmaşık görevler bile etkili bir şekilde çözülebilmektedir (Pirim, 2006).

Elektriğin keşfinden sonra insan hayatında önemli değişiklikler olmuş ve insanlar elektriği sanayi, taşımacılık, turizm, endüstri, tıp gibi akla gelen hemen her alanda kullanmaya başlamışlardır. Bilgisayarın icadı da elektrik gibi insanların hayatını etkilen başka bir gelişme olarak karşımıza çıkmaktadır. Bilgisayarlar günümüzde iletişim, eğitim, sağlık, finans gibi birçok alanda çok önemli bir yer edindi.

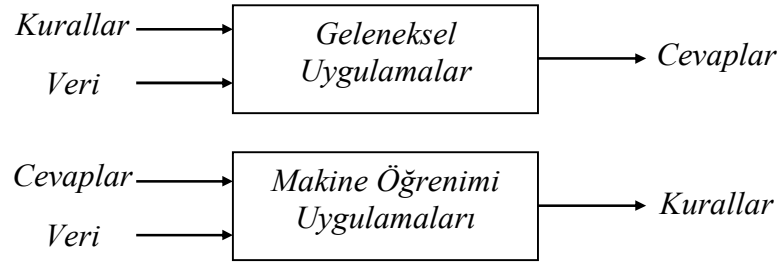
Bilgisayarların icadıyla birlikte, öğrenme, problem çözme, çevre algılama gibi bilişsel işlevleri taklit edebilen yapay zekâ sistemlerine ait çalışmalar (Turing, 2009) özellikle 1940'lı yıllardan sonra birçok teknolojik araştırma ve geliştirmenin odağı haline gelmiştir (McCorduck ve diğerleri, 1977).

Makine Öğrenmesi

Pratikte makine öğrenimi (ML) olarak tanımlanan makine öğrenmesi, matematik ve istatistik bilimlerini kullanarak, eldeki verilerden verilere bağlı sonuçlar (çıktılar) üreten, bu sonuçlar ile bilinmeyene yönelik tahminlerde bulunarak yöntemlerin geliştirilmesi sağlayan yapay zekânın bir alt dalıdır (Ray, 2019).

Geleneksel programlamada veri ve kurallar giriş olarak kullanılır ve uygulamanın istenilen sonuçları üretmesi beklenir. Makine öğrenmesi uygulamalarında ise veri ve sonuçlar uygulamaya girilir ve uygulamanın insan zihni ile zor çözülebilecek problemleri çok daha kolay hızlı ve çözebilecek, anlamlı ve verimli kurallar üretmesi istenir. Bu yüzden geleneksel programlamada uygulamanın alan uzmanlığına çok fazla ihtiyaç duyulur ve uygulama, insan müdahalesine ve hatasına karşı çok duyarlıdır. Makine öğrenmesinde ise matematik, istatistik ve bilgisayar bilimini temel alan disiplinler arası bir yaklaşım ile uygulamanın kuralları sonuçtan öğrenmesini sağlanmaktadır. Böylece bilgisayarların işlem gücünden faydalanılarak verileri öğrenme modeline dayalı en karmaşık sorunları çözme konusunda bile insan

kapasitesinin ötesine geçen uygulamalar geliştirilebilmektedir. Geleneksel ve makine öğrenmesi uygulamaları girdi ve çıktılarının gösterimi Şekil 1’de verilmiştir.



Şekil 1. Geleneksel ve makine öğrenmesi uygulamaları girdi ve çıktıları.

Otomotiv, tıp, eğlence, finans, pazarlama, savunma sistemleri gibi pek çok alanda kullanılan makine öğrenimi; sınıflandırma, kümeleme, kestirim-tahminleme, ortam öğrenme çözümleri için sıklıkla kullanılmaktadır. Makine öğrenmesi uygulamalarında genel olarak, gözetimli, gözetimsiz ve pekiştirmeli öğrenme türleri kullanılır (Mittal & Sangwan, 2019). Şekil 2’de makine öğrenimi türleri gösterilmiştir.



Şekil 2. Makine öğrenimi türleri.

Gözetimli öğrenim

Gözetimli veya Denetimli öğrenme olarak bilinen (SL) bu yöntem, girdi verisi ve bu girdilere karşılık gelen çıktılarının kullanımı ile geliştirilen makine öğrenimi türüdür. Girdiler öğrenim verisi olarak kullanılır ve bu veriler ışığında öğrenim sağlanır (Singh ve diğerleri, 2016). Öğrenme başarısı genelleme hatası ile ölçülen ve tahminleme ve sınıflandırma uygulamalarında sıklıkla kullanılan gözetimli makine öğrenme teknikleri aşağıda sıralanmıştır.

- Destek Vektör Makinaları (SVM)
- Karar Ağaçları (DT)
- En Yakın Komşuluk (KNN)
- Naive Bayes
- Doğrusal Regresyon ()
- Lojistik Regresyon (Logistic Regression)
- Yapay Sinir Ağları (ANN)

- Derin Öğrenme (DL)

Gözetimsiz öğrenim

Denetimsiz olarak ta bilinen gözetimsiz öğrenme (UL) yöntemi girdi verisi verilerek bu verilere dayalı kümeleme-yoğunluk tahmini ve ilişki-anormallik tespiti için geliştirilen makine öğrenimi türüdür (Usama ve diğerleri, 2019). Gözetimsiz öğrenmede veriler hakkında daha fazla bilgi edinmek ve verilerdeki temel yapıyı veya dağılımı modellemek amaçlanmaktadır. Gözetimli öğrenmeye göre fazla miktardaki eğitim verileri nedeniyle hesaplama karmaşıklığı olabileceği gibi, daha fazla eğitim süresi gerektiren ve yüksek hatalı sonuç riski olan bir yöntemdir. Çıktıların doğrulanması için insan müdahalesine ihtiyaç duyulmaktadır. Sıklıkla kullanılan gözetimsiz makine öğrenme teknikleri aşağıda sıralanmıştır.

- K-Ortalama Kümeleme (K-means Clustering)
- Birliktelik Kuralları (ARM)
- Boyutsal küçülme (DR)
- Tekil değer ayrışımı(SVD)
- Temel bileşen Analizi (PCA)

Pekiştirmeli öğrenim

Takviye öğrenimi olarakta bilinen pekiştirmeli öğrenim (RL), davranışçılıktan esinlenen ve bilinmeyen ortamda en yüksek ödül miktarına ulaşabilmesi için hangi işlemlerin yapılması gerektiğinin öğrenilmesi ilgili bir makine öğrenimi türüdür (François-Lavet ve diğerleri, 2018). Pekiştirmeli öğrenme yöntemi, oyun, kontrol, yöneylem araştırması, bilgi, simülasyon yöntemi ile optimizasyon, sürü zekâsı ve istatistik gibi birçok alanda kullanılmaktadır (Qiang & Zhongli, 2011). Sıklıkla kullanılan pekiştirmeli makine öğrenme teknikleri aşağıda sıralanmıştır.

- Monte Carlo
- Q-Öğrenme (Q-Learning)
- Durum – Eylem – Ödül – Durum – Eylem (SARSA)
- Derin Q Ağı (DQN)
- Boyut İndirgeme (DR)

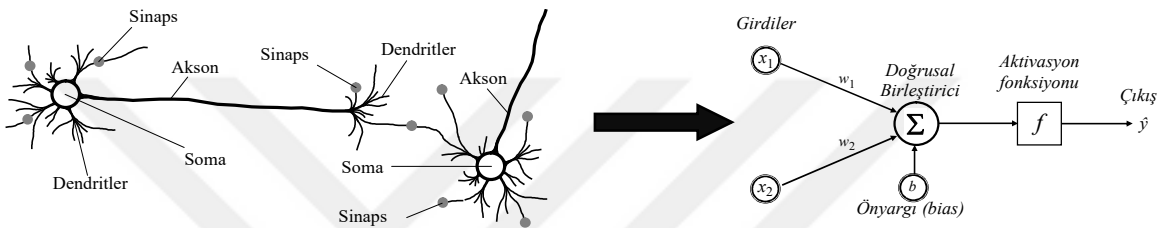
Yapay Sinir Ağları

Bu bölümde YSA'nın dönüm noktalarına göre gelişim süreci, YSA yapısına ve YSA'nın biyolojik sinir ağları ile olan ilgisi anlatılacaktır. YSA algoritmasının işleyişi,

hesaplaması, optimizasyonu ve performansı ile ilgili detaylara “Derin öğrenme” bölümünde yer verilmiştir.

YSA mimarisi

YSA algoritmaları insan beyninden ilham alınarak, görme, anlama, tanıma, anlamlandırma ve motor kontrol gibi karmaşık işlevlerin modellenmesi için geliştirilmiştir (Öztürk & Şahin, 2018). Biyolojik nöron hücrelerinin ve bu hücreler arasındaki sinaps bağlantılarının modellemesini esas alan YSA’da öğrenme, ağa girilen veriler ve çıktılar arasındaki ilişkinin tespit edilebilmesi ile sağlanmaya çalışılır. Şekil 3 biyolojik sinir hücresinin yapısını (solda) göstermektedir.



Şekil 3. Biyolojik ve yapay sinir hücresinin (nöron) yapısı (Negnevitsky & Intelligence, 2005)

Biyolojik sinir hücrelerindeki dentritleri YSA’da toplama fonksiyonu, hücre gövdesini transfer fonksiyonu, aksonları yapay nöron çıkışı, sinapsları ağırlıklar temsil etmektedir. Biyolojik sinir yapısı ile YSA karşılıkları Tablo 1’de verilmiştir.

Tablo 1. Biyolojik Sinir Yapısı ile YSA Karşılıkları

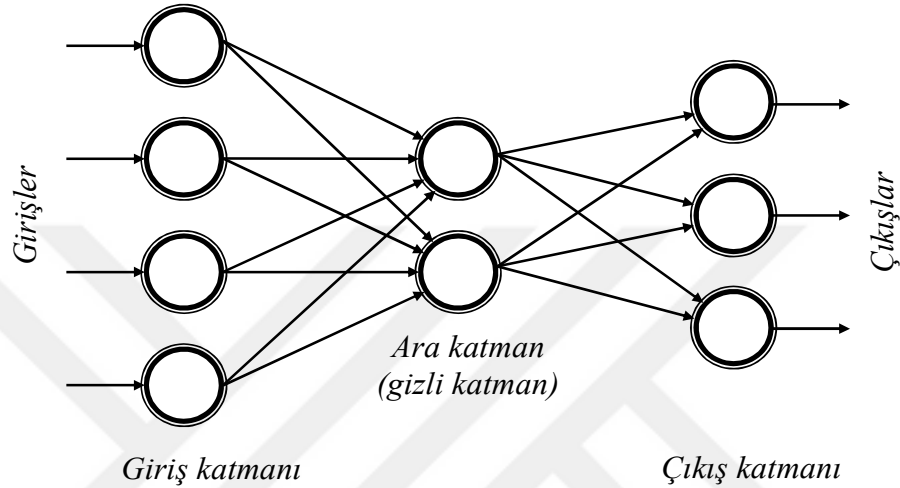
Biyolojik Sinir Sistemi	YSA
Nöron	Nod
Dentrit	Toplama işlemi
Hücre gövdesi	Aktivasyon işlemi
Akson	Nod çıkışı
Sinaps	Ağırlık

YSA’nın sinir hücrelerinin en temel ögesi olarak nöronlar (nod, algılayıcı veya perceptron olarak ta bilinen) tasarlanmıştır. Nöronlar, en basit tek katmanlı sinir ağı modelidir. Nöronlar, giriş değeri veya giriş katmanı, ağırlık ve katman önyargı (bias) değeri, toplam değeri, aktivasyon fonksiyonu olmak üzere 4 temel bileşenden oluşur. Şekil 3 yapay sinir hücresinin (nöron) yapısını (sağda) göstermektedir.

Teorik olarak bir YSA, birbirine bağlantılı yapay nöronlar koleksiyonu olarak tanımlanabilir (Fukushima, 1975). Nöronlardaki her ağırlık, beyindeki sinapslar gibi, diğer nöronlara bir sinyal iletebilir. Sinyali alan yapay bir nöron, onu işler ve kendisine bağlı nöronlara sinyal gönderir. Burada bağlantı sağlayan sinyal yani ağırlık gerçek bir sayıdır; her bir nöronun çıkışı, girdilerinin toplamının genellikle doğrusal olmayan bir fonksiyon ile

hesaplanır ve buna göre öğrenme sırasında bağlantılardaki sinyallerin gücü artar veya azalır (Basheer & Hajmeer, 2000).

Yapay sinir ağlarının üç katmanı bulunur. Bunlar; giriş katmanı, ara katman yani gizli katman ve çıkış katmandır (Öztemel, Yapay sinir ağları, 2003). Katmanlarda en az bir nöron bulunur. Veriler giriş katmanı ile ağı alınır. Giriş katmanının çıktıları bir sonraki katmana olan gizli katmana giriş değeri olarak sağlanır. Gizli katmanda veriler işlenerek bu katmanın çıktıları çıkış katmanına girdi olarak alınır. Örnek YSA mimarisi Şekil 4’de verilmiştir.



Şekil 4. Örnek YSA mimarisi

Geleneksel makine öğrenmesi yöntemleri ile YSA; öğrenme, veri ve kaynak ihtiyacı, hesaplama yöntemleri, hata toleransları ve performans gibi konularda bazı değişiklikler gösterir (Pirim, 2006). Geleneksel makine öğrenmesi algoritmaları ile YSA modelinin karşılaştırılması Tablo 2’de verilmiştir.

Tablo 2. Geleneksel Algoritmalar ve YSA Modeli Karşılaştırılması (Sabak & Başar, 2020)

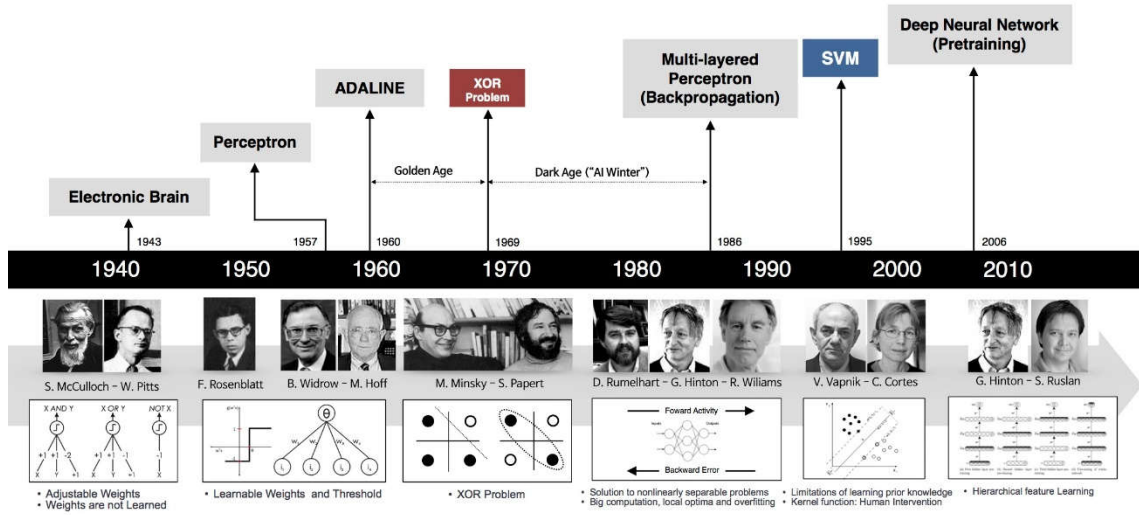
Geleneksel Algoritmalar	YSA
Çıkışlar, kurallardan göre elde edilir.	Çıkış bilgilerinden kurallar öğrenilir.
Veriler ve algoritma kuralları vardır.	Veriden deneyim çıkarılır.
Öğrenme; merkezi, senkron ve sıralıdır.	Öğrenme; birleşik, asenkron ve eğitimden sonra paraleldir.
Bellek paketlidir, hazır veri depolanmıştır.	Bellek ayrıdır, veri sinir ağına yayılmıştır.
Hata toleransı bulunmaz.	Hata toleransı bulunur.
Öğrenme hızlıdır.	Daha yavaş ve donanım gerektirir.

Bu bölümde YSA’nın temel çalışma metodunun anlatılması amaçlandığından ileri yayılım, geri yayılım nöron değeri, aktivasyon fonksiyonları hata fonksiyonları ve

optimizasyon ile ilgili hesaplamalara bir sonraki bölüm Derin Öğrenme başlığı altında yer verilmiştir.

YSA gelişim süreci

YSA, 1940'lı yıllarda mantıksal operatörlerle tasarlanmış mantıksal operatörlerin kullanılması ile ağ eğitime çalışılmıştır. 1950'li yıllarda ise ilk defa perseptron kuramı ortaya atılmış ve ağın eğitilmesi, ağdaki ağırlıkların eşik değerleri ile birlikte XOR operatörlerini kullanarak güncellenmesi ile sağlanmıştır. 1960'lı yıllarda ise Adaptif Lineer Nöron yaklaşımı ile ağ eğitime çalışılmış ve bu on yılın sonuna doğru XOR mantık operatörlerinin kullanımından vaz geçilmiştir (Serengil, Evolution of Neural Networks and Deep Learning, 2017).



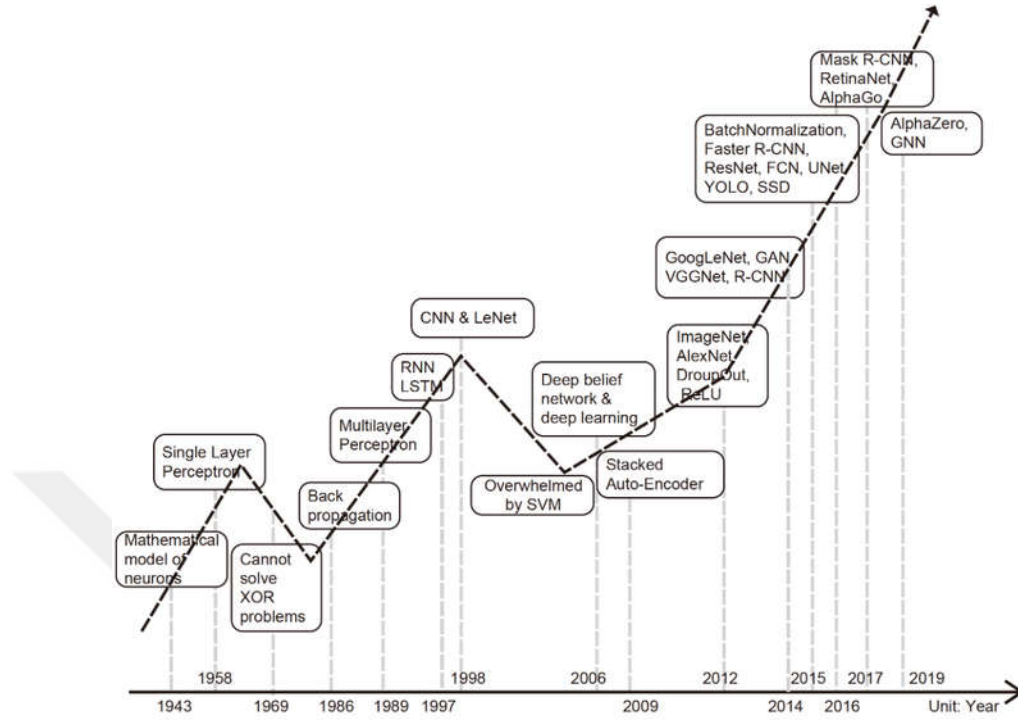
Şekil 5. Yapay sinir ağlarının kilometre taşları (Li ve diğerleri, 2021).

XOR Mantıksal operatörlerin kullanımının efektif olmadığı kanısından sonra 1986 yılına YSA popülerliğini kaybetmiş ve YSA yöntemi karanlık çağına girmiştir. 1986 yılında David Rumelhart, Geoffrey Hinton ve Ronald Williams'in YSA'yı 3 katmanlı olarak tasarlamasından ve geri yayılım yöntemi ile eğitmesinden sonra, YSA bu günkü yapısını almış ve araştırma-geliştirme çalışmaları hız kazanmıştır.

1986'dan sonra çok katlı mimarideki YSA için her ne kadar RNN, LSTM ve CNN uygulamaları kullanılmaya başlandıysa da o günlerde YSA'nın ihtiyacı olan hesaplama gücüne sahip donanımlar olmadığından, makine öğrenmesi çalışmaları SVM'e doğru eğilimlendirmiştir.

2012 yılından sonra ise Geoffrey Hinton ağdaki gizli katman sayısını artırılarak yüksek hesaplama yeteneğine sahip donanımların da yardımı ile derin öğrenme konseptini geliştirmiştir. Yüksek işlem gücüne sahip CPU ve GPU'lar, doğrusal olmayan aktivasyon

fonksiyonları (ReLU'lar, sigmoid, tanh, softmax vb.) ve optimizasyon için kullanılan yeni yaklaşımlarla YSA, Derin öğrenme türevi ile günümüzde üzerinde sıkça çalışılan bir çalışma alanı haline gelmiştir.



Şekil 6. Derin öğrenmenin tarihi ve gelişimi (Serengil, *Evolution of Neural Networks and Deep Learning*, 2017).

Şekil 5’de YSA ağlarının kilometre taşları ve Şekil 6’da Derin öğrenmenin tarihi ve gelişimi gösterilmiştir.

Kaynak Özetleri

Yapılan literatür taraması çalışmasında makine öğrenmesi yöntemlerinin tahmin üretme uygulamalarında oldukça kullanışlı olduğu ve başarılı sonuçlar verdiği gözlemlenmiştir. Son yıllarda DL tabanlı çalışmalar araştırmacıların yoğun ilgisini çektiği görülmektedir. Zaman seri uygulamalarda RNN modeli ve RNN modelinden esinlenerek geliştirilen LSTM, GRU ve bu modelleri türevleri ve hibrit kullanımlarından elde edilen tahmin değerlerinin gerçek değerlere yakın sonuçlar verdiği görülmüştür. Tarama sırasında incelenen bazı çalışmalar bu bölümde özetlenmiştir.

ML tahminleme çalışmaları

Konut binalarının soğutma ve ısıtma yüklerini tahmin etmek yapılan çalışmada (Duarte ve diğerleri, 2017) 748 binadan toplanan veri ve DT, RF, çok katmanlı NN ve SVM yöntemleri kullanılmış ve bu yöntemlerin tahminleme performansı değerlendirilmiştir. Çalışmada en iyi tahminleme performansının NN modeli ile elde edildiği ve kullanılan metrik sonuçlarından

MAE değerinin 0,565, RMSE değerinin 0,837, MAPE değerinin 2,342 ve R^2 değerinin 0,991 olarak elde edildiği görülmektedir.

Ham petrol fiyatı tahmini için ANN-Q Makine Öğrenimi yönteminin kullanıldığı çalışmada (Abdullah & Zeng, 2010), en iyi öğrenme sonuçları normalleştirilmiş verilerin kullanılması ile 5 gizli katmanla oluşturulan modelle elde edilmiştir. Yapılan çalışmada modelin gerçek değerlere uyumlu ve başarılı sonuçlar ürettiği, metrik olarak kullanılan RMSE değerinin 2,2690, NMSE değerinin 0,00896 ve Dstat değerinin %93,33 olarak elde edildiği belirtilmiştir.

Elektrik enerjisi ihtiyacının tahminlemesi için yapılan çalışmada (Vivas ve diğerleri, 2020), klasik istatistiksel ve matematiksel yöntemler ile makine öğrenimi modellerinin ürettiği değerler MAPE metriğine göre karşılaştırılmıştır. Çalışmada klasik yöntem olarak istatistiksel ve matematiksel zaman serisi regresyon yöntemi ARIMA modeli; ML modeli olarak YSA, RNN, SVM ve bulanık sinir ağı tabanlı tahmin yöntemlerinin kullanılmış ve ML modellerinin daha iyi tahmin doğruluğuna sahip olma eğiliminde olduğu sonucuna varılmıştır.

Malezya'nın Terengganu eyaletinde bulunan Kenyir gölü verilerinin kullanılması ile yapılan çalışmada (Ridwan ve diğerleri, 2021) ise; BLR, BDTR, DFR, NNR makine öğrenmesi modelleri kullanılarak gölün yağış tahminlemesi yapılmıştır. Çalışmanın bulgularında çalışılan makine öğrenmesi yöntemlerinin yağışları kabul edilebilir doğruluk düzeyinde tahmin edebildiği; çalışmaya hibrit makine öğrenme algoritmaları eklenmesi ve farklı iklim değişikliği senaryolarının dahil edilmesiyle daha doğru yağış tahmininin yapılabileceğini belirtilmiştir.

Hava kirliliği tahmini üzerine yapılan ANN, SVM, FL ve DNN yapay zekâ uygulamaları ile literatürde yapılmış çalışmaların incelendiği araştırmada (Masood & Ahmad, 2021) ise, bu alanda DNN'in en uygun teknik olduğu, modelin başarılı sonuçlar verdiği tespit edilmiş ve araştırmacılara gelecekteki çalışmalar için DNN'in farklı modellerini geliştirmeye odaklanmaları konusunda tavsiyelerde bulunulmuştur.

ML ile su parametrelerinin tahminlemesi çalışmaları

Kuveyt'in Ardiya ilindeki atıksu arıtma tesisinin 16 aylık verilerini kullanarak AKM ve BOİ parametrelerinin tahminlemesi üzerine yapılan çalışma (Hamoda ve diğerleri, 1999) YSA modeli kullanılarak gerçekleştirilmiştir. Çalışmada kullanılan modelin AKM parametresini %94,6 oranında BOİ parametresini ise %97,3 doğrulukla tahminleyebildiği ve YSA'nın atık su arıtma tesislerinin performansını tahmin etmek için alternatif bir metodoloji sağladığı belirtilmiştir.

YSA kullanılarak yapılan başka bir çalışmada (Öztemel & DÜGENCİ, Atıksu arıtma tesis kontrolde yapay sinir ağı ile kirlilik parametre tahmini, 2016) YSA modeli kullanılarak Sakarya Atıksu Arıtma Tesisinin kirlilik parametrelerinin tahminlemesi yapılmıştır. Çalışmada tesisin 652 günlük ölçüm sonuçları giriş parametresi olarak kullanılmış (Sıcaklık, Debi, pH, AKM, KOİ) ve tesisin çıkış değerlerinin (Sıcaklık, pH, AKM, KOİ) tahmini yapılmıştır. Çalışma sonucunda Sıcaklık değerinin %89,1 doğrulukla, pH değerinin %92,9 doğrulukla, AKM değerinin %78,9 doğrulukla, KOİ değerinin %86,5 doğrulukla olduğu ve YSA modelinin ortalama %86,9 doğrulukta değerler üretirken tesis kontrolünde karar vericilere yardımcı olacak nitelikte olduğu özetlenmiştir.

Arıtma tesisi atıksuyunun giriş KOİ ve BOİ konsantrasyonunu tahmin etmek için yapılan çalışmada (Yu ve diğerleri, 2018) ise, özellik çıkarımı ve boyutsallığının azaltılması için KPCA, tahminleme için ELM kullanılmıştır. Çalışmada kullanılan KPCA-ELM yönteminin MAE, MAPE ve RMSE sonuçlarına göre iyi bir tahmin kapasitesine sahip olduğu tespit edilmiştir.

Biyolojik Atıksu Arıtma Tesislerinin KOİ parametresinden BOİ parametresinin tahmini için yapılan başka bir çalışmada ise (Sezer M. , 2007), YSA modeli ve iki farklı atıksu arıtma tesisinin verileri kullanılarak yapılmıştır. Çalışmada eğitim için A tesisinin 365 deneysel verisiyle kullanılarak eğitim yapılmış model oluşturulmuş, daha sonra elde edilen model ile B tesisinin 66 adet deneysel verisi test edilmiştir. Çalışmada YSA modelinin tahminlemede başarılı sonuçları verdiği, modelin atıksu arıtma tesislerinde performans parametrelerinin üretilmesinde alternatif bir metot olabileceği görülmüştür.

Tebriz Atıksu arıtma tesisinin çıkış suyundaki BOİ ve KOİ parametrelerinin tahmini için FFNN, SVR ve ANFIS modelleri kullanılarak yapılan çalışmada (Nourani ve diğerleri, 2021), giriş verisi olarak tesise günlük giren BOİ, KOİ, AKM, pH (günlük ve bir gün önceki), BOİ ve KOİ parametreleri kullanılmıştır. Modellerin eğitimi için tesisin 2016 ile 2018 yılları arası günlük ölçüm sonuçlarının kullanıldığı çalışmada SVR modelinin kullanılan diğer modellere göre daha doğru sonuçlar verdiği gözlemlenmiştir.

Bangladeş Surma nehrinin BOİ ve KOİ miktarını tahmin etmek için FFNN ve RBFNN modelleri ve nehrin 3 yıllık deneysel verilerinin kullanıldığı çalışmada (Ahmed, 2017), oluşturulan YSA modellerinin performansı, korelasyon katsayısı (R^2), MSE ve verimlilik katsayısı (E) metrikleri kullanılarak değerlendirilmiştir. Çalışılan RBFNN ve FFNN ile modellerinin ürettiği değerlerin deneysel verilerle karşılaştırılması ile, sinir ağı modelinin uygun sonuçlar verdiğini tespit edilmiştir (Dingli & Fournier, 2017).

DL (RNN) zaman serilerinin tahminlemesi çalışmaları

Konutların elektrik enerjisi tüketiminin tahmini için yapılan çalışmada (Mocanu ve diğerleri, 2016), ANN, SVM, RNN, CRBM ve FCRBM modelleri kullanılmıştır. Kullanılan tüm yöntemlerin öğrenme süresinin çok kısa olduğu belirtilen çalışmada FCRBM'in ürettiği tahmin değerlerinin ANN, SVM, RNN ve CRBM gibi son teknoloji yöntemlerinden daha iyi performans gösterdiği belirtilmiştir.

Sezer vd. tarafından yapılan literatür incelemesinde (Sezer ve diğerleri, 2020), 2005–2019 yılları arasında yayınlanmış RNN tabanlı (LSTM ve GRU dahil) DL modelleri ile yapılan finansal zaman serisi tahmini çalışmaları taranmıştır. Yapılan araştırmada, incelenen çalışmaların çoğunda DL modellerinin performansının geleneksel makine öğrenimi modellerinin performansından daha iyi olduğu gözlemlenmiştir. Çalışmada ortaya çıkan başka bir bulgu da; finansal tahmin çalışmalarında DL modellerine yönelik genel ilginin arttığının tespit edilmesidir.

LSTM modeli ile yapılan başka bir çalışmada (Abbasimehr ve diğerleri, 2020), bir mobilya firmasının verileri kullanılarak elde edilen talep tahmin sonuçları istatistiksel ve makine öğrenmesi yöntemleri ile karşılaştırılmıştır. Çalışmada istatistiksel olarak ARIMA ve ETS yöntemleri makine öğrenmesi yöntemi olarak ise ANN, KNN, RNN ve SVM seçilmiştir. Çalışma sonucunda LSTM yönteminin tahmin sonuçları diğer makine öğrenmesi ve istatistiksel yöntemlerin sonuçlarından daha yüksek doğrulukta olduğu gözlemlenmiştir.

İki farklı petrol sahasının ham üretim verilerinin kullanıldığı başka bir zaman serilerinin tahminlemesi çalışması (Sagheer & Kotb, 2019), derin LTSM modeli ile gerçekleştirilmiştir. Çalışılan modelin ürettiği sonuçlar derin RNN ve derin GRU'dan daha iyi performans göstermiştir. Çalışmada ayrıca derin LSTM sonuçlarının istatistiksel ARIMA modelinden çok daha iyi olduğu gözlemlenmiştir.

Elektrik yükü tahmini üzerine RNN, LSTM ve GRU modellerinin kullanıldığı zaman serisi tahmin çalışmasında (Tokgöz & Ünal, 2018) ise, Metrik olarak MAPE'nın kullanılmıştır. Çalışmada kullanılan RNN, LSTM ve GRU derin öğrenme modellerinin ARIMA ve ANN modellerinden daha iyi tahmin başarısına sahip olduğu gözlemlenmiştir.

Hisse fiyat tahmini için yeni bir düzenlenleştirilmiş GRU-LSTM modelinin kullanıldığı çalışmada (Liu ve diğerleri, 2019), elde edilen sonuçlar GRU ve LSTM modelleri ile karşılaştırılmıştır. Deneysel sonuçları, GRU-LSTM modelinin GRU ve LSTM modellerine göre tahminlemede yüksek doğruluğu ve yüksek öğrenim hızına sahip olduğunu göstermiştir.

Hisse senedi fiyatı zaman serilerinin tahmini konusunda yapılan başka bir çalışma (Althelaya ve diğerleri, 2018), çift yönlü ve yığınlaştırılmış LSTM ve GRU modelleri ile yapılmıştır. S&P500 endeksinin geçmiş verilerinin kullanılarak yapıldığı çalışmada, yığınlaştırılmış LSTM mimarisinin hem kısa hem de uzun vadede en yüksek tahmin performansını gösterdiği tespit edilmiştir.

Kısa süreli trafik akışını tahmin etmek için LSTM GRU yöntemlerinin kullanıldığı başka bir zaman serisi çalışmasında (Fu ve diğerleri, 2016), deneysel sonuçların LSTM ve GRU mimarisindeki RNN tabanlı derin öğrenme yöntemlerinin ARIMA modelinden daha iyi performans gösterdiği gözlemlenmiştir.

Döviz (Forex) piyasası para birimlerinin gelecekteki kapanış fiyatlarını tahmin etmek için yapılan çalışmada (Islam & Hossain, 2020) ise GRU ve LSTM modelleri ile zaman serisi tahmini çalışması yapılmıştır. İki yıllık döviz piyasası verilerinin kullanıldığı çalışmada modelin performansının ölçülmesi için MSE, RMSE, MAE ve R^2 metrikleri kullanılmıştır. Çalışmada, oluşturulan hibrit GRU-LSTM modelinin 10 ve 30 dakika tahmini için tüm modellerden daha iyi performans gösterdiği sonucu görülmüştür.

DL (RNN) ile su parametrelerinin zaman serisi tahminlemesi çalışmaları

Mısır'daki El-Berka atık su arıtma tesisinde toplam AKM, KOİ, BOİ, amonyak ve sülfürün giderim verimliliğini tahmin etmek için yapılan çalışmada (El-Rawy ve diğerleri, 2021), tesisin 8 yıllık verisi ile YSA, DL (4 gizli katman kullanılarak) ve LSTM modelleri kullanılmıştır. Çalışılan modellerin tahmin sonuçları El-Berka Atıksu Arıtma Tesisinin parametre değerlerini etkili bir şekilde tahmin edebileceğini göstermiştir. Metrik olarak MSE, MAE, RMSE, MAPE ve R^2 nin kullanıldığı çalışmada en iyi tahmin performansı LSTM modeli ile elde edilmiştir.

Şehir drenaj ağlarının izlenmesi ve kontrol edilmesi için kanalizasyon suyu kalitesinin değerlendirilmesini esas alan çalışmada (Jiang ve diğerleri, 2021) atıksuda bulunan BOİ, KOİ, NH_4^+-N , TN, ve TP miktarlarının tahmini yapılmıştır. Çalışmada kullanılan derin öğrenme algoritmaları geleneksel makine öğrenimi algoritmaları ile karşılaştırılmış ve derin öğrenme modelleri (özellikle RNN, LSTM ve GRU) RMSE ve R^2 metriklerine daha iyi tahmin performansı göstermiştir. DL algoritmalarından ise GRU'nun en iyi performansa, daha hızlı öğrenme kabiliyetine ve daha basit mimariye sahip olma gibi avantajlara sahip olduğu belirtilmiştir.

LSTM yöntemi ile yapılan başka bir zaman serisi çalışması (Anuar ve diğerleri, 2021), Malezya Selangor nehrinin BOİ, KOİ, NH_3-N , TN, ve TP miktarlarının tahminlemesi için

yapılmıştır. Çalışmada metrik olarak RMSE kullanılmıştır. Modelin tahmin değerleri ile gerçek değerlerin uyumlu olduğu gözlemlenmiş ve modelin gelecekte su kalitesi parametrelerini tahmin edebileceğini ve su kalitesi tahmini için uygun bir yaklaşım olduğu belirtilmiştir.

Hindistan Delhi bölgesindeki Yamuna Nehir suyunun BOİ ve KOİ parametrelerinin tespit edilmesi için yapılan çalışmada (Khullar & Singh, 2022) derin öğrenme tabanlı bir Bi-LSTM modeli kullanılmıştır. Nehir suyunun 2013-2019 yılları arası aylık örneklerinden oluşan 6 yıllık veri setinin kullanıldığı modelin performans ölçümü için RMSE, MAE, MSE ve MAPE metrikleri kullanılmıştır. Yapılan çalışmanın deney sonuçlarına göre KOİ tahminlemesi için MSE, RMSE, MAE ve MAPE değerlerini sırasıyla 0.015, 0.117, 0.115 ve 20.32 olarak; BOİ tahminlemesi için MSE, RMSE, MAE ve MAPE değerleri ise sırasıyla 0.107, 0.108, 0.124 ve 18.22 olarak gözlemlenmiştir.

Şanghay'daki bir nehir suyunun kalitesinin izlemesi amaçlı yapılan çalışmada (Ye ve diğerleri, 2019) ise sudaki KOİ miktarının tahminlemesi çalışması yapılmıştır. Geleneksel GM, ve LSTM-RNN modelinin kullanıldığı çalışmada performans ölçümü için metrik olarak seçildiği çalışmada RMSE ve MAPE kullanılmıştır. Çalışmanın deney sonuçları, LSTM-RNN modelinin nehir suyu kalitesinin tahmin performansının geleneksel GM'den daha iyi olduğunu ortaya koymaktadır.

