



**SİNYAL İŞLEME VE MAKİNE ÖĞRENMESİ YÖNTEMLERİ
KULLANILARAK BÜYÜKBAŞ HAYVAN DAVRANIŞLARININ TAHMİNİ**

Ahmet PÜTÜN

**YÜKSEK LİSANS TEZİ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

OCAK 2025

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Ahmet PÜTÜN

16/01/2025

SİNYAL İŞLEME VE MAKİNE ÖĞRENMESİ YÖNTEMLERİ KULLANILARAK BÜYÜKBAŞ HAYVAN DAVRANIŞLARININ TAHMİNİ

(Yüksek Lisans Tezi)

Ahmet PÜTÜN

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Ocak 2025

ÖZET

Bu çalışmada, büyükbaş hayvan davranışlarının izlenmesi ve sınıflandırılması amacıyla, ivmeölçer sensörlerinden elde edilen veriler analiz edilmiştir. Büyükbaş hayvanlardan elde edilen veriler, farklı davranış kategorilerini içermektedir. Veri setindeki dengesizlikler nedeniyle az sayıda örneğe sahip bazı davranış kategorileri yeniden değerlendirilmiş ve veri işleme aşamasında uygun yöntemlerle ele alınmıştır. Bu işlemler ile sınıflandırma modellerinin azınlık davranış kategorilerinden kaynaklanan dengesizliklerin etkisinin azaltılması amaçlanmıştır. Çalışmada istatistiksel, entropi, yineleme grafiği tabanlı özelliklerin çıkarımı gerçekleştirilmiştir. Bu özellikler, özellik seçimi ile optimize edilerek en az sayıda özellikte en iyi modeli üretmek hedeflenmiştir. Çok sayıda geleneksel makine öğrenmesi sınıflayıcıları ve LSTM algoritması kullanılarak sınıflandırmalar gerçekleştirilmiştir. Performans değerlendirmeleri sonucunda, Ekstra Ağaçlar sınıflayıcısı %98,18 doğruluk değeri ile literatürde bu veri seti için en az özellikte elde edilmiş en iyi performansı göstermiştir. LSTM modeli ise %92,70 doğruluk oranına ulaşmıştır. Bu çalışma, modern teknolojilerin hayvancılık sektörüne entegrasyonu ile hayvan davranışlarının doğru bir şekilde izlenebileceğini ve sınıflandırılabilirliğini göstermesinin yanında, elde edilen yüksek doğruluk oranıyla hayvan sağlığının izlenmesi ve refahının artırılması için iyi bir potansiyel sunmaktadır. Ayrıca, önerilen yöntemlerin farklı veri setlerinde uygulanabilir olması, tarım ve hayvancılık alanındaki diğer çalışmalara da katkı sağlayacaktır.

Bilim Kodu : 93421

Anahtar Kelimeler : İvmeölçer, sınıflandırma, LSTM, veri analizi, zaman serisi analizi, Ekstra Ağaçlar sınıflayıcısı

Sayfa Adedi : 87

Danışman : Doç. Dr. Derya YILMAZ

PREDICTION OF CATTLE BEHAVIORS BY USING SIGNAL PROCESSING AND MACHINE LEARNING METHODS

(M. Sc. Thesis)

Ahmet PÜTÜN

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

January 2025

ABSTRACT

In this study, data obtained from accelerometer sensors were analyzed for the purpose of monitoring and classifying cattle behaviors. Data obtained from cattle include different behavioral categories. Due to imbalances in the data set, some behavioral categories with few examples were re-evaluated and handled with appropriate methods in the data processing phase. These operations aimed to reduce the effect of imbalances originating from minority behavioral categories of classification models. In the study, statistical, entropy, recurrence plots based features were extracted. These features were optimized with feature selection and the aim was to produce the best model with the least number of features. Classifications were performed using a large number of traditional machine learning classifiers and LSTM algorithms. As a result of performance evaluations, Extra Trees classifier showed the best performance obtained with the least number of features in the literature for this data set with 98.18% accuracy value. The LSTM model reached 92.70% accuracy rate. This study demonstrates that animal behavior can be accurately monitored and classified with the integration of modern technologies into the livestock sector and offers a good potential for monitoring animal health and improving animal welfare with the high accuracy rate obtained. In addition, the applicability of the proposed methods to different data sets will contribute to other studies in the field of agriculture and livestock.

Science Code : 93421

Key Words : Accelerometer, classification, LSTM, data analysis, time series analysis, Extra Trees Classifier

Page Number : 87

Supervisor : Associate Professor Dr. Derya YILMAZ

TEŐEKKÜR

Bu alıőmanın gerekleőtirilmesinde, deęerli bilgilerini bizlerle paylaőarak bana yol gsteren, akademik ve kiőtisel geliőtimde her zaman destekleyici bir rehber olan, sabır ve zverisiyle beni motive eden kıymetli danıőtmanım Do. Dr. Derya YILMAZ'a en iten teőtakkürlerimi sunarım. Bu srete bana kattıęınız bilgi, deneyim ve ilham iin minnettarım.

Ayrıca, bu zorlu srete sevgileri, anlayıőtları ve sabırlarıyla her zaman yanımda olan sevgili aileme sonsuz teőtakkür ederim. Onların varlıęı, alıőtmamı tamamlamamda en byk motivasyon kaynaęı olmuőtur.



İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xi
1. GİRİŞ.....	1
2. MATERYAL VE YÖNTEM	7
2.1. Veri Seti ve Kullanılan Teknolojiler	8
2.1.1. İvmeölçer.....	8
2.1.2. Japon siyah sığır ineği davranış sınıflandırması veri seti.....	9
2.2. Normalizasyon	10
2.3. Pencereleme	11
2.4. Eksik ve Hatalı Verilerin İşlenmesi.....	14
2.5. Özellik Çıkarımı	15
2.5.1. Temel istatistiksel özellikler	16
2.5.2. Sinyal ve vektör tabanlı özellikler.....	22
2.5.3. Entropi veya fraktal özellikler	24
2.5.4. Zaman frekans alanı özellikleri.....	28
2.5.5. Yineleme grafiği tabanlı özellikler.....	30
2.6. Sınıf Dengesi	36
2.6.1. Azınlık sınıfların çıkarılması	37

	Sayfa
2.7. Sentetik Azınlık Aşırı Örnekleme Tekniği	38
2.8. Sınıflama	39
2.8.1. Ağaç tabanlı sınıflayıcılar	40
2.8.2. Doğrusal yöntemli sınıflayıcılar	41
2.8.3. Topluluk yöntemleri sınıflayıcılar.....	43
2.8.4. Çekirdek tabanlı sınıflayıcılar	45
2.8.5. Bayes yöntemli sınıflayıcılar.....	46
2.8.6. Yakınlık tabanlı sınıflayıcılar.....	47
2.8.7. Diğer sınıflayıcılar	48
3. UYGULAMALAR VE ELDE EDİLEN SONUÇLAR	51
3.1. Kullanılan Donanım ve Yazılım Altyapısı.....	52
3.2. Algoritmanın İlerleyişi.....	53
3.3. Özellik Seçimi Süreci	54
3.4. Model Eğitim Süreci.....	56
3.5. Performans Değerlendirme ve Karşılaştırma	58
4. TARTIŞMA.....	73
5. SONUÇ VE ÖNERİLER.....	77
KAYNAKLAR	79
EKLER.....	83
EK-1 Optimizasyonda kullanılan parametreler	84
ÖZGEÇMİŞ	87

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 2.1. Veri setindeki çeşitli davranışlar ve ineklere göre dağılımları	10
Çizelge 2.2. Veri setinde yer alan ivmeölçer verilerinin sütunları ve açıklamaları	10
Çizelge 2.3. Çıkarılan özelliklerin listesi	17
Çizelge 3.1. Özellik seçimi işlemi sonrası belirlenen önem değerleri	54
Çizelge 3.2. Kullanılan özelliklerin eksenlerle ilişkisi	58
Çizelge 3.3. Kullanılan özelliklerin ortalaması ve standart sapması	59
Çizelge 3.4. Azınlık sınıfları çıkarılmadan sınıflama sonuçları	60
Çizelge 3.5. Sınıflandırma modellerinin test doğruluğu performans karşılaştırması	62
Çizelge 3.6. Sınıfların yanlış reddetme oranları (FRR, %)	64
Çizelge 3.7. Yineleme grafiği özellikleri kullanılarak elde edilen sınıflama performansları	66
Çizelge 4.1. Literatürde yapılan çalışmalar	75

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Hayvan Davranışlarının Sınıflandırılmasında İzlenen Sürecin Genel Gösterimi.....	7
Şekil 2.2. Ham ve normalize edilmiş verilerin karşılaştırılması.....	12
Şekil 2.3. Veri setinden örnek pencereleme gösterimi	14
Şekil 2.4. Veri setinden azınlık sınıfları çıkarılmasından sonra etiket dağılımları.....	37
Şekil 2.5. Veri setindeki ham etiket dağılımları	38
Şekil 3.1. Çalışmanın uygulama süreci.....	53
Şekil 3.2. Seçilen özelliklerin gösterimi	56
Şekil 3.3. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (a) Linear Ridge (b) LDA (c) QDA (d) SGD Classifier (e) GPC (f) Gaussian Navie Bayes	67
Şekil 3.4. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (g) Bernoulli Navie Bayes (h) Karar Ağacı Sınıflayıcısı (i) Bagging Classifier (j) ET Sınıflayıcısı (k) Ada Boost Classifier (l) Gradyan Artırma Sınıflayıcısı.....	68
Şekil 3.5. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (l) HGBC (m) Rastgele Orman Sınıflayıcısı (n) KNN Sınıflayıcı (o) RBC (ö) NCC (p) Lojistik Regresyon	69
Şekil 3.6. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (r) Lasso Lojistik Regresyon (s) ENC (ş) SVC (t) Kuadratik Kernel SVC (u) RBF Kernel SVC (ü) Linear SVC	70
Şekil 3.7. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (v) Pasif Agresif Sınıflayıcı (y) Perceptron (z) Voting Hard Classifier (w) Voting Soft Classifier (x) LSTM	71
Şekil 3.8. Tüm sınıfların kullanıldığı makine öğrenmesi modelleri için karışıklık matrisleri (aa) ET Sınıflayıcısı (bb) KNN Sınıflayıcısı (cc) Kuadratik Kernel SVC (dd) Bagging Sınıflayıcısı	72

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Simgeler

Açıklamalar

Hz

Hertz

g

Yer çekimi

Kısaltmalar

Açıklamalar

ATT

Saldırma

BMN

Çiftleşme

DT

Karar Ağacı

DRN

Su içme

DRR

Çapraz Yineleme Oranı

ESC

Kaçma

ENC

Esnek Ağ Sınıflayıcısı

ET

Ekstra Ağaçlar

FES

Yemlikte yeme

FRR

Yanlış Reddetme Oranı

FFT

Hızlı Fourier Dönüşümü

FRT

İlk Yineleme Zamanı

GB

Gradyan Artırma

GLCM

Gray Level Co-Occurrence Matrix

GPC

Gauss Süreci Sınıflayıcı

GRZ

Otlama

HGBC

Hist Gradient Boosting Sınıflayıcı

HFD

Higuchi Fraktal Boyutu

KNN

K-en yakın komşular

LCK

Yalama

LDA

Doğrusal Diskriminant Analizi

Kısaltmalar**Açıklamalar**

LSTM	Uzun Kısa Süreli Bellek
LR	Lojistik Regresyon
NB	Naive Bayes
NCC	En Yakın Merkez Sınıflayıcı
MEMS	Mikro-Elektro-Mekanik Sistemler
MOV	Hareket
PPV	Hassasiyet
RBC	Radius Neighbors Classifier
REL	Yatarak dinlenme
RES	Ayakta dinlenme
RF	Rastgele Orman
RNN	Yinelemeli sinir ağı
RMS	Karekök Ortalama
RP	Yineleme Grafiği
RUS	Ayakta geniş getirme
RQA	Yineleme Nicelik Belirleme Analizi
URI	İdrar yapma
SGD	Stochastic Gradient Descent
SMA	Sinyal büyüklük alanı
SMOTE	Sentetik Azınlık Aşırı Örneklemme Tekniği
SLT	Tuz yalama
SVC	Destek Vektör Sınıflayıcı
SVD	Singular Value Decomposition
SVM	Destek Vektör Makinesi
TT	Yakalama Süresi
VRR	Dikey Yineleme Oranı
XGB	Aşırı Gradyan Artırma
QDA	Karesel Diskriminant Analizi

1. GİRİŞ

Küresel nüfus artışı ve buna bağlı olarak artan gıda talebi, tarım ve hayvancılık sektörlerini daha sürdürülebilir ve verimli yöntemler geliştirmeye yönlendirmektedir. Hayvancılık gıda üretiminin temel taşlarından biridir. Hayvancılığın sürdürülebilirliği için her zaman mevcut kaynakların etkin bir şekilde kullanılması ve üretim süreçlerinin optimize edilmesi gerekmektedir. Hayvancılıkta verimliliği artırmanın temel yollarından biri de hayvanların davranışlarını doğru bir şekilde izlemek ve bu davranışları anlamlandırmaktır. Bunun için hayvanların günlük aktivitelerinin, hareket alışkanlıklarının ve davranış kalıplarının doğru bir şekilde izlenmesi gerekmektedir. Bu aktiviteler onların sağlık durumları, stres seviyeleri, beslenme alışkanlıkları ve genel refahları hakkında kritik bilgiler sağlamaktadır [1]. Bu bilgiler sadece üretim kalitesini artırmakla kalmayıp, aynı zamanda hayvan refahını ön planda tutarak sürdürülebilir hayvancılık hedeflerinin gerçekleştirilmesine de doğrudan katkıda bulunmaktadır.

Hayvan davranışlarını izlemek ve anlamlandırmak hem bireysel hem de sürü düzeyinde önemli faydalar sunmaktadır fakat geleneksel yöntemlerle hayvan davranışlarının izlenmesi genellikle zaman alıcı, maliyetli ve insana dayalı bir süreçtir. Bununla birlikte modern teknoloji ve otomasyon sistemlerinin hayvancılıkta kullanılması, bu süreçleri daha verimli ve daha kolay yönetilebilir hale getirmiştir. Özellikle ivmeölçer gibi giyilebilir sensör teknolojileri, hayvanların davranışlarının anlık olarak izlenmesini ve analiz edilmesini mümkün kılmaktadır [2]. Bu veriler genellikle büyük miktarlarda ve karmaşık yapıda olduğundan, anlamlı bilgiye dönüştürülmeleri için gelişmiş veri işleme ve analiz yöntemlerine ihtiyaç duyulmaktadır. Bu sensörler sayesinde hayvanların yürüme, dinlenme ve geviş getirme gibi temel davranışları sınıflandırılabilir. Bu davranışların sınıflandırılması ile anomaliler tespit edilebilir ve sürü yönetimi optimize edilebilir.

Bir hayvanın anormal bir davranış sergilemesi sağlık sorunlarının erken teşhisinde hayati bir rol oynayabilir. Benzer şekilde bir sürünün hareket ve dinlenme düzenlerinin gözlemlenmesi, çevresel ve beslenme koşullarının sürü üzerindeki etkilerinin değerlendirilmesine olanak tanımaktadır. Doğru bir şekilde yapılan davranış izleme ile hastalıkların erken teşhis edilmesini, tedavi süreçlerinin hızlandırılmasını ve hayvanların

genel performansının artırılmasını sağlamak mümkündür. Bu da hem ekonomik verimlilik hem de hayvanların yaşam kalitesi açısından önemli bir avantaj sunmaktadır.

Hayvanların hareket miktarında artış genellikle kızgınlık dönemine girdiklerinin bir göstergesi olarak kabul edilir. Bu bilgi suni tohumlama veya çiftleşme zamanlamasının doğru bir şekilde belirlenmesini sağlar. Örneğin bir ineğin kızgınlık döneminde yürüyüş aktivitesindeki %80'e varan artış, çiftlik yönetimine üreme planlaması için kesin veriler sunmaktadır [3]. Kızgınlığın kaçırılması, süt üretimi yapan işletmelerde laktasyon dönemini uzatabilir ve işletmeye önemli mali kayıplar yaşatabilir. Davranış analizi ile bu sürecin kontrol altına alınması üreme başarısını artırmaktadır. Bir hayvanın düşük hareketlilik göstermesi sağlık problemlerinin erken bir işareti olabilmektedir. Özellikle ayak hastalıkları, topallık veya metabolik bozukluklar gibi sorunlar bu tür davranışlarla ilişkilendirilmektedir [4]. Geviş getirme davranışında belirgin bir azalma, sindirim problemlerine veya enerji yetersizliğine işaret edebilir. Enerji yetersizliği ineklerin süt oranında ciddi azalmalar yaratabilmektedir. Doğum sonrası geviş getirme sürelerinin takip edilmesi, doğum sonrası komplikasyonların erken teşhis edilmesine olanak sağlamaktadır [5]. Hayvanların yem yeme ve otlama sürelerinin izlenmesi yem israfını azaltmaya ve yem verimliliğini artırmaya yardımcı olur. Hayvanların otlama sürelerindeki artış yem tüketiminin azalmasıyla sonuçlanabilir. Bu da yem stratejilerinin yeniden değerlendirilmesini gerektirir [5].

Bu çalışmada, ivmeölçer sensörlerinden elde edilen büyükbaş hayvan davranış verileri üzerinde sinyal işleme ve makine öğrenmesi yöntemleri uygulanarak davranışların tahmin edilmesi amaçlanmaktadır. Bu bağlamda verilerin işlenmesi ve özellik çıkarımı aşamaları ele alınacaktır. Özellik çıkarımı sonrasında elde edilen özellikler hem geleneksel sınıflayıcılar hem de uzun kısa süreli bellek (Long Short Term Memory- LSTM) modelleri üzerinde eğitilerek davranış tahmini yapılacaktır. Model eğitimi sırasında özellik seçimi uygulanacak ve yalnızca gerekli özellikler kullanılacaktır. Modelin performansı optimize edilerek gereksiz işlemlerden arındırılması amaçlanmıştır. Araştırmanın temel amacı, büyükbaş hayvanların günlük aktivitelerinin otomatik olarak sınıflandırılması için etkili bir model geliştirmektir. Böylece, hayvancılık faaliyetleri daha verimli bir şekilde yönetilebilirken, hayvan refahı da iyileştirilebilecektir.

Balasso, Marchesini, Ughelini, Serva ve Andrighetto (2021) çalışmalarında, büyükbaş süt

ineklerinde davranış ve postür tespiti için, büyükbaş süt ineklerinin sol yan bölgesine yerleştirilen bir üç eksenli ivmeölçer kullanmıştır [6]. Çalışmada, 5 farklı davranış (hareket, dinlenme, beslenme, geviş getirme, hareketsiz durma) ve 3 farklı postür (ayakta, sağ sternal yatar, sol sternal yatar) sınıflandırılmıştır. Rastgele orman (Random Forest-RF), K-en yakın komşular (K-Nearest Neighbors-KNN) Aşırı Gradyan Artırma (eXtreme Gradient Boosting-XGB) ve Destek Vektör Makinesi (Support Vector Machine-SVM) gibi makine öğrenimi algoritmaları kullanılarak davranış ve postür sınıflandırması yapılmış, özellikle postür tespitinde 99,2% doğruluğa ulaşılmıştır. Davranış tespitinde ise %76 gibi bir başarı elde edilmiştir. Çalışma düşük frekanslı (5 Hz) veri toplama ve 8 saniyelik pencere uzunluğu ile veri analizine yönelik yenilikçi bir yaklaşım sunmaktadır. Bu çalışma, süt ineklerinde davranış ve postür tespiti için başarılı bir temel oluşturarak, düşük maliyetli ve yüksek doğruluklu sınıflandırma yöntemleri geliştirilmesine katkı sağlamaktadır. Ayrıca farklı algoritmaların davranış tespiti üzerindeki etkilerinin detaylı bir şekilde incelenmesi, hayvan davranışı analizinde hangi yöntemlerin daha uygun olduğunu göstermesi açısından önemlidir.

Martono, Sawado, Nonaka, Terada ve Ohwada (2023) çalışmalarında, kapalı ortamda yaşayan büyükbaş sığırların davranışlarını sınıflandırmıştır [7]. Yaptıkları çalışmada, sığırların boyunlarına yerleştirdikleri 3 eksen ivmeölçer sensörlerini kullanmıştır. Veriler yem yeme ve geviş getirme gibi temel davranışların analiz edilmesi için 10, 30, 60 ve 180 saniyelik zaman pencerelerine ayrılmıştır. Çalışma, 15 farklı özelliğin çıkarımı ile etiketli bir veri seti oluşturmuş ve dört farklı makine öğrenimi algoritmasını RF, XGB, Karar Ağacı (Decision Tree-DT), Lojistik Regresyon (Logistic Regression-LR) kullanarak sınıflandırma performansını değerlendirmiştir. Sonuçlar, daha kısa zaman pencerelerinin (%10 ve %30) daha iyi sınıflandırma doğruluğu sağladığını göstermiştir. Bu çalışma, hayvan davranışlarını anlamada sensör teknolojilerinin ve makine öğrenimi algoritmalarının potansiyelini vurgularken, pencereleme seçimi hakkında temel bilgi sağlamaktadır.

Pavlovic ve diğerleri (2022) çalışmalarında, boyun bölgesine monte edilmiş ivmeölçer tabanlı cihazları kullanarak yeme, geviş getirme ve diğer davranışların sınıflandırmıştır [8]. Çalışma 18 sığırdan elde edilen verilerle gerçekleştirilmiştir. Bu veriler farklı davranış sınıflarını doğru bir şekilde ayırt etmek için çeşitli makine öğrenimi algoritmalarını eğitmekte kullanılmıştır. Bu çalışmada, özellikle boyun hareketlerinden türetilen özelliklerin seçilmesine odaklanılmış ve karşılıklı bilgi (Mutual Information-MI) ile geriye

dönük özelliğın ortadan kaldırılması (Backward Feature Elimination-BFE) gibi özellik seçme yöntemleri kullanılarak veri boyutlarının düşürülmesi sağlanmıştır. Doğrusal diskriminant analizi (Linear Discriminant Analysis-LDA) algoritması, hem yüksek doğruluk (%83) hem de düşük hesaplama maliyeti sunması nedeniyle en uygun model olarak belirlenmiştir. Çalışma, düşük maliyetli cihazlar üzerinde kullanılabilir kadar hafif ve etkili bir model sunmaktadır.

Riaboff ve diğerleri (2020) çalışmalarında, süt ineklerinin otlama, yürüme, geviş getirme (yatar ve ayakta) gibi davranışlarını sınıflandırmak için üç eksenli ivmeölçer verilerini kullanmıştır [9]. Boyuna monte edilen sensörlerden elde edilen veriler, %90 örtüşme ile 10 saniyelik pencereye bölünerek analiz edilmiştir. Çalışma, XGB algoritması ile %98 doğruluk ve %96 Cohen's Kappa gibi yüksek başarı oranları elde etmiştir. Ayrıca, zaman serisi analizlerinde Gizli Markov Modeli (Hidden Markov Model-HMM) tabanlı Viterbi algoritması kullanılarak zamansal yapının daha iyi anlaşılması sağlanmıştır.

Wu ve diğerleri (2022) çalışmalarında, büyükbaş hayvan davranışlarının tanımlanması için derin öğrenme yöntemleri kullanarak yeni bir yaklaşım sunmuşlardır [10]. Çalışmalarında, 12 büyükbaş hayvanın hareket verilerini izlemek için dokuz eksenli bir ataletsel ölçüm birimi (IMU) kullanılmıştır. Davranış sınıflandırması, geleneksel LSTM-RNN ve derin rezidüel çift yönlü LSTM (Bidirectional LSTM) modelleriyle gerçekleştirilmiştir. Çalışmada, altı farklı davranış türü (yem yeme, yatma, ruminasyon, boyun kaşıma, bacak kaşıma ve sosyal yalama) tanımlanmıştır. Derin rezidüel çift yönlü LSTM modeli, özellikle deri hastalıklarıyla ilişkili davranışların sınıflandırılmasında yüksek bir doğrulukla (%94,9) en iyi performansı göstermiştir.

Bu çalışmada kullanılan veri seti ile yapılan 4 adet çalışma bulunmaktadır. Bu çalışmalardan birini gerçekleştiren Li ve Chai (2023), AnimalAccML isimli kullanıcı dostu bir grafik arayüz geliştirmiştir [11]. Bu arayüz, 3 eksen ivmeölçer verilerini kullanarak davranış analizi ve makine öğrenimi modeli geliştirme süreçlerini otomatikleştirmiştir. Bu çalışma, pencere boyutu ve adım boyutunun etkilerini optimize ederek davranış sınıflandırmasında yüksek doğruluk elde etmeyi amaçlamıştır. Ayrıca azınlık sınıflarını birleştirerek veri dengesizliği problemini çözmek için yenilikçi yaklaşımlar sunmuştur. Çalışmalarında 97,2% doğruluk oranı elde etmişlerdir. Li ve diğerleri (2021) çalışmalarında, sınırlı etiketli veriyle çalışan derin öğrenme modellerinde veri artırma

yöntemlerinin önemini vurgulamıştır [12]. En iyi sınıflandırma performansını, bilgi kaybını telafi etme (Compensate for Information Loss-CIL) ve geri alma (Reversal-REV) yöntemlerini kullanarak F1 Skoru 94,43% elde etmiştir. Bu yöntemler özellikle küçük ölçekli veri setlerinde derin öğrenme performansını artırmak için uygun maliyetli çözümler sunmaktadır. Bartels ve diğerleri (2022), TinyCowNet isimli düşük enerji tüketimli RNN modelleri geliştirmiştir [13]. Bu modeller, küçük uç cihazlarda kullanılabilir ve uzun süreli batarya ömrüyle davranış izlemeye olanak tanır. Çalışma, davranışların zaman içindeki dağılımını tahmin etmek için regresyon algoritmalarını uygulamış ve 95,2% doğruluk oranına ulaşmıştır. Ayrıca, farklı davranışların aynı pencere içinde karıştığı durumlarda, doğru sınıflandırma yapılabilmesi için yenilikçi yöntemler önermiştir. Bu çalışmalar, sığır davranışlarının izlenmesi ve sınıflandırılması alanında yenilikçi teknolojileri kullanarak elde edilen ilerlemeleri ortaya koymaktadır. Özellikle veri dengesizliği, enerji tüketimi ve kullanıcı dostu arayüzler gibi problemlere yönelik çözüm önerileri sunulmuş, bu da tarım ve hayvancılık sektöründe hassas hayvancılık uygulamalarının yaygınlaşmasına katkı sağlamıştır.

Bu tez çalışması, literatürde yer alan mevcut yaklaşımların ötesine geçerek daha hassas ve genellenebilir sonuçlar elde etmeyi hedeflemektedir. Kullanılan veri seti, hızlanma sensörlerinden elde edilen ve farklı davranış sınıflarını temsil eden etiketli verilerdir. Çalışma hem sinyal işleme hem de makine öğrenmesi yöntemlerini bir araya getirerek bu verilerin etkin bir şekilde analiz edilmesini ve sınıflandırılmasını sağlayacaktır.

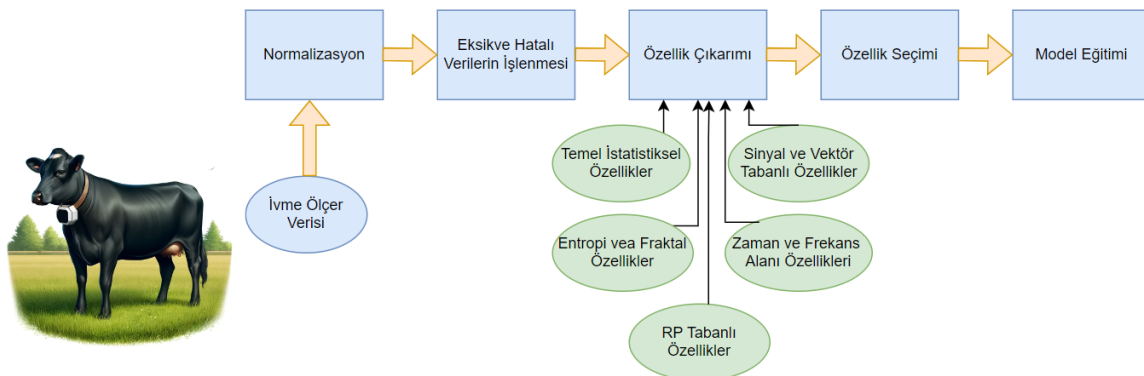
Sonuç olarak, bu araştırmanın hayvancılık sektöründe teknoloji kullanımının artırılmasına ve hayvan davranışlarının daha iyi anlaşılmasına katkı sağlaması beklenmektedir. Ayrıca geliştirilen yöntemlerin, benzer veri setleri üzerinde uygulanabilir olması diğer araştırmalara da ışık tutacaktır.



2. MATERYAL VE YÖNTEM

Bu çalışmada, Japon Siyah Sığırlarına ait ivmeölçer verileri kullanılarak hayvan davranışlarının doğru bir şekilde sınıflandırılması amaçlanmıştır. Çalışmanın temel amacı hem geleneksel makine öğrenimi algoritmalarını hem de LSTM gibi derin öğrenme modellerini kullanarak davranış sınıflandırma performanslarını karşılaştırarak en uygun yöntemi belirlemektir. Bu doğrultuda veri işleme, özellik çıkarımı, özellik seçimi ve model eğitimi gibi yöntemler uygulanmıştır.

Bu çalışmada ivmeölçer verilerinden hayvan davranışlarını sınıflandırmak amacıyla veri ön işleme adımı gerçekleştirilmiştir. Bu işlem ile verilerin anlamlı özelliklere dönüştürülmesini ve modellerin daha yüksek doğrulukla çalışmasının sağlanması hedeflenmiştir.



Şekil 2.1. Hayvan davranışlarının sınıflandırılmasında izlenen sürecin genel gösterimi

Şekil 2.1.'de ilk adımı, hayvanların günlük hareket verilerini ivmeölçer sensörleri ile toplanmış veri seti oluşturmuştur. Ardından veri ön işleme aşamasında ham veriler analiz ve modelleme için uygun hale getirilmiştir. Bu aşamada normalizasyon işlemi uygulanmış ve eksik veya hatalı veriler temizlenmiştir. Özellik çıkarımı aşamasında ise temel istatistiksel özellikler, entropi ve fraktal tabanlı özellikler, zaman ve frekans alanına dayalı özellikler ve yineleme grafiği (Recurrence Plot- RP) tabanlı özellikler hesaplanmıştır. Daha sonra, çıkarılan özellikler arasından en anlamlı olanlar özellik seçimi gerçekleştirilerek belirlenmiştir. Son olarak, seçilen özellikler kullanılarak RF, ekstra ağaçlar (Extra Trees-ET) ve LSTM gibi algoritmalarla model eğitimi yapılmıştır.

2.1. Veri Seti ve Kullanılan Teknolojiler

2.1.1. İvmeölçer

İvmeölçerler, hızlanmayı algılamak ve ölçmek için kullanılan hassas cihazlardır. Mikro-elektro-mekanik sistemler (MEMS) tabanlı ivmeölçerler, küçük boyutları ve düşük maliyetleri ile geniş uygulama alanlarına sahiptir. Temel olarak, bir kütlenin bir yay üzerinde hareket etmesine dayalı olarak çalışan bu sensörler, yerçekimi kuvveti ve titreşimleri algılayarak eğim, hızlanma ve diğer dinamikleri ölçebilmektedir [14]. Newton'un ikinci yasasına dayanan MEMS ivmeölçerler, bir kütlenin yer değiştirmesi veya titreşimi sırasında oluşan değişiklikleri algılar. Çoğu ivmeölçer, bu hareketlerin oluşturduğu kapasitans değişikliklerini ölçer ve bu verilerden hızlanma ve konum bilgisi türetir. Bu sensörlerin temel prensiplerinden biri, kütlenin frekans değişikliklerinin gerilim seviyelerine dönüştürülmesidir [15].

Kartezyen koordinat sistemi, üç boyutlu uzayda bir cismin konumunu ve hareketini tanımlamak için kullanılan temel bir referans sistemidir. Bu sistem birbirine dik üç eksen oluşmaktadır. Eksen, bir cismin uzaydaki hareketini ve konumunu ifade etmek için kullanılan hayali doğrulardır. Kartezyen koordinat sistemi ise bu doğruların birbirine dik şekilde uzandığı bir sistemdir. Bu eksenler X, Y ve Z olarak isimlendirilmektedir. X eksen genellikle yatay doğrultuda sağa ve sola uzanır, Y eksen ileri ve geri yönündeki hareketleri temsil eder ve Z eksen ise yukarı ve aşağı yönündeki hareketleri ifade eder. Bu eksenler, bir cismin üç boyutlu bir uzaydaki konumunu belirlemek için kullanılan referans noktalarını oluşturur. Her eksen, cismin farklı bir yöndeki hareketini veya hızlanmasını ölçmek için ayrı bir bileşen sağlar.

MEMS ivmeölçerler genellikle üç eksen (X, Y ve Z) ölçüm yapma kapasitesine sahiptir ve bu sayede bir cismin hareket dinamiklerini kapsamlı bir şekilde analiz edebilmektedir. Bu cihazlar sadece doğrusal hızlanmayı değil, aynı zamanda dönme hareketleri gibi farklı hareketleri de algılayabilir. Küçük boyutları sayesinde taşınabilir cihazlarda ve enerji kısıtlaması olan uygulamalarda tercih edilmektedirler. Bunun yanı sıra yüksek hassasiyet ve düşük enerji tüketimi gibi özellikleri, ivmeölçerleri modern teknolojinin vazgeçilmez bir parçası haline getirmiştir. Ancak sıcaklık değişimleri, manyetik alanlar ve titreşim gibi çevresel faktörler sensör performansını etkilemektedir. Bu durumlarda gelişmiş sinyal

işleme teknikleri ve kalibrasyon yöntemleri kullanılarak doğruluk artırılabilir.

2.1.2. Japon siyah sığır ineği davranış sınıflandırması veri seti

Bu çalışmada, Japon Siyah Sığırlarının davranışlarını sınıflandırmak amacıyla toplanmış üç eksenli (X, Y ve Z) ivmeölçer verilerini içermektedir. Çalışmada kullanılan veri seti [16] tarafından sağlanmıştır. Veri seti her biri 16-bit çözünürlüğe ve $\pm 2g$ hassasiyete sahip Kionix KX122-1037 ivmeölçerlerin altı farklı Japon Siyah Sığırının boynuna yerleştirilmesiyle oluşturulmuştur. Veri toplama işlemi, 12 Haziran 2020 tarihinde Japonya'nın Nagano şehrinde bulunan Shinshu Üniversitesi'ndeki bir çiftlikte gerçekleştirilmiştir. Sığırlar serbest dolaşım alanları olan çim alan ve ahırlarda gözlemlenmiştir. Bu süreç boyunca, ivmeölçerlerden alınan veriler, Sony FDR-X3000 4K video kameralarla çekilen görüntülerle eşleştirilmiştir.

Veri toplama sürecinde davranışların doğru bir şekilde etiketlenebilmesi için video kayıtları davranış uzmanları ve gözlemciler tarafından analiz edilmiştir. Bu süreç toplamda 69 kişi-saat sürmüştür. Ham olarak toplanan 567 dakikalık verinin, yüksek kaliteli 197 dakikalık etiketli veri haline getirilmesi için üç gözlemcinin çoğunluk oylamasına dayalı bir etiketleme yöntemi kullanılmıştır. Bu etiketleme işlemi sonucunda, veri seti 13 farklı davranış kategorisine ayrılmıştır. 13 farklı davranış kategorisine ayrılmıştır. Bu davranışlar, hayvanların dinlenme, yürüme, otlama ve geviş getirme gibi aktivitelerini kapsamaktadır. Her bir davranışın örnek sayısı Çizelge 2.1.'de gösterilmektedir.

Veri toplama sürecinde kullanılan ivmeölçer saniyede 25 örnek (25Hz) toplama kapasitesine sahiptir. Elde edilen veriler, altı farklı dosyada (cow1.csv-cow6.csv) saklanmıştır. Her dosya, ilgili sığırın zaman damgası, üç eksenli hızlanma verileri (AccX, AccY, AccZ) ve davranış etiketlerini içermektedir. Veri setindeki sütunların tanımları Çizelge 2.2.'de verilmiştir.

Veri toplama sürecinde, Küresel Konumlama Sistemi (Global Positioning System-GPS) sinyali ile zaman damgaları düzeltilmiş ve yüksek doğruluk elde edilmiştir. Ayrıca bu veri setinin etik standartlara uygunluğu, Shinshu Üniversitesi Hayvan Bakım ve Kullanım Komitesi tarafından onaylanmıştır.

Çizelge 2.1. Veri setindeki çeşitli davranışlar ve ineklere göre dağılımları

	İnek 1	İnek 2	İnek 3	İnek 4	İnek 5	İnek 6	Toplam	Davranış Tanımı
RES	35 814	47 059	20 501	15 735	11 025	19 996	150 130	Ayakta dinlenme
RUS	1 620	25 930	11 156	14 523	0	0	53 229	Ayakta geniş getirme
MOV	6 376	8 437	7 532	17 248	4 846	5 760	50 199	Hareket
GRZ	2 416	2 199	0	2 707	2 442	7 849	17 613	Otlanma
SLT	204	0	10 654	0	0	0	10 858	Tuz yalama
FES	6 809	0	0	0	1 125	0	7 934	Yemlikte yeme
DRN	1 176	0	1 300	0	0	0	2 476	Su içme
LCK	0	0	649	297	0	356	1 302	Yalama
REL	0	360	0	404	0	0	764	Yatarak dinlenme
URI	239	0	383	0	0	0	621	İdrar yapma
ATT	57	50	0	62	0	197	366	Saldırma
ESC	0	0	0	128	0	0	128	Kaçma
BMN	0	54	0	0	0	0	54	Çiftleşme

Çizelge 2.2. Veri setinde yer alan ivmeölçer verilerinin sütunları ve açıklamaları

TimeStamp_UNIX	TimeStamp_JST	AccX [g]	AccY [g]	AccZ [g]	Label [-]
GPS Zaman Damgası	JST Zaman Damgası	X Eksenini	Y Eksenini	Z Eksenini	Davranış

2.2. Normalizasyon

Normalizasyon, çeşitli ölçüm araçları ve/veya sensörlerle elde edilmiş verilerin belirli bir aralıkta yeniden ölçeklendirilmesidir. Amaç veriler kaynaklarındaki değişimlerin belirli bir aralık içinde değerlendirilmesi ile hesaplanacak özelliklerin bu değişimlerden kaynaklı yanlış değerler üretmesini önlemektir. Normalizasyon, özelliklerin farklı değer aralıklarında olması durumunda, makine öğrenimi algoritmalarının bu farklılıklar nedeniyle yanlış öğrenme veya performans kaybı yaşamamasını önlemeyi amaçlar [17]. Özellikle gradyan tabanlı öğrenme algoritmaları veya uzaklık tabanlı algoritmalar kullanan modellerde normalizasyon son derece kritik bir adımdır. Bir veri setinde her özellik farklı ölçüm birimlerine ve aralıklara sahip olabilir. Büyük aralıklara sahip özellikler, küçük aralıklara sahip özelliklere göre modelin daha fazla dikkatini çekmektedir. Bu durum modelin dengeli öğrenmesini engellemektedir [18].

$$x_{norm} = \frac{x - \min(x)}{\max(X) - \min(X)} \quad (2.1)$$

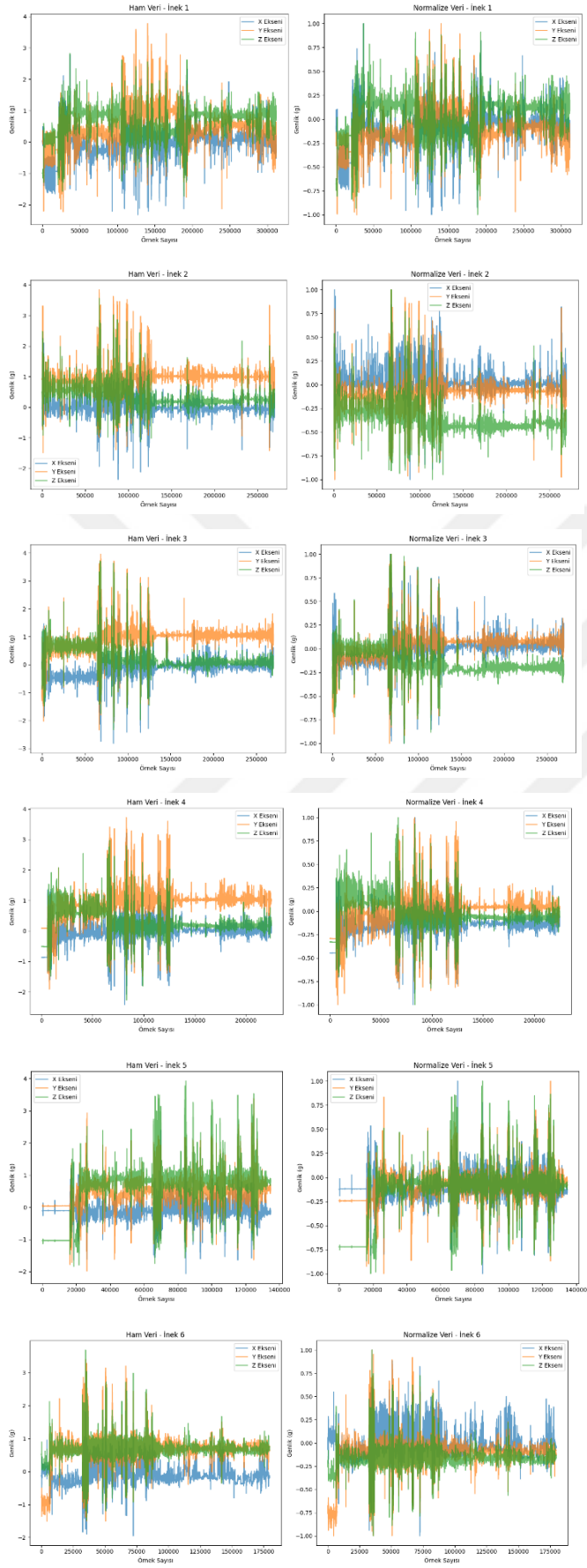
Bu çalışmada Min-Max Normalizasyonu yapılmış olup, sinyaller $[-1,1]$ aralığında normalize edilmiştir. Özellikle ivmeölçer verilerinin x, y ve z eksenlerindeki ölçüm aralıkları göz önüne alındığında, bu yöntem verilerin $[-1, 1]$ aralığında temsil edilmesini sağlamış ve sınıflayıcı modellerin kararlı çalışmasına katkıda bulunmuştur. Her bir eksendeki (AccX, AccY, AccZ) veriler için minimum ve maksimum değerler bulunmuştur.

Bu çalışmada kullanılan 6 farklı ineğe ait ivmeölçer sensör verilerinin ham ve normalize edilmiş halleri Şekil 2.2.'de gösterilmiştir. Her bir grafik çiftinde sol tarafta ham veriler (X, Y, Z eksenleri) sunulmuş, sağ tarafta ise bu verilerin Min-Max Normalizasyon yöntemi ile $[-1, 1]$ aralığına ölçeklenmiş hali gösterilmiştir. Ham verilerde her eksendeki ölçümlerin farklı aralıklarda olduğu ve bu durumun özellikler arasında ölçek farklılıklarına yol açtığı açıkça görülmektedir. Bu tür farklılıklar makine öğrenimi algoritmalarında modelin belirli özelliklere aşırı odaklanmasına neden olabilir ve dolayısıyla öğrenme sürecinde dengesizliklere yol açabilir.

Normalize edilmiş verilere bakıldığında tüm eksenlerin aynı ölçek aralığında temsil edildiği ve bu sayede modelin tüm özelliklere eşit derecede önem verebileceği bir yapı oluşturulduğu anlaşılmaktadır. Özellikle zaman serisi verilerde ölçek farklılıklarını gidermek algoritmaların daha kararlı çalışmasını sağlamaktadır. Bu durum hem model doğruluğunu artırmakta hem de eğitim sürecinin hızını optimize etmektedir.

2.3. Pencereleme

Pencereleme, zaman serisi verilerini belirli uzunluklardaki segmentlere (pencerelere) bölme işlemidir. Zaman serisi verileri genellikle sürekli bir veri akışı içerir ve bu veriler belirli zaman aralıklarında analiz edilerek anlamlı özellikler çıkarılmaktadır. Pencereleme hem veri boyutunu azaltır hem de verilerdeki zamana bağlı değişimlerin modellenmesine yardımcı olmaktadır [19]. Zaman serisi verilerinin doğası gereği belirli olaylar (bir adım atma hareketi veya kısa bir duraklama) yalnızca belirli zaman aralıklarında meydana gelmektedir. Hareket veya davranışların başlangıç ve bitiş noktalarının doğru tespiti pencerelerin yeterli hassasiyette olması ile sağlanabilmektedir.



Şekil 2.2. Ham ve normalize edilmiş verilerin karşılaştırılması

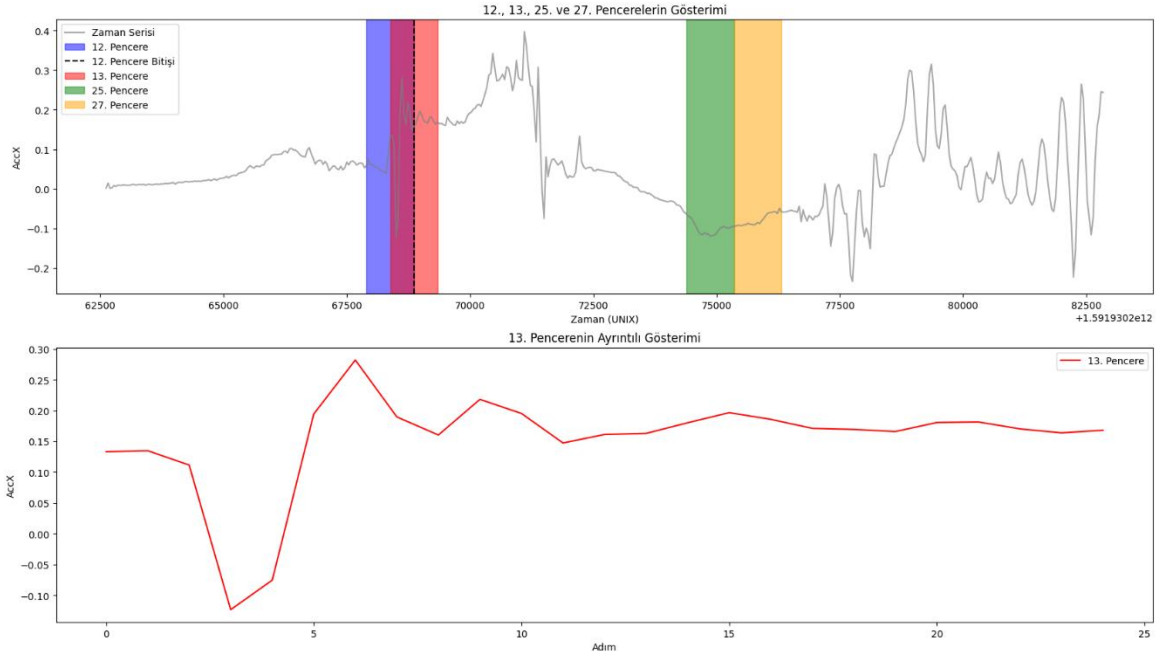
Bu olayların doğru bir şekilde analiz edilmesi ve tespit edilmesi için verilerin uygun uzunlukta pencerelere bölünmesi gerekmektedir. Hareketlerin hızlı değişim gösterdiği ve detaylı analiz gerektirdiği durumlarda kısa uzunluktaki pencerelerin kullanımı oldukça önemlidir. Hayvanların dinlenme, hareket etme veya sıçrama gibi davranışlarının ayrıştırılabilmesi, bu tür kısa pencerelerle daha hassas bir şekilde mümkündür. Pencereleme ile her bir segment üzerinde özellikler (ortalama, standart sapma, maksimum değer) çıkarılarak veriler daha anlamlı hale getirilmektedir. Çıkarılan bu veriler modeller için kullanılmaktadır [20].

Bu çalışmada 25 Hz örnekleme hızına sahip ivmeölçer verileri kullanılmıştır. Bu, saniyede 25 veri noktası toplandığı anlamına gelir. Genel analize göre, 1-2 saniyelik aralığın tanıma hızı ile doğruluk arasındaki en iyi dengeyi sağladığı gözlemlenmiştir [21]. Bu çalışmada, 1 saniyelik pencere uzunluğu tercih edilmiştir. Her bir pencerede 25 veri noktası bulunmaktadır. Pencere zaman serisinin toplam uzunluğuna göre %50 örtüşme ile ilerletilmiş ve bir pencerenin yarısı bir sonraki pencereye dahil edilmiştir. Bu yaklaşım bir önceki pencerenin yarısını bir sonraki pencereye taşıyarak veri kaybını en aza indirmekte ve veri sürekliliğini korumaktadır. Hayvanın ivme değişimi sırasında hareketin başlangıç ve bitiş noktalarının birden fazla pencereye dahil edilmesi sağlanır ve sürekliliğin korunması sağlanmaktadır. Ayrıca hareketlerin daha doğru bir şekilde analiz edilmesine de olanak tanımaktadır [20].

Şekil 2.3.'te zaman serisi verilerinin pencereleme sürecine ilişkin bir örnek görselleştirme yer almaktadır. Bu görselleştirme, zaman serisi verilerinin belirli aralıklardaki segmentlere nasıl bölündüğünü ve bu segmentlerin analizine yönelik yöntemleri açıklamaktadır. Şekilde, özellikle 12. ve 13. pencereler gösterilmiştir. Bu pencerelerin %50 örtüşme ile ilerlediği ve birbirini tamamlayacak şekilde veri sürekliliğini koruduğu gözlemlenmektedir. Ayrıca 12. pencerenin bitiş noktası, siyah bir kesikli çizgiyle belirtilmiş ve bu pencerelerin sınırlarının net bir şekilde gösterilmesi sağlanmıştır. Ayrıca Şekil 2.3.'te 13. pencere ayrı bir alt grafikte daha ayrıntılı olarak gösterilmektedir. Çalışmada tüm pencereler ayrı ayrı incelenip ileride bahsedilecek olan tüm özellikler hesaplanmıştır.

Sonuç olarak bu çalışmada kullanılan pencereleme yöntemi, hayvan davranışlarının zaman içindeki değişimlerini detaylı bir şekilde analiz etmek ve sınıflandırma modellerinin doğruluğunu artırmak için etkili bir araç olarak kullanılmaktadır. Bu yöntem özellikle

hareket ve davranış analizi gibi uygulamalar için ideal bir çerçeve sunmaktadır.



Şekil 2.3. Veri setinden örnek pencereleme gösterimi.

2.4. Eksik ve Hatalı Verilerin İşlenmesi

Eksik ve hatalı veriler, veri analizi ve makine öğrenimi süreçlerinde yaygın olarak karşılaşılan sorunlardan birisidir. Ham veriler, sensör hataları, veri aktarım problemleri veya diğer teknik aksaklıklar nedeniyle eksik ya da hatalı değerler olabilir. Bu tür veriler makine öğrenimi modellerinin performansını olumsuz etkiler ve yanlış sonuçlara yol açabilir. Eksik veya hatalı veriler, özellikle zaman serisi verileri gibi düzenli bir yapıya sahip olan veri setlerinde daha büyük bir problem oluşturmaktadır. Çünkü bu tür verilerde eksiklikler analizlerdeki sürekliliği bozabilir ve modelin doğruluğunu düşürebilmektedir. Bu çalışmada, eksik ve hatalı verilerin analizi ve işlenmesi için aşağıdaki adımlar izlenmiştir:

Veri setindeki sonsuz (inf) ve boş (nan) değerler, Python'da isnull() ve isinf() gibi fonksiyonlar kullanılarak tespit edilmiştir [22]. Bu adım, veri setindeki sorunlu noktaların belirlenmesini sağlamıştır. Tespit edilen sonsuz ve boş değerler, ilgili sütunun ortalama değeriyle değiştirilmiştir [23]. Ortalama değer kullanılması veri setinin genel istatistiksel yapısını korumuştur. Bir sütundaki eksik bir değer (Eş. 2.2) ile doldurulmuştur.

$$x_{new} = \frac{\sum_{i=1}^n x_i}{n} \quad (2.2)$$

Bu yöntem özellikle zaman serisi verilerinde verinin doğal yapısını korumak ve sürekliliği sağlamak için etkili bir çözüm sunmaktadır.

Ortalama ile doldurma nispeten basit bir yöntemdir ve küçük miktarda eksik veriye sahip veri setlerinde etkili sonuçlar sağlamaktadır [23]. Ancak eksik verilerin büyük oranlarda olduğu durumlarda, interpolasyon yöntemleri veya makine öğrenimi tabanlı eksik veri tamamlama algoritmaları da kullanılabilir. Bu çalışmada eksik veriler nispeten az olduğu için ortalama ile doldurma yöntemi etkili ve yeterli bir yöntem olmuştur.

2.5. Özellik Çıkarımı

Özellik çıkarımı, ham verilerden anlamlı ve işlenebilir bilgilerin elde edilmesi sürecidir. Bu işlem veri boyutunu azaltırken aynı zamanda verilerin daha kompakt ve model için daha anlamlı hale getirilmesini sağlamaktadır. Makine öğrenimi modellerinin doğruluğunu ve etkinliğini artırmak için doğru ve etkili özelliklerin seçimi kritik öneme sahiptir. Özellikle zaman serisi verilerinde özellik çıkarımı, verinin dinamiklerini anlamak için gereklidir ve her pencere için özet bilgiler sağlayarak daha verimli bir model eğitimi sunmaktadır.

Bu çalışmada geleneksel istatistiksel metrikler ve zaman serisi analizi için kullanılan ileri düzey yöntemler bir arada kullanılmıştır. Yineleme Grafiği (Recurrence Plot-RP) tabanlı metrikler ekleyerek, sinyalin zaman içindeki tekrar eden yapısını analiz etmek ve literatürde kullanılanlardan farklı özelliklerin eldesi amaçlanmıştır. Özelliklerin analiz ve sınıflandırma sürecinde etkili bir şekilde kullanılabilmesi amacıyla elde edilen tüm özellikler beş ana gruba ayrılmıştır. Bu ana gruplar temel istatistiksel özellikler, sinyal ve vektör tabanlı özellikler, entropi veya fraktal tabanlı özellikler, zaman frekans alanı özellikleri ve RP tabanlı özelliklerdir. Toplam çıkarılan özellik sayısı 109'dur.

Bu gruplama, farklı analiz ihtiyaçlarını karşılamak ve verinin hem basit hem de karmaşık yapısını modellemek için yapılmıştır. Temel istatistiksel özellikler sinyalin genel yapısını özetlerken, fraktal ve entropi tabanlı metrikler daha karmaşık ve kaotik yapıları incelemeye olanak tanımaktadır. Sinyal ve vektör tabanlı özellikler sinyalin büyüklüğü, yön bilgisi ve

enerji dağılımı gibi özelliklerini değerlendirir. Zaman frekans alanı özellikleri hayvan davranışlarının frekans uzayındaki dinamiklerini analiz etmeye odaklanırken, RP tabanlı metrikler sinyalin zaman içindeki tekrar eden davranışlarını detaylı bir şekilde modellemeyi amaçlamaktadır. Bu veri setinin kullanıldığı önceki çalışmalarda kullanılanlardan farklı olarak RP tabanlı özellikler, zaman ve frekans alanı özellikleri ve entropi veya fraktal özellikler kullanılmıştır. Bu yöntemlerin birlikte kullanımını veri setinin farklı boyutlarını kapsayarak sınıflandırma performansını artırmak için etkili bir yaklaşım sunmuştur. Bu kadar farklı modeller kullanılmasının bir diğer sebebi de özellik seçimi işleminin gerçekleştiriliyor olmasıdır.

Çizelge 2.3.'te kullanılan tüm özellikler verilmiştir. Elde edilen tüm bu özellikler çalışmada sınıflandırma modeli için değerlendirilmiştir. Ancak özelliklerin tamamı kullanılmamış, model performansını optimize etmek için özellik seçimi yöntemi uygulanmıştır. Özellik seçimi süreci ve sonuçları ilerleyen bölümlerde detaylı olarak ele alınacaktır.

2.5.1. Temel istatistiksel özellikler

Bu bölümde zaman serisi verilerinden çıkarılan temel istatistiksel özelliklerin hesaplama yöntemleri açıklanmıştır. Bu özellikler, sinyalin genel yapısını özetlemek ve sınıflandırma modeline anlamlı bilgi sağlamak amacıyla kullanılmıştır.

Basıklık katsayısı

Basıklık katsayısı (kurtosis), bir veri dağılımının uç değerlerinin yoğunluğunu ölçen bir metriktir. Pozitif değerler, dağılımın sivri uçlara sahip olduğunu, negatif değerler ise daha düz bir yapıya sahip olduğunu gösterir.

$$Kurtosis^{axis} = \frac{N \sum_{j=1}^N (X_j - \bar{X})^4}{\sum_{j=1}^N (X_j - \bar{X})^2} - 3 \quad (2.3)$$

X_j : X, Y veya Z eksenindeki j veri noktası.

\bar{X} : Pencere içerisindeki verinin ortalama değeri

N : Pencere boyutu

-3: Fisher kurtosis için gereken normalizasyon sayısı

Çizelge 2.3. Çıkarılan özelliklerin listesi

Kategori	Özellikler
Temel İstatistiksel Özellikler	Basıklık Katsayısı (Kurtosis) X, Y, Z
	Medyan Mutlak Sapma (MAD) X, Y, Z
	Ortalama Değer (Mean) X, Y, Z
	Medyan Değer (Median) X, Y, Z
	Minimum Değer X, Y, Z
	Maksimum Değer X, Y, Z
	25. Yüzdellik Değer X, Y, Z
	75. Yüzdellik Değer X, Y, Z
	Çarpıklık Katsayısı (Skewness) X, Y, Z
	Toplam Değer X, Y, Z
	Standart Sapma X, Y, Z
	Varyans X, Y, Z
Sinyal ve Vektör Tabanlı Özellikler	Sinyal Büyüklük Alanı (Signal Magnitude Area)
	Vektör Büyüklüğü (Vector Magnitude)
	Enerji
	Ortalama Vektör Mesafesi
Entropi veya Fraktal Özellikler	Shannon Entropisi X, Y, Z
	Higuchi Fraktal Boyutu X, Y, Z
	Katz Fraktal Boyutu X, Y, Z
	Permütasyon Entropisi X, Y, Z
	Spektral Entropi X, Y, Z
	SVD Entropi (Singular Value Decomposition) X, Y, Z
	Yaklaşık Entropi X, Y, Z
	Örnek Entropi X, Y, Z
Zaman Frekans Alanı Özellikleri	Baskın Frekans (Dominant Frequency) X, Y, Z
	Karekök Ortalama (Root Mean Square) X, Y, Z
	Sıfır Geçiş Oranı (Zero Crossing Rate) X, Y, Z
RP Tabanlı Özellikler	Kontrast X, Y, Z
	Korelasyon X, Y, Z
	Yineleme Oranı (Recurrence Rate, RR) X, Y, Z
	Determinizm (DET) X, Y, Z
	Laminarity X, Y, Z
	Çapraz Yineleme Oranı (Diagonal Recurrence Rate, DRR) X, Y, Z
	Dikey Yineleme Oranı (Vertical Recurrence Rate VRR) X, Y, Z
	Yineleme Oranı Tabanlı Entropy X, Y, Z
	Yakalama Süresi (Trapping Time, TT) X, Y, Z
	Ortalama Çapraz Çizgi Uzunluğu (Mean Diagonal Line Length, Lmean) X, Y, Z
	İlk Yineleme Zamanı (First Recurrence Time, FRT) X, Y, Z
	Eğilim X, Y, Z

Medyan mutlak sapma

Medyan mutlak sapma (MAD), bir veri setindeki tüm değerlerin medyandan mutlak farklarının ortalamasıdır.

$$MAD_i^{axis} = \text{median}(|X_j - \text{median}(X_j)|), \quad j \in [i, i + N - 1] \quad (2.4)$$

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

X_j : X, Y veya Z eksenleri verileri

Ortalama değer

Ortalama değer, bir veri kümesinin merkezini temsil eden en temel istatistiksel ölçütlerden biridir. Zaman serisi analizinde, penceredeki tüm değerlerin aritmetik ortalamasını alarak hesaplanır. Bu metrik, sinyalin genel eğilimini anlamak için kullanılır.

$$Mean_i^{axis} = \frac{1}{N} \sum_{j=1}^{i+N-1} X_j \quad (2.5)$$

$axis$: Hesaplamanın yapıldığı X, Y veya Z eksen

N : Pencere boyutu

X_j : X, Y veya Z eksenleri üzerindeki j veri noktası.

Medyan değer

Medyan, bir veri kümesinin sıralanmış elemanlarının ortasında bulunan değeri temsil eder. Veri kümesi çift sayıda elemandan oluşuyorsa, ortadaki iki elemanın aritmetik ortalaması medyan olarak alınır. Medyan, özellikle veri kümesinde aşırı uç değerlerin etkisini azaltmak için kullanılır.

$$Median_i^{axis} = \text{median}(X_j), \quad j \in [i, i + N - 1] \quad (2.6)$$

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

X_j : X, Y veya Z eksenleri verileri

Minimum değer

Minimum değer, bir veri kümesindeki en küçük elemanı ifade eder. Zaman serisi verilerinde her bir pencere içindeki minimum değer ayrı ayrı hesaplanır. Bu özellik, sinyalin en düşük seviyesini anlamak için kullanılır.

$$Min_i^{axis} = \min(X_j), j \in [i, i + N - 1] \quad (2.7)$$

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

X_j : X, Y veya Z eksenleri verileri

Maksimum değer

Maksimum değer, bir veri kümesindeki en büyük elemanı ifade eder. Zaman serisi verilerinde her bir pencere içindeki maksimum değer ayrı ayrı hesaplanır. Bu özellik sinyalin en yüksek seviyesini anlamak için kullanılır.

$$Max_i^{axis} = \max(X_j), j \in [i, i + N - 1] \quad (2.8)$$

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

X_j : X, Y veya Z eksenleri verileri

Yüzdellik değer

25. Yüzdellik değer, pencere içerisindeki verilerin sıralanmasını ve sıralı veri kümesinden %25'lik bölgeye karşılık gelen değerini seçilmesini sağlar. Özellikle dağılımın alt kısımlarda bulunan değerlerini anlamak ve incelemek için kullanılır. Bu değer verinin alt çeyreğini temsil ederek küçük değerlerin analizi için önemli bir referans noktası oluşturur.

$$Q_i^{Lower} = X \left[\frac{(N + 1)}{4} \right] \quad (2.9)$$

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

X : X, Y veya Z eksenleri verileri

75. Yüzdellik değeri

75. Yüzdellik değeri, pencere içerisindeki verilerin sıralanmasını ve sıralı veri kümesinden %75'lik bölgeye karşılık gelen değeri seçilmesini sağlar. Özellikle dağılımın daha yüksek uç değerlerini anlamak ve incelemek için kullanılır. Bu değeri verinin üst çeyreğini temsil ederek büyük değerlerin dağılımı hakkında bilgi sağlar.

$$Q_i^{Upper} = X \left[\frac{3 \times (N + 1)}{4} \right] \quad (2.10)$$

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

X : X, Y veya Z eksenleri verileri

Çarpıklık katsayısı

Çarpıklık katsayısı (skewness), bir veri dağılımının simetrisini ölçen bir metriktir. Veri kümesinin ortalamaya göre simetrik olup olmadığını anlamak için kullanılır. Pozitif bir çarpıklık, dağılımın sağa eğimli olduğunu, negatif bir çarpıklık ise sola eğimli olduğunu gösterir. Çarpıklık katsayısı, pencere yöntemiyle her pencere için ayrı ayrı hesaplanabilir.

$$Skewness_i^{axis} = \frac{\sum_{j=i}^{i+N-1} (X_j - \bar{X}_i)^3}{(N-1) \cdot (\sigma_i^{axis})^3} - 3 \quad (2.11)$$

X_j : X, Y veya Z eksenleri verileri

\bar{X}_i : Pencere içerisindeki verinin ortalama değeri

σ_i^{axis} : Pencere içerisindeki standart sapma

N : Pencere boyutu

i : Pencerenin başlangıç indeksidir.

Toplam deęer

Toplam deęer, bir pencere içindeki tüm verilerin toplamını ifade eder. Zaman serisi analizinde, her pencere için toplam deęer ayrı ayrı hesaplanır. Bu özellik, sinyal gücü ve büyüklüğünü anlamada kullanılabilir.

$$Sum_i^{axis} = \sum_{j=i}^{i+N-1} X_j \quad (2.12)$$

X_j : X, Y veya Z eksenleri j noktasındaki verisi

N : Pencere boyutu

i : Pencerenin başlangıç indeksidir.

Standart sapma

Standart sapma, bir veri setinin ortalamadan ne kadar saptığını ölçen bir metriktir. Zaman serisi analizinde, her pencere için standart sapma ayrı ayrı hesaplanır. Bu özellik, sinyalin deęişkenliğini ve dalgalanma seviyesini analiz etmek için kullanılır.

$$Std_i^{axis} = \sqrt{\frac{1}{N-1} \sum_{j=i}^{i+N-1} (X_j - \bar{X}_i)^2} \quad (2.13)$$

X_j : X, Y veya Z eksenleri j noktasındaki verisi

\bar{X}_i : Pencere içerisindeki verinin ortalama deęeri

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

Varyans

Bir veri setindeki deęerlerin, ortalamadan sapmalarının karesinin ortalaması olarak tanımlanır. Zaman serisi analizinde, her pencere için varyans ayrı ayrı hesaplanır. Varyans, sinyalin yayılımını ve dalgalanma derecesini ölçer.

$$Var_i^{axis} = \frac{1}{N-1} \sum_{j=i}^{i+N-1} (X_j - \bar{X}_i)^2 \quad (2.14)$$

X_j : X, Y veya Z eksenleri j noktasındaki verisi

\bar{X}_i : Pencere içerisindeki verinin ortalama değeri

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

2.5.2. Sinyal ve vektör tabanlı özellikler

Sinyal ve vektör tabanlı özellikler, sinyalin genel yapısını ve hareketin dinamiklerini anlamak için önemli ipuçları sunar. Özellikle çok boyutlu sensör verilerinde, vektörel büyüklük ve sinyal enerji dağılımı gibi metrikler, hareketin kapsamlı bir şekilde incelenmesine olanak tanır.

Bu özellikler, yalnızca belirli bir eksendeki X, Y veya Z verilerinin analiziyle sınırlı kalmaz; aynı zamanda eksenler arasındaki ilişkileri değerlendirerek hareketin üç boyutlu bir yapısını da ortaya çıkarır. Örneğin, sinyal büyüklük alanı (Signal Magnitude Area-SMA) toplam hareket enerjisini ölçerken, vektör büyüklüğü (vector magnitude) sinyalin uzaysal büyüklüğünü analiz eder. Bunun yanı sıra bu tür özellikler farklı hareket türlerinin ayrıştırılmasına yardımcı olur ve sınıflandırma modellerinin doğruluğunu artırabilmektedir.

Sinyal büyüklük alanı

SMA, çok eksenli bir hareket verisinin toplam büyüklüğünü ölçmek için kullanılan bir metriktir. Tüm eksenlerdeki (X, Y ve Z) sinyalin mutlak değerlerinin toplamının alınmasıyla hesaplanır. Zaman serisi verilerinde, her kaydırmalı pencere için ayrı ayrı hesaplanır. SMA, toplam hareketi belirlemek için kullanılır. Genellikle fiziksel aktiviteleri sınıflandırmada yaygın olarak kullanılır.

$$SMA_i = \frac{1}{N} \sum_{j=i}^{i+N-1} (|X_j| + |Y_j| + |Z_j|) \quad (2.15)$$

X_j, Y_j, Z_j : X, Y ve Z eksenleri üzerindeki j. veri noktaları.

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

Vektör büyüklüğü

Vektör büyüklüğü, çok eksenli bir hareket verisinin genel büyüklüğünü hesaplamak için kullanılan temel bir metriktir. Bu metrik, X, Y ve Z eksenlerindeki ivme değerlerinin karelerinin toplamının karekökü alınarak hesaplanır. Zaman serisi verilerinde, her kaydırmalı pencere için ayrı ayrı hesaplanır. Vektör büyüklüğü, sinyalin büyüklüğünü bir bütün olarak temsil ettiği için özellikle fiziksel aktivitelerin şiddetini belirlemek veya farklı hareket türlerini sınıflandırmak için sıklıkla kullanılır.

$$VM_i = \sqrt{\frac{1}{N} \sum_{j=i}^{i+N-1} (X_j^2 + Y_j^2 + Z_j^2)} \quad (2.16)$$

X_j, Y_j, Z_j : X, Y ve Z eksenleri üzerindeki j. veri noktaları.

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

Enerji

Enerji, bir sinyalin karelerinin toplamı olarak tanımlanır. Bu metrik, sinyalin gücünü veya yoğunluğunu ölçmek için kullanılır.

$$MV_i = \sum_{j=i}^{i+N-1} X_j^2 + \sum_{j=i}^{i+N-1} Y_j^2 + \sum_{j=i}^{i+N-1} Z_j^2 \quad (2.17)$$

X_j, Y_j, Z_j : X, Y ve Z eksenleri üzerindeki j. veri noktaları.

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

Ortalama vektör mesafesi

Bu formül, bir pencere içerisindeki ardışık veri noktaları arasındaki ortalama vektör mesafesini hesaplamak için kullanılır.

$$d = \frac{1}{N-1} \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (2.18)$$

N : Pencere boyutu

X_i, Y_i, Z_i : X, Y ve Z eksenleri üzerindeki i . veri noktaları.

2.5.3. Entropi veya fraktal özellikler

Entropi ve fraktal özellikler, sinyallerin karmaşıklığını ve düzensizliğini ölçmek için kullanılan ileri düzey matematiksel metriklerdir. Entropi tabanlı metrikler, bir sinyalin bilgi içeriğini ve rastgeleliğini değerlendirir. Fraktal tabanlı metrikler, bir sinyalin geometrik veya kaotik yapısını modellemeye olanak tanır ve onlar hakkında fikir verir. Bu özellikler, zaman serisi verilerindeki karmaşık yapıları analiz etmek ve sinyalin altında yatan dinamikleri anlamak için kritik öneme sahiptir.

Zaman serisi analizinde entropi ve fraktal ