T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DENGESİZ VERİ SETLERİNDE AŞIRI ÖRNEKLEME TEKNİKLERİ İLE MAKİNE ÖĞRENMESİ YAKLAŞIMLARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Ümit DİLBAZ

Akıllı Sistemler Mühendisliği Anabilim Dalı

OCAK 2023

T.C.
BURSA TEKNİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

DENGESİZ VERİ SETLERİNDE AŞIRI ÖRNEKLEME TEKNİKLERİ İLE MAKİNE ÖĞRENMESİ YAKLAŞIMLARININ KARŞILAŞTIRILMASI

YÜKSEK LİSANS TEZİ

Ümit DİLBAZ
(19324814024)

Akıllı Sistemler Mühendisliği Anabilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Mustafa Özgür CİNGİZ

OCAK 2023

BTÜ, Lisansüstü Eğitim Enstitüsü'nün 19324814024 numaralı Yüksek Lisans Öğrencisi Ümit DİLBAZ, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "DENGESİZ VERİ SETLERİNDE AŞIRI ÖRNEKLEME TEKNİKLERİ İLE MAKİNE ÖĞRENMESİ YAKLAŞIMLARININ KARŞILAŞTIRILMASI" başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı: **Dr. Öğr. Üyesi Mustafa Özgür CİNGİZ** .....
Bursa Teknik Üniversitesi

Jüri Üyeleri: **Prof. Dr. Turgay Tugay BİLGİN** .....
Bursa Teknik Üniversitesi

Doç. Dr. Murtaza CİCİOĞLU .....
Uludağ Üniversitesi

Teslim Tarihi :
Savunma Tarihi : 27 Ocak 2023

20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, Bursa Teknik Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Lisansüstü Eğitim Enstitüsü'nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

İNTİHAL BEYANI

Bu tezde görsel, işitsel ve yazılı biçimde sunulan tüm bilgi ve sonuçların akademik ve etik kurallara uyularak tarafımdan elde edildiğini, tez içinde yer alan ancak bu çalışmaya özgü olmayan tüm sonuç ve bilgileri tezde kaynak göstererek belgelediğimi, aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim.

Ümit DİLBAZ

Eşime ve oğluma,

ÖNSÖZ

Nesnelerin interneti kavramının hayatımıza girmesi ile birlikte birçok alanda dijital ve fiziksel sistemlerin entegrasyonu her geçen gün katlanarak artmış ve bu sayede insanlığa büyük faydalar sağlayan teknolojik araçlar ve metotlar daha da etkin ve güçlü hale gelmiştir. Özellikle son dönemlerde gelişen siber fiziksel sistemlerin sunduğu veriye erişebilirlik olanakları, veri bilimi ve yapay zekâ alanındaki gelişmeleri arkasına aldığında birçok alanda kişi ve kurumlara karar destek mekanizması olarak finansal ve stratejik kazanımların yolunu açmıştır. Şüphesiz ki bu imkanlardan yaygın bir şekilde faydalanılan alanlardan biri de endüstriyel tesislerin iş verimliliğini ve üretkenliğini arttıran bakım stratejileri ve uygulamalarıdır. Bu çalışmada, yapay zekâ yöntemlerinden faydalanılarak kestirimci bakım uygulamaları için oluşturulan sınıflandırma modellerinin performansını en iyi hale getirebilmek maksadıyla farklı yaklaşımlar denenmiştir. Ele alınan yaklaşımların model performansları üzerindeki olumlu ve olumsuz etkileri incelenip ortaya konularak, bu alanda yapılacak olan çalışmalara katkı sağlanılması amaçlanmıştır.

Öncelikle, profesyonel iş hayatımı ve ilgi alanımı göz önünde bulundurarak üzerinde çalışmak istediğim tez konusu hususunda benden desteğini hiç esirgemeyen ve kendi tecrübeleri ışığında beni sürekli yönlendirerek çalışmamı bilimsel temeller çerçevesinde tamamlamamı sağlayan saygıdeğer danışman hocam Dr. Öğr. Üyesi Mustafa Özgür CİNGİZ'e saygı ve teşekkürlerimi sunarım.

Ayrıca eğitim öğretim hayatım boyunca her daim yanımda olan ve bugünlere gelmemde maddi ve manevi desteklerini hiçbir zaman esirgemeyen sevgili anneme, babama ve kardeşlerime teşekkür ederim.

Son olarak, hayatımın her evresinde olduğu gibi, tez çalışmamı hazırlarken de desteğini bir an olsun eksiltmeyen değerli eşime ve oğluma sonsuz teşekkür ederim.

Ocak 2023

Ümit DİLBAZ
Elektronik ve Haberleşme Mühendisi

İÇİNDEKİLER

Sayfa

KISALTMALAR

AdaBoost : Adaptive Boosting
ADASYN : Adaptive Synthetic
ANN : Artificial Neural Network
CatBoost : Categorical Bossting
CLSTM : CNN + LSTM
CNN : Convolutional Neural Network
CRA : Collaborative Recommendation Approach
DL : Deep Learning
DNN : Deep Neural Network
EFB : Exclusive Feature Bundling
ENN : Edited Nearest Neighbour
FFT : Fast Fourier Transform
FSPS : Frequency Spectrum Partition Summation
GB : Gradient Boosting
GBM : Gradient Boosting Machine
GOSS : Gradient Based One Side Sampling
IMS : Intelligent Maintenance Systems
IoT : Internet of Things
IR : Imbalanced Ratio
KNN : K-Nearest Neighbour
LGBM : LightGBM
LSTM : Long Short-Term Memory
ML : Machine Learning
MLP : Multi Layer Perceptron
MSE : Mean Squared Error
NASA : National Aeronautics and Space Administration
PCA : Principal Component Analysis
ReLU : Rectified Linear Unit
RF : Random Forest
RMSE : Root Mean Square Error

RNN **:** Recurrent Neural Network

RUL **:** Remaining Useful Lifetime

SMOTE **:** Synthetic Minority Over-Sampling Technique

SVM **:** Support Vector Machines

SVRM : Support Vector Regression Machines

XGBoost **:** Extreme Gradient Boosting

SEMBOLLER

κ	**:** Cohen Kappa
σ	**:** Sigmoid Fonksiyonu

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

DENGESİZ VERİ SETLERİNDE AŞIRI ÖRNEKLEME TEKNİKLERİ İLE MAKİNE ÖĞRENMESİ YAKLAŞIMLARININ KARŞILAŞTIRILMASI

ÖZET

Endüstriyel bir tesisin faaliyetlerini kesintisiz bir şekilde devam ettirebilmesi için, o tesisi oluşturan ekipman ve sistemlerin kullanım ömürlerini uzatmak ve arızi duruşları engellemek amacıyla yapılan teknik ve idari işlerin tümüne bakım denir. Endüstriyel bakım faaliyetlerin yetersizliğinden ileri gelen plansız duruşların ve kazaların ortaya çıkardığı maliyetler işletmeler için ciddi riskler teşkil etmektedir. Birçok işletme geleneksel bakım yaklaşımları ile bu riskleri yönetmeye çalışsa da başarıları kısıtlı kalabilmektedir. Teknolojideki gelişmelerin ışığında bakım stratejilerini güncelleyen ve ileri taşıyan şirketler ilgili riskleri ve kayıpları daha etkin bir şekilde yönetme imkanına sahiptir.

Sahadan ve ekipmanlar üzerinden toplanan verilerin analiz edilerek potansiyel arızaların henüz oluşmadan önce tahmin edilebilmesine ve buna yönelik yürütülen bakım faaliyetlerine kestirimci bakım denir. Nesnelerin interneti (Internet of Things – IoT) alanındaki gelişmeler ve siber fiziksel sistemlerin entegrasyonu ile endüstriyel ekipmanlar üzerinden verilerin gerçek zamanlı bir şekilde toplanması kolaylaşmıştır. Bu verilerin yapay zekâ algoritmaları ile işlenmesinden elde edilen tahmine dayalı analitik çıkarımlar ise kestirimci bakım stratejilerine yeni bir boyut kazandırmıştır.

Kestirimci bakım ve arıza tespiti gibi problemlerde tahmine dayalı bir analitik model ortaya koymak istediğimizde, bu olayların doğası gereği dengesiz sınıf dağılımına sahip bir veri kümeleri karşımıza çıkar. Genellikle arıza olmayan bir durumu ifade eden gözlem sayısı, arıza durumunu temsil eden gözlemlerden çok fazladır. Dengesiz veri setlerinde yapay zekâ algoritmaları ile sınıflandırma yapmak önemli zorluklar içerir. Çünkü algoritmalar daha fazla sayıda gözlemin olduğu arıza olmama durumunu ifade eden bilgileri ezberleme eğiliminde olur. Bu sorun ise gerçek hayat uygulamasında algoritmalar ile arızaların tespit edilmesini zorlaştırır.

Bu çalışmada, endüstriyel ortamdaki ekipmanlardan toplanmış verileri temsil eden 2 farklı veri seti üzerinde yapay zekâ algoritmaları kullanılarak bir sınıflandırma görevi gerçekleştirilmiştir. Dengesiz sınıf verisi dağılımına sahip olan her iki veri setinde, aşırı uyumlanma (overfitting) probleminin önüne geçebilmek için eğitim verileri üzerinde çeşitli aşırı örnekleme yöntemler ve hibrit yöntemler denenerek veri setleri dengeli hale getirilmiştir. Dengelenmiş veri setleri kullanılarak bağımsız (tekil) makine öğrenme algoritmaları, kollektif öğrenmeye dayalı makine öğrenmesi algoritmaları ve derin öğrenme algoritmaları ile oluşturulan modeller vasıtasıyla sınıflandırma yapılmıştır. Modellerin başarı performansları başta Cohen Kappa skoru olmak üzere, F1 skoru, duyarlılık ve doğruluk metrikleri açısından değerlendirilmiştir. Kolektif öğrenmeye dayalı makine öğrenmesi algoritmalarının her iki veri setinde de diğer algoritmalardan daha yüksek performans gösterdiği görülmüştür. Ayrıca veri seti dengeleme yöntemlerinin, kolektif öğrenmeye dayalı makine öğrenmesi algoritmalarının başarı performansına etkisi incelenmiştir. Farklı

yöntemlerle dengelenen veri setlerinde kolektif öğrenme modellerinin performansları değişkenlik gösterirken, genel olarak rastgele örnekleme yöntemi ile dengeli hale getirilen veri setlerinde daha iyi performans elde edilmiştir. Yapılan bu tez çalışmasında, dengesiz veri setlerinde sınıflandırma görevi için model başarımına etki eden parametreler çok yönlü bakış açısıyla ortaya konulmuştur.

Anahtar kelimeler: Aşırı Örnekleme, Kestirimci Bakım, Kolektif Öğrenme, Derin Öğrenme, Makine Öğrenmesi, Yapay Zekâ

COMPARISON OF MACHINE LEARNING APPROACHES BY USING OVERSAMPLING TECHNIQUES ON IMBALANCED DATASETS

SUMMARY

Industrial maintenance covers all the technical and administrative operations to extend the lifetime of the equipments and avoid from unplanned system failures in order to ensure the manufacturing facilities uninterruptedly. The cost of unplanned breakdowns and accidents caused by insufficient maintenance operations brings serious risks for the industrial organizations. Although many facilities try to manage these risks with traditional maintenance approaches, their success may be limited. Companies that update and advance their maintenance strategies in the way of technological developments have the opportunity to manage related risks and losses more effectively.

The analysis of data collected from sensors and equipment, allowing for the prediction of potential failures before they occur, and the maintenance activities conducted based on these predictions, is called predictive maintenance. With the developments in the Internet of Things (IoT) and the integration of cyber-physical systems, it has become easier to collect data in real time via industrial equipment. Analytic insights based on the processing of this data with artificial intelligence algorithms have brought a new dimension to predictive maintenance strategies.

When we aim to devise an analytical model based on prediction for problems such as predictive maintenance and fault detection, we are faced with a dataset with an imbalanced class distribution, due to the nature of these events. The number of observations representing the non-faulty condition usually far exceeds the observations representing the faulty condition. Classifying imbalanced datasets using artificial intelligence algorithms poses significant challenges, as algorithms tend to overfit with the information representing the non-faulty condition, where there is a larger number of observations. This issue makes it difficult to detect failures using algorithms in real-life applications.

In this study, artificial intelligence algorithms were used to perform a classification task on two different datasets representing data collected from equipment in an industrial environment. To prevent the problem of overfitting on both datasets with imbalanced class data distributions, various oversampling methods and hybrid methods were applied on the training data to balance the datasets. Classification was performed using models created with standalone machine learning algorithms, ensemble learning algorithms, and deep learning algorithms, using the balanced datasets. The performance of the models was evaluated in terms of the Cohen Kappa score, F1 score, recall, and accuracy. Ensemble learning algorithms carried out higher performance than the other algorithms in both datasets. The effect of the data balancing methods on the performance of ensemble learning algorithms was also analysed. The performance of the ensemble learning models varied in the datasets balanced using different methods, but generally performed better in datasets balanced

using random sampling. In this thesis, the parameters affecting model performance for the classification task in imbalanced datasets were presented with a multidimensional perspective.

Keywords: Oversampling, Predictive Maintenance, Ensemble Learning, Deep Learning, Machine Learning, Artificial Intelligence

1. GİRİŞ

Bir tesisin veya sistemin, faaliyetlerini kesintisiz ve sorunsuz bir şekilde devam ettirebilmesi için o sistem üzerinde yapılan teknik, idari ve yönetsel faaliyetlerin tümüne endüstriyel bakım denir [1]. Üretim sektöründe etkin bir bakım sistemi; maliyet, ürün kalitesi, sektörel itibar gibi birçok kritik konuya direkt olarak etki etmektedir ve bu unsurların birleşimi olarak da organizasyonun rekabet gücünde önemli rol oynamaktadır. Bu durum ise makine ve ekipmanların sağlık durumlarının sürekli olarak takip edilmesini ve plansız duruşların oluşmadan önce gerekli önlemlerin alınmasını önemli kılmaktadır.

Geçmişten günümüze teknolojinin gelişmesiyle birlikte, endüstriyel bakım süreçlerinin yönetim ve uygulama biçimleri de değişim göstermiştir. İlk dönemlerde arıza meydana geldikten sonra bozulan parçayı değiştirme ya da onarma mantığıyla başlayan bakım süreçleri, artık sistemler üzerinden toplanan verilerin yapay zekâ algoritmalarıyla analiz edilerek muhtemel plansız duruşların çok önceden tespit edilmesine ve tam zamanında bakım felsefesine evrilmiştir. Potansiyel arızaların önceden tahmin edilmesine dayalı bu bakım türü ise kestirimci bakım ya da öngörücü bakım olarak ifade edilmektedir. Kestirimci bakım, üretim çıktı verimliliğinin artması, bakım maliyetlerinin düşürülmesi ve iş akış süreçlerinin iyileştirilmesi için gerekli bir yaklaşımdır.

Makine öğrenmesi (ML) ve makine öğrenmesinin alt türü olan derin öğrenme (DL) algoritmaları ile yoğun hacimde ve çeşitlilikteki veri yığınlarının anlamlandırılması ve bu anlamlandırılmış verilerin bir karar mekanizması olarak sunulması mümkündür [2].

Endüstri 4.0 çağı ile birlikte dijital ve fiziksel sistemlerin entegrasyonu hızla artmıştır [3]. Üretim ortamlarında ölçme, veri toplama ve işleme enstrümanlarının yaygınlaşması ile verinin gücünü arkasına alan bakım faaliyetleri yapay zekâ destekli uygulamalar ile son derece etkin hale gelmiştir.

1.1 Tezin Amacı ve Literatüre Katkısı

Kestirimci bakım, ilgili ekipmanlardan ve çevresel koşullardan elde edilen bilgilerin işlenip analiz edilerek faydalı bilgi çıkarımı yapılması esasına dayanır. Burada kestirimci bakımın etkinliği, toplanan verilerin yapısına ve amaçlanan çıktıya uygun olan doğru algoritma ve yöntemlerin seçilmesine bağlıdır [4]. Bir başka önemli husus ise, veri ön işleme aşamasında yapılan tüm veri manipülasyon işlemlerinin model performansı üzerine olan etkisidir.

Bir arızanın tespiti ve sınıflandırılması için toplanan veri kümeleri, doğası gereği dengesiz dağılmış veri sınıflarından oluşmaktadır. Verilerin büyük bir çoğunluğu genellikle arıza olmama durumuna ait iken, arıza durumuna ait verilerin oranı çok düşük kalır. Bu durum ise problemin çözümünde "doğruluk paradoksu" olarak karşımıza çıkar. Örneğin elimizde bulunan toplam 100 adet gözlemden sadece 2 adeti arıza olma durumunu temsil ediyorsa, modelimiz tüm sınıflandırma tahminlerini çoğunluk sınıfına ait "arıza olmama durumu" olarak sınıflandırsa model başarım doğruluğu %98 olacaktır. Ancak matematiksel olarak oldukça yüksek performanslı görünen bu tahmin modelinde, asıl olan arızalı ekipmanı tahmin etme amacı tamamen göz ardı edilmiş olur [5].

Yukarıda bahsedilen doğruluk paradoksu sebebiyle, dengesiz sınıf dağılımına sahip veri setleri makine öğrenmesi ya da derin öğrenme algoritmalarının başarım performansları açısından ciddi zorluklar içermektedir. Dengesiz veri setleri iyi yönetilemediğinde eğitim modellerinde başarımı yüksek görünmesine rağmen, gerçek hayat uygulamasında başarım oranı düşük uygulamalar ile neticelenebilir.

Bu çalışmada yapay zekâ modellerinin eğitilmesinde kullanılan dengesiz veriler ile başa çıkmak için literatürde yer alan çeşitli aşırı örnekleme yöntemleri denenmiştir ve bunların sınıflandırma modellerinin başarımları üzerine etkisi incelenmiştir. Ayrıca çalışmada kullanılan makine öğrenmesi ve derin öğrenme ile sınıflandırma modellerinin, birbirinden farklı 2 dengesiz veri seti üzerindeki performansları da gözlemlenerek, yüksek başarım oranı için modellere ilişkin genelleme yapılıp yapılamayacağı değerlendirilmiştir.

1.2 Tezin Organizasyonu

Bu çalışma 7 ana başlık altında toplanmıştır.

Giriş bölümünde tez çalışmasının amacı, literatür katkısı ve tezin organizasyonundan bahsedilmiştir.

İkinci bölümde ise literatürdeki klasik makine öğrenmesi ve derin öğrenme yöntemleri ile kestirimci bakım çalışmaları incelenmiştir ve çeşitli yaklaşımlara ilişkin araştırmalara yer verilmiştir.

Üçüncü bölümde, endüstriyel bakım süreçlerine ve endüstriyel bakım çeşitlerine dair bilgiler sunulmuştur. Etkin sonuçları ile ön plana çıkan kestirimci bakım detaylandırılmıştır ve kestirimci bakımın avantajlarına yer verilmiştir.

Dördüncü bölümde yapay zekâ uygulamaları ile ilgili bilgiler verilmiştir. Tezde kullanılan klasik makine öğrenmesi ve derin öğrenme algoritmalarının açıklamalarına yer verilmiştir.

Beşinci bölüm olan materyal ve metot bölümünde ise öncelikle çalışmada faydalanılan teknolojilerden kısaca bahsedilmiştir. Daha sonra bu çalışmada kapsamında ele alınan veri setleri ile ilgili bilgiler sunulmuştur. Dengesiz sınıf dağılımına sahip veri seti problemi ile başa çıkmada kullanılan yaklaşımlara ilişkin açıklamalar yapılmıştır.

Altıncı bölümde öncelikle modellerin değerlendirmesinde kullanılmış olan performans metriklerine ilişkin açıklamalar yapılıp hesaplama yöntemlerine değinilmiştir. Ardından önerilen klasik makine öğrenmesi ve derin öğrenme sınıflandırma modellerinin deneysel sonuçları ve performansları ortaya konulmuştur.

Son bölüm olan tartışma ve sonuçlar kısmında ise yapılan çalışmanın sonucu farklı perspektiflerden değerlendirilerek, gelecek çalışmalara yönelik önerilerde bulunulmuştur.

2. KAYNAK ARAŞTIRMASI

Kestirimci bakım çalışmaları hakkında yapılan literatür araştırmasında birçok farklı yaklaşım olduğu görülmüştür. İstatistiksel metotların yanında son yıllarda özellikle yapay zekâ destekli kestirimci bakım uygulamaları ile ilgili olan akademik çalışmalar oldukça fazladır. Bazı çalışmalarda problemlere makine öğrenmesi yöntemleri ile yaklaşım sergilenmişken bazılarında derin öğrenme yöntemleri tercih edilmiştir. Bir kısım çalışma ise, bu çalışmada olduğu gibi hem makine öğrenmesi hem de derin öğrenme yöntemleri ile gerçekleştirilmiştir ve performans karşılaştırılmaları yapılmıştır. İncelenen literatür çalışmaları burada 3 alt başlık halinde sunulmuştur.

2.1 Klasik Makine Öğrenmesi Yöntemleri ile Kestirimci Bakım Çalışmaları

Makine öğrenmesi yöntemleri kestirimci bakım için yaygın olarak kullanılmaktadır.
Ayvaz S. Ve Alpay K. yaptıkları çalışmada [6] sensörler ve algılayıcılar vasıtasıyla endüstriyel tesisten toplanan verilerden oluşmuş gerçek hayat verileri üzerinde çalışma yapılarak kestirimci bakım sistemi kurulmuştur. Yapılan çalışmada veri ön işleme adımlarından sonra öznitelik seçimi kısmında boyut azaltma işlemi için temel bileşenler analizi (PCA) yöntemi kullanılarak, değişkenlerin önem sırasının belirlenmiştir. Veri seti farklı makine öğrenmesi algoritmalarıyla eğitilerek başarım oranları test edilmiştir ve rassal orman (RF) algoritmasının en iyi sonucu veren algoritma olduğu görülmüştür. Bu performansa en yakın diğer algoritma ise XGBoost algoritmasıdır [6].
NASA'nın yayınlamış olduğu Turbo Fan Motor veri seti üzerinde bir başka çalışma gerçekleştirilmiştir. Veri seti; tanımlama numarası, zaman bilgisi, 3 farklı ayar kademesi ve 21 farklı sensör ölçüm verilerini içermektedir. Yapılan çalışmada kalan faydalı ömür (RUL) tahminlemesi yapılmıştır. Bu tahminleme için lineer regresyon, karar ağaçları, destek vektör makineleri (SVM), rassal orman (RF) algoritmaları başta olmak üzere birçok algoritma denenmiştir. Tahmin edilen ve gerçek RUL değerleri karşılaştırıldığında ortalama hata kareleri kökü (RMSE) metriği açısından en iyi sonucu veren algoritmanın rassal orman algoritması olduğu görülmüştür [7].

Rüzgâr tribününden toplanan vibrasyon verileri üzerinde gerçekleştirilen bir başka çalışmada rulmanlara ilişkin oluşan hata tipini tahminleme konusuna odaklanılmıştır. Elde edilen vibrasyon sinyalleri kullanılarak SVM, K-En yakın komşu (KNN) ve K ortalama (K-Means) algoritmaları ile sınıflandırma performansı ölçülmüş. Yapılan performans karşılaştırmasında en yüksek doğruluk oranı %87 ile KNN modelinde görülmüştür. Bunun haricinde bir ileri aşama olarak iş birliğine dayalı tavsiye yaklaşımı (CRA) ile nihai doğruluk oranının %93'lere kadar çıktığı görülmüştür. İşbirliğine dayalı tavsiye yaklaşımı ile her bir algoritmaya belli ağırlıklandırmalar verilerek ortak bir yaklaşımla daha yüksek doğrulukta tahminleme yapılması amaçlanmaktadır [8].

Bir diğer gerçek hayat uygulaması ise kimya sektöründe faaliyet gösteren bir fabrikadaki 30 farklı endüstriyel pompadan 2,5 yıl boyunca toplanan vibrasyon verileri üzerinden gerçekleştirilmiştir. Yapılan çalışmada rassal orman algoritması kullanılarak 7 güne kadarki titreşim kaynaklı hataların tespiti yapılmıştır [9].

Otomobil dişli kutusu arıza tanısı için yapılan bir çalışmada vibrasyon verilerinden yararlanılmıştır. Her bir dişli kutusu iki farklı hız ve yük şartlarında test edilmiştir. Yapılan çalışmada iyi durumdaki ve hatalı durumdaki dişli kutularından toplanan vibrasyon verileri SVM algoritması ile eğitilerek arıza teşhisi çalışması gerçekleştirilmiştir. Sonuç olarak 4 farklı dişli durumu için sırasıyla %96,62 , % 92,375 , %98,75 ve %100'lük doğru sınıflandırma elde edilerek tasarlanan çalışmada SVM yönteminin arıza teşhisinde verimli bir şekilde kullanılabileceği ortaya konulmuştur [10].

Bir gerçek hayat uygulamasının ele alındığı başka çalışmada döküm fabrikasından 6 ay boyunca toplanan veriler ile kalan faydalı ömür tahmini konusunda çalışma yapılmıştır. Tahmin için rassal orman algoritması ve temel bileşenler analizi (PCA) yöntemi kullanılmıştır. Rassal orman algoritması ile elde edilen en yüksek doğruluk oranı %85,17 iken, temel bileşenler analizi ise %73,4 oranında anomali tespiti yapılabilmiştir [11].

Soylu B. ve arkadaşları tarafından, yatak üretim tesisindeki paketleme makinesinden elde edilen geçmiş arıza verileri ve anlık sensör ölçüm verileri baz alınarak bir bakım karar destek sistemi uygulaması gerçekleştirilmiştir. Karar ağaçları ve destek vektör makine algoritmalarının ele alındığı çalışmada öncelikle veri seti sırasıyla %75 ve %25 olmak üzere eğitim ve test verisi olarak bölünmüştür. Dengesiz bir sınıf dağılımına sahip bu veri seti ile 5 katlı çapraz doğrulama yöntemi uygulanarak model

performansları test edilmiştir ve karmaşıklık matrisi açısından karşılaştırılmıştır. Karar ağaçları algoritması arıza olan ve arıza olmayan durumların tamamını neredeyse doğru tahmin ederek yüksek bir doğruluk tahmini ortaya koymuştur. Destek vektör makinesi algoritmasında ise arıza olmama durumlarının doğru tahmin edilmesinde sıkıntı yaşanmazken, arıza durumlarının büyük çoğunluğu yanlış tahmin edilmiştir [12].

Otomotiv endüstrisinde yapılan bir gerçek hayat uygulamasında kullanılan transfer pres makinesi için bakım ihtiyacı zamanının tahmin edilmesi ve yaklaşan anormal durumların tespitine yönelik kestirimci bakım çalışması ortaya konulmuştur. Transfer pres makinesine yerleştirilen sensörler vasıtasıyla 2 ayrı motor için toplanan veriler üzerinde farklı makine öğrenmesi algoritmalarının performansları karşılaştırılmıştır. Denenmiş olan KNN, Naive Bayes, SVM, rassal orman ve lojistik regresyon algoritmaları arasından, rassal orman algoritmasının her iki motor için de en yüksek doğruluk performansına sahip algoritma olduğu ortaya konulmuştur. Rassal orman algoritması ile hidrolik pompa motorunda %99,5 ve yağlama motorunda ise %99,9'luk doğru sınıflandırma performansı gerçekleşmiştir [13].

2.2 Derin Öğrenme Yöntemleri ile Kestirimci Bakım Çalışmaları

Literatürde kestirimci bakım problemlerine derin öğrenme yaklaşımları ile çözüm üretilen birçok çalışma mevcuttur.

Bir çalışmada, NASA'nın 7 adet özniteliğe sahip IMS rulman veri setinden faydalanılarak, uzun ömürlü kısa dönem bellek (LSTM) tabanlı kestirimci bakım modeli oluşturulmuştur. Oluşturulan model için genetik algoritmalar kullanılarak hiper parametreler optimize edilmiş ve deneysel sonuçlar karşılaştırılmıştır. Her bir öznitelik ayrı ayrı değerlendirilerek ve çeşitli kombinasyonlarda öznitelik vektörleri oluşturularak en uygun öznitelik ve öznitelik vektörlerinin performansları doğruluk metriği bakımından değerlendirilmiştir. Yığılma, çarpıklık ve tepe faktörü (creast factor) kombinasyonlarından oluşan öznitelik vektörü ile en yüksek doğruluk oranı %92,34 olarak elde edilmiştir. Daha sonra genetik algoritmalar ile hiper parametre optimizasyonu neticesinden doğruluk performansı % 98,14'e çıkarılmıştır. En başarılı LSTM modeli 8 zaman adımlı, 3 katmanlı, 115 gizli sinir hücreli yapıdan oluşmuştur [14].

21 adet sensör verisi içeren uçak motoru veri seti ile yapılan diğer bir çalışmada ise, sırasıyla %70 ve %30 oranlarında eğitim ve test verisi olarak bölünmüştür. Bu çalışmada konvolüsyonel sinir ağları (CNN), uzun ömürlü kısa dönem bellek (LSTM) ve bu derin öğrenme algoritmalarının kombinasyonu olacak şekilde (CNN + LSTM) toplam 3 farklı yaklaşımla sonuçlar elde edilmiştir. Önerilen CNN ve LSTM kombinasyonuna sahip hibrit model %99,60 ile en yüksek doğruluk oranına sahip performansı göstermiştir [15].

Bir önceki çalışmaya benzer bir şekilde yapılan diğer çalışmada ise CNN ile LSTM'in kombinasyonuyla oluşturulan hibrit model ile klasik LSTM modelinin karşılaştırılması yapılmıştır. Hibrit derin öğrenme modelinin doğruluk performansı klasik CNN modelinden %4,44 daha yüksek çıkmıştır. Bu çalışmada yazarlar, makine öğrenmesi ve derin öğrenme yaklaşımıyla yapılan diğer çalışmalar ile kendi çalışma sonuçlarını kıyasladığında önerdikleri hibrit derin öğrenme yapısının %97,48 ile en yüksek doğruluk oranına sahip olduğu görülmüştür [16].

Gerçek bir rüzgâr türbininin çalışma koşullarını taklit edecek biçimde tasarlanmış olan bir tezgâh üstü deney sisteminde Biswal S. ve Sabareesh G.R. tarafından çalışma yapılmıştır. Sağlıklı bir dişli ve rulman; diş dibi çatlağı ve sağlıklı rulman, sağlıklı dişli ve eksenel çatlağa sahip rulman olmak üzere 3 farklı koşul için deneyler gerçekleştirilmiştir. Kurulan deney düzeneğinde farklı senaryolar için elde edilen vibrasyon verilerinden yola çıkılarak kritik bileşenler üzerinde oluşmaya başlayan arızaların erken safhada teşhis edilebilmesi için yapay sinir ağı modeli önerilmiştir. Önerilen model ile eğitilen verilerde %92,6 doğruluk performansıyla sağlıklı durum ile arızalı durum sınıflandırması yapılmıştır [17].

Motora ait sensör verilerini içeren ve NASA tarafından yayınlanmış olan bir veri seti üzerindeki bir çalışmada LSTM ağları ile motor durum tahminine yönelik durum incelenmiştir. Kullanılan veri setinde öncelikle varyansı sıfır olan ve birbirleri arasında korelasyonu yüksek olan öznitelikler veri setinden çıkartılmış ve geri kalan öznitelikler ile çalışma yapılmıştır. Önerilen 2 katmanlı LSTM yapısında15 adet girişe karşılık motor durumunu belirten 4 farklı etiket (sağlıklı, dikkat, onarım, arıza durumu) mevcuttur. Önerilen LSTM ağı yapısı ile kalan faydalı ömür tahmini %85 doğruluk oranında performans göstermiştir [18].

2.3 Klasik Makine Öğrenmesi ve Derin Öğrenme Yöntemlerinin Karşılaştırıldığı Kestirimci Bakım Çalışmaları

Makine öğrenmesi yöntemleri ve derin öğrenme yöntemleri çalışılan veri setlerinin yapısına, veri setindeki gözlem miktarına, özniteliklerin sayısına ve karakteristiğine bağlı olarak performans açısından birbirlerine göre üstünlük sağladıkları durumlar olabilir. Bu tez çalışmasında da olduğu gibi literatürde bazı çalışmalar ise makine öğrenmesi ve derin öğrenme yöntemlerini aynı çalışma içerisinde ele alıp performans karşılaştırması yapmışlardır.

Microsoft tarafından yayınlanmış olan bir veri seti üzerinde yapay zekâ algoritmaları kullanılarak belirlenen zaman dilimi içerisinde hangi tip arızanın çıkacağının tahminlemesi yapılmıştır. Veri seti öncelikle KNN, karar ağaçları, rassal orman, Naive Bayes ve yapay sinir ağları algoritmaları ile test edilmiş, en iyi 2 sonucu veren rassal orman ve yapay sinir ağları (ANN) üzerinde detaylı çalışma yapılmıştır. Veri seti eğitim, test ve doğrulama olmak üzere 3'e bölünmüştür ve modellerin oluşturulup test edilmesi bu şekilde yapılmıştır. Veri seti, dengesiz bir veri seti olduğu için performans değerlendirmesinde doğruluk metriğinden ziyade, duyarlıklık, kesinlik ve F1 skor metrikleri kullanılmıştır. Arıza olmama durumu ile birlikte 4 farklı komponentteki arıza olma durumlarının sınıflandırma tahmini bu metrikler ile ortaya konulmuş ve tatmin edici sonuçlar elde edildiği görülmüştür [19].

Bir önceki çalışmada [19] kullanılan veri setini ele alan başka bir çalışma daha vardır. Bu çalışmada 8 farklı makine öğrenmesi ve derin öğrenme algoritması denenip model performansları karşılaştırılmıştır. Optimizasyon çalışmaları neticesinde Gradient Boosting, rassal orman ve ANN modellerinin en iyi performans sonuçları verdiği gözlemlenmiştir [20].

Yapılan başka bir çalışmada, fan ve ölçüm ekipmanları ile oluşturulmuş olan deneysel ortamdan alınan vibrasyon verileri işlenmiştir. Vibrasyon ölçümleri üzerinde öncelikle Hızlı Fourier Dönüşümü (FFT) yapılarak veri seti yeniden yapılandırılmıştır. Daha sonra K-Katlamalı Çapraz Doğrulama ile ANN modellemesi yapılarak modelin başarımı RMSE metriği ile ortaya konulmuştur. Daha sonra regresyon ağaçları, rassal orman ve karar destek makineleri yöntemleri ile karşılaştırılmıştır. ANN modelinin diğer klasik makine öğrenmesi yöntemlerine oranla daha yüksek performanslı sonuç verdiği görülmüştür [21].

Diğer bir çalışma rulman arızası veri seti üzerinde gerçekleştirilmiştir. LSTM tabanlı özyinelemeli sinir ağları (RNN) ile klasik makine öğrenmesi yöntemlerinden SVM modelinin başarım oranları karşılaştırılmıştır. Yapılan çalışmada kullanılan veri seti 28 adet özniteliğe sahiptir. Veri setindeki etiketler ise Normal I, Normal II, Uyarı ve Hata olarak 4 farklı kategoride işaretlenmiştir. Ortaya konulan LSTM tabanlı RNN modelinin SVM yöntemine göre daha üstün performans verdiği görülmüştür [22].

Bir diğer çalışmada motor rulman verileri üzerinde LSTM ve destek vektör regresyon makinesi (SVRM) yöntemleri ile mekaniksel durum tahmini çalışması gerçekleştirilmiştir ve model sonuçları ortalama kare hata (MSE) metriği açısından karşılaştırılmıştır. Farklı zaman çerçeveleri baz alınarak (1, 3 ve 10) performansları test edilen modellerden, LSTM modelinin aynı zaman çerçeveleri dahilinde SVRM modeline göre daha başarılı performans ortaya koyduğu görülmüştür [23].

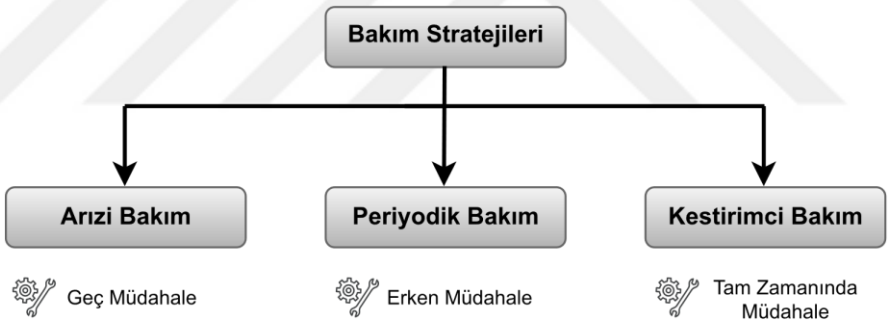
Derin otomatik kodlayıcı ve derin sinir ağları (DNN) kullanılarak rulmanlar üzerinde kalan faydalı ömür tahmini için yapılan bir çalışmada derin öğrenme tabanlı model önerilmiştir. RUL tahmini için kullanılan veri setinde vibrasyon verileri yer almaktadır. İlgili vibrasyon verilerinden, zaman alanı özellikleri, frekans alanı özellikleri ve zaman-frekans alanı özellikleri dahil olmak üzere farklı özellikler çıkarılmıştır. Tahmin modeli olarak otomatik kodlayıcılı derin sinir ağları, otomatik kodlayıcısız derin sinir ağları, frekans spektrum bölümü toplamı (FSPS) ile derin sinir ağlarının yanında klasik SVM yöntemleri karşılaştırılmıştır ve model performansları ortalama kare hata (MSE) metriği açısından incelenmiştir. En iyi sonucu 0.200 MSE değeri ile otomatik kodlayıcılı derin sinir ağlarının verdiği görülmüştür. Bu çalışmada en yetersiz kalan modelin ise 0.347 MSE değeri ile SVM algoritmasının olduğu görülmüştür [24].

Başka bir çalışmada ise, CNC freze tezgâhı üzerinde yapılan bir sensör entegrasyonu ile makineden vibrasyon verileri toplanmıştır. Yıpranmış ve tamamen yeni takımlar ile yapılan deneylerden elde edilen veriler ANN, SVM ve KNN sınıflandırıcıları ile eğitilerek, sonrasında eğitim seti dışında yer alan veriler ile test edilmiştir. ANN modelinin 0,9444 doğruluk değerinin yanında, duyarlılık ve kesinlik parametreleri açısından da en yüksek performansı gösterdiği ortaya konulmuştur [25].

3. ENDÜSTRİYEL BAKIM STRATEJİLERİ

Şirketlerin hizmet ettikleri sektörler, mevcut varlıkları ve uçtan uca üretim süreçleri farklılık gösterdiğinden dolayı bakım yönetim ihtiyaçları da birbirinden farklıdır [26]. Bakım planlamalarında ve faaliyetlerinde her ne kadar ortak amaç üretim sürekliliğini sağlamak, verimliliği artırmak gibi benzer olsa da tercih edilen bakım stratejileri şirketlerin kendi öz değerlendirmeleri neticesinde belirlenir. Her bakım stratejisinin bazı avantajları ve dezavantajları vardır. Dolayısıyla tüm işletmelere uyan tek bir bakım modeli söz konusu değildir.

Şekil 3.1'de görüldüğü üzere endüstriyel bakım stratejileri Arızi Bakım, Periyodik Bakım ve Kestirimci Bakım olarak 3 sınıfa ayrılmaktadır [27]. İlerleyen kısımlarda her bir bakım stratejisi ayrıca açıklanacaktır.



Şekil 3.1 : Bakım stratejilerinin sınıflandırılması.

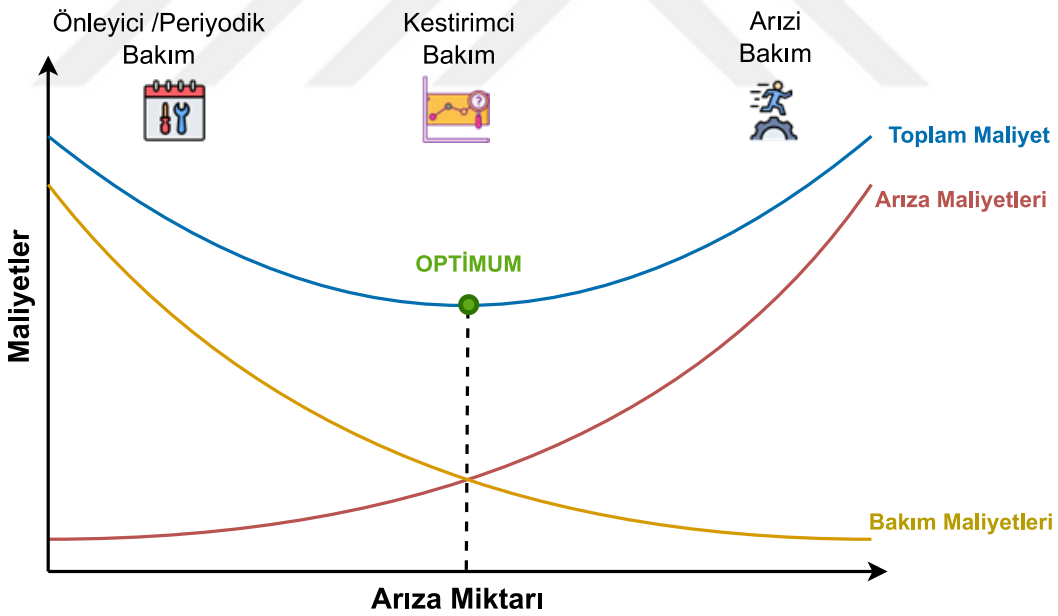
Şirketler tarafından tercih edilen bakım stratejileri çoğu zaman şirket içerisindeki varlıkların kritiklik duruma göre farklılık gösterir ve şirket genelinde optimum bakım maliyetleri açısından karma yöntemler kullanılabilir. Örneğin; bir şirket kritiklik seviyesi yüksek ekipmanları için kestirimci bakım yöntemini tercih ederken, kritiklik seviyesi görece daha az olan ekipmanları için periyodik bakımı tercih edebilir.

Makinelerin kritiklik seviyelerine göre sınıflandırılması, makinelerin prosese ve işletme maliyetine olan etkisine göre çok yönlü analizler vasıtasıyla yapılır. Makine sınıflandırma kriterleri fabrikaların dinamiklerine göre değişir. Örneğin;

- Harcanan iş gücü (toplam bakım süresi)

- Bakım maliyetleri
- Stok maliyetine etkisi
- Diğer proseslerin devamlılığını etkileme oranı
- Toplam ekipman verimliliği hedefi
- Makinenin hassasiyeti
- İş güvenliği riskleri (patlama, yangın vb.)
- Müşteri dinamikleri

Bu kriterler, doğru bir bakım stratejisinin oluşturulması için detaylı bir şekilde irdelenmelidir. Aşırı ve gereksiz bakım yapmak bakım maliyetlerini artırırken, eksik ve gereğinden az bakım yapmak da arızi duruşların maliyetlerinin artmasına ve bunun yanında iş güvenliği, ürün kalitesi gibi çeşitli risklerin orta çıkmasına neden olur. Bakım stratejilerini şekillendirmenin odağında Şekil 3.2'de de görüldüğü gibi toplam optimum bakım maliyet dengesini bulmak vardır. Bakım yönetimi; personel yönetimi, varlık yönetimi ve bir bakım bütçesi yönetimini kapsayan topyekûn bir disiplini gerekli kılar.



Şekil 3.2 : Optimum bakım maliyet eğrisi [28].

3.1 Arızi Bakım

Arızi bakım, bozulan bir sistemin tekrar standart çalışma koşullarına döndürülmesi için yapılan düzeltici faaliyetlerin tümünü kapsayan bir yaklaşımdır. Bu bakım

yöntemi, makinenin bozulana kadar çalıştırılması mantığına dayanan en ilkel bakım yöntemidir. Arızi bakım uygulamalarında ekipman arızası gerçekleşene kadar herhangi bir kaynak ayrılmaz [29]. Ancak ne zaman gerçekleşeceği belli olmayan bu tip arızi duruşlar, uzun süreli üretim duruşlarını ve kayıplarını beraberinde getirir. Ayrıca plansız oluşan bu tip arızanın makinedeki başka ekipmanlara da zarar verme ve onları da çalışamaz duruma getirme riski bulunmaktadır. Yüksek miktarda malzeme stoğuna ihtiyaç duyulur. Bu tür gelişen ani bakım faaliyetleri için bakım ekibinin teknik yetkinliklerinin yüksek olması beklenir. Her ne kadar arıza oluşumu öncesinde herhangi bir finansal kaynak ihtiyacı duymasa da arıza sonucu oluşan maliyetler göz önüne alındığında en yüksek maliyetli bakım türüdür. Sadece arıza sayısının az, emniyet risklerinin bulunmadığı ya da yedek malzeme stoğuna kolay ulaşılabilen kritik olmayan durumlarda uygulanması uygundur [30].

3.2 Periyodik Bakım

Periyodik bakım, planlı bir bakım türü olup, arıza oluşmadan önce gerekli tedbirlerin alınarak arızaya sebep olabilecek olan ekipmanların bakımlarının yapılması esasına dayanır [31]. Önleyici bakım olarak da ifade edilebilmektedir. Periyodik bakım günümüzde endüstriyel ortamlarda en yaygın olarak tercih edilen bakım yöntemlerinden biridir. Makine üreticilerinden ya da uzman görüşlerinden elde edilen bilgiler doğrultusunda bakım faaliyetleri bir bakım planına yerleştirilir ve uygulanır. Bu bakım planı takvimsel bir periyot olabileceği gibi, makine çalışma döngüsü de olabilir. Örneğin; 6 ayda bir değiştirilmesi gereken bir filtre ya da 10.000 çalışma döngüsünde bir yağlanması gereken mekanizma gibi.

Periyodik bakımlar planlı olduğu için hangi noktaya ne zaman ve ne şekilde müdahale edileceği önceden belirlenmiştir. Bu durum bakım ekiplerinin, faaliyetlerini daha organize bir şekilde gerçekleştirilmesine imkân sağlar. Periyodik bakım yaklaşımının sağladığı en büyük avantajlardan biri plansız duruş maliyetlerini minimize etmesidir. Ayrıca belli periyodlarda yapılan rutin bakımlar ile ekipmanların yaşam süresi uzatılmış ve daha büyük ölçekli olası arızaların önüne geçilmiş olur.

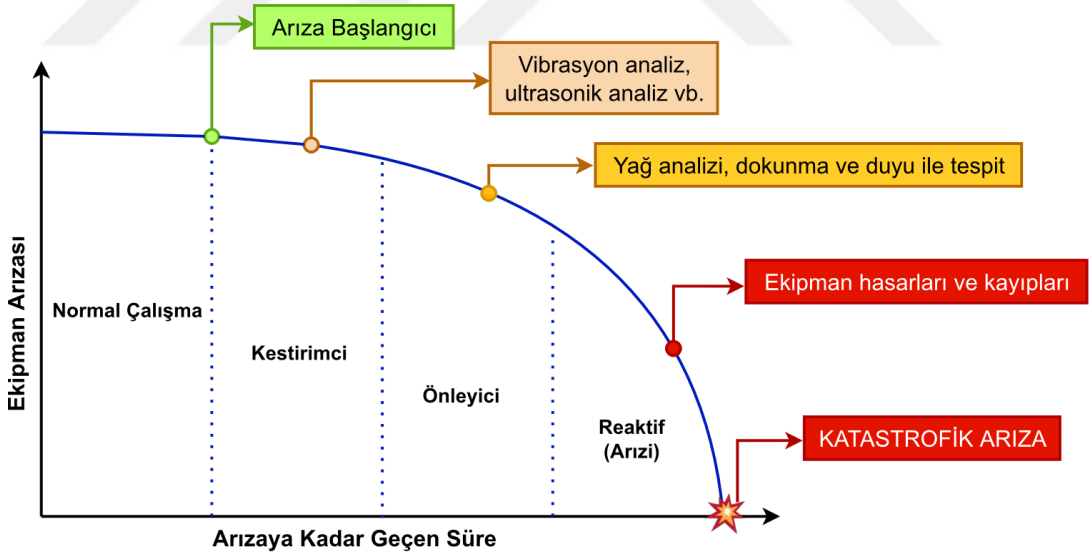
Periyodik bakım yönteminin uygulandığı bir işletmede optimum ekonomik fayda sağlamak için oluşturulacak bakım periyodunun doğru belirlenmesi kritik öneme sahiptir. İyi planlanamayan bir periyodik bakım, bazen henüz kondisyon olarak bakıma ihtiyaç duymayan ekipmanlara bakım yapılmasına neden olur. Her türlü

gereksiz bakım faaliyeti, bakım maliyetlerini arttırır ve ekipman yaşam süresinden yeterince faydalanamamayla sonuçlanabilir. Bir diğer durumda ise üretim ortamı dinamiklerindeki fark edilemeyen bir anormallikten dolayı ekipmanın beklenenden daha önce arıza yapması durumuyla karşılaşılabilir. Bu tür durumlarda arızi bakım faaliyetleri uygulanır.

3.3 Kestirimci Bakım

Kestirimci bakım, sistemlerden toplanan verilere dayanarak, potansiyel arızaları, plansız duruşa sebebiyet vermeden önce istatistiksel metotlar ile ya da çeşitli yapay zekâ algoritmaları ile tahmin ederek zamanında gerekli müdahaleyi yapmayı öngören durum bazlı bir bakım metodudur [31].

Şekil 3.3'te görüldüğü üzere kestirimci bakım ile potansiyel arızaya ilk oluşum evrelerinden itibaren teşhis konulabilmektedir. Ekipman durumu izlenerek, arızanın oluşma seyrine girmesi günler, hatta haftalar öncesinden tahmin edilebilir ve buna göre de plansız duruş gerçekleşmeden üretim faaliyetlerini en az etkileyecek uygun bir plan dahilinde gerekli müdahale yapılır.



Şekil 3.3 : Kestirimci bakım yöntemi ile erken teşhis [29].

Kestirimci bakım ile, periyodik bakımın en önemli dezavantajından biri olan gereğinden fazla bakım yapma ve ekipmanın yaşam döngüsünden maksimum seviyede faydalanamama sorununun önüne geçilmiş olur. Dolayısı ile kestirimci bakım, diğer bakım metotları ile kıyaslandığında Şekil 3.1'de de görüldüğü üzere "tam zamanında bakım" olarak nitelendirilebilir. Ekipmanlara yapılması gereken

bakım aralıkları elde edilen verilerden çıkarılan öngörücü bilgiler doğrultusunda tayin edilir. Bu sayede bakım ihtiyacı duyulan makine veya ekipman potansiyel bir arızadan hemen önce durdurularak gerekli müdahale yapılır. Hatta bazı durumlarda duruşa gerek kalmadan yapılan ayar ve müdahaleler ile olumsuz durum bertaraf edilebilir. Bu avantajları sayesinde kestirimci bakım duruş maliyetlerinin önemli seviyede azaltılmasına ve üretim verimliliğinin artmasına yardımcı olur.

Bunun yanından kestirimci bakım uygulamalarında yoğun miktarda verinin toplanması, depolanması, işlenmesi için yüksek maliyetli ekipman ve altyapı yatırımlarına gereksinim duyulur. Ayrıca bu yöntem, süreçte görev alacak olan yüksek yetkinlikte personellerinin eğitimini ve istihdamını gerekli kılar. Diğer bakım yöntemlerine kıyasla görece yüksek ön maliyetler sebebiyle kısa vadeli düşünceye sahip firmalar ve yöneticileri tarafından bu bakım yöntemi göz ardı edilebilir ya da kısıtlı ve geleneksel yöntemlerle idame ettirilmeye çalışılır [30]. Kestirimci bakım olgunluk seviyesini yukarıya taşıma hedefindeki firmalar daha yüksek harcamalar neticesinde kestirimci bakım uygulamalarını daha etkin hale getirebilirler.

3.3.1 Kestirimci bakım olgunluk seviyesi

Kestirimci bakımın özünde “veri” bulunmaktadır. Dolayısıyla kestirimci bakımı, veriye dayalı bir bakım modeli olarak adlandırabiliriz. Geçmişten günümüze kestirimci bakım için kullanılan yöntemler nesnelerin interneti (IoT) teknolojilerinin ve cihazlarının gelişmesiyle daha etkin hale gelmiştir. En eski kestirimci bakım yöntemlerinden olan yağ ve parçacık analizi, ultrasonik test, termal görüntüleme gibi yöntemlerde analiz edilen veriler günümüzde IoT cihazlarından toplanan gerçek zamanlı verilere oranla daha basit ve anlaşılabilir bir yapıya sahiptir ve işlenmesi daha kolaydır [32]. Ancak günümüzde dijital fiziksel sistemlerin gelişimi ve endüstriyel ortamlara entegrasyonu sayesinde, geleneksel yöntemler ile başlayan kestirimci bakım uygulamaları olgunlaşarak daha yüksek kabiliyetler kazandı.

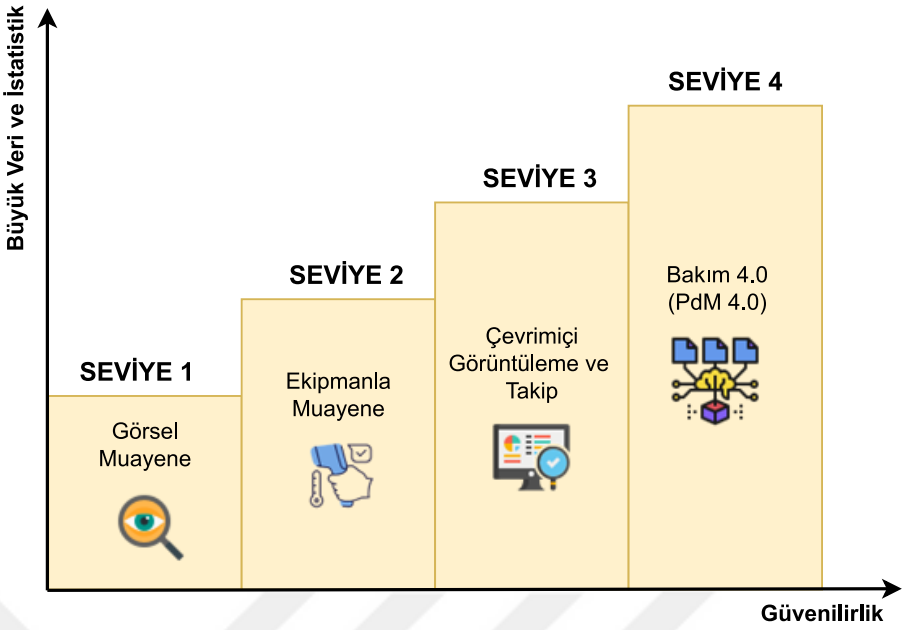
Kestirimci gelişim aşamaları Şekil 3.4’te de görüldüğü üzere 4 seviyeli bir olgunluk matrisi üzerinde ifade edilebilir.

Seviye 1: Süreç daha çok periyodik denetleme ve kontrol listeleri üzerinden ilerler. Bu aşamada genellikle görsel kontroller yapılır. Gözlenen veriler genelde kâğıt üzerinde tutulur ve kâğıt dokümanlar üzerinden trend takibi yapılır. Organizasyonda ise deneyimli ustaların tecrübelerinden faydalanılır.

Seviye 2: Periyodik denetlemeler cihazlar vasıtasıyla yapılır. Farklı zamanlarda sistemler üzerinden ölçümler alınır. Gözlem sonuçları dijital olarak basit yazılımlar ile tutulur. Organizasyon olarak eğitimli denetleyicilerden faydalanılır.

Seviye 3: Sürekli uzaktan denetleme ve sensörler vasıtasıyla ölçümler yapılır. Dijital ölçüm verileri çeşitli durum izleme ve analiz yazılımları ile incelenir ve trend takibi yapılır. Durum veri tabanı oluşturulmuştur. Organizasyonda daha çok bakım ve güvenilirlik mühendisleri vardır.

Seviye 4: Sürekli elde edilen sensör ölçümlerinin yanından birçok farklı tesis verisi bir yerde toplanır. Dijital veri geçmişi ve depolaması gelişmiştir. Analizler; durum izleme yazılımları, büyük veri platformları ve gelişmiş karar destek yazılımları vasıtasıyla yapılır. Organizasyonda güvenilirlik mühendisleri ve veri bilimciler bulunur. PdM4.0 olarak ifade edilir [33].



İçerik (Veri Kaynağı)	• Periyodik fiziksel muayene • Kontrol listeleri • Kağıt dökümantasyon	• Periyodik fiziksel muayene • Enstrümanlar • Dijital kayıtlar	• Uzaktan sürekli muayene • Sensörler • Dijital kayıtlar	• Uzaktan sürekli muayene • Sensörler ve diğer veriler • Dijital kayıtlar
Veri Toplama ve İşleme	• Kağıt tabanlı veri • Çoklu muayene noktası	• Dijital durum kayıtları • Tekli muayene noktası	• Dijital durum kayıtları • Çoklu muayene noktası	• Dijital durum kayıtları • Çoklu muayene noktası • Dijital çevresel veri • Dijital bakım geçmişi
Teknolojiler	• MS Excell / MS Access vb.	• Gömülü enstrüman yazılımları	• Durum takip yazılımı • Durum veri tabanı	• Durum takip yazılımı • Büyük veri platformları • İstatistiksel yazılımlar

Şekil 3.4 : Kestirimci bakım olgunluk matrisi [33].

3.3.2 Yapay zekâ destekli kestirimci bakım

Kestirimci bakımın, gücünü veriden aldığını belirtmiştik. Kestirimci bakım uygulamalarında daha öncelerden uzman görüşlerinden, eğitimli personellerin tecrübelerinden ve veri yorumlama kabiliyetlerinden faydalanırken, günümüzde yapay zekanın alt dalları olan makine öğrenmesi ve derin öğrenme gibi metotlar kullanılarak isabet oranı yüksek öngörücü tahminlerle etkili kestirimci bakım uygulamaları yapılabilmektedir.

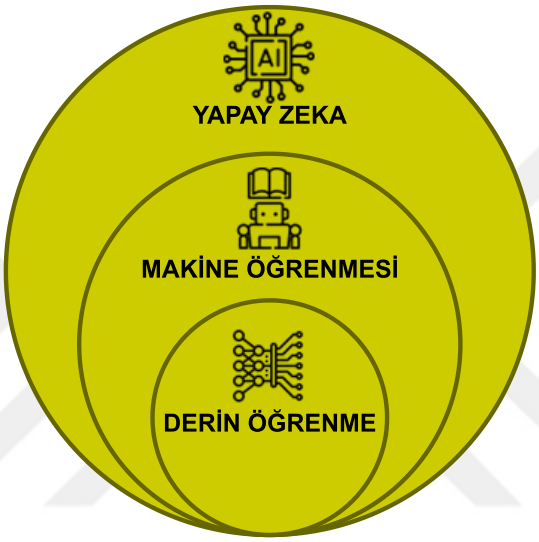
Yapay zekâ destekli kestirimci bakım modelleri 3 temel aşamada ele alınabilir [34]:

- Ekipmanlardan verileri toplanması ve uygun formatlara getirilmesi
- Arıza tespitine uygun özniteliklerin belirlenip seçilmesi
- Kestirimci bakım uygulanacak noktada bize en doğru ve etkili tahminleri verebilecek modelin oluşturulması

Bu 3 aşamanın her biri beraberinde çeşitli zorluklar getirir. Ekipmanlardan veri toplanması aşamasında eksik ya da hatalı sensör verileri, birden fazla tekrarlayan veriler, alakasız veriler titizlikle yönetilmesi gereken hususlardır. Veriler problemin amacına uygun hale getirilirken, veri setinde kullanılacak olan özniteliklerin seçimi de son derece önemlidir. Gereksiz öznitelikler veri eğitiminde modeli daha karmaşık hale getirebilir ve modelin performansını düşürebilir. Sonuca etkisi büyük olan herhangi bir nitelik, veri seti içerisinde yer almıyorsa yine hedeflenen model performansının yakalanmasında güçlük çekilebilir.

4. MAKİNE ÖĞRENMESİ METODLARI

Yapay zekâ, makine öğrenmesi ve derin öğrenme kavramları sıklıkla birbirlerinin yerine kullanılan ve birbirleri ile karıştırılan kavramlardır. Bu kavramların kendi aralarındaki ilişkileri Şekil 4.1'deki gibi bir gösterimle açıklanabilir [35].



Şekil 4.1 : Yapay zekâ, makine öğrenmesi ve derin öğrenme ilişkisi.

Yapay zekâ, makine öğrenmesi ve derin öğrenmeyi kapsayan en genel terim olarak ifade edilir. Yapay zekâ, bir bilgisayarın ya da bilgisayar kontrollü makinenin, insanların muhakeme, akıl yürütme, anlam çıkartma ve geçmiş deneyimlerden öğrenme gibi bilişsel yeteneklerini ve işlevlerini taklit etme yeteneğidir [36].

Yapay zekâ günümüzde oldukça popüler hale gelmiş olsa da geçmişi sanılan aksine oldukça eskiye dayanmaktadır. Bu kavram, ilk kez 1950'lerde Alan Turing tarafından yayınlanan bir makalede makinelerin insanlar gibi düşünüp düşünemeyeceği fikri ortaya atılarak gündeme gelmiştir [37]. Daha sonra 1956 yılında ise yapay zekanın mimarlarından kabul edilen John McCarthy, bir konferansta yapay zekâyı "akıllı makineler üretim bilimi ve mühendisliği" olarak dile getirmiştir [38].

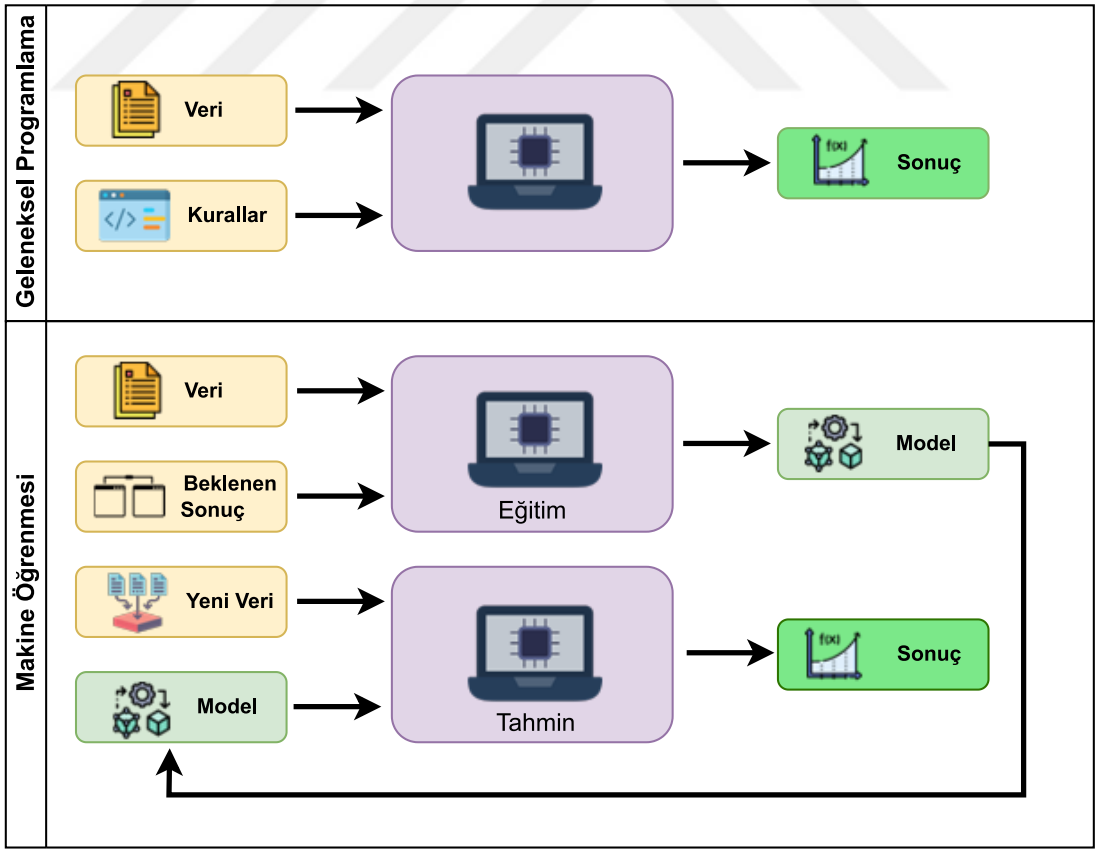
Geçmişten günümüze bu alanda yapılan birçok çalışma neticesinde, artık yapay zekâ uygulamaları günlük hayatımızın bir parçası olarak her alanda karşımıza

çıkmaktadır. Ulaşım teknolojilerinden sağlık alanındaki çalışmalara, askeri uygulamalardan mühendislik uygulamalarına, bankacılık ve finans hizmetlerinden siber güvenlik ve her türlü bilişim uygulamalarına kadar hayatımızın her alanına giren yapay zekâ uygulamaları ile otomatikleştirilen birçok süreç, karar destek uygulamaları ile insanlara oldukça fayda sağlamaya devam etmektedir.

İlerleyen bölümlerde yapay zekanın birer alt çalışma alanı olan klasik makine öğrenmesi ve derin öğrenme kavramları detaylıca açıklanmıştır.

Günümüzde, internet ortamından, endüstriyel ortamdaki algılayıcılardan, çeşitli veri tabanlarından ve birçok farklı veri kaynağından yüksek miktarlarda ham veriler elde edilmektedir. Makine öğrenmesi ise bu ham verilerin bir çıktı olarak bilgiye dönüştürülmesini sağlayan yapay zekâ dalıdır [39].

Makine öğrenmesini geleneksel programlamadan ayıran en önemli özelliği ise girdi olarak belirli kurallar bütününe odaklanmak yerine, geçmiş verilerden faydalanmasıdır. Geleneksel programlama ile makine öğrenmesi arasındaki işleyiş farkı Şekil 4.2'de görülebilir [40].



Şekil 4.2 : Geleneksel programlama ile makine öğrenmesinin şematik karşılaştırılması.

Makine öğrenmesi modelleri tarafından geçmiş verilerin kullanılıp, bu veriler arasında istatistiksel bağıntılar kurularak bir bilgi elde edilmesi, tıpkı bir insanın daha önceden deneyimlediği bir olayın sonucunda neyle karşılaşacağını tahmin etmesi gibidir. Deneyimleme ne kadar farklı varyasyonlarda ve çok sayıda gerçekleşirse çıktıyı da tahmin etme doğruluğu o kadar artacaktır.

Makine öğrenmesi ile geleneksel programlamayı kıyaslayacak olursak, makine öğrenmesinin avantajları şu şekilde ifade edilebilir:

- Geleneksel programlamada girdi olarak geniş bir kurallar bütünü gerekir. Bunu sağlamak makine öğrenmesine göre daha zordur.
- Geleneksel programlamada girdi verisinde bir değişim olursa, doğru bir çıktı için kurallar manuel olarak yeniden güncellenmelidir. Makine öğrenmesi ise verilerdeki değişikliği kendisi keşfeder.
- Makine öğrenmesi, görüntü, yazı ve ses gibi karmaşık problemlerde daha iyi sonuç verir [41].

Denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve pekiştirmeli öğrenme gibi farklı makine öğrenmesi yaklaşımları mevcuttur. Problem çözümünde kullanılacak olan öğrenme çeşidi ise kişinin tercihinden ziyade, problemde girdi olarak kullanılacak olan veri setine, problemin kapsamına ve problemin çözümünde ihtiyaç duyulan çıktı tipine göre belirlenir. Yani problemin kapsam ve amacı hangi yöntemin kullanılacağı konusunda belirleyici husustur. Bu tez çalışması kapsamında ele alınan kestirimci bakım probleminde, klasik makine öğrenmesi tarafında denetimli makine öğrenmesi yaklaşımı kullanılmıştır.

Bir veri seti, satırlardan ve sütunlardan oluşan bağımlı ve bağımsız değişkenleri içeren diziler şeklinde ifade edilebilir. Veri bilimi ve makine öğrenmesi terminolojisine göre her bir satır “gözlem” veya “örnek” olarak isimlendirilirken her bir sütun ise bir “öznitelik” olarak ifade edilir. Genellikle veri setinin son sütununda yer alarak özniteliklerin bağıntılarını ifade eden bağımlı değişkenler ise “etiket” veya sınıf olarak isimlendirilir. Her bir etiket, ilgili gözlem satırındaki özniteliklerin bağıntılarını temsil eder.

Denetimli öğrenme, bir makine öğrenmesi algoritmasına girdi olarak verilen özniteliklerin yanında etiket bilgisini de içeren veri setleri ile gerçekleştirilir. Yani

algoritma gelecekteki olayları tahmin etmek için, geçmişteki etiketli örnekler arasındaki bağıntıyı kurarak yeni verilere uygular ve hedef etiketi tahmin eder [42].

Sınıflandırma, bir denetimli öğrenme yaklaşımıdır. Veri kümesinde yer alan örneklere ait öznitelikler üzerinde kurulan bağıntıya göre, adından da anlaşılacağı üzere o örneğin hangi sınıfa dahil olduğunun tespiti ile ilgilenen denetimli öğrenme görevidir [43].

Sınıflandırma problemlerinde etiket bilgileri ayrık değerlerden oluşur. Yani etiket bilgileri kategorik verilerdir. Örneğin, bir elektronik postanın spam olup olmadığı, bir resimdeki hayvanın ne olduğunun tahmini, cinsiyet tahmini, bir hastalık olup olmadığının teşhisi vb. problemler sınıflandırma yöntemi ile yapılır.

Bu tez çalışmasında, kullanılan veri setleri üzerinde endüstriyel ortamdan elde edilen verilere dayanarak arıza oluşup oluşmadığının tespiti yapılmaktadır. Bu açıdan bakıldığında, çalışmada denetimli makine öğrenmesi yöntemi ile sınıflandırma görevi icra edilmiştir.

4.1 Temel Sınıflandırma Algoritmaları

Klasik makine öğrenmesi uygulamaları için geliştirilmiş çok sayıda özel sınıflandırma algoritması vardır. Bunların bir kısmı başlı başına kullanılabilen tekli sınıflandırıcılar iken, bazıları ise daha etkin bir sınıflandırma yapabilme adına aynı anda birden fazla sınıflandırma algoritmasından faydalanan kolektif öğrenme algoritmalarıdır.

Tez kapsamında ele alınan vaka çalışmalarında denetimli öğrenme yaklaşımıyla sınıflandırma görevi icra edildiği için, aşağıda sadece tezde kullanılmış olan sınıflandırma algoritmalarına yer verilmiştir. Fakat makine öğrenmesi algoritmaları bunlarla sınırlı değildir.

4.1.1 Tekil (Bağımsız) sınıflandırma algoritmaları

Çalışmada kullanılan tekli sınıflandırma algoritmaları karar ağaçları, lojistik regresyon, destek vektör makineleri ve K en yakın komşu sınıflandırıcılarıdır.

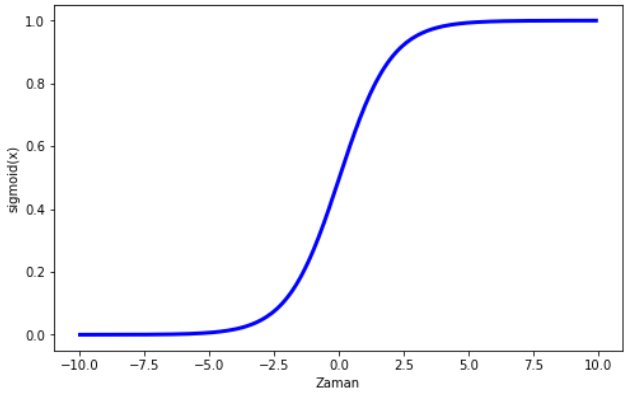
4.1.1.1 Karar ağaçları

Uzun yıllardır yaygın olarak kullanılan denetimli makine öğrenmesi algoritmalarından biridir. Karar ağaçları veri kümesi içerisinde yer alan gözlemleri, gözlemlerin özelliklerin değerleri üzerinde belirli karar kuralları oluşturarak etiket atamayı hedefleyen sınıflandırıcıdır. Karar ağaçları hiyerarşisinin en tepesinde yer alan hücreye kök düğüm denir. Belirlenen karar kuralları dizisine göre kök düğümden çocuğu olan iç düğümlere doğru uygulanan sınıf atamaları nihai olarak çocuksuz düğümlere (yapraklara) giden yolu izleyerek gerçekleştirilir. Karar ağaçları, sinir ağları ve destek vektör makineleri gibi diğer sınıflandırıcılara kıyasla daha yorumlanabilir yapıdadır. Çünkü verilerle ilgili sınıf ayrımlarını basit sorularla anlaşılır bir şekilde ortaya koyabilir. Ayrıca karar ağaçları; lojistik regresyon, destek vektör makineleri gibi tek bir karar sınırına dayanan sınıflandırıcılarla kolayca elde edilemeyen verilerin birçoğunu modelleyebilmek için yeterince esnek ve anlamlıdırlar. [44]

Bir karar ağacı modelinde düğümlerin sayısının artması modelin karmaşıklığını arttırır. Ezbere öğrenme durumu ile karşılaşılırsa, bu sorunu aşmak için ağaçta budama işlemine gidilebilir. Budama ise modelin tahmin doğruluğuna katkısı bulunmayan dalların karar ağacı yapısından çıkarılmasıdır.

4.1.1.2 Lojistik regresyon

Regresyon kelimesi sürekli verilerden oluşan bağımsız değişkenlerin tahmin edilmesini çağrıştırıyor olsa da, lojistik regresyon kategorik bağımlı değişkenlerden oluşan verilerin sınıflandırmaları için kullanılan yöntemlerinden biridir. Lojistik regresyon ile sınıflandırma yapılırken kategorik değişken ile bağımsız bir ya da birden çok değişken arasındaki ilişkinin ölçülmesi lojistik fonksiyonuna dayanır [45]. Lojistik fonksiyonu Şekil 4.3'te görüldüğü üzere "S" harfi şeklindedir ve Sigmoid eğrisi olarak tanımlanır.



Şekil 4.3 : Sigmoid eğrisi.

Sigmoid fonksiyonunun matematiksel ifadesi Denklem 4.1'de görüldüğü gibidir:

$$f(x)=\frac{1}{1+e^{-x}} \tag{4.1}$$

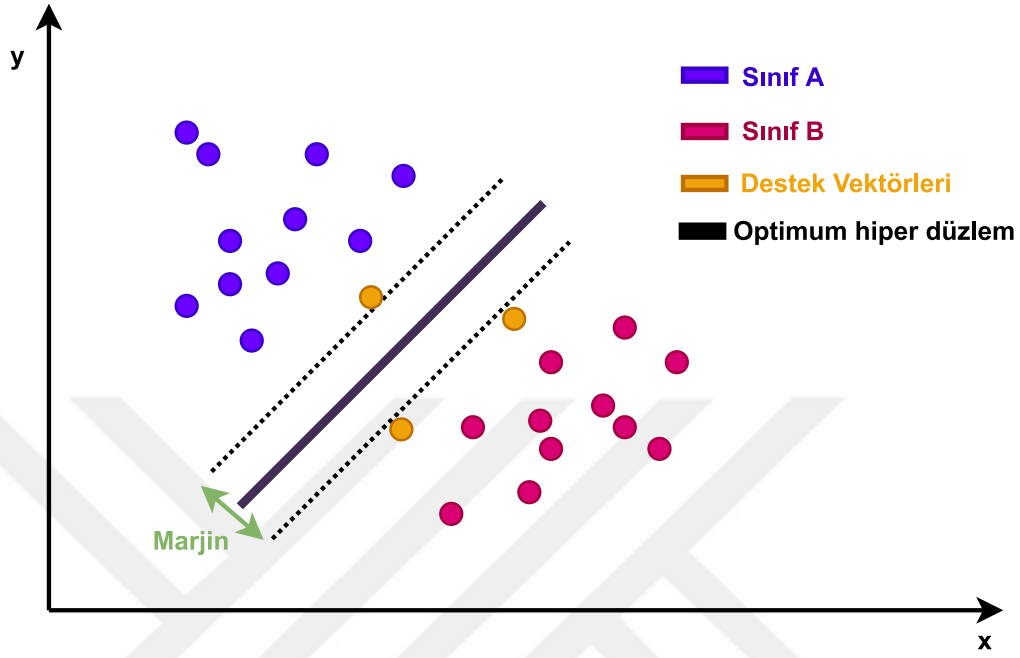
Sigmoid fonksiyonunda Şekil 4.4'te görüldüğü gibi x değerleri -∞ ile +∞ arasında değişmektedir. X eksenindeki değerler negatif sonsuza doğru gittikçe y fonksiyonu 0'a yakınsamaktadır; artı sonsuza doğru gittikçe ise y fonksiyonu 1'e yakınsamaktadır. Özetle -∞ ile +∞ arasındaki değerleri girdi olarak alan fonksiyon, çıkış olarak belirli bir eşik değeri yardımı ile bizlere kategorik sınıflandırma yapılmasına imkan tanıyan 0 veya 1 değeri türetebilmektedir [46].

4.1.1.3 Destek vektör makineleri

Destek vektör makineleri, sınıflandırmada kullanılan bir diğer denetimli makine öğrenmesi yöntemlerindendir. Destek vektör makinelerinin altında yatan fikir ise sınıflandırılmak istenen veriler arasındaki ayrımı en iyi ifade eden olan hiper düzlemi bulmaktır. Şekil 4.5'te görüldüğü üzere optimum hiper düzlem ise iki farklı sınıfa ait örneklemler arasındaki uzaklığı temsil eden marjinin maksimize edilmesi ile sağlanır. Bu sınıflandırma yönteminde marjin ne kadar geniş olursa, sınıfları birbirinden ayrıştırma işlemi o kadar iyi yapılır.

Ancak gerçek dünyadaki birçok veri Şekil 4.5'te görülen hiper düzlemle doğrusal bir şekilde kolaylıkla ayrılabilecek dağılıma sahip değildir. Bu gibi durumlarda "çekirdek hilesi (kernel trick)" denilen yönteme başvurulur. Çekirdek hilesi ile veri boyutuna fazladan boyut eklenerek veri uzayı doğrusal bir düzlem ile sınıflandırılabilir hale getirilir. Doğrusal, polinom, radyal temel işlevi ve sigmoid

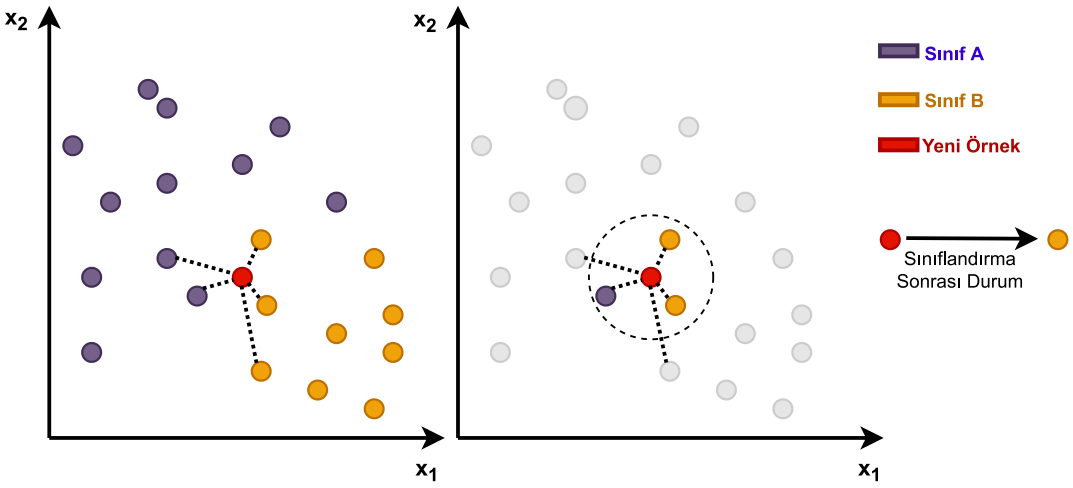
gibi çeşitli çekirdek işlevleri mevcuttur. Bu çalışmada oluşturulan SVM modelinde radyal temel işlevi (radial basis function – RBF) kullanılmıştır.



Şekil 4.4 : Hiper düzlem ve maksimum marjin.

4.1.1.4 K-En yakın komşu

K en yakın komşu algoritması bir veri setinde yer alan benzer gözlemlerin birbirine yakın olacağı varsayımı üzerine çalışan bir algoritmadır. Bu benzerlik ölçüt değeri ise Öklit, Manhattan, Minkowski gibi çeşitli uzaklık yöntemlerinden biri kullanılarak tayin edilir. Şekil 4.5'te görüleceği üzere sınıfı tayin edilmek istenen yeni bir gözlem noktasının, K adet en yakın komşunun ait oldukları sınıfa göre sıklık miktarı baz alınarak, ilgili gözlem için yeni sınıf belirlenmiştir [47].



Şekil 4.5 :K-En yakın komşu algoritması ile sınıflandırma [47].

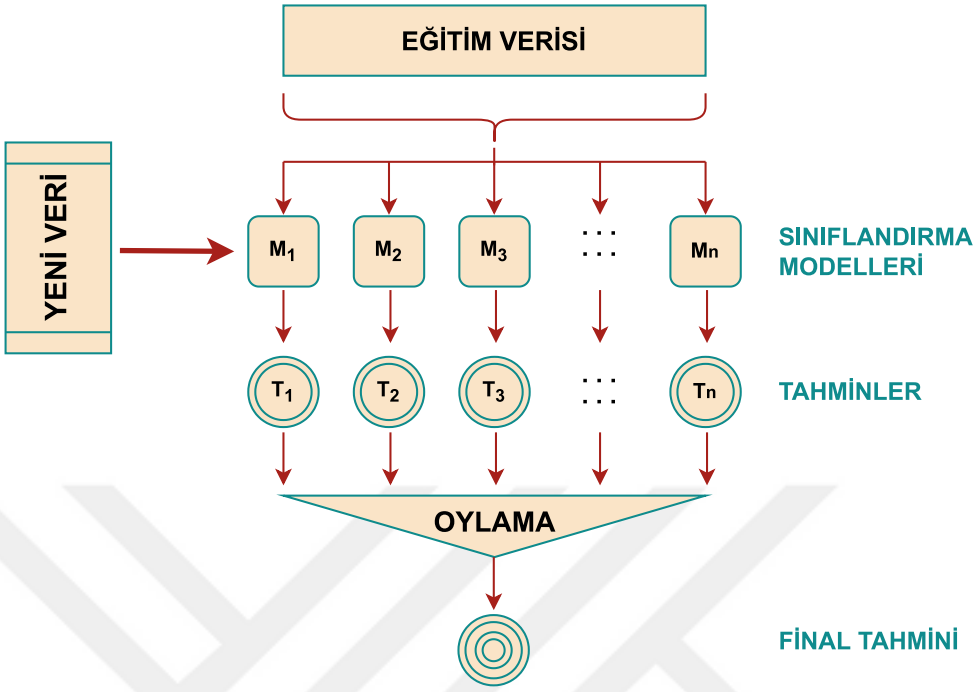
4.1.2 Kollektif öğrenme tabanlı sınıflandırma algoritmaları

İnsanlar gündelik hayatlarında karar vermesi gereken birçok durum ile karşılaşır ve çeşitli değerlendirme ölçütlerine göre kararlar verir. Kararlar neticesinde elde edilecek olan çıktı maliyetleri kritik ise durumu daha dikkatli bir şekilde irdelemek gerekir. Karar kriterlerinin mümkün olan en doğru şekilde değerlendirilmesi için faklı kaynaklardan veriler toplanır. Örneğin; bir araba satın alma sürecinde karar kriterlerinin sağlıklı bir biçimde şekillendirildiğinden emin olmak için öncelikle ilgili aracın satış bayisine gidilip test sürüşü ile kişisel deneyimler elde edilebilir. Buna ek olarak, ilgilenilen araç ile ilgili internet üzerinden çeşitli araştırmalar yapılıp, farklı insanların görüşleri incelenir. Bunun haricinde aracı yeteri kadar uzun süre deneyimlemiş ve güven duyulan kişiler varsa, aracın olumlu ve olumsuz yanları hakkındaki görüşleri öğrenilip onların deneyimlerinden faydalanılır. Nihai karar ise, tüm araştırmalardan elde edilen kolektif çıkarımlar doğrultusunda oluşturulur.

Araba satın alma süreciyle ilgili durumu kolektif öğrenmeye benzetilebilir. Kolektif öğrenme birden çok makine öğrenmesi modelinden gelen tahminleri birleştirerek daha etkin bir sınıflandırma yapmayı amaçlayan yaklaşımdır. Araba analojisinde bahsedilen kişisel deneyimler, başka kullanıcıların yorumları, ilgilenilen arabaya sahip olan ve güven duyulan kişilerin görüşleri aslında farklı birer modelin çıktısı olarak nitelendirilebilir.

Kolektif öğrenmede, nihai karar için her bir modelin ağırlıkları eşit alınabildiği gibi, belli modellerin önemine istinaden kendilerine farklı ağırlıklar da tayin edilebilir. Bu

durum daha çok güven duyulan modelin karar verme sürecinde etkisinin daha fazla olmasını sağlar. Kolektif öğrenme akış şeması Şekil 4.6'da verilmiştir.



Şekil 4.6 : Temel kolektif öğrenme [48].

Çalışmada kullanılan kollektif öğrenme tabanlı makine öğrenmesi algoritmaları Rassal Orman, GBM, LightGBM, XGBoost, AdaBoost ve CatBoost algoritmalarıdır.

Rassal orman algoritması torbalama (bagging) yaklaşımı kullanırken, GBM, LightGBM, XGBoost, AdaBoost ve CatBoost yükseltme (boosting) yaklaşımı kullanan algoritmalardır.

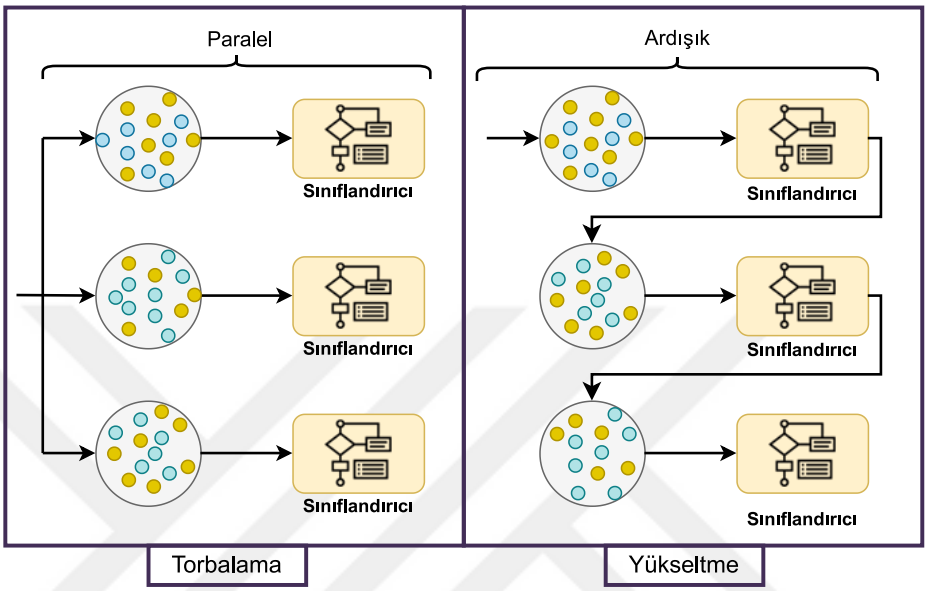
En temel ifadeyle, kolektif öğrenme modeli içerisindeki zayıf öğrenicilerin her biri birbirinden bağımsız ve paralel çalışıyorsa buna torbalama (bagging), eğer her bir zayıf öğrenici birbiri ile ardışıl olarak çalışıyorsa buna yükseltme (boosting) denir.

Bagging yöntemi "Bootstrap Aggregating (Önyükleme Toplaması)" kelimelerinin kısaltılmış bir biçimde ifade edilmesi ile isimlendirilen ve 1996 yılında Leo Breiman tarafından ortaya konulan bir yöntemdir. Belirlenen temel sınıflandırıcının eğitilmesi için eğitim verisinden rastgele örnekleme yoluyla çok sayıda alt veri kümesi oluşturulur ve her bir alt küme ile birbirinden bağımsız ve paralel bir şekilde temel sınıflandırıcılar eğitilir.

Boosting yöntemi ise, çok sayıda zayıf sınıflandırıcının ardışıl bir şekilde eğitilmesi ve nihai olarak güçlü bir sınıflandırıcının elde edilmesi yoluyla ortaya konulan bir

yöntemdir. Bu yöntemde her bir iterasyonda yanlış sınıflandırılmış verilerin ağırlıkları artırılarak bir sonraki iterasyonda ilgili zayıf sınıflandırıcının daha önceden yanlış sınıflandırılmış verilere odaklanması sağlanır. Nihai olarak ise güçlü bir sınıflandırıcı elde edilir. [49]

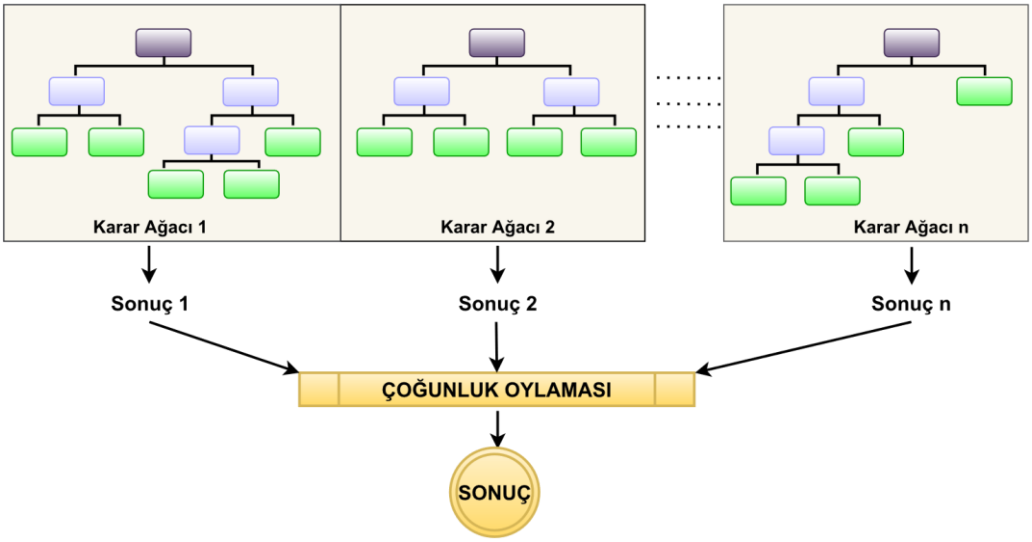
Torbalama ve yükseltme yaklaşımlarının işleyiş biçimleri Şekil 4.7'de gösterilmiştir.



Şekil 4.7 : Torbalama ve yükseltme işleyiş biçimleri [50].

4.1.2.1 Rassal orman (Random forest -RF)

Rassal orman kolektif makine öğrenmesi algoritmalarından biri olup, Şekil 4.8'de görüldüğü biçimde içerisinde birden fazla karar ağacı barındırır.

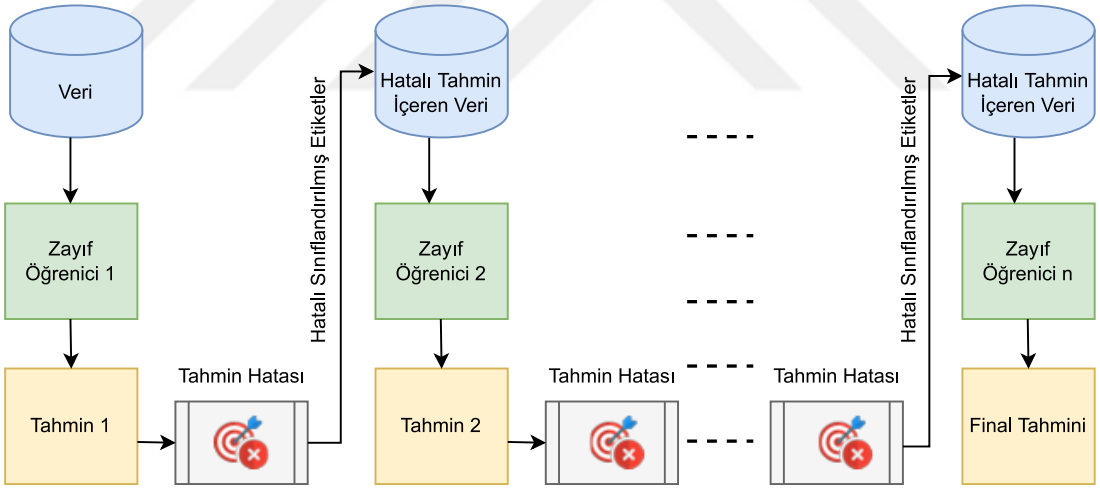


Şekil 4.8 : Rassal orman algoritması [51].

Karar ağaçları için önemli bir problemlerden biri veriyi ezberleme sorunudur. Rassal orman algoritmasında bu sorunun çözümü için, her bir tekil karar ağacında birbirinden farklı rastgele öznitelikler kullanılır. Bu sayede birbiri ile yüksek korelasyona sahip bağımsız ağaç yapılarından kaçınılmış olunur ve ağaç çeşitliliği artar [51]. Eğitilen her bir ağaç yapısı, kendi tahmin sonuçlarını oluşturur. Bağımsız karar ağaçlarının tahmin sonuçları çoğunluk oylaması ilkesine göre değerlendirilerek en çok tekrar eden sınıflandırma sonucu nihai sınıflandırma sonucu olarak kabul edilir [52].

4.1.2.2 Gradyan yükseltme (Gradient boosting - GB)

Gradyan yükseltme 2001 yılında Friedman tarafından ortaya konulmuştur. Sınıflandırma ve regresyon problemleri için kullanılan güçlü algoritmalardan biridir [53]. Bu makine öğrenimi algoritması, modellerin hızını ve performansını artırmak için ardışıl karar ağaçlarına dayanan bir kolektif model olarak çalışır. Güçlü bir öğrenici oluşturmak için yükseltme tekniği kullanılarak bir dizi zayıf öğreniciden faydalanılır. Gradyan yükseltme algoritmasının işleyişi Şekil 4.9'da verilmiştir.



Şekil 4.9 : Gradyan yükseltme [53].

Şekil 4.9'da görüleceği üzere ardışıl bağlanan her bir ağaç kendisinden bir önceki ağacın hatasını minimuma indirmeye çalışır. İşleyişi kısaca şu şekildedir:

1. Öncelikle ilk karar ağacı eğitilir.
2. Eğitilen karar ağacı ile tahmin yapılır.
3. Tahmin sonucunda elde edilen etiket değerleri gerçek etiket değerleri ile karşılaştırılır ve hatalı etiketler tespit edilir.

4. Hatalı olarak tespit edilen etiketler ardışıl olarak bir sonraki ağacı eğitmek için kullanılır.
5. Ardışıl ağaçlardan oluşmuş olan eğitimler tamamlanana kadar yukarıdaki adımlar tekrarlanır.
6. Birçok ardışıl zayıf öğreniciden güçlü bir öğrenici elde edilerek son tahmin yapılır.

4.1.2.3 Ekstrem gradyan yükseltme (Extreme gradient boosting - XGBoost)

Ekstrem gradyan yükseltme algoritması, gradyan yükseltme algoritmasının gelişmiş bir uygulaması olup 2016 yılında Chen ve Guestrin tarafından önerilmiştir. Hız ve tahmin gücü ile ön plana çıkmaktadır. Gradyan yükseltme algoritması ile kıyaslandığında yaklaşık 10 kat daha hızlı çalışır. Yüksek seviyede esnekliğe sahiptir. Hesaplanan benzerlik puanlarına göre kazancı hesaplar ve geriye doğru budama işlemi gerçekleştirir. Ayrıca aşırı öğrenmeyi azaltan ve performansı iyileştiren birtakım düzenlemeler içerir [54].

XGBoost algoritması, bir GB algoritma çerçevesidir ve bu algoritma ile aynı yöntemi izlemektedir. Aralarındaki fark ise bir dizi sistem optimizasyonuna ve algoritmik iyileştirme detaylarında yatmaktadır. XGBoost algoritması farklı regülarizasyon teknikleri kullanarak ve ağaçların karmaşıklığını kontrol ederek daha yüksek performans sergileyebilmektedir [55]. XGBoost ağaç yapısında önceliği derinliğe vererek hesaplama performansını önemli ölçüde arttırır. Ayrıca sistemsel optimizasyon sayesinde XGBoost algoritması benzerlik skoru ve ağaç çıktılarını ön bellekte gerçekleştirebildiğinden dolayı hızlı hesaplamalar yapabilmektedir.

4.1.2.4 Hızlı ekstrem gradyan yükseltme (Light extreme gradient boosting - Light GB)

Microsoft tarafından geliştirilen bu algoritma, XGBoost algoritmasının performansının artırılmış halidir. Özellikle büyük veri kümelerinden diğer algoritmalara göre üstünlük sağlar. Daha az hafıza kullanımı ile daha yüksek performans sergiler. LightGBM algoritmasını diğer ağaç tabanlı algoritmalardan ayıran özelliği ise seviye bazında büyüme yapan diğer algoritmaların aksine yaprak tabanlı büyüme stratejisinde çalışmasıdır. Yaprak bazında büyüme stratejisi bu algoritmanın daha hızlı öğrenmesini sağlar. Ancak bu strateji küçük veri setlerinde

modelin aşırı öğrenmeye yatkın olmasına sebebiyet verir. Bu yüzden daha iyi bir performans için büyük veri setleri ile kullanılması önerilir [56].

LightGBM algoritmasını, geleneksel gradyan yükseltme algoritmasından ayıran en önemli nokta ise, bünyesinde gradyan tabanlı tek taraflı örnekleme (Gradient Based One Side Sampling, GOSS) ve ayrıcalıklı öznitelik desteleme (Exclusive Feature Bundling, EFB) algoritmalarını barındırmasıdır [57]. GOSS yöntemi ile küçük gradyanlara sahip veri örneklerinin önemli bir kısmı hariç tutularak bilgi kazancının tahmin edilmesi için yalnızca kalan önemli kısımlar kullanılır. EFB yöntemiyle de değişkenlerin boyutunu azaltmak amacıyla seyrek yapıdaki özellikler daha sık ve yoğun özelliklere dönüştürerek işlem karmaşıklığını azaltmaktadır [58]. Bu özellikler LightGBM algoritmasının geleneksel gradyan yükseltme algoritmasından yaklaşık 20 kat daha hızlı çalışmasını sağlar.

4.1.2.5 Uyarlamalı yükseltme (Adaptive boosting - AdaBoost)

"Adaptive" ve "Boosting" kelimelerinden türetilmiş olan AdaBoost algoritması 1997 yılında Freund ve Schapire tarafından önerilen bir kolektif öğrenme algoritmasıdır. Çok miktarda ardışıl zayıf sınıflandırıcılardan faydalanılarak nihai bir güçlü sınıflandırıcı elde edilmesi esasına dayanır. İlk başta veri setindeki her bir gözlem eşit ağırlıklara sahiptir. Her bir öznitelik için "stump" adı verilen, sadece 1 kök düğümünden ve 2 yapraktan meydana gelen çok sayıda yapı oluşturulur [59]. Zayıf öğreniciler ile model eğitilir ve tahminleme yapılır. Zayıf öğrenicinin tahminindeki hataların ağırlığı artırılarak bir sonraki zayıf öğreniciye aktarılır. Bu şekilde kümülatif olarak hataların ağırlığı her iterasyonda güncellenir. En az hataya sahip olan zayıf öğrenici nihai tahminde daha fazla söz sahibi olur. Son olarak da tüm zayıf öğrenicilerden nihai bir tahmin ortaya konulur [60].

4.1.2.6 Kategorik yükseltme (Categorical boosting – CatBoost)

"Category" ve "Boosting kelimelerinden türetilmiş olan CatBoost algoritması 2017 yılında Yandex firmasındaki mühendisler ve araştırmacılar tarafından LightGBM ve XGBoost algoritmalarına alternatif olarak tanıtılmıştır. CatBoost algoritması karar ağaçlarında gradyan arttırma (yükseltme, güçlendirme) yöntemlerinden biridir. İsmindeki "kategorik" kelimesinde rağmen sadece kategorik veriler ile çalışmaz, bunun yanından nümerik ve metin verileri ile de çalışabilen bir algoritmadır. Kategorik veriler ile çalışabilmek için diğer algoritmalarda kodlama yöntemleri ile

kategorik değişkenlerin sayısal değerlere dönüştürülmesi işlemi yapılırken, CatBoost algoritması ile çalışırken buna gerek yoktur.

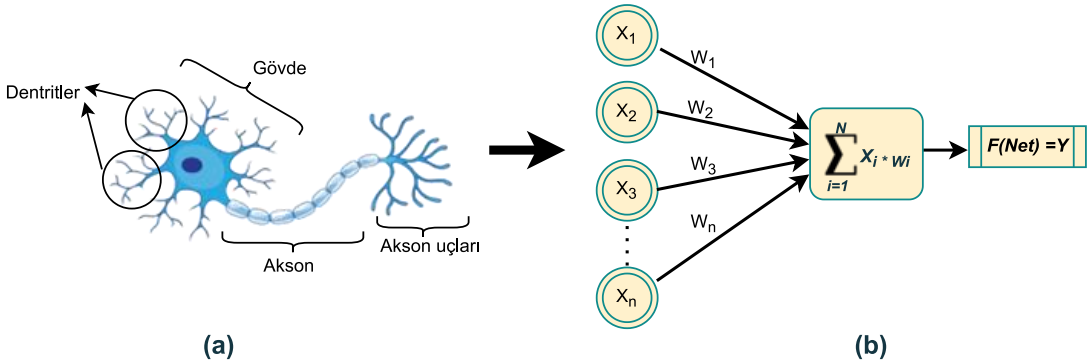
4.2 Derin Öğrenme

Derin öğrenme, yapay sinir ağları olarak adlandırılan ve insan beyninin yapısından, işleyişinden ilham alan algoritmalara dayalı bir makine öğrenmesi alt alanıdır [61].

Derin öğrenme konseptini daha anlaşılır kılmak adına, bu bölümde öncelikle beynin temel birimi niteliğindeki biyolojik sinir hücresinin yapısı, işleyişi ve kendisine ait terminolojileri ele alınmıştır. İlerleyen kısımlarda biyolojik sinir hücresi ile yapay sinir ağlarının açıklamalarına ve ne şekilde ilişkilendirildiklerine yer verilmiştir.

Yapay sinir ağları insan beynindeki biyolojik sinir hücrelerini taklit eder. İlk kez 1943 yılında biyolojik sinir hücrelerinin yapısı ve çalışma mantığı örnek alınarak, öğrenme sürecinin matematiksel olarak modellenmesi sonucunda yapay sinir ağları ortaya çıkmıştır [62].

Yapay sinir ağları, karmaşık ve komplike işleri esnek bir şekilde temsil edebilmeleri açısından günümüzde son derece ilgi görür hale gelmiştir [63]. Şekil 4.10 üzerinde biyolojik sinir hücresinin yapısı ve yapay sinir hücresi benzetimi görülmektedir.



Şekil 4.10 : Biyolojik sinir hücresinin ve yapay sinir hücresinin benzetimi. (a) Biyolojik sinir hücresi, (b) Yapay sinir hücresi (perceptron) modeli [63].

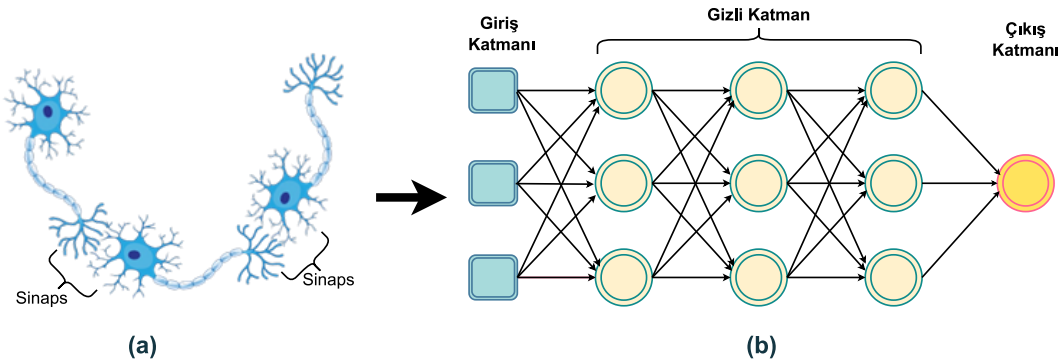
Biyolojik sinir hücresi temelde dentritler, gövde ve akson olmak üzere 3 ana bölümde incelenebilir. Dentritler, giriş bilgilerinin toplayıp gövdeye ileten birimlerdir ve yapay sinir ağlarında birleştirme (toplama) fonksiyonuna karşılık gelmektedir. Şekilde 4.11-b'de X_1, X_2, X_3 ve X_n ile belirtilen her birim bir dentriti temsil eder. W_1, W_2, W_3 ve W_n ise dentritlere ait ağırlıkları gösterir. Biyolojik sinir hücresinin gövde kısmı ise, yapay sinir ağı modelinde matematiksel olarak her bir

dentritten gelen sinyallerin kendi ağırlık değerleri ile çarpılarak toplanması ile elde edilen "Net" denklemine eşittir. (Denklem 4.2)

$$Net = \sum_{i=1}^{N} X_i * W_i \tag{4.2}$$

F(Net) olarak belirtilen transfer fonksiyonu ise "Net" değerinin bir sonraki sinir hücresine iletilecek olan çıktısıdır [63]. Burada hücreye gelen net bilgiye karşılık hücrenin çıktı olarak üreteceği cevap aktivasyon fonksiyonu ile belirlenir. En bilinen ve yaygın şekilde kullanılan aktivasyon fonksiyonlarına, sigmoid fonksiyon, hiperbolik tanjant fonksiyonu, ReLU (Düzenlenmiş lineer birim) fonksiyonu, softmax fonksiyonu örnek olarak verilebilir. Bu aktivasyon fonksiyonları doğrusal olmayan gerçek dünya koşullarının yapay sinir ağlarına tanıtılması amacıyla kullanılır [64]. Biyolojik sinir hücresindeki aksonlar ise transfer fonksiyonu bilgisinin bir sonraki sinir hücresine aktarılmasını sağlayan çıkış birimi olarak nitelendirilebilir.

Algılayıcılar, doğrusal olmayan problemlere yeteri kadar iyi çözümler üretemediği için 1960 yılı itibariyle Widrow ve Hoff'un çalışmaları ile ilk kez çok katmanlı sinir ağı (MLP) yapısına geçilmiştir [65]. Bu yapı Şekil 4.11'de gösterilmiştir.



Şekil 4.11 : Biyolojik sinir ağının ve çok katmanlı yapay sinir ağlarının benzetimi. (a) Biyolojik sinir ağı, (b) Çok katmanlı yapay sinir ağı [63].

Tek katmanlı yapay sinir ağları sadece giriş ve çıkış katmanlarından oluşurken, çok katmanlı sinir ağlarında ise giriş katmanı ile çıkış katmanı arasında 1 veya daha fazla sayıda gizli katman bulunmaktadır. Her bir gizli katman çok sayıda nörondan oluşmaktadır. (Şekil 4.11)

Bir eğitim kümesindeki özniteliklerin değerleri yapay sinir ağlarına giriş olarak verilirken, çıkışta elde edilen sonucun ise gerçek etiket değerine yakın olması

beklenir. Gerçek etiket değerine en yakın çıktıların bulunabilmesi için W ağırlık katsayılarının sürekli güncellenmeleri sağlanır. Bu süreçte yapay sinir ağlarında öğrenme işlemi geri yayılım algoritması (gradient descent algoritması) ile gerçekleştirilerek nöronlar arasındaki bağlantı ağırlıkları güncellenmiş olur.

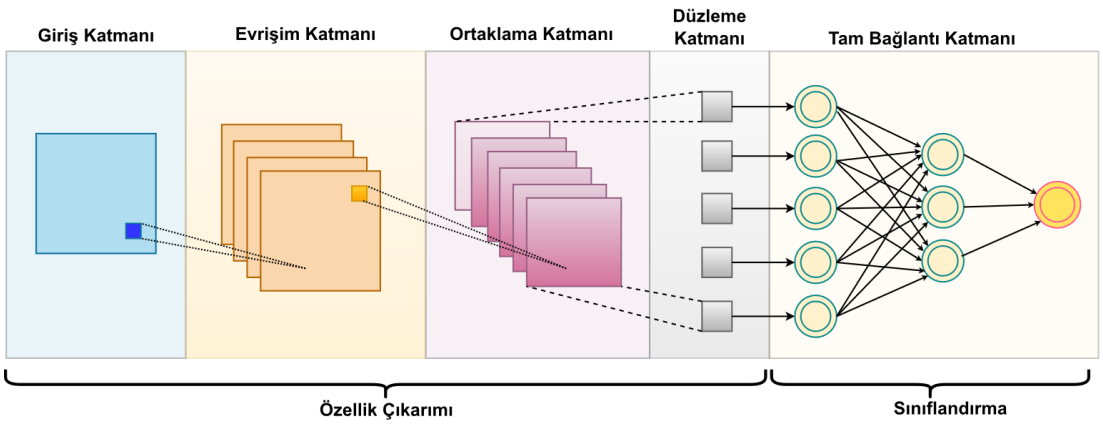
Ele alınan çalışma kapsamında derin öğrenme algoritmaları olarak;

- Evrişimli sinir ağları (CNN),
- Özyinelemeli sinir ağlarından türetilmiş olan uzun ömürlü kısa dönem bellek (LSTM),
- CNN tabanlı LSTM yapısının kullanıldığı hibrit bir model

incelenmiştir.

4.2.1 Evrişimli sinir ağları (CNN)

Derin öğrenmenin en bilinen algoritmalarından biri evrişimli sinir ağlarıdır. Evrişimli sinir ağları başta görüntü işleme ve analiz, nesne tanıma ve sınıflandırma gibi işler olmak üzere pek çok alanda kullanılır [66]. "Evrişimli Sinir Ağı" diye nitelendirilmesinin sebebi ise giriş verisi üzerinde öznitelik çıkarımı için öncelikle evrişim ya da konvolüsyon denilen matematiksel bir işlemin uygulanmasıdır. Evrişimli sinir ağlarının mimarisi, Şekil 4.12'de gösterildiği gibi, giriş katmanı, evrişim (konvolüsyon) katmanı, ortaklama (pooling) katmanı ve tamamen bağlı katman olarak incelenebilir.



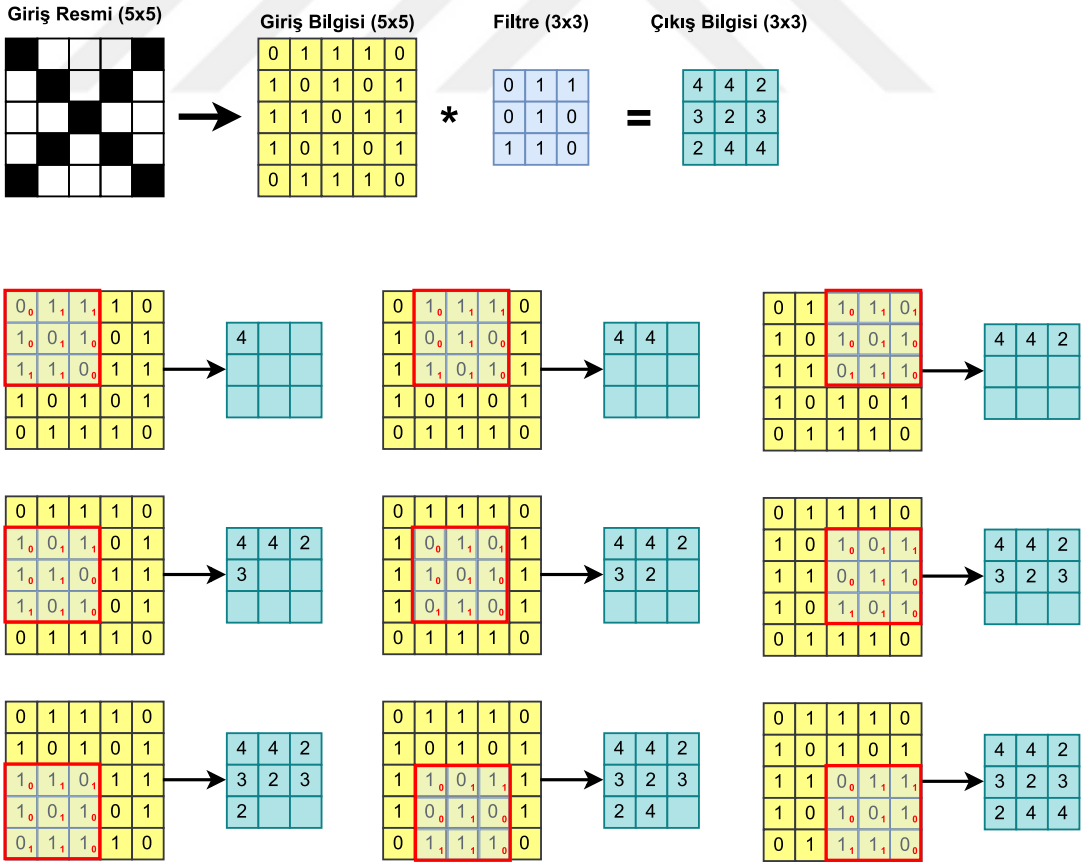
Şekil 4.12 : CNN mimarisi [66].

Giriş katmanı: CNN mimarisinin ilk kısmı, giriş katmanı adı verilen ve dış dünyadan alınan ham verilerin konvolüsyon işlemi için ağa verilmesini sağlayan

katmandır. [67] Girdilerin veri çözünürlüğü ve boyutu modelin mimarisine göre iyi bir şekilde seçilmelidir. Başarı oranının artırılması için çok sayıda veri girdisi gerekse de bu aynı zamanda modelin eğitim süresi ve test süresinin uzun olmasına ve daha fazla bellek tüketimine sebep olacaktır. Bu durum daha yüksek bir işlemci ve bellek kapasitesini gerekli kılar [68].

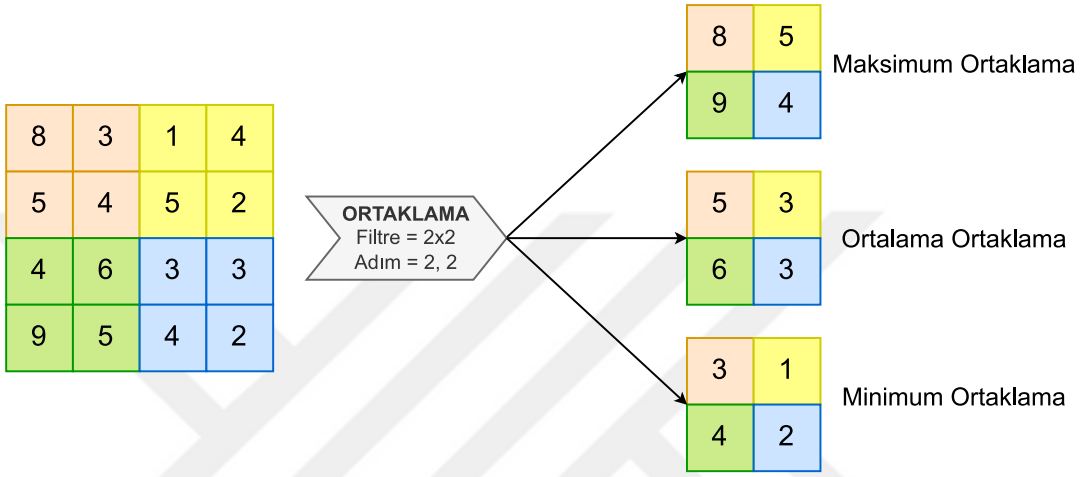
Evrişim katmanı: Evrişimli sinir ağlarının temelini oluşturan bu katman evrişim, konvolüsyon ya da dönüşüm katmanı adıyla bilinir. Bu işlem esnasında belirli bir filtrenin tüm giriş verisi üzerinde adım sayısı (stride) kadar kaydırılarak dolaştırılması neticesinde daha küçük boyutta öznitelik çıkarımının yapılması sağlanır [69]. Giriş verisini temsil eden yeni öznitelik, her bir adımda giriş verisi ile filtrenin örtüştüğü değerlerde matris iç çarpım işlemi yapılarak hesaplanır. (Şekil 4.13)

Bir CNN mimarisinde birden fazla ardışıl evrişim katmanı bulunabilir. Konvolüsyon katmanındaki filtre 2x2, 3x3, 4x4 gibi farklı boyutlara sahip olabilir. Filtre boyutu parametresi sınıflandırma görevindeki başarıya etki eden bir parametredir.



Şekil 4.13 : Filtrenin giriş verisi üzerinde kaydırılarak yeni öznitelik çıkarımın yapılması [70].

Ortaklama (pooling) katmanı: Bu katman ardışık evrişim katları arasına hesaplama karmaşıklığını azaltmak amacıyla sıklıkla konulur. Burada öğrenilen herhangi bir parametre bulunmaz. En popüleri maksimum ortaklama (maximum pooling) olmak üzere, ortalama ortaklama (mean pooling) ve minimum ortaklama (minimum pooling) gibi farklı ortaklama çeşitleri de mevcuttur. Şekil 4.14 üzerinde minimum, maksimum ve ortalama ortaklama işlemleri gösterilmiştir.



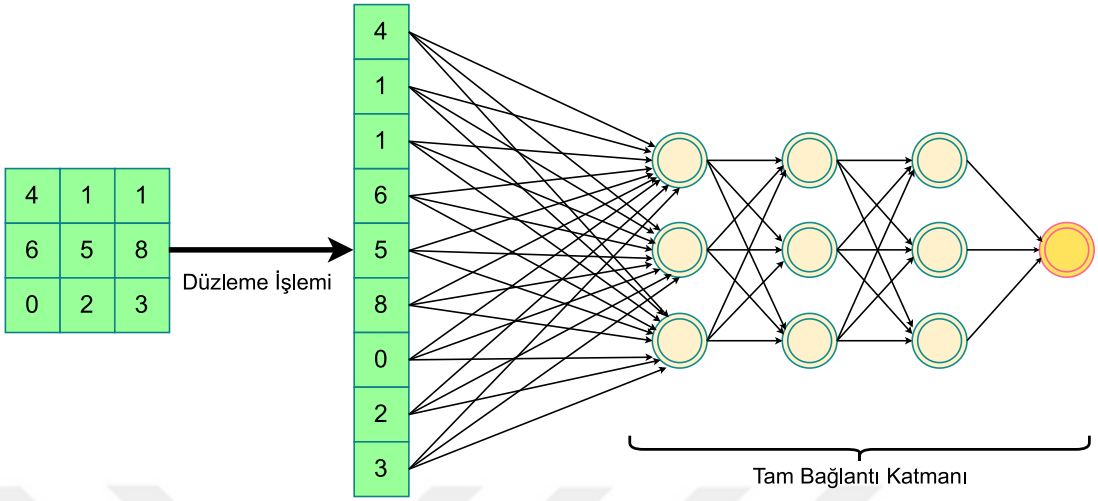
Şekil 4.14 : Maksimum, ortalama ve minimum ortaklama işlemi [71].

Maksimum ortaklama, özellik haritasının ortaklama çerçevesi ile kesiştiği değerler arasından en büyüğünü, ortalama ortaklama tüm değerlerin aritmetik ortalamasını, minimum ortaklama ise değerler arasından minimum değeri seçer. Bu sayede önemli öznitelikler korunarak özellik haritasının boyutu azaltılır. Bir nevi "aşağı örnekleme" işlemi yapılmış olur. Ortaklama işlemi ile tekrarlanan bilgiler ortadan kaldırıldığı için hesaplama karmaşıklığının yanında aynı zaman aşırı öğrenme (ezberleme) probleminin de üstesinden gelinir.

Düzleme (flattening) katmanı: Bu katman evrişim ve ortaklama katmanları sonrası elde edilen çok boyutlu matrisi tam bağlantı katmanına aktarılmadan önce tek boyutlu vektör haline çevirir. Bu sayede veri sinir ağlarının işleyebileceği şekilde tek boyutlu hale getirilebilir. (Şekil 4.15)

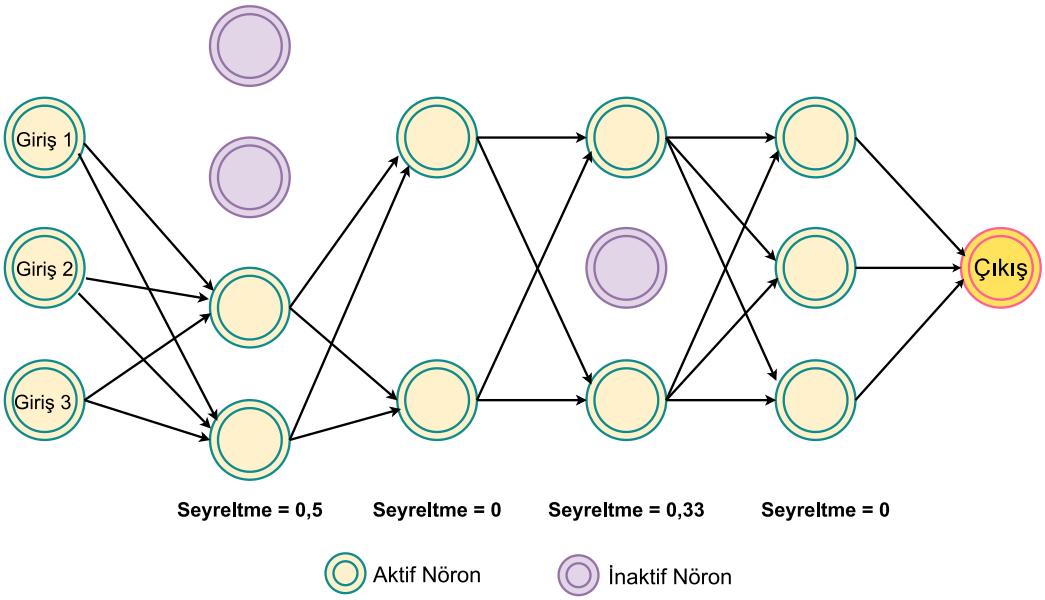
Tam bağlantı katmanı: Bu katman CNN yapısının son ve en önemli katmanıdır. Tam bağlantı katmanına kadar olan katmanlarda tamamen öznitelik çıkarımı üzerine işlemler yapılmıştır. Bu katman ise çok katmanlı sinir ağı yapısına tekabül eder ve tek boyutlu dizi halinde beslendiği verilerle sınıflandırma görevini yerine getirir [72].

Buradaki tam bağlantı katmanı da birden çok sayıda gizli katmana sahip olacak şekilde tasarlanıp optimum sınıflandırma sonucu için çeşitlendirilebilir. (Şekil 4.15)



Şekil 4.15 : Düzleme ve tam bağlantı katmanı [72].

Seyreltme (dropout): Bir CNN modelinde özellikle tam bağlantı katmanındaki çeşitlilik modelin çok büyümesine ve karmaşık hale gelmesine neden olabilir. Büyük modeller eğitim süresi uzunsa ve veri sayısı azsa aşırı öğrenmeye meyleder. Yapay sinir ağları eğitiminde modelin aşırı öğrenmesini önlemek amacıyla bir sonraki katmana geçmeden önce rastgele belirlenmiş bazı nöronların geçici olarak göz ardı edilmesi işlemine seyreltme (dropout) denir. Seyreltme işlemi genelde tamamen bağlı katmanda kullanılır. Öğrenme sürecinin her bir iterasyonunda belirlenen olasılık miktarına göre nöronlar Şekil 4.16'da görüldüğü gibi göz ardı edilir. Örneğin 0,5 olarak belirlenmiş seyreltme değeri için nöronların %50'si rastgele bir şekilde geçici süre unutulur [73].



Şekil 4.16 : Seyreltme işlemi [74].

Aktivasyon fonksiyonları:

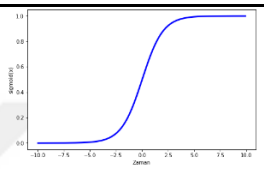
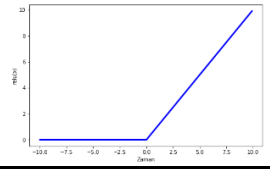
Yapay sinir ağlarını optimize ederken seçmemiz gereken parametrelerden biri, sinir ağlarının çıkışını belirleyen aktivasyon fonksiyonunun çeşididir. Aktivasyon fonksiyonları, sinir ağının video, resim, ses gibi verilerdeki karmaşık örüntüleri öğrenmesini sağlar. Aktivasyon fonksiyonu kullanılmadığında sinyal çıkışı basit bir doğrusal fonksiyon olacaktır ve lineer regresyon gibi davranacaktır. Buna karşın video, resim, ses gibi gerçek dünya verilerindeki karmaşık örüntüleri öğrenme kabiliyeti tek dereceli polinom şeklinde ifade edilebilecek bir doğrusal fonksiyon ile değerlendirilmeye çalışıldığında, öğrenme yeteneği çok kısıtlı kalacaktır [64]. Sigmoid, tanh, ReLU, Leaky ReLU, softmax gibi birçok aktivasyon fonksiyonu literatürde kullanılmaktadır. Bu aktivasyon fonksiyonlarından softmax, çoklu sınıflandırma problemlerinde kullanılmaktadır. İkili sınıflandırma yapılan bu çalışma kapsamında ise tam bağlantı katmanlarında ReLU, yapay sinir ağı çıkış katmanlarında ise Sigmoid fonksiyonları kullanılmıştır.

ReLU (doğrultulmuş lineer birimler) aktivasyon fonksiyonu: Derin öğrenme algoritmalarında en yaygın olarak kullanılan aktivasyon fonksiyonlarının başında gelir. Bu fonksiyon 0 ile $+\infty$ arasında bir çıktı değeri üretir. Eğer giriş değeri 0'dan büyük ise çıkışa aynen aktarılır. Ama giriş değeri 0'dan küçük ise 0 olarak çıkışa aktarılır [72].

Sigmoid aktivasyon fonksiyonu: "S" şeklinde bir eğri grafiğe sahip olan aktivasyon fonksiyonudur. Gelen girdi değerini 0 ile 1 arasındaki aralığa konumlandırır. Tahmin edilen değeri olasılıklar ile eşleştirmek için bu fonksiyon kullanılabilir. Örneğin, bir elma resmini giriş olarak verdiğimiz yapay sinir ağının çıkışında yer alan sigmoid fonksiyonu 0.90 değeri üretmişse, bu resmin %90 olasılıkla elma olduğunu belirtir.

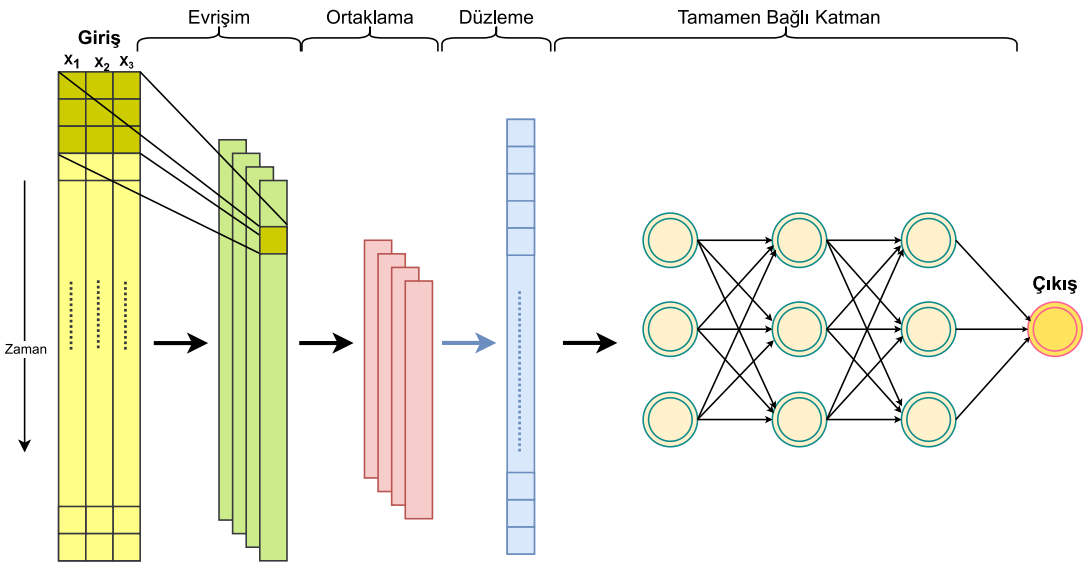
ReLU ve Sigmoid fonksiyonlarının formülleri Çizelge 4.1'de verilmiştir.

Çizelge 4.1 : ReLU ve Sigmoid aktivasyon fonksiyonları.

Aktivasyon Fonksiyonu	Denklem	Çıktı Aralığı	Fonksiyon Grafiği
Sigmoid	$f(x)=\frac{1}{1+e^{-x}}$	(0, 1)	
ReLU	$f(x)=\begin{cases} 0 \text{ için } x<0 \\ x \text{ için } x\geq 0 \end{cases}$	(0, ∞)	

Evrişimli sinir ağları görüntü analiz ve sınıflandırma gibi işlerde yaygın olarak kullanılsa da zaman serisi verileri ve doğal dil işleme gibi işlerde de kullanılmaktadır. Bu gibi durumlarda CNN yapısına girdi olarak verilecek verilerin yapısı da farklılık göstermektedir. Giriş ve çıkış boyutlarına göre CNN yapıları 1 boyutlu, 2 boyutlu ve 3 boyutlu CNN olmak üzere sınıflandırılmaktadır. Bu çalışma kapsamında kestirimci bakım uygulama örnekleri ile çalışıldığından dolayı 1 boyutlu CNN yapıları kullanılmıştır. (Şekil 4.17)

1 boyutlu CNN'lerde giriş karakteristiği ve zaman adımları olmak üzere 2 boyutlu bir veri giriş ve çıkışı vardır [75]. Bu tip CNN yapılarında evrişim işlemi esnasında filtre sadece 1 boyutta (zaman ekseninde) ilerletilir.

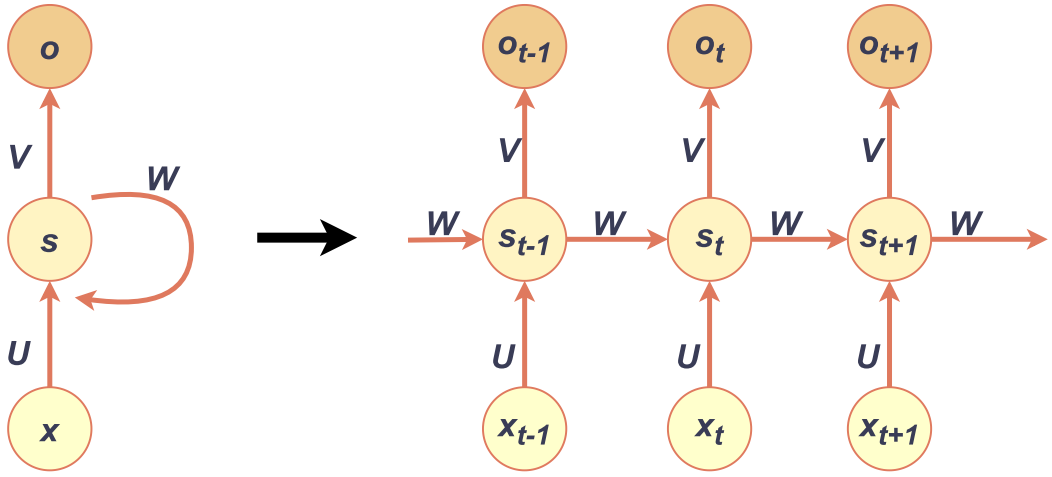


Şekil 4.17 : 1 Boyutlu evrişimli sinir ağı yapısı [75].

4.2.2 Özyinelemeli (Tekrarlayan) sinir ağları (RNN)

Özyinelemeli sinir ağlarının arkasındaki temel fikir sıralı bilgilerin kullanılmasıdır. Diğer sinir ağlarını hatırlayacak olursak, model girdileri birbirinden bağımsızdır. Örneğin bir görüntü sınıflandırmasında modele girdi olarak verilen bir fotoğraf karesinin ait olduğu sınıfı tayin edilirken, bu fotoğrafın bir sonraki sırada modele verilecek olan fotoğraf karesinin sınıflandırılmasında hiçbir etkisi yoktur. Yani sıraya ve zamana bağlı bir durum söz konusu değildir. Ancak bu durum, her problem için uygun olmayabilir. Örneğin, bir metin analizi işleminde sıradaki kelimeyi tahmin edebilmek için daha önce geçmiş olan kelimelerin bilinmesi önem arz eder. Metin analizleri, konuşma tanıma, zaman serileri biçiminde veri üreten sensör veya istatistiksel bilgilerin analizi gibi işlerde RNN kullanımı uygundur.

RNN'lerde üretilen çıkışlar, mevcut ve daha önceki bilgilerin birlikte kullanılması ile elde edilir. Şekil 4.18'de görüleceği üzere bu ağlarda bir işlemin çıkışı, aynı zamanda hemen sonraki işlemde giriş olarak kullanıldığı için diğer sinir ağlarından farklıdır [76]. Yani RNN'ler sıradaki adımı takip edebilmek için girdiler arasında bağıntı kurar ve eğitim esnasında kurdukları bu bağıntıları hatırlarlar. Diğer sinir ağlarından ayıran en önemli özelliği RNN'lerin bir belleğinin olmasıdır.



Şekil 4.18 : Özyinelemeli sinir ağı yapısı [77].

Şekil 4.18'de ağ yapısı gösterilen RNN için matematiksel ifadelerin anlamı şu şekildedir:

U, V ve W: sırasıyla gizli katman, çıkış katmanı ve durum katmanı ağırlıkları

x_t : t anındaki giriş bilgisi

s_t *:* t anındaki gizli durum bilgisi

o_t *:* t anındaki çıkış bilgisi

t anındaki durum bilgisi s_t, o andaki giriş bilgisi ve bir önceki adımdan (t-1 anından) gelen durum bilgisinin fonksiyonu olarak;

$$s_t = f(Ux_t + Ws_{t-1}) \tag{4.3}$$

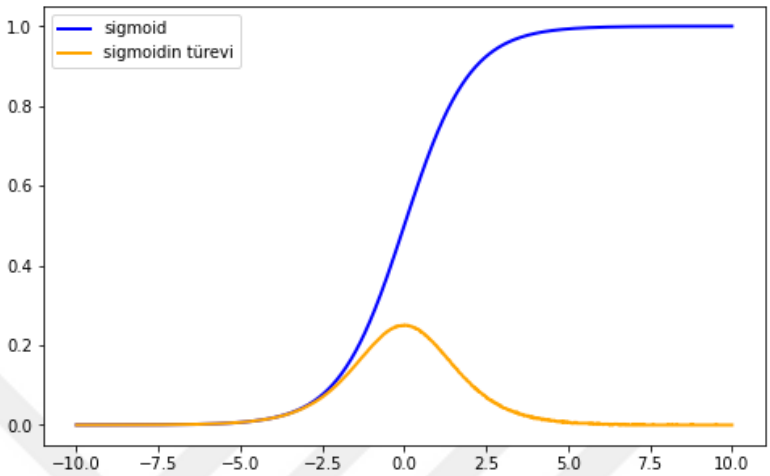
şeklinde ifade edilir.

4.2.3 Uzun ömürlü kısa dönem bellek (LSTM) ağları

LSTM ağları, RNN ağlarının özelleştirilmiş halidir. RNN ağları sıralı bilgilerden ve zaman serilerinden oluşan verileri iyi bir şekilde modelleyebilse de kaybolan/azalan gradyan (vanishing gradient) problemi sebebiyle uzun süreli bağımlılıkları hatırlayamaz [78]. RNN ağlarındaki uzun süreli bellek kayıpları sorununa etkin bir çözüm olarak 1997 yılında Sepp Hochreiter ve Jurgen Schmidhuber tarafından LSTM ağları önerilmiştir. [79]

Kaybolan Gradyan Problemi: Bir veri kümesindeki değerler aktivasyon fonksiyonları ile istenilen aralığa indirgenebilir. Örneğin sigmoid fonksiyonunu ele

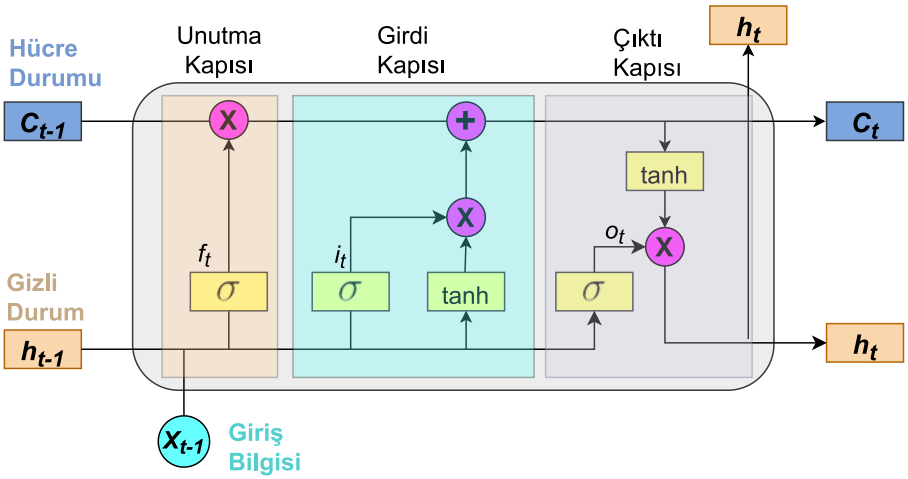
aldığımızda bu fonksiyon tüm çıkış değerlerini 0 ile 1 arasına sığdırır. Sigmoid fonksiyonunun ilk türevinde ise değerler 0 ile 0,25 arasında yer alır. (Şekil 4.19) Kaybolan gradyan problemi ise çok katmanlı sinir ağlarında her işlem adımında ardışık bir şekilde çok sayıda türev alındığında ortaya çıkar.



Şekil 4.19 : Sigmoid fonksiyonu ve onun türevi.

Sigmoid gibi aktivasyon fonksiyonları ile çalışırken, türevlerinin hata fonksiyonunda bulunan küçük değerleri, ilk katmanlara yaklaştıkça birkaç kez çarpılır. Bu çarpımlar neticesinde türev değeri gittikçe 0'a yakınsanır. Böyle bir durumda ise derin sinir ağları çok az güncelleneceğinden dolayı öğrenme olayı minimum düzeye iner ve yerel minimumlara düşmeye neden olabilir [80].

LSTM ağları RNN ağlarına kıyasla daha karmaşık yapıya sahiptir. Tipik bir LSTM yapısı Şekil 4.20'de gösterilmiştir. LSTM ağları içerisinde önceki durum bilgisi ile girdi bilgisini tutan hafıza hücreleri yer alır.



Şekil 4.20 : LSTM yapısı [81].

Şekil 4.20'de de görüldüğü üzere bir LSTM'in çıktısı 3 şeye bağlıdır:

- Hücre Durumu (Cell State) : Ağın mevcut uzun süreli belleği
- Gizli Durum (Hidden State) : Bir önceki noktaya ait çıktı bilgisi
- Mevcut adımdaki girdi bilgisi

LSTM'ler, bir veri dizisindeki bilgilerin ağa nasıl girdiğini, depolandığını ve ağdan nasıl ayrıldığını kontrol eden bir dizi kapı (gate) kullanır. Bir LSTM modelinde unutma kapısı, girdi kapısı ve çıktı kapısı olmak üzere 3 kapı mevcuttur. Bu kapılar birer filtre gibi düşünülebilir ve her birinin kendi sinir ağı yapısı vardır [81].

Unutma kapısı (forget gate): Hem önceki gizli durum hem de yeni girdi verileri göz önünde bulundurularak hangi bilgilerin bellekte tutulacağına ve hangi bilgilerin unutulacağına karar verilen kapıdır. Sigmoid aktivasyon fonksiyonu ile 0 ile 1 arasında değerler üretir. Unutma kapısında bu işlem neticesinde 0'a yakın olan değerler unutulacak, 1'e yakın olan değerler ise bellekte tutulacak verilerdir. Tutulan veriler hücre durumu (cell state) ile taşınmaya devam eder. Unutma kapısı Denklem 4.4'te verilmiştir.

Girdi kapısı (input gate): Bu kısımda hangi bilgilerin hücre durumunda saklanacağını belirlenir. Öncelikle sigmoid aktivasyon fonksiyonu ile bir sinir ağı hangi değerlerin güncelleneceğine karar verir. Bunun haricinde ağı düzenleyen bir tanh aktivasyon fonksiyonu kullanılır. Bu fonksiyon ile yeni aday değer vektörü ortaya çıkarılır. Son olarak sigmoid ve tanh fonksiyonlarının çıkışı çarpılarak hücre durumuna eklenir. Girdi kapısı Denklem 4.5'te verilmiştir.

Çıktı kapısı (output gate): Bu aşamada neyin çıktı olarak gönderileceğine karar verilir ve bu karar verilen çıktı ise bir sonraki hücrenin girişi olur. Çıktı olarak neyin gönderileceğine karar verilirken öncelikle sigmoid fonksiyonlu bir sinir ağı çalıştırılır. Bunun yanından hücre durumu tanh fonksiyonundan geçirilerek sigmoid ağın çıkışı ile çapılır. Böylece çıkışa karar verilmiş olur. Çıktı kapısı Denklemi 4.6'da verilmiştir.

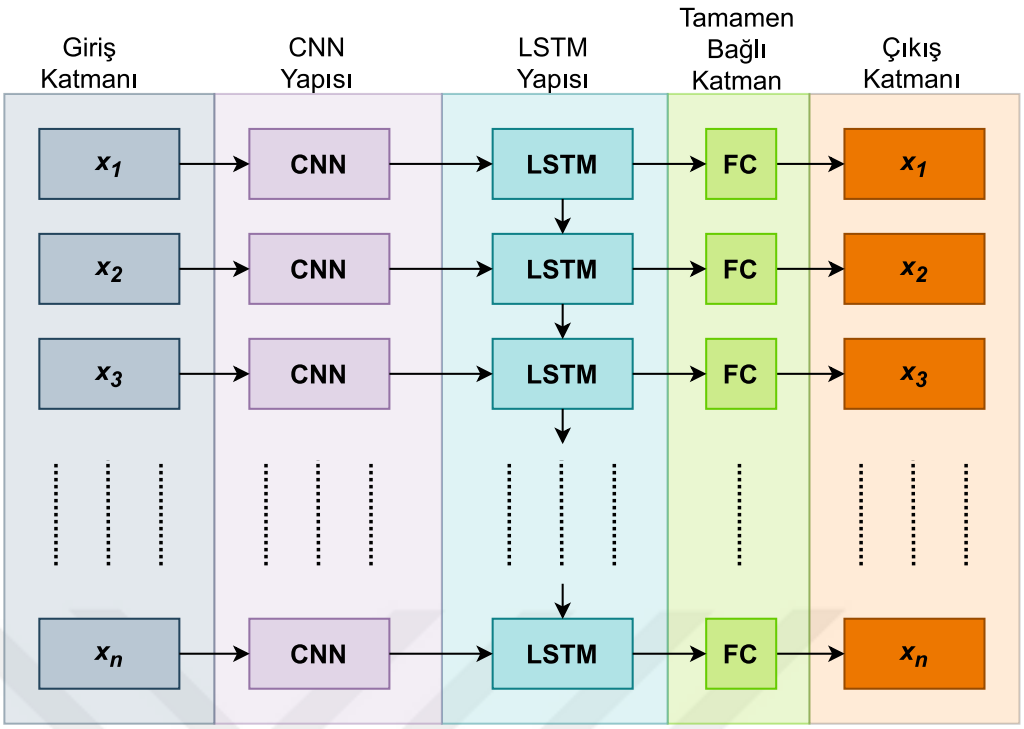
Unutma Kapısı $$F_t=\sigma(W_F h_{t-1}+W_F x_t+b_F)$$ (4.4)

Girdi Kapısı $$I_t=\sigma(W_I h_{t-1}+W_I x_t+b_I)$$ (4.5)

Çıktı Kapısı $$O_t=\sigma(W_O h_{t-1}+W_O x_t+b_o)$$ (4.6)

4.2.4 CNN tabanlı LSTM (CNN-LSTM)

CNN, verilerden etkili bir şekilde öznitelik çıkarabilen güçlü bir derin öğrenme yöntemi olarak bilinirken, LSTM ise özellikle zaman serisi verilerinin sınıflandırılmasında etkin ve güçlü bir derin öğrenme yöntemi olarak bilinir. CNN tabanlı LSTM mimarilerde bu her iki yöntemin avantajları bir araya getirilmiştir. CNN ile öznitelik çıkarımı yapılır ve bu LSTM modeline verilir. CNN-LSTM mimarisi Şekil 4.21'de gösterilmektedir.



Şekil 4.21 : CNN-LSTM yapısı [82].

CNN-LSTM genellikle hareket tanıma, görüntü sınıflandırma ve video sınıflandırma için kullanılır. Görsel zaman serisi tahmin problemlerinin uygulanması ve görüntü dizilerinden metinsel açıklamaların üretilmesi için geliştirilmişlerdir. [82] Ancak kullanım alanları bunlarla sınırlı değildir. A. Nasser ve H. Al-Khazraji kestirimci bakım çalışmasında bir CNN-LSTM modeli yaklaşımını önermişlerdir. [16] Bu çalışmada CNN-LSTM modeli klasik LSTM yöntemine göre daha yüksek performans göstermiştir.

5. MATERYAL VE YÖNTEM

5.1 Kullanılan Teknolojiler

Bu tez çalışmasında, deney tasarımı Python programlama dili ile Anaconda Navigator yazılımı içerisinde yer alan Jupyter Notebook programlama geliştirme aracı kullanılarak gerçekleştirilmiştir. Kodlama kısmında Python'ın numpy, pandas, scikit-learn, matplotlib, seaborn, tensorflow, imblearn ve os kütüphanelerinden (modüllerinden) faydalanılmıştır.

Python programlama dili, 1990'lı yılların başlarında geliştirilmeye başlanmış olan yorumlamalı, etkileşimli, nesne yönelimli bir yüksek seviye programlama dilidir [83]. Windows, Linux, Mac, Symbian gibi hemen hemen her türlü platformda çalışmaktadır. Python dilinin en önemli özelliklerden biri C, C++ gibi diğer dillerin aksine derlenmeye gerek duyulmadan çalıştırılabilir olmasıdır. Bunun yanında diğer programlama dillerine kıyasla basit söz dizimi sayesinde son yıllarda popülerlik kazanmıştır ve dünya genelinde kullanıcı sayısı bir hayli artmıştır. Modüler yapısı ve kolaylıkla erişilebilir çok çeşitli kütüphanelerinin bulunması sayesinde, özellikle yapay zekâ ve veri bilimi alanındaki çalışmalarda yüksek oranda tercih edilmesini sağlamıştır.

Anaconda Navigator ise yapay zekâ ve veri bilimi uygulamaları için açık kaynaklı, ücretsiz ve tümleşik bir Python dağıtımıdır. İçerisinde yapay zekâ ve veri bilimi çalışmaları başta olmak üzere birçok alanda kullanılabilecek yüzlerce açık kaynak modül barındırmasının yanında, Jupyter Notebook, Spyder, JupyterLab, RStudio gibi geliştirici arayüzleri de içerir. Bu çalışmada program geliştirme arayüzü olarak Jupyter Notebook tercih edilmiştir.

Jupyter Notebook, web tarayıcısı üzerinden çeşitli kod ve metin içeriklerini düzenleyip çalıştırmayı sağlayan bir sunucu-istemci uygulamasıdır. Python haricinde, Julia, R, Ruby gibi programlama dillerini de desteklemektedir [84]. Oluşturulan program parçacıklarını farklı hücrelere bölüp yazma ve her bir hücreyi

istenilen sıra ile çalıştırma özellikleri vardır. Veri işleme, veri görselleştirme ve model oluşturma gibi birçok işlem yapılabilir.

Numpy, çok sayıda kişinin katkıda bulunduğu, açık kaynaklı ve Python'un yoğun olarak kullanılan güçlü bir veri manipülasyonu kütüphanesidir. Birçok kaynak tarafından Python'un temel bilimsel bilgi işlem paketi olduğu kabul edilir. Numpy kullanılarak çok boyutlu diziler ve matrisler üzerinde kolayca işlemler gerçekleştirilebilir [85].

Pandas, farklı tipteki ilişkili sayısal ve zaman serisi gibi veri yapılarını birleştirme, ekleme, bölme, yeniden şekillendirme gibi operasyonlar ile veri manipülasyonu ve analitiği için kullanılan popüler Python kütüphanelerinden biridir. Numpy paketi üzerine kurulmuş olan bir kütüphanedir [85].

Scikit-learn, karmaşık verilerle çalışmak için yaygın olarak bilinen bir Python kütüphanesidir. Makine öğrenimini destekleyen açık kaynaklı bir kütüphanedir. Doğrusal regresyon, sınıflandırma, kümeleme vb. gibi çeşitli denetimli ve denetimsiz algoritmaları destekler.

Matplotlib, sayısal verilerin çizilmesinden ve görselleştirilmesinde kullanılan kütüphanedir. Bu yüzden veri analizinde kullanılır. Aynı zamanda açık kaynaklı bir kitaplıktır ve pasta grafikler, histogramlar, dağılım grafikleri gibi birçok tipte grafik ve şekilleri çizmeye olanak sağlar [86].

Seaborn, veri bilimi çalışmalarında oldukça yaygın bir şekilde kullanılan bir diğer veri görselleştirme kütüphanesidir.

Tensorflow, üst düzey hesaplamalar için kullanılan açık kaynaklı bir kitaplıktır. Ayrıca makine öğrenimi ve derin öğrenme algoritmalarında da kullanılır. Çok sayıda tensör işlemi içerir. Genellikle bu Python kütüphanesi matematik ve fizikteki karmaşık hesaplamaları çözmek için kullanır. Bu çalışmada derin öğrenme algoritmaları için kullanılmıştır.

Os, Python'da dizin, dosya ve klasör işlemleri için kullanılan bir kütüphanedir. Farklı işletim sistemlerinde dosya ve dizin operasyonları esnasında notasyon uyumsuzluğunu ortadan kaldırması açısından oldukça faydalı bir kütüphanedir.

Imblearn, dengesiz veri setlerinin dengeli hale getirmek için birçok farklı metot içeren Python kütüphanesidir. Aşırı örnekleme, örnek seyreltme, sentetik veri

oluşturma gibi işlemlerin yapılmasını sağlar. Dengesiz veri kümeleri ile çalışılırken veri ön işleme arasında sıklıkla kullanılmaktadır.

5.2 Veri Setlerinin Tanıtılması

Elinde verisi olan ve bu veriden içgörüler çıkarabilen kurumlar rekabet açısından çok büyük avantajlar elde etmektedir. Bu verilerin toplanması, depolanması ve yönetilmesi ise maliyetli bir iştir.

Makine öğrenmesi ya da derin öğrenme destekli kestirimci bakım uygulamaları için üretim sahasında yoğun ve dağınık bir veri toplama alt yapısına sahip olmak gerekir. Gerekli finansal yatırımını yapmış olan firmaların elde ettikleri bilgiler artık o firmanın özvarlıklarından biri sayılabilir. Doğal olarak yüksek değerdeki ve gizlilikteki bu tarz veri tabanlarını kamuya açık erişim kaynaklarında bulmak oldukça güçtür. Literatürde yer alan, kestirimci bakım alanındaki bazı akademik çalışmalarda kişilerin kendi görev yaptıkları kurumlardan elde edilen gerçek hayat verileri kullanılmış olsa da, literatüre katkı sağlamak isteyen bilim kuruluşları ya da kişiler tarafından laboratuvar ortamında gerçekleştirilen deney tasarımlarından elde edilen veriler ya da sentetik olarak oluşturulan veriler kamunun kullanımına açık olarak paylaşılmaktadır. Örneğin NASA'nın turbo fan motorları ile ilgili yayınlamış olduğu veriler üzerinde onlarca akademik çalışma mevcuttur.

Bu tez kapsamında kestirimci bakım çalışması için literatürde yer alan 2 farklı veri setinden yararlanılmıştır. İki farklı veri setinin kullanılmasının sebebi ise uygulanan benzer yöntemler neticesinde performans başarımlarının veri setine bağlı olarak değişim gösterip göstermeyeceğini görebilmek ve bazı yüksek performanslı sınıflandırma yöntemlerinin genelleştirilip genelleştirilemeyeceği yönünde bir çıkarım yapmaktır.

Çalışmada kestirimci bakım verilerini temsil eden 2 farklı veri setinden faydalanılmıştır:

1) AI4I 2020 Kestirimci Bakım Veri Seti (UCI) [87]

2) Makine Arızası Veri Seti (BigML) [88]

Çalışmanın bundan sonraki kısmında yukarıdaki veri setleri sırasıyla "Veri Seti 1" ve "Veri Seti 2" olarak isimlendirilecektir.

Veri seti 1: Bu veri seti, gerçek kestirimci bakım veri setlerinin elde edilmesi ve özellikle de yayınlanması genellikle zor olduğundan dolayı, S. Matzka tarafından oluşturulan ve endüstriyel ortamda kestirimci bakımı temsil eden sentetik bir veri setidir [89]. Veri seti toplam 10.000 gözlemden oluşmaktadır. Veri seti içerisinde ikili sınıflandırma için "Machine Failure" adında arızanın olup olmadığını belirten bir etiket bulunurken, aynı zamanda oluşan arızanın tipinin belirten 5 farklı etiket daha vardır. Ancak bu çalışma kapsamında ikili sınıflandırma görevi gerçekleştirileceği için odak dışında kalan bu etiketler veri setinden çıkartılarak çalışma gerçekleştirilmiştir. Veri seti Çizelge 5.1'de özetlenmiştir.

Çizelge 5.1 : Veri Seti 1'in öznitelik ve etiket bilgileri özeti.

Özellik	Veri Tipi	Tanım
Tip	Kategorik	Ürün Kalitesi Varyantı (Low, Medium, High)
Hava Sıcaklığı [K]	Nümerik	Ortam sıcaklığı
Proses Sıcaklığı [K]	Nümerik	Proses Çalışma Sıcaklığı
Dönme Hızı [rpm]	Nümerik	2860 W gücündeki çalışma hızı
Tork [Nm]	Nümerik	Makine torku
Ekipman Aşınması [min]	Nümerik	Proseste kullanılan ekipman aşınma süresi
Makine Arızası	Kategorik	Makine Arıza Durumu (İkili etiket – "0": Makine arızası yok, "1": Makine arızası var.)

Veri seti 1 içerisinde yer alan özniteliklerden "Tip" kategorik değişken olduğu için veri ön işleme sırasında bu veri için bir sıcak kodlama (OHE, one hot encoding) işlemi uygulanarak sayılara dönüştürülmesi sağlanmıştır. "Makine Arızası" ise veri setinde etiketi ifade etmektedir. Bu özellik için belirtilen değer eğer 0 ise bu durum makine arızasının olmadığını, eğer 1 ise makine arızası olduğunu gösterir.

Çizelge 5.2'de ise ilgili veri setinden rastgele seçilmiş birkaç gözleme yer verilmiştir.

Çizelge 5.2 : Veri Seti 1'e ait bazı rastgele seçilmiş gözlem örnekleri.

Tip	Hava Sıcaklığı	Proses Sıcaklığı	Dönüş Hızı	Tork	Ekipman Aşınması	Makine Arızası
L	298.1	308.6	1527	40.2	16	0
M	298.3	308.7	1667	28.6	18	0
M	298.5	309	1741	28.0	21	0
H	298.4	308.9	1782	23.9	24	0
H	295.6	306.1	1372	55.6	215	1
L	296.3	307.2	1319	68.3	24	1
H	297	307.8	1385	56.4	202	1
M	296.9	307.8	1549	35.8	206	1

Veri seti 1 içerisindeki 10.000 gözlemden 339 tanesi makinede arızanın olduğu durumlara ait gözlemlerdir. Geri kalan 9661 tanesi ise arıza olmayan durumu temsil

etmektedir. Bu durumda ilgili veri seti sınıf dağılımları bakımından dengesiz bir veri setidir. Dengesizlik oranı (IR) yaklaşık 28.50'dir.

Dengesizlik oranı (IR – imbalance ratio), bilinen en temel sınıf dengesizlik boyutu ölçüsüdür. Basitçe Denklem 5.1'de gösterildiği gibi çoğunluk sınıfına ait örnek sayısının azınlık sınıfına ait örnek sayısına oranından elde edilir [90].

$$IR = \frac{N_{maj}}{N_{min}} \tag{5.1}$$

Bu denklemde IR dengesizlik oranını, N_{maj} çoğunluk sınıfına ait gözlem sayısı, N_{min} ise azınlık sınıfına ait gözlem sayısını ifade eder. IR değeri ne kadar büyük olursa dengesizlik boyutu da o kadar büyük olur. IR değerinin 1 olduğu veri setleri ise sınıf dağılımı ve denge bakımından ideal kabul edilir. Ancak bu çalışmada ele alınan kestirimci bakım gibi gerçek hayata dair problemlerde karşımıza genellikle dengesiz veri setleri çıkar ve bu da başa çıkılması gereken kritik bir husustur.

Veri seti 2: Bu veri seti, saatlik bazda 1 yıl boyunca düzenli olarak veri toplanan bir makineden elde edilmiş veri kümesinden oluşmaktadır. Veri toplamda 8784 adet gözlem içermektedir. (366 gün boyunca, her gün 24'er adet gözlem) Veri seti içerisinde "Failure (Arıza)" isminde makinede arıza olup olmadığını belirten 1 adet sınıflandırma etiketi mevcuttur. Bunun haricinde tarih ve saat bilgileri ile birlikte diğer öznitelikler yer almaktadır. Veri seti 2, Çizelge 5.3'te özetlenmiştir.

Çizelge 5.3 : Veri Seti 2'in öznitelik ve etiket bilgileri özeti.

Özellik	Veri Tipi	Tanım
Tarih	Zaman verisi	Gözlemin tarih ve saat bilgisi
Sıcaklık	Nümerik	Sıcaklık bilgisi
Nem	Nümerik	Nem bilgisi
Operatör	Kategorik	Ölçümü yapan operatör
Ölçüm 1	Nümerik	Ölçüm 1 değeri
Ölçüm 2	Kategorik	Ölçüm 2 tipi
Ölçüm 3	Kategorik	Ölçüm 3 tipi
Ölçüm 4	Nümerik	Ölçüm 4 değeri
Ölçüm 5	Nümerik	Ölçüm 5 değeri
Ölçüm 6	Nümerik	Ölçüm 6 değeri
Ölçüm 7	Nümerik	Ölçüm 7 değeri
Ölçüm 8	Nümerik	Ölçüm 8 değeri
Ölçüm 9	Nümerik	Ölçüm 9 değeri
Ölçüm 10	Nümerik	Ölçüm 10 değeri
Ölçüm 11	Nümerik	Ölçüm 11 değeri
Ölçüm 12	Nümerik	Ölçüm 12 değeri
Ölçüm 13	Nümerik	Ölçüm 13 değeri
Ölçüm 14	Nümerik	Ölçüm 14 değeri
Ölçüm 15	Nümerik	Ölçüm 15 değeri
Son arızadan beri geçen süre	Nümerik	Son arıza oluşumundan örnek alınma saatine kadar geçen süre
Arıza	Kategorik	Arıza durumu (İkili etiket – "No": Makine arızası yok, "Yes": Makine arızası var.)

Veri seti 2 içerisinde yer alan özniteliklerden “Operatör”, “Ölçüm 2” ve “Ölçüm 3” kategorik değişkenlerdir. Bu değişkenlere veri ön işleme sırasında one hot encoding işlemi uygulanarak sayılara dönüştürülmesi sağlanmıştır. “Arıza” ise veri setinde etiketi ifade etmektedir. Bu özellik için belirtilen değer eğer “No” ise bu durum makine arızasının olmadığını, eğer “Yes” ise makine arızası olduğunu gösterir. “Tarih” sütunu ise tarih ve saat bilgisini içermektedir, modellere eğitim için verilecek veri kümesinden çıkartılmıştır.

Çizelge 5.4’te ise ilgili veri setinden rastgele seçilmiş birkaç gözleme yer verilmiştir.

Çizelge 5.4 : Veri Seti 2’ye ait bazı rastgele seçilmiş gözlem örnekleri.

Sıcaklık	Nem	Operatör	Ölçüm 1	Ölçüm 2	Ölçüm 3	Ölçüm n	Ölçüm 15	Geçen Süre	Arıza
67	82	Operator1	291	1	1	…	1842	90	No
67	75	Operator2	298	2	2	…	1301	114	No
60	78	Operator4	317	3	2	…	551	198	No
63	84	Operator8	199	2	1	…	1146	154	No
71	65	Operator3	576	0	0	…	1821	367	Yes
67	76	Operator7	177	3	0	…	1012	7	Yes
69	72	Operator2	1550	1	1	…	1693	165	Yes
75	77	Operator6	174	0	1	…	1645	29	Yes

Veri seti 2 içerisindeki 8.784 adet gözlemden sadece 81 tanesi makinede arızanın olduğu durumlara ait gözlemlerdir. Geri kalan 8.703 tanesi ise arıza olmayan durumu temsil etmektedir. Bu durumda ilgili veri seti sınıf dağılımları bakımından dengesiz bir veri setidir. Dengesizlik oranı 107.44’e tekabül etmektedir. Veri seti 1’e kıyasla dengesizlik ölçütü daha fazladır.

5.3 Dengesiz Veri Seti Problemi ve Dengesiz Veri Setleri ile Başa Çıkma Yöntemleri

Sınıflandırma yapılacak olan bir veri seti içerisinde yer alan gözlemlerin ait oldukları sınıfların dağılımları her zaman dengeli bir şekilde olmaz. Örneğin, kredi kartı sahtekarlığı tespiti, ağ saldırıları tespiti, istenmeyen e-posta tespiti, tıbbi vaka teşhisi, makine arızalarının teşhisi gibi durumlara ilişkin gerçek hayat uygulamalarına dair veri setlerinde dengesiz bir sınıf dağılımı ile karşılaşılması çok olasıdır. Bu karakteristiğe sahip veri setlerinde gözlemlerin büyük bir kısmı bir sınıfa ait iken, az bir kısmı diğer sınıfa ait olmaktadır. Verilerin ait olduğu sınıfın büyük çoğunlukta olduğu kısmına “çoğunluk (majortiy) sınıfı”, gözlem sayısı çok daha az olan sınıfa ise “azınlık (minority) sınıfı” denir [91].

Bir veri seti içerisinde yer alan sınıfların dağılımlarının birbirine yakın olmadığı durumlarda dengesiz veri seti problemi ile karşılaşılır. Dengesiz veri problemi, veri bilimi uygulamalarında dikkat edilmesi gereken kritik bir husustur. Çünkü yapay zekâ algoritmaların birçoğu veri setlerindeki sınıf dağılımlarının dengeli bir şekilde olduğunu varsayarak modelin doğruluk oranını maksimize etmeye çalışır.

Çalışma kapsamında ele alınan veri setlerinin yüksek dengesizlik oranına sahip olduğu daha önceden belirtilmiştir. Çizelge 5.5'te "Veri Seti 1" ve "Veri Seti 2"'nin özeti mevcuttur.

Çizelge 5.5 : Veri Seti 1 ve Veri Seti 2 özet bilgileri.

Veri Seti	Toplam Gözlem Sayısı	Çoğunluk Sınıfı Gözlem Sayısı	Azınlık Sınıfı Gözlem Sayısı	Dengesizlik Oranı (IR)
Veri Seti 1	10.000	9.661	339	28,50
Veri Seti 2	8.784	8.703	81	107,44

Çizelge 5.5'te de görüldüğü üzere her iki veri setindeki sınıf dağılımlarında arıza olmama durumu arıza durumuna göre daha dominanttır. Özellikle Veri Seti 2'ye ait dengesizlik oranı Veri Seti 1'e kıyasla çok daha fazladır. Her iki veri setindeki bilgiler eğer bu haliyle makine öğrenmesi ve derin öğrenmesi algoritmalarına eğitim için girdi olarak verilirse, arıza olma durumunu temsil eden veriler çok az sayıda olduğundan dolayı model sadece arıza olmama durumunu öğrenmeye yatkın olacaktır. Bu ise karşımıza doğruluk paradoksu olarak çıkacaktır. Yani model tahminlerinin tamamında arıza durumunu belirten etiketlerin hiçbirini bilemeyip tahminlerin tümüne "Arıza Yok" şeklinde bir tespit yapsa bile Veri Seti 1 için yaklaşık %97, Veri Seti 2 için yaklaşık %99 oranında doğruluk elde edilecektir. İstatistiksel olarak arzulanana çok yakın değerler olarak görünse de, diğer yandan bizim için önemli olan arıza olma durumlarının hiçbirini tespit edemediği için modellerin gerçek hayata uygulanabilirliği mümkün olmayacaktır. Dengesiz veri setlerinde değerlendirme metriği olarak tek başına "Doğruluk (Accuracy)" parametresi doğru ve yeterli bir ölçüt değildir.

Dengesiz veri setlerinde bu sorunun önüne geçilebilmesi maksadıyla muhtelif yöntemler mevcuttur. Yanlış sınıflandırma maliyetlerinin değerlendirildiği maliyet duyarlı çözümler olsa da bu çalışma da dengesiz sınıf dağılımı problemine veri düzeyinde çözüm yolları denenmiştir. Veri düzeyindeki çözümler dengesiz sınıf dağılımına sahip veri setlerindeki sınıf dağılımlarını ayarlamaya ve dengesizlik

oranını 1'e yaklaştırmaya yönelik çeşitli yaklaşımlardır. Bunlar 3 başlık altında toplanabilir:

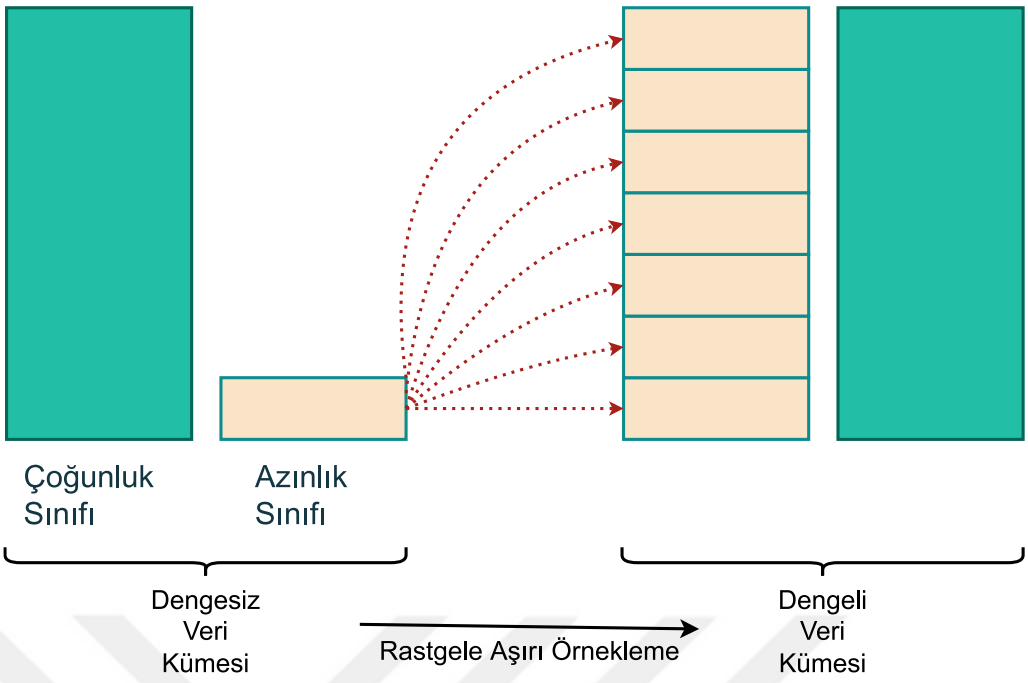
1) Aşırı Örnekleme (Oversampling)
2) Eksik Örnekleme (Undersampling)
3) Aşırı örnekleme ve eksik örnekleme kombinasyonları

Dengesiz sınıf dağılımına sahip veri setlerinde, çoğunluk sınıfına ait gözlem sayısını azınlık sınıfındaki gözlem sayısı miktarına yaklaştırmak ya da eşitlemek için çoğunluk sınıfına ait bazı gözlemlerin çeşitli yöntemler ile seçilerek eğitim verisi dışında bırakılmasına eksik örnekleme ya da aşağı örnekleme denir. Ancak dengesizlik oranı yüksek olan veri setlerinde eksik örnekleme yöntemlerini tercih edilmesi durumunda çok fazla veri kaybı meydana gelecektir. Bu durumda model eğitimini ciddi seviyede etkileyebilecek birçok gözlemin veri seti dışında kalma ihtimali çok yüksektir. Aşağı örnekleme yöntemleri kullanıldığında Veri Seti 1 için toplam verinin yaklaşık %94'ü, Veri Seti 2 için tüm verinin yaklaşık %98'i model eğitim verisinin dışında bırakılacaktır. Geri kalan örnekler ise özellikle yüksek miktarlarda veriye ihtiyaç duyulan derin öğrenme algoritmaları için son derece yetersizdir. Bu sebeple bu tez çalışması kapsamında sadece aşırı örnekleme ve aşırı örnekleme ile eksik örneklemenin kombinasyonlarının olduğu yaklaşımlar ele alınmıştır.

5.3.1 Aşırı örnekleme yöntemleri (Oversampling)

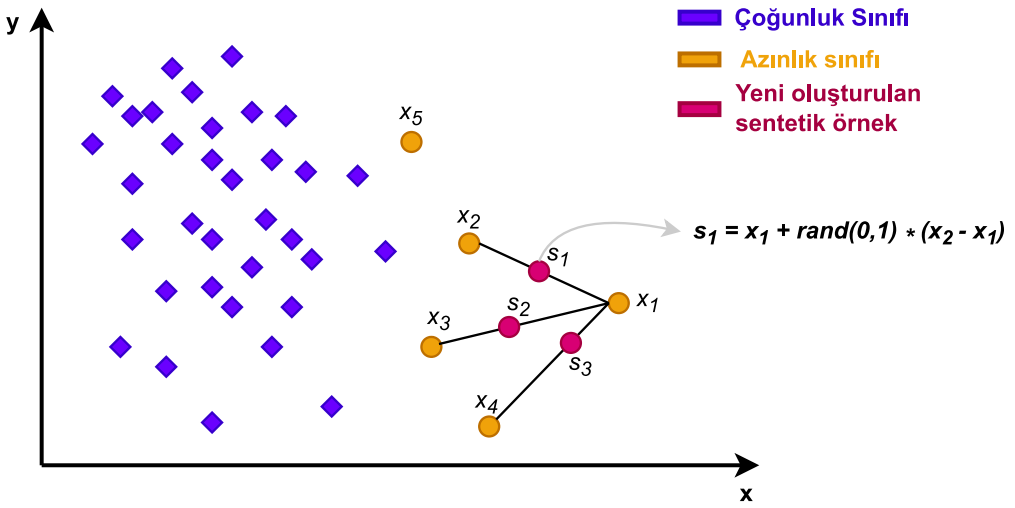
Aşırı örnekleme yaklaşımlarında azınlık sınıfına ait gözlemlerin sayısı arttırılarak çoğunluk sınıfındaki gözlem sayısına yaklaştırılmaktadır ve bu sayede dengesizlik oranı 1'e yakınsamaktadır.

Rastgele aşırı örnekleme: Veri kümesinde yer alan azınlık sınıfına ait örneklerden rastgele bazı örnekleri basit bir şekilde kopyalayıp çoğaltarak, çoğunluk sınıfındaki örnek sayısına eşit miktarda yeni bir veri kümesi oluşturur. Ancak bu yeni veri kümesi çok sayıda tekrarlayan veri içerdiği için model eğitimlerinde aşırı uyum (overfitting) sorunu ile karşılaşılabilir. Rastgele aşırı örnekleme Şekil 5.1'de gösterilmiştir.



Şekil 5.1 : Rastgele aşırı örnekleme.

Sentetik azınlık aşırı örnekleme tekniği (Synthetic minority over-sampling technique - SMOTE): Bu örnekleme sıklıkla kullanılan aşırı örnekleme yaklaşımlarından biridir. Ancak bu yöntem, rastgele seçilen örnekler yerine, veri kümesinde zaten bulunan azınlık sınıfı örnekleri arasındaki k adet komşunun uzaklıkları kullanarak yeni sentetik veriler üretir [92]. SMOTE yönteminin avantajı, örneklerin çoğaltılması yerine sentetik örnekler üretildiği için rastgele aşırı örnekleme yönteminde karşılaşılan aşırı uyum sorunundan kaçınılmasıdır. Şekil 5.2'de SMOTE yöntemi gösterilmiştir.



Şekil 5.2 : SMOTE yöntemi [93].

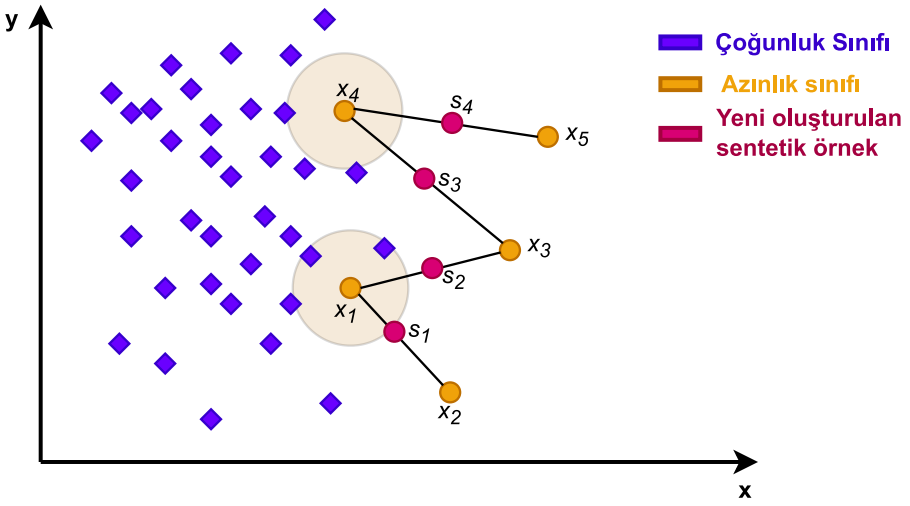
SMOTE yöntemiyle sentetik örnekler aşağıdaki adımları izleyerek oluşturulur:

- Azınlık sınıfından baz alınmak üzere rastgele bir örnek belirlenir.
- Belirlenen baz örnek için k adet en yakın komşu tespit edilir. (Örneğin k =3)
- Bu komşular arasındaki uzaklık farkı alınır ve 0 ile 1 arasında rastgele bir sayı ile çarpılır.
- Çıkan sonuç daha önceden rastgele seçilmiş olan örnek vektörüne eklenerek yeni örnek oluşturulur.

Şekil 5.2'de görüldüğü üzere seçilen x_1 örneği için 3 adet en yakın komşusu (x_2, x_3 ve x_4) arasındaki mesafeler baz alınarak yeni sentetik veri olarak s_1, s_2 ve s_3 örnekleri oluşturulmuştur. [93]

Adaptif sentetik örnekleme (ADASYN): ADASYN yöntemi 2008 yılında Haibo He ve diğerleri tarafından tanıtılmış olan ve sentetik veri üretmek için SMOTE tekniğine dayanan bir çeşit aşırı örnekleme tekniğidir [94]. ADASYN yönteminin SMOTE yönteminden farkı, azınlık sınıfının daha düşük yoğunlukta olduğu bölgelerde daha fazla sayıda sentetik veri üretebilmek için, çoğunluk sınıfının hâkim olduğu alanlarda yer alan azınlık sınıfı örneklerini saptayan bir yöntem uygulamasıdır. Sentetik veri üretmek için baz alınan azınlık sınıfı örneğinin k-en yakın komşu alanı içerisinde ne kadar fazla çoğunluk sınıfından örnek varsa o kadar o kadar fazla sentetik veri üretilir. Eğer azınlık sınıfına ait bir gözlemin k- en yakın komşu alanında çoğunluk sınıfına ait hiçbir örnek yoksa o gözlem için sentetik veri üretilmez [95].

Şekil 5.3'te görüldüğü üzere öncelikle azınlık sınıfının yoğunluğunun az olduğu bölgedeki azınlık sınıfı gözlemleri tespit ediliyor. Ve bu gözlemlerin k-en yakın komşu alanındaki çoğunluk sınıfı örneklerinin sayısına bağlı olarak aldığı ağırlıklandırma değerince sentetik veri üretiliyor.

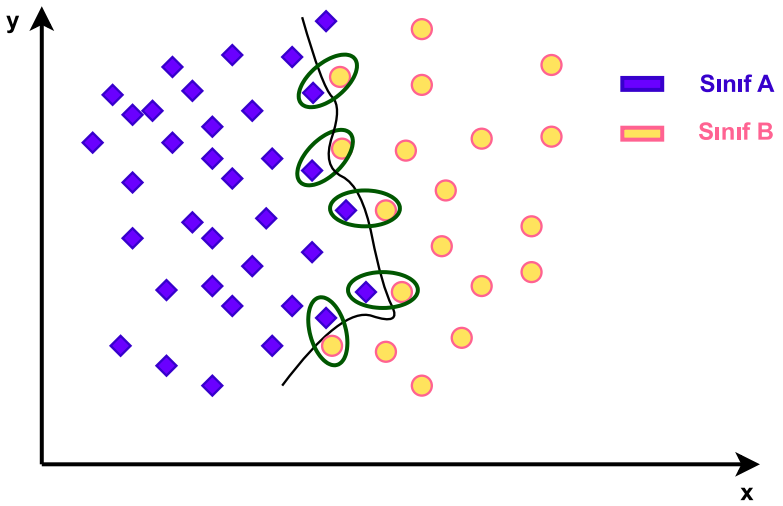


Şekil 5.3 : ADASYN yöntemi [93].

5.3.2 Aşırı örnekleme ve eksik örnekleme yöntemlerinin kombinasyonları

Aşırı örnekleme ve eksik örnekleme kombinasyonlarının uygulandığı hibrit yaklaşımlardan en bilinenleri olan SMOTE-Tomek ve SMOTE-ENN yöntemlerine yer verilmiştir. Ancak hibrit örnekleme yaklaşımları sadece bu iki yöntemle sınırlı değildir.

Sentetik azınlık aşırı örnekleme-Tomek linkleri (SMOTE-Tomek): Sınıf dağılımları dengeli olmayan veri kümelerinin dengeli hale getirmenin bir diğer yolu da sentetik azınlık aşırı örnekleme tekniği ile Tomek linklerinin birlikte kullanıldığı SMOTE-Tomek yöntemidir. Tomek linkleri çoğunluk ve azınlık sınıfı olmak üzere birbirine en yakın zıt sınıf örnek çiftlerinden oluşur. (Şekil 5.4)

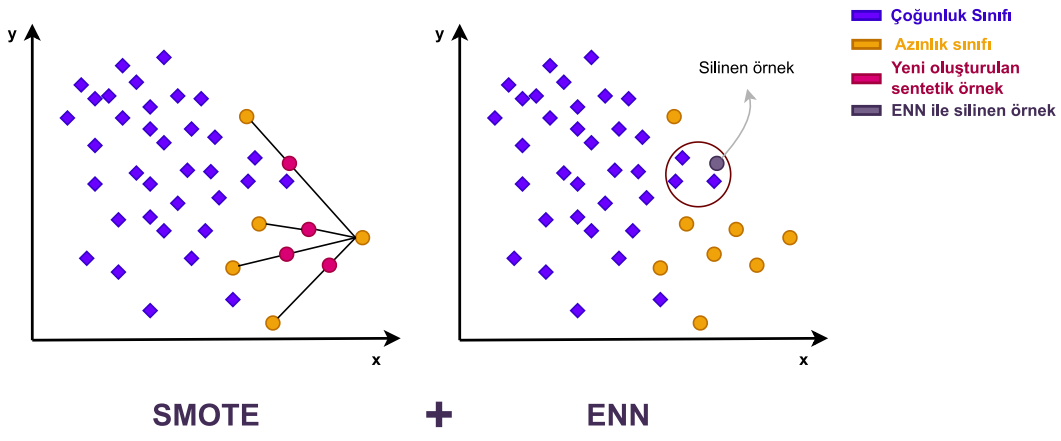


Şekil 5.4 : Tomek linkleri [96].

Tomek linkleri Şekil 5.4'te de görüldüğü gibi iki sınıf ayrımının çok net olmadığı ve sınıflandırma algoritmaları için en çok sorun çıkaracak olan örneklerdir. Bu sebeple Tomek linkleri yaklaşımında birbirine en yakın ama zıt sınıfa ait örneklerin her ikisi ya da çoğunluk sınıfına ait olanlar tespit edilir ve çıkartılır. Böylece iki farklı sınıf arasındaki karar sınırı daha da belirginleştirilmiş olur ve algoritmaların daha güvenli bir şekilde sınıflandırma yapmasına imkân verilir. Bu yaklaşım bir örnek eksiltme uygulamasıdır.

SMOTE-Tomek hibrit yaklaşımında ise öncelikle azınlık sınıfına ait olan örneklerin sayısını çoğunluk sınıfına eşitlemek için SMOTE yöntemi ile sentetik veriler üretilir. Daha sonra iki farklı sınıf arasındaki ayrımı belirginleştirmek adına Tomek linkleri yöntemi uygulanır [97].

Sentetik azınlık aşırı örnekleme - Düzenlenmiş en yakın komşu (SMOTE-ENN): Aşırı örnekleme ve örnek eksiltme yöntemlerinin bir arada kullanıldığı hibrit yöntemlerden biri 2004 yılında Batista ve diğerleri tarafından önerilen SMOTE-ENN yöntemidir. Düzenlenmiş en yakın komşu yönteminde Şekil 5.5'te olduğu gibi, k adet komşu miktarınca (k parametresi belirlenmemişse 3 olarak kabul edilir) gözlem arasından çoğunluğun ait olduğu sınıftan farklı olan gözlemler silinir. Bu yöntem veri seti içerisinde sınıflandırma algoritmalarının performansını olumsuz etkileyebilecek gürültü verilerinin ortadan kaldırılması için etkili bir örnek eksiltme yöntemidir.



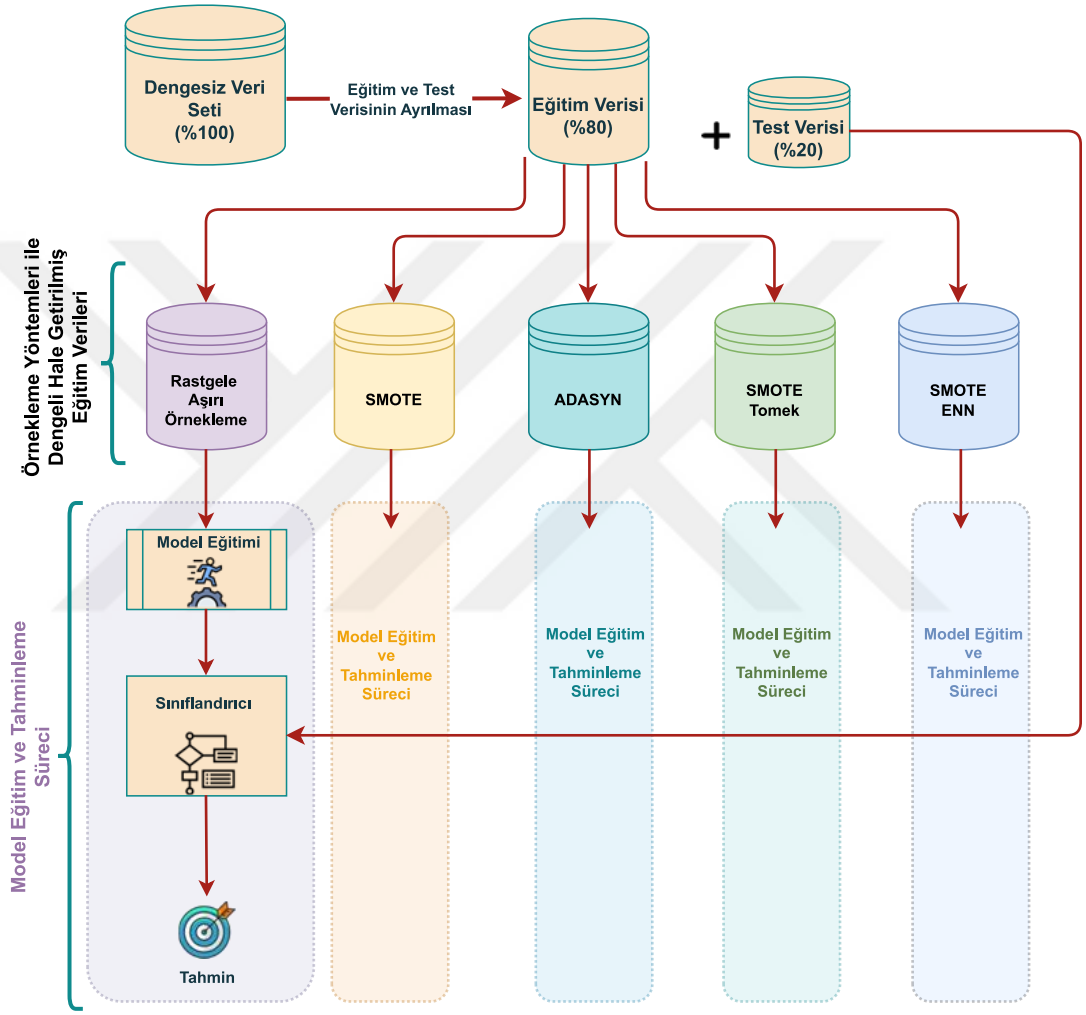
Şekil 5.5 : SMOTE ENN [98].

SMOTE-ENN hibrit yaklaşımında tıpkı SMOTE-Tomek'te olduğu gibi öncelikle azınlık sınıfına ait olan örneklerin sayısını çoğunluk sınıfına eşitlemek için SMOTE

yöntemi ile sentetik veriler üretilir. Daha sonra gürültü oluşturacak verilerin kaldırılması için örnek eksiltme yöntemlerinden ENN yöntemi uygulanır [99].

5.4 Deneysel Tasarımın Açıklanması

Ele alınan çalışma kapsamında bir veri seti üzerinde yapılan sınıflandırma işleminin aşamaları Şekil 5.6'da gösterilmiştir.



Şekil 5.6 : Bir makine öğrenmesi algoritmasının 5 farklı örnekleme yöntemi ile denenmesi.

Deney tasarım aşamaları şu şekildedir:

1) Veri setlerinin her ikisinde de eksik ya da anlamsız veri sorunu yoktur. Bu sebeple eksik veri tamamlama ya da bazı verileri silme gibi işlemler uygulanmamıştır.

2) Veri sızıntısı olgusunu engellemek amacıyla, ilk iş olarak eğitim ve test verileri sırasıyla %80 ve %20 oranında ayrılmıştır. %20'lik test verisine hiçbir şekilde dokunulmamıştır. Veri seti, eğitim ve test verileri olarak ayrılırken katmanlama (stratification) yöntemi kullanılarak, sınıflandırma etiket yüzdelerinin hem eğitim verilerinde hem de test verilerinde aynı oranda kalması sağlanmıştır. Böylece eğitim ve test verileri ayrılırken, herhangi bir veri kümesinde azınlık sınıfına ait gözlemlerin yeteri kadar yer almaması ya da hiç yer almamasının önüne geçilmiştir. Örneğin eğitim verilerinin %3'ü arıza olma durumunu ifade ederken, test verilerinin de %3'ünün arıza olma durumunu ifade edecek şekilde dağılıma sahip olması sağlanmıştır.

3) Bu aşama itibariyle, kullanılan veri setlerinin karakteristikleri sebebiyle elde dengesiz sınıf dağılımına sahip eğitim ve test veriler mevcuttur. Model eğitiminde kullanılacak olan eğitim verileri, modellerin aşırı uyuma meyletmesini engellemek amacıyla Şekil 5.6'da görüldüğü gibi rastgele aşırı örnekleme, SMOTE, ADASYN, SMOTE Tomek ve SMOTE ENN olmak üzere 5 farklı örnekleme yöntemiyle dengeli hale getirilmiştir. Dengesiz veri seti problemi ile başa çıkabilmek için uygulanan bu örnekleme işlemleri sadece eğitim verisine uygulanmıştır. Makine arızasını bulmaya yönelik yapılan bu kestirimci bakım uygulamasını gerçek hayat olgusundan koparmamak adına test verilerinde hiçbir örnekleme işlemi yapılmamıştır. Çünkü her ne kadar modellerin eğitilmesinde aşırı öğrenmenin önüne geçmek için eğitim verileri dengeli hale getirilse de, gerçek hayatta endüstriyel ortamdan toplanan anlık veriler dengesiz halde gelmeye devam edecektir.

4) Deney tasarımında her bir veri seti için 5 çeşit örnekleme yöntemi ile 14 farklı makine öğrenmesi algoritması denenmiştir. Burada karşılaştırması yapılacak olan 14 farklı makine öğrenmesi yönteminden her birinin farklı gözlemleri içeren veri setleri ile eğitilmesi ve test edilmesi, model performanslarının objektif bir şekilde karşılaştırılması açısından uygun olmayacaktır. Tüm modellerin aynı gözlemlere sahip veri kümeleri ile eğitilip sınıflandırma yapabilmesi amacıyla, dengeli hale getirilen her bir eğitim seti, öncelikle bilgisayarda farklı dosyalar halinde kaydedilmiştir. (Örneğin; SMOTE eğitim verisi, SMOTE test verisi, ADASYN eğitim verisi, ADASYN test verisi vb.) Her bir farklı makine öğrenmesi algoritması için,

dengeli hale getirilip kayıt altına alınmış olan eğitim verileri girdi olarak verilmiştir. Aynı şekilde en başta ayrılmış ve hiçbir işlem yapılmamış olan test verisi gözlemleri tüm modeller için aynı tutulmuştur. Çizelge 5.6 ve Çizelge 5.7'de her bir veri setindeki azınlık sınıfı ve çoğunluk sınıfına ait gözlemlerin örnekleme neticesindeki miktarlarına yer verilmiştir.

5) Klasik makine öğrenmesi algoritmalarında modeli en iyilemek için rastgele arama (random search) yöntemi ile yapılarak parametre optimizasyonu gerçekleştirilmiştir. Derin öğrenme yöntemlerinde de aynı şekilde rastgele arama yöntemleri ile optimizasyon yapılmıştır. Ancak derin öğrenme modellerinin eğitim süresinin uzun olması sebebiyle hiper parametre optimizasyon uzayı görece sınırlı tutulmuştur. Rastgele arama yöntemi ile yapılan parametre optimizasyonu neticesinde her bir model için elde edilen parametreler Ek A'da verilmiştir. Ek A'da yer almayan parametreler varsayılan değerleri ile kullanılmıştır.

6) 14 adet farklı makine öğrenmesi yönteminin her biri ile aynı test verisi üzerinde sınıflandırma yapılmıştır ve modellerin tahmin performansları bu tezin 6 numaralı başlığı altında belirtilen metrikler ile ölçülerek kıyaslanmıştır.

7) Yukarıdaki 6 aşamada bahsedilen deney tasarımı, 2 farklı veri seti için tekrarlanmıştır.

Aşağıda Veri Seti 1 ve Veri Seti 2'ye ait gözlemlerin örnekleme işleminden önce ve örnekleme işlemlerinden sonraki adetleri yer almaktadır.

Çizelge 5.6 : Veri Seti 1 için aşırı örnekleme öncesi ve sonrasında gözlem sayıları.

	EĞİTİM VERİSİ			**TEST VERİSİ**		
	Çoğunluk Sınıfı Gözlem Sayısı	Azınlık Sınıfı Gözlem Sayısı	Dengesizlik Oranı (IR)	Çoğunluk Sınıfı Gözlem Sayısı	Azınlık Sınıfı Gözlem Sayısı	Dengesizlik Oranı (IR)
Örnekleme Öncesi	7729	271	28,520	1932	68	28,412
Rastgele Aşırı Örnekleme	7729	7729	1,000	1932	68	28,412
SMOTE	7729	7729	1,000	1932	68	28,412
ADASYN	7729	7671	1,008	1932	68	28,412
SMOTE Tomek	7664	7664	1,000	1932	68	28,412
SMOTE ENN	6665	7322	0,910	1932	68	28,412

Çizelge 5.6'da görüldüğü üzere örnekleme öncesi dengesiz dağılıma sahip Veri Seti 1'e ait gözlemler, çeşitli aşırı örnekleme yöntemleri ile dengeli hale getirilmiştir. Örneklemeler neticesinde eğitim için kullanılacak olan veri kümelerinin dengesizlik oranları 1'e yakınsamıştır. Ancak makine öğrenmesi modellerinin performansları, dengesiz sınıf dağılımına sahip test verileri ile yapılan sınıflandırma tahminlerine göre ölçülmüştür.

Çizelge 5.7 : Veri Seti 2 için aşırı örnekleme öncesi ve sonrasında gözlem sayıları.

	EĞİTİM VERİSİ			**TEST VERİSİ**		
	Çoğunluk Sınıfı Gözlem Sayısı	Azınlık Sınıfı Gözlem Sayısı	Dengesizlik Oranı (IR)	Çoğunluk Sınıfı Gözlem Sayısı	Azınlık Sınıfı Gözlem Sayısı	Dengesizlik Oranı (IR)
Örnekleme Öncesi	6962	65	107,108	1741	16	108,813
Rastgele Aşırı Örnekleme	6962	6962	1,000	1741	16	108,813
SMOTE	6962	6962	1,000	1741	16	108,813
ADASYN	6962	6975	0,998	1741	16	108,813
SMOTE Tomek	6962	6962	1,000	1741	16	108,813
SMOTE ENN	6542	6962	0,940	1741	16	108,813

Çizelge 5.7'de görüldüğü üzere örnekleme öncesi dengesiz dağılıma sahip Veri Seti 2'ye ait gözlemler, çeşitli aşırı örnekleme yöntemleri ile dengeli hale getirilmiştir. Örneklemeler neticesinde eğitim için kullanılacak olan veri kümelerinin dengesizlik oranları 1'e yakınsamıştır. Test verilerinin dengesizlik durumu ise olduğu gibi bırakılmış ve makine öğrenmesi modellerinin performansları bu şekilde ölçümlenmiştir.

6. PERFORMANSLARIN ÖLÇÜTLERİ VE DENEYSEL SONUÇLAR

6.1 Performans Ölçütleri

Yapay zekâ algoritmaları ile oluşturulmuş modellerin performanslarının uygun açıdan değerlendirilebilmesi için doğru metriklerin seçilmesi önemlidir.

Sınıflandırma problemlerinde performans ölçütlerinin temelinde karışıklık matrisi yer alır. Karışıklık matrisi, "m" sınıf sayısını belirtmek üzere, gerçek sınıf etiketlerinin ve tahmin edilen sınıf etiketlerinin kombinasyonlarını içeren m x m boyutunda bir çeşit tablodur. Bu çalışmada ele alınan problem ikili sınıflandırma görevi olduğu için karışıklık matrisi 2x2 boyutundadır. Karışıklık matrisi Şekil 6.1'de yer alan kombinasyonlardan oluşur.

		Sınıflandırma Tahmini	
		Pozitif	**Negatif**
Gerçek Durum	**Pozitif**	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Negatif	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Şekil 6.1 : Karışıklık matrisi.

Karışıklık matrisinde 4 temel terminoloji ile karşılaşılır:

Doğru pozitif (DP): Gerçek değer pozitif iken tahmin edilen değer de pozitiftir.

Doğru negatif (DN): Gerçek değer negatif iken tahmin edilen değer de negatiftir.

Yanlış pozitif (YP): Gerçek değer negatif iken tahmin edilen değer pozitiftir. (Hatalı Tahmin)

Yanlış negatif (YN): Gerçek değer pozitif iken tahmin edilen değer negatiftir. (Hatalı Tahmin)

Karışıklık matrisinden türemiş olan çeşitli metrikler aşağıda yer almaktadır:

Doğruluk (Accuracy): Doğru olarak sınıflandırılan örneklerin tüm örneklere oranını ifade eden metriktir. Bir sınıflandırma modelinin değerlendirilmesinde en temel metrik olsa da tek başına kullanılması yeterli değildir. Hatta bu çalışmada ele alınan veri setlerinde olduğu gibi dengesiz bir sınıf dağılımına sahip verilerdeki sınıflandırmalarda tek başına performans değerlendirme metriği olarak kullanılması yanıltıcı olacaktır.

$$Doğruluk = \frac{DP+DN}{DP+YP+DN+YN} \tag{6.1}$$

Kesinlik (Precision): Pozitif olarak tahmin edilen örneklerin gerçekte ne kadarının pozitif olduğunun ölçütüdür. Kesinlik değerinin maksimize edilmesi için yanlış pozitif değerinin minimum olması beklenir.

$$Kesinlik = \frac{DP}{DP+YP} \tag{6.2}$$

Duyarlılık (Recall): Gerçekte pozitif örneklerin ne kadarının doğru olarak tahmin edildiğini ifade eden metriktir. Duyarlılık değerinin maksimum olabilmesi için yanlış negatif değerlerinin minimum olması beklenir.

$$Duyarlılık = \frac{DP}{DP+YN} \tag{6.3}$$

Dolandırıcılık tespiti, hastalık teşhisi, arıza tespiti gibi vakalarda anomali durumunu doğru olarak tahmin edememenin maliyeti, anomali olmayan durumu yanlış tahmin etmekten daha önemlidir. Örneğin, gerçek bir dolandırıcılık girişimi (Fraud tespiti), olağan bir işlem olarak sınıflandırıldığında, bunun kişi ve kurumlara ciddi seviyelerde maddi zararlar verebileceği açıktır. Bu örnekten yola çıkılacak olunursa; duyarlılık, gerçekte dolandırıcılık girişimi olan hareketlerin kaç tanesinin doğru bilindiğinin ölçütüdür.

F1 skoru (F1 score): Bir modelin performansını daha etkin bir şekilde değerlendirebilmek için duyarlılık ve kesinlik ölçütlerinin birlikte kullanıldığı bir metriktir. Duyarlılık ve kesinlik metriklerinin harmonik ortalamalarının bulunması ise hesaplanır. Dengesiz veri setlerinin değerlendirilmesi açısından iyi bir metriktir.

$$F1\ Skoru = 2 * \frac{Kesinlik * Duyarlılık}{Kesinlik + Duyarlılık} \quad (6.4)$$

Cohen Kappa skoru (Cohen Kappa score): Gözlenen ve beklenen değerler arasındaki uyuşmanın güvenirliğinin ifade eden bir metriktir [100].

Gözlenen değer (Po): Modelin çıktısı (tahmin)

Beklenen değer ($Pc - Pe$): Beklenen çıktı (gerçek)

Cohen Kappa skoru, Po ve Pc olasılıkları kullanılarak Denklem 6.5'teki gibi hesaplanır.

$$\kappa = \frac{P_o - P_c}{1 - P_c} \quad (6.5)$$

Kappa skoru -1 ile +1 arasında değer alabilir. Kappa skorunun pozitif bir değer alması sınıflandırıcılar arasındaki uyumun şans eseri beklenen uyumdan daha fazla olduğunu, negatif bir değer alması ise şans eseri beklenen uyumdan daha az olduğunu gösterir [95]. Bu değer 1'e yakınsadıkça uyum mükemmele doğru yaklaşır. Kappa skoruna göre uyum düzeyi Çizelge 6.1'de gösterilmiştir.

Çizelge 6.1 : Kappa skoru uyum düzeyinin yorumlanması.

Kappa Skoru	**Uyum Düzeyi Yorumu**
0,81-1,00	Mükemmel Uyum
0,61-0,80	Önemli Derecede Uyum
0,41-0,60	İyi Derecede Uyum
0,21-0,40	Orta Derecede Uyum
0-0,20	Kötü Uyum
< 0	Şansa Bağlı Uyum

Kappa skoru ile ilgili hesaplamalar için somut bir örnek düşünelim. Tablo 7'de arıza olup olmama durumunun sınıflandırıldığı ve toplam 10 adet gözlemden oluşan küçük bir veri kümesi üzerindeki her bir gözlem için ait oldukları gerçek sınıfları ve eğitilmiş olan model tarafından tahmin edilen sınıfları gösterilmiştir.

Çizelge 6.2 : Örnek veri kümesi üzerinde gerçek ve tahmin edilen gözlemler.

	(G: Gözlem)	G-1	G-2	G-3	G-4	G-5	G-6	G-7	G-8	G-9	G-10
Arıza Durumu (Var / Yok)	**Gerçek**	Yok	Yok	Yok	Var	Yok	Yok	Yok	Var	Yok	Yok
	Tahmin edilen	Var	Yok	Yok	Var	Yok	Yok	Var	Var	Yok	Yok

Çizelge 6.2'de de görüldüğü üzere gerçek sınıflar ile tahmin edilen sınıflar arasında %100 bir uyum söz konusu değildir. Tablo 8'de uyum gösteren ve göstermeyen gözlemlerin niceliklerine yer verilmiştir.

Çizelge 6.3 : Gerçek sınıf ve tahmin sınıfları arasındaki uyum.

		TAHMİN	
		Arıza Var	**Arıza Yok**
GERÇEK	**Arıza Var**	2	0
	Arıza Yok	2	6

"Arıza var" sınıfı olarak 2 adet gözlemde, "Arıza yok" sınıfı olarak ise toplam 6 adet gözlemde gerçek ve tahmin modellerinin mutabık kaldığını görüyoruz. Bu durumda modelin gerçek duruma uyumu yani $P_o = 8 / 10 = 0{,}8$ olarak bulunur.

Gerçek sınıfların ve tahmin edilen sınıfların şans eseri uyum sağlama olasılıklarını hesaplayacak olursak;

1) Gerçek sınıflar göz önüne alındığında toplam 10 adet gözlem içerisinden 2 tane arıza durumu, 8 tane de arıza olmama durumu mevcuttur. O halde gerçek durumdaki arıza olasılığı 2/10 =0,2'dir. Arıza olmama olasılığı ise 8/10=0,8'dir.

2) Modelin tahmini incelenecek olursa, 10 adet gözlem içerisinden 4 tane arıza durumu, 6 tane arıza olmama durumu mevcuttur. O halde model tahminine göre arıza olma olasılığı 4/10=0,4'tür. Arıza olmama olasılığı ise 6/10=0,6'dır.

3) Hem gerçek durumun hem de model tahmininin arızalı sınıf olasılıkları ise 0,2*0,4 =0,08'dir.

4) Hem gerçek durumun hem de model tahmininin arızalı olmama durumuna ilişkin sınıf olasılıkları ise 0,8*0,6 =0,48'dir.

5) 3 ve 4 numaralı maddelerde yer alan arıza olma ve arıza olmama olasılıkları açısından gerçek durum ve model tahminin rastgele uyuşma olasılıkları Pe = 0,08 + 0,48 = 0,488 olarak bulunur.

6) Po ve Pe olasılık değerlerinin Kappa skoru formülüne uygulayacak olursak;

$$\kappa = \frac{\text{Po-Pe}}{\text{1-Pe}} = \frac{(0.80-0.488)}{(1-0.488)} = 0.312/0.512 \cong 0{,}6094$$

sonucu elde edilir.

6.2 Deneysel Sonuçlar

Bu çalışma kapsamında kestirimci bakım verilerini temsil eden 2 farklı veri seti kullanılmıştır. Her veri seti üzerinde toplam 14 klasik makine öğrenmesi ve derin öğrenme algoritması denenmiştir. Ayrıca bu 14 farklı algoritma, dengesiz sınıf dağılımına sahip veri setlerinin farklı yukarı örnekleme yöntemleri ile dengeli hale getirildiği 5 farklı senaryo altında test edilmiştir.

Deneysel çalışma kapsamında elde edilen yapay zekâ modellerinin başarı performanslarının değerlendirmesinde;

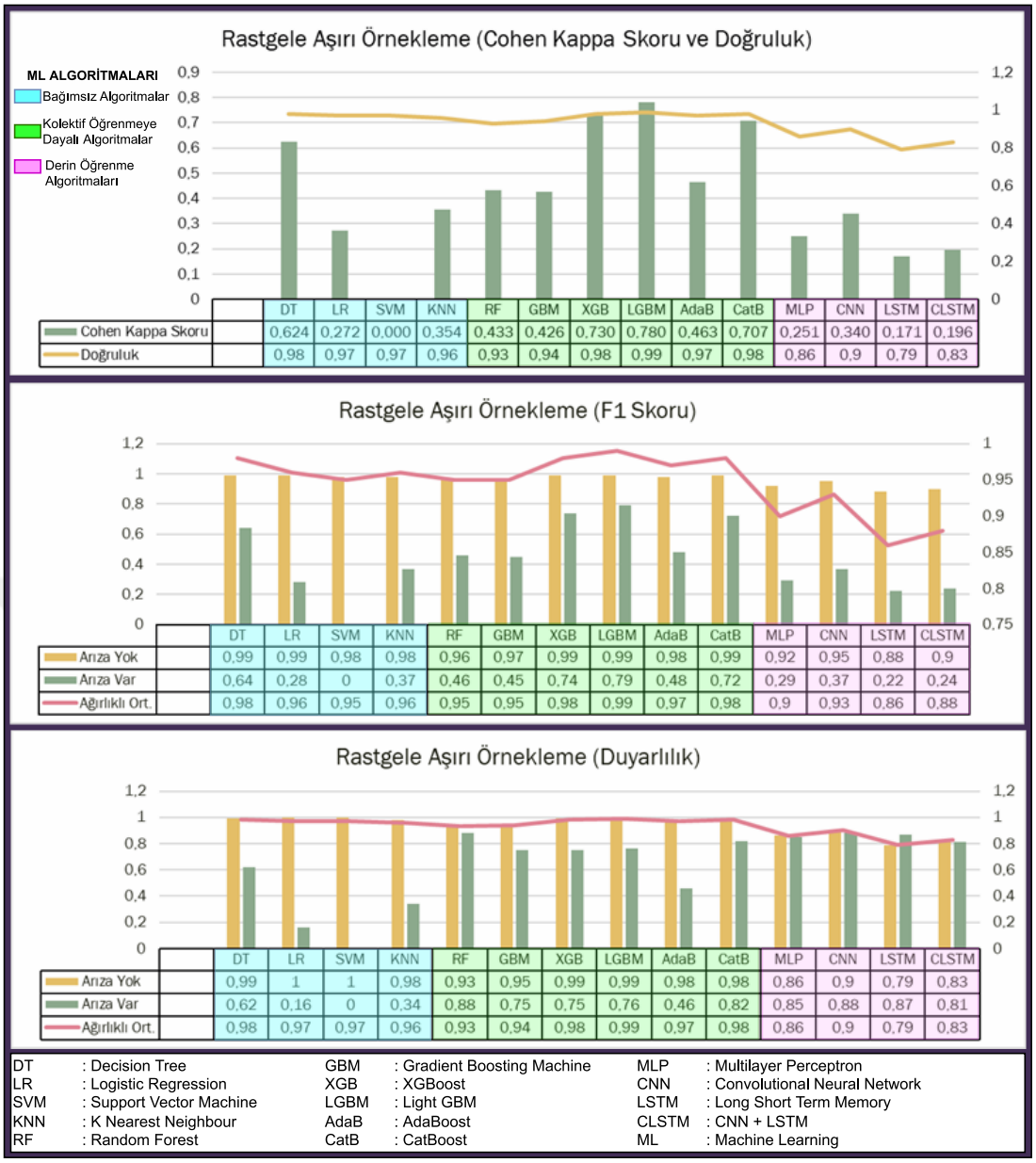
- Doğruluk (Accuracy),
- Duyarlılık (Recall),
- F1 Skoru (F1 Score)
- Cohen Kappa Skoru

metrikleri kullanılmıştır.

Bu bölümde “Veri Seti 1” ve “Veri Seti 2”ye ilişkin sonuçlar sırası ile ortaya konulmuştur.

6.2.1 Veri seti 1’e ait performans sonuçlarının incelenmesi

İlgili veri setinin performans metrikleri öncelikle 5 farklı veri seti dengeleme yöntemi kullanılarak ölçülmüştür. Rastgele aşırı örnekleme yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.2’deki gibidir:



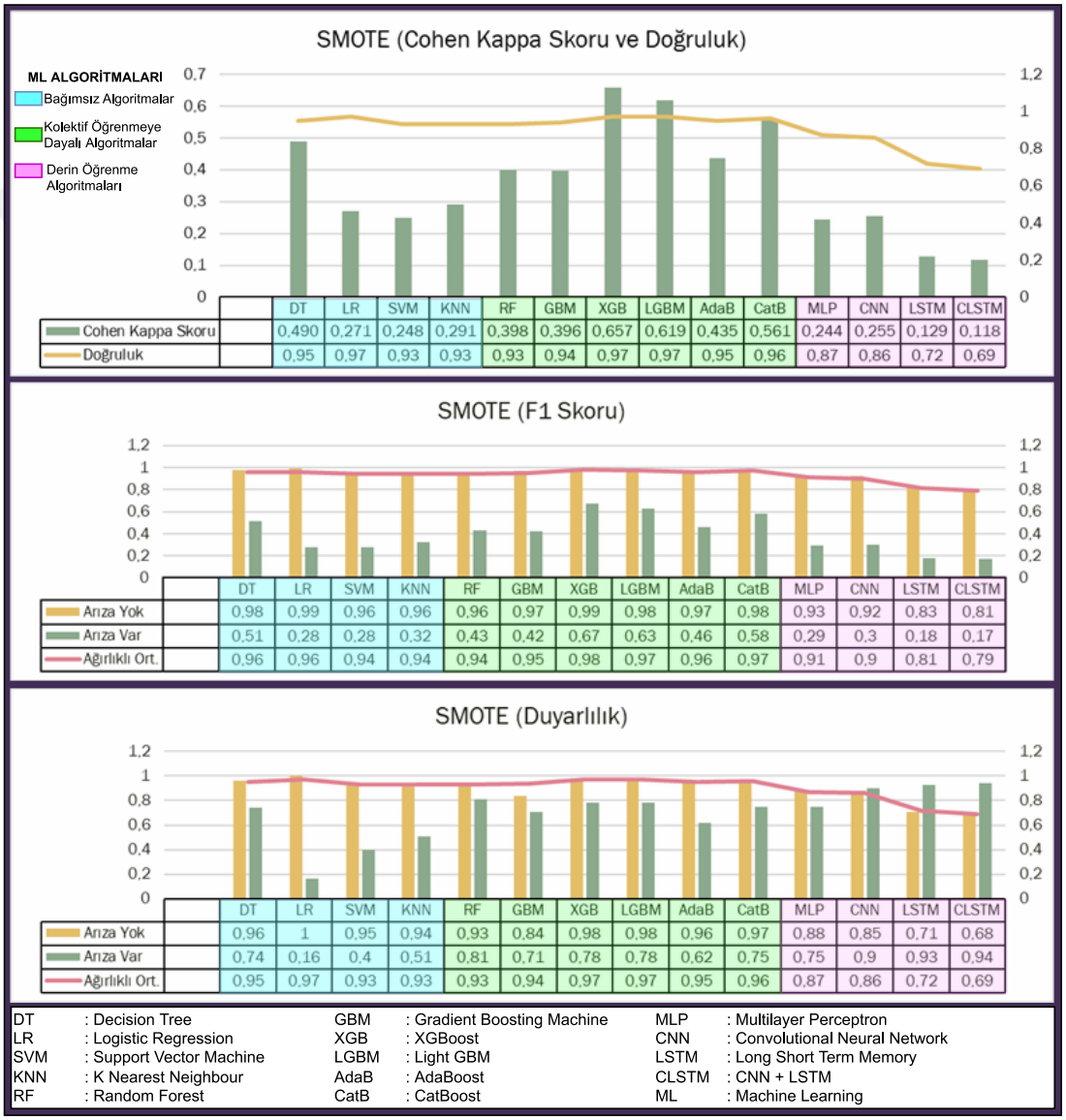
Şekil 6.2 : Rastgele aşırı örnekleme yöntemi kullanıldığında performans metrikleri (Veri Seti 1).

Cohen Kappa skoru açısından 0,780'lik skor ile en yüksek performansı LGBM algoritması göstermiştir. Daha sonra sırasıyla XGBoost algoritması 0,730 ve CatBoost algoritması 0,707 Cohen Kappa skoru ile en yüksek performansa sahip algoritmalar olarak görünmektedir.

F1 skoru incelendiğinde "Arıza Yok" durumu için DT, LR, XGBoost, LGBM ve CatBoost algoritmalarının 0,99 değerine sahip olduğu görülebilir. "Arıza Var" durumu için ise en yüksek F1 skorunun 0,79 ile LGBM algoritmasına ait olduğu görülüyor. Daha sonra en yüksek F1 skoruna sahip algoritmalar ise 0,74 ile LR ve 0,72 ile Catboost algoritmalarıdır.

Duyarlılık metriği göz önüne alındığında “Arıza Yok” durumu için en yüksek skorun LR ve SVM algoritmalarında 1 olduğu görülüyor. Ancak bu iki algoritma aynı zamanda “Arıza var” durumu için de diğer algoritmalara kıyasla en küçük değerler olan 0,16 ve 0 skorlarına sahipler. Bu durum modellerin aşırı uyumlandığını ve arıza tespiti için kullanılamayacağını gösterir

SMOTE yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.3’teki gibidir:



Şekil 6.3 : SMOTE Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 1).

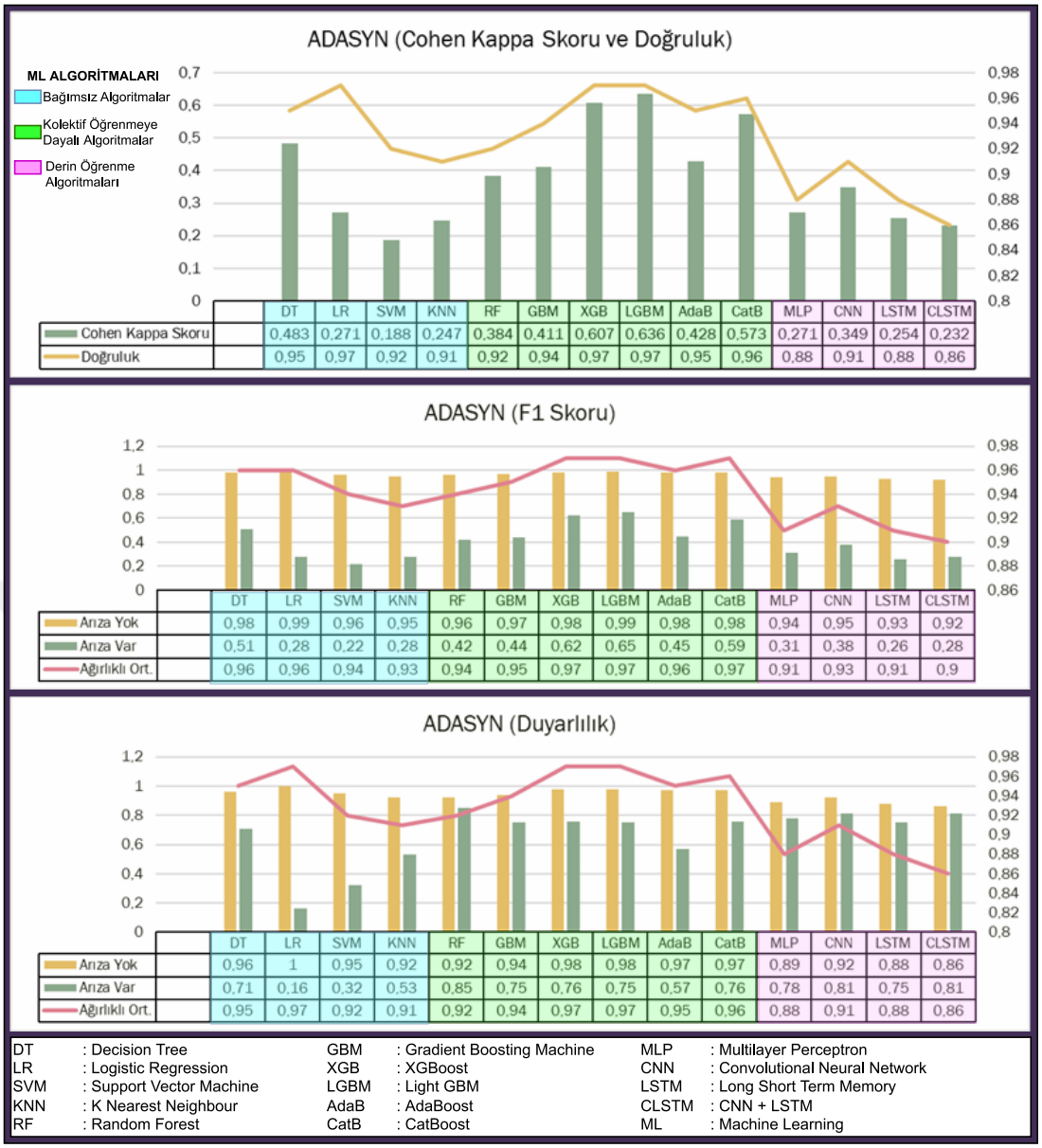
Tüm algoritmalar SMOTE yöntemi kullanılarak eğitildiğinde en yüksek Cohen Kappa skorunun 0,657 olarak XGBoost algoritması tarafından elde edildiği görülmektedir. Daha sonra en yüksek performansa sahip algoritmalar ise sırasıyla 0,619 ve 0,561 Cohen Kappa skorları ile LGBM ve CatBoost algoritmalarıdır. Cohen

Kappa değeri açısından en düşük performans ise derin öğrenme algoritmaları tarafından ortaya konulmuştur.

F1 skoru incelendiğinde "Arıza Yok" durumu için en yüksek skora (0,99) sahip olan LR ve XGBoost algoritmaları ilk sırada yer almaktadır. "Arıza Var" durumu için ise en yüksek F1 skorunun 0,67 ile XGBoost algoritmasına ait olduğu görülüyor. Daha sonra en yüksek F1 skoruna sahip algoritmalar ise 0,63 ile LGBM ve 0,58 ile CatBoost algoritmalarıdır.

Duyarlılık metrikleri karşılaştırıldığında "Arıza Yok" durumu için en yüksek skorun LR algoritmasında 1 olduğu görülüyor. Daha sonra en yüksek duyarlılık değerine sahip olan algoritmalar XGBoost ve LGBM'dir. (0,98) "Arıza yok" durumu için en yüksek duyarlılık skorlarının ise derin öğrenme algoritmaları tarafından (0,75 – 0,94 arası) sergilendiği görülmektedir. Derin öğrenme algoritmaları arıza olma durumunu gayet iyi bir şekilde tespit edebilse de arıza olmama durumu için düşük duyarlılık sebebiyle çok sayıda hatalı çıktı üretmektedir.

Adasyn yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.4'te gösterilmiştir:



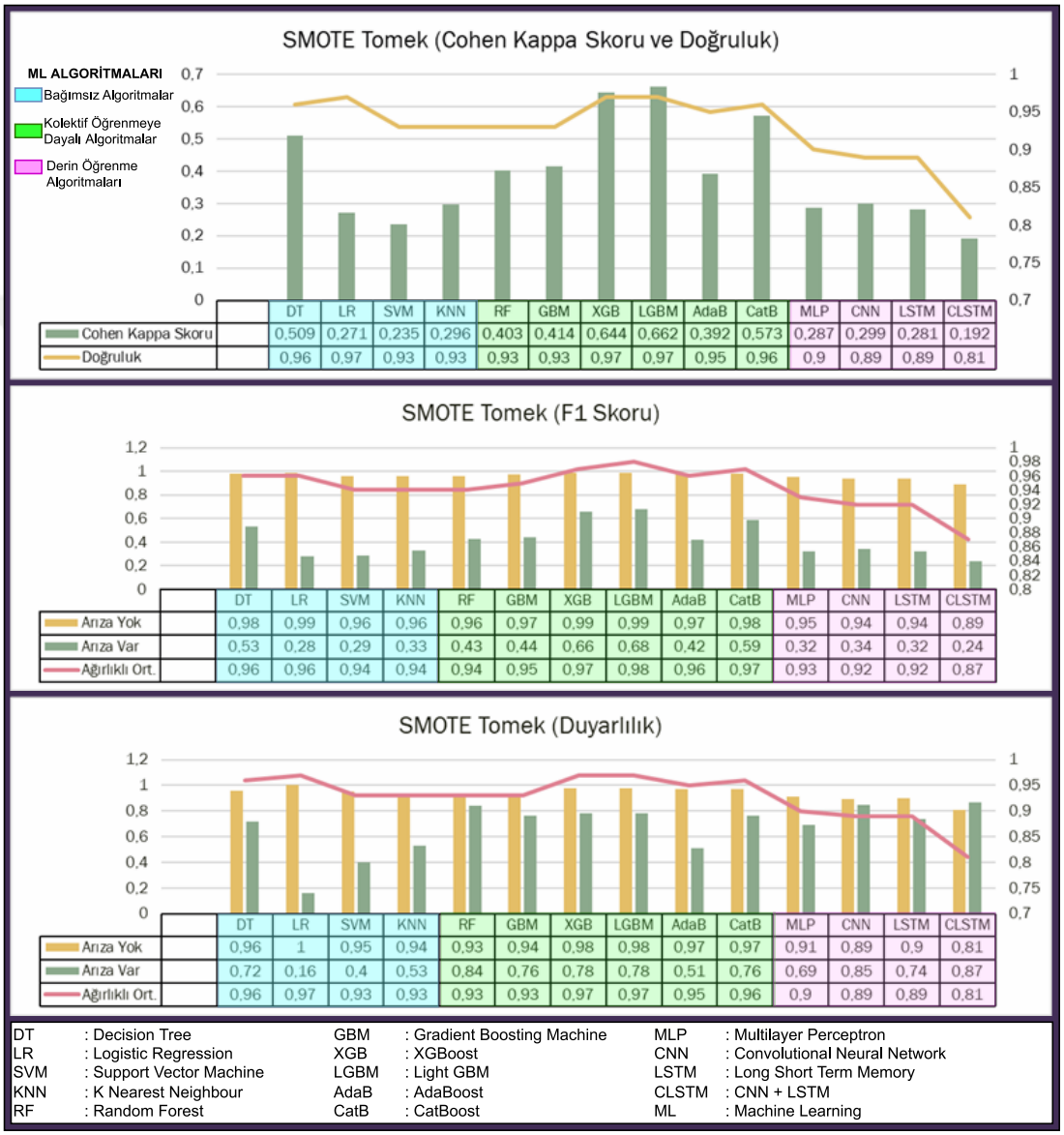
Şekil 6.4 : Adasyn Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 1).

Verileri dengeli hale getirmek için Adasyn yöntemi kullanıldığında, eğitilen modellerden en yüksek Cohen Kappa skoruna sahip olan 3 algoritma sırasıyla LGBM(0,636), XGBoost (0,607) ve CatBoost'tur. (0,573)

F1 skoru incelemesinde "Arıza Yok" durumu için LR ve LGBM algoritmaları 0,99 , XGBoost, AdaBoost ve CatBoost algoritmaları ise 0,98 değerine sahiptir. "Arıza Var" durumu için ise en yüksek F1 skorunun 0,65 ile LGBM algoritmasına ait olduğu görülüyor. Daha sonra en yüksek F1 skoruna sahip algoritmalar ise 0,62 ile XGBoost ve 0,59 ile CatBoost algoritmalarıdır.

Duyarlılık metrikleri karşılaştırıldığında “Arıza Yok” durumu için en yüksek skorun LR algoritmasında 1 olduğu görülüyor. Ancak LR algoritması aynı zamanda arıza olma durumunu en düşük doğrulukta tahmin eden algoritmadır.

Şekil 6.5’te SMOTE Tomek yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları gösterilmiştir:



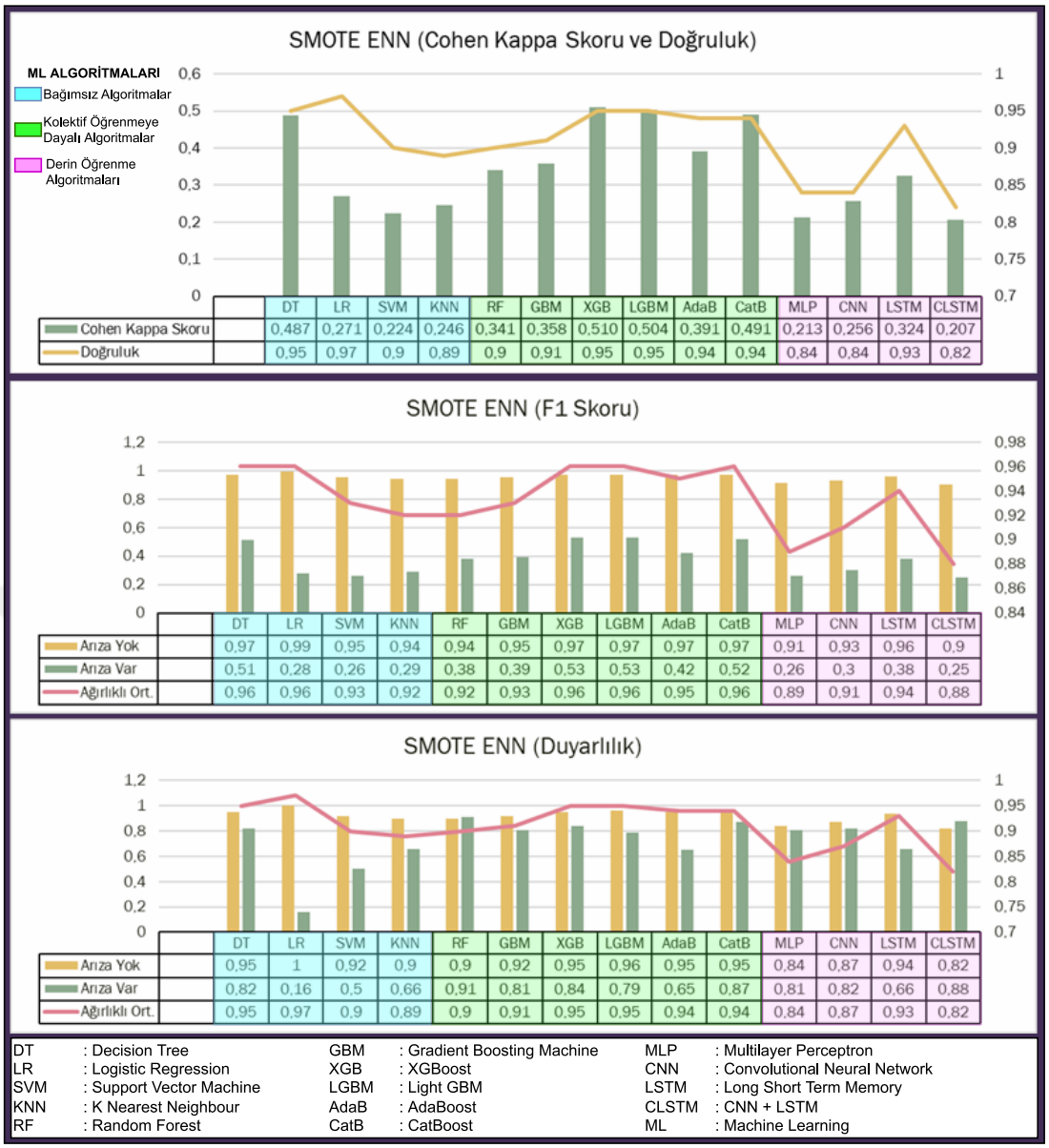
Şekil 6.5 : SMOTE Tomek Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 1).

Eğitim verisine SMOTE Tomek yöntemi uygulandığında, Adaysn yöntemine benzer şekilde en yüksek Cohen Kappa skoruna sahip olan 3 algoritmanın sırasıyla LGBM (0,662), XGBoost (0,644) ve CatBoost (0,573) olduğu görülüyor.

F1 skoru incelemesinde “Arıza Yok” durumu için LR, XGBoost ve LGBM algoritmaları 0,99 değerine sahipken bundan sonraki en yüksek F1 skorlarına 0,98 ile DT ve CatBoost algoritmaları sahiptir. “Arıza Var” durumu için ise en yüksek F1 skorunun 0,68 ile LGBM algoritmasına ait olduğu görülüyor. Daha sonra en yüksek F1 skoruna sahip algoritmalar ise 0,66 ile XGBoost ve 0,59 ile CatBoost algoritmalarıdır.

Duyarlılık metriklerine göre “Arıza Yok” durumu için en yüksek skorun LR algoritmasında 1 olduğu görülüyor. Ancak LR algoritması aynı zamanda arıza olma durumunu 0,16 değeri ile en düşük doğrulukta tahmin eden algoritmadır. Bu durum LR algoritmasının aşırı uyumlanmaya meyilli olduğunu gösterir. XGBoost ve LGBM algoritmaları arıza olmama durumu için 0,98 değerine sahipken, arıza durumu için 0,78’dir.

SMOTE ENN yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.6’da gösterilmiştir:



Şekil 6.6 : SMOTE ENN Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 1).

Tüm algoritmalar SMOTE ENN yöntemi kullanılarak eğitildiğinde en yüksek Cohen Kappa skorunu 0,510 olarak XGBoost algoritmasına aittir. Daha sonra en yüksek performansa sahip algoritmalar ise sırasıyla 0,504 ve 0,4911 Cohen Kappa skorları ile LGBM ve CatBoost algoritmalarıdır.

F1 skoru incelendiğinde "Arıza Yok" durumu için LR algoritması 0,99 skor ile ilk sırada yer almaktadır. Bunun ardından en yüksek F1 skorlarının (0,97) ise DT, XGBoost, LGBM, AdaBoost ve CatBoost algoritmalarına ait olduğu görülüyor. "Arıza Var" durumu için ise en yüksek F1 skorunun 0,53 ile XGBoost ve LGBM algoritmalarait olduğu görülüyor.

Duyarlılık metrikleri açısından "Arıza Yok" durumu için en yüksek skorun LR algoritmasında 1 olduğu görülüyor. Ancak LR algoritması bu yöntemle de aşırı uyumlanmaya meyil etmiş ve arıza olmama durumunu 0,16 duyarlılık değeri ile tahmin edebilmiştir. "Arıza Yok" durumu için ikinci en yüksek duyarlılık değerine sahip olan algoritma ise 0,96 değeri ile LGBM algoritmasıdır. "Arıza var" durumunu 0,91 duyarlılık değerine sahip RF algoritması en yüksek performans ile sınıflandırabilirken, ikinci olarak en iyi sınıflandırma yapan algoritmanın derin öğrenme modellerinden hibrit yapıdaki CNN tabanlı LSTM modelinin olduğu görülüyor. CNN tabanlı LSTM ile elde edilen duyarlılık skoru 0,88'dir.

Veri seti 1 için genel değerlendirme: Veri Seti 1 için 14 farklı makine öğrenmesi algoritması 5 farklı veri seti örnekleme yöntemi ile denendiğinden dolayı toplamda 70 farklı model ortaya çıkarılmıştır. Cohen Kappa skoruna göre bu 70 model içerisinden en yüksek performanslı 10 model Çizelge 6.4'te verilmiştir.

Çizelge 6.4 : Cohen Kappa skoruna göre en yüksek performansa sahip 10 model (Veri Seti 1).

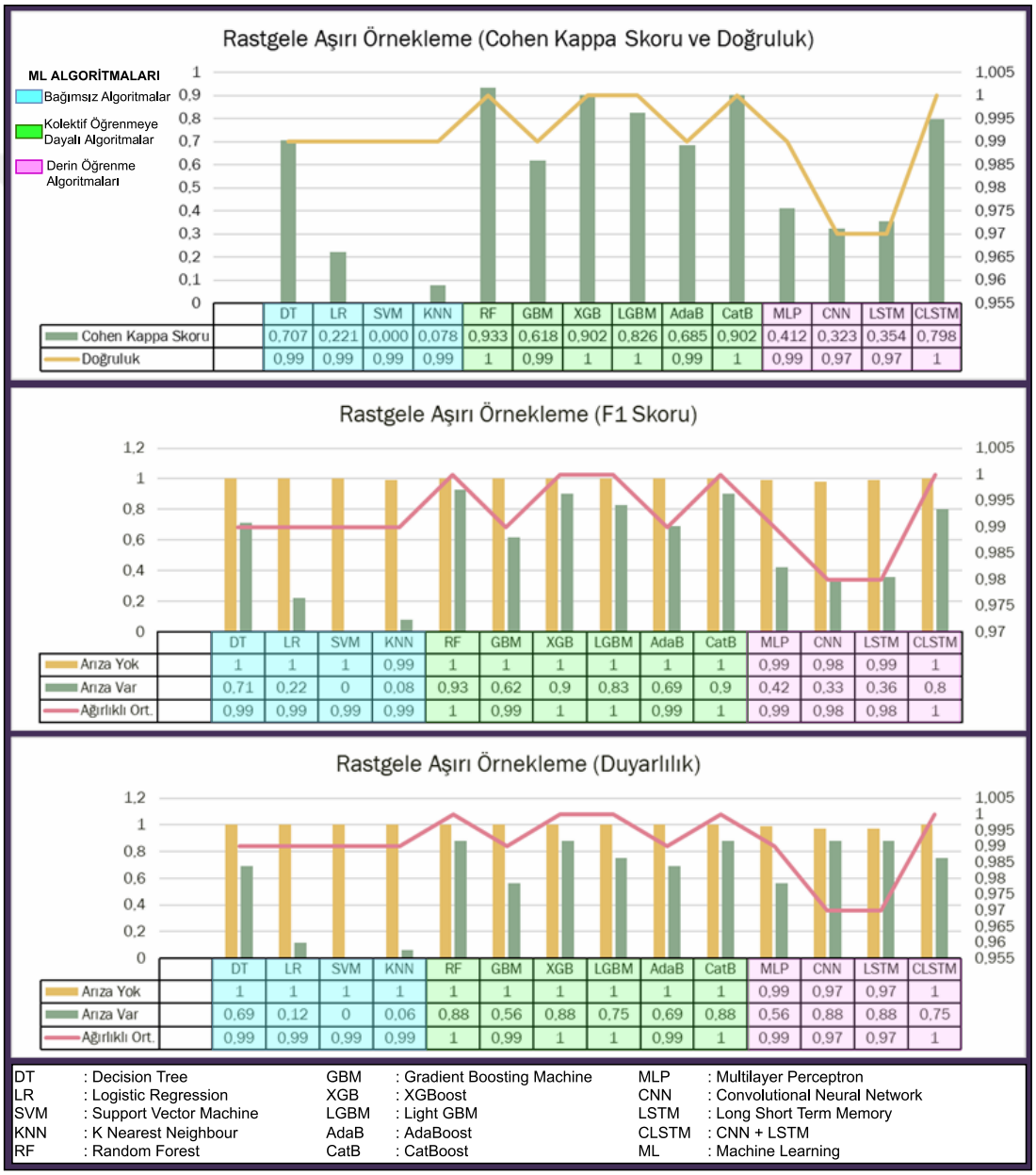
Sıra No	Modeller	Cohen Kappa Skoru	Kappa Skoru Uyum Düzeyi	Algoritma Türü	Geçen Süre
1	LightGBM	0,78	Önemli Derecede Uyum	Kolektif Öğrenme	Rastgele Aşırı Örnekleme
2	XGBoost	0,73	Önemli Derecede Uyum	Kolektif Öğrenme	Rastgele Aşırı Örnekleme
3	CatBoost	0,707	Önemli Derecede Uyum	Kolektif Öğrenme	Rastgele Aşırı Örnekleme
4	LightGBM	0,662	Önemli Derecede Uyum	Kolektif Öğrenme	SMOTE Tomek
5	XGBoost	0,657	Önemli Derecede Uyum	Kolektif Öğrenme	SMOTE
6	XGBoost	0,644	Önemli Derecede Uyum	Kolektif Öğrenme	SMOTE Tomek
7	LightGBM	0,636	Önemli Derecede Uyum	Kolektif Öğrenme	ADASYN
8	Decision Tree	0,624	Önemli Derecede Uyum	Kolektif Öğrenme	Rastgele Aşırı Örnekleme
9	LightGBM	0,619	Önemli Derecede Uyum	Kolektif Öğrenme	SMOTE
10	XGBoost	0,607	Önemli Derecede Uyum	Kolektif Öğrenme	ADASYN

Çizelge 6.4'te görüldüğü üzere Veri Seti 1 için en yüksek performansı gösteren algoritma 0,780 Cohen Kappa skoruna sahip olan LightGBM algoritmasıdır. Ayrıca en yüksek performanslı 10 modelin tamamı istisnasız bir şekilde kolektif öğrenme tabanlı algoritmalardan elde edilmiş olan modellerdir. Örnekleme türüne göre ise en

yüksek Cohen Kappa skoruna sahip ilk 3 modelin tamamı rastgele aşırı örnekleme sonucu elde edilen eğitim verileri ile eğitilmiş olan modellerdir.

6.2.2 Veri seti 2'ye ait performans sonuçlarının incelenmesi

Bu veri setinin performans metrikleri de aynen Veri Seti 1'de olduğu gibi eğitim verisi üzerinde 5 farklı örnekleme yöntemi kullanılarak ölçülmüştür. Rastgele aşırı örnekleme yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performans sonuçları Şekil 6.7'deki gibidir:



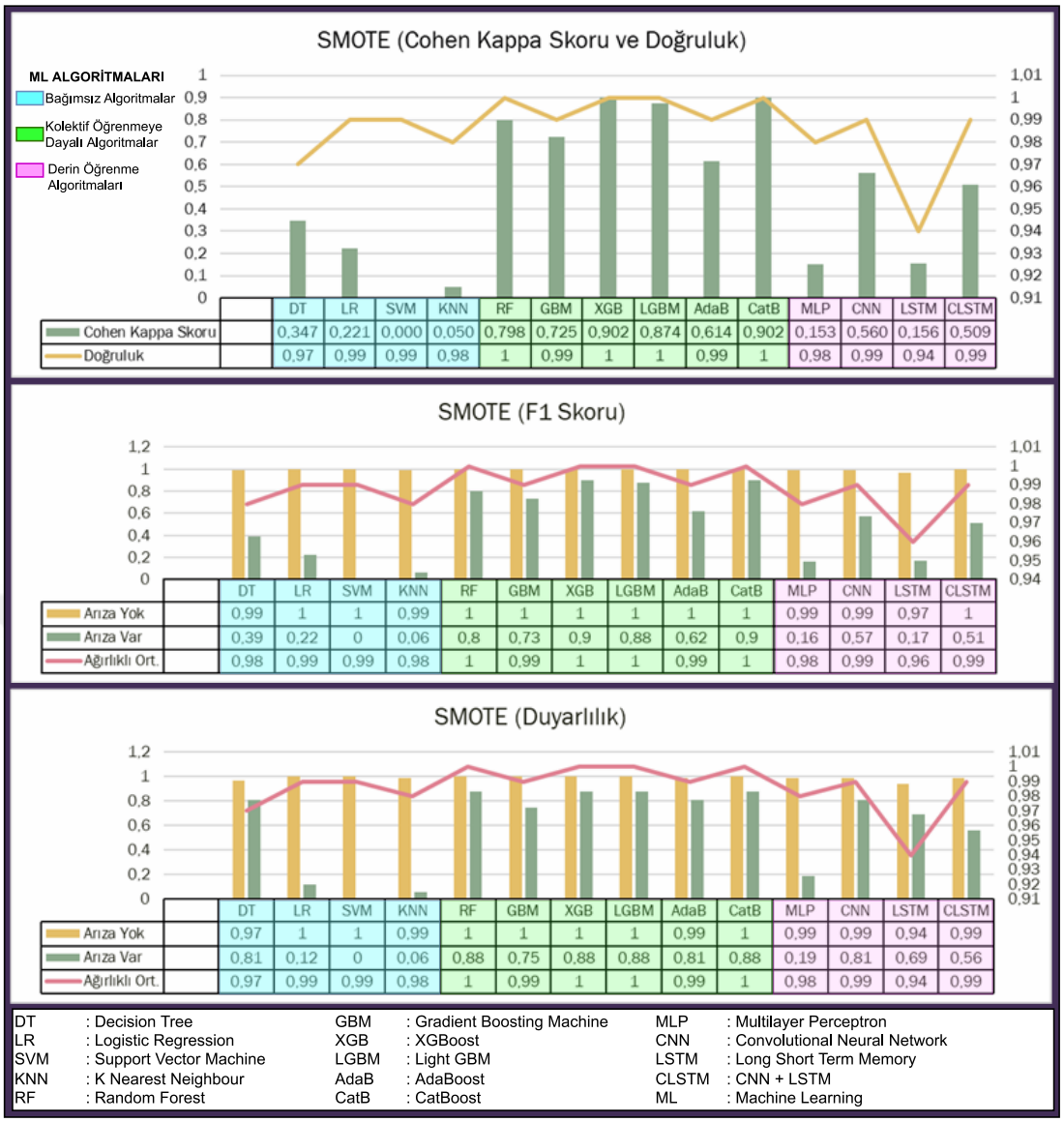
Şekil 6.7 : Rastgele Aşırı Örnekleme Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 2).

Rastgele aşırı örnekleme yöntemi kullanılarak eğitilmiş olan modellerde en yüksek Cohen Kappa skoruna 0,933 puanla RF algoritmasının sahip olduğu görülüyor. Bunu ise 0,902 Cohen Kappa skoru ile XGBoost ve CatBoost algoritmaları takip etmektedir.

F1 skoru açısından "Arıza Yok" durumu için algoritmaların tümü 0,98 ile 1 arasında skora sahiptir. "Arıza Var" durumu için ise en yüksek F1 skoruna sahip ilk 3 algoritmanın sırasıyla RF (0,93), XGBoost (0,90) ve CatBoost (0,90) olduğu görülmektedir.

Duyarlılık metrikleri incelendiğinde "Arıza Yok" durumu için tüm bağımsız (tekil) makine öğrenmesi algoritmaları ve kolektif öğrenme algoritmalarının 1 değerine sahip olduğu görülüyor. Ancak derin öğrenme algoritmalarından sadece CSLTM modeli 1 değerine sahiptir. Diğer derin öğrenme algoritmalarında duyarlılık değeri 0,97 ile 0,99 arasında değişmektedir. "Arıza Var" durumu için azınlık sınıfı gözlemlerini en yüksek performans ile sınıflandırabilen algoritmalar 0,88 skoru ile RF, XGBoost, CatBoost, CNN ve LSTM'dir. LR, SVM ve KNN algoritmalarında ise duyarlılık değerinin 0 ile 0,12 arasında değiştiği görülmektedir. Bu ise ilgili algoritmaların anomali durumunu tespit etmekte zayıf kaldığının ve modellerin aşırı uyuma meyil gösterdiğinin ifadesidir.

SMOTE yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.8'de yer almaktadır:



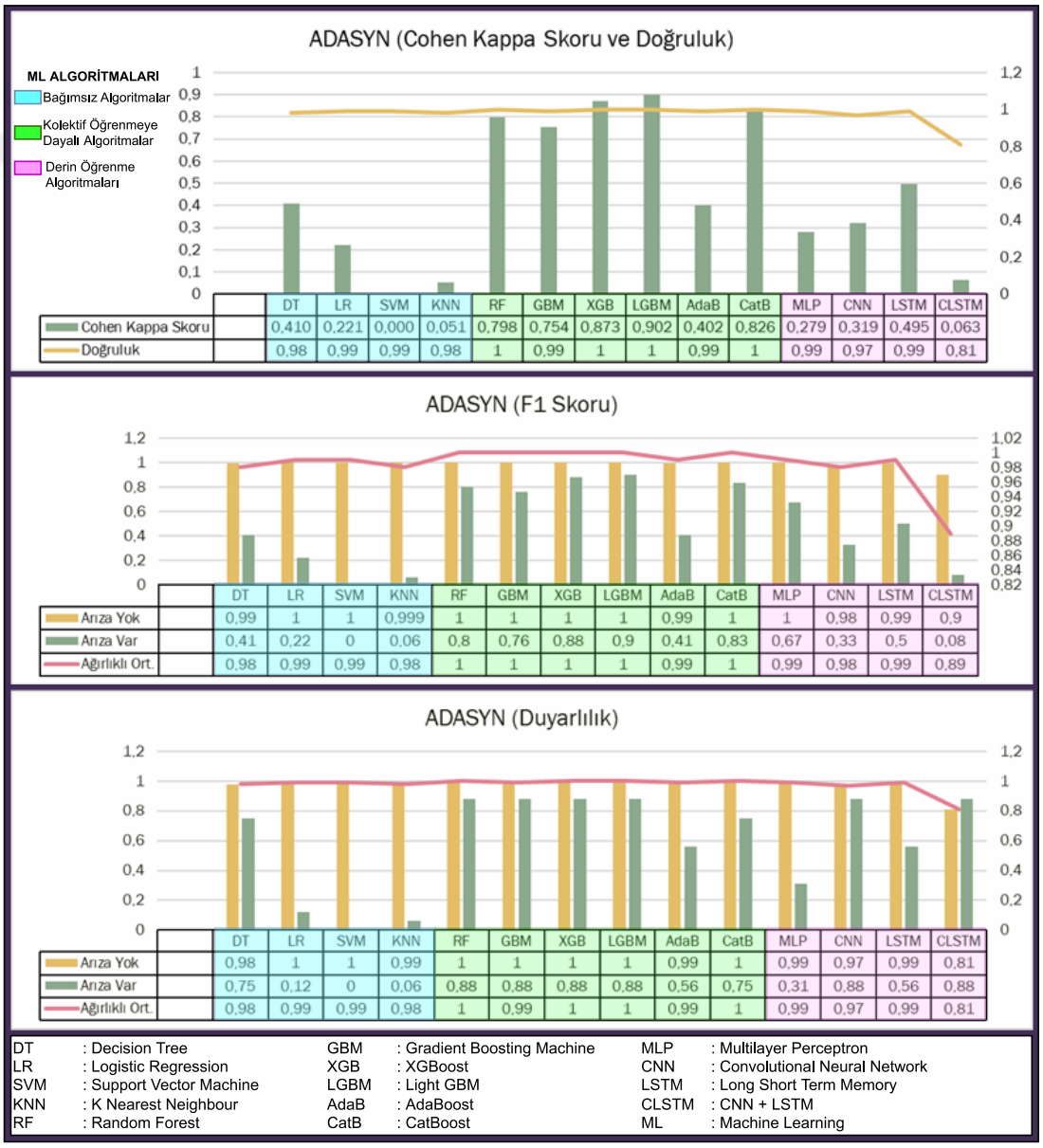
Şekil 6.8 : SMOTE Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 2).

SMOTE aşırı örnekleme yöntemi kullanıldığında eğitilen modeller arasından en yüksek Cohen Kappa skorunun 0,902 olarak XGBoost ve CatBoost algoritmalarında olduğu görülüyor. Bu 2 algoritmadan sonra en yüksek Cohen Kappa skoruna sahip olan algoritma ise 0,874'lük değer ile LGBM algoritmasıdır.

F1 skorları incelendiğinde "Arıza Yok" durumu LSTM haricindeki algoritmaların tamamı 0,99 ile 1 değerlerini almıştır. LSTM modeli ise 0,97 Cohen Kappa skoruna sahiptir. "Arıza Var" durumu için en yüksek F1 skoru 0,90 olarak XGBoost ve CatBoost algoritmalarına tarafından ortaya konulmuştur. XGBoost ve CatBoost algoritmalarını takip eden en yüksek performanslı algoritma ise 0,88 skoruyla LGBM'dir.

Duyarlılık metriği açısından bakıldığında “Arıza Yok” durumunu için DT ve LSTM modelleri haricindeki tüm modeller 0,99 ile 1 değerlerini almıştır. DT modeli 0,97 ve LSTM ise 0,94 duyarlılık değerine sahiptir. Arızanın olduğu anomali durumunu sınıflandırma da ise en yüksek performansı ise 0,88 duyarlılık değeri ile kolektif öğrenme algoritmalarından RF, XGBoost, LGBM ve CatBoost modelleri göstermiştir.

Adasyn yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.9’da yer almaktadır:



Şekil 6.9 : ADASYN Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 2).

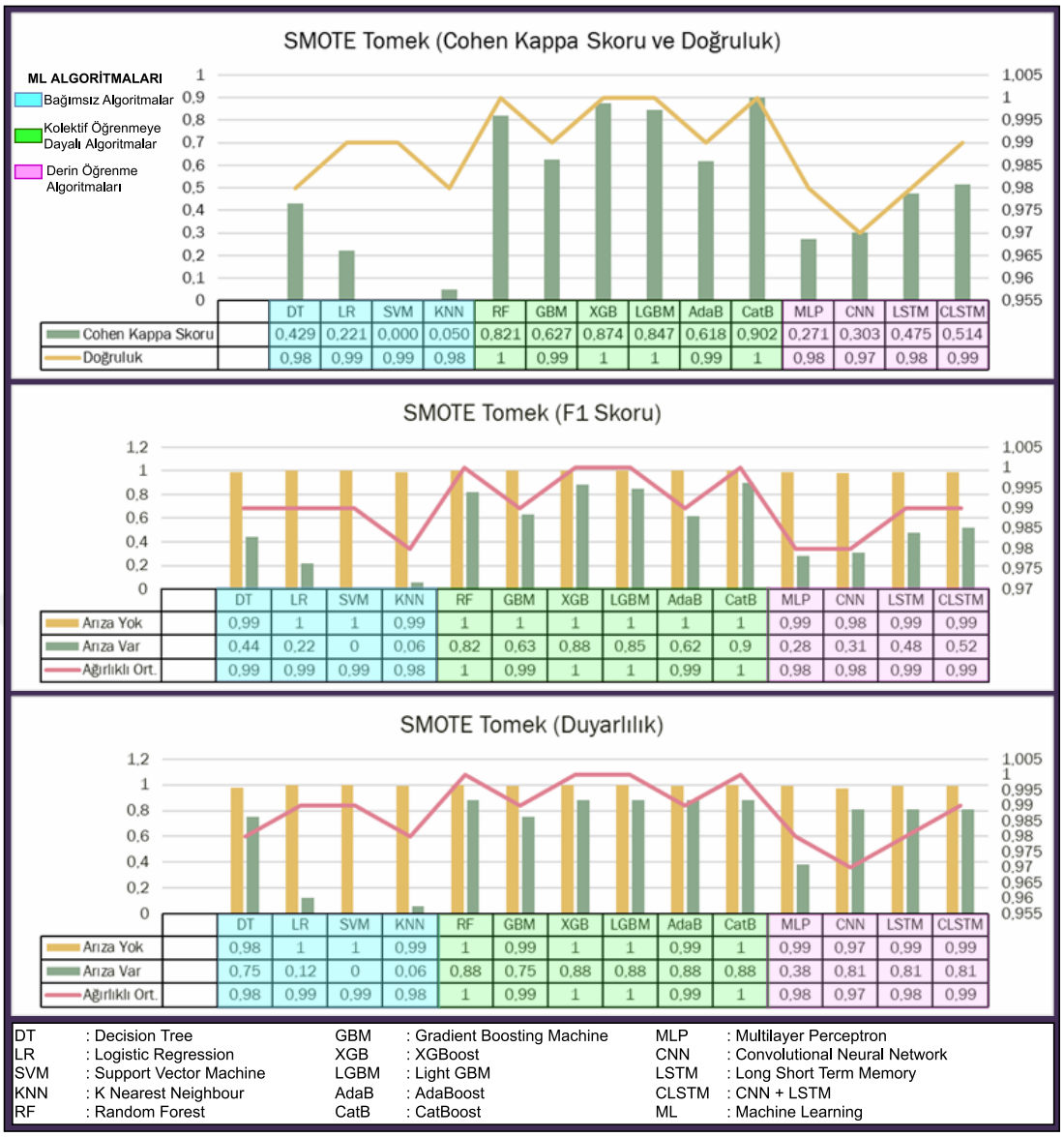
Eğitim verilerinin Adasyn yöntemi ile örneklendiği durumda Cohen Kappa skorunun en yüksek olduğu modelin 0,902’lik skorla LGBM olduğu görülüyor. LGBM’in

performansını ise 0,826 Cohen Kappa skoruyla CatBoost, 0,798 Cohen Kappa skoruyla RF modelleri takip etmektedir.

F1 skorları açısından "Arıza Yok" durumu için tüm algoritmaların 0,98 ile 1 arasında bir F1 skoruna sahip olduğu görülüyor. Arıza olma durumunun sınıflandırılmasında en yüksek performans 0,90 skoruyla LGBM tarafından ortaya konulmuştur. LGBM algoritmasının performansını ise sırasıyla 0,88 ve 0,83 skorlarıyla XGBoost ve CatBoost modelleri takip etmektedir.

Duyarlılık metrikleri incelendiğinde "Arıza Yok" durumunu için CLSTM modeli haricindeki tüm modeller 0,97 ile 1 arasında değerler almıştır. CLSTM modeli ise en düşük performans ile 0,81 değerine sahiptir. "Arıza Var" durumu için ise kolektif öğrenme algoritmalarından RF, GBM, XGBoost ve LGBM algoritmalarının ve derin öğrenme algoritmalarından ise CNN ve CLSTM algoritmalarının 0,88 değeriyle en iyi sınıflandırma performansına sahip olduğu görülmüştür.

SMOTE Tomek yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.10'da yer almaktadır:



Şekil 6.10 : SMOTE Tomek Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 2).

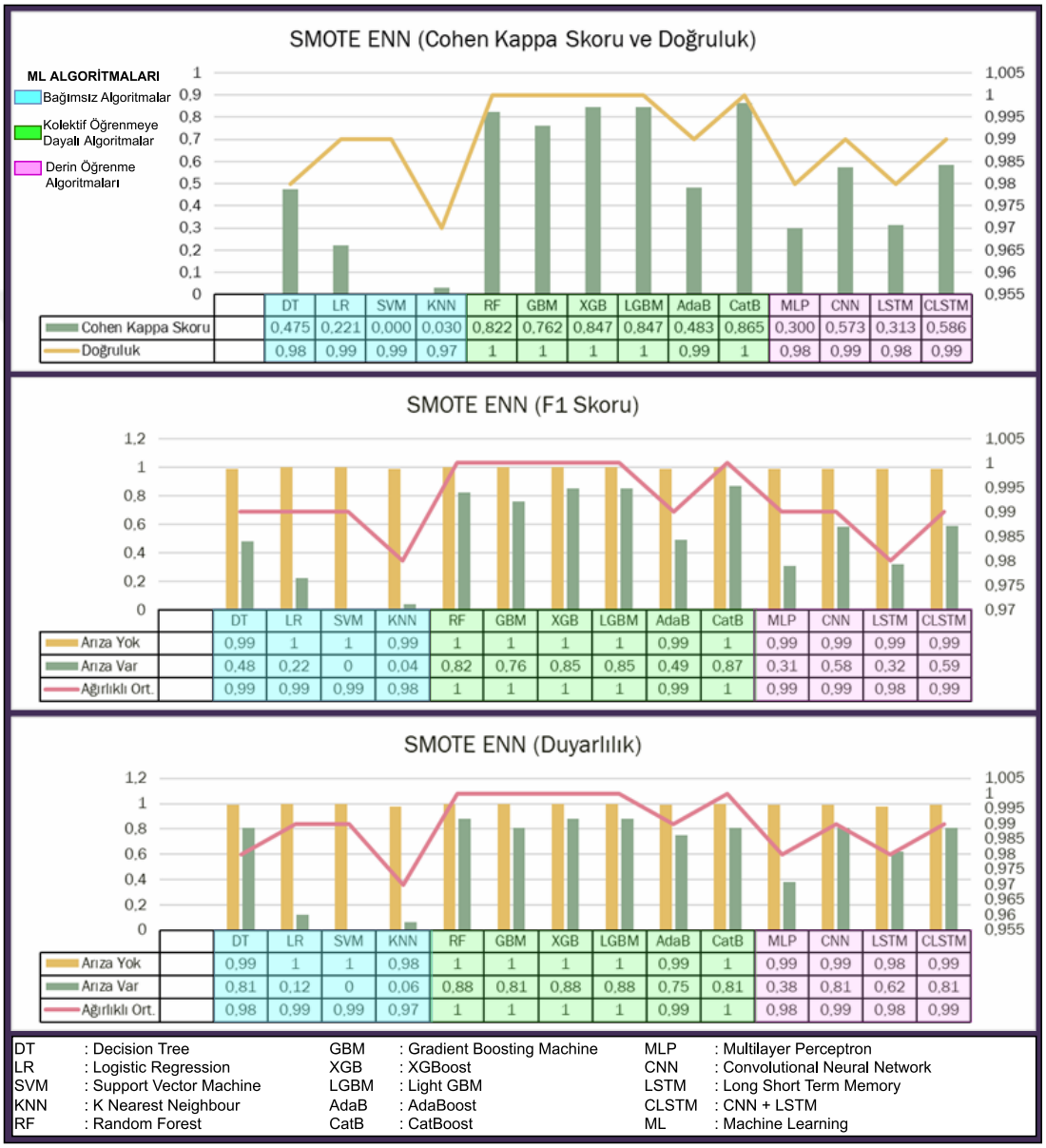
SMOTE Tomek yöntemi kullanılarak eğitilmiş olan modellerde en yüksek Cohen Kappa skoruna 0,902 değeri ile CatBoost algoritmasının sahip olduğu görülüyor. Bunu ise 0,874 Cohen Kappa skoruyla XGBoost ve 0,847 Cohen Kappa skoru ile LGBM takip etmektedir.

F1 skoru olarak "Arıza Yok" durumu için tüm algoritmalar 0,98 ile 1 arasında değerler almıştır. Ancak "Arıza Var" durumu için 0,88 F1 skoru ile en yüksek performans gösteren model XGBoost'tur. Bunu ise 0,85 ile LGBM ve 0,82 ile RF modelleri takip etmektedir.

Duyarlılık parametresi olarak da "Arıza Yok" durumu için tüm algoritmaların 0,97 ile 1 arasında performans gösterdiği görülüyor. Ancak anomali tespitini gösteren

"Arıza Var" sınıflandırması için 0,88 değeri ile en yüksek performans gösteren modellerin tamamının kolektif öğrenme algoritmaları olduğu gözlemlenmiştir. (RF, XGBoost, LGBM, AdaBoost ve CatBoost)

SMOTE ENN yöntemi kullanılarak elde edilen verilerle eğitilmiş olan modellerin, test verisi üzerindeki performansları Şekil 6.11'de gösterilmiştir:



Şekil 6.11 : SMOTE ENN Yöntemi Kullanıldığında Performans Metrikleri (Veri Seti 2).

SMOTE ENN örnekleme yöntemi kullanıldığında eğitilen modeller arasından en yüksek Cohen Kappa skorunun 0,865 olarak CatBoost algoritması tarafından sergilendiği görülüyor. CatBoost algoritmasından sonra en yüksek Cohen Kappa

skoruna sahip olan algoritmalar ise 0,847 Cohen Kappa skoru ile XGBoost ve GBM algoritmalarıdır.

F1 skorları incelendiğinde "Arıza Yok" durumu için algoritmaların tamamı 0,99 ile 1 değerlerini almaktadır. "Arıza Var" durumu için ise en yüksek F1 skoruna 0,87 ile CatBoost modeli sahiptir. CatBoost algoritmasını ise 0,85 F1 skoru ile XGBoost ve LGBM modelleri takip etmektedir.

Duyarlılık metrikleri açısından "Arıza Yok" durumunu için tüm modeller 0,98 ile 1 arasında değerler almıştır. Arızanın olduğu anomali durumu tespitinde ise en başarılı algoritmaların 0,88 duyarlılık değeri ile RF, XGBoost ve LGBM algoritmaları olmak üzere kolektif öğrenme sınıflandırıcıları tarafından sergilendiği görülmektedir.

Veri seti 2 için genel değerlendirme: Veri Seti 2 için 14 farklı makine öğrenmesi algoritması 5 farklı veri seti örnekleme yöntemi ile denendiğinden dolayı toplamda 70 farklı model ortaya çıkarılmıştır. Cohen Kappa skoruna göre bu 70 model içerisinden en yüksek performansa sahip olan 10 tanesi Çizelge 6.5'teki gibidir.

Çizelge 6.5 : Cohen Kappa skoruna göre en yüksek performanslı 10 model (Veri Seti 2)

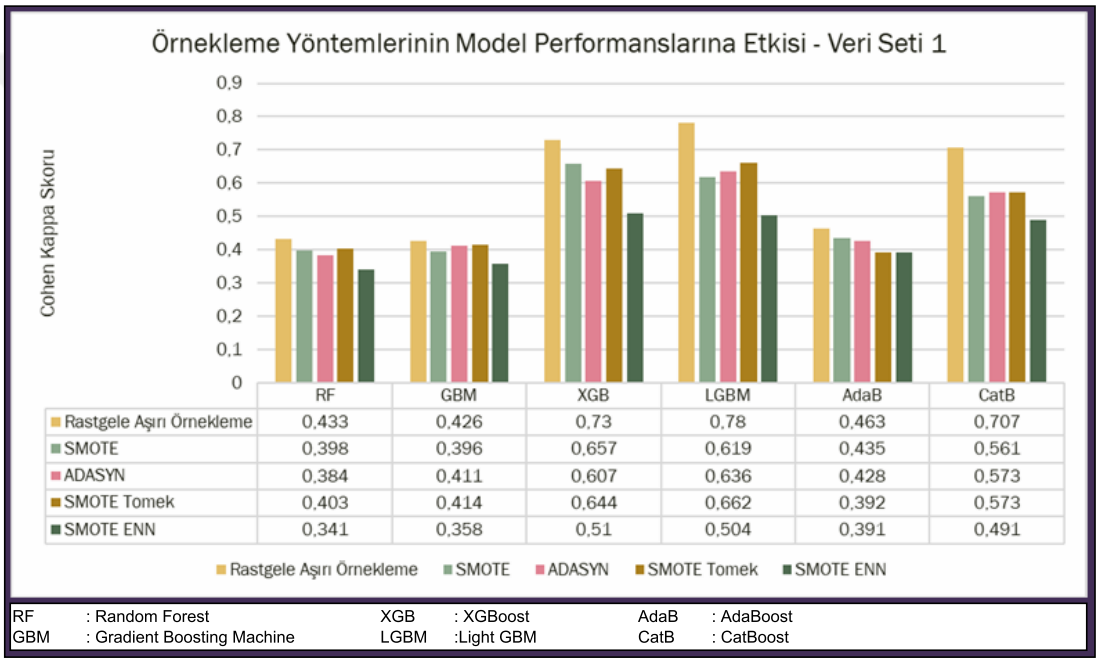
Sıra No	Modeller	Cohen Kappa Skoru	Kappa Skoru Uyum Düzeyi	Algoritma Türü	Geçen Süre
1	RF	0,933	Mükemmel Uyum	Kolektif Öğrenme	Rastgele Aşırı Örnekleme
2	XGBoost	0,902	Mükemmel Uyum	Kolektif Öğrenme	Rastgele Aşırı Örnekleme
3	CatBoost	0,902	Mükemmel Uyum	Kolektif Öğrenme	Rastgele Aşırı Örnekleme
4	XGBoost	0,902	Mükemmel Uyum	Kolektif Öğrenme	SMOTE
5	CatBoost	0,902	Mükemmel Uyum	Kolektif Öğrenme	SMOTE
6	LightGBM	0,902	Mükemmel Uyum	Kolektif Öğrenme	ADASYN
7	CatBoost	0,902	Mükemmel Uyum	Kolektif Öğrenme	SMOTE Tomek
8	LightGBM	0,874	Mükemmel Uyum	Kolektif Öğrenme	SMOTE
9	XGBoost	0,874	Mükemmel Uyum	Kolektif Öğrenme	SMOTE Tomek
10	XGBoost	0,873	Mükemmel Uyum	Kolektif Öğrenme	ADASYN

Çizelge 6.5'te görüldüğü üzere Veri Seti 2 için en yüksek performansı gösteren algoritma 0,933 Cohen Kappa skoruna sahip olan Rassal Orman (RF) algoritmasıdır. Ayrıca en yüksek performanslı 10 modelin tamamı istisnasız bir şekilde kolektif öğrenme tabanlı algoritmalardan elde edilmiş olan modellerdir. Örnekleme türüne göre ise yüksek performansı gösteren model rastgele aşırı örnekleme sonucu elde edilen eğitim verileri ile eğitilmiş olan modeldir. Onun ardından 0,902 Cohen Kappa skoru ile gelen modellerin 2 tanesi rastgele aşırı örnekleme, 2 tanesi SMOTE

örnekleme, 1 tanesi Adasyn örnekleme ve 1 tanesi de SMOTE Tomek örnekleme yöntemiyle elde edilen eğitim verileri ile eğitilmiş olan modellerdir.

6.2.3 Örnekleme yöntemlerinin model performansına etkisinin incelenmesi

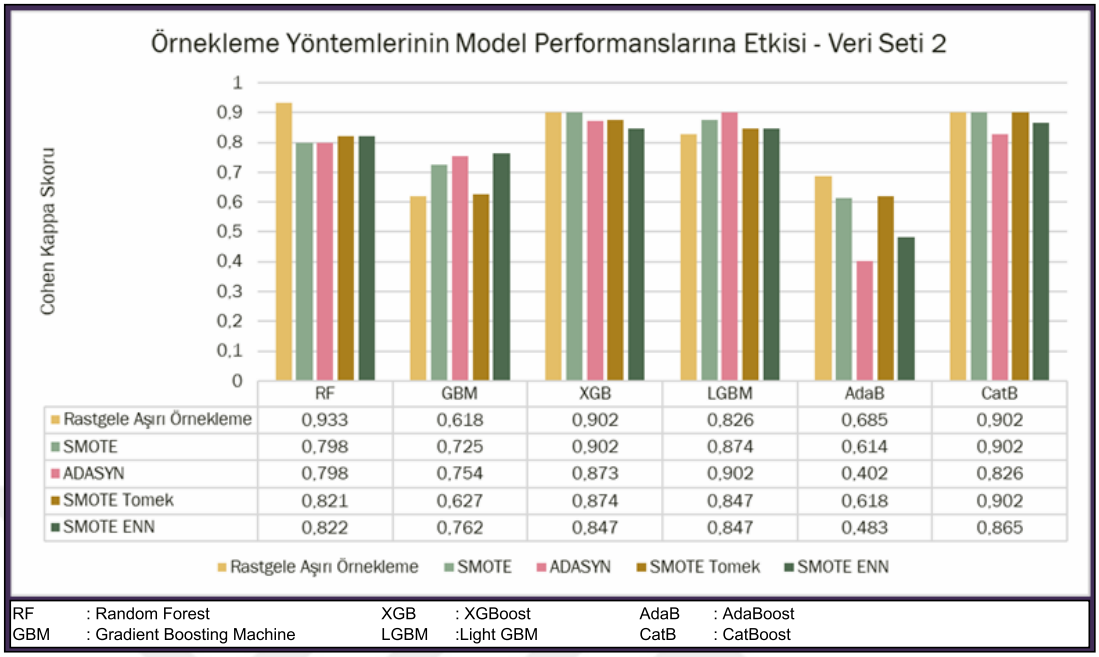
Her 2 veri seti için de Cohen Kappa Skoru açısından en yüksek performansa sahip modeller, istisnasız olarak kolektif öğrenmeye dayalı algoritmalardan elde edilen modellerdir (bknz. Çizelge 6.4 ve Çizelge 6.5) Bu sebeple örnekleme yöntemlerinin kolektif öğrenme algoritmalarının performansına etkisi ayrıca incelenmiştir. Şekil 6.12'de Veri Seti 1 için örnekleme yöntemlerinin Cohen Kappa skoruna etkisi kolektif öğrenme algoritmaları üzerinde gösterilmiştir.



Şekil 6.12 : Örnekleme yöntemlerinin model performansına etkisi (Veri Seti 1).

Veri Seti 1 için rastgele aşırı örnekleme kullanılarak eğitilen modellerin tamamında en yüksek Cohen Kappa skoru elde edilmiştir. Ayrıca her bir model için en düşük Cohen Kappa skoru ile sonuçlanan örnekleme türü ise SMOTE ENN'dir. Diğer örnekleme türlerinin performansları ise modelden modele göre değişiklik göstermektedir. Örneğin; SMOTE örneklemesi XGBoost modelinde en iyi 2. Cohen Kappa skorunu (0,657) verirken, aynı örnekleme türü LGBM algoritmasında 0,619 Cohen Kappa skoru ile rastgele aşırı örnekleme, SMOTE Tomek ve Adasyn'dan sonra ancak 4. sırada yer almıştır.

Şekil 6.13'te Veri Seti 2 için örnekleme yöntemlerinin Cohen Kappa skoruna etkisi kolektif öğrenme algoritmaları üzerinde gösterilmiştir.



	RF	GBM	XGB	LGBM	AdaB	CatB
Rastgele Aşırı Örnekleme	0,933	0,618	0,902	0,826	0,685	0,902
SMOTE	0,798	0,725	0,902	0,874	0,614	0,902
ADASYN	0,798	0,754	0,873	0,902	0,402	0,826
SMOTE Tomek	0,821	0,627	0,874	0,847	0,618	0,902
SMOTE ENN	0,822	0,762	0,847	0,847	0,483	0,865

RF : Random Forest
GBM : Gradient Boosting Machine
XGB : XGBoost
LGBM :Light GBM
AdaB : AdaBoost
CatB : CatBoost

Şekil 6.13 : Örnekleme yöntemlerinin model performansına etkisi (Veri Seti 2).

Veri Seti 2 için en yüksek Cohen Kappa skoru (0,933) RF modelinde rastgele aşırı örnekleme yöntemiyle elde edilmiştir. Ancak Veri Seti 1'de görülenin aksine rastgele aşırı örnekleme yönteminin tüm modellerin performansına olumlu etkisi genelleştirilemez. Örneğin rastgele aşırı örnekleme sonrası eğitilen GBM modeli en düşük Cohen Kappa skoruna (0,618) sahiptir. Yine de RF, XGBoost, AdaBoost ve CatBoost olmak üzere 4 modelin de ulaştığı en yüksek Cohen Kappa skorları rastgele aşırı örnekleme yöntemiyle gerçekleşmiştir.

7. SONUÇLAR VE DEĞERLENDİRME

Bu tez çalışmasında, endüstriyel ortamdaki ekipmanların arızi durumlarını temsil eden 2 farklı veri seti üzerinde klasik makine öğrenmesi ve derin öğrenme algoritmaları denenerek çok yönlü bir bakış açısı ile değerlendirilmeye çalışılmıştır. Yapılan çalışmada model performansları açısından ciddi zorluklara neden olan dengesiz sınıf dağılımına sahip veriler ile başa çıkabilmek için çeşitli örnekleme yöntemleri denenmiş ve bunların model performanslarına etkisi gözlemlenmiştir.

Elde edilen bulgular farklı açılarla ele alındığında:

Model performansları açısından bakıldığında; her 2 veri seti için de dengesiz verileri sınıflandırmada istisnasız bir şekilde kolektif öğrenme sınıflandırıcılarının en yüksek performansa sahip olduğu görülmüştür. Bağımsız makine öğrenmesi algoritmalarından lojistik regresyon, karar destek makineleri ve K-En yakın komşu algoritmalarının son derece zayıf performans sergilediği ve genellikle aşırı uyuma meyil ederek anomali verilerinin tespitini yapmaktan uzak kaldığı görülmüştür. Özellikle dengesizlik oranı daha yüksek olan Veri Seti 2 için bu durum daha da belirgin şekilde göze çarpmaktadır. Derin öğrenme algoritmaları ise bağımsız makine öğrenmesi algoritmalarından görece iyi performans sergilemesine rağmen kolektif öğrenme algoritmaları kadar yüksek bir sınıflandırma performansına erişememişlerdir.

Bu çalışmada ele alınan veri setleri 10.000'i aşmayan sayıdaki gözlemden oluşmaktadır. Derin öğrenme algoritmaları ise genellikle yüzbinlerce, milyonlarca gözlemden oluşan daha büyük veri setlerinde tercih edilmektedir. Ayrıca derin öğrenme algoritmalarının eğitimi ve optimizasyonu klasik makine öğrenmesi algoritmalarından çok daha fazla işlem yükü gerektirmektedir. Bu sebeple kişisel bilgisayarlarda yapılan derin öğrenme model optimizasyonları kısıtlı kalabilmektedir. Bu tez çalışmasında ele alınan veri setlerinden daha büyük miktarlarda veriler ile çalışıldığında ve/veya işlem gücü yeterince yüksek bilgisayarlarda daha fazla

çeşitlilikte kombinasyonlar kullanarak optimizasyon yapıldığında derin öğrenme algoritmalarında daha iyi sonuçlar elde edilebilir.

Örnekleme yöntemleri açısından bakıldığında; Veri Seti 1 ile yapılan çalışmada rastgele aşırı örneklemenin kullanıldığı tüm kolektif öğrenme algoritmalarında en yüksek Cohen Kappa skoru elde edilmiştir. Bunun yanında Veri Seti 2 ile yapılan çalışmada da toplam 6 tane kolektif öğrenme algoritmasından 4 tanesinde (RF, XGBoost, AdaBoost ve CatBoost) en yüksek Cohen Kappa skorları ile sonuçlanan modeller rastgele aşırı örnekleme yoluyla elde edilmiştir. Rastgele aşırı örneklemede azınlık sınıfından rastgele olarak seçilen gözlemlerin yeniden veri setine eklenmesi ile sınıf dengesizliği problemini çözmek amaçlanır. Bu durum ise sürekli tekrarlanan aynı veriler sebebiyle aşırı uyumu mümkün kılmaktadır. Rastgele aşırı örneklemedeki bu probleme çözüm getirebilmek için diğer yaklaşımlar tercih edilir. Ancak tez çalışmasında ele alınan veri setlerinde rastgele aşırı örnekleme yöntemiyle en yüksek Cohen Kappa skorları veren modeller incelendiğinde aşırı uyum sorunu görülmemiştir. Örneğin Veri Seti 1'de rastgele aşırı örnekleme yoluyla 0,780 ile en yüksek Cohen Kappa skorunun elde edildiği LightGBM modelinde "Arıza Yok" durumu %99 oranında doğru sınıflandırılırken, "Arıza Var" durumu ise %76 oranında doğru sınıflandırmıştır. Benzer şekilde Veri Seti 2'de rastgele aşırı örnekleme yoluyla elde edilen en yüksek Cohen Kappa skoru 0,933 ile Rassal Orman modeline aittir. Bu modelde ise "Arıza Yok" durumun tamamını doğru tahmin ederken, "Arıza Var" durumunu %88 oranında doğru sınıflandırma başarısı göstermiştir.

Çalışmamız kapsamında ele alınan Veri Seti 1, aynı zamanda bir başka çalışmada [5] çeşitli makine öğrenmesi algoritmaları ve veri seti dengeleme yöntemleri kullanılarak ele alınmıştır. İlgili çalışmada öncelikle herhangi bir örnekleme yapılmadan makine öğrenmesi algoritmaları denenmiş ve Rassal Orman algoritmasının en yüksek performansı verdiği bulunmuştur. Daha sonra ise bu algoritma SMOTE, SMOTE Tomek, Adasyn, SMOTE ENN gibi çeşitli örnekleme yaklaşımları ile test edilmiştir. Rassal orman algoritmasında SMOTE yönteminin en yüksek performansı ortaya koyduğu belirtilmiştir. Yazar bu performans sıralaması için Fowlkes Mallow skorunu esas değerlendirme metriği olarak ele almıştır. Fowlkes Mallow skoruna göre en yüksek performansı gösteren modelde "Arıza Yok" durumu %98, Arıza Var" durumu ise %76 doğrulukta sınıflandırılabilmiştir. Bu tez kapsamında rastgele aşırı

örnekleme yaklaşımıyla elde edilen CatBoost modelinde "Arıza Yok" durumu %98, "Arıza Var" durumu ise %82 doğrulukta sınıflandırılabilmiştir.

Veri setleri açısından bakıldığında; aynı yöntem ve aynı algoritmalar kullanılmasına rağmen modellerin performansları önemli derecede farklılık göstermiştir. Veri Seti 1 ile yapılan çalışmada en yüksek performanslı 10 modelin Cohen Kappa skorları 0,607 ile 0,780 arasında değişmektedir. Veri Seti 2 ile yapılan çalışmada ise en yüksek performanslı 10 modelin Cohen Kappa skorları 0,873 ile 0,933 arasında değişmektedir.

Veri setlerinde sınıf dağılımlarındaki dengesizlik sorununun anomali tespitini zorlaştırdığı ve makine öğrenmesi algoritmaları için ciddi zorluklar oluştuğunu belirtmiştik. Bu çalışmada kullanılan Veri Seti 1'in dengesizlik oranı 25,80 ve Veri Seti 2'nin dengesizlik oranı ise 107,44'tür. Bu durumu somutlaştırmak istediğimizde Veri Seti 1 için her 1 adet azınlık sınıfı gözlemine 25,8 adet çoğunluk sınıfı gözlemi denk geldiğini; Veri seti 2 için ise her 1 adet azınlık sınıfı gözlemine 107,44 adet çoğunluk sınıfı gözlemi denk geldiğini söyleyebiliriz. Yani Veri Seti 2 içerisinde yer alan anomali gözlemlerine daha seyrek rastlanıldığı için burada azınlık sınıflarını doğru bir şekilde sınıflandırmanın diğer veri setine oranla daha zor olacağı düşünülebilir. Ancak elde edilen sonuçlar ise bunun tersi bir durumu bize göstermiştir. Veri Seti 2'de yer alan azınlık sınıfları daha doğru bir şekilde sınıflandırılmıştır. Bunun farklı sebepleri olabilir. Kamu erişimine açık şekilde gerçek bir kestirimci bakım uygulaması verilerine ulaşmak kolay değildir. Bu çalışmada ele alınan Veri Seti 1, S. Matzka tarafından yapılan çalışmada [89] rassal yürüyüş işlemi ile oluşturulmuş olan sentetik bir veri setidir. Bu sentetik veri setinde yer alan gözlemler, kendilerini temsil eden sınıf etiketlerini birbirinden yeteri kadar belirgin bir şekilde ayrıştıramayacak karakteristikte olabilir. Ya da Veri seti 2'de yer alan gözlemlerden "Arıza Var" etiketine sahip olanların, arıza olma durumunu temsil etmede belirgin bir fark yaratmış olmasından dolayı daha yüksek doğrulukta tahmin edilmiş olabilir.

Uygulanmış olan yöntemlerin farklı özellikteki veri setleri üzerinde göstereceği performanslar ilerideki çalışmalarda irdelenebilir. Ayrıca dengesiz sınıf dağılımına sahip veri setleri ile başa çıkabilmek için bu çalışmada veri düzeyinde çözüm aranarak çeşitli örneklendirme yöntemlerine yer verilmiştir. İleri çalışmalarda

maliyet duyarlı çözümler de dengesiz veri setleri ile başa çıkabilmek için bir diğer bir yaklaşım olarak ele alınabilir.

KAYNAKLAR

[1] **Öteyaka H.C.** (2020). *Endüstriyel Bir Üretim Hattı İçinOptimnal Bakım Stratejisinin ve Periyodunun Belirlenmesi.* (Doktora tezi). Kütahya Dumlupınar Üniversitesi, Fen Bilimleri Enstitüsü, Kütahya.

[2] **Yıldız E.** (2022). *Öngörülü Model İşaretleme Dili Kullanılarak Yapay Zeka ve Makine Öğrenmesi Modellerinin Web Servisi Olarak Servis Edilmesi.* (Yüksek Lisans Tezi). Bursa Teknik Üniversitesi, Akıllı sistemler Mühendisliği Anabilim Dalı, Bursa

[3] **Çınar Z.M., Nuhu A.A., Zeeshan Q., vd**. (2020). Machine Learning in Predictive Maintenance towards Sustainable Smart Manufacturing in Industry 4.0. *Sustainability 2020*, 12, 8211; doi:10.3390/su12198211

[4] **Carvalho T.P., Soares F.A.A.M.N., Vita R., vd.** (2019). A systematic literature review of machine learning methods applied to predictive maintenance. https://doi.org/10.1016/j.cie.2019.106024.

[5] **Sridhar S., Sanagavarapu S.** (2021). Handling Data Imbalance in Predictive Maintenance for Machines using SMOTE-based Oversampling. İçinde: *13th International Conference on Computational Intelligence and Communication Networks (CICN) | 978-1-7281-7696-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/CICN51697.2021.9574668.* Lima, Peru: September 22-23.

[6] **Ayvaz S., Alpay K.** (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, Volume 173, 114598, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2021.114598.

[7] **Mathew V., Toby T., Singh V., vd.** (2017). Prediction of Remaining Useful Lifetime (RUL) of turbofan engine using machine learning. İçinde *Proceedings of 2017 IEEE International Conference on Circuits and Systems (ICCS 2017) | 978-1-5090-6480-9/17/$31.00©2017 IEEE.* Thuckalay, India: December 20-21

[8] **Durphaka G.K., Selvaraj B.** (2016). Predictive maintenance for wind turbine diagnostics using vibration signal analysis based on collaborative recommendation approach. İçinde: *2016 Intl. Conference on Advances in Computing, Communications and Informatics (ICACCI).* Jaipur, India: Sept. 21-24

[9] **Amihai I., Gitzel R., Kotriwala A.M.** (2018). An Industrial Case Study Using Vibration Data and Machine Learning to Predict Asset Health. İçinde: *2018 IEEE 20th Conference on Business Informatics*. Vienna, Austria : July 11-14

[10] **Praveenkumar T., Saimurugan M., Krishnakumar P., vd.** (2014). Fault Diagnosis of Automobile Gearbox Based on Machine Learning Techniques. İçinde: *12th Global Congress on Manufacturing and Management, GCMM 2014*. doi: 10.1016/j.proeng.2014.12.452.

[11] **Ceyhan H. ve Kasapbaşı M.C.** (2022). Üretim Sistemlerinde Makine Öğrenmesi ile Kestirimci Bakım Uygulaması ve Modellemesi. *Avrupa Bilim ve Teknoloji Dergisi, Sayı 33, S. 167-175*. DOI: 10.31590/ejosat.1019210

[12] **Soylu B., Yiğiter H., Sarıkaya V., vd.** (2022). Kestirimci Bakım Planlama İçin Makine Öğrenmesi Temelli Bir Karar Destek Sistemi ve Bir Uygulama. *Verimlilik Dergisi, ss. 48-66,* Oca. 2022, doi:10.51551/verimlilik.988104

[13] **Yiğit E.** (2021). *Makine Öğrenmesi Kullanılarak Endüstriyel Pres Makinesi için Kestirimci Bakım Uygulaması.* (Yüksek Lisans Tezi). Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Kocaeli

[14] **Kim D.G. ve Choi J.Y.** (2021). Optimization of Design Parameters in LSTM Model for Predictive Maintenance. *Appl. Sci. 2021*, 11(14):6450. https://doi.org/10.3390/app11146450

[15] **Hermawan A.P., Kim D.S. ve Lee J.M.** (2020). Predictive Maintenance of Aircraft Engine using Deep Learning Technique, İçinde: *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 1296-1298, doi: 10.1109/ICTC49870.2020.9289466.

[16] **Nasser A. ve Al-Khazraji H.** (2022). A hybrid of convolutional neural network and long short-term memory network approach to predictive maintenance. *International Journal of Electrical and Computer Engineering (IJECE)*. DOI:10.11591/ijece.v12i1.pp721-730.

[17] **Biswal S. ve Sabareesh G.R.** (2015). Design and Development of a Wind Turbine Test Rig for Condition Monitoring Studies. İçinde: *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, pp. 891-896, doi: 10.1109/IIC.2015.7150869. Pune, India : May 28-30

[18] **Aydin O. ve Guldamlasioglu S.** (2017). Using LSTM Networks to Predict Engine Condition on Large Scale Data Processing Framework. İçinde: *2017 4th International Conference on Electrical and Electronic Engineering (ICEEE)*, pp. 281-285, doi: 10.1109/ICEEE2.2017.7935834. Ankara, Turkey.

[19] **Cardoso D. ve Ferreira L.** (2021). Application of Predictive Maintenance Concepts Using Artificial Intelligence Tools. *Applied Sciences*. 2021; 11(1):18. https://doi.org/10.3390/app11010018

[20] **Gęca Y.** (2020). Performance comparison of machine learning algotihms for predictive maintenanc. *Informatyka, Automatyka, Pomiary W Gospodarce I Ochronie Środowiska, 10(3).* 32-35. https://doi.org/10.35784/iapgos.1834

[21] **Scalabrini Sampaio G., Vallim Filho A.R.dA., Santos da Silva L., vd.** (2019). Prediction of Motor Failure Time Using An Artificial Neural Network. *IEEE Sensors*. 19(19):4342. https://doi.org/10.3390/s19194342

[22] **Wu H., Huang A. ve Sutherland J.** (2020). Avoiding Environmental Consequences of Equipment Failure via an LSTM-Based Model for Predictive Maintenance. *17th Global Conference on Sustainable Manufacturing*. 43. 666-673. 10.1016/j.promfg.2020.02.131.

[23] **Chen Z., Liu Y. ve Liu S.** (2017). Mechanical State Prediction Based on LSTM Neural Netwok. İçinde: *2017 36th Chinese Control Conference (CCC)*, pp. 3876-3881, doi: 10.23919/ChiCC.2017.8027963. Dalian, China : July 26-28

[24] **Len R., Sun Y., Cui J., vd.** (2018). Bearing remaining useful life prediction based on deep autoencoder and deep neural networks, *Journal of Manufacturing Systems*. Volume 48, Part C, Pages 71-77, ISSN 0278-6125, https://doi.org/10.1016/j.jmsy.2018.04.008

[25] **Hesser D.F. ve Markert B.** (2018). Tool wear monitoring of a retrofitted CNC milling machine using artificial neural networks, *Manufacturing Letters*. Volume 19, Pages 1-4, ISSN 2213-8463, https://doi.org/10.1016/j.mfglet.2018.11.001.

[26] **Bakım Yönetimi İçin Temel Kılavuz.** (19 Ağustos 2021). Erişim: 23 Ekim 2022. *https:// www.asius.com.tr/post/bakım-yönetimi-i-çin-temel-kılavuz*

[27] **Gürsoy M.Ü., Çolak U.C., Gökçe M.H., vd.** (2019). Endüstri İçin Kestirimci Bakım. *Iinternational Journal Of 3D Printing Technologies And Digital Industry*. 3:1 (2019) 56-66

[28] **Reactive Maintenance Vs Preventive Maintenance Vs Predictive Maintenance.** (3 Haziran 2020). Erişim: 23 Ekim 2022. *https://www.assetinfinity.com/blog/reactive-vs-preventive-vs-predictive*

[29] **Güngör Kankaya G.** (2022). *Kestirimci Bakım Yaklaşımıyla Bir Turbofan Motrounun Çözümlenmesi*. (Yüksek Lisans Tezi). Hacettepe Üniversitesi, Elektrik ve Elektronik Muhendisliği Anabilim Dalı, Ankara

[30] **Öztanır O.** (2018). *Makine Öğrenmesi Kullanılarak Kestirimci Bakım*. (Yüksek Lisans Tezi). Hacettepe Üniversitesi, Elektrik ve Elektronik Muhendisliği Anabilim Dalı, Ankara

[31] **Ceyhan H.** (2022). *Üretim Sistemlerinde Makine Öğrenmesi ile Kestirimci Bakım Uygulaması ve Modellemesi*. (Yüksek Lisans Tezi). İstanbul Ticaret Üniversitesi, Bilgisayar Mühendisliği Anabilim Dalı, İstanbul.

[32] **Kestirimci Bakım Nedir.** (12 Mayıs 2018). Erişim: 24 Ekim 2022. *https://www.muhendisbeyinler.net/kestirimci-bakim-nedir/*

[33] **Jasiulewicz - Kaczmarek M. ve Gola A.** (2019). Maintenance 4.0 Technologies for Sustainable Manufacturing - an Overview. *IFAC-PapersOnLine*. Volume 52, Issue 10, Pages 91-96, https://doi.org/10.1016/j.ifacol.2019.10.005.

[34] **K. Dhingra P.** (4 Aralık 2018). *ML for Predictive Maintenance.* [PowerPoint slides]. Retrieved from https://www.slideshare.net/prashdhingra/machine-learning-for-predictive-maintenance-external

[35] **Jakhar D. ve Kaur I.** (2019). Artificial intelligence, machine learning and deep learning: definitions and differences. *Clinical and Experimental Dermatology*. https://doi.org/10.1111/ced.14029

[36] **Yapay Zeka Nedir? Nasıl Çalışır? İnsan Gibi Düşünen Makine.** (30 Haziran 2022) Erişim: 30 Ekim 2022. *https://soyleki.com/yapay-zeka-nedir-nasil-calisir-insan-gibi-dusunen-makine/*

[37] **Şahinci D.** (2020) *Yapay Zeka ve Reklamcılığın Geleceği*. (Doktora Tezi). İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul

[38] **McCarthy J.** (2004) *What is Artificial Intelligence?*. Computer Science Department, Stanford University, Stanford, CA 94305

[39] **Uğuz S.** (2019). *Makine Öğrenmesi -Teorik yönler ve Python uygulamaları ile Bir Yapay Zeka Ekolü.* (ss. 69) Ankara : Nobel

[40] **Süslü A.** (2019). *Doğa ve İnsan Bilimlerinde Yapay Zekâ Uygulamaları*. Akademia Doğa ve İnsan Bilimleri Dergisi , 5 (1) , 1-10 . Retrieved from https://dergipark.org.tr/tr/pub/adibd/issue/56415/787597

[41] **Biswas A.** (8 Ağustos 2020) *Machine Learning & Its Application in Predictive Maintenance*. [PowerPoint slides]. Retrieved from https://www.slideshare.net/arnabblr/machine-learning-predictive-maintenance

[42] **Makine Öğrenmesi (Machine Learning) Nedir?.** (26 Kasım 2021) Erişim: 5 Kasım 2022. https://blog.turhost.com/makine-ogrenmesi-machine-learning-nedir/

[43] **Candan H.** (2021). *Adım Adım Makine Öğrenmesi Bölüm 2: Denetimli Öğrenme Nedir?*. Erişim: 5 Kasım 2022. https://medium.com/machine-learning-turkiye/adım-adım-makine-ogrenmesi-bolum-2-denetimli-ogrenme-nedir-80ffb1322e4f

[44] **Akca M.F. (2020).** *Karar Ağaçları (Makine Öğrenmesi Serisi-3).* Erişim: 15 Kasım 2022. https://medium.com/deep-learning-turkiye/karar-agacları-makine-ogrenmesi-serisi-3-a03f3ff00ba5

[45] **Polamuri S.** (2 Mart 2017) *How the Logistic Regression Model Works in Machine Learning*. Erişim: 15 Kasım 2022. https://dataaspirant.com/how-logistic-regression-model-works/

[46] **Ulgen E.K.** (12 Şubat 2018). *Python ile Lojistik Regresyon — Makine Öğrenimi Bölüm-7*. Erişim: 16 Kasım 2022. https://medium.com/@k.ulgen90/lojistik-regresyon-makine-ogrenimi-bolum-7-c6bc685a4084

[47] **K-Nearest Neighbor (KNN) Algorithm for Machine Learning.** (t.y.). Erişim: 17 Kasım 2022. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[48] **Verma A. K., Pal S. ve Kumar S.** (2019). Classification of Skin Disease using Ensemble Data Mining Techniques. *Asian Pacific Journal of Cancer Prevention, Vol 20.* DOI:10.31557/APJCP.2019.20.6.1887

[49] **Ankara N.** (2019). *Dengesiz Kredi Skorlama Veri Setlerinde Kolektif Öğrenme Algoritmalarının Performans Değerlendirmesi.* (Yüksek Lisans Tezi). Yıldız Teknik Üniversitesi, Matematik Mühendisliği Anabilim Dalı, İstanbul

[50] **Nazman E.** (6 Eylül 2020). *Customer churn prediction.* Erişim: 15 Kasım 2022. https://medium.com/@ezginazman/prediction-of-the-churn-decision-of-customers-c2cce303d716

[51] **Baştürk O.** (2021). *Hava trafik akışının iyileştirilmesi için rassal orman ve derin sinir ağları tabanlı varış zamanı ve yörünge tahmini*. (Doktora Tezi). Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Eskişehir

[52] **Gök E.C.** (2021). *Kan değerleri ile covıd-19 enfekte düzeyinin rassal orman sınıflandırıcı ile tahmin edilmesi*. (Yüksek Lisans Tezi). Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta

[53] **Yangın G.** (2019). *XGboost ve karar ağacı tabanlı algoritmaların diyabet veri setleri üzerine uygulaması.* (Yüksek Lisans Tezi). Mimar Sinan Üniversitesi Güzel Sanatlar Fakültesi, Fen Bilimleri Enstitüsü, İstanbul

[54] **Şahin D.** (2022). *Comparative analysis of XGBoost and LightGBM methods for day ahead spot natural gas price forecasting.* (Yüksek Lisans Tezi). İstanbul Technical University, Energy Science and Technology Division, İstanbul

[55] **Patrous Z. S.** (2018). *Evaluating XGBoost for User Classification by using Behavioral Classification by using Behavioral Smartphone Sensors.* (Master in Computer Science). KTH Royal Institute of Technology, School of electrical Engineering and Computer Science, Stockholm

[56] **Muratlar E.R.** (29 Nisan 2020). *LightGBM.* Erişim: 23 Kasım 2022. https://www.veribilimiokulu.com/lightgbm/

[57] **Üstüner M., Abdikan S., Bilgin G. vd.** (2020). *Hafif Gradyan Artırma Makineleri ile Tarımsal Ürünlerin Sınıflandırılması.* Turkish Journal of Remote Sensing and GIS. Eylül 2020, 1(2): 97-105

[58] **Aksu Ş.H. ve Çakıt E.** (2022). *A machine learning approach to classify mental workload based on eye tracking data.* Journal of the Faculty of Engineering and Architecture of Gazi University. 38:2 (2023) 1027-1039. DOI: 10.17341/gazimmfd.1049979

[59] **Yenigün O.** (2021). *AdaBoost from Scratch.* Erişim: 23 Kasım 2022. https://medium.com/mlearning-ai/adaboost-from-scratch-f8979d961948

[60] **Acet A.** (2022). *SVM, NB, KNN, Adaboost ve Random Forest Sınıflandırma Algoritmaları Kullanılarak Meme Kanserinin Tahmini.* (Yüksek Lisans Tezi). İnönü Üniversitesi, Fen Bilimleri Enstitüsü, Malatya

[61] **Brownlee J.** (14 Ağustos 2020). *What is Deep Learning?*. Erişim: 25 Kasım 2022. https://machinelearningmastery.com/what-is-deep-learning/

[62] **Doğan F. ve Türkoğlu İ.** (2019). *Derin Öğrenme Modelleri ve Uygulama Alanlarına İlişkin Bir Derleme.* DÜMF Mühendislik Dergisi 10:2 (2019) : 409-445. DOI: 10.24012/dumf.411130

[63] **Meng Z., Hu Y. ve Ancey C.** (2020). Using a Data Driven Approach to Predict Waves Generated by Gravity Driven Mass Flows. *Water*. 12(2):600. https://doi.org/10.3390/w12020600

[64] **Kızrak A.** (4 Şubat 2019). *Derin Öğrenme İçin Aktivasyon Fonksiyonlarının Karşılaştırılması.* Erişim: 07 Kasım 2022. https://ayyucekizrak.medium.com/derin-ogrenme-icin-aktivasyon-fonksiyonlarının-karşılaştırılması-cee17fd1d9cd

[65] **Caceres P.** (20 Mart 2020). *The Multilayer Perceptron - Theory and Implementation of the Backpropagation Algorithm.* Erişim: 08 Kasım 2022. https://pabloinsente.github.io/the-multilayer-perceptron

[66] **Abubakar M.M.** (2022). *Heart Sounds Classification Using Deep Learning Algorithms.* (Master's Thesis). Fırat University, Graduate School Of Natural and Applied Sciences, Elazığ

[67] **Ezer E.** (2022). *Kolonoskopi Görüntülerindeki Poliplerin Evrişimli Sinir Ağları ile Tespiti.* (Yüksek Lisans Tezi). Marmara Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul

[68] **Bastem T.U.** (2022). *Deep Learning Based Phishing Web Page Detection.* (Master's Thesis). Çankaya University, Graduate School Of Natural and Applied Sciences, Ankara

[69] **İnik Ö. ve Ülker E.** (2017). *Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri* . Gaziosmanpaşa Bilimsel Araştırma Dergisi , 6 (3) , 85-104 . Retrieved from https://dergipark.org.tr/tr/pub/gbad/issue/31228/330663

[70] **12.2 Convolutional Neural Network.** (t.y.). Erişim: 27 Kasım 2022. https://scientistcafe.com/ids/convolutional-neural-network.html

[71] **Camgözlü Y.** (2021). *Evrişimli Sinir Ağı Kullanılarak Yaprak Resimlerinin Sınıflandırılması.* (Yüksek Lisans Tezi). İskenderun Teknik Üniversitesi, Mühendislik ve Fen Bilimleri Enstitüsü, Hatay

[72] **Javadov İ.** (2020). *Evrişimli Sinir Ağları ile Plaka Tanımada Algortimaların Karşılaştırılması.* (Yüksek Lisans Tezi). İstanbul Aydın Üniversitesi, Lisansüstü Eğitim Enstitüsü, İstanbul

[73] **Mohammed S.H.** (2021). *Detection of Cancer Area in Lung Images with the Help of Deep Learning Algortihm.* (Master's Thesis). Fırat University, Graduate School Of Natural and Applied Sciences, Elazığ

[74] **Ergin T.** (2 Ekim 2018). *Evrişimsel Sinir Ağında Niçin Dropout Kullanmamalısınız.* Erişim: 27 Kasım 2022. https://medium.com/@tuncerergin/evrisimsel-sinir-aginda-nicin-dropout-kullanmamalisiniz-7e31941f8bb0

[75] **Moradi S.** (2022). *Real-Time Crash Risk Analysis Using Deep Learning.* (M.Sc. Thesis). İstanbul Technical University, Department of Civil Engineering, İstanbul

[76] **Tuncer A.** (2022). *LSTM Metodu Kullanılarak Rüzgar Hızının Tahmin Edilmesi.* (Yüksek Lisans Tezi). Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul

[77] **Bao W., Yue J. ve Rao Y.** (2017). A deep learning framework for financial time series using stacked autoencoders and long short term memory. *PLoS ONE* 12(7): e0180944. https://doi. org/10.1371/journal.pone.0180944

[78] **Palangi H., Ward R. ve Deng L.** (2016). Distributed Compressive Sensing : A Deep Learning Approach. *arXiv:1508.04924 [cs.LG]*. https://doi.org/10.48550/arXiv.1508.04924

[79] **Hochreiter S. ve Schmidhuber J.** (1997). Long Short-Term Memory. İçinde:

Neural Computation 9(8): 1735-1780.

[80] **Kaytan M., Aydilek İ.B., Yeroğlu C., vd.** (2022). *Sigmoid-Gumbel: Yeni Bir Hibrit Aktivasyon Fonksiyonu.* Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, c. 11, sayı. 1, ss. 29-45, Mar. 2022, doi:10.17798/bitlisfen.990508

[81] **Dolphin R.** (21 Ekim 2020). *LSTM Networks | A Detailed Explanation.* Erişim: 28 Kasım 2022. https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9

[82] **Tasdelen A. ve Sen, B.** (2021). A hybrid CNN-LSTM model for pre-miRNA classification. *Sci Rep* 11, 14125 (2021). https://doi.org/10.1038/s41598-021-93656-0

[83] **General Python FAQ.** (t.y.). Erişim: 29 Kasım 2022. https://web.archive.org/web/20121024164224/http://docs.python.org/faq/general.html#what-is-python

[84] **Eryılmaz F.** (26 Mayıs 2019). *Jupyter Notebook Nedir? Nasıl Kullanılır? Veri Bilimi!*. Erişim: 28 Kasım 2022. https://farukeryilmaz.com/jupyter-notebook-nedir-nasil-kullanilir-veri-bilimi/

[85] **Pandas Vs NumPy: What's The Difference?.** (2022). Erişim: 04 Aralık 2022. https://www.interviewbit.com/blog/pandas-vs-numpy/

[86] **Libraries in Python.** (2021). Erişim: 04 Aralık 2022. https://www.geeksforgeeks.org/libraries-in-python/

[87] **AI4I 2020 Predictive Maintenance Dataset Data Set.** (2020). Erişim: 04 Aralık 2022. https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset

[88] **Url-1** < https://bigml.com/user/czuriaga/gallery/dataset/587d062d49c4a16936000810> Erişim: 04 Aralık 2022. BigML, Inc. Corvallis, Oregon, USA, 2011

[89] **Matzka S.** (2020). Explainable Artificial Intelligence for Predictive Maintenance Applications. İçinde: *2020 Third International Conference on Artificial Intelligence for Industries (AI4I)*, 2020, pp. 69-74, doi: 10.1109/AI4I49448.2020.00023.

[90] **Zhu R., Guo Y. ve Xue J-H.** (2020) Adjusting the imbalance ratio by the dimensionality of imbalanced data. *Pattern Recognit. Lett., 133*, 217-223.

[91] **Pir M.Ş.** (2022). *77- Dengesiz Veri Setlerinde Sınıflandırma Problemlerinin Çözümünde Melez Yöntem Uygulaması.* (Yüksek Lisans Tezi). Bursa Uludağ Üniversitesi, Fen Bilimleri Enstitüsü, Bursa

[92] **Chawla N., Bowyer K., Hall L., vd.** (2002). SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res. (JAIR).* 16. 321-357. 10.1613/jair.953.

[93] **López F.** (01 Mart 2021). *SMOTE: Synthetic Data Augmentation for Tabular Data*. Erişim: 04 Aralık 2022. https://towardsdatascience.com/smote-synthetic-data-augmentation-for-tabular-data-1ce28090debc

[94] **He H., Bai Y., Garcia E.A., vd.** (2008). Adaptive synthetic sampling approach for imbalanced learning. *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. doi: 10.1109/ijcnn.2008.4633969.

[95] **Brandt J. ve Lanzén E.** (2021). *A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification.* (Dissertation). Uppsala University, Department of Statistics, Uppsala, Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-432162

[96] **Handling imbalanced dataset in supervised learning using family of SMOTE algorithm.** (2017). Erişim: 07 Aralık 2022. https://www.datasciencecentral.com/handling-imbalanced-data-sets-in-supervised-learning-using-family/

[97] **Başarır S.** (2022). *Suç Veri Setini Analiz Etmek İçin Makine Öğreniminde Örnekleme Teknikleri ve Uygulanması.* (Yüksek Lisans Tezi). Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, İstanbul

[98] **Lin G-M. ve Zeng H-C.** (2021). Electrocardiographic Machine Learning to Predict Mitral Valve Prolapse in Young Adults. *IEEE Engeneering in Medicine and Biology Society Section*. Volume 9, 2021.

[99] **Milli M.F.E.** (2022). *Sınıf Dengeleme Yöntemlerinin Makine Öğrenmesi Teknikleri Üzerine Etkisi : Kredi Risk Örneği.* (Yüksek Lisans Tezi). Dokuz Eylül Üniversitesi, Sosyal Bilimler Enstitüsü, İzmirkappa

[100] **Bıkmaz Bilgen Ö. ve Doğan N.** (2017). *Puanlayıcılar Arası Güvenirlik Belirleme Tekniklerinin Karşılaştırılması.* Journal of Measurement and Evaluation in Education and Psychology , 8 (1) , 63-78 . DOI: 10.21031/epod.294847

EKLER

EK A : Makine öğrenmesi modellerinin parametreleri

Çizelge A.1 : Veri Seti 1 üzerinde çalışılan optimize edilmiş makine öğrenmesi modellerinin parametreleri.

Algoritmalar	Rastgele Aşırı Örnekleme	SMOTE	ADASYN	SMOTE Tomek	SMOTE ENN
DT	criterion: gini max_depth: 14 min_samples_split: 2	criterion: entropy max_depth: 13 min_samples_split: 3	criterion: entropy max_depth: 12 min_samples_split: 2	criterion: entropy max_depth: 13 min_samples_split: 3	criterion: entropy max_depth: 13 min_samples_split: 2
LR	C: 0.01 penalty: l2	C: 0.01 penalty: l2	C: 0.01 penalty: l2	C: 0.01 penalty: l2	C: 0.01 penalty: l2
SVM	C: 0.1 gamma: 1 kernel: rbf	C: 10 gamma: 0.01 kernel: rbf	C: 10 gamma: 0.01 kernel: rbf	C: 10 gamma: 0.01 kernel: rbf	C: 10 gamma: 0.01 kernel: rbf
KNN	n_neighbors: 1 p: 2	n_neighbors: 2 p: 1	n_neighbors: 1 p: 1	n_neighbors: 2 p: 1	n_neighbors: 1 p: 2
RF	criterion: gini max_depth: 6 max_features: auto n_estimators: 200	criterion: gini max_depth: 6 max_features: auto n_estimators: 600	criterion: entropy max_depth: 6 max_features: auto n_estimators: 600	criterion: gini max_depth: 6 max_features: auto n_estimators: 400	criterion: entropy max_depth: 6 max_features: auto n_estimators: 600
GBM	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: sqrt min_samples_leaf: 3 min_samples_split: 3 n_estimators: 10 subsample: 1.0	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: sqrt min_samples_leaf: 3 min_samples_split: 3 n_estimators: 10 subsample: 1.0	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: sqrt min_samples_leaf: 3 min_samples_split: 3 n_estimators: 10 subsample: 1.0	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: sqrt min_samples_leaf: 8 min_samples_split: 3 n_estimators: 10 subsample: 1.0	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: sqrt min_samples_leaf: 3 min_samples_split: 8 n_estimators: 10 subsample: 1.0
XGBoost	learning_rate: 0.1 max_depth: 7 n_estimators: 180	learning_rate: 0.1 max_depth: 8 n_estimators: 180	learning_rate: 0.1 max_depth: 7 n_estimators: 180	learning_rate: 0.1 max_depth: 7 n_estimators: 180	learning_rate: 0.1 max_depth: 7 n_estimators: 180
Light GBM	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 30 num_leaves: 127 reg_alpha: 0.1	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 30 num_leaves: 31 reg_alpha: 0.1	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 50 num_leaves: 127 reg_alpha: 0.1	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 50 num_leaves: 127 reg_alpha: 0.1	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 30 num_leaves: 127 reg_alpha: 0.1
AdaBoost	base_estimator__criterion: gini base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: entropy base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: entropy base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: gini base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: entropy base_estimator__splitter: best n_estimators: 1

Çizelge A.1 (devam) : Veri Seti 1 üzerinde çalışılan optimize edilmiş makine öğrenmesi modellerinin parametreleri.

Algoritmalar	Rastgele Aşırı Örnekleme	SMOTE	ADASYN	SMOTE Tomek	SMOTE ENN
CatBoost	max_depth: 5 n_estimators: 200	max_depth: 5 n_estimators: 300	max_depth: 5 n_estimators: 300	max_depth: 5 n_estimators: 300	max_depth: 5 n_estimators: 200
MLP	activation: relu alpha: 0.05 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam	activation: relu alpha: 0.05 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam	activation: relu alpha: 0.0001 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam	activation: relu alpha: 0.0001 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam	activation: relu alpha: 0.05 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam
CNN	Batch Size : 256 Epoch Size : 75 filters: 16 kernel_size: 2 learning_rate: 0.001 units in denses : (256, 192, 192, 192, 1)	Batch Size : 256 Epoch Size : 75 filters: 16 kernel_size: 8 learning_rate: 0.001 units in denses : (224, 128, 224, 128, 160, 256, 96, 1)	Batch Size : 128 Epoch Size : 75 filters: 32 kernel_size: 2 learning_rate: 0.001 units in denses : (64, 96, 192, 64, 224, 192, 192, 1)	Batch Size : 64 Epoch Size : 75 filters: 16 kernel_size: 8 learning_rate: 0.001 units in denses : (160, 160, 224, 64, 160 ,128, 224, 128, 1)	Batch Size : 128 Epoch Size : 25 filters: 16 kernel_size: 8 learning_rate: 0.01 units in denses : (224, 192, 256, 1)
LSTM	Batch Size : 128 Epoch Size : 40 LSTM units : 100 units in denses : (192, 128, 160, 32, 224, 32, 1)	Batch Size : 128 Epoch Size : 40 LSTM units : 200 units in denses : (192,160, 224, 128, 1)	Batch Size : 128 Epoch Size : 40 LSTM units : 200 units in denses : (128, 160, 128, 192, 224, 1)	Batch Size : 128 Epoch Size : 40 LSTM units : 200 units in denses : (160, 128, 224, 128, 64, 192, 160, 160, 192, 1)	Batch Size : 128 Epoch Size : 40 LSTM units : 200 units in denses : (192, 64 ,160, 160, 128, 192, 1)
CNN+ LSTM	Batch Size : 128 Epoch Size : 40 filters: 32 kernel_size: 4 LSTM units : 200 learning_rate: 0.001 units in denses : (96, 224, 224, 256, 160, 64, 192, 224, 160, 1)	Batch Size : 128 Epoch Size : 40 filters: 16 kernel_size: 8 LSTM units : 200 learning_rate: 0.001 units in denses : (96, 256, 64, 96,1)	Batch Size : 128 Epoch Size : 40 filters: 16 kernel_size: 8 LSTM units : 200 learning_rate: 0.001 units in denses : (128, 96, 192, 1)	Batch Size : 128 Epoch Size : 40 filters: 16 kernel_size: 8 LSTM units : 200 learning_rate: 0.001 units in denses : (64, 96, 32, 64, 1)	Batch Size : 128 Epoch Size : 40 filters: 16 kernel_size: 8 LSTM units : 50 learning_rate: 0.001 units in denses : (96, 96, 96, 224, 96 , 96, 1)

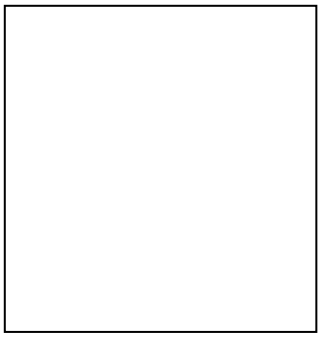
Çizelge A.2 : Veri Seti 2 üzerinde çalışılan optimize edilmiş makine öğrenmesi modellerinin parametreleri.

Algoritmalar	Rastgele Aşırı Örnekleme	SMOTE	ADASYN	SMOTE Tomek	SMOTE ENN
DT	criterion: gini max_depth: 13 min_samples_split: 2	criterion: gini max_depth: 14 min_samples_split: 3	criterion: gini max_depth: 14 min_samples_split: 2	criterion: gini max_depth: 14 min_samples_split: 2	criterion: gini max_depth: 14 min_samples_split: 2
LR	C: 10.0 penalty: l2	C: 10.0 penalty: l2	C: 10.0 penalty: l2	C: 10.0 penalty: l2	C: 10.0 penalty': l2
SVM	C: 1 gamma: 1 kernel: rbf	C: 10 gamma: 1 kernel: rbf	C: 10 gamma: 1 kernel: rbf	C: 10 gamma: 1 kernel: rbf	C: 1 gamma: 1 kernel': rbf
KNN	n_neighbors: 1 p: 1	n_neighbors: 2 p: 1	n_neighbors: 1 p: 1	n_neighbors: 2 p: 1	n_neighbors: 2 p: 1
RF	criterion: gini max_depth: 6 max_features: auto n_estimators: 200	criterion: gini max_depth: 6 max_features: auto n_estimators: 600	criterion: gini max_depth: 6 max_features: auto n_estimators: 200	criterion: gini max_depth: 6 max_features: auto n_estimators: 400	criterion: gini max_depth: 6 max_features: auto n_estimators: 400
GBM	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: log2 min_samples_leaf: 8 min_samples_split: 3 n_estimators: 10 subsample: 1.0	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: log2 min_samples_leaf: 3 min_samples_split: 8 n_estimators: 10 subsample: 1.0	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: sqrt min_samples_leaf: 3 min_samples_split: 3 n_estimators: 10 subsample: 0.8	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: sqrt min_samples_leaf: 3 min_samples_split: 8 n_estimators: 10 subsample: 0.8	criterion: mae learning_rate: 0.2 loss: deviance max_depth: 8 max_features: log2 min_samples_leaf: 3 min_samples_split: 3 n_estimators: 10 subsample: 1.0
XGBoost	learning_rate: 0.1 max_depth: 4 n_estimators: 180	learning_rate: 0.1 max_depth: 5 n_estimators: 180	learning_rate: 0.1 max_depth: 5 n_estimators: 180	learning_rate: 0.1 max_depth: 5 n_estimators: 180	learning_rate: 0.1 max_depth: 5 n_estimators: 180
Light GBM	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 30 num_leaves: 127 reg_alpha: 0.1	lambda_l1: 1 lambda_l2: 0 min_data_in_leaf: 30 num_leaves: 31 reg_alpha: 0.1	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 100 num_leaves: 31 reg_alpha: 0.1	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 30 num_leaves: 127 reg_alpha: 0.1	lambda_l1: 0 lambda_l2: 0 min_data_in_leaf: 30 num_leaves: 127 reg_alpha: 0.1

Çizelge A.2 (devam) : Veri Seti 2 üzerinde çalışılan optimize edilmiş makine öğrenmesi modellerinin parametreleri.

Algoritmalar	Rastgele Aşırı Örnekleme	SMOTE	ADASYN	SMOTE Tomek	SMOTE ENN
AdaBoost	base_estimator__criterion: gini base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: entropy base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: gini base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: entropy base_estimator__splitter: best n_estimators: 1	base_estimator__criterion: entropy base_estimator__splitter: best n_estimators: 1
CatBoost	max_depth: 4 n_estimators: 200	max_depth: 5 n_estimators: 200	max_depth: 4 n_estimators: 100	max_depth: 5 n_estimators: 300	max_depth: 5 n_estimators: 300
MLP	activation: tanh alpha: 0.0001 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam	activation: relu alpha: 0.0001 hidden_layer_sizes: (20,) learning_rate: constant solver: adam	activation: relu alpha: 0.0001 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam	activation: relu alpha: 0.0001 hidden_layer_sizes: (20,) learning_rate: constant solver: adam	activation: relu alpha: 0.0001 hidden_layer_sizes: (10, 30, 10) learning_rate: constant solver: adam
CNN	Batch Size: 128 Epoch Size: 25 filters: 32 kernel_size: 2 learning_rate: 0.01 units in denses : (96, 224, 256, 96, 128, 256, 96, 160, 1)	Batch Size: 64 Epoch Size: 25 filters: 16 kernel_size: 2 learning_rate: 0.01 units in denses : (160, 192, 96, 32, 224, 224, 128, 96, 1)	Batch Size: 256 Epoch Size: 25 filters: 64 kernel_size: 2 learning_rate: 0.01 units in denses : (128, 160, 224, 96, 64, 160, 1)	Batch Size: 256 Epoch Size: 50 filters: 64 kernel_size: 2 learning_rate: 0.01 units in denses : (256, 32, 224, 1)	Batch Size: 64 Epoch Size: 25 filters: 32 kernel_size: 2 learning_rate: 0.01 units in denses: (64, 192, 32, 96, 160, 192, 160, 32 ,128, 1)
LSTM	Batch Size: 128 Epoch Size: 40 LSTM units: 100 units in denses: (32, 128, 64, 160, 96, 64, 1)	Batch Size: 128 Epoch Size: 40 LSTM units: 100 units in denses s: (96, 100, 32, 32, 32, 32, 32, 32, 32, 32, 1)	Batch Size: 128 Epoch Size: 40 LSTM units: 100 units in denses: (64, 64, 1)	Batch Size: 128 Epoch Size: 40 LSTM units: 100 units in denses: (256, 128, 64, 192, 256, 1)	Batch Size: 128 Epoch Size: 40 LSTM units: 50 units in denses: (32, 224, 192, 224, 160, 256, 1)
CNN+ LSTM	Batch Size: 128 Epoch Size: 40 filters: 64 kernel_size: 2 LSTM units: 50 learning_rate: 0.01 units in denses: (256, 64, 64, 1)	Batch Size: 128 Epoch Size: 40 filters: 32 kernel_size: 4 LSTM units: 200 learning_rate: 0.001 units in denses: (64, 128, 96, 224, 192, 160, 1)	Batch Size: 128 Epoch Size: 40 filters: 16 kernel_size: 2 LSTM units: 100 learning_rate: 0.01 units in denses: (64, 256, 256, 1)	Batch Size: 128 Epoch Size: 40 filters: 32 kernel_size: 2 LSTM units: 200 learning_rate: 0.001 units in denses: (32, 192, 32, 1)	Batch Size: 128 Epoch Size: 40 filters: 32 kernel_size: 4 LSTM units: 50 learning_rate: 0.01 units in denses: (224, 128, 128, 64 ,1)

ÖZGEÇMİŞ



Ad-Soyad : Ümit DİLBAZ

Doğum Tarihi ve Yeri :

E-posta :

ÖĞRENİM DURUMU:

- **Lisans :** 2010, Süleyman Demirel Üniversitesi, Mühendislik ve Mimarlık Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2012 - ... Türkiye Şişe ve Cam Fabrikaları A.Ş – Elektrik Elektronik Bakım Onarım Mühendisi

TEZDEN TÜRETİLEN ESERLER, SUNUMLAR VE PATENTLER:

- Dilbaz Ü. and Cingiz M.Ö. (2022). "Predictive Maintenance on Industrial Data Using Soft Voted Ensemble Classifiers," in *International Conference on Computing, Intelligence and Data Analytics (ICCIDA2022)*