Atık su kalite standardının artırılması için atıksu arıtma tesisi giriş suyundaki KOİ parametresinin tahminlemesinin yapıldığı başka bir çalışmada (Wang ve diğerleri, 2019), hibrit CNN-LSTM, CNN ve LSTM modelleri kullanılmıştır. Çalışmanın sonucunda, hibrit CNN-LSTM modelinin, bağımsız CNN veya bağımsız LSTM modelinden daha yüksek doğrulukta tahminler ürettiği gözlemlenmiştir.

Güney Kore'deki Geum Nehri'nin su kalitesini izlenmesi amaçlı yapılan çalışmada (Lim ve diğerleri, 2019) ise, BOİ, KOİ ve AKM su kalitesi indekslerinin RNN ve LSTM modelleri ile tahmini yapılmıştır. Makine öğrenimi algoritmalarının uygulanması için Google tarafından geliştirilmiş olan açık kaynak kodlu TensorFlow Kütüphanesi kullanılmıştır. Kullanılan modellerde girdi olarak on yıl boyunca ölçülmüş günlük meteorolojik (sıcaklık ve rüzgar hızı) ve hidrolojik (su seviyesi ve akış debisi) veriler kullanılmıştır. RMSE ve R2'nin performans metriği olarak kullanıldığı çalışmanın deneysel sonuçları, her iki modelinde birbirleri ile göz ardı edilebilecek seviyede yakınlıkta ve yüksek doğrulukta olduğunu göstermiştir.

MATERYAL ve YÖNTEM

Materyal

Çalışmada materyal olarak Erzurum Biyolojik Atıksu Arıtma Tesisi (ERBASAT) ve tesisin işletimi sırasındaki ölçülen deneysel verileri seçilmiştir. Çalışmada tahmin üretmek için ise Python¹ dili ve TensorFlow² ve Keras³ derin öğrenme kütüphaneleri kullanılmıştır. Bu bölümde tesis, tesisin proses süreçleri, çalışmada kullanılan veri seti ve çalışılan modellerin tasarlanması için kullanılan yazılımlar ile ilgili açıklamalara yer verilmiştir.

Erzurum atıksu arıtma tesisi

Tesis, Erzurum Aziziye ilçesinin Kuşçu mevkii Ferah Mahallesi sınırındaki Pulur çayı ve Karasu nehrinin kuzeyinde yer almakta olup şehir merkezine ortalama 20 km mesafededir. Arıtılan suyun deşarj noktası, Fırat Nehri'nin başlangıcı ve en büyük kollarından biri olan Karasu Nehri'dir. ERBASAT'ın üstten görünümü **Şekil 7. ERBASAT'ın üstten görünümü** Şekil 7'de verilmiştir.



Şekil 7. ERBASAT'ın üstten görünümü (ESKİ, 2021).

Tesis hatlarının işleyiş özeti

Kanalizasyon sistemi ile atıksu arıtma tesisine gelen atıksular önce belirli aralıklı ızgaralardan geçirilerek atıksu içerisindeki katı atıklar tutulur. Izgara ünitelerinde filtre edilen atıksu havalandırılmalı kum-yağ tutucu proseslere girerek kumlar vb. inorganik maddeler hava ile havuz dibine çökertilerek bertaraf edilir. Atıksuda bulunan köpük ve yağ gibi yüzen

¹ <https://www.python.org>

² <https://www.tensorflow.org>

³ <https://keras.io>

Bu çalışmada tesisi girişindeki sudan alınan örneklerden ve ERBASAT'ın Şekil 9'da gösterilen havalandırma ve son çökertme havuzlarındaki süreçlerinden elde edilen deneysel veriler ve kullanılarak arıtılmış sudaki BOİ ve KOİ miktarlarının tahmini yapılacaktır.

Veri seti parametreleri

Literatürde BOİ ve KOİ konsantrasyonu tahminlemesi üzerine yapılan çalışmalar incelendiğinde, modellemelerde girdi parametresi olarak sadece KOİ değerlerini kullanarak BOİ değerlerinin tahmin çalışmasının (Sezer M. , 2007) yapıldığı ve ayrıca suyun debisi, sıcaklığı, zaman(t) , sertliği (pH), iletkenlik, AKM, toplam azot (tN) ve toplam fosfor (tP) parametrelerin kullanıldığı gözlemlenmiştir (Osman Tuğrul & Egemen, 2018). Havalandırma havuzundaki çözünmüş oksijen miktarı, hava miktarı(Nm³/sa) gibi tesisin proseslerinin işletilmesi sırasında ölçülen parametreler de BOİ ve KOİ miktarlarını etkileyebileceğinden bu parametreler de çalışılan modellere girdi parametresi olarak kullanılmıştır. Çalışmada kullanılan modellerin girdi parametreleri Tablo 3'de gösterilmiştir.

Tablo 3. Çalışmada Kullanılan Modellerin Girdi Parametreleri.

Alan Adı	Birimi	Açıklama
Debi	m ³ /gün	Tesis giriş debisi
Sıcaklık	°C	Tesis giriş suyunun sıcaklık değeri
pH	-	Tesis giriş suyunun pH değeri
İletkenlik	µs/cm	Tesis giriş suyunun iletkenliği
AKM	mg/l	Askıda Katı Madde miktarı
Giriş KOİ Miktarı	mg/l	Giriş suyunun KOİ miktarı
Giriş BOİ5 Miktarı	mg/l	Giriş suyunun BOİ5 miktarı
Giriş TN	mg/l	Giriş suyunun toplam azot miktarı
Ortalama Redoks	mV	Redoks reaksiyon ortalaması
Anoksik Çöz. Oks Ort.	m ³ /h	Anoksik havuzda bulunan ortalama çözünmüş oksijen miktarı
Çöz. Oks Ort.-1	m ³ /h	1. aerobik tankında bulunan ortalama çözünmüş oksijen miktarı
Çöz. Oks Ort.-2	m ³ /h	2. aerobik tankında bulunan ortalama çözünmüş oksijen miktarı
Çıkış TN	m ³ /h	Çıkış suyunun toplam azot miktarı
Hava Miktarı	mg/l	Havuzlarda verilen toplam hava miktarı

Çalışmada atıksu arıtma tesisinin atıksuyun arıtılması sırasında ERBASAT'ın akredite laboratuvarında yapılan 01.01.2019-31.07.2021 tarihleri arası 943 günlük analiz değerleri kullanılmıştır.

Derin öğrenme kütüphaneleri

Kolaylık ve esnekliğinden dolayı geliştiriciler ve araştırmacılar tarafından veri işlemede ve makine öğrenmesi uygulamalarında Python, C ve C++ programlama dilleri tercih

edilmektedir. Python dilinde geliştirilmiş başlıca derin öğrenme kütüphaneleri Tensorflow, Keras, Caffe, Pytorch, Microsoft Cognitive Toolkit, MXNet ve Deeplearning4j olarak sıralanabilir (Erickson ve diğerleri, 2017). Bu çalışmada uygulama geliştirilmesi için Python dili, TensorFlow 2.0 ve Keras API kullanılmıştır.

Çalışmada, Python ve R programlama dillerinin bilimsel hesaplama kullanımında paket yönetimini kolaylaştırmak için tasarlanmış ücretsiz ve açık kaynaklı dağıtım aracı Anaconda⁴ paketi içerisinde bulunan Jupyter Notebook⁵, Visual Studio Code⁶ geliştirme araçları kullanılarak kod geliştirme yapılmıştır.

Çalışma kapsamında geliştirilen uygulamada veri işleme ve kullanımını için Python programlama dilinde makine öğrenimi uygulamaları için geliştirilmiş çok boyutlu dizileri ve matrisleri destekleyen Numpy⁷ kütüphanesi, dosya işlemleri için Pandas⁸ kütüphanesi, veri görüntülemek için Matplotlib⁹ kütüphanesi, veri normalizasyonu, verinin bölünmesi ve metrik kullanımı için ise Scikit-Learn¹⁰ kütüphanesi kullanılmıştır.

Tensorflow

İlk sürümü 2017 yılında yayınlanan ve Google (Ertam & Aydın, 2017) tarafından makine öğrenimi amaçlı geliştirilen TensorFlow, büyük ölçekli ve sayısal hesaplama gerektiren uygulamalarda kullanılan açık kaynaklı C++ dilinde hazırlanmış, C++/Python dillerinde uygulama geliştirmeye olanak sağlayan bir yazılım kütüphanesidir. TensorFlow; ANN, DL SVM gibi makine öğrenimi modellerini ve algoritmalarını bir araya getirir ve bunların ortak bir platformda kullanımı sağlar.

Keras

Keras, açık kaynaklı ve geliştirilmesi devam eden bir sinir ağı kütüphanesidir. İlk sürümü 2015 yılı mart ayında yayınan Keras, François Chollet tarafından NN uygulamaları için Python dilinde (Chollet & others, 2018) geliştirilmiştir. Keras API içerisinde özel amaçlara uygun layer yapıları aktivasyon, normalizasyon, optimizasyon ve hata fonksiyonları gibi araçlar bulunur. Keras öğrenme modeli aşağıdaki adımlarla oluşturulur:

- Modelin oluşturulması (katmanların eklenmesi)
- Modelin derlenmesi,

⁴ <https://www.anaconda.com>

⁵ <https://jupyter.org>

⁶ <https://code.visualstudio.com>

⁷ <https://numpy.org>

⁸ <https://pandas.pydata.org>

⁹ <https://matplotlib.org>

¹⁰ <https://scikit-learn.org>

- Modelin eğitimi,
- Performansın değerlendirilmesi.

Keras kütüphanesinde yazılımın geliştirilmesi için kullanılan ve eğitilen modelin kaydedilmesi, test edilmesi ve farklı uygulamalarla kullanılması için olanak sağlayan özellikleri bulunmaktadır.

Yöntem

Makine öğrenme algoritmalarının tahmin üretme alanında önemli bir başarı sağladığı yapılan çalışmalarla (Duarte ve diğerleri, 2017) belirlenmiştir. Bu çalışmada, YSA'nın katmanlarının artırılması ile geliştirilen derin öğrenme tabanlı tahmin yöntemi kullanılacaktır. Gözetimli, yarı gözetimli, gözetimsiz (Yann LeCun & Hinton, 2015) ve pekiştirmeli öğrenme (François-Lavet ve diğerleri, 2018) alanlarında başarılı sonuçlar veren derin öğrenme, istatistik, sanayi, tıp, konuşma tanıma, görüntü işleme (nesne tanıma ve algılama vb.) gibi birçok alanda son teknolojiye önemli faydalar sağlamıştır (Bengio ve diğerleri, 2017). Tahmin üretme alanında ise enerji tüketimin belirlenmesi (Mocanu ve diğerleri, 2016), üretim (Wang ve diğerleri, 2018), finansal beklenti analizi (Dingli & Fournier, 2017) gibi alanlarda kullanılmıştır.

Bu çalışmada, arıtma tesisinin geçmiş verilerinin kullanılması ile gelecekte değerinin tahmin edilmesi için zaman serisi tahminine dayalı bir derin öğrenme yaklaşımı olan “Tekrarlayan Sinir Ağı” (RNN) yöntemi, RNN metodunda karşılaşılan yok olan ve patlayan gradyan probleminin çözümü geliştirilen “Uzun Kısa Süreli Bellek” (LSTM) yöntemi ve “Kapılı Tekrarlayan Hücre” (GRU) derin öğrenme yöntemi kullanılmıştır.

Yöntem bölümünde derin öğrenme, derin öğrenme modellerinin optimizasyonu ve performans iyileştirmeleri için kullanılacak yöntemler, RNN, LSTM ve GRU mimarileri açıklanacaktır.

Derin öğrenme

Derin Öğrenme (DL) algoritmaları, insan beyninden esinlenilerek geliştirilen YSA'nın daha gelişmiş bir türevi olarak tasarlanmıştır. Derin Öğrenme algoritmalarının, diğer makine öğrenme algoritmalarından ayrılan özelliği; karmaşık yapısı ve çok yüksek miktarda veriyi işleyebilecek çok yüksek işlem gücü olan donanım kaynaklarına ihtiyaç duymasıdır (Ng, 2013). Derin öğrenme algoritmaları esnek, güçlü ve genellemede iyi sonuçlar vermekte ve kendi özelliklerini (girdilerini) geliştirip öğrendiği için makine öğrenmesi uygulamalarında kullanışlı bir yöntemdir.

Derin Öğrenme, iyi sonuç elde edilebilmesi için çok fazla eğitim verisi örneğine ihtiyaç duyar ve binlerce hatta milyonlarca parametreye sahip olma eğilimindedir. Bu yüzden hesap maliyeti oldukça yüksektir ve özel hesaplama amaçlarına uygun donanım kaynaklarına ihtiyaç duyar. Özellikle bilgisayar donanımlarının performanslı hale gelmesi ve GPU'ları (Grafik İşlemci Ünitesi) kullanılması ile popülerlik kazanmıştır.

Bununla birlikte Derin Öğrenme algoritmaları günümüzde Facebook, Google, Microsoft, Nvidia gibi yüksek veriye ve hesaplama teknolojilerine sahip firmalar tarafından kullanılıp ürünlerine entegre edilmektedir. Bu teknoloji firmaları TensorFlow ve Pytorch gibi geliştirdikleri kütüphanelerini açık kaynak kodlu olarak geliştirici topluluklarla paylaşarak derin öğrenme algoritmalarının ilerlemesine destek olmaktadır.

Büyük verinin (BD) artması ve bu verilerle daha efektif sistemler ve ürünlerin geliştirilmesi ile derin öğrenme günümüzde akademik ve endüstriyel alanlarda üzerine sıkça çalışılan bir alan haline gelmiştir. (Khan ve diğerleri, 2019)

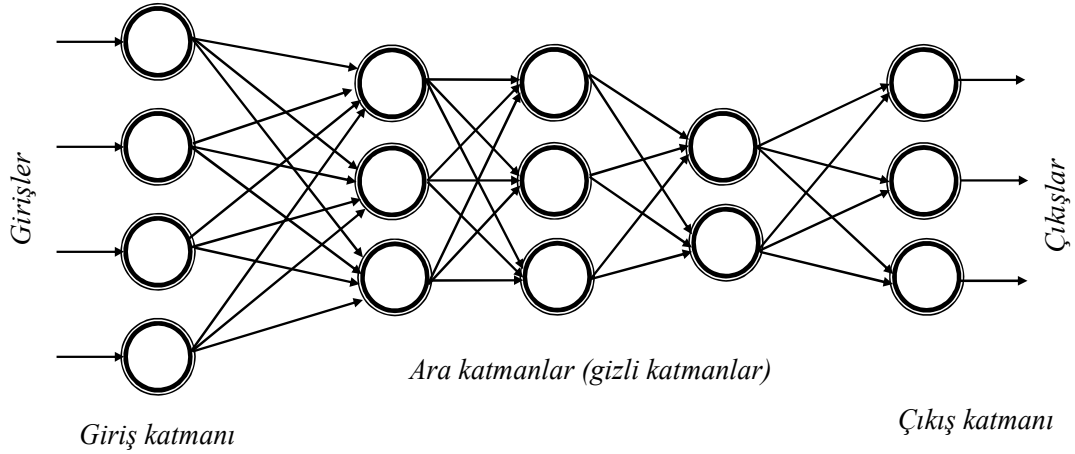
Sınıflandırma (classification), regresyon (regression), küme analizi (clustering) ve birliktelik kural çıkarımı (association rule mining) amaçlı kullanılan derin öğrenme algoritmalarının görüntü işleme (görüntü tanıma, yüz, iris, parmak izi vb. biyometrik algılama vb.), konuşma tanıma, metinden konuşmaya çeviri, tahmin üretme (regresyon), reklam uygulamaları, arama motorları, sosyal medya tanıma önerileri gibi uygulamaları bulunmaktadır (Deng & Yu, 2014).

Derin öğrenme modelleri kullanılacağı uygulamaların yapısına göre seçilmektedir. Örneğin sınıflandırma ve tahminleme uygulamalarında evrişimli sinir ağı (CNN) görüntü tanıma işlemlerinde, tekrarlayan sinir ağı (RNN) zaman serileri, konuşma tanıma gibi uygulamalarda kullanılır (Yann LeCun & Hinton, 2015).

Derin öğrenme modeli

Derin öğrenme modelleri de YSA gibi ve düğümler olarak da bilinen yapay nöronlardan oluşan sinir ağları kullanılarak geliştirilmiştir. YSA'nın birden fazla gizli katman (ara katman-hidden layer) olduğu durumlarda model, derin öğrenme algoritması olarak adlandırılır. Derin Öğrenme algoritması, gizli katmanları sayesinde giriş değerlerinin ayırt edici özelliklerini öğrenebilir ve bu değerler üzerinden yeni ilişkiler sağlayabilir. Derin öğrenme modelinin alt katmanlarındaki değerler daha az ayırt edici özelliği barındırırken bu katmanlardaki değerlerle oluşan modelin üst seviye katmanlardaki değerleri fazla ayırt edici özelliğe sahiptir. Algoritmanın performansı ve başarısı için uygun veri miktarının sağlanması ve bu verilerle

algoritmanın yeterince eğitimi gereklidir. Örnek derin öğrenme mimarisi Şekil 10'da gösterilmektedir.



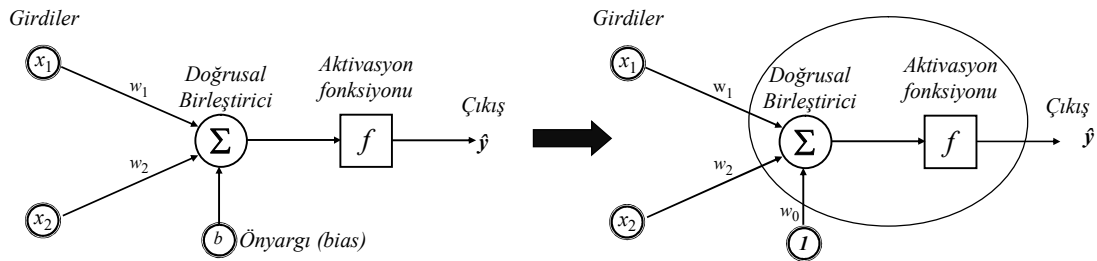
Şekil 10. Örnek derin öğrenme mimarisi.

Bu bölümde ilk olarak yapay sinir ağlarının temel taşı olan nöronlar ve aktivasyon fonksiyonları anlatılmıştır. Daha sonra nöron ve katman sayıları artırılarak çok katmanlı sinir ağları oluşturulmuş ve nöron değerlerinin hesaplanmasına yer verilmiştir.

Nöronlar

YSA Modeli bölümünde de bahsedildiği gibi, derin öğrenmenin ve her sinir ağının temel yapı taşı algılayıcı olarak da bilinen nöronlardır. Nöronlar girdi değerlerinden, her bir giriş değerine karşılık gelen ağırlık değerlerinden ve bias değerinden oluşur. Ağırlık değerleri için model geliştirilmeye başlanınca 0 ile 1 arası rastgele üretilen sayılar atanır. Nöronun değeri(z), giriş değerleri(x) ve bu değerlere karşılık ağırlıkların(w) çarpımının toplamı ile hesaplanır. Nöronun aktive edilmesi ise eşik değerine (önyargı-bias'a) bağlıdır. Eğer aktivasyon fonksiyonu sonucu ağırlıklar toplamı önyargı değerinden büyük ise nöron aktive edilir.

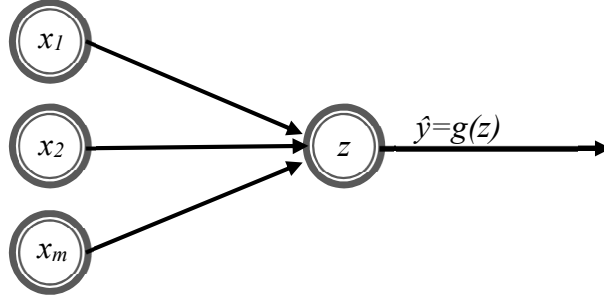
Şekil 11'de yapay nöron bileşenleri (solda) gösterilmiştir.



Şekil 11. Yapay nöron önyargı ilişkisi.

Model tasarlanırken nöronun önyargı değeri 1 olarak alınır ve giriş gibi düşünülerek önyargı için rastgele bir önyargı ağırlık değeri (w_0) atanarak üretilir. Yani önyargı değeri de bir

giriş olarak kabul edilir. Şekil 11’de (sağda) bias değerinin 1 alınarak ağırlığı ile çarpılıp nöron değerine eklendiği gösterilmiştir.



Şekil 12. Tek Katmanlı ve Tek Nöronlu Sinir Ağı.

$$z = w_o + \sum_{j=1}^m x_j w_j \quad (1)$$

$$\hat{y} = g(z) \quad (2)$$

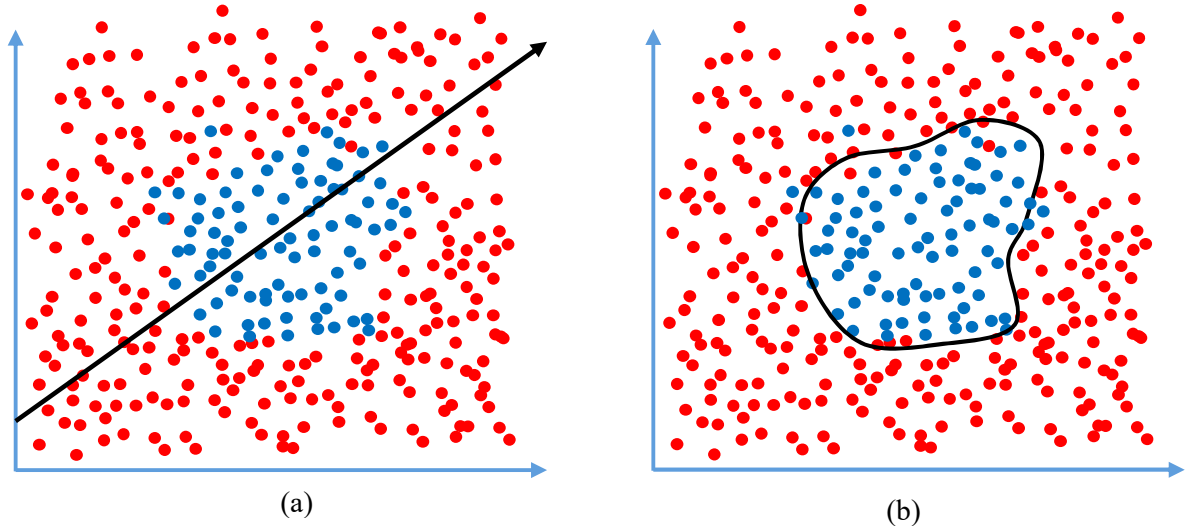
$$\hat{y} = g(w_o + X^T \cdot W) \quad (3)$$

Hesaplamaların daha kolay gösterilebilmesi en sade yapay sinir ağı olan tek katmanlı ve tek nöronlu sinir ağı (nöron) Şekil 12’de gösterilmiştir. Nöronun değerinin (z) hesaplanması için denklem 1’de gösterildiği gibi nöronun giriş değerleri (x_j), giriş değerlerine karşılık gelen ağırlıklar (w_j) ile çarpılır ve çıkan sonuca önyargı değeri (w_o) eklenir. Nöronun çıkış değeri ($\hat{y}$) ise Denklem 2’de gösterildiği gibi nöron değerinin doğrusal olmayan aktivasyon fonksiyonunun sonucu ile belirlenir.

Nöronun çıkış değerinin matrislerle ifade etmek için, giriş değerleri $\mathbf{X}$ matrisi ile ve girişlere ait ağırlıklar ise $\mathbf{W}$ matrisi ile gösterilmiştir. Nöronun çıkış değerleri ise Denklem 3’te gösterildiği gibi $\mathbf{X}$ matrisinin transpozunun $\mathbf{W}$ matrisi ile nokta çarpımından çıkan skalar değere önyargı değerinin eklenmesi ile elde edilen değerinin (nöron değeri) aktivasyon fonksiyonuna göre sonucu ile elde edilebilir.

Aktivasyon fonksiyonları

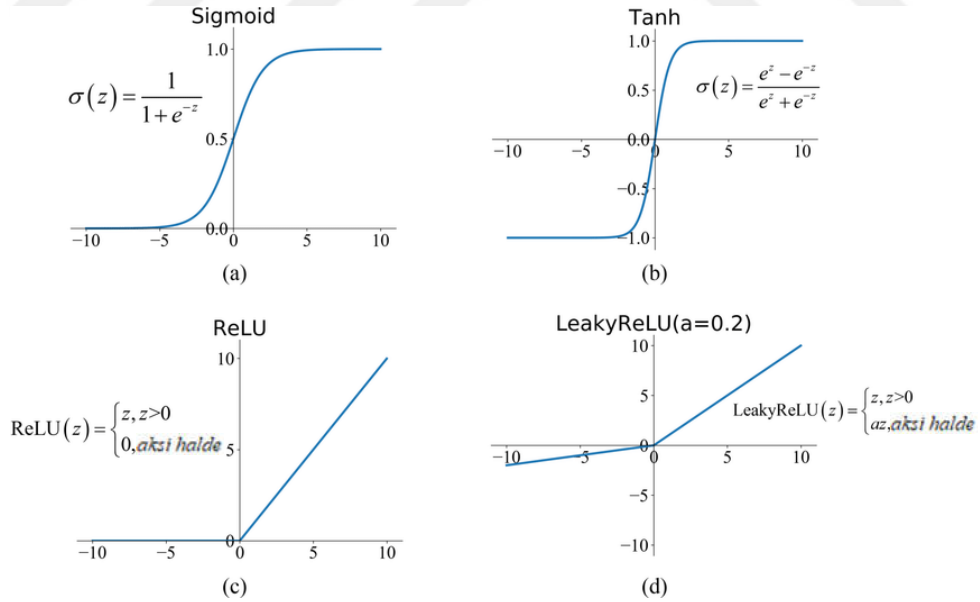
Ağıdaki nöronların toplam değerlerinin etkinleştirme işlevi için aktivasyon fonksiyonları (g) kullanılır (Hou ve diğerleri, 2016). Aktivasyon fonksiyonları performans ve öğrenme konusunda kilit bir role sahiptir. Derin öğrenme yönteminde kullanılan veri setinin neredeyse tamamı doğrusal olmayan karmaşık bir yapıda olduğundan veriler bir doğru ile temsil edilebilecek yapıda değildir. Doğrusal olmayan aktivasyon fonksiyonunun amacı bu verilerin ağı dâhil edilebilmesidir (Vijayaprabakaran & Sathiyamurthy, 2020).



Şekil 13. Örnek veri setinin doğrusal ve doğrusal olmayan fonksiyonlarla temsil edilmesi.

Aktivasyon fonksiyonlarının başka bir özelliği ise türevlenebilir yapıda olmalarıdır. Ağın eğitimi sırasında her bir nöronun ağırlığının hataya göre türevi alınacağından kullanılacak aktivasyon fonksiyonunun türevlenebilir olması gerekir. Şekil 13-a'da örnek veri setinin doğrusal olan fonksiyonla ve Şekil 13-b'de doğrusal olmayan fonksiyonla temsil edilmesi gösterilmiştir.

Derin öğrenme araştırmalarında ve uygulamada kullanılan sıkça kullanılan sigmoid, Tanh, ReLU ve LeakyReLU aktivasyon fonksiyonları (Nwankpa ve diğerleri, 2018) Şekil 14'de verilmiştir.

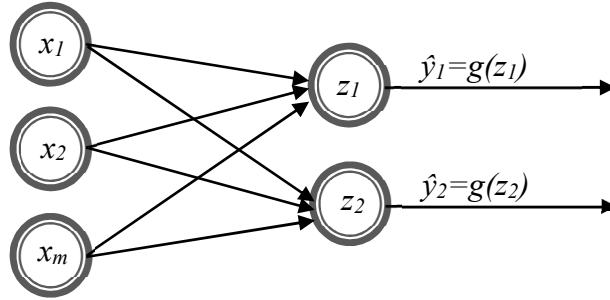


Şekil 14. Aktivasyon fonksiyonları.

Düzeltilmiş doğrusal birim (ReLU) aktivasyon fonksiyonu, derin öğrenme uygulamalarında bugüne kadar en yaygın kullanılan fonksiyondur (Nair & Hinton, 2010).

Derin öğrenme modelinin oluşturulması

Birden fazla çıkışı olan bir yapay sinir ağının gösterilmesi için Şekil 12’de gösterilen nörona bir nöron daha eklenmiş ve Şekil 15’deki gibi iki nöronu, iki çıkışı olan en sade yapay sinir ağı oluşturulmuştur. Burada her nöronun kendi ağırlığı olduğu görülmektedir. Her iki nöronun da önyargı değerleri aynıdır.



Şekil 15. İki nöronlu yapay sinir ağı.

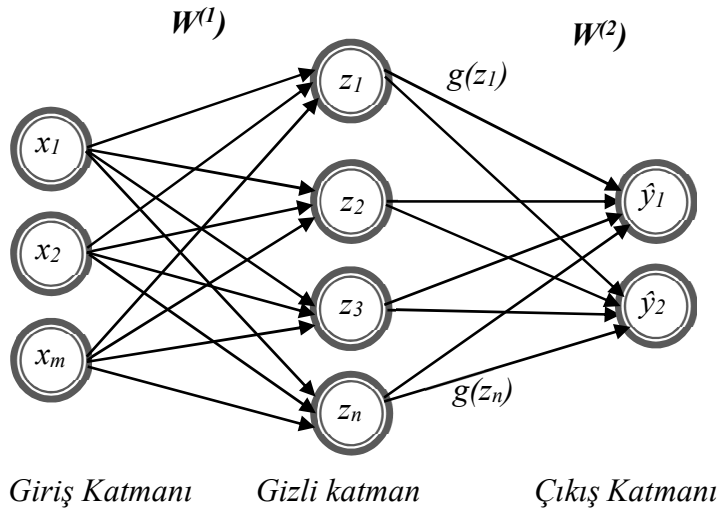
$$z_i = w_{o,i} + \sum_{j=1}^m x_j w_{j,i} \quad (4)$$

$$\hat{y}_i = g(z_i) \quad (5)$$

$$\hat{y}_i = g(w_o + X^T \cdot W) \quad (6)$$

Nöronların değerlerinin (z_i) hesaplanması Denklem 4’de gösterildiği gibi aynı girişler ile nöronun kendi ağırlık değerlerinin çarpımlarına önyargı değerinin eklenmesi ile bulunur. Nöronların çıkış değerleri (y_i) ise Denklem 5’de gösterildiği gibi her bir nöron değerlerinin doğrusal aktivasyon fonksiyonuna göre sonucu ile belirlenir.

Şekil 15’deki girişler ve çıkışlar arasına gizli katman eklenerek Şekil 16’deki gibi katmanlı bir yapay sinir ağı oluşturulmuştur. Yapay sinir ağı içinde sadece bir ara katman olduğunda “Tek katmanlı yapay sinir ağı” diye adlandırılır. Ara katmanlar giriş ve çıkış katmanları gibi işlemler sırasında gözlemlenmediğinden, gizli katman olarak tanımlanır.



Şekil 16. Tek katmanlı yapay sinir ağı.

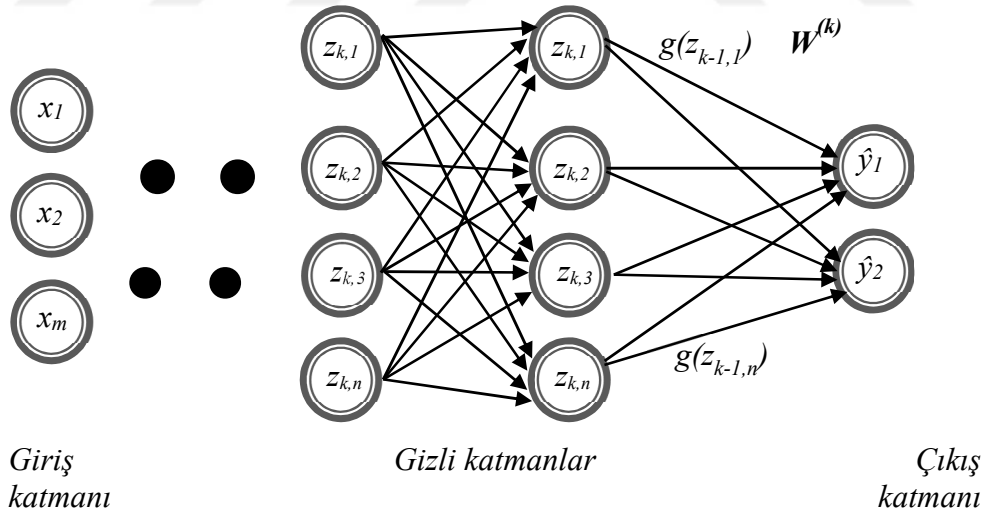
$$z_i = w_{oi}^{(1)} + \sum_{j=1}^m x_j w_{ji}^{(1)} \quad (7)$$

$$\hat{y}_i = g \left(w_{oi}^{(2)} + \sum_{j=1}^n g(z_j) w_{ji}^{(2)} \right) \quad (8)$$

Ağa gizli bir katmanın eklenmesi sonunda, ağı giriş değerleri ile gizli katmanlar arasında ve gizli katmanlar ile çıkış değerleri arasında yeni $W^{(1)}$ ve $W^{(2)}$ ağırlık vektörleri oluşturulmuş olur. Bu yüzden ağı çıkış değerlerinin hesaplanması için önce gizli katmandaki nöronların değerlerinin hesaplanması ve bu gizli katman çıkış değerlerinin ağ çıkış nöronlarının değerinin bulunmasında giriş değeri olarak kullanılması gerekir.

Gizli katmandaki nöronların değerleri Denklem 7’de gösterildiği gibi giriş değerleri ile $W^{(1)}$ ağırlık vektörünün değerlerinin nokta çarpımlarının toplamına ve önyargı değerinin eklenmesi ile bulunur. Çıkış değerlerinin bulunması için ise Denklem 8’de gösterildiği gibi gizli katman nöron değerlerinin aktivasyon sonucu ve $W^{(1)}$ ağırlık vektörünün değerlerinin nokta çarpımlarının toplamına ve önyargı değerinin eklenmesinin aktivasyon fonksiyon sonucu ile bulunur.

YSA’ya gizli katmanlar eklenerek oluşturulan Derin Öğrenme modelinin gösterimi karmaşıklığı önlemek amacı ile Şekil 17’de basitleştirilerek verilmiştir. Modele eklenen her bir ağı yeni $W^{(1)}, W^{(2)} \dots W^{(k)}$ ağırlık vektörleri bulunmaktadır.



Şekil 17. Çok katmanlı YSA (Derin öğrenme modeli).

$$z_{ki} = w_{oi}^{(k)} + \sum_{j=1}^{n_{k-1}} g(z_{k-1,j}) w_{ji}^{(k)} \quad (9)$$

$$\hat{y}_{k,i} = g \left(w_{oi}^{(k+1)} + \sum_{j=1}^n g(z_{k,j}) w_{ji}^{(k+1)} \right) \quad (10)$$

Denklem 9’da gösterildiği gibi her bir katmanın nöron değerlerinin hesaplanması için bir önceki katmanın nöron değerlerinin aktivasyon sonuçları alınır ve katmanın ağırlıklarıyla olan çarpımına önyargı değeri eklenir ve aktivasyon fonksiyonu sonucu alınıp işlem çıkış

katmanına kadar sürdürülür. Çıktı katmanının değerleri ise Denklem 10'da gösterildiği gibi son gizli katmanının nöron değerlerinin aktivasyon sonucu çıktı katmana giriş değeri olarak kabul edilir ve tahmin değerleri üretilir.

Şekil 17'te gösterilen Derin Öğrenme modeli sıralı model(sequential model), model katmanların birbirleri ile olan ilişkisi tam bağlantılı (fully connected), gizli katmanlar ise (dense layer) olarak adlandırılır.

Derin öğrenme modelinin eğitilmesi

Derin öğrenme modelinin eğitilmesi, en az hataya sahip ağırlık değerlerinin belirlenmesi işlemidir (Basheer & Hajmeer, 2000). Oluşturulan modelin eğitilmesi için veri setinin modele gösterilmesi gerekmektedir. Modelin eğitilmesi ileri yayılım ve geril yayılım yöntemleri ile yapılır. İleri yayılım işleminde veri seti girdi olarak kullanılır ve ağırlık ve çıkış değerleri bulunur. Ağırlık ileri yayılım işleminden sonra bulunan çıkış değerleri gerçek sonuçlar ile karşılaştırılır ve ağırlıklar çıkan hataya göre geri yayılım işlemi ile güncellenir (Yann LeCun & Hinton, 2015). Eğitim Sırasında veriler modele tekrar tekrar gösterilir ve modelin eğitilmesi sağlanır.

Veri setindeki her bir veri için yapılan ileri yayılım ve geri yayılım işlemine iterasyon, eğitim için kullanılan verinin tamamının kullanılarak ağırlık eğitime ise çağ veya ipok (epoch) adı verilir. Ağırlık eğitimi, önceden belirlenen bir ipok sayısı tamamlanınca veya ipoklar arasında hata oranında büyük bir değişiklik yoksa sonlandırılır.

Derin öğrenme ağlarında ileri yayılım (Forward propagation)

İleri Yayılma, veri setindeki girdi değerlerinin gizli katmanlar üzerinden çıktı katmanlarına aktarılması işlemidir. İleri yayılım işleminde katmanlardaki nöronların değerleri ve ağırlıklar ve çıktı değerleri “derin öğrenme modelinin oluşturulması” bölümünde anlatıldığı gibi hesaplanır ve bu işlem çıktı katmanına kadar sürdürülür. İleri yayılım sonunda ağırlık hesaplanan çıktı değeri gerçek çıktı değeri ile karşılaştırılır ve hata fonksiyonlarıyla ağırlıkların doğruluğuna göre ağırlık başarısı tespit edilir.

Hata fonksiyonları ve metrikler

Ağırlık hatanın tespiti için hata fonksiyonları (kayıp fonksiyonu veya maliyet fonksiyonu olarak ta adlandırılır) kullanılır. Modelin veri setini ne kadar iyi modellediğini değerlendirme için kullanılır. Hata fonksiyonunda $\hat{y}$ hesaplanan yani tahmini çıkış değerlerini y ise gerçek değeri temsil eder. Hata fonksiyonları bu iki değeri karşılaştırır ve ağırlık hatasını hesaplar. Hata fonksiyonu sonucu çıkan hata, ağırlık her bir ağırlığın hatasının tamamını temsil

eder. Hata fonksiyonlarının değeri her zaman pozitif sonuçlar üretmektedir ve hata mümkün olduğunca minimize edilmeli ve sifıra yakın olmalıdır. Tahmini değer ile gerçek değerler birbirlerine yakınlığı olursa modelin başarısını gösterir. Tahminler gerçek sonuçlardan çok fazla saparsa, hata fonksiyonun değeri artar ve tahminler gerçek değerlere ne kadar yakın olursa hata değeri sifıra yaklaşmış olur.

Hata fonksiyonları modelin kullanılacağı problem türüne göre seçilir. Derin öğrenme yönteminde sıklıkla kullanılan ve çalışmada kullanılmış metrikler olan; Ortalama Mutlak Hata fonksiyonu (MAE) Denklem 11’de, Kareli Ortalama Hatası (MSE) fonksiyonu Denklem 12’de, Ortalama Kare Kök Hata fonksiyonu (RMSE) Denklem 13’de, Ortalama mutlak yüzde hatası Denklem 14’de, Belirleme Katsayısı (R^2) Denklem 15’de ($\bar{y}$: gerçek değerlerin ortalaması) verilmiştir.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (11)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (12)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (13)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_i} \quad (14)$$

$$R^2 = 1 - \frac{SS_{err}}{SS_{total}} = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (15)$$

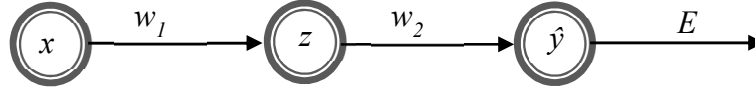
Derin öğrenme ağlarında geri yayılım (Back propagation)

Ağın eğitimiyle, ileri yayılım işleminde hesaplanan çıkış değerlerinin yani tahmini değerlerin, veri setindeki çıkış değerlerine yani gerçek değerlere mümkün olduğunca yakın olması sağlanır. İleri yayılım işlemi sonunda hesaplanan hatanın miktarı, tahmini ve gerçek çıkış değerlerinin birbirlerine yakınlığını gösterir. Eğer hata tahmini değerler ile gerçek değerler arasında fark fazla ise hata değeri büyük, fazla değilse hata değeri küçük hesaplanacaktır.

Ağda tespit edilen hatanın giderilmesi geri yayılım işlemi ile yapılır. Geri yayılım, sinir ağındaki ileri yayılım sonrasında hesaplanan çıktılar ile gerçek değerler arasında oluşan hataya göre nöronların ağırlık değerlerinin güncellenmesi işlemidir.

Geri yayılım işleminde önce ağdaki her bir ağırlığın hatadaki etkisi hesaplanır, daha sonra ağdaki tüm ağırlıklar modele dışardan girilen hiper parametre olan öğrenme oranına göre

tespit edilen hata miktarı kadar güncellenir. Ağırlıkların toplam hatadaki etkisinin tespiti, ilgili ağırlığın toplam hataya göre kısmi türevi alınarak hesaplanır.



Şekil 18. Geri yayılım için örnek yapay sinir ağı.

Geri yayılım işleminde ağırlıkların türevlerinin alınması zincir kuralına göre alınır. Ağırlıkların türevinin alınmasının daha iyi ifade edilebilmesi için Şekil 18'deki bir girişi, bir ara katmanı ve bir çıkışı olan örnek yapay sinir ağı kullanılmıştır. Örnek yapay sinir ağındaki w_2 ağırlığının toplam hatadaki etkisi Denklem 16'da, w_1 ağırlığının toplam hatadaki etkisi ise Denklem 17'de gösterilmiştir.

$$\frac{\partial E}{\partial w_2} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2} \quad (16)$$

Denklem 16'da da görüldüğü gibi w_2 'nin toplam hatadaki etkisinin bulunması için, w_2 'nin toplam hataya göre türevi alınmıştır. Hata fonksiyonu (E), w_2 türünden ifade edilemediği için önce toplam hatanın tahmini değere göre kısmi türevi $\left(\frac{\partial E}{\partial \hat{y}}\right)$ daha sonra w_2 ağırlığının tahmini değere göre kısmi türevi $\left(\frac{\partial \hat{y}}{\partial w_2}\right)$ alınmış ve zincir kuralı uygulanmıştır.

$$\frac{\partial E}{\partial w_1} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_1} = \frac{\partial E}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z} \cdot \frac{\partial z}{\partial w_1} \quad (17)$$

Denklem 17'de ise w_1 'in toplam hatadaki etkisinin bulunması için, w_1 'in toplam hataya göre türevi alınmıştır. Hata fonksiyonu (E), w_1 türünden ifade edilemediği için önce toplam hatanın tahmini değere göre kısmi türevi $\left(\frac{\partial E}{\partial \hat{y}}\right)$, tahmini değer z nöronun çıkış değerine göre türevi $\left(\frac{\partial \hat{y}}{\partial z}\right)$ sonra da w_1 ağırlığının tahmini değere göre kısmi türevi $\left(\frac{\partial z}{\partial w_1}\right)$ alınmış ve zincir kuralı uygulanmıştır.

Modelin tüm ağırlıklarının güncelleme işlemi için, ağırlıklar vektörünün hata fonksiyonuna göre türevini oluşturan Denklem 18'de gösterilmiş olan gradyan vektörü (∇) kullanılır. Gadyan yukarı yönde bir vektör alanını verir ve değişimin en büyük olduğu değer büyüklüğündedir. Gadyan, artımın en çok olduğu yeri gösterdiğinden, hatanın minimize edilmesi için gradyan vektörünün tersi yönünde (negatif) işlem yapılır. Ağırlıkların güncellenmesi Denklem 19'de gösterildiği gibi öğrenme oranı miktarınca yapılır. Hiper parametre olan öğrenme oranı skaler bir değer olup modele dışardan parametre olarak girilir.

Denklem 20’de modeldeki her bir ağırlığın güncellenme miktarına göre güncellenmesi gösterilmiştir.

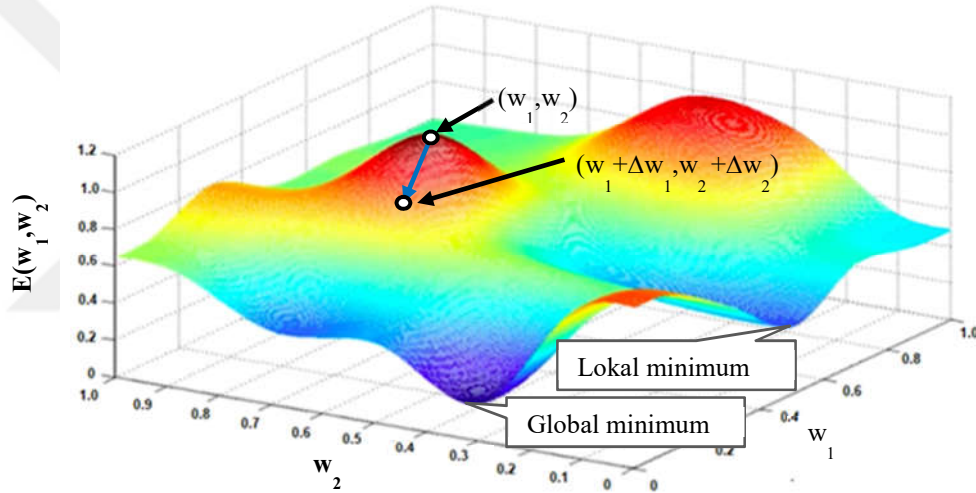
Öğrenme oranı hiper parametre olarak sabit veya adaptif (uyarlanabilir) olarak eğitimin değişimine göre modele uygulanabilir. Ağırlıkların adaptif olarak güncellenmesi “Optimizasyon yöntemleri” bölümünde anlatılmıştır.

$$\nabla E(\mathbf{W}) = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \dots, \frac{\partial E}{\partial w_n} \right) \quad (18)$$

$$\Delta \mathbf{W} = -\eta \cdot \nabla E(\mathbf{W}) \quad (19)$$

$$\mathbf{W} = \mathbf{W} + \Delta \mathbf{W} \quad (20)$$

Şekil 18’de gösterilen sinir ağı için w_1 ve w_2 ağırlıklarının hata fonksiyonuna göre oluşturabileceği gradyan grafiği Şekil 19’da verilmiştir.



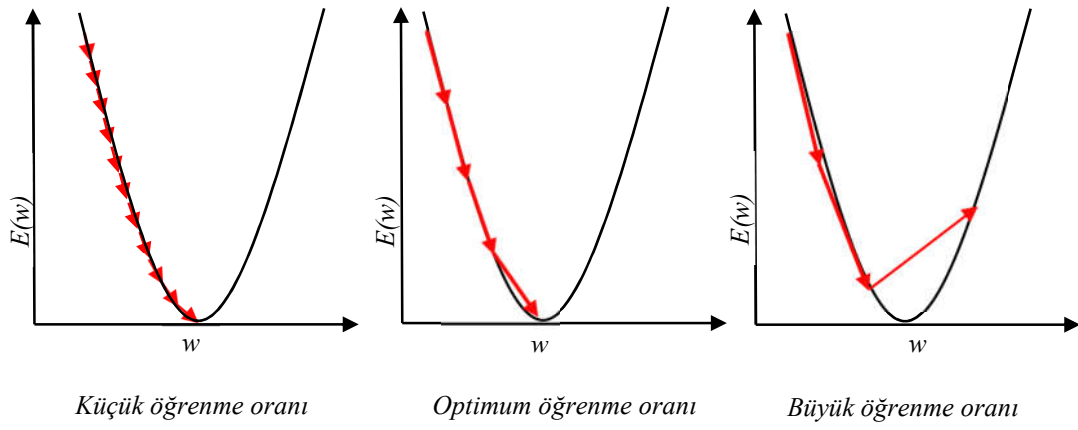
Şekil 19. Gradyan inişi grafiği.

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i} \quad (21)$$

$$w_i = w_i + \Delta w_i \quad (22)$$

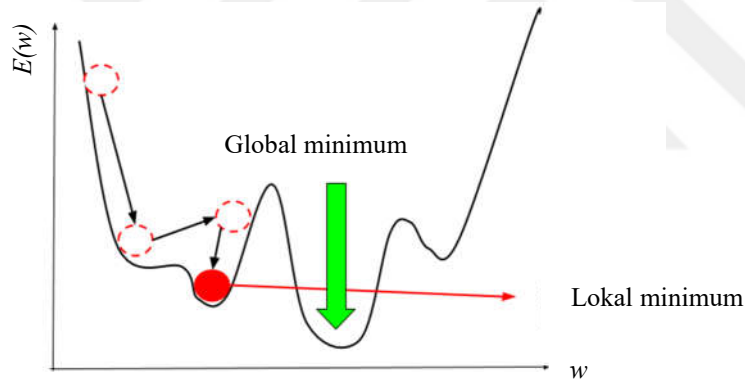
$$w_1 = w_1 - \eta \frac{\partial E}{\partial w_1}, w_2 = w_2 - \eta \frac{\partial E}{\partial w_2} \quad (23)$$

Değişim miktarı için gradyanın tersi yönde hareket edildiğinden, öğrenme miktarı her bir ağırlığın öğrenme oranı kadar, Denklem 21’de ki gibi gradyanın negatifi alınarak hesaplanır ve Denklem 22’de gösterildiği gibi ağırlıklar güncellenir. Denklem 23’de w_1 ve w_2 ağırlıklarının güncellenen değerleri verilmiştir. Güncellenen yeni ağırlıklar ile modelin tekrar ileri yayılım işlemi uygulanarak ağırlık doğruluk oranının artması sağlanır.



Şekil 20. Öğrenme oranının modelin eğitimine etkisi.

Öğrenme oranının modelin eğitimine etkisi Şekil 20’de verilmiştir. Öğrenme oranının küçük seçilmesi ile (solda) ağırlıklar hatasız bir şekilde güncellenir fakat bu işlem fazla zaman alır. Öğrenme oranının büyük seçilmesiyle (sağda) ise eğitim zamanı kısalır fakat ağırlıkların büyük oranda güncellenmesi ağırlıkların başarısını ters yönde etkiler (Basheer & Hajmeer, 2000). Bu yüzden hem ağırlıkların eğitiminin optimum sürede olması hem de başarı oranının yüksek olması için öğrenme oranının optimum (ortada) değerinin tespit edilmesi gerekir.



Şekil 21. Global minimum ve lokal minimum.

Derin öğrenme modelinin hata fonksiyonlarının gradyan iniş grafiği Şekil 21’de görüldüğü gibi genelde dışbükey olmayan (non-convex) formdadır. Modelin başarı seviyesinin artırılması için hatanın gradyan inişine göre en düşük değerine yani global minimuma ulaşması gerekmektedir. Öğrenme hızının çok düşük seçilmesi hatanın lokal optimum değere takılarak, global optimum değere hiç ulaşamamasına neden olabilir. Global minimum değere ulaşılması için kullanılan momentum ve mini toplulaştırma gibi yöntemlere “Optimizasyon yöntemleri” bölümünde yer verilmiştir.

Gradyan kaybolması ve gradyan patlaması

Gradyan kaybolması ve gradyan patlaması, ağırlık eğitimi sırasında gradyan vektöründeki değerlerin aşırı düşük veya aşırı yüksek çıkması durumudur. Gradyan vektöründeki değerlerin 0'a yakın olması gradyan kaybolması, aşırı yüksek çıkması ise gradyan patlaması olarak tanımlanır. Gradyan kaybolması ve gradyan patlaması ağırlık eğitimi engelleyen durumlardır (Skansi, 2018).

Gradyan kaybolmasına ağırlıkların küçük seçilmesi ve aktivasyon fonksiyonu sebep olur. Gradyan kaybolması sıklıkla ağırlık çok katmanlı olduğu durumlarda, ağırlık giriş katmanına yakın olan ağırlıklarda meydana gelmektedir. Gradyan vektöründeki değerler zincir alma kuralına göre türev alınarak bulunduğu, 0'a çok yakın değerlerin birbiri ile çarpılmasıyla gradyan gittikçe 0'a yaklaşarak kaybolacaktır. Gradyan kaybolması durumunda, ağırlıklar gittikçe küçülen değerlerle güncellenir ve bu yüzden eğitimi aşırı yavaşlar veya durma noktasına gelir.

Gradyan patlamasında ise tam tersi bir durum gerçekleşir. Ağırlık değerlerinin büyük seçilmesiyle büyük olan bu değerler birbirleri ile çarpılırken aşırı yüksek sonuçlar verecektir. Gradyan patlamasında ağırlıkların güncellenmesi çok yüksek değerler ile yapılacağından ağırlık değerleri gittikçe büyüyecek ve ağırlıklar uygun değerlere ulaşamayacaktır.

Gradyan kaybolması ve gradyan patlaması ağırlık eğitimi engellediğinden, ağırlık doğru sonuçla almak için bu sorunların çözülmesi gerekir. Gradyan kaybolması ve gradyan patlaması sorununun çözümünde ağırlıkların başlangıç değerlerinin seçimi ve aktivasyon fonksiyonun doğru seçilmesi önemli rol oynamaktadır. Örneğin ağırlık geri yayılımı sırasında sigmoid aktivasyonunun türevi 0 ile 0,25 arasında değerler alacağından, sigmoid fonksiyonu yerine RELU aktivasyon fonksiyonunun kullanılması veya sigmoid fonksiyonuna uygun ağırlık değerlerinin belirlenmesi gradyan kaybolması sorununa çözüm olmaktadır. Gradyan kaybolması ve gradyan patlamasını önlemek için ağırlıkların ilk değerlerinin seçilmesi konusuna ise çalışmanın “

Ağırlık başlangıç değerlerinin atanması” bölümünde yer verilmiştir.

Optimizasyon yöntemleri

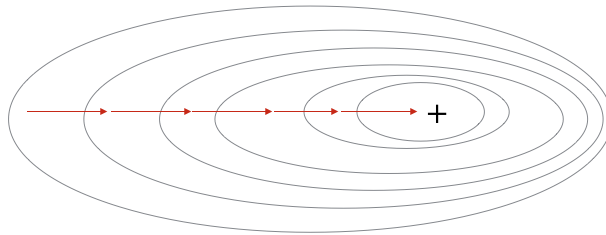
Derin Öğrenme algoritmalarında gerek veri setlerinin büyük olması ve gerekse gizli katmanların sayılarının fazlalığı modelin eğitimini zaman ve donanım kaynaklarının tüketilmesi konusunda olumsuz etkiler. Bu yüzden bellek ve işlemci gibi daha az donanım kaynağı ile ve daha az zamanda modelin eğitiminin yapılması için bir takım optimizasyon yöntemleri kullanılır. Optimizasyon yöntemleri, hata fonksiyonuna göre gradyan inişini kullanarak global minimum hatanın en verimli şekilde minimize edilebilmesi için, ağırlıkların güncellenme miktarının tespit edilmesini amaçlar. Optimizasyon yöntemleri ağırlıkların nasıl güncelleneceği ve modelin nasıl öğreneceği konusunda performans bakımından kilit öneme sahiptir.

Optimizasyon yöntemlerinin problem türü ve eğitim performansı bakımından avantaj ve dezavantajları bulunur. Her bir Optimizasyon yönteminin avantaj ve dezavantajlarından aşağı bahsedilmiştir.

Gradyan inişi

Gradyan inişi (GD), küçük adımlar halinde ağırlıkların güncellenmesi yöntemidir. Gradyan inişi yöntemi, modele veri setinin tamamının girilmesi ve daha sonra ağırlıkların güncellenmesini yöntemidir. Bu yöntemde, ağırlıklar her bir ipok sonrasında güncellenerek model eğitildiğinden ağırlık eğitimi fazla zaman almaktadır ve gradyan vektörü büyük olduğundan hata fonksiyonun hesaplanması için çok fazla bellek ve işlemci kaynağına ihtiyaç duyulmaktadır. Şekil 22’de gradyan inişine göre ağırlıkların güncellenmesi gösterilmiştir.

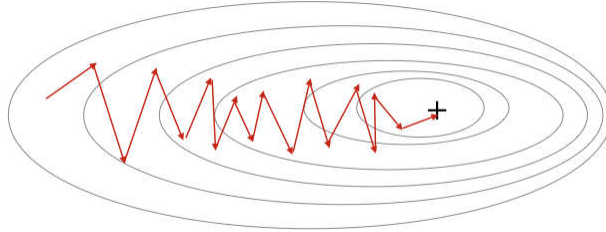
Hatanın giderilmesi için ağırlıklar doğru bir şekilde güncellenmektedir fakat modelin eğitimi büyük veri setlerinde oldukça yavaştır. Modelin hızlı öğrenmesi için öğrenme oranının büyük seçilmesi gerekmektedir. Öğrenme oranının büyük seçilmesi ise zamanı kısaltmasına rağmen ağırlıkların doğru güncellenmesini etkilemektedir.



Şekil 22. Gradyan inişine göre ağırlıkların güncellenmesi.

Stokastik gradyan inişi

Stokastik gradyan inişı (SGD), gradyan inişinde ihtiyaç duyulan bellek ve işlemci kaynağının azaltılması ve öğrenmenin hızlandırılması için ağırlıkların sık sık güncellenmesi yöntemidir. Stokastik gradyan inişı yönteminde ağırlıklar her bir ipokta veri setinin rastgele alınan bir örneğinde güncellenir.

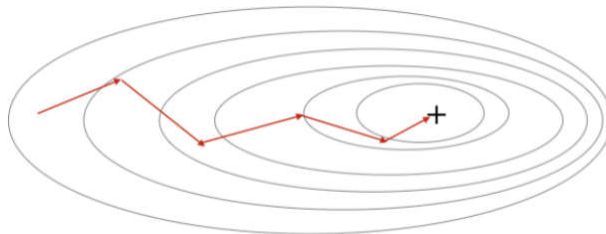


Şekil 23. Stokastik gradyan inişine göre ağırlıkların güncellenmesi.

Stokastik gradyan inişinde ağırlıkların veri setinin rastgele alınan bir örneği için sık sık güncellenmesi ise Şekil 23’de de görüldüğü gibi gürültülü adımlar meydana getirir.

Mini-gruplandırılmış stokastik gradyan inişı

Mini-gruplandırılmış stokastik gradyan inişı (MBSGD), stokastik gradyan inişinde yaşanan gürültünün azaltılması için veri seti, öğrenme oranı gibi dışarıdan girilen bir parametreye göre gruplara ayrılır ve bu gruplar içinden rastgele alınan bir örneğe göre ağırlıkların güncellenmesi yapılır.

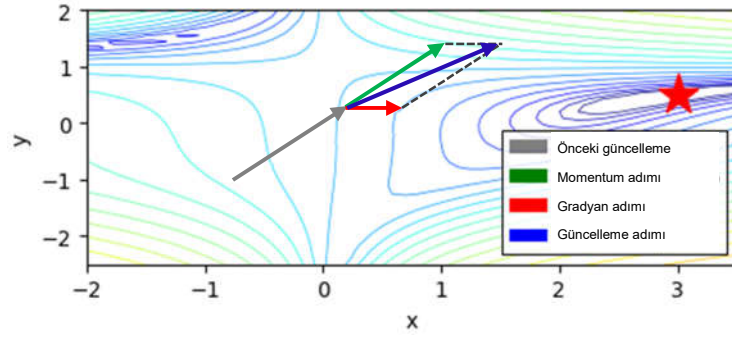


Şekil 24. Mini-gruplandırılmış stokastik gradyan inişine göre ağırlıkların güncellenmesi.

Mini-gruplandırılmış stokastik gradyan inişı yönteminde, ağırlıklar her bir ipok sırasında girilen gruplandırma miktarı kadar ağırlıklarda güncelleme yapılır. Şekil 24’de görüldüğü gibi bu yöntemde stokastik gradyan inişı yöntemi kadar sapmalar olmaz.

Momentum tabanlı gradyan inişı

Gradyan İnişinde ağırlıkların güncellenmesi zaman aldığından, momentum tabanlı gradyan inişı yöntemi ile modelin eğitiminin hızlandırılması amaçlanmıştır. Ağırlıklardaki değişim arka arkaya aynı yönde olduğunda, ağırlıklar üssel (eksponansiyel) olarak daha büyük adımlarla güncellenir. Ağırlıkların değişim hızının (v_t) hesaplanmasında - pratikte çoğunlukla 0,5 ile 0,9 arasında seçilen - öğrenme oranı gibi parametrik bir değer olan, momentum faktörü (γ) kullanılır.



Şekil 25. Gradyan inişi ve momentum.

$$\mathbf{v}_{(t)} = \gamma \mathbf{v}_{(t-1)} - \eta \nabla E(\mathbf{w}_{(t)}) \quad (24)$$

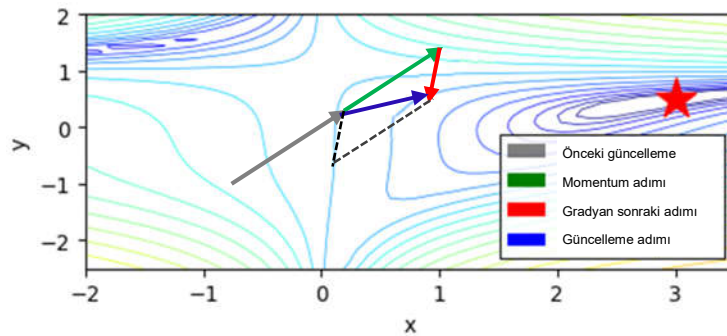
$$\mathbf{w}_{(t+1)} = \mathbf{w}_{(t)} + \mathbf{v}_{(t)} \quad (25)$$

Şekil 25 Şekil 25. Gradyan inişi ve momentum.

'te gösterildiği gibi Ağırlık vektörünün bir önceki güncelleme değerinin momentum faktörü oranında artışına, gradyan iniş vektörünün toplanması ile alınması gereken adım hesaplanır. Bu işlem Denklem 24'de gösterilmiştir. Ağırlıklar hesaplanan güncelleme miktarınca Denklem 25'te gösterildiği gibi güncellenir.

Nesterov hızlandırılmış gradyan inişi

Gradyan İnişinde yaşanan yavaşlık probleminin başka bir çözümü için Nesterov Hızlandırılmış Gradyan İnişi yöntemi kullanılmaktadır. Bu yöntemde de ağırlıklarda aynı yönde arka arkaya değişim olduğunda, ağırlıklar üssel (eksponansiyel) olarak daha büyük adımlarla güncellenir. Bu yöntemde, momentum tabanlı gradyan inişi yönteminden farklı olarak önceki verilere değil ağırlığın alabileceği bir sonraki değerine bakılarak ağırlıklar güncellenir.



Şekil 26. Gradyan inişi ve Nesterov momentum.

$$\mathbf{v}_{(t)} = \gamma \mathbf{v}_{(t-1)} - \eta \nabla E(\mathbf{w}_{(t)} + \gamma \mathbf{v}_{(t-1)}) \quad (26)$$

$$\mathbf{w}_{(t+1)} = \mathbf{w}_{(t)} + \mathbf{v}_{(t)} \quad (27)$$

Şekil 26’da gösterildiği gibi Ağırlık vektörünün bir önceki güncelleme değerinin momentum faktörü oranında artışına, gradyan iniş vektörü ve bir sonraki ağırlığın toplanması ile alınması gereken adım hesaplanır. Bu işlem Denklem 26’da gösterilmiştir. Ağırlıklar hesaplanan güncelleme miktarınca Denklem 27’de gösterildiği gibi güncellenir.

AdaGrad optimizasyonu

SGD, GD momentum yöntemlerinde veri setindeki bütün örnekler için yani ağdaki her bir giriş değeri için aynı öğrenme oranı kullanılmaktadır. Genellikle veri setlerinde birden fazla örnek türü bulunur ve bu türlerin her biri modelin eğitimini farklı ağırlıklarla etkiler. Modelin eğitiminde veri setindeki bazı örnekler için yüksek, bazı örnekleri için düşük öğrenme oranı ile ağırlıkların güncellenmesi gerekir. Bu yüzden ağırlıklar dinamik ve girişlerin özelliklerine göre ölçeklendirilen, uyarlanabilir öğrenme oranı (ALR) yöntemleri ile güncellenir. Bu yöntemde ağırlıkların güncellenmesi, ağırlıklara göre modelin öğrenmiş olduğu uyarlanabilir öğrenme oranı ile yapılır.

Uyarlanabilir öğrenme oranı yöntemlerinden AdaGrad, öğrenme hızının parametrelerle uyarlanması için önceki gradyan adımlarının büyüklüğünü kullanır. Gradyan değişiminin büyük veya küçük olduğunu tespit etmek için Denklem 28’de görüldüğü G gibi ölçeklendirme vektörü kullanılmıştır. Eğitimin başında her bir ağırlık aynı öğrenme oranı ile güncellenirken, sonraki güncellenmelerde ağırlıklar birbirlerinden bağımsız olarak ağırlıkların değişimine göre ölçeklendirilmiş öğrenme oranı ile güncellenir. Her bir ağırlığın öğrenme oranının ölçeklendirme miktarlarının tespiti, ilgili ağırlığa ait olduğu değişimlerin toplamı ile hesaplanır. Ölçeklendirme miktarının pozitif bir değer olmasını sağlamak için değişimlerin kareleri alınır.

Denklem 29’da gösterildiği gibi öğrenme oranının G vektörüne göre ölçeklendirilmesi ile ağırlıklar güncellenir. Böylece, gradyanın büyük olma durumunda öğrenme oranı (hata fonksiyonunda eğimin fazla olma durumunda) azaltılmış, gradyanın küçük olma durumunda(hata fonksiyonunda eğimin az olma durumunda) ise artırılmış olur. Değişimin çok az olduğu durumlarda gradyan eğimi sıfır değerini alabileceği için sıfıra bölünmeyi engellemek için sonucu etkilemeyecek çok küçük ε sayısı(Ör: $\varepsilon=10^{-7}$) eklenir.

$$\mathbf{G}_{i,(t)} = \sum_{t=1}^t E(\mathbf{w}_{i,(t)})^2 \quad (28)$$

$$\mathbf{w}_{i,(t+1)} = \mathbf{w}_{i,(t)} - \frac{\eta}{\sqrt{\mathbf{G}_{i,(t)} + \varepsilon}} \nabla E(\mathbf{w}_{(t)}) \quad (29)$$

AdaGrad yöntemi ile her ne kadar öğrenme performansını artırmış olsa dahi, güncelleme vektörü $\mathbf{G}$ bir toplam vektörü olduğundan öğrenme oranı zaman içinde azalır ve bu

durumda öğrenme yavaşlar. Bununla birlikte öğrenmenin başında çok büyük gradyan olacağından ve bu büyüklükler toplamı, ilerleyen zamanda gradyan küçük olsa dahi eğitimi yavaşlatacaktır. Bu durumlar AdaGrad yönteminde karşılaşılan en büyük sorunlardır.

RMSProp optimizasyonu

Uyarlanabilir öğrenme oranı yöntemlerinden RMSProp optimizasyonu, momentum tabanlı gradyan inişi yöntemi ile aynı mantığa sahiptir. Öğrenme oranının ölçeklenmesi için her bir ağırlığın geçmişteki hareketli ortalamasının karelerini kullanır. RMSProp optimizasyonu gradyan hareketlerini esas aldığından ilerleme dikey yönde sağlanır. Ağırlıkların her birinin güncellenmesi için, öğrenme oranı Denklem 30'da verilen ölçeklendirme vektörü ($\mathbf{m}$) kullanılır. Ölçeklendirme için kullanılan parametrik hareketli ortalama faktörü (γ) geçmişteki güncellemelerin ne kadar önemli olduğunu belirler. Güncel hareketli ortalamasının karelerinin etkisi fazla, eski hareketli ortalamasının karelerinin etkisi daha az olacağından, ağırlıklar hareketli ortama değerlerine bağlı olarak üssel (eksponansiyel) olarak güncellenir.

$$\mathbf{m}_{i,(t)} = \gamma \mathbf{m}_{i,(t-1)} + (1 - \gamma) \nabla E(\mathbf{w}_{i,(t)})^2 \quad (30)$$

$$\mathbf{w}_{i,(t+1)} = \mathbf{w}_{i,(t)} - \frac{\eta}{\sqrt{\mathbf{m}_{i,(t)} + \varepsilon}} \nabla E(\mathbf{w}(t)) \quad (31)$$

Denklem 31'de gösterildiği gibi öğrenme oranının $\mathbf{m}$ vektörüne ölçeklendirilmesi ile ağırlıkla güncellenir. AdaGrad yönteminde olduğu gibi paydada sıfıra bölünmeyi engellemek için sonucu etkilemeyecek çok küçük ε sayısı eklenir.

Adam optimizasyonu

Uyarlanabilir öğrenme oranlarını hesaplayan yöntemlerinden biri olan Adam optimizasyon yöntemi, RMSprop ve momentumun optimizasyonlarının birlikte kullanılmasıdır. Her bir ağırlığın öğrenme oranının ölçeklenmesi Denklem 32'deki RMSprop yöntemi kullanılmış (γ_1 = hareketli ortalama faktörü) ve ağırlıkların değişim hızı için ise Denklem 33'deki momentum yöntemi (γ_2 =momentum faktörü) kullanılmıştır. Denklem 34'te ise öğrenme oranı her bir ağırlığın geçmişteki hareketli ortalamasının karelerine göre ölçeklendirilerek momentum vektörüne göre yeni ağırlıklar belirlenmiştir.

$$\mathbf{m}_{i,(t)} = \gamma_1 \mathbf{m}_{i,(t-1)} + (1 - \gamma_1) \nabla E(\mathbf{w}_{i,(t)})^2 \quad (32)$$

$$\mathbf{v}_{i,(t)} = \gamma_2 \mathbf{v}_{i,(t-1)} + (1 - \gamma_2) \nabla E(\mathbf{v}_{i,(t)}) \quad (33)$$

$$\mathbf{w}_{i,(t+1)} = \mathbf{w}_{i,(t)} - \frac{\eta}{\sqrt{\frac{\mathbf{m}_{i,(t)}}{1-\gamma_1}} + \varepsilon} \frac{\mathbf{v}_{i,(t)}}{1-\gamma_2} \quad (34)$$

Regülerizasyon yöntemleri

Makine öğrenmesinde verilere uymayan basit bir model önyargı (bias), verilere gereğinden fazla uyan karmaşık bir model varyans (variance) olarak adlandırılır. Tablo 4’de önyargı ve varyans ilişki durumlarından görüldüğü gibi modelde eğitim ve test hatasının yüksek olduğu durumlarda yüksek önyargı, eğitim hatası düşük ve test hatası yüksek olduğu durumlarda yüksek varyans durumu ile karşılaşılır.

Modelin, eğitim ve test verilerinde düşük performans göstermesi eksik öğrenme (underfitting); eğitim veri kümesinde iyi performans gösterdiği halde test verilerinde yeterli performansı gösterememesi ise aşırı öğrenme (overfitting) olarak tanımlanır.

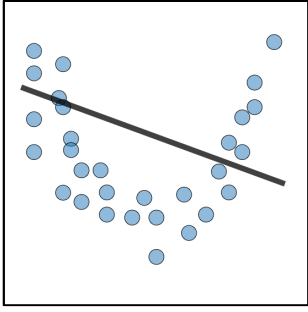
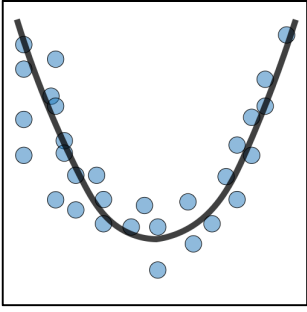
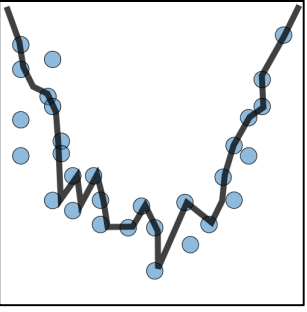
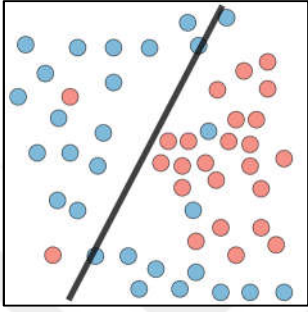
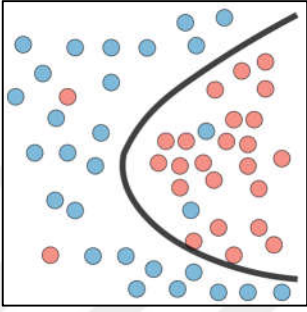
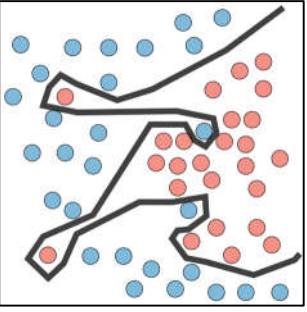
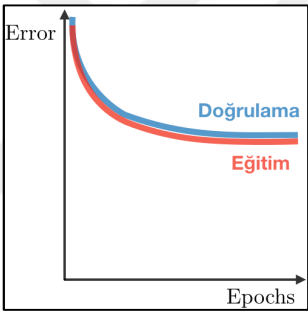
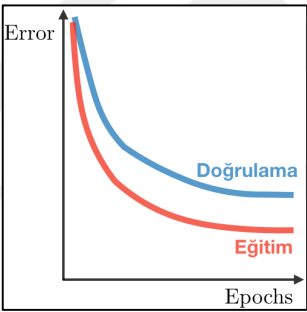
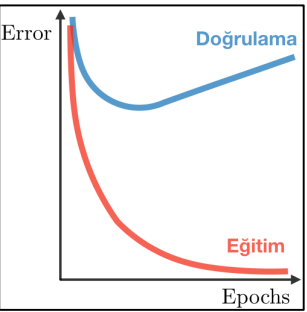
Tablo 4. Önyargı - Varyans İlişki Durumları

Eğitim Hatası	Test Hatası	Sebebi
Yüksek	Yüksek	Yüksek önyargı
Düşük	Yüksek	Yüksek varyans
Düşük	Düşük	Uygun ilişki

Modelin yeteri kadar eğitilememesi veya modelin veri setine uygun olmaması eksik öğrenmenin en büyük sebeplerindendir. Tablo 5’te görüldüğü gibi (sol sütün) model eğitim verisini temsil etmemekte ve modelin eğitim ve test verilerinde hata oranı yüksektir. Bu durumda modelin karmaşıklığının ve Eğitim verisinin artırılması gerekir.

Modelin gereğinden fazla eğitilmesi durumunda ise eğitim verisinde hata oranının az olduğu ancak test verilerinde hata oranı yüksek olduğu Tablo 5’de (sağ sütün) görülmektedir. Eksik öğrenme ve aşırı öğrenme durumları sadece derin öğrenmede değil, tüm makine öğrenimi yöntemlerinin en temel sorunlarından biridir. Derin öğrenme uygulamalarında regülerizasyon yöntemleri model genelleştirilmesi için kullanılan yöntemlerdir.

Tablo 5. Öğrenmenin Genelleştirilmesi

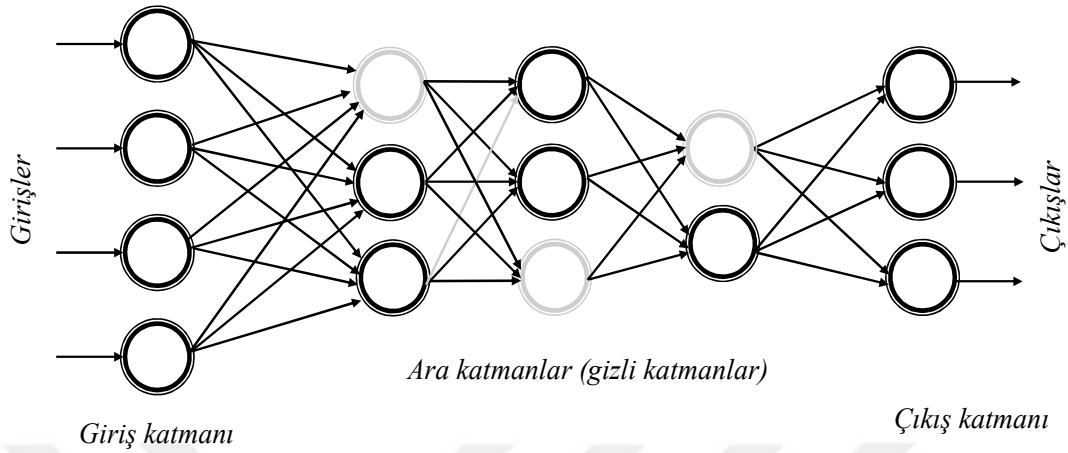
Belirtiler	Underfitting Eksik öğrenme	Just right Optimal sonuç	Overfitting Aşırı öğrenme
Regresyon			
Sınıflandırma			
Derin öğrenme eğitimi			

Tablo 5’te (orta sütün) optimal sonucun eğitim verisini temsil ettiği (genelleme oranı yüksek olduğu), eğitim ve test verisinde hata oranlarının kabul edilebilir seviyede düşük olduğu görülmektedir. Diğer makine öğrenme yöntemlerinde olduğu gibi aşırı öğrenme sorunu, derin öğrenme yönteminin de temel sorunlarından biridir. Bu bölümde modelin sadece eğitim verisinde değil, daha önceden görmediği veriler üzerinde de başarılı olmasını sağlamak için kullanılan regülerizasyon yöntemleri anlatılmıştır.

Seyreltme

Seyreltme (dropout) regülerizasyon yöntemi, eğitim sırasında gizli katmanlardaki belirli oranda nöronun seyreltilmesi işlemidir. Seyreltme işlemi, belirlenen orandaki rastgele seçilen nöronların aktivasyonlarının sıfıra uyarlanması ile yapılır. Nöronların seyreltilmesi ile, eğitim farklı nöronları kombinasyonları üzerinden yapılır ve ağırlıklar her biri farklı mimariye sahip ağlar üzerinden hesaplanır (Skansi, 2018). Ağın kapasitesini düşürüldüğünden eğitim daha az zaman alır ve daha az kaynak tüketir. Ayrıca model her defasında farklı nöronlar üzerinden

oluşturulan ağlar üzerinden öğrendiğinden, model öğrenme hem de test verisinde daha yüksek performans göstermektedir. Şekil 27’de gizli katmandaki nöronlardan rastgele seçilerek yapılmış örnek seyreltme işlemini göstermektedir.



Şekil 27. Örnek seyreltme işlemi.

L1 ve L2 regülarizasyonu

Derin öğrenme modelinin eğitimi için kullanılan hata fonksiyonlarının minimize edilmesi sonucunda hata miktarının aşırı azalması ve sıfır değerine yaklaşması modelin aşırı öğrenmesi, eğitim verisini ezberlemesi olarak sonuçlanır. L1 (Lasso) ve L2 (Ridge Regression) regülarizasyon yöntemleri modelinin karmaşıklığını azalmak ve aşırı öğrenme sorununu çözmek için kullanılırlar (Skansi, 2018). L1 ve L2 regülarizasyonunda hata fonksiyonuna skaler bir hiper parametre olarak regülarizasyon parametresi (λ) kullanılır.

$$E = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^n |w_j| \quad (35)$$

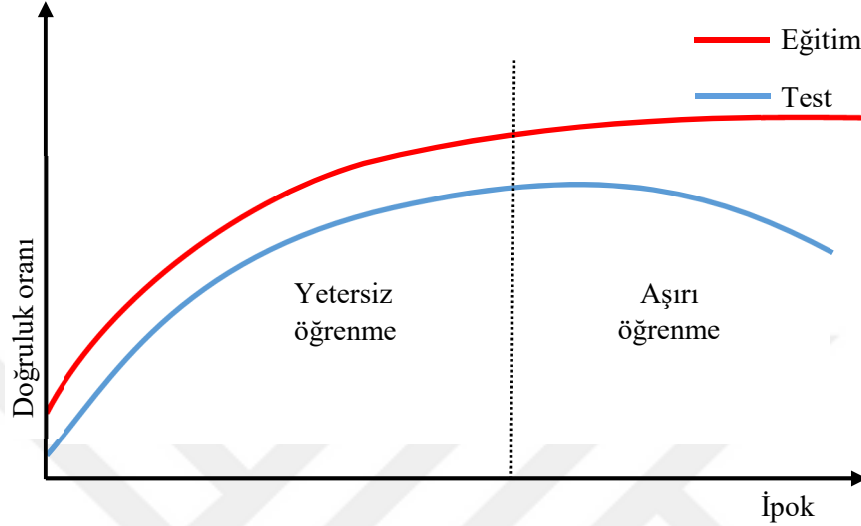
$$E = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^n w_j^2 \quad (36)$$

Denklem 35’te gösterilen L1 regülarizasyon yönteminde ağırlıkların mutlak değer toplamı regülarizasyon parametresi ile çarpılarak, Denklem 36’da gösterilen L2 regülarizasyon yönteminde ise ağırlıkların kare toplamı regülarizasyon parametresi ile çarpılarak, hata değerine eklenmiş ve modelin genelleştirilmesi sağlanmıştır.

L1 regülarizasyonu yönteminde ağırlıkların mutlak değerleri toplamı alındığından, ağırlık değerleri çok küçük olan giriş değerlerinin modeldeki etkisi de az olacak ve böylece öznelik seçimi (feature selection) işlemi yapılmış olacaktır.

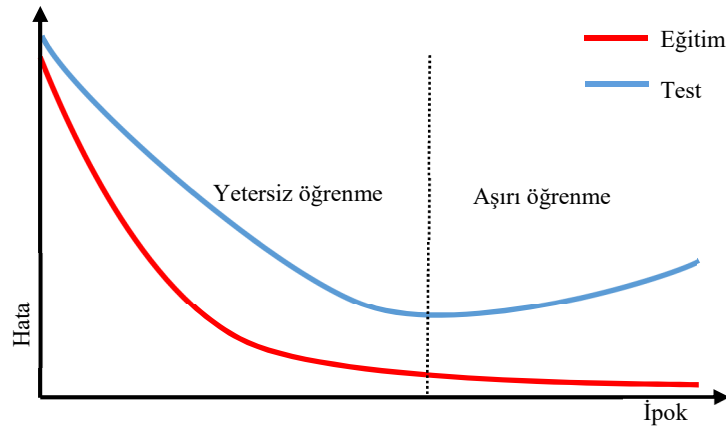
Erken durdurma

Modelin eğitimi için kullanılan ipok sayısının az olması durumunda eksik öğrenme, fazla olması durumunda ise aşırı öğrenme problemi ile karşılaşılır. Erken durdurma (early stopping), modelin aşırı öğrenmesini engellenmek için kullanılan bir regülerizasyon yöntemidir.



Şekil 28. Erken durdurma doğruluk oranı epok ilişkisi.

Şekil 28’de görüldüğü gibi modelin eğitim verilerine doğruluk oranı her ipokta artar ve model eğitim verilerine göre ağırlıkları günceller. Modelin aşırı öğrenmeye başlaması durumunda eğitim verileri ezberleneceğinden, modelin daha önce hiç görmediği test verileri üzerindeki başarısı azalır.



Şekil 29. Erken durdurma hata epok ilişkisi.

Şekil 29’da görüldüğü gibi aşırı öğrenmenin başladığı noktadada, modelin test veri kümesi üzerindeki hata oranı da artmaya başlayacak ve model eğitim verisini ezberlemeye başlayacaktır. Bu yüzden aşırı öğrenmenin başladığı durumda eğitim sonlandırılmalıdır.

Ağırlık başlangıç değerlerinin atanması

Derin öğrenme modelinin eğitimine başlamadan önce ağırlıkların başlangıç değerleri atanır daha sonra ileri ve geri yayılım işlemleri ile ağırlıkların güncellemesi yapılır. Ağırlıkların ilk değerlerinin atanması hem modelin başarısı hem de performansı için hayati önem taşımaktadır. Ağırlıkların yüksek seçilmesi eğitim sırasında gradyanların çok büyük olmasına (gradyan patlaması: exploding gradient problem) aşırı düşük seçilmesi ise gradyanın aşırı düşük olmasına (gradyan kaybolması: vanishing gradient problem) sebep olacaktır. Tüm ağırlıkların aynı seçilmesi ise, katmanlardaki nöronların eğitim sırasında aynı özellikleri öğrenmesine neden olacaktır.

Derin öğrenme modelinin eğitiminde ilk ağırlıkları geleneksel olarak rastgele atanabilir veya modelin türüne ve veri setinin özelliğine göre geliştirilmiş özel başlangıç değeri atama yöntemleri kullanılabilir. Modelin performansının ve doğruluk değerlerinin artırılması için geliştirilen bu yöntemlerden başlıcaları: Denklem 37’de gösterilen Xavier/Gorat ve Denklem 38’de gösterilen Kaiming He yöntemleridir.

$$W^{(l)} = W^{(l)} \cdot \sqrt{\frac{1}{m^{(l-1)}}} \quad \text{veya} \quad W^{(l)} = W^{(l)} \cdot \sqrt{\frac{1}{m^{(l-1)} + m^{(l)}}} \quad (37)$$

$$W^{(l)} = W^{(l)} \cdot \sqrt{\frac{2}{m^{(l-1)}}} \quad (38)$$

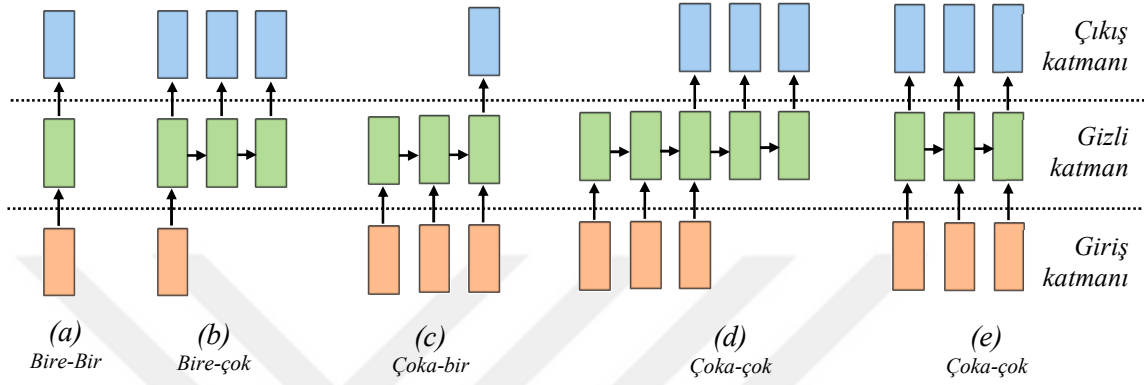
Xavier/Gorat yönteminde normal dağılım veya Gauss dağılım yöntemleri ile oluşturulan ağırlık vektörleri (ortalama=0 ve standart sapması=0,01 gibi) ve ön yargı değerleri 37’deki gibi ölçeklendirilir. Denklemde kullanılan $m^{(l-1)}$ bir önceki katmandaki nöron sayısı, $m^{(l)}$ bir sonraki katmandaki nöron sayısıdır. Xavier/Gorat yönteminin sigmoid ve TanH fonksiyonu ile kullanımı gradyan kaybolması bakımından verimli sonuçlar üretmektedir. Denklem 38’de gösterilen Kaiming He yöntemi ReLu aktivasyonu ile verimli sonuçlar vermektedir ve ölçeklendirme faktörü olarak 2 kullanmaktadır.

Ağırlıkların ilk değerinin atanması için aktivasyon fonksiyonları türünü ve veri setini esas alan yöntemlerin kullanılması, ağırlık performansını ve başarısını olumlu yönde etkileyeceği gibi yukarıda bahsedilen gradyan patlaması ve gradyan kaybolması gibi sorunlara da çözüm sağlar.

RNN (Tekrarlayan sinir ağları)

İleri beslemeli sinir ağlarında her ne kadar gizli katmanların boyutlarında ve sayılarında değişiklikler olsa da; ağırlık sabit boyutlu veri girdisi ve sabit boyutlu çıktısı bulunur. Stok analizi

ve hisse senedi fiyatlarının tahmini gibi yapay sinir ağı uygulamalarda sabit boyutlu veri girişi ve veri çıkışı (Şekil 30 - a Bire-bir), resim etiketleme uygulamalarında sabit boyutlu giriş ve değişken boyutlu veri çıkışı (Şekil 30 - b Bire-çok), duyu analizi uygulamalarda değişken boyutlu veri girişi ve sabit veri çıkışı (Şekil 30 - c çok-bir), doğal dil işleme ve konuşma tanıma, video etiketleme uygulamalarında (Şekil 30 - d çok-çok), ve metin özetleme uygulamalarında değişken veri girişi ve değişken uzunlukta veri çıkışı (Şekil-1e çok-çok) bulunduğu yapay sinir ağı modelinin veri girişi ve tahmin üretme bakımından esnekliğe sahip olması gerekir.



Şekil 30. Tekrarlayan sinir ağlarında veri giriş ve çıkış çeşitlilikleri.

Regresyon ve sınıflandırma problemlerinde oldukça kullanışlı olan ve başarılı sonuçlar üreten ileri beslemeli sinir ağlarında (FFNN) veri, giriş katmanından gizli katmanlara, gizli katmanlardan çıkış katmanına doğru yalnızca ileri yönde ilerler. Ağda dönme-döngü veya geçmişle ilgili bir bağlantı ve hafıza işlemi olmadığından ileri beslemeli sinir ağları sonuçları ağa girilen önceki verilerden bağımsız olarak sadece mevcut giriş verisi kullanılarak hesaplanır. Ses-konuşma tanıma, doğal dil işleme, stok analizi, hisse senedi fiyatlarının tahmini gibi zaman serisi uygulamaları için kullanılan verilerde zaman bileşeni bulunur veya ağda o anda işlenen veriler geçmiş veriler ile bağımlıdır (Yann LeCun & Hinton, 2015).

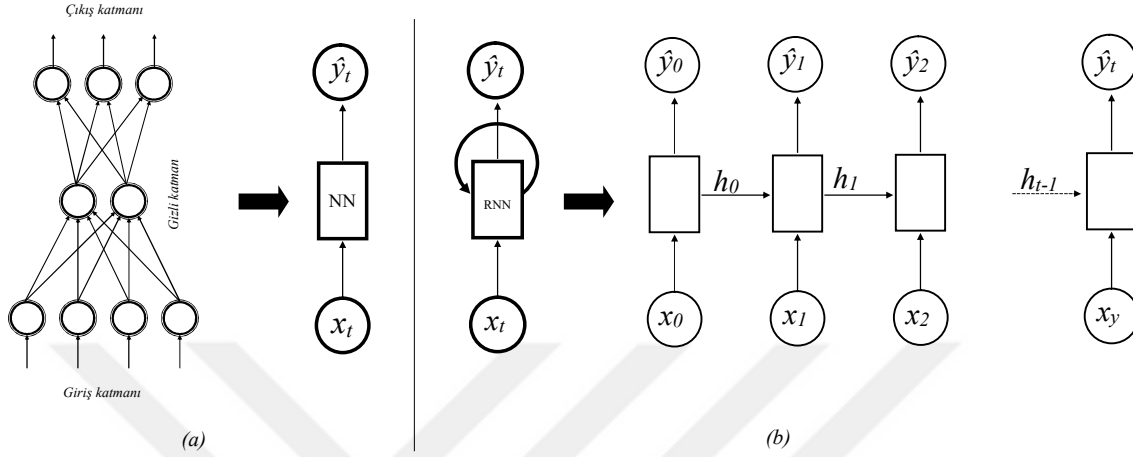
Tekrarlayan sinir ağları (RNN), değişik boyutlarda veri girişi ve veri çıktısı üretebilmesi, sıralı verilerin işlenmesi ve sonuçların tahmin edilmesinde ağa girilen veriler geçmiş verilerin de kullanılmasıyla geliştirilmiş derin öğrenme bir modeldir (Almalaq & Edwards, 2017).

İleri beslemeli yapay sinir ağları ile yapılamayan değişken uzunlukta giriş ve çıkışa sahip sinir ağı modelleri tekrarlayan sinir ağları ile oluşturulabilirler. Tekrarlayan sinir ağına girilen farklı sayıdaki zaman adımları ile değişik uzunlukta verinin ağa girişi sağlanır ve zaman içinde kayıp kontrol edilerek çıkış vektörünün uzunluğu belirlenebilir. Böylece tekrarlayan sinir ağının değişken uzunlukta veri dizisini işleyebilmesi sağlanır.

Tekrarlayan sinir ağlarında, ağın bir önceki ($t-1$) gizli katman değerleri gizli durum (hidden state) olarak (h_{t-1}) ağın t anındaki verisi ile kullanılır ve ağın önceden girilen veriler ile

ilişkisi sağlanır. Tekrarlayan sinir ağlarında çıkış yani tahmin değerleri ağın önceki verileri ve mevcut giriş verisi kullanılarak hesaplanır.

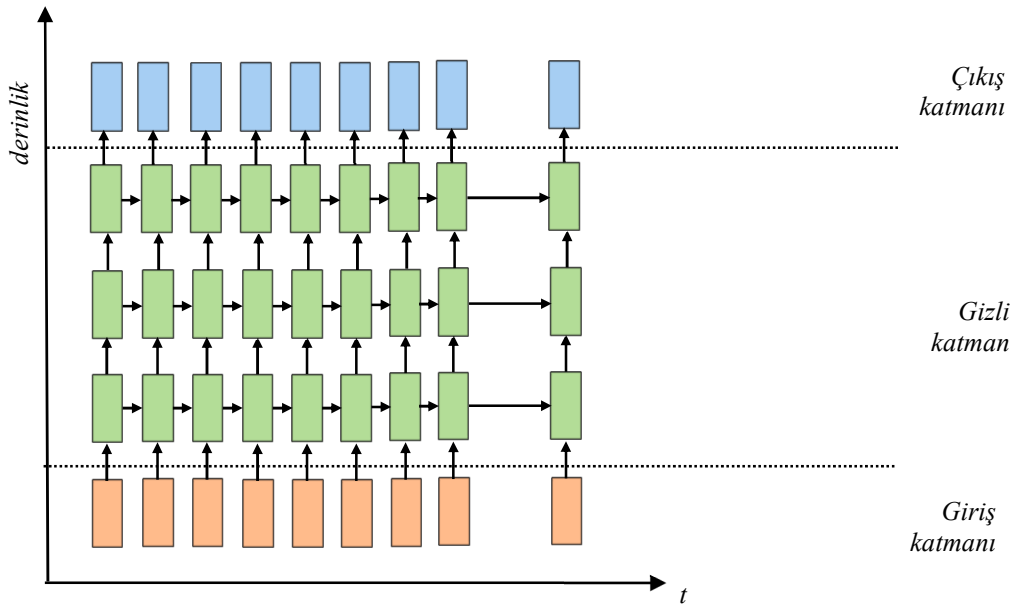
Şekil 31-a’da ileri beslemeli yapay sinir ağının t zamanında giriş katmanı, gizli katman durumu ve çıkış katmanı gösterilmiştir. Şekil 31-b’de tekrarlayan sinir ağı yapısı ve gizli durum değerlerinin bir sonraki gizli katmanlarda giriş verisi olarak kullanıldığı gösterilmiştir.



Şekil 31. Tekrarlayan sinir ağlarının yapısı.

$$\hat{y}_t = f(x_t, h_{t-1}) \quad (39)$$

RNN sinir ağı hücresinde gizli durum ve ağın veri girişi olmak üzere iki giriş (h_{t-1} ve x_t) ve bir çıkış ($\hat{y}_t$) değeri bulunur. Denklem 39’da gösterildiği gibi ağın t anındaki çıkış değeri (y_t), t anındaki giriş değerleri ve bir önceki gizli durumu (h_{t-1}) kullanılarak hesaplanır ve böylece yineleme ilişkisi ağı sağlanmış olur. Denklem 39’da gösterilen f fonksiyonu standart RNN sinir ağı aktivasyon işlemlerini ifade etmektedir.



Şekil 32. Derin tekrarlayan sinir ağı (Deep RNN).

Katman sayısının artırılması ile derin tekrarlayan sinir ağı modeli oluşturulur. Şekil 32’de derin tekrarlayan sinir ağı modeli gösterilmiştir. Derin tekrarlayan sinir ağlarının giriş ve çıkış katmanındaki gizli katmanların birbirleriyle olan ilişkisi derin ileri beslemeli yapay sinir ağlarında olduğu gibi, modelin çıktısının bir sonraki gizli katmandaki RNN modeline giriş değeri olarak alınması ile oluşturulur.

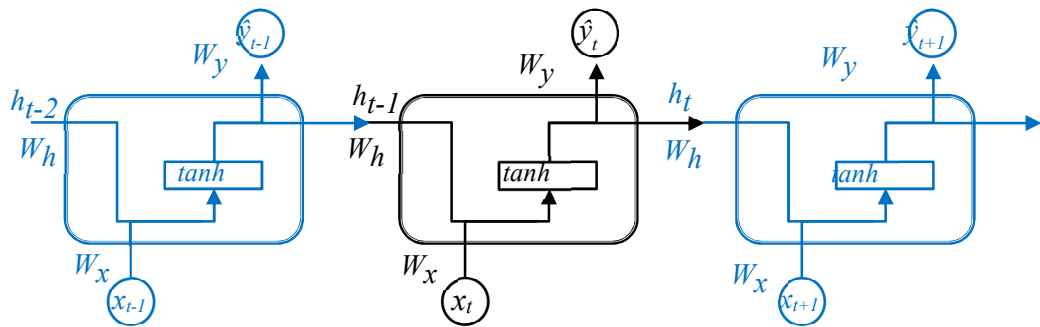
Tekrarlayan sinir ağı giriş, ağırlık ve çıkış vektörlerinin gösterimi ve ağı değerlerinin hesaplama işlemleri RNN Ağına eğitimi başlığının altında “Tekrarlayan Sinir Ağlarında İleri Yayılım” bölümünde anlatılmıştır.

RNN ağına eğitilmesi

RNN modelinin eğitilmesi işleminde ileri beslemeli sinir ağlarında olduğu gibi veri seti modele tekrar tekrar gösterilir ve zamana göre ileri-geri yayılım yapılarak en az hataya sahip ağırlık değeri belirlenir. İleri yayılım işleminde ağırlık değerlerine göre bulunan çıkış değeri gerçek sonuçlar ile karşılaştırılır ve “Derin Öğrenme Modelinin Eğitilmesi” bölümünde anlatılan hata fonksiyonlarına göre ağı hatası tespit edilir. Belirlenen hata değerlerine göre ağıdaki ağırlıklar güncellenir. Ağı eğitimi, önceden belirlenen bir ipok sayısına göre veya hata oranında büyük bir değişiklik olmama durumuna göre sonlandırılır.

Tekrarlayan sinir ağlarında ileri yayılım

Şekil 33’de Tekrarlayan sinir ağı modelinde gösterildiği gibi tekrarlayan sinir ağına girişine (X_t) ait giriş ağırlık vektörü (W_x), gizli durum değerlerine (h_t) ait gizli durum ağırlık vektörü (W_h) ve çıkışına (y_t) ait çıkış ağırlık vektörü (W_y) bulunur. Denklem 40’de gösterildiği gibi giriş değeri giriş ağırlıkları ile, bir önceki gizli durum değeri (h_{t-1}) gizli durum ağırlıkları ile çarpılır ve bu iki sonuç toplanarak aktivasyon fonksiyonuna göre sonucu ile modelin t anındaki gizli durum değeri belirlenir.



Şekil 33. Tekrarlayan sinir ağı.

Ağın t anındaki çıkış yani tahmin değerinin hesaplanması için Denklem 40'ta hesaplanan gizli durumu ve çıkış ağırlık vektörü ile çarpılır. Bu işlem Denklem 41'de gösterilmiştir.

$$h_t = \tanh(W_h \cdot h_{t-1} + W_x \cdot x_t) \quad (40)$$

$$\hat{y}_t = W_y \cdot h_t \quad (41)$$

Derin tekrarlayan sinir ağı modelinin t anındaki gizli durumunun hesaplanması için Denklem 42'de gösterildiği gibi ağın son katmanın gizli durum ağırlıkları bir önceki katmanın gizli durumu ile; giriş ağırlık vektörü ise son katmanın giriş gizli durumu ile çarpılır ve elde edilen değerlerin aktivasyon fonksiyonuna göre sonucu alınır. Derin tekrarlayan sinir ağı modelinin t anındaki tahmin değeri ise Denklem 43'de gösterildiği gibi de Denklem 42'de bulunan gizli durum değeri ile son katmanındaki çıkış ağırlıklarının çarpımıyla hesaplanır.

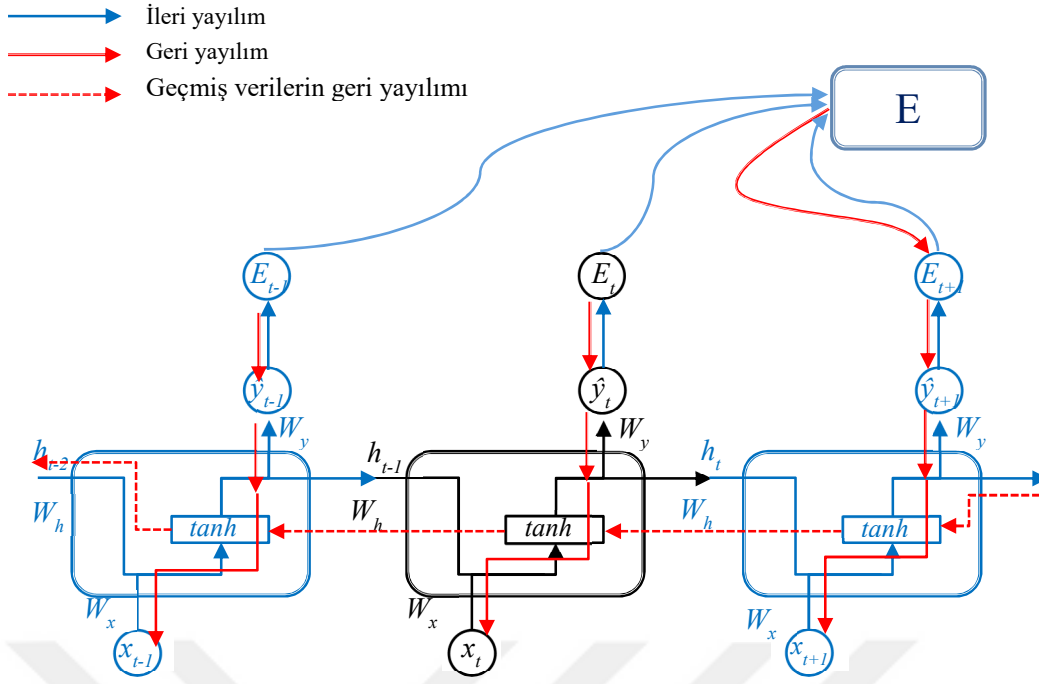
$$h_t^{(k)} = \tanh(W_h^{(k)} \cdot h_{t-1}^{(k-1)} + W_x^{(k)} \cdot h_t^{(k)}) \quad (42)$$

$$\hat{y}_t = W_y^{(k)} \cdot h_t^{(k)} \quad (43)$$

Tekrarlayan sinir ağlarında geri yayılım

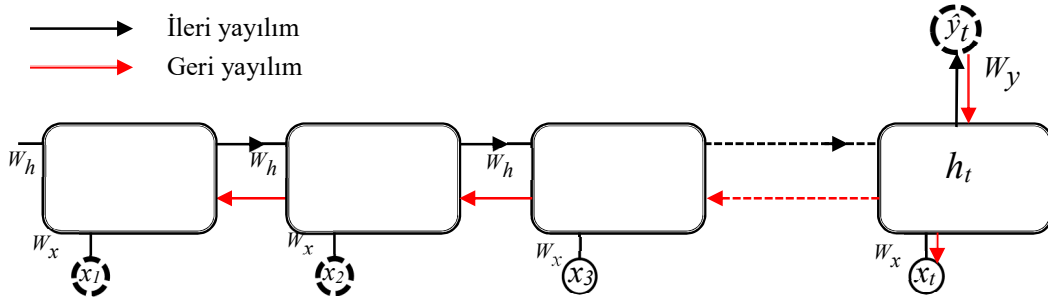
Tekrarlayan sinir ağlarında geri yayılım zamana göre yapılır. Bu işlem için, ileri yayılım sonunda ağın t anında hesaplanan tahmin yani çıktı değeri gerçek çıktı değeri ile karşılaştırılır ve hata fonksiyonlarıyla ağın t zamanındaki hatası tespit edilir. Her bir t anında hesaplanan hata değerleri toplanarak ağın toplam hatası tespit edilir. Ağdaki ağırlıkların güncellenmesi ise ağın her bir t anındaki hata değeri üzerinden değil, Denklem 44 de gösterildiği gibi ağın her bir t anında bulunan hata değerlerinin toplamı üzerinden yapılır.

$$E = \sum_{t=1}^n E_t \quad (44)$$



Şekil 34. Tekrarlayan sinir ağı ileri ve geri yayılım.

Ağırlıkların toplam hatadaki etkisi, her bir ağırlığın toplam hataya göre kısmi türevinin alınması ile hesaplanır. Türev alma işlemi ileri beslemeli derin öğrenme sinir ağlarının geri yayılım bölümünde anlatıldığı gibi zincir kuralı ile yapılır. Her bir ağırlığın türevinin alınması ile gradyan vektörü bulunur ve gradyan vektörüne göre öğrenme oranı kullanılarak ağırlıklar güncellenir. Şekil 34’de tekrarlayan sinir ağının t anındaki hata değerleri, toplam hata değeri ve ağırlıkların zamana göre güncellenmesi gösterilmiştir.



Şekil 35. Tekrarlayan sinir ağlarında gizli durum ağırlıklarının güncellenmesi.

RNN’lerde ağırlıkların zamana göre güncellemeleri yapıldığından gizli durum ağırlık vektörünün (W_h) güncellenmesi için zamana göre yayılım dolayısı ile ağına t_0 anına kadar geri yayılım yapılır. Gizli durum ağırlık değerlerinin t_0 anına kadar güncellenmesi sırasında başlangıçta ağırlıkların küçük seçilmesi ile gradyan kaybolması çok büyük seçilmesi ile gradyan patlaması sorunu oluşur ve geçmiş veriler ile yeni girilen veriler arasındaki ilişki bağı zayıflamış olur. Tekrarlayan sinir ağlarında gizli durum ağırlıklarının güncellenmesi ve ağına t

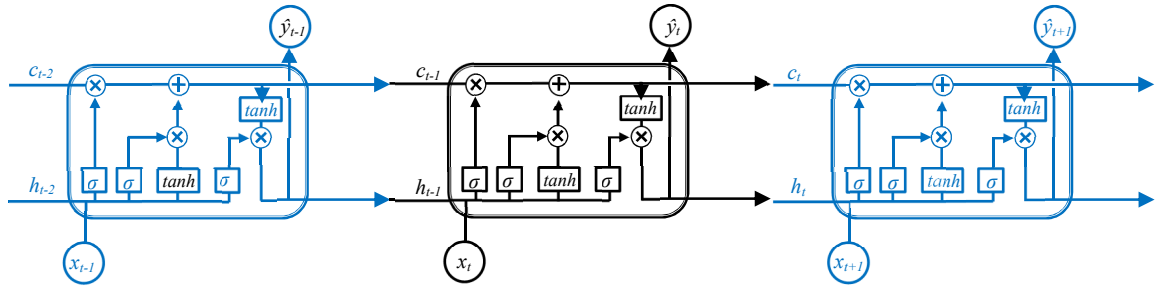
anındaki çıkış değerlerinin ($\hat{y}_t$) ağı t_1 ve t_2 anında girilen değerler ile bağımlı olduğu Şekil 35’de gösterilmiştir.

Gizli durum ağırlıklarının aşırı büyük çıkması durumunda oluşan gradyan patlaması sorununa ağırlıkların uygun başlangıç değerlerinin atanması veya gradyan kırpması (gradient clipping) yapılarak ağırlıkların ölçeklendirilmesi ile çözüm sağlanır. Gradyan kaybolması sorununa ise, ağırlıkların uygun başlangıç değerlerinin atanması ve uygun aktivasyon fonksiyonunun kullanılması veya Uzun Kısa Süreli Bellek Ağları (LSTM) ve Kapılı Tekrarlayan Birim (GRU) modelleri ile çözüm sağlanır. LSTM ve GRU modellerinde kullanılan kapılar verinin ne kadarının bir sonraki hücreye aktarılacağına, aday değerler ise hangi verinin bir sonraki hücreye aktarılacağını belirler. Verinin ne kadarının önemli olduğunu belirleyen kapılar için 0-1 arasında değerler üreten sigmoid fonksiyonları ve hangi verinin aktarılacağını belirleyen aday değerler için ise aktivasyon fonksiyonları kullanılır.

Tekrarlayan sinir ağlarında veri setinde toplulaştırma (batch) yapılmış ise ve her toplulaştırılmış veri diğer toplulaştırılmış veriler ile ilgili ise (fiyat, stok ve tüketim analizi gibi uygulamalarda) durumlu (stateful) modunda, toplulaştırılmış veriler diğer toplulaştırılmış veriler ile ilgili değilse (doğal dil işleme vb uygulamalarda) durumsuz(stateless) modunda tekrarlayan sinir ağı modeli kullanılır. Stateful modundaki modelde toplu işin sonunda elde edilen gizli durum ağırlıkları (h_t) bir sonraki batch için ilk gizli durum ağırlığı (h_0) olarak kullanılır. Stateless modundaki modelde ise her toplu iş için ilk gizli durum ağırlığı bir önceki batch’den bağımsız (h_0) rastgele olarak belirlenir (Fang & Yuan, 2019).

Uzun kısa süreli bellek (UKSB) modeli

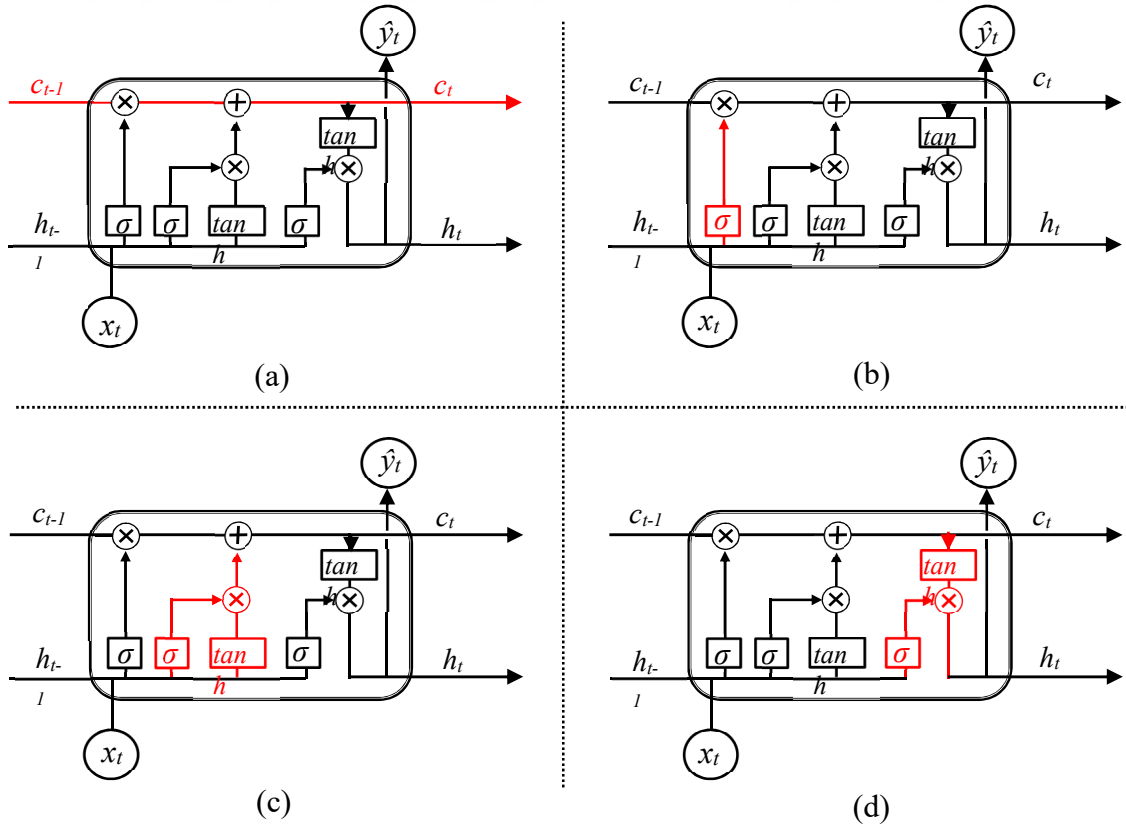
Uzun kısa süreli bellek (LSTM) modeli, ağı daha önceden girilen verilerden ilgisiz olanları unutmayı ve ilgili olanları hatırlamayı esas alan ve gradyan kaybolması sorununu çözen tekrarlayan sinir ağı modelidir (Olah, 2015). LSTM modeli yapısal olarak standart RNN gibi tekrarlayan modüllerden oluşur ve ağı daha önceden girilen verilerle ilişki kurmak için ağı bir önceki ($t-1$) gizli durum değerlerini (h_{t-1}) kullanır. LSTM, standart RNN den farklı olarak hangi verinin hatırlanması ve hangi verilerin unutulması gerektiğini hücre durumu (c_t) bellek mekanizması (memory cell) ile tutar ve eğitimin her bir t anında verilerin ilişkisine göre hücre durumu güncellenir. Hücrenin bir önceki durumunun (c_{t-1}), ağı girilmesi ve her bir t anında güncellenmesi ile ağı uzun süreli bellek yeteneği sağlanmış olur ve gradyan kaybolması sorununa çözüm sağlanır (Hochreiter & Schmidhuber, 1997). Şekil 36’daki Uzun Kısa Süreli Bellek Modelinde hücrelerin gizli durumu ve hücre durumu değerlerinin zamana bağlı olarak birbirleriyle olan ilişkileri gösterilmiştir.



Şekil 36. Uzun kısa süreli bellek modeli.

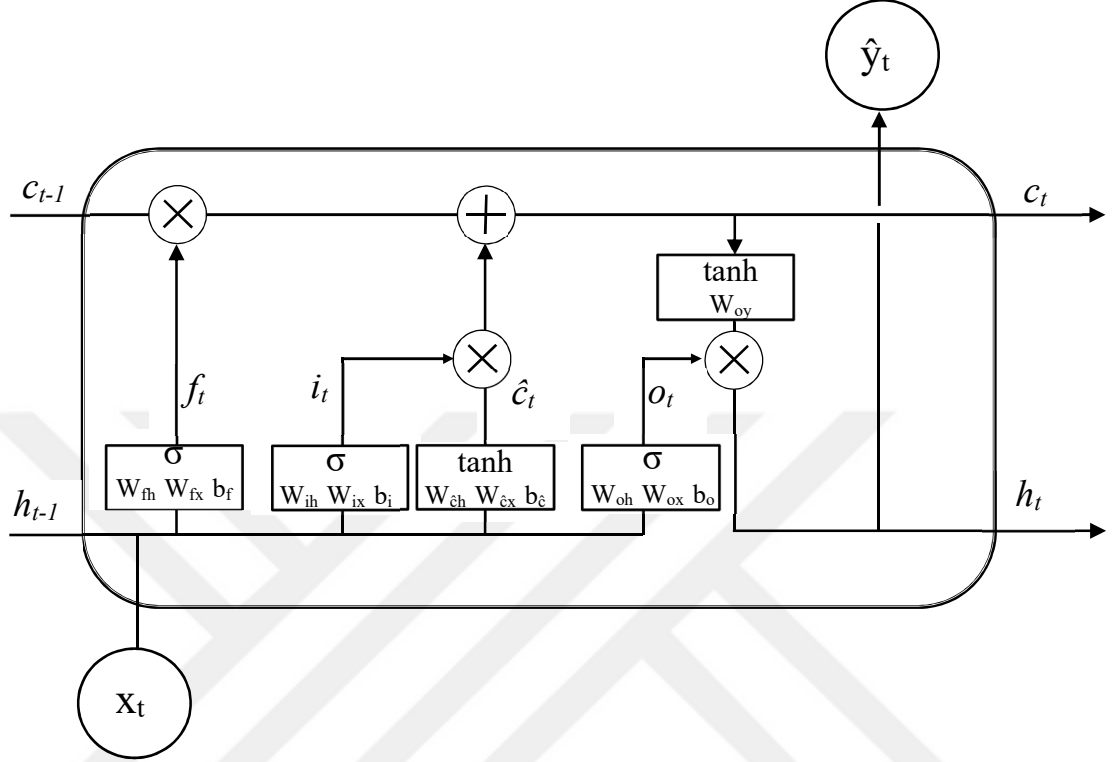
LSTM Modülünde uzun kısa süreli bağımlılığı sağlayan hücre durumu(ct) (Şekil 37-a), alakasız geçmişin hücre durumundan çıkarılmasını sağlayan unutmak kapısı (Şekil 37-b), alakalı yeni bilgilerin hücre durumuna eklenmesini sağlayan giriş kapısı (Şekil 37-c) ve bir sonraki modelin gizli durumunun üretilmesini sağlayan çıkış kapısı (Şekil 37-d) bulunur. LSTM modülü kapıları sigmoid ve aktivasyon (tanh aktivasyon fonksiyonu kullanılarak anlatılacaktır.) fonksiyonları kullanılarak oluşturulan sinir ağlarıdır. Sigmoid fonksiyonları hangi verilerin unutulması yada kullanılması gerektiğini, tanh fonksiyonları ise verilerin ne kadar kullanılması gerektiğini belirler.

LSTM modelinde bulunan kapılar birer sinir ağını (tek katmanlı ve tek nöronlu sinir ağını) temsil eder. Sigmoid fonksiyonların kullanılması ile ağı eğitimi sırasında hangi verinin unutulacağı hangi verinin ağı ekleneceği gibi kararlara ağı kendi kendini eğiterek karar verir. Kapıların çıkış boyutu hücre durumu ve gizli katman durumu boyutundadır.



Şekil 37. Uzun kısa süreli bellek hücre durumu ve kapıları.

LSTM hücresinin her bir kapısında giriş ve gizli durumuna karşılık gelen ağırlık vektörleri bulunur. Kapıların değerleri; kapı ağırlık vektörleri, giriş değerleri ve gizli durum değerleri kullanılarak hesaplanır. Şekil 38’de Uzun Kısa Süreli Bellek modeli ve ağırlık vektörleri gösterilmiştir.



Şekil 38. Uzun kısa süreli bellek modeli ağırlıkları.

Unutma kapısı (forget gate), ağda gerçek değerler ile ilişkili olan verilerin tutulmasını, ağda gerçek değerler ile ilişkili olmayan verilerin ise unutulmasını ve hücre durumundan çıkarılmasını sağlar. Unutma kapısı (f_t) için, ağın bir önceki gizli durum değerleri (h_{t-1}) ile bu değerlerin unutma kapısı ağırlıkları (W_{fh}) ; ağın giriş değerleri kapının giriş ağırlıkları (W_{fx}) ile çarpılır ve çıkan sonuç unutma kapısının biası (b_f) ile toplanır. Elde edilen sonuca sigmoid fonksiyonu uygulanır ve unutma kapısı belirlenmiş olur. Denklem 42’de unutma kapısının hesaplanması gösterilmiştir.

$$f_t = \sigma [(W_{fh} \cdot h_{t-1}) + (W_{fx} \cdot x_t) + b_f] \quad (42)$$

Unutma kapısında kullanılan sigmoid fonksiyonu sonucunda, ağın her bir t anında ağa girilen verilerin gerçek değerler ile ilişkisine göre 0-1 arası değerler oluşur. Ağa girilen verinin gerçek değer ile ilgisinin olmadığı durumlarda 0 veya 0’a yakın değer, giriş verisinin gerçek değer ile ilişkili yani bağımlılığı olan durumlarda ise 1 e yakın değerler elde edilir. Veri setindeki ilişkisiz girişlere karşılık gelen değer 0 veya 0 a yakın olacağından, zaman içinde ilişkisiz ve devamlı değişen giriş değerlerinin ağırlıkları unutulacak; sadece ilişkili olan giriş değerlerinin ağırlıkları hatırlanmış olacaktır.

LSTM modülünde ağa yeni girilen verilerin hücre durumuna eklenmesi için giriş kapısı (i_t) ve aday hücre durumu ($\hat{c}_t$) değerlerini kullanır. Giriş kapısı (i_t) t anında ağa girilen verilerin hücre durumuna eklenip eklenmeyeceğini tespit etmek için, aday hücre durumu ($\hat{c}_t$) ise ağa yeni girilen verilerin hücre durumuna ne kadarının ekleneceğinin tespit edilmesi için kullanılır.

Giriş kapısı değeri için, ağın bir önceki gizli durum değerleri (h_{t-1}) ile bu değerlerin giriş kapısı ağırlıkları (W_{ih}); ağın giriş değerleri ile giriş kapısının ağırlıkları (W_{ix}) çarpılır ve ile çarpılır ve çıkan sonuç giriş kapısının biası (b_i) ile toplanır. Elde edilen sonuca sigmoid fonksiyonu uygulanır ve giriş kapısı belirlenmiş olur. Denklem 43'te giriş kapısının hesaplanması gösterilmiştir. Giriş kapısının (i_t) hesaplanması sonucunda sigmoid fonksiyonu kullanılması ile 0 – 1 aralığında değer elde edilir. Bu değer ağa yeni girilen verinin gerçek değerler ile ilişkisini gösterir. Ağa yeni girilen verilerin gerçek değerler ile ilişkili olduğu durumlarda sigmoid fonksiyonu 1 veya 1 e yakın değer, girilen veri gerçek değerler ile ilişkili olmadığı durumlarda ise sigmoid fonksiyonu 0 veya 0'a yakın değer üretir. Böylece yeni girilen veri ağırlıklarının hücre durumuna ne miktarda ekleneceği tespit edilmiş olur.

$$i_t = \sigma [(W_{ih} \cdot h_{t-1}) + (W_{ix} \cdot x_t) + b_i] \quad (43)$$

$$\hat{c}_t = \tanh [(W_{ch} \cdot h_{t-1}) + (W_{cx} \cdot x_t) + b_c] \quad (44)$$

Ağın hücre durumunun (c_t) tespit edilmesi için önce aday hücre durum ($\hat{c}_t$) değeri hesaplanır ve hücre durumu (c_t) güncellenmesi için hesaplanan bu aday hücre durumu değeri, unutmama kapısı (f_t) ve giriş kapısı (i_t) kullanılır. Aday hücre durum değeri ($\hat{c}_t$) denklem 44'de gösterildiği gibi ağın bir önceki gizli durum değerleri (h_{t-1}) ile aday hücre durum ağırlıkları (W_{ch}) ile, ağın giriş değerleri aday hücre durum ağırlıkları (W_{cx}) ile çarpılır ve çıkan sonuç aday hücre durumu biası (b_c) ile toplanır ve sonuca tanh fonksiyonu uygulanır.

Hücre durumunun (c_t) hesaplanması için ağın bir önceki t anındaki hücre durumu ile (c_{t-1}) unutmama kapısı vektörü eleman bazlı çarpılır ve ilgisiz olan verilerin hücre durumundan çıkarılması, ilgili olanların ise hücre durumunda tutulması sağlanır. Giriş kapısı aday veri değerlerinin hücre durumuna eklenmesi için ise giriş kapısı (i_t) ile aday hücre durumu değerleri ($\hat{c}_t$) eleman bazlı çarpılır ve hücre durumuna eklenecek veriler bulunmuş olur. Elde edilen her iki vektörün toplanması ile ağın t anında hücre durumu güncellenmiş olur. Denklem 45'de hücre durumu vektörünün güncellenmesi işlemi gösterilmiştir.

$$c_t = f_t \circ c_{t-1} + i_t \circ \hat{c}_t \quad (45)$$

Ağın tahmin değerleri ve ağın bir sonraki anında ($t+1$) kullanılacak gizli durum değerleri (h_t) çıkış kapısı (output gate) tarafından üretilir. Çıkış kapısı değerlerinin (o_t) bulunması için, ağın bir önceki gizli durum değerleri (h_{t-1}) ile bu değerlerin çıkış kapısı ağırlıkları (W_{oh}); ağın giriş değerleri ile kapının çıkış ağırlıkları (W_{ox}) çarpılır ve çıkan sonuç çıkış kapısının biası (b_o)

ile toplanır. Elde edilen sonuca sigmoid fonksiyonu uygulanır ve çıkış kapısı belirlenmiş olur. Denklem 46’da çıkış kapısının hesaplanması gösterilmiştir.

$$o_t = \sigma[(W_{oh} \cdot h_{t-1}) + (W_{ox} \cdot x_t) + b_o] \quad (46)$$

Ağın gizli durum değerlerinin bulunması için Denklem 47’de gösterildiği gibi güncellenmiş olan gizli durumun çıkış ağırlık vektörü ile (W_{oy}) çarpımının tanh fonksiyonuna göre değerleri alınır ve böylece çıkış verilerinin -1 ve +1 aralığına alınması sağlanır. Elde edilen sonuç çıkış kapısı vektörü ile eleman bazlı çarpılır ve çıkış kapısındaki ağırlıklara göre ağın gizli durum değerleri (h_t) tespit edilmiş olur. LSTM ağının t anındaki gizli durum değerleri ağın bir sonraki anında giriş gizli durum değeri olarak kullanır ve bu değerler Denklem 48’de gösterildiği gibi ağın t anındaki tahmin değerleridir.

$$h_t = o_t \circ \tanh(W_{oy}c_t) \quad (47)$$

$$\hat{y} = h_t \quad (48)$$

LSTM modelinin eğitilmesi için ileri-geri yayılım yapılarak en az hataya sahip ağırlık değerleri belirlenir. Geri yayılım işleminde standart RNN modelinde olduğu gibi t anındaki hata değeri üzerinden değil, hesaplanan toplam hata üzerinden yapılır. LSTM modelinde bulunan bütün ağırlıkların ($W_{fh}, W_{fx}, b_f, W_{ih}, W_{ix}, b_i, W_{ch}, W_{cx}, b_c, W_{oh}, W_{ox}, b_o, W_{oy}$) toplam hatadaki etkisi, her bir ağırlığın toplam hataya göre kısmi türevinin alınması zincir kuralı ile hesaplanır. Ağırlıklarda yapılan güncellemelerde standart RNN modelinde olduğu gibi t_0 anına kadar zamana göre geri yayılım olmadığından standart RNN modelinde yaşanan gradyan kaybolması veya gradyan patlaması sorunu çözülmüş olur.

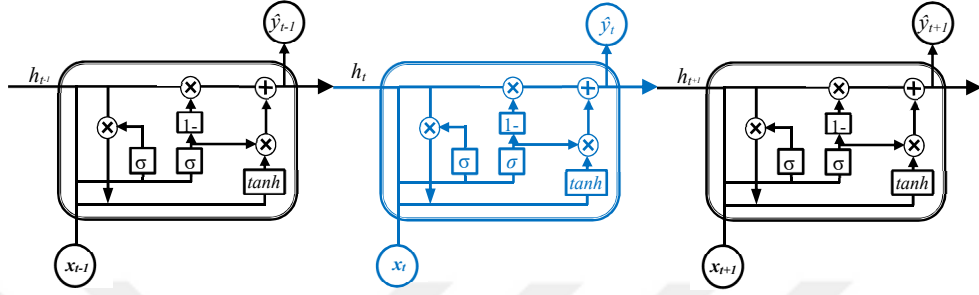
LSTM modelinde anlatılan tanh aktivasyon fonksiyonları yerine “Derin Öğrenme Modelinin Oluşturulması” bölümünde bahsedilen farklı aktivasyon fonksiyonları da kullanılmaktadır. Aktivasyon fonksiyonlarının başarısı verinin türüne ve çıkış türüne göre değişiklik göstermektedir (Fang & Yuan, 2019).

GRU

LSTM modelinde kapı sayısının fazla olmasından ve geri yayılım sürecinde güncellenmesi gereken birçok parametre bulunması LSTM modelinin eğitim süresinin uzun olmasına sebep olmaktadır (Yang ve diğerleri, 2020). Bununla birlikte LSTM modelinin karmaşık yapısından dolayı uygulamasında zorluklar yaşanmaktadır. LSTM modelinde yapılan iyileştirmeye geliştirilmiş olan GRU modeli, iki kapı ve bir gizli durum değeri ile tasarlanmış LSTM’in basitleştirilmiş bir versiyonudur (Fu ve diğerleri, 2016). GRU modelinin yapısal

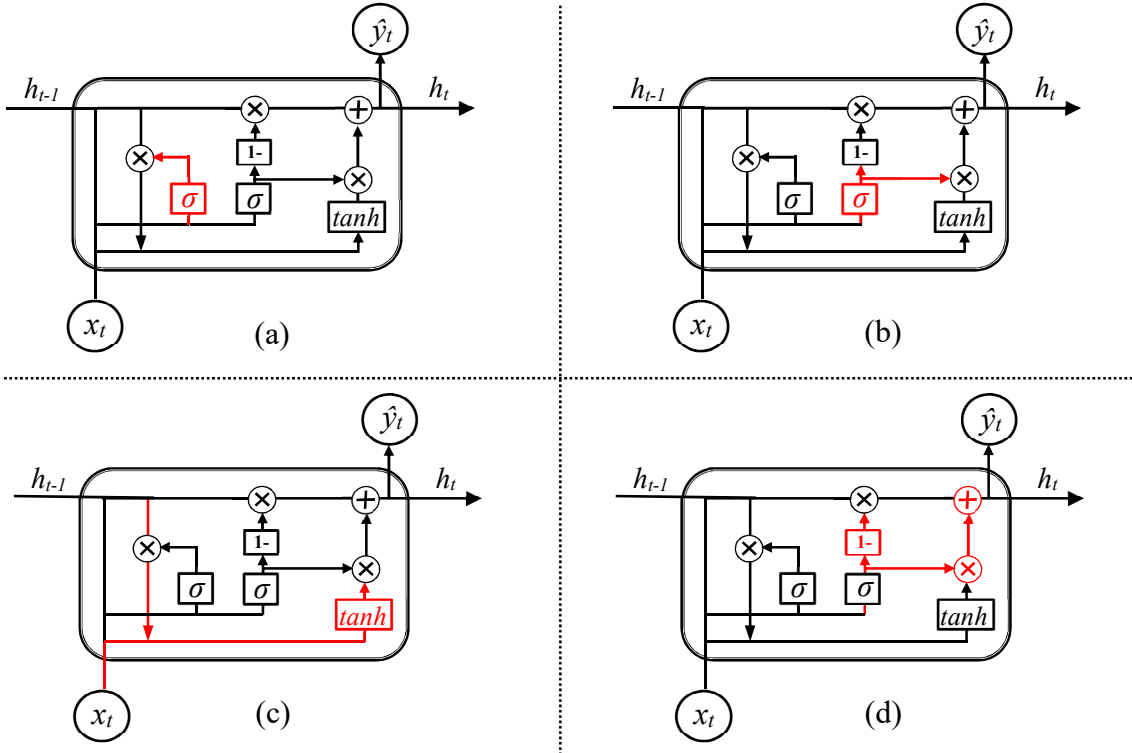
olarak daha basit olması uygulamada kolaylıklar sağladığı gibi modelin öğrenme hızını da artırmakta ve LSTM modeline göre avantajlar sağlamaktadır.

GRU modelinin sıfırlama kapısı (reset gate) ve güncelleme kapısı (update gate) bir önceki gizli katman durumunun (h_{t-1}) GRU ünitesinden bir sonraki GRU ünitesine (h_t) geçişini kontrol ederler. Şekil 39’ de Kapılı Tekrarlayan Ünite (GRU) Modelinde hücre durumu değerlerinin zamana bağlı olarak birbirleriyle olan ilişkileri gösterilmiştir.



Şekil 39. Kapılı tekrarlayan ünite (GRU) modeli.

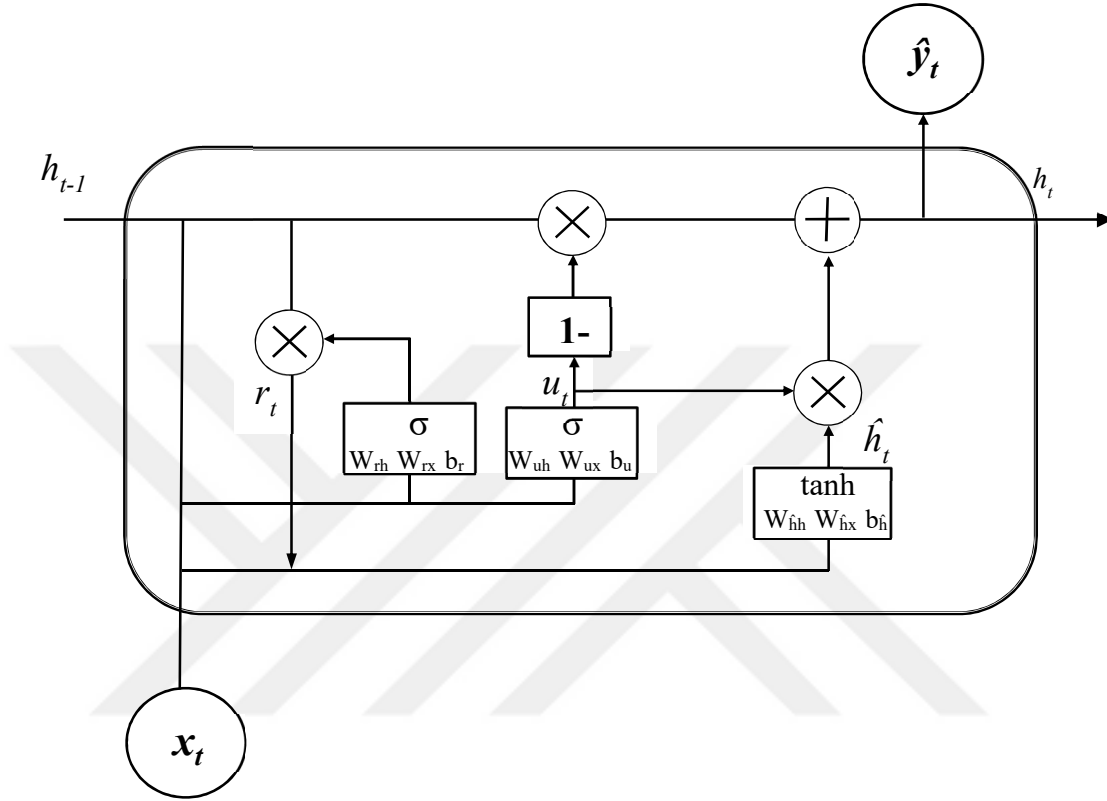
GRU Modülünde, alakasız geçmişin ağır gizli durumundan çıkarılmasını sağlayan sıfırlama kapısı (Şekil 40-a), alakalı yeni bilgilerin ağır gizli durumuna eklenmesini sağlayan giriş kapısı (Şekil 40-b), bir sonraki modelin gizli durumunun üretilmesini sağlayan aday gizli durum değeri (Şekil 40-c) bulunur. GRU Modülünde ağır gizli durumunun güncellenmesi LSTM modelinde olduğu gibi yeni bir kapı ile değil, giriş kapısının tamlayanı (Şekil 40-d) alınarak yapılır.



Şekil 40. Kapılı tekrarlayan ünite (GRU) modeli gizli durum ve kapıları.

LSTM modülünde olduğu gibi GRU modelinde kullanılan sigmoid fonksiyonları hangi verilerin unutulması yada kullanılması gerektiğini, tanh fonksiyonları ise verilerin ne kadar kullanılması gerektiğini belirler.

GRU hücresinin kapı değerleri hesaplanırken her bir kapıya ait ağırlık değerleri kullanılır. Şekil 41’de GRU modeli ve ağırlık vektörleri gösterilmiştir.



Şekil 41. Kapılı tekrarlayan ünite (GRU) modeli ağırlıkları.

LSTM unutmaya kapısı gibi çalışan sıfırlama kapısı (reset gate) yeni bilgilerin belleğe nasıl ekleneceğine, yani bilgilerin ne kadarının unutulabileceğini belirlemek için kullanılır. Sıfırlama kapısı (r_t) için, ağırlık bir önceki gizli durum değerleri (h_{t-1}) ile bu değerlerin sıfırlama kapısı ağırlıkları (W_{rh}) ; ağırlık giriş değerleri ile kapının giriş ağırlıkları (W_{rx}) ile çarpılır ve çıkan sonuç unutmaya kapısı biası (b_r) ile toplanır. Elde edilen sonuca sigmoid fonksiyonu uygulanır ve sıfırlama kapısı belirlenmiş olur. Denklem 49’de sıfırlama kapısının hesaplanması gösterilmiştir.

$$r_t = \sigma[(W_{rh} \cdot h_{t-1}) + (W_{rx} \cdot x_t) + b_r] \quad (49)$$

Sıfırlama kapısında kullanılan sigmoid fonksiyonu sonucunda, ağa girilen verinin gerçek değer ile ilgisinin olmadığı durumlarda 0 veya 0’a yakın değerler, giriş verisinin gerçek değer ile ilişkili yani bağımlılığı olan durumlarda ise 1 e yakın değerler elde edilir böylece ilişkili verilerin ağırlıklarının ağda kalması ve ilişkisiz verilerin sıfırlanması yani unutulması sağlanır.

Güncelleme kapısı (update gate) önceki gizli durum değerinden (h_{t-1}) hangi bilgilerin bir sonraki adıma (h_t) iletebileceğinin belirlenmesi için kullanılır. Güncelleme kapısı (u_t) için, ağırlıklı bir önceki gizli durum değerleri (h_{t-1}) ile bu değerlerin güncelleme kapısı ağırlıkları (W_{uh}) ; ağırlıklı giriş değerleri ile kapının giriş ağırlıkları (W_{ux}) ile çarpılır ve çıkan sonuç güncelleme kapısı biası (b_u) ile toplanır. Elde edilen sonuca sigmoid fonksiyonu uygulanır ve sıfırlama kapısı belirlenmiş olur. Denklem 50’de sıfırlama kapısının hesaplanması gösterilmiştir.

$$u_t = \sigma[(W_{uh} \cdot h_{t-1}) + (W_{ux} \cdot x_t) + b_u] \quad (50)$$

Ağırlıklı gizli durum değerlerinin (h_t) güncellenmesi için önce aday gizli durum ($\hat{h}_t$) değerleri hesaplanır ve hesaplanan bu aday gizli durumu değerleri ve güncelleme kapısı (u_t) kullanılarak gizli durum değerinin güncellenmesi yapılır. Aday gizli durum değerleri ($\hat{h}_t$) denklem 51’de gösterildiği gibi ağırlıklı bir önceki gizli durum değerleri (h_{t-1}) sıfırlama kapısı (r_t) ile eleman bazlı çarpılır ve ağırlıklı sonraki gizli katmanlarına iletmeyecek veriler gizli durumdan çıkarılmış olur. Daha sonra elde edilen vektör aday gizli durum ağırlıkları ($W_{\hat{h}h}$) ile, ağırlıklı giriş değerleri aday gizli durum ağırlıkları ($W_{\hat{h}x}$) ile çarpılır ve çıkan sonuç aday gizli durumu biası ($b_{\hat{h}}$) ile toplanır ve daha sonra elde edilen sonuca tanh fonksiyonu uygulanır.

$$\hat{h}_t = \tanh[(W_{\hat{h}h} \cdot r_t \cdot h_{t-1}) + (W_{\hat{h}x} \cdot x_t) + b_{\hat{h}}] \quad (51)$$

Gizli durum değerlerinin (h_t) güncellenmesi için gizli durum değerlerinin bir önceki değerleri (h_{t-1}) ve aday gizli durum ($\hat{h}_t$) değerleri kullanılır. Gizli durum değerlerinin nasıl güncelleneceğini ise güncelleme kapısı belirler. Gizli durum değerleri denklem 37’de gösterildiği gibi güncelleme kapısı değerleri (u_t) ile aday gizli durum ($\hat{h}_t$) değerlerinin çarpımına, güncelleme kapısının tamlayanı ($1-u_t$) ile bir önceki gizli durum değerinin (h_{t-1}) değerinin çarpımının eklenmesi ile bulunur. Gizli durum değerlerinin (h_t) güncellenmesi sırasında güncelleme kapısı değerlerinin (u_t) 1 veya 1’e yakın olması durumunda aday gizli durum ($\hat{h}_t$) değerleri, 0 veya 0’a yakın olduğu durumlarda ise bir önceki gizli durum değerleri (h_{t-1}) bir sonraki gizli durum değerleri olarak kullanılır.

GRU ağırlıklı t anındaki gizli durum değerleri standart RNN ve LSTM modelinin modelinde olduğu gibi ağırlıklı bir sonraki anında giriş gizli durum değeri olarak kullanılır. Denklem 53’de gösterildiği gibi ağırlıklı t anındaki tahmin değerleri ($\hat{y}_t$) gizli durum değerleridir.

$$h_t = u_t \cdot \hat{h}_t + (1 - u_t) \cdot h_{t-1} \quad (52)$$

$$\hat{y}_t = h_t \quad (53)$$

GRU modelinin eğitilmesi için ileri ve geri yayılım işlemleri yapılır ve geri yayılım işleminde ağırlıklı her bir t anındaki hatasına göre değil toplam hataya göre ağırlıklar güncellenir.

Ağırlıkların güncellenmesi zamana göre geri yayılım ile yapılmadığından gradyan kaybolması veya gradyan patlaması sorunu çözülmüş olur.

GRU modeli ağırlıklarının LSTM modeline göre daha az olmasından dolayı daha hızlı eğitilmektedir (Yang ve diğerleri, 2020). Tekrarlayan sinir ağı uygulamalarında oldukça popüler olan LSTM ve GRU modellerinin doğruluk değerleri problemin ve veri setinin türüne göre değişiklikler göstermektedir.

Veri setinin hazırlaması

Bu bölümde Derin Öğrenme ağının daha performanslı eğitilebilmesi için veri setinde bulunan eksik verilerin düzenlenmesi, veri setindeki uyumsuzlukların giderilmesi, veri setinin ölçeklendirilmesi-normalleştirilmesi gibi konulara yer verilecektir.

Eksik verilerin tamamlanması ve aykırı değerlerin düzeltilmesi

Makine öğrenmesi uygulamalarında başarılı tahminler üretilebilmesi için veri setinin tam ve güvenilir olması gerekmektedir. Veri setinde eksik öznitelik kayıtların ya veri setinde çıkarılması veya eksik olan özniteliklerin diğer özniteliklerle uyumlu olarak tamamlanması gerekmektedir (Kaiser, 2014). Zaman serisi uygulamalarında eksik olan özniteliklerin bulunduğu kayıtların silinmesi veri setinde uyumsuzluklara neden olacağından, eksik özniteliğin önceki ve sonraki değerlerinden faydalanılarak üretilmesi işlemi yapılır (Krizhevsky ve diğerleri, 2012). Eksik özniteliklerin üretilmesi; seri ortalaması (series mean) yakın noktaların ortalaması(mean of nearby points), yakın noktaların medyanı (median of nearby points), doğrusal enterpolasyon (linear interpolation) ve noktadaki doğrusal eğilim (linear trend at point) istatistiksel yöntemler ile tamamlanır. Denklem 53’de doğrusal enterpolasyon, Denklem 54’de doğrusal Enterpolasyon eğimi, Denklem 55 ve Denklem 56’da ise veri seti doğrusal Enterpolasyon işlemi gösterilmiştir.

$$y_i = y_k + m(x_i - x_k) \quad (53)$$

$$m = \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \quad (54)$$

$$\frac{f(x_k) - f(x_i)}{x_k - x_i} = \frac{f(x_k) - f(x_{k+1})}{x_k - x_{k+1}} \quad (55)$$

$$f(x_i) = f(x_k) + \frac{f(x_{k+1}) - f(x_k)}{x_{k+1} - x_k} (x_i - x_k) \quad (56)$$

Veri seti aykırı değerlerin düzeltilmesi

Modelin başarısını etkileyen önemli etkenlerden bir diğeri de veri setindeki aykırı (outlier) değerlerdir (Rashid & Waheed, 2020). Aykırı değerler veri toplama ve veri üreten

ekipmanlardan kaynaklanan sorunlar olabilmektedir. Veri setinde aykırı öznitelik modelin performansını etkileyecek ve yanlış tahminler üretilmesine sebep olacaktır. Veri setindeki aykırı özniteliklere ortalama medyan, mod gibi istatistiki yöntemlerle yeni değer ataması (imputation) yapılabilir; aykırı özniteliklerin sınırlandırılması (winsorization) yapılabilir veya aykırı özniteliklerin dönüştürülmesi (transformation) yapılabilir.

Veri ölçekleme ve normalleştirme

Normalleştirme kavramı, özellik ölçekleme (feature scaling), normalize etme / olağanlaştırma (normalisation), standardize etme (standardisation) olarak ta adlandırılmaktadır. Normalleştirme, verilerin hazırlanması sırasında verinin farklı aralıklara sahip olduğu durumlarda ortak bir ölçek kullanmak için bir veri setinin sayısal sütunların değerlerini değiştirmek için uygulanan bir yöntemdir. Çoğunlukla x' ile gösterilir. Regresyon uygulamalarında sıklıkla min-max ve fix zero normalleştirme yöntemleri kullanılmaktadır. Denklem 57'de Min-Max, Denklem 58'de Fix Zero yöntemi ile normalizasyon ve ölçeklendirme işlemi gösterilmiştir.

$$x' = \frac{x - \min}{\min - \max} \quad (57)$$

$$x' = \frac{x}{\max} \quad (58)$$

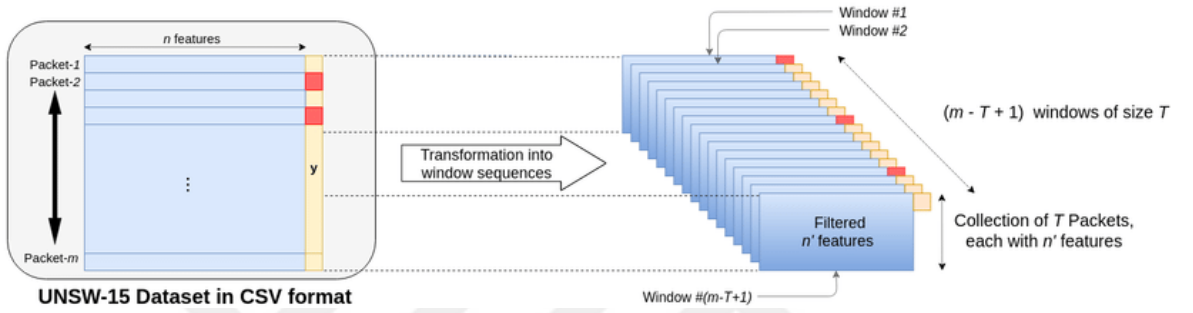
Veri setinin bölünmesi

Veri setinde bulunan verinin bir kısmı eğitim (training data) için, bir kısmı doğrulama (validation data), bir kısmı da modelin başarısını test (test data) etmek için kullanılır. Ağ eğitilmeğe başlanırken her bir nöron için ağırlık değerleri üretilip, ileri yayılım ve geri yayılım işlemi ile ağırlıkların doğru değerlerinin bulunması yani ağın eğitilmesi sağlanır. Bu işlem için ağın eğitim verileri kullanılır. Ağın eğitimi sırasında üretilen nöron ağırlıklarının gerçek değerlerle ilişkisinin karşılaştırılması için doğrulama verileri, eğitim tamamlandıktan sonra ağın performansının tespit edilmesi için ise test verileri kullanılır.

Veri setinin çerçeve dizisine dönüştürülmesi

RNN, LSTM ve GRU zaman serisi modelleri veri dizilerini giriş ve çıkış olarak kullanan makine öğrenme modelleridir. Model veri giriş ve çıkışının belirli bir formatta yapılması gerekmektedir (Singh ve diğerleri, 2016). Veri setinin çerçeve dizisine dönüştürülmesi işleminde satırlar halinde bulunan veri dizisi tahmin verisinin ilişkili olacağı zaman dizisi boyutuna çevrilir. Verinin çerçeve dizisine çevrilmesi işlemi için veri setinin ilk satırından hedef değer (tahmini yapılacak değer) ilgili olduğu zaman aralığına (çerçeve boyutu:t) kadar

olan satır sayısı miktarınca öznitelik çerçeve haline getirilir ve bu çerçeveye karşılık gelen değer hedef değeri olarak seçilir. Bir sonraki çerçeve için ise Veri setinin ikinci satırından t zaman aralığı kadar öznitelik seçilir ve ilk seçilen hedef değerden sonraki değer ikinci çerçevenin hedef değeri olarak belirlenir. Bu işlem veri setinin sonuna kadar tekrarlanarak veri seti 2 boyutlu bir diziden 3 boyutlu bir diziye çevrilir(veri setinin 3 boyutlu olması durumunda ise veri setinin çerçeve dizisine dönüştürülmesi işlemi ile veri 4 boyutlu bir diziye çevrilmiş olur). Şekil 42’de veri setinin çerçeve dizisine dönüştürülmesi işlemi gösterilmiştir (Singh ve diğerleri, 2021). Model çıktısının bir dizi olması durumunda veri setinin çerçeve dizisine dönüştürülmesi işlemi hem giriş hem de çıkış işlemi için yapılır.



Şekil 42. Veri Setinin çerçeve dizisine dönüştürülmesi (Singh ve diğerleri, 2016).

ARAŞTIRMA BULGULARI

Bu çalışmanın amacı Erzurum Biyolojik Atıksu Arıtma Tesisinin çıkış suyunda bulunan BOI ve KOI konsantrasyonlarının zaman serisi derin öğrenme yöntemleri ile tahmin edilmesidir. Tahminleme modeli olarak RNN, LSTM ve GRU modelleri kullanılmıştır. Çalışılan modellere arıtma tesisinden alınan verilerin girdi olarak verilmesinden önce hazırlık işlemi olarak veri setinde ön işleme (data pre-processing) işlemleri gerçekleştirilmiş ve veri setinin modele uygun olarak bölümlenmesi ve boyutlandırılması yapılmıştır.

Veri Seti Hazırlık Aşaması

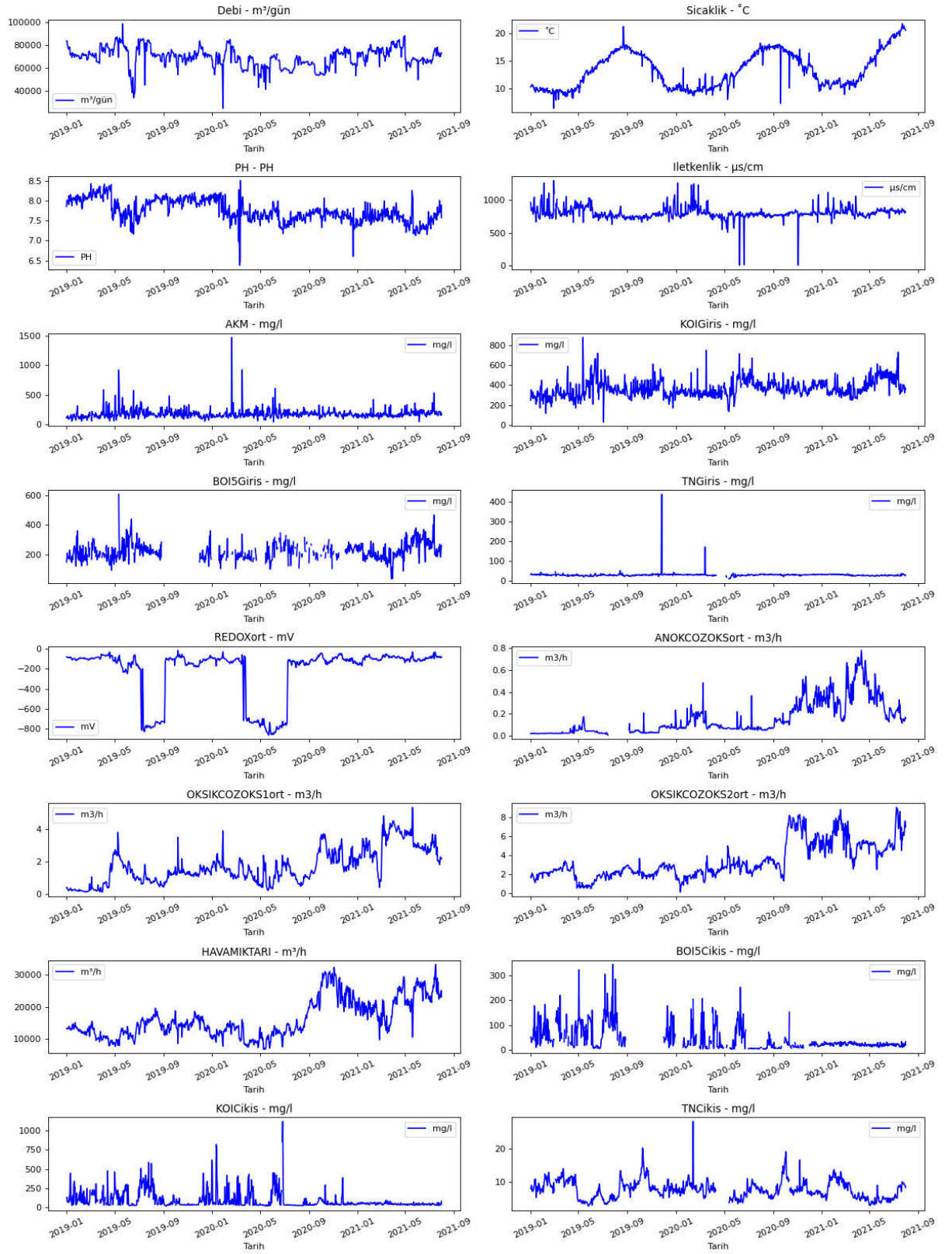
Eksik verilerin tamamlanması

ERBASAT'ın tesis çıkış suyunun BOI ve KOI miktarlarının tahminlemesi için 01.01.2019 ve 31.07.2021 tarihleri arası günlük deneysel ölçüm değerleri kullanılmıştır. 943 adet kayıttan ve 14 özniteliğe sahip veri setinde gerek tesisin bakım ve onarım işlemlerinden ve gerekse ölçüm yapılan cihazlardaki arızalardan dolayı tesisin giriş ve çıkış suyu ölçümlerinde *Tablo 1* Tablo 6'da gösterilen parametrelerde eksiklikler bulunmaktadır.

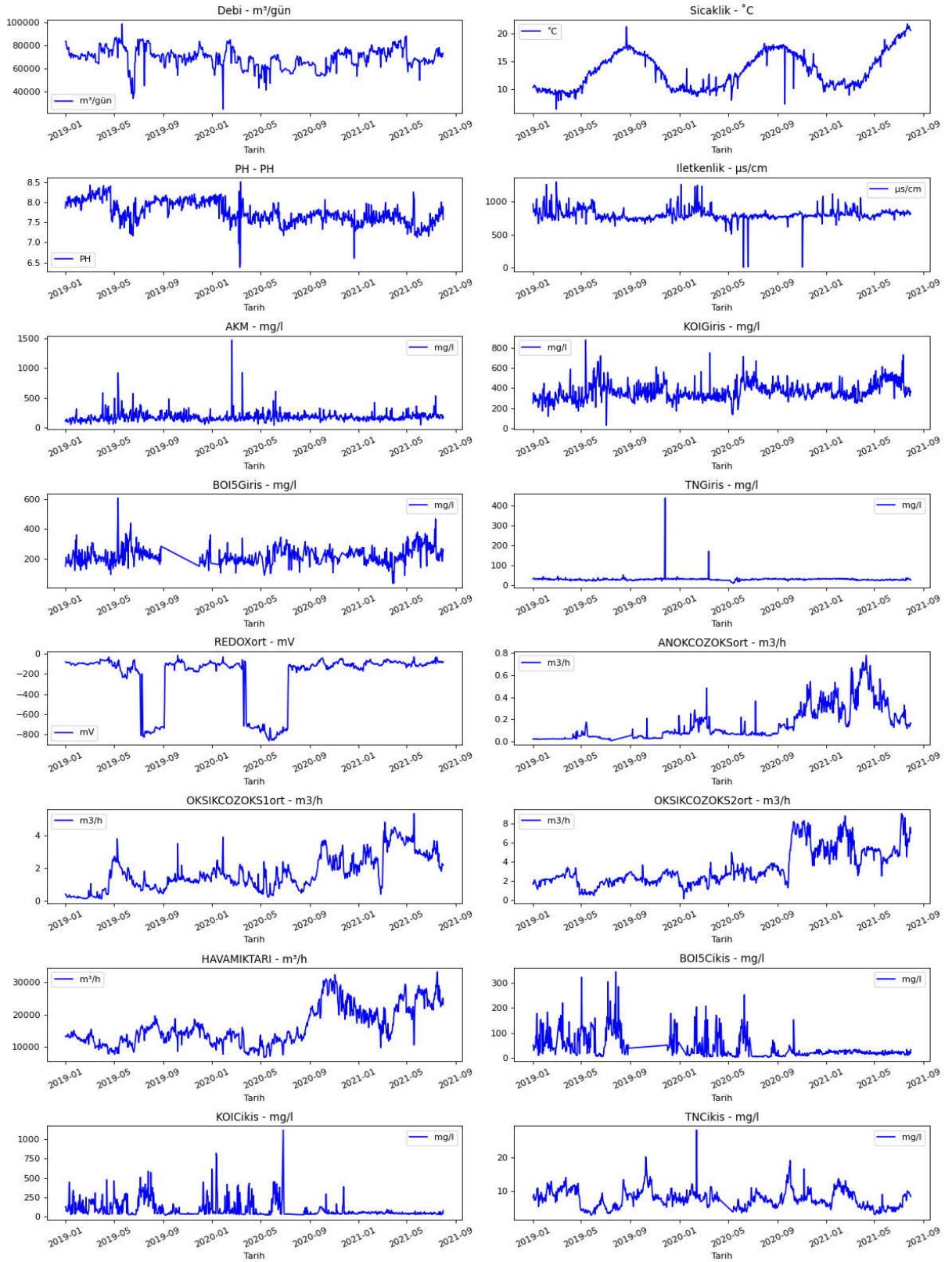
Tablo 6. Veri Setinde Bulunan Eksik Öznitelik Bilgileri.

Parametre	Eksik Veri Sayısı	Eksik Veri Yüzdesi
Debi	2	0,21
Sıcaklık	6	0,64
PH	2	0,21
İletkenlik	9	0,95
AKM	10	1,06
KOI Giriş	10	1,06
BOI5 Giriş	248	26,30
TN Giriş	37	3,92
Anoksik Çöz Oks. Ort.	52	5,51
BOI5 Çıkış	14	1,48
KOI Çıkış	209	22,16
TN Çıkış	40	4,24

Çalışılan derin öğrenme yöntemleri zaman serisi modelleri olduğundan özniteliklerin eksik olduğu kayıtların silinmesi zaman serisinde uyumsuzlara neden olacaktır. Veri setindeki eksik değerlerin tamamlanması için eksik özniteliklerin önceki ve sonraki değerlerinden faydalanılarak doğrusal enterpolasyon (linear interpolation) yöntemi kullanılmıştır. Şekil 43'de ham veri seti özniteliklerinin zaman serisi grafiği, Şekil 44'de ise doğrusal enterpolasyon kullanılarak eksik öznitelikleri tamamlanan veri seti özniteliklerinin zaman serisi grafiği gösterilmiştir.

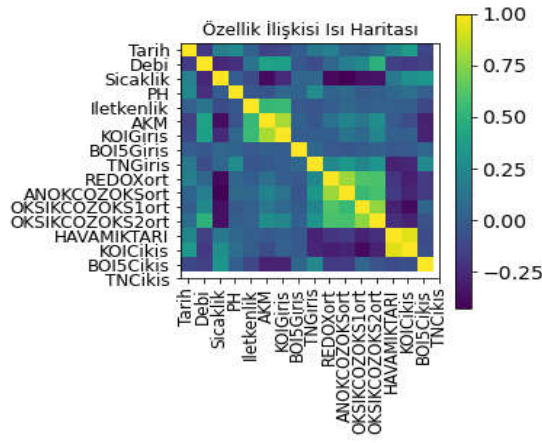


Şekil 43. Ham veri seti özneteliklerinin zaman serisi grafiği.

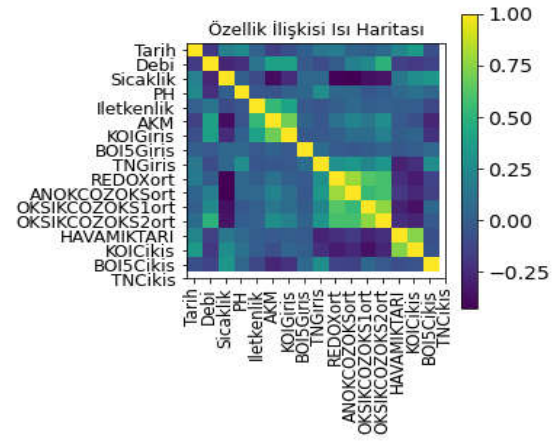


Şekil 44. Eksik öznitelikleri tamamlanan veri seti özniteliklerinin zaman serisi grafiği.

Veri setinin ham ve özniteliklerinin özellik ilişkisi ısı haritası grafiği Şekil 45-a'da ve eksik verileri tamamlanan özniteliklerinin özellik ilişkisi ısı haritası grafiği Şekil 45-b'de Korelasyon matrisi olarak gösterilmiştir.



(a)

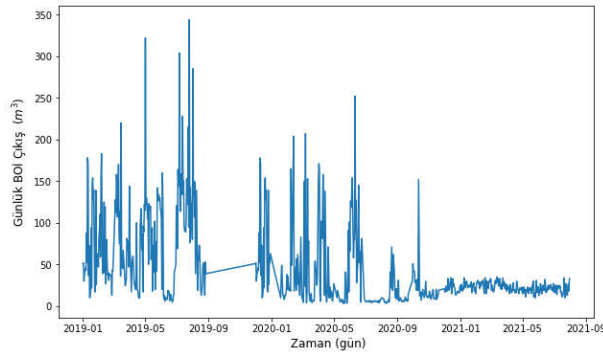


(b)

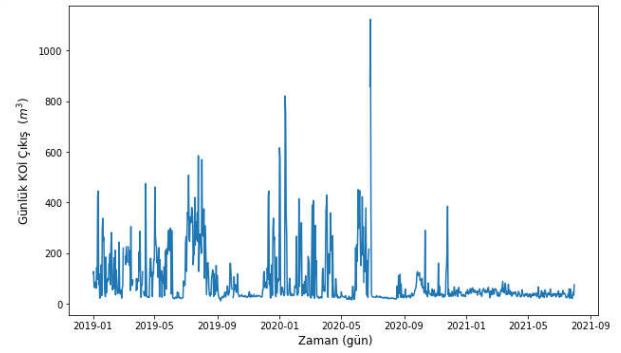
Şekil 45. Ham ve eksik verileri tamamlanan özniteliklerinin özellik ilişkisi ısı haritası.

Aykırı değerlerin düzeltilmesi

Şekil 46-a ve Şekil 46-b de bulunan zaman serisi grafiklerinde görüldüğü gibi hem BOİ ve hem de KOİ özniteliklerinde aykırı değerler bulunmaktadır ve bu değerler tahminleme sırasında çalışılan modellerin doğru değerler üretmesini olumsuz yönde etkileyeceğinden BOİ ve KOİ özniteliklerin aykırı değerlerinde enterpolasyon yöntemi ile düzenleme işlemi yapılmıştır. Aykırı değerleri düzenlenmiş BOİ ve KOİ özniteliklerinin dağılım grafiği Şekil 47-a ve Şekil 47-b de verilmiştir.

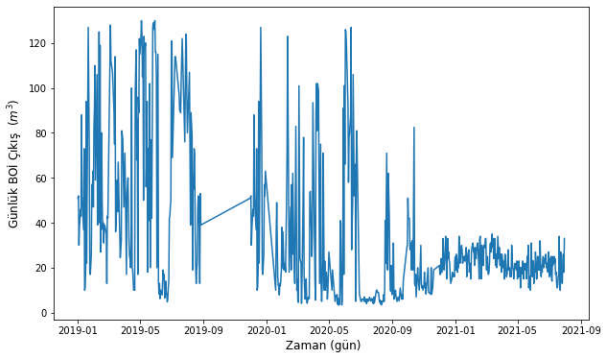


(a)

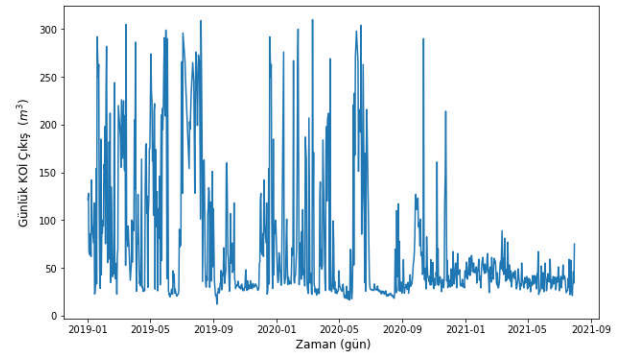


(b)

Şekil 46. BOİ ve KOİ özniteliklerinin zaman serisi grafiği.



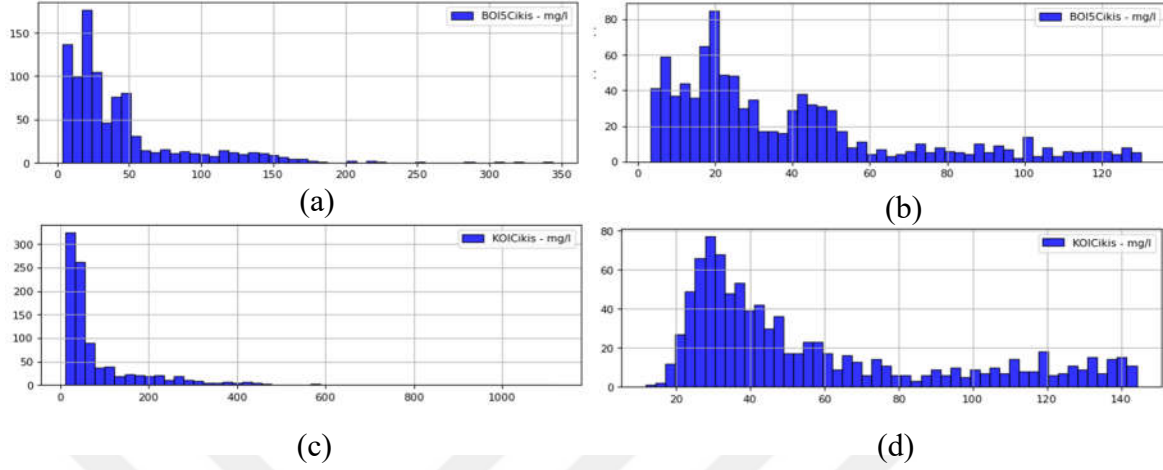
(a)



(b)

Şekil 47. Aykırı değerleri düzenlenmiş BOİ ve KOİ özniteliklerinin zaman serisi grafiği.

Ham öznitelik değerleri ve enterpolasyon yöntemi ile aykırı değerleri düzenlenmiş öznitelik değerlerinin gösterildiği Şekil 48'de; Şekil 48-a BOI değerlerinin ham halini Şekil 48-b BOI değerlerinin aykırı değerlerinin düzenlenmiş halini; Şekil 48-c KOI değerlerinin ham halini ve Şekil 48-d ise KOI değerlerinin aykırı değerlerinin düzenlenmiş halini göstermektedir.



Şekil 48. BOI ve KOI özniteliklerinin ham ve aykırı değerleri düzenlenmiş dağılım grafiği.

Veri normalleştirme

Tablo 7'de Veri seti örnek girdi parametrelerinde de görüldüğü gibi öznitelikler arasındaki farklılıklar bulunmaktadır. Veriler arasında farklılığın giderilmesi için ve verileri tek bir düzen içerisinde ölçeklendirmek için veri seti parametrelerine 0 ve 1 arasında normalleştirme işlemi uygulanmıştır. Tablo 8 veri setinin normalleştirilmiş örnek girdi parametrelerini göstermektedir.

Tablo 7. Veri Seti Örnek Girdi Parametreleri.

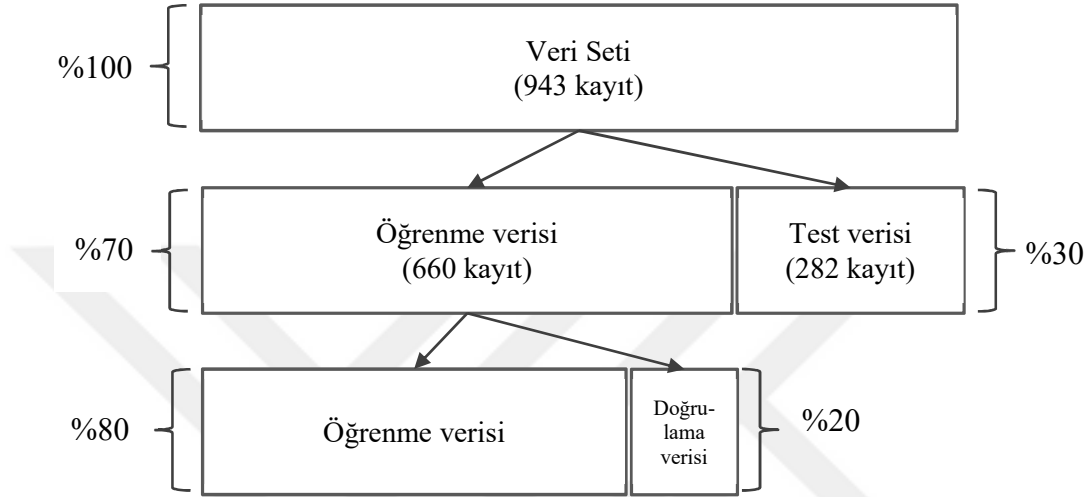
Tarih	Debi	Sıcaklık	PH	İletkenlik	AKM	KOI Giriş	BOI5 Giriş	TN Giriş
1.01.2019	83729	10,3	7,85	960	110	249	149	30,4
2.01.2019	83729	10,3	8,01	850	134	354	178	32,4
3.01.2019	79891	10,6	7,98	831	112	301	210	35,4
4.01.2019	79689	10,7	7,99	864	144	295	177	33,3
5.01.2019	77063	10,5	7,96	871	105	300	180	31,3
6.01.2019	76659	10,6	7,9	885	90	266	172	30,5
1.01.2019	83729	10,3	7,85	960	110	249	149	30,4
2.01.2019	83729	10,3	8,01	850	134	354	178	32,4

Tablo 8. Veri Seti Normalleştirilmiş Örnek Girdi Parametreleri.

Tarih	Debi	Sıcaklık	PH	İletkenlik	AKM	KOI Giriş	BOI5 Giriş	TN Giriş
1.01.2019	0,435657	0,662162	0,622642	0,607486	0,0948396	0,377061	0,313102	0,0426600
2.01.2019	0,481233	0,641892	0,632075	0,617610	0,1143650	0,425341	0,219653	0,0473300
3.01.2019	0,538874	0,635135	0,580189	0,624619	0,0711297	0,387659	0,256262	0,0473300
4.01.2019	0,634048	0,635135	0,599057	0,616831	0,0948396	0,422986	0,292871	0,0543349
5.01.2019	0,634048	0,641892	0,613208	0,623841	0,1004180	0,329958	0,175337	0,0426600
6.01.2019	0,626005	0,621622	0,627358	0,655771	0,1115760	0,375883	0,192678	0,0496649
1.01.2019	0,611260	0,648649	0,613208	0,637080	0,1004180	0,434762	0,344894	0,0519999
2.01.2019	0,435657	0,662162	0,622642	0,607486	0,0948396	0,377061	0,313102	0,0426600

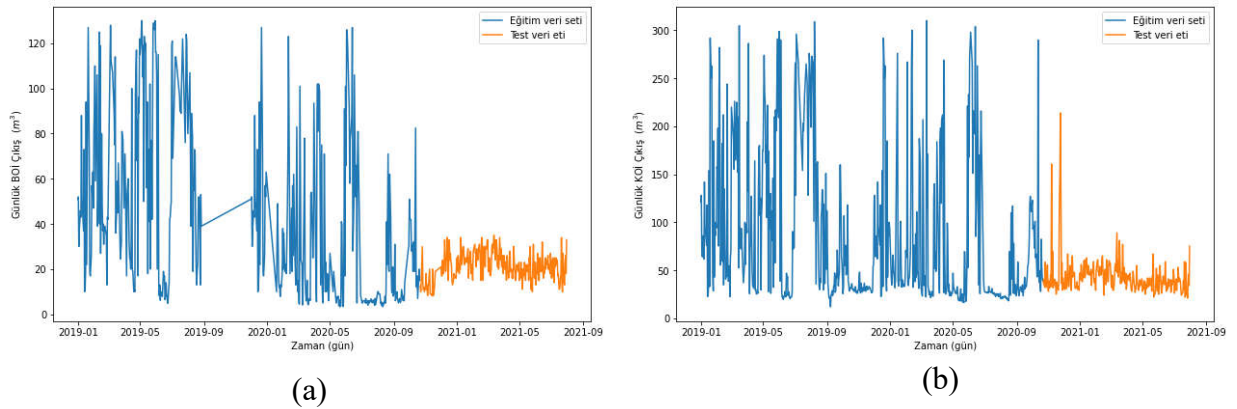
Veri setinin bölünmesi

Arıtma tesisininim çıkış suyunun BOI ve KOI tahminlemesi için toplam 943 kayıttan faydalanılmış olup veri setinin %70'i eğitim ve %30'u ise test amaçlı kullanılmıştır. Veri seti Şekil 49'da görüldüğü gibi ilk 660 kayıt öğrenme verisi olarak seçilmiş son 282 kayıt ise test verisi olarak seçilmiştir. Öğrenme verisinin ise %80'lik kısmı (528 kayıt) öğrenme amaçlı, %20'lik kısmı (132 kayıt) ise, öğrenme sırasında modelin öğrenmesinin ne doğrulukta olduğunu belirlemek üzere doğrulama verisi olarak kullanılmıştır.



Şekil 49. Eğitim, doğrulama ve test veri setleri.

Çalışılan modellerde BOI ve KOI tahmini için kullanılan parametrelerinin eğitim ve test değerlerinin zaman serisi grafiği Şekil 50'de verilmiştir. BOI (Şekil 50-a) ve KOI (Şekil 50-b) parametreleri aykırı değerleri düzenlenmiş olarak gösterilmiştir.

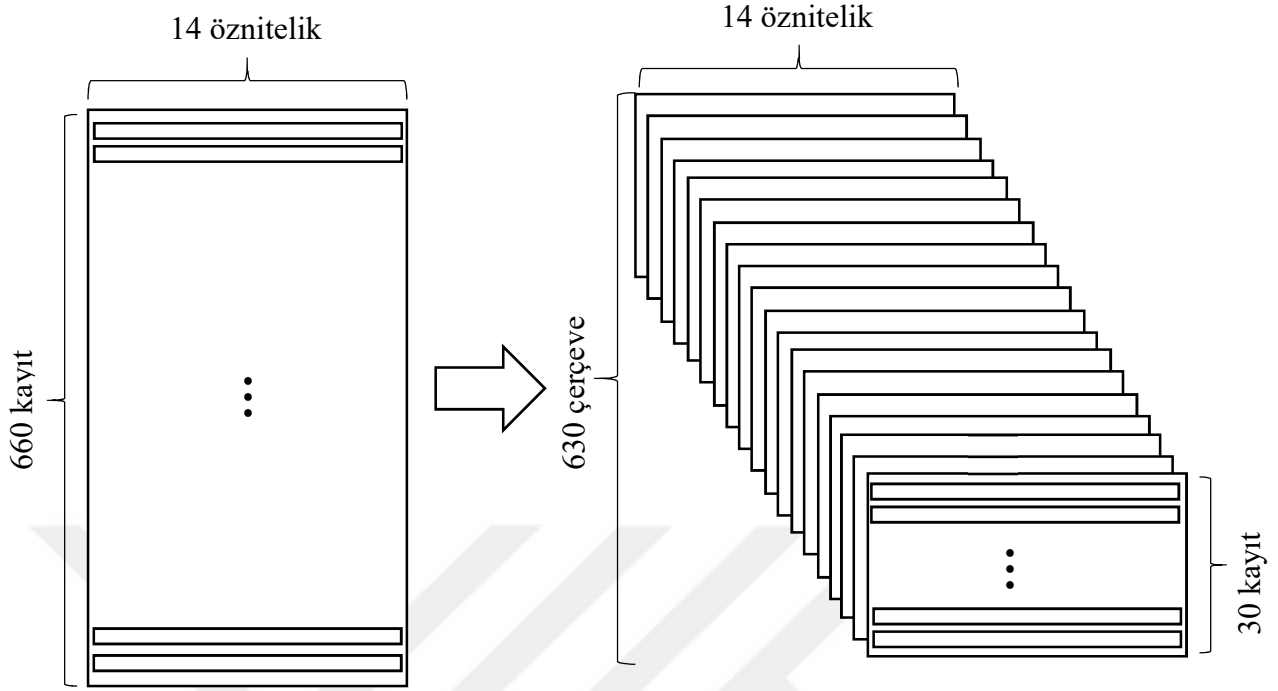


Şekil 50. BOI ve KOI parametrelerinin eğitim ve test değerlerinin zaman serisi grafiği.

Veri seti çerçeve dizisinin oluşturulması

Erzurum Atıksu arıtma tesislerinde kullanılan çamurun yaşı (θ_c) 20-30 gün aralığında olduğundan (Aydın, 2020), BOİ ve KOİ miktarı tahmini için çalışmada kullanılan veri setinden çerçeve dizisinin oluşturulması işleminde çerçeve boyutu (t) 30 olarak seçilmiştir. Eğitim için kullanılan 14 öznitelik ve 660 kayıttan oluşan veri seti, 630 adet 14 öznitelikli 30 satır

büyükliğünde çerçevelere dönüştürülmüştür. Şekil 51’de çalışmada kullanılan Veri setinin pencere dizisine dönüştürülmesi işlemi gösterilmiştir.



Şekil 51. Veri setinin pencere dizisine dönüştürülmesi.

Veri seti çerçeve dizisinin oluşturulması sonrasında $[660,14]$ boyutlu veri seti $[630,30,14]$ boyutlu bir dizi haline getirilmiştir. Modelin eğitimi sırasında $[630,30,14]$ boyutlu bir dizinin ilk %80 lik kısmı olan $[504,30,14]$ boyutlu dizi eğitim için, kalan %20’lik kısım $[136,30,14]$ boyutlu dizi ise eğitimin doğrulanması için kullanılmıştır.

Modellerin Oluşturulması

Erzurum Atıksu arıtma tesislerinde arıtılan suda bulunan BOİ ve KOİ miktarı tahmini için yapılan çalışmada kullanılan RNN, LSTM ve GRU modelleri karşılaştırmanın eşit şartlarda yapılabilmesi için aynı özelliklerde oluşturulmuştur. Modeller hem stateful hem de stateless olarak ve eşit iki gizli katmandan oluşacak şekilde tasarlanmıştır. Modellerde genelleştirmeyi önlemek için her iki gizli katmana %20 oranında seyreltme uygulanmıştır. Modellerde aktivasyon fonksiyonu olarak sigmoid fonksiyonu, optimizasyon yöntemi olarak Adaptif Moment Tahmini (Adam) yöntemi kullanılmıştır. Modellerin eğitimi sırasında hesaplanan hata değerinin lokal minimum değerlerinde kalmaması için toplulaştırma mini-batch yöntemi kullanılmıştır. Stateful olarak tasarlanan modellerde batch size’ın hem eğitim hem de test verisini tam bölmesi gerektiğinden, batch size olarak hem 630 kayıttan oluşan öğrenim veri setini hem de 252 kayıttan oluşan test veri setini tam olarak bölen 18 sayısı seçilmiştir. Stateful modellerde eğitim için 28 batch, test için 14 batch kullanılmıştır. Eğitim Stateless olarak tasarlanan modellerde ise batch size 8 olarak seçilmiştir.

BOİ ve KOİ miktarlarının tahmini 2 gizli katmanlı RNN, LSTM ve GRU modelleri ile yapılmış ve kullanılan modeller her iki gizli katmanında 8, 16, 32, 64, 128, 256, 512, 1024 ve 2048 ünite olacak şekilde test edilmiştir.

Modellerde bulunan hücre ve ön yargı ağırlıklarının başlangıç değerleri 0 ile bir arasında normal dağılım (gauss dağılımı) yöntemi ile atanmış, öğrenme oranı olarak 0,001 değeri seçilmiş ve eğitimin ya 500 ipokta ya da doğrulama kayıp verisinin 20 ipokta değişmemesi durumunda erken durdurma işlemi ile tamamlanması sağlanmıştır.

Kullanılan metrikler

BOİ ve KOİ miktarı tahmini için çalışmada kullanılan modellerin eğitimi sırasında oluşan hatanın belirlenmesi için hata fonksiyonu olarak MAE fonksiyonu, modellerin tahminleme başarısının kontrolü için ise metrik olarak MAE, MSE, RMSE ve R^2 fonksiyonları kullanmıştır.

Tahmin Aşaması

Erzurum Atıksu arıtma tesislerinde arıtılan suda bulunan BOİ ve KOİ miktarı tahmini için çalışılan RNN, LSTM ve GRU modelleri stateless ve stateful olarak test edilmiştir. Her iki parametre için oluşturulan modellerin tahmin doğruluk performansları çalışmanın bu bölümünde özetlenmiştir.

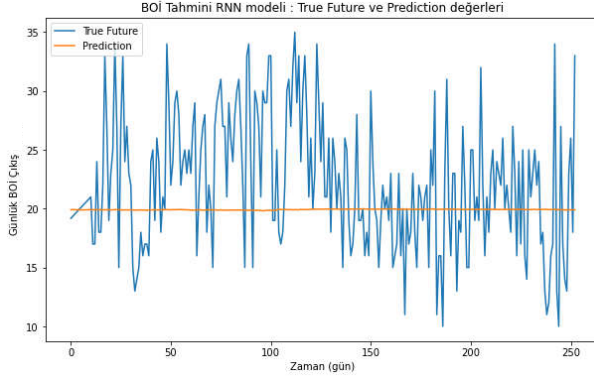
RNN modelinin uygulanması

Çalışmada BOİ tahmini için kullanılan stateless RNN modelinin tahmin sonuçları Tablo 9'da verilmiştir.

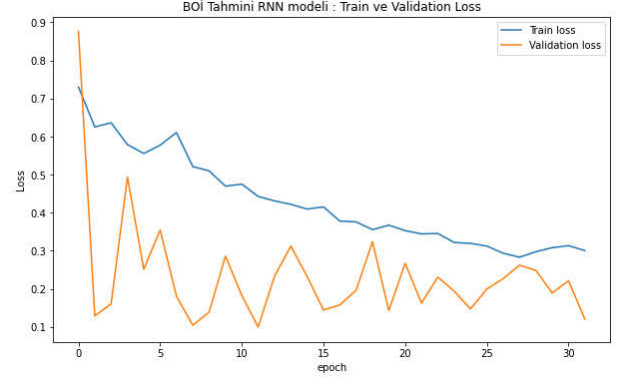
Tablo 9. Stateless RNN Modelinin BOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R^2
8	12,3615	184,135	13,5696	66,3679	-5,0546
16	11,2041	155,448	12,4679	60,6619	-4,1113
32	6,5467	59,9806	7,7447	36,3364	-0,9722
64	6,7648	63,79	7,9869	37,6136	-1,0975
128	11,2674	156,846	12,5238	47,5229	-4,1573
256	4,7155	35,5992	5,9665	21,0372	-0,1705
512	36,9207	1.393,56	37,3304	170,999	-44,822
1024	53,9332	2.939,20	54,2144	266,66	-95,645
2048	23,492	582,287	24,1306	120,019	-18,146

Stateless RNN modeli ile BOİ miktarının tahmini için yapılan çalışmada MAE, MSE, RMSE, MAPE ve R^2 metriklerine göre en başarılı sonuç 256 ünite ile oluşturulan model ile elde edilmiştir. Kullanılan tüm metriklerde en başarılı sonuçların alındığı 256 üniteden oluşan modelin tahmin ve hata grafiği Şekil 52'de verilmiştir.



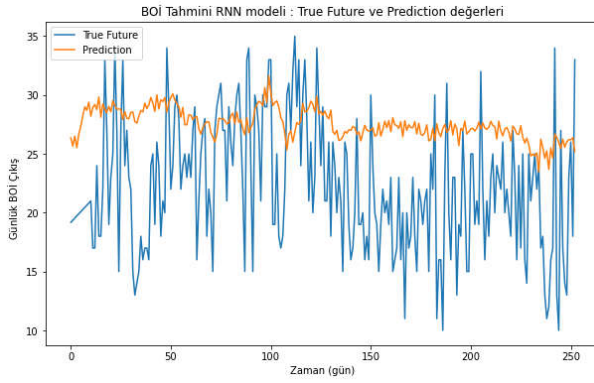
(a)



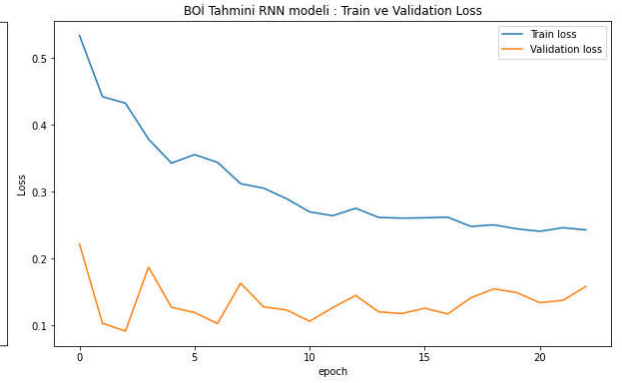
(b)

Şekil 52. BOİ tahmini için 256 ünite ile oluşturulan stateless RNN modelinin tahmin ve hata grafiği

256 ünite ile oluşturulan stateless RNN modelinin tahmin grafiği linear bir formda iken modelin 32 ünite ile oluşturulan tahmin değerleri **Şekil 53**'da görüldüğü gibi nonlinear bir grafiğe sahiptir.



(a)



(b)

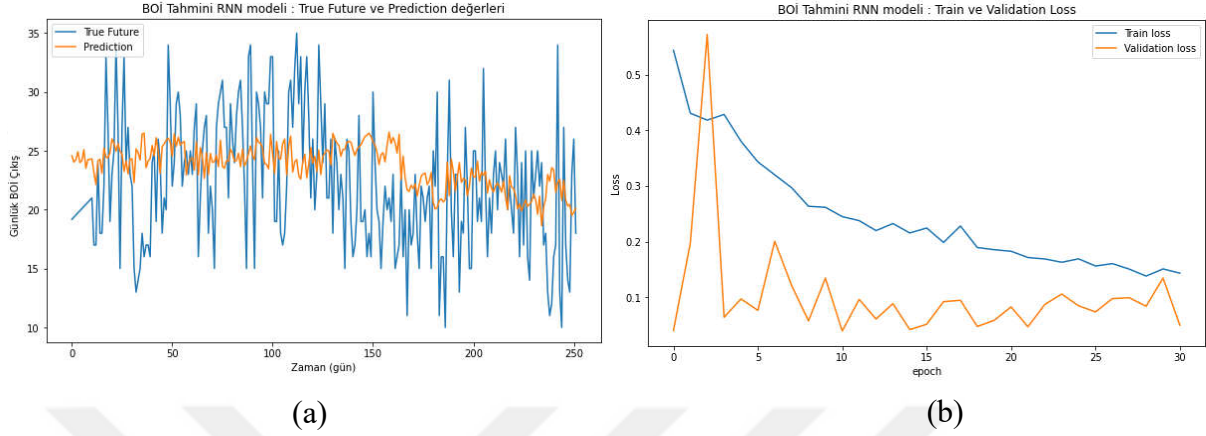
Şekil 53. BOİ tahmini için 32 ünite ile oluşturulan stateless RNN modelinin tahmin ve hata grafiği.

Çalışmada BOİ tahmini için kullanılan stateful RNN modelinin tahmin sonuçları Tablo 10'da verilmiştir.

Tablo 10. Stateful RNN Modelinin BOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	12,5977	197,338	14,0477	67,7419	-5,5633
16	12,2588	180,555	13,4371	65,797	-5,0051
32	10,4861	137,203	11,7134	56,7711	-3,5633
64	10,5007	136,055	11,6642	56,652	-3,5251
128	4,762	31,9928	5,6562	24,7959	-0,0641
256	12,7744	234,683	15,3194	64,2851	-6,8053
512	21,4709	491,064	22,16	96,7777	-15,332
1024	15,9588	284,752	16,8746	83,7942	-8,4706
2048	66,6414	4.471,15	66,8666	328,301	-147,71

Stateful RNN modeli ile BOİ miktarının tahmini için yapılan çalışmada MAE, MSE, RMSE, MAPE ve R^2 metriklerine göre en başarılı sonuç 128 ünite ile oluşturulan model ile elde edilmiştir. Kullanılan tüm metriklerde en başarılı sonuçların alındığı 128 üniteden oluşan stateful modelin tahmin ve hata grafiği Şekil 54'te verilmiştir.



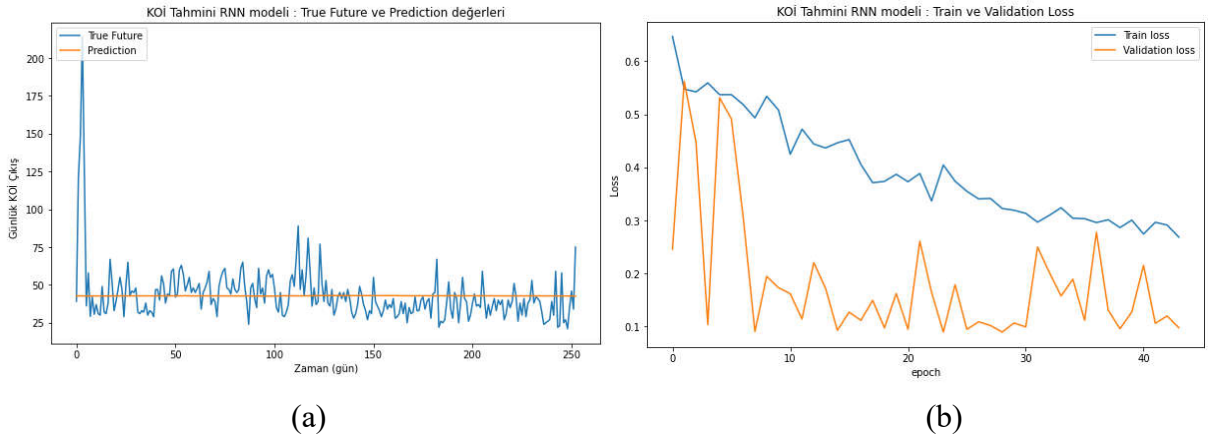
Şekil 54. BOİ tahmini için 128 ünite ile oluşturulan stateful RNN modelinin tahmin ve hata grafiği.

Çalışmada KOİ tahmini için kullanılan stateless RNN modelinin tahmin sonuçları Tablo 11'de verilmiştir.

Tablo 11. Stateless RNN Modelinin KOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R^2
8	14,2743	412,3791	20,3071	37,8535	-0,2122
16	12,0764	364,4792	19,0913	29,8791	-0,0714
32	12,9351	373,4135	19,3239	33,3936	-0,0976
64	12,8967	373,6059	19,3289	32,997	-0,0982
128	22,9498	722,9182	26,8871	63,0199	-1,125
256	10,8274	340,8326	18,4617	25,1225	-0,0019
512	48,0237	2.646,50	51,4441	112,3884	-6,7794
1024	26,8218	901,5615	30,026	73,5318	-1,6501
2048	227,3899	52.046,36	228,1367	593,0339	-151,99

Stateless RNN modeli ile KOİ miktarının tahmini için yapılan çalışmada MAE, MSE, RMSE, MAPE ve R^2 metriklerine göre en başarılı sonuç 256 ünite ile oluşturulan model ile elde edilmiştir. Kullanılan tüm metriklerde en başarılı sonuçların alındığı 256 üniteden oluşan modelin tahmin ve hata grafiği Şekil 55'te verilmiştir.



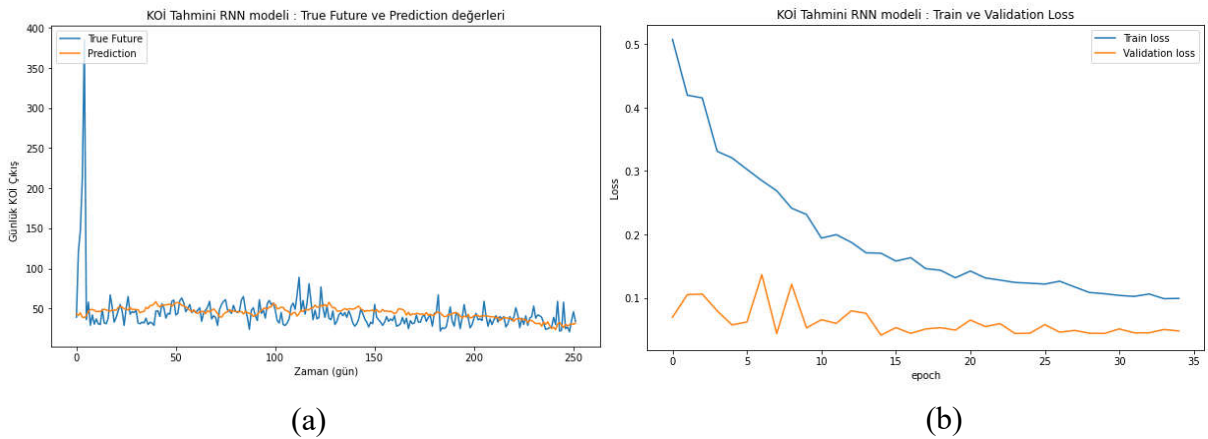
Şekil 55. KOİ tahmini için 256 ünite ile oluşturulan stateless RNN modelinin tahmin ve hata grafiği

Çalışmada KOİ tahmini için kullanılan stateful RNN modelinin tahmin sonuçları Tablo 12’de verilmiştir.

Tablo 12. Stateful RNN Modelinin KOİ Tahmin Sonuçları.

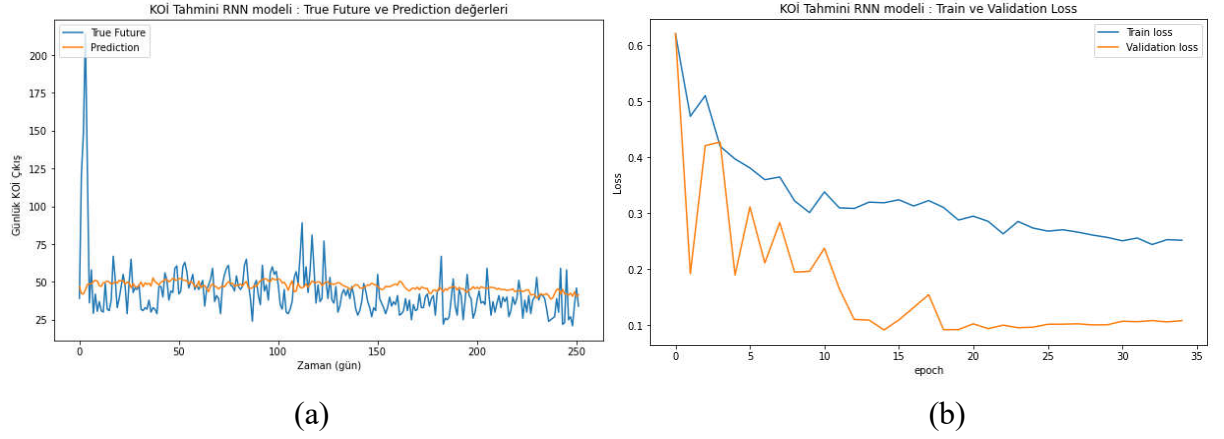
Ünite sayısı	MAE	MSE	RMSE	MAPE	R^2
8	13,8007	408,9464	20,2224	36,1879	-0,2116
16	15,9833	465,2565	21,5698	42,979	-0,3785
32	12,2636	806,8796	28,4056	26,6167	-0,0428
64	11,4745	348,9716	18,6808	28,6659	-0,034
128	38,8226	1.684,34	41,0407	105,1216	-3,9905
256	83,7191	7.205,34	84,8843	223,4396	-20,3484
512	20,6893	624,2169	24,9843	56,913	-0,8495
1024	49,7809	2.626,26	51,247	134,4474	-6,7812
2048	68,9131	4.907,31	70,0522	184,7198	-13,5397

Stateful RNN modeli ile KOİ miktarının tahmini için yapılan çalışmada MAPE metriğine göre en başarılı sonuç 32 ünite ile oluşturulan model ile; MAE, MSE, RMSE ve R^2 metriklerine göre en başarılı sonuç 64 ünite ile oluşturulan model ile elde edilmiştir. MAPE metriğine göre en başarılı sonuçların alındığı 32 üniteden oluşan stateful modelin tahmin ve hata grafiği Şekil 56’da verilmiştir.



Şekil 56. KOİ tahmini için 32 ünite ile oluşturulan stateful RNN modelinin tahmin ve hata grafiği.

Stateful RNN modeli ile KOİ tahminlemede MAE, MSE, RMSE ve R^2 metriklerine en başarılı sonuçların alındığı 64 üniteden oluşan stateful modelin tahmin ve hata grafiği Şekil 57’de verilmiştir.



Şekil 57. KOİ tahmini için 64 ünite ile oluşturulan stateful RNN modelinin tahmin ve hata grafiği.

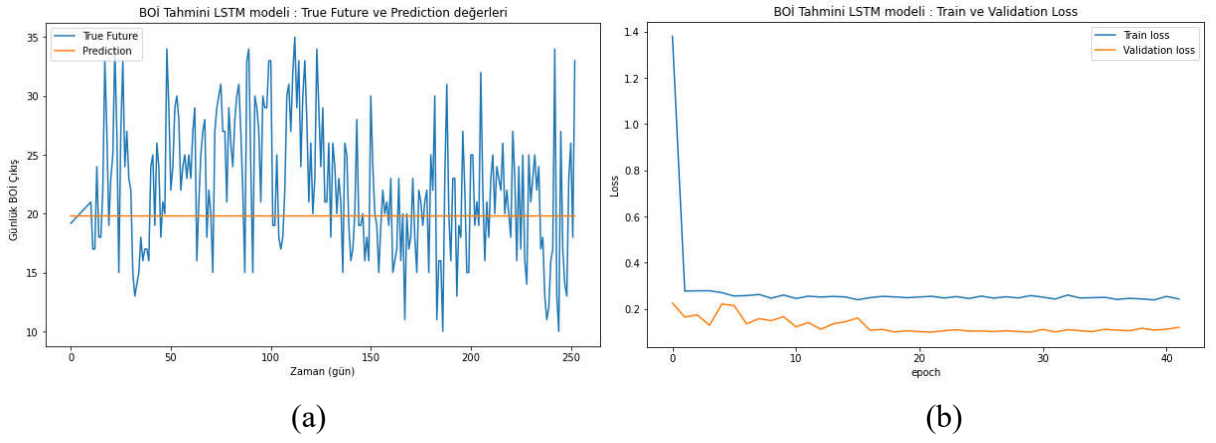
LSTM modelinin uygulanması

Çalışmada BOİ tahmini için kullanılan stateless LSTM modelinin tahmin sonuçları Tablo 13’de verilmiştir.

Tablo 13. Stateless LSTM Modelinin BOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R^2
8	11,0138	153,9114	12,4061	59,7066	-4,0608
16	12,2064	178,4494	13,3585	65,4681	-4,8676
32	6,7769	63,5519	7,9719	37,6561	-1,0897
64	10,5063	138,7298	11,7784	57,2743	-3,5616
128	11,8162	169,8766	13,0337	63,7674	-4,5858
256	10,0124	127,4643	11,29	54,7859	-3,1912
512	10,1894	131,3885	11,4625	55,6734	-3,3202
1024	7,1312	69,8724	8,359	39,7335	-1,2975
2048	4,7478	36,0496	6,0041	21,0809	-0,1854

Stateful LSTM modeli ile BOİ miktarının tahmini için yapılan çalışmada MAE, MSE, RMSE, MAPE ve R^2 metriklerine göre en başarılı sonuç 2048 ünite ile oluşturulan model ile elde edilmiştir. Kullanılan tüm metriklerde en başarılı sonuçların alındığı 2048 ünitenden oluşan stateful modelin tahmin ve hata grafiği Şekil 58’de verilmiştir.



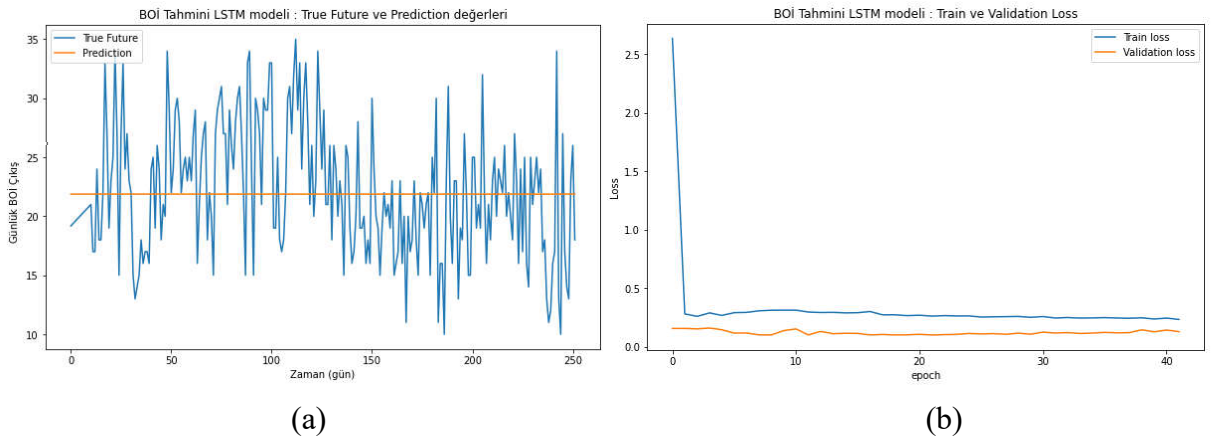
Şekil 58. BOİ tahmini için 2048 ünite ile oluşan stateless LSTM modelinin tahmin ve hata grafiği.

Çalışmada BOİ tahmini için kullanılan stateful LSTM modelinin tahmin sonuçları Tablo 14’te verilmiştir.

Tablo 14. Stateful LSTM Modelinin BOİ Tahmin Sonuçları.

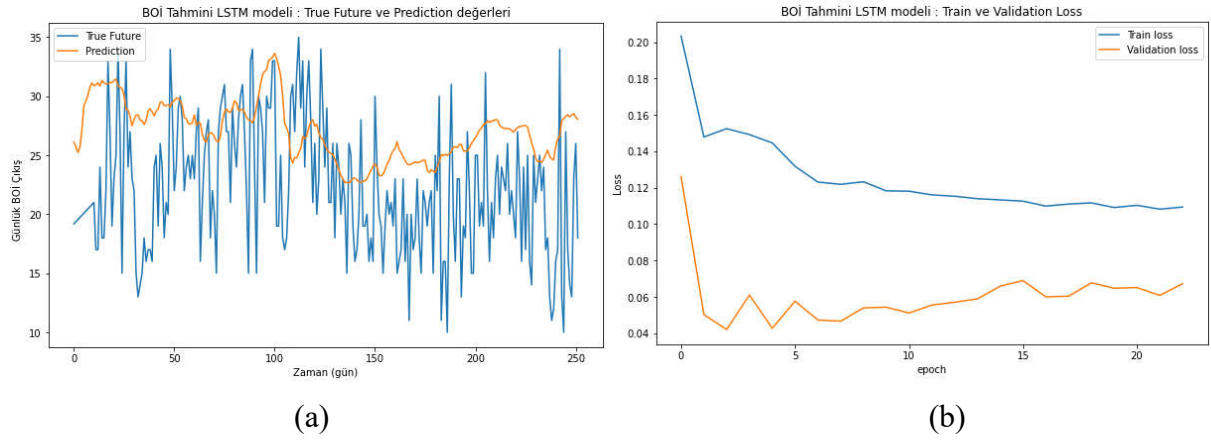
Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	8,3149	100,2666	10,0133	45,1376	-2,3348
16	12,2814	179,8403	13,4105	65,643	-4,9813
32	6,1051	53,9694	7,3464	33,6587	-0,795
64	15,1931	259,8433	16,1197	79,9438	-7,6422
128	12,2235	178,0125	13,3421	65,569	-4,9205
256	14,4805	239,8934	15,4885	76,6818	-6,9786
512	15,1984	260,9201	16,153	80,1135	-7,678
1024	15,3461	265,4388	16,2923	80,8239	-7,8283
2048	4,4749	30,1315	5,4892	21,9428	-0,0021

Stateful LSTM modeli ile BOİ miktarının tahmini için yapılan çalışmada MAE, MSE, RMSE, MAPE ve R² metriklerine göre en başarılı sonuç 2048 ünite ile oluşturulan model ile elde edilmiştir. Kullanılan tüm metriklerde en başarılı sonuçların alındığı 2048 ünitiden oluşan stateful modelin tahmin ve hata grafiği Şekil 59’da verilmiştir.



Şekil 59. BOİ tahmini için 2048 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.

BOİ tahmini için 2048 ünite ile oluşturulan stateless LSTM modelinin tahmin grafiği lineer bir formda iken modelin 32 ünite ile oluşturulan tahmin değerleri Şekil 60'ta görüldüğü gibi nonlinear bir grafiğe sahiptir.



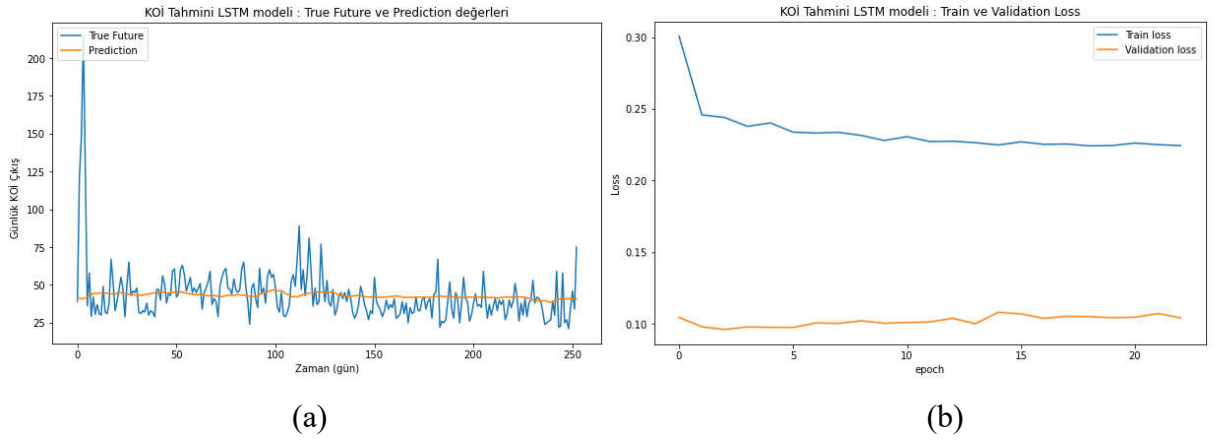
Şekil 60. BOİ tahmini için 32 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.

Çalışmada KOİ tahmini için kullanılan stateless LSTM modelinin tahmin sonuçları Tablo 15'te verilmiştir.

Tablo 15. Stateless LSTM Modelinin KOİ Tahmin Sonuçları.

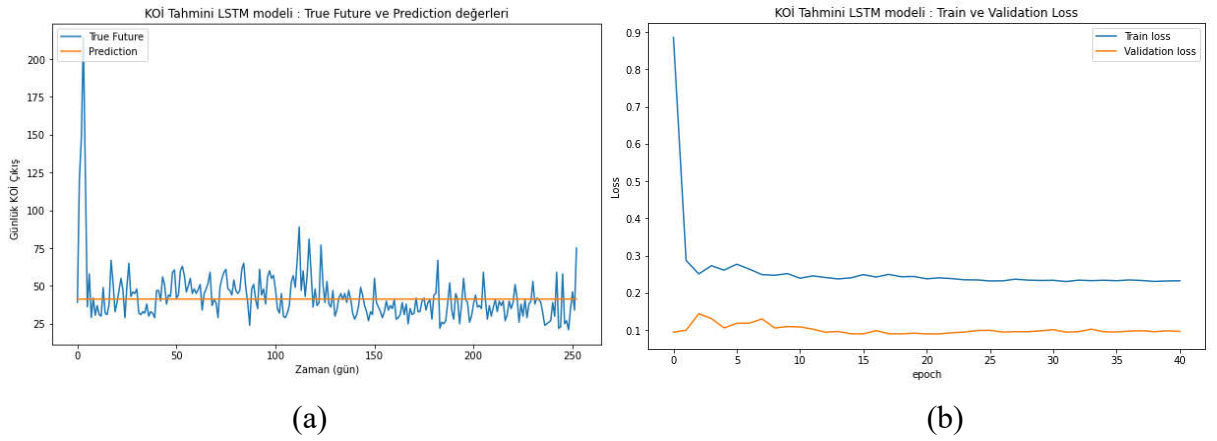
Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	11,0208	349,342	18,6907	26,1802	-0,0269
16	10,4314	338,6465	18,4024	24,0657	0,0045
32	10,6586	338,9262	18,4099	24,9489	0,0037
64	14,1978	401,0811	20,027	37,5306	-0,179
128	12,4931	363,302	19,0605	31,9593	-0,0679
256	14,4502	407,8855	20,1962	38,3649	-0,199
512	17,294	496,0702	22,2726	47,067	-0,4582
1024	10,5626	343,6825	18,5387	23,6489	-0,0103
2048	10,9184	340,2057	18,4447	25,6473	-0,033

Stateless LSTM modeli ile KOİ miktarının tahmini için yapılan çalışmada MAE, MSE, RMSE ve R² metriklerine göre en başarılı sonuç 16 ünite ile oluşturulan model ile; MAPE metriğine göre en başarılı sonuç 1024 ünite ile oluşturulan model ile elde edilmiştir. MAE, MSE, RMSE ve R² metriklerine göre en başarılı sonuçların alındığı 16 ünitiden oluşan Stateless modelin tahmin ve hata grafiği Şekil 61'de verilmiştir.



Şekil 61. KOİ tahmini için 16 ünite ile oluşan stateless LSTM modelinin tahmin ve hata grafiği.

Stateless LSTM modeli ile KOİ tahminlemede MAPE metriğine göre en başarılı sonuçların alındığı 1024 ünitelerden oluşan stateless modelin tahmin ve hata grafiği Şekil 62’de verilmiştir.



Şekil 62. KOİ tahmini için 1024 ünite ile oluşan stateless LSTM modelinin tahmin ve hata grafiği.

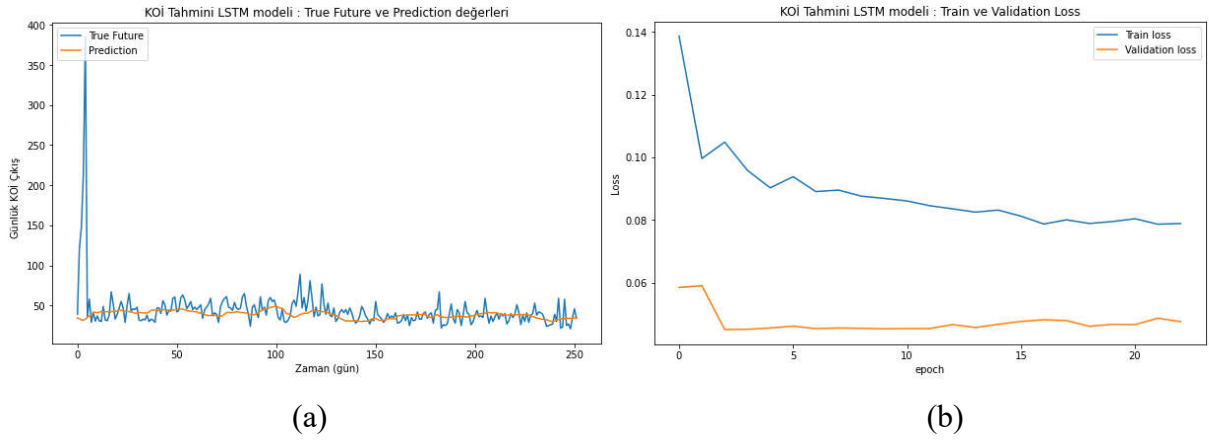
Çalışmada KOİ tahmini için kullanılan stateful LSTM modelinin tahmin sonuçları Tablo 16’da verilmiştir.

Tablo 16. Stateful LSTM Modelinin KOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	11,77	837,8165	28,9451	23,5806	-0,0828
16	11,1207	824,2519	28,7098	20,9574	-0,0653
32	10,3026	356,1463	18,8718	22,206	-0,0552
64	12,9939	375,1297	19,3683	33,8098	-0,1115
128	12,942	372,1613	19,2915	33,6307	-0,1027
256	13,3294	380,3524	19,5026	34,9237	-0,1269
512	14,1035	399,2165	19,9804	37,4152	-0,1828
1024	15,5513	439,3358	20,9603	41,9049	-0,3017
2048	82,9353	7.069,40	84,0797	84,0797	-19,9456

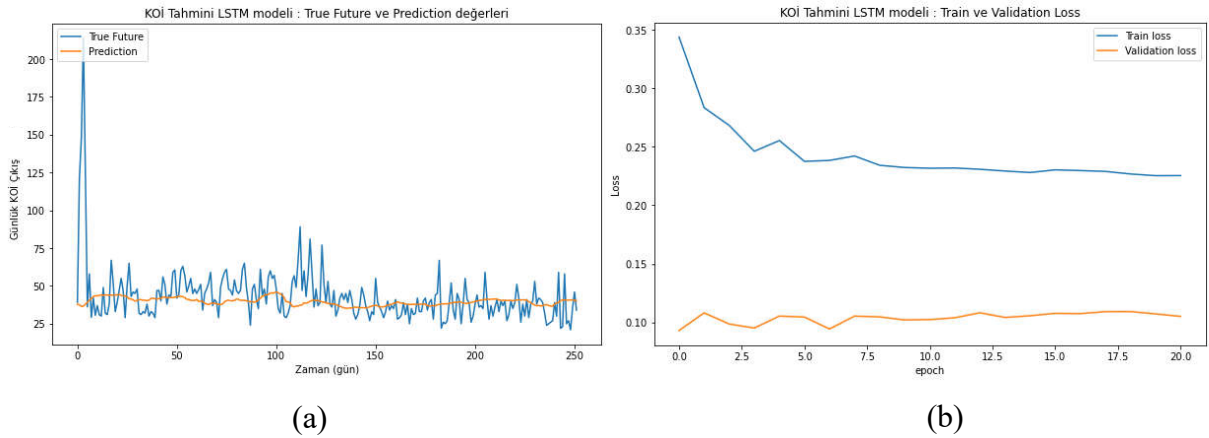
Stateful LSTM modeli ile KOİ miktarının tahmini için yapılan çalışmada MAPE metriğine göre en başarılı sonuç 16 ünite ile oluşturulan model ile; MAE, MSE, RMSE ve R²

metriklerine göre ise en başarılı sonuç 32 ünite ile oluşturulan model ile elde edilmiştir. MAPE metriğine göre en başarılı sonuçların alındığı 16 üniteden oluşan stateful modelin tahmin ve hata grafiği Şekil 63’de verilmiştir.



Şekil 63. KOİ tahmini için 16 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.

Stateful LSTM modeli ile KOİ tahminlemede MAE, MSE, RMSE ve R^2 metriklerine göre en başarılı sonuçların alındığı 32 üniteden oluşan stateful modelin tahmin ve hata grafiği Şekil 64’te verilmiştir.



Şekil 64. KOİ tahmini için 32 ünite ile oluşan stateful LSTM modelinin tahmin ve hata grafiği.

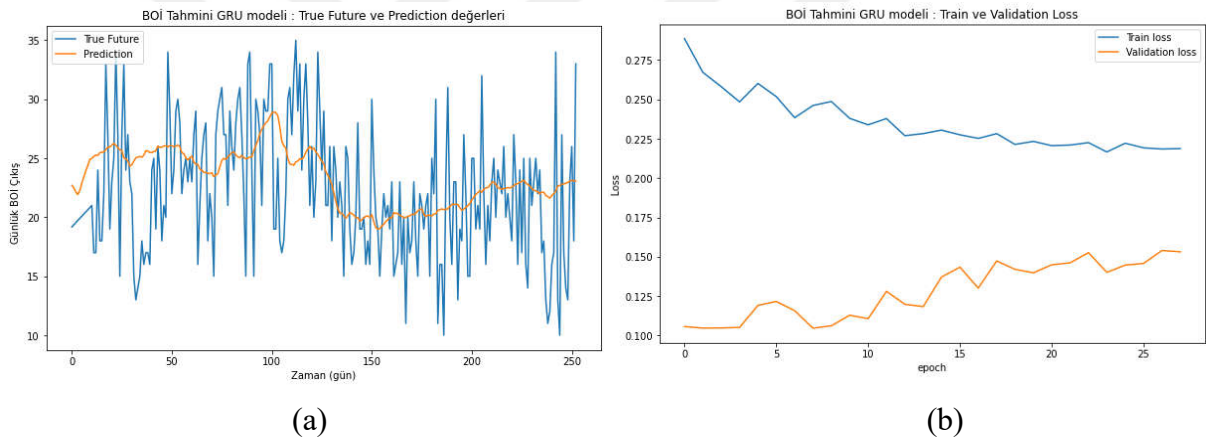
GRU modelinin uygulanması

Çalışmada BOİ tahmini için kullanılan stateless GRU modelinin tahmin sonuçları Tablo 17’de verilmiştir.

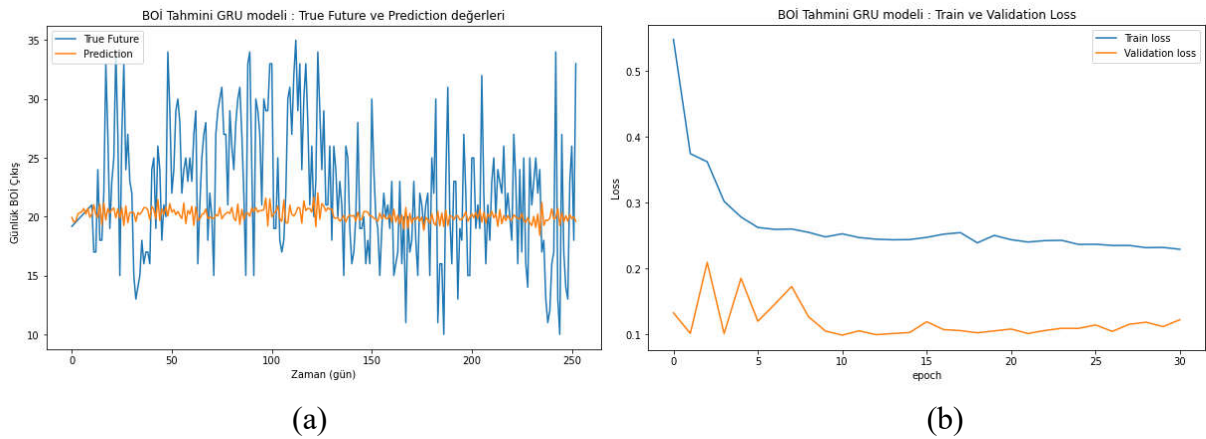
Tablo 17. Stateless GRU Modelinin BOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	13,2883	207,022	14,3883	70,8681	-5,8071
16	5,1734	40,8601	6,3922	28,0781	-0,3435
32	4,2401	27,6266	5,2561	22,0336	0,0916
64	5,2228	39,2987	6,2689	28,7065	-0,2922
128	5,2865	39,7912	6,308	28,935	-0,3084
256	4,632	34,0073	5,8316	20,8385	-0,1182
512	6,0318	56,862	7,5407	24,7166	-0,8697
1024	9,4117	114,1651	10,6848	51,7035	-2,7539
2048	9,3868	113,4573	10,6516	51,5582	-2,7306

Stateless GRU modeli ile BOİ miktarının tahmini için yapılan çalışmada MAE, MSE RMSE ve R² metriklerine göre en başarılı sonuç 32 ünite ile oluşturulan model ile; MAPE metriğine göre en başarılı sonuç 256 ünite ile oluşturulan model ile elde edilmiştir. MAE, MSE RMSE ve R² metriklerine göre en başarılı sonuçların alındığı 32 ünitelerden oluşan Stateless modelin tahmin ve hata grafiği Şekil 65’te verilmiştir.

**Şekil 65.** BOİ tahmini için 32 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.

Stateless GRU modeli ile BOİ tahminlemede R² metriğine göre en başarılı sonuçların alındığı 256 ünitelerden oluşan stateless modelin tahmin ve hata grafiği Şekil 66’da verilmiştir.

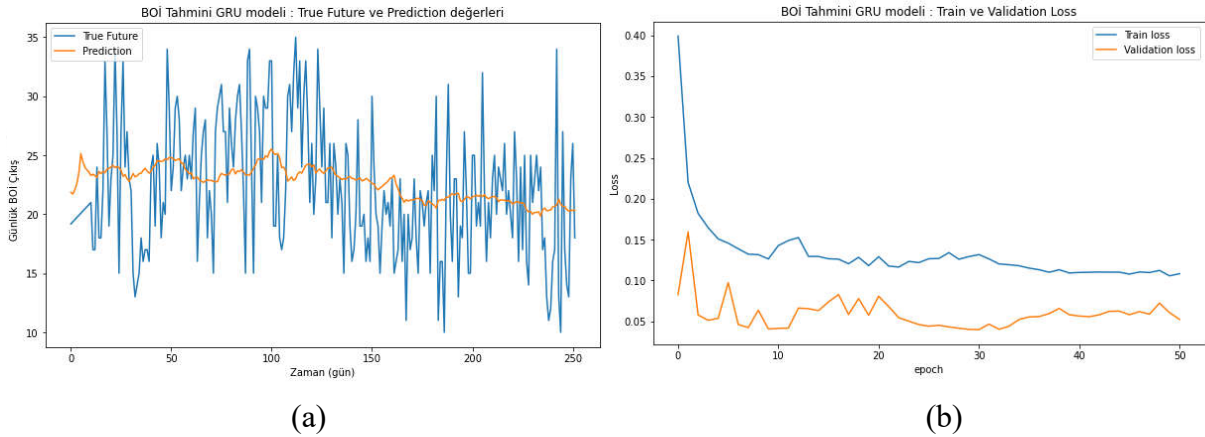
**Şekil 66.** BOİ tahmini için 256 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.

Çalışmada BOİ tahmini için kullanılan stateful GRU modelinin tahmin sonuçları Tablo 18’de verilmiştir.

Tablo 18. Stateful GRU Modelinin BOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	14,5161	240,2691	15,5006	76,7597	-6,9911
16	5,8511	52,3504	7,2354	31,3316	-0,7411
32	5,0343	39,1315	6,2555	27,68	-0,3015
64	4,7777	32,8984	5,7357	24,9255	-0,0942
128	4,3434	27,1374	5,2094	21,9196	0,0974
256	5,1435	37,538	6,1268	28,021	-0,2485
512	9,5516	119,6081	10,9365	39,6908	-2,9781
1024	4,7287	31,9933	5,6563	24,9845	-0,0641
2048	46,8969	2.229,43	47,2169	233,0268	-73,149

Stateful GRU modeli ile BOİ miktarının tahmini için yapılan çalışmada MAE, MSE, RMSE, MAPE ve R² metriklerine göre en başarılı sonuç 128 ünite ile oluşturulan model ile elde edilmiştir. MAE, MSE, RMSE, MAPE ve R² metriklerine göre en başarılı sonuçların alındığı 128 üniteden oluşan stateful modelin tahmin ve hata grafiği Şekil 67’de verilmiştir.



Şekil 67. BOİ tahmini için 128 ünite ile oluşan stateful GRU modelinin tahmin ve hata grafiği.

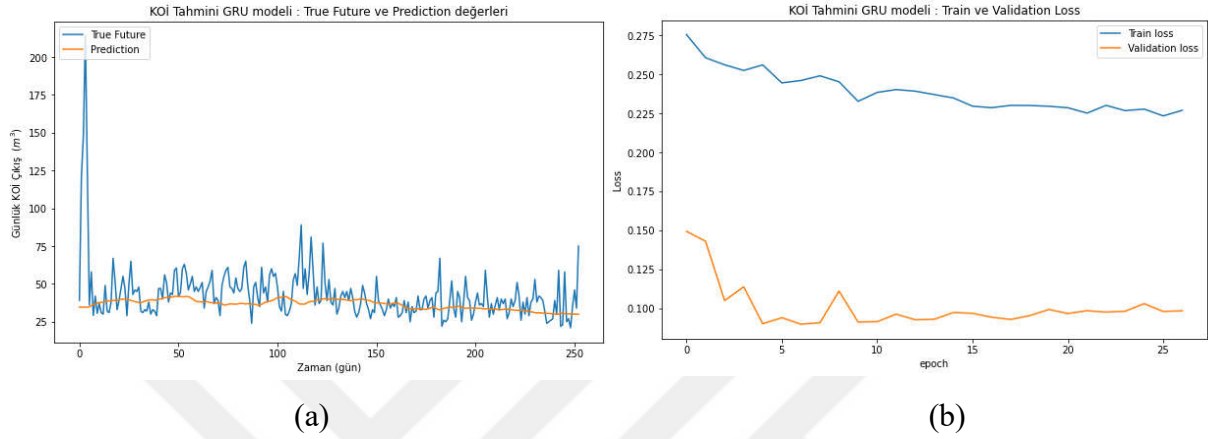
Çalışmada KOİ tahmini için kullanılan stateless GRU modelinin tahmin sonuçları Tablo 19’da verilmiştir.

Tablo 19. Stateless GRU Modelinin KOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	10,4202	340,0369	18,4401	24,252	0,0005
16	10,7355	382,5686	19,5594	21,2201	-0,1246
32	9,9851	337,2981	18,3657	21,4996	0,0085
64	10,5524	370,48	19,2479	21,2819	-0,089
128	10,3386	347,2435	18,6345	22,3068	-0,0207
256	12,7316	366,0783	19,1332	32,7713	-0,0761
512	11,7226	348,6731	18,6728	29,1552	-0,0249
1024	27,6226	948,0428	30,7903	75,6924	-1,7868
2048	271,4626	74.031,17	272,0867	705,8655	-216,614

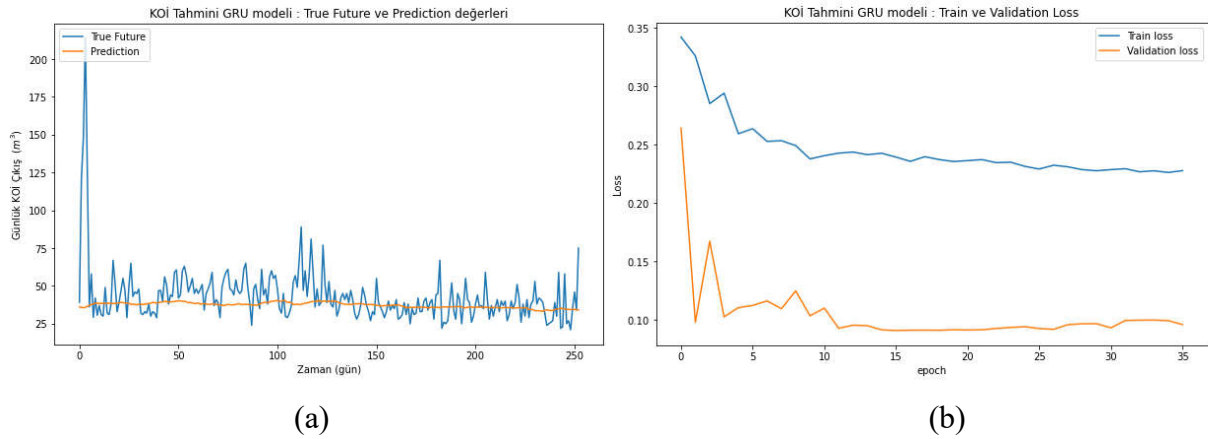
Stateless GRU modeli ile KOİ miktarının tahmini için yapılan çalışmada MAPE metriğine göre en başarılı sonuç 16 ünite ile; MAE, MSE, RMSE ve R^2 metriklerine göre en başarılı sonuç 32 ünite ile oluşturulan model ile elde edilmiştir.

Stateless GRU modeli ile KOİ tahminlemede MAPE metriğine göre en başarılı sonuçların alındığı 16 ünitelerden oluşan stateless modelin tahmin ve hata grafiği Şekil 68’de verilmiştir.



Şekil 68. KOİ tahmini için 16 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.

Stateless GRU modeli ile KOİ tahminlemede MAE, MSE, RMSE ve R^2 metriklerine göre en başarılı sonuçların alındığı 32 ünitelerden oluşan stateless modelin tahmin ve hata grafiği Şekil 69’da verilmiştir.



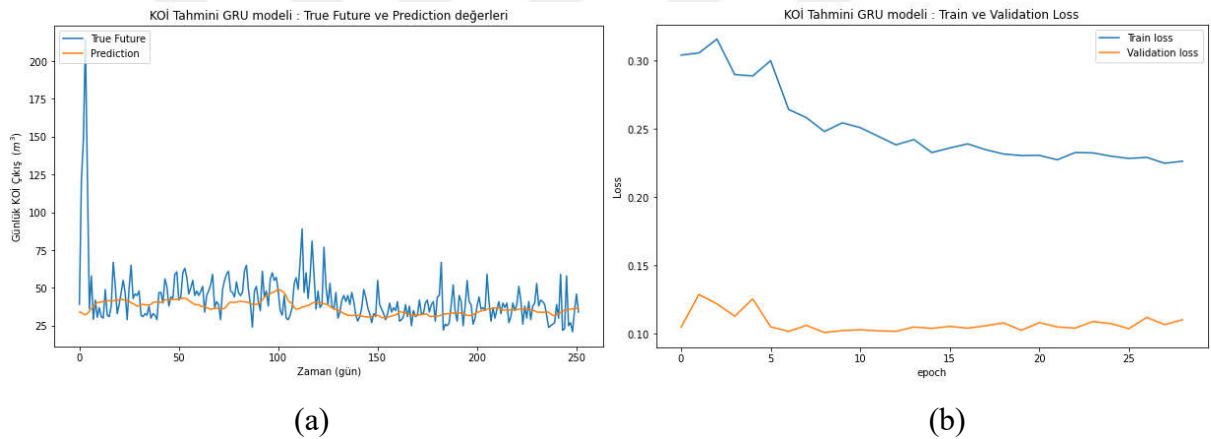
Şekil 69. KOİ tahmini için 32 ünite ile oluşan stateless GRU modelinin tahmin ve hata grafiği.

Çalışmada KOİ tahmini için kullanılan stateful GRU modelinin tahmin sonuçları Tablo 20’de verilmiştir.

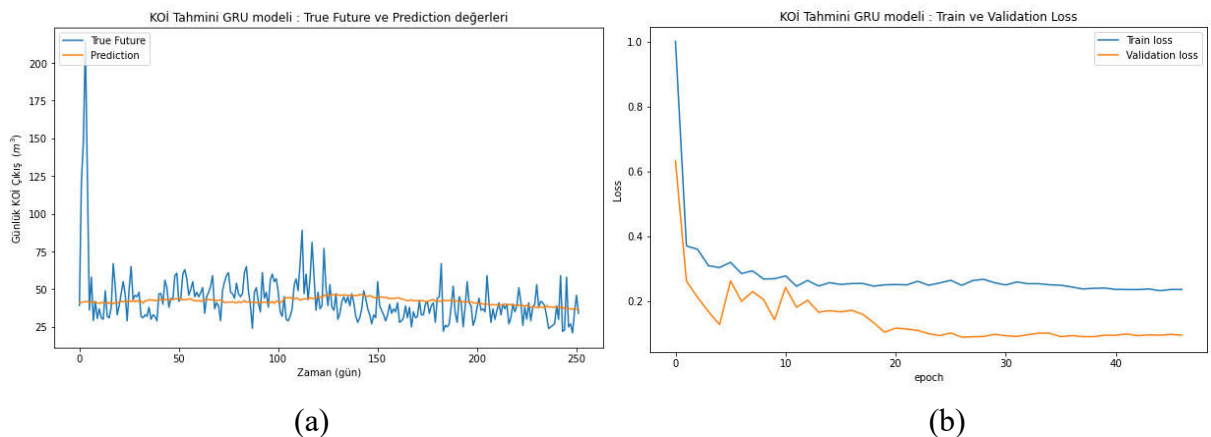
Tablo 20. Stateful GRU Modelinin KOİ Tahmin Sonuçları.

Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²
8	13,6471	783,36	27,9886	32,4423	-0,0124
16	11,086	362,3493	19,0355	25,7946	-0,0736
32	10,4319	376,0722	19,3926	21,029	-0,1142
64	11,0916	396,2305	19,9055	21,6471	-0,174
128	11,2563	341,6158	18,4829	27,6554	-0,0122
256	10,4764	337,159	18,3619	23,8037	0,001
512	51,0248	2.748,97	52,4306	137,5954	-7,1448
1024	57,4486	3.447,80	58,7179	154,5499	-9,2153
2048	122,1267	15.159,70	123,1247	323,1429	-43,916

Stateful GRU modeli ile KOİ miktarının tahmini için yapılan çalışmada MAE ve MAPE metriklerine göre en başarılı sonuç 32 ünite ile oluşturulan model ile; MSE, RMSE ve R² metriklerine göre en başarılı sonuç 256 ünite ile oluşturulan model ile elde edilmiştir. MAE ve MAPE metriklerine göre en başarılı sonuçların alındığı 32 ünitelerden oluşan stateful modelin tahmin ve hata grafiği Şekil 70’de verilmiştir.

**Şekil 70.** KOİ tahmini için 32 ünite ile oluşan stateful GRU modelinin tahmin ve hata grafiği.

Stateful GRU modeli ile KOİ tahminlemesinde MSE, RMSE ve R² metriklerine en başarılı sonuçların alındığı 256 ünitelerden oluşan stateful modelin tahmin ve hata grafiği Şekil 71’de verilmiştir.

**Şekil 71.** KOİ tahmini için 256 ünite ile oluşan stateful GRU modelinin tahmin ve hata grafiği.

Modellerin Performans özeti

Çalışılan RNN, LSTM ve GRU modellerinin MAPE metriğine göre sıralanması ile en iyi BOİ tahmin sonuçları Tablo 21’de, en iyi KOİ tahmin sonuçları Tablo 22’de ve literatürde yapılan çalışmaların BOİ ve KOİ tahmin sonuçları ise Tablo 23 ve Tablo 24’te özetlenmiştir.

Tablo 21. Çalışılan Modellerin En İyi BOİ Tahmin Sonuçları.

Model	Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²	Başarı Oranı (%)
Stateless GRU	256	4,632	34,0073	5,8316	20,8385	-0,1182	79,1615
Stateless RNN	256	4,7155	35,5992	5,9665	21,0372	-0,1705	78,9628
Stateless LSTM	2048	4,7478	36,0496	6,0041	21,0809	-0,1854	78,9191
Statefull GRU	128	4,3434	27,1374	5,2094	21,9196	0,0974	78,0804
Statefull LSTM	2048	4,4749	30,1315	5,4892	21,9428	-0,0021	78,0572
Stateless GRU	32	4,2401	27,6266	5,2561	22,0336	0,0916	77,9664
Statefull RNN	128	4,762	31,9928	5,6562	24,7959	-0,0641	75,2041

Tablo 22. Çalışılan Modellerin En İyi KOİ Tahmin Sonuçları.

Model	Ünite sayısı	MAE	MSE	RMSE	MAPE	R ²	Başarı Oranı (%)
Statefull LSTM	16	11,1207	824,2519	28,7098	20,9574	-0,0653	79,0426
Statefull GRU	32	10,4319	376,0722	19,3926	21,029	-0,1142	78,9710
Stateless GRU	16	10,7355	382,5686	19,5594	21,2201	-0,1246	78,7799
Stateless GRU	32	9,9851	337,2981	18,3657	21,4996	0,0085	78,5004
Statefull LSTM	32	10,3026	356,1463	18,8718	22,206	-0,0552	77,7940
Stateless LSTM	1024	10,5626	343,6825	18,5387	23,6489	-0,0103	76,3511
Statefull GRU	256	10,4764	337,159	18,3619	23,8037	0,001	76,1963
Stateless LSTM	16	10,4314	338,6465	18,4024	24,0657	0,0045	75,9343
Stateless RNN	256	10,8274	340,8326	18,4617	25,1225	-0,0019	74,8775
Statefull RNN	32	12,2636	806,8796	28,4056	26,6167	-0,0428	73,3833
Statefull RNN	64	11,4745	348,9716	18,6808	28,6659	-0,034	71,3341

Tablo 23. Literatürde Yapılan Çalışmaların BOİ Tahmin Sonuçları.

Çalışma	Parametre	Model	Kullanılan Metrikler				R ²
			MAE	MSE	RMSE	MAPE	
El-Rawy vd, 2021	BOİ	LSTM	-	-	1,92	-	-
Anuar vd, 2021	BOİ	LSTM	-	-	2,7382	-	-
Khullar vd, 2022	BOİ	Bi-LSTM	0,124	0,107	0,108	18,22	-
Lim vd, 2019	BOİ	LSTM	-	-	0,0000080	-	0,931
Hamoda vd, 1999	BOİ	ANN	-	-	-	2,7	0,72
Yu vd, 2018	BOİ	KPCA-ELM	1,125	1,340	-	1,321	-
Bu çalışma	BOİ	GRU	4,632	34,0073	5,8316	20,8385	-0,1182

Tablo 24. Literatürde Yapılan Çalışmaların KOİ Tahmin Sonuçları.

Çalışma	Parametre	Model	Kullanılan Metrikler				R ²
			MAE	MSE	RMSE	MAPE	
El-Rawy vd, 2021	KOİ	LSTM	-	-	3,36	-	-
Anuar vd, 2021	KOİ	LSTM	-	-	3,6519	-	-
Khullar vd, 2022	KOİ	Bi-LSTM	0,115	0,015	0,117	20.32	-
Ye vd, 2019	KOİ	LSTM-RNN	-	-	1,174	0,594	-
Lim vd, 2019	KOİ	LSTM	-	-	0,0000186	-	0,994
Öztemek vd, 2016	KOİ	ANN	-	-	-	13,5	-
Yu vd, 2018	KOİ	KPCA-ELM	2,322	3,108	-	1,223	-
Bu çalışma	KOİ	LSTM	11,1207	824,2519	28,7098	20,9574	-0,0653

Modellerin Eğitim Süreleri

Erzurum Atıksu arıtma tesislerinde arıtılan suda bulunan BOİ ve KOİ miktarı tahmini için test edilen stateless ve stateful RNN, LSTM ve GRU modellerinin eğitim sürelerinin incelemesi bu bölümünde özetlenmiştir.

Çalışmada BOİ ve KOİ tahmini için kullanılan stateless ve stateful RNN modellerinin eğitim süreleri Tablo 25’te verilmiştir.

Tablo 25. RNN Modelinin Eğitim Süreleri

Ünite Sayısı	BOİ		KOİ	
	Stateless Model Eğitim Süresi (Saniye)	Stateful Model Eğitim Süresi (Saniye)	Stateless Model Eğitim Süresi (Saniye)	Stateful Model Eğitim Süresi (Saniye)
8	51,838771	42,6214	90,70049	66,8298
16	59,263516	62,6365	54,74663	31,1389
32	50,163932	46,5583	69,6035	36,6305
64	86,267273	45,5686	93,26007	38,9135
128	82,155179	31,6598	57,4665	21,2129
256	67,247059	65,1186	91,0602	52,8401
512	63,274651	33,5787	67,44232	45,7995
1024	82,356462	59,9924	87,57968	27,8458
2048	183,632751	53,0845	119,4716	94,9011

BOİ tahmini için kullanılan RNN modellerinden 32 ünitelerden oluşan stateless modeli 50,16 saniye ile, 128 ünitelerden oluşan stateful modeli ise 31,65 saniye ile eğitimi en az sürede tamamlayan modellerdir.

KOİ tahmini için kullanılan RNN modellerinden 16 ünitelerden oluşan stateless modeli 54,74 saniye ile, 128 ünitelerden oluşan stateful modeli ise 21,21 saniye ile eğitimi en az sürede tamamlayan modellerdir

Çalışmada BOİ ve KOİ tahmini için kullanılan stateless ve stateful LSTM modellerinin eğitim süreleri Tablo 26’da verilmiştir.

Tablo 26. LSTM Modelinin Eğitim Süreleri

Ünite Sayısı	BOİ		KOİ	
	Stateless Model Eğitim Süresi (Saniye)	Stateful Model Eğitim Süresi (Saniye)	Stateless Model Eğitim Süresi (Saniye)	Stateful Model Eğitim Süresi (Saniye)
8	41,135191	26,8746	36,166961	25,9098
16	39,229138	40,7258	40,998828	20,2997
32	36,847597	19,4907	46,182772	17,6841
64	39,063343	22,7490	47,680170	21,5139
128	40,207315	26,7036	46,383834	24,1202
256	42,074165	19,7845	50,209001	20,6681
512	44,689336	22,2200	48,676584	33,1516
1024	95,304478	53,2812	143,521807	65,7724
2048	470,949028	259,0500	306,397151	143,3059

BOİ tahmini için kullanılan LSTM modellerinden 32 ünitelerden oluşan stateless modeli 36,84 saniye ile, 32 ünitelerden oluşan stateful modeli ise 19,49 saniye ile eğitimi en az sürede tamamlayan modellerdir.

KOİ tahmini için kullanılan LSTM modellerinden 8 ünitelerden oluşan stateless modeli 36,16 saniye ile, 32 ünitelerden oluşan stateful modeli ise 17,68 saniye ile eğitimi en az sürede tamamlayan modellerdir.

Çalışmada BOİ ve KOİ tahmini için kullanılan stateless ve stateful GRU modellerinin eğitim süreleri Tablo 27’de verilmiştir.

Tablo 27. GRU Modelinin Eğitim Süreleri

Ünite Sayısı	BOİ		KOİ	
	Stateless Model Eğitim Süresi (Saniye)	Stateful Model Eğitim Süresi (Saniye)	Stateless Model Eğitim Süresi (Saniye)	Stateful Model Eğitim Süresi (Saniye)
8	44,702037	51,3040	59,400861	21,0294
16	51,926117	40,3418	53,66429	20,5742
32	54,399376	49,7696	57,535039	26,8738
64	50,093552	22,6428	71,637048	32,5973
128	59,953925	49,2043	48,994134	35,5499
256	60,940513	27,7792	54,867949	44,9186
512	90,867604	32,4185	68,76783	23,1708
1024	175,752943	67,0971	106,087451	51,3399
2048	504,125188	136,8600	472,023378	159,0845

BOİ tahmini için kullanılan GRU modellerinden 8 ünitelerden oluşan stateless modeli 44,70 saniye ile, 64 ünitelerden oluşan stateful modeli ise 22,64 saniye ile eğitimi en az sürede tamamlayan modellerdir.

KOİ tahmini için kullanılan GRU modellerinden 128 ünitelerden oluşan stateless modeli 48,99 saniye ile, 16 ünitelerden oluşan stateful modeli ise 20,57 saniye ile eğitimi en az sürede tamamlayan modellerdir.

SONUÇLAR

Bu tez çalışmasında; derin öğrenme modelinin zaman serisi tekrarlayan sinir ağı yöntemlerinden RNN, LSTM ve GRU modellerini kullanılarak ERBASAT'ın arıtılmış suyunda bulunan biyolojik oksijen ihtiyacı (BOİ) ve kimyasal oksijen ihtiyacı (KOİ) miktarlarının tahmin edilmesi amaçlanmıştır. Çalışılan modeller stateful ve stateless olarak ve her model 2 gizli katmana sahip olacak şekilde tasarlanmıştır. Çalışmada, ERBASAT'da ölçülen 943 günlük deney sonuçlarının eksik verilerinin tamamlanması, aykırı değerlerin düzenlenmesi ve normalize edilmesinden sonra verinin %70'i eğitim, %30'u test için kullanılmıştır. Çalışılan modellerin tahminlemedeki başarı performanslarının değerlendirilmesi ve model tahmin sonuçlarının gerçek değerler ile karşılaştırılması için MAE, MSE, RMSE, MAPE ve R^2 metrikleri kullanılmıştır. Tez çalışmasında yapılan BOİ ve KOİ tahminlemeleri sonucunda aşağıdaki bulgular tespit edilmiştir:

- R^2 metriği göz önünde bulundurulduğunda, çalışılan 3 modelinde tahmin sonuçları gerçek değerler ile yüksek korelasyon göstermemiştir.
- Çalışılan modeller içerisinde BOİ parametresini MAPE metriğine göre en yüksek doğrulukta GRU modeli %79,16 başarı ile tahmin edebilmiştir.
- Çalışılan modeller içerisinde KOİ parametresini MAPE metriğine göre en yüksek doğrulukta LSTM modeli %79,04 başarı ile tahmin edebilmiştir.
- Modellerin stateless ve stateful olarak tasarlanmasının model başarısını gözlemlenebilir bir derecede etkilemediği ancak stateful modellerin eğitim süreleri stateless modellerin eğitim süresinden daha az olduğu görülmüştür.

Elde edilen bulgular; çalışılan RNN, LSTM ve GRU derin öğrenme modellerinin atıksu arıtma tesislerinin BOİ ve KOİ performans parametrelerinin tahmin edilmesinde kullanılabilecek alternatif bir yöntem olduğunu, modellerin eğitiminin hızlı yapılabilmesi için stateful olarak tasarlanmalarının daha uygun olacağını göstermektedir. Soğuk hava koşullarında çamurdaki bakterinin üremesi yavaş olduğundan, Erzurum Atıksu arıtma tesisinin sert ve uzun kış şartlarına bağlı olarak tesisin işletilmesinde birtakım zorluklar yaşanmaktadır. Erzurum Biyolojik Atıksu arıtma tesisi hem rakım hem de iklim olarak farklı bir konumda bulunduğundan, yapılan çalışma yüksek rakım ve uzun kış şartlarında bulunan AAT çıkış suyunun konsantrasyonlarının tahminlemesi çalışmalarına faydalar sağlayacaktır. Tez çalışmasında kullanılan modellerden elde edilen tahmin performansları gelecekte;

- Kış mevsimlerine ait verilerin zaman serisi dokusunu bozmayacak uygun bir yöntemle veri setinden çıkarılması,
- Eksik verilerin giderilmesi için farklı yöntemler kullanılması,
- Modellerin gizli katman sayılarının artırılması
- Veri seti çerçeve dizisinin oluşturulması için kullanılan çerçeve boyutunun (t) farklı büyüklükler kullanılması

gibi yöntemlerle kontrol edilerek modellerin performansları iyileştirmeleri için yöntemler araştıracaktır. Yapılacak çalışmalardan elde edilen sonuçlar istatistiksel yöntemler ile karşılaştırılarak performans değerlendirilmesi yapılacaktır.



KAYNAKLAR

- Abbasimehr, H., Shabani, M., & Yousefi, M. (2020). An optimized model using LSTM network for demand forecasting. *Computers & industrial engineering*, 143, 106435.
- Abdullah, S. N., & Zeng, X. (2010). Machine learning approach for crude oil price prediction with Artificial Neural Networks-Quantitative (ANN-Q) model. *The 2010 International Joint Conference on Neural Networks (IJCNN)*, (s. 1–8).
- Ahmed, A. M. (2017). Prediction of dissolved oxygen in Surma River by biochemical oxygen demand and chemical oxygen demand using the artificial neural networks (ANNs). *Journal of King Saud University-Engineering Sciences*, 29, 151–158.
- Aksungur, N., & Firidin, Ş. (2008). Su kaynaklarının kullanımı ve sürdürülebilirlik. *Aquaculture Studies*, 2008.
- Almalaq, A., & Edwards, G. (2017). A review of deep learning methods applied on load forecasting. *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, (s. 511–516).
- Althelaya, K. A., El-Alfy, E.-S. M., & Mohammed, S. (2018). Stock market forecast using multivariate analysis with bidirectional and stacked (LSTM, GRU). *2018 21st Saudi Computer Society National Computer Conference (NCC)*, (s. 1–7).
- Anuar, N. N., Fauzi, N. F., Ab Halim, H. Z., Khairudin, N. I., Bakhtiar, N. S., & Shafii, N. H. (2021). Implementation of Long-Short Term Memory Neural Network (LSTM) for Predicting The Water Quality Parameters in Sungai Selangor. *Journal of Computing Research and Innovation*, 6, 40–49.
- Aydın, S. (2020). Erzurum Biyolojik Atıksu Arıtma Tesisi Arıtma Çamuru Yönetiminin İncelenmesi. *Yüksek Lisans Tezi*.
- Basheer, I. A., & Hajmeer, M. (2000). Artificial neural networks: fundamentals, computing, design, and application. *Journal of microbiological methods*, 43, 3–31.
- Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Cilt 1). MIT press Massachusetts, USA:.
- Chen, Q., Chai, W., & Qiao, J. (2010). Modeling of wastewater treatment process using recurrent neural network. *2010 8th World Congress on Intelligent Control and Automation*, (s. 5872–5876).
- Chollet, F., & others. (2018). Keras: The python deep learning library. *Astrophysics source code library*, ascl–1806.
- Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and trends in signal processing*, 7, 197–387.
- Dingli, A., & Fournier, K. S. (2017). Financial time series forecasting—a deep learning approach. *International Journal of Machine Learning and Computing*, 7, 118–122.
- Duarte, G. R., da Fonseca, L. G., Goliatt, P. V., & de Castro Lemonge, A. C. (2017). Comparison of machine learning techniques for predicting energy loads in buildings. *Ambiente Construído*, 17, 103–115.
- El-Rawy, M., Abd-Ellah, M. K., Fathi, H., & Ahmed, A. K. (2021). Forecasting effluent and performance of wastewater treatment plant using different machine learning techniques. *Journal of Water Process Engineering*, 44, 102380.
- Erickson, B. J., Korfiatis, P., Akkus, Z., Kline, T., & Philbrick, K. (2017). Toolkits and libraries for deep learning. *Journal of digital imaging*, 30, 400–405.

- Ertam, F., & Aydın, G. (2017). Data classification with deep learning using Tensorflow. *2017 international conference on computer science and engineering (UBMK)*, (s. 755–758).
- ESKİ. (2021, December). Erzurum Su ve Kanal idaresi Genel Müdürlüğü. *Erzurum Su ve Kanal idaresi Genel Müdürlüğü*. <http://www.eski.gov.tr/> adresinden alındı
- Fang, X., & Yuan, Z. (2019). Performance enhancing techniques for deep learning models in time series forecasting. *Engineering Applications of Artificial Intelligence*, 85, 533–542.
- François-Lavet, V., Henderson, P., Islam, R., Bellemare, M. G., & Pineau, J. (2018). An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*.
- Fu, R., Zhang, Z., & Li, L. (2016). Using LSTM and GRU neural network methods for traffic flow prediction. *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, (s. 324–328).
- Fukushima, K. (1975). Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, 20, 121–136.
- Gücker, B., Brauns, M., & Pusch, M. T. (2006). Effects of wastewater treatment plant discharge on ecosystem structure and function of lowland streams. *Journal of the North American benthological society*, 25, 313–329.
- Hamoda, M. F., Al-Ghusain, I. A., & Hassan, A. H. (1999). Integrated wastewater treatment plant performance evaluation using artificial neural networks. *Water Science and Technology*, 40, 55–65.
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9, 1735–1780.
- Hou, L., Samaras, D., Kurc, T. M., Gao, Y., & Saltz, J. H. (2016). Neural networks with smooth adaptive activation functions for regression. *arXiv preprint arXiv:1608.06557*.
- Islam, M. S., & Hossain, E. (2020). Foreign exchange currency rate prediction using a GRU-LSTM Hybrid Network. *Soft Computing Letters*, 100009.
- Jiang, Y., Li, C., Sun, L., Guo, D., Zhang, Y., & Wang, W. (2021). A deep learning algorithm for multi-source data fusion to predict water quality of urban sewer networks. *Journal of Cleaner Production*, 318, 128533.
- Kaiser, J. (2014). Dealing with Missing Values in Data. *Journal of Systems Integration (1804-2724)*, 5.
- Khan, M., Jan, B., & Farman, H. (2019). *Deep Learning: Convergence to Big Data Analytics*. Springer.
- Khullar, S., & Singh, N. (2022). Water quality assessment of a river using deep learning Bi-LSTM methodology: forecasting and validation. *Environmental Science and Pollution Research*, 29, 12875–12889.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097–1105.
- Li, Q., Yang, Y., Guo, Y., Li, W., Liu, Y., Liu, H., & Kang, Y. (2021). Performance Evaluation of Deep Learning Classification Network for Image Features. *IEEE Access*, 9, 9318–9333.
- Lim, H., An, H., Kim, H., & Lee, J. (2019). Prediction of pollution loads in the Geum River upstream using the recurrent neural network algorithm. *Korean Journal of Agricultural Science*, 46, 67–78.

- Liu, Y., Wang, Z., & Zheng, B. (2019). Application of regularized GRU-LSTM model in stock price prediction. *2019 IEEE 5th International Conference on Computer and Communications (ICCC)*, (s. 1886–1890).
- Maddocks, A., Young, R. S., & Reig, P. (2015). Ranking the world's most water-stressed countries in 2040.
- Masood, A., & Ahmad, K. (2021). A review on emerging artificial intelligence (AI) techniques for air pollution forecasting: Fundamentals, application and performance. *Journal of Cleaner Production*, 322, 129072.
- McCorduck, P., Minsky, M., Selfridge, O. G., & Simon, H. A. (1977). History of artificial intelligence. *IJCAI*, (s. 951–954).
- Mittal, S., & Sangwan, O. P. (2019). Big data analytics using machine learning techniques. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, (s. 203–207).
- Mocanu, E., Nguyen, P. H., Gibescu, M., & Kling, W. L. (2016). Deep learning for estimating building energy consumption. *Sustainable Energy, Grids and Networks*, 6, 91–99.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. *Icml*.
- Negnevitsky, M., & Intelligence, A. (2005). A guide to intelligent systems. *Artificial Intelligence*.
- Ng, A. (2013). Machine Learning and AI via Brain simulations. *Accessed: May, 3, 2018*.
- Nourani, V., Asghari, P., & Sharghi, E. (2021). Artificial intelligence based ensemble modeling of wastewater treatment plant using jittered data. *Journal of Cleaner Production*, 291, 125772.
- Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.
- Olah, C. (2015). Understanding LSTM Networks Colah's Blog. *Colah. github. io*.
- Osman Tuğrul, B. A., & Egemen, A. R. (2018). Atıksu Aritma Tesislerinde Biyokimyasal Oksijen İhtiyacının Farklı Regresyon Modelleriyle Tahmin Edilmesi. *Engineering Sciences*, 13, 96–105.
- Öztemel, E. (2003). Yapay sinir ağları. *PapatyaYayincılık, İstanbul*.
- Öztemel, E., & DÜGENCİ, M. (2016). Atıksu arıtma tesis kontrolde yapay sinir ağı ile kirlilik parametre tahmini. *3rd*.
- Öztürk, K., & Şahin, M. E. (2018). Yapay sinir ağları ve yapay zekâ'ya genel bir bakış. *Takvim-i Vekayi*, 6, 25–36.
- Pirim, A. G. (2006). Yapay zeka. *Journal of Yaşar University*, 1, 81–93.
- Qiang, W., & Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, (s. 1143–1146).
- Rashid, J., & Waheed, K. (2020). Missing Values and Outliers in Research Data. *Pakistan Postgraduate Medical Journal*, 31, 167–167.
- Ray, S. (2019). A quick review of machine learning algorithms. *2019 International conference on machine learning, big data, cloud and parallel computing (COMITCon)*, (s. 35–39).

- Rice, E. W., Baird, R. B., Eaton, A. D., Clesceri, L. S., & others. (2012). *Standard methods for the examination of water and wastewater* (Cilt 10). American public health association Washington, DC.
- Ridwan, W. M., Sapitang, M., Aziz, A., Kushiar, K. F., Ahmed, A. N., & El-Shafie, A. (2021). Rainfall forecasting model using machine learning methods: Case study Terengganu, Malaysia. *Ain Shams Engineering Journal*, 12, 1651–1663.
- Sabak, S., & Başar, Ö. D. (2020). Enflasyonu Etkileyen Faktörlerin Ülkeler Bazında Yapay Sinir Ağları Analizi İle İncelenmesi. *Working Paper Series*, 1, 27–38.
- Sagheer, A., & Kotb, M. (2019). Time series forecasting of petroleum production using deep LSTM recurrent networks. *Neurocomputing*, 323, 203–213.
- Serengil, S. I. (2017, 10 14). *Evolution of Neural Networks and Deep Learning*. 01 2021, 08 tarihinde <https://sefiks.com/2017/10/14/evolution-of-neural-networks/> adresinden alındı
- Serengil, S. I. (2017). Evolution of Neural Networks and Deep Learning | sefiks.com. *Evolution of Neural Networks and Deep Learning* | sefiks.com.
- Sezer, M. (2007). *Yapay sinir ağları (YSA) kullanılarak KOİ parametresinden BOİ parametresinin tahmin edilmesi*. Master's thesis, Sakarya Üniversitesi.
- Sezer, O. B., Gudelek, M. U., & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90, 106181.
- Siarni-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, (s. 1394–1401).
- Sibil, R., Berkun, M., & Bekiroglu, S. (2014). The comparison of different mathematical methods to determine the BOD parameters, a new developed method and impacts of these parameters variations on the design of WWTPs. *Applied Mathematical Modelling*, 38, 641–658.
- Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, (s. 1310–1315).
- Singh, P., Jishnu Jaykumar, P., Pankaj, A., & Mitra, R. (2021). Edge-Detect: Edge-Centric Network Intrusion Detection using Deep Neural Network. *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, (s. 1-6). <https://doi.org/10.1109/CCNC49032.2021.9369469>
- Skansi, S. (2018). *Introduction to Deep Learning: from logical calculus to artificial intelligence*. Springer.
- Tokgöz, A., & Ünal, G. (2018). A RNN based time series approach for forecasting turkish electricity load. *2018 26th Signal Processing and Communications Applications Conference (SIU)*, (s. 1–4).
- Turing, A. M. (2009). Computing machinery and intelligence. *Parsing the turing test* (s. 23–65). içinde Springer.
- Usama, M., Qadir, J., Raza, A., Arif, H., Yau, K.-L. A., Elkhatib, Y., . . . Al-Fuqaha, A. (2019). Unsupervised machine learning for networking: Techniques, applications and research challenges. *IEEE access*, 7, 65579–65615.

- Vijayaprabakaran, K., & Sathiyamurthy, K. (2020). Towards activation function search for long short-term model network: a differential evolution based approach. *Journal of King Saud University-Computer and Information Sciences*.
- Vivas, E., Allende-Cid, H., & Salas, R. (2020). A systematic review of statistical and machine learning methods for electrical power forecasting with reported mape score. *Entropy*, 22, 1412.
- Wang, J., Ma, Y., Zhang, L., Gao, R. X., & Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48, 144–156.
- Wang, Z., Man, Y., Hu, Y., Li, J., Hong, M., & Cui, P. (2019). A deep learning based dynamic COD prediction model for urban sewage. *Environmental Science: Water Research & Technology*, 5, 2210–2218.
- Yang, J., & Chen, B. (2021). Energy efficiency evaluation of wastewater treatment plants (WWTPs) based on data envelopment analysis. *Applied Energy*, 289, 116680.
- Yang, S., Yu, X., & Zhou, Y. (2020). LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example. *2020 International workshop on electronic communication and artificial intelligence (IWECAI)*, (s. 98–101).
- Yann LeCun, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436.
- Ye, Q., Yang, X., Chen, C., & Wang, J. (2019). River water quality parameters prediction method based on LSTM-RNN model. *2019 Chinese Control And Decision Conference (CCDC)*, (s. 3024–3028).
- Yilmaz, E. C., & Dogan, E. (2008). Artificial neural networks application for modelling of wastewater treatment plant performance. *Electronic Letters on Science and Engineering*, 4, 1–9.
- Yu, T., Yang, S., Bai, Y., Gao, X., & Li, C. (2018). Inlet water quality forecasting of wastewater treatment based on kernel principal component analysis and an extreme learning machine. *Water*, 10, 873.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı:	Sırrı BALLI
Doğum tarihi:	
Doğum Yeri:	
Uyruğu:	
Adres:	
Tel:	
E-mail:	
Eğitim	
Lise:	Atatürk Teknik Lisesi
Lisans:	Doğu Akdeniz Üniversitesi
Yüksek lisans:	Atatürk Üniversitesi Fen Bilimleri Enstitüsü
Doktora:	
Yabancı Dil Bilgisi	
İngilizce:	
Almanca:	
Rusça:	
Diğer	
Üye Olunan Mesleki Kuruluşlar	
Tezden Üretilmiş Yayınlar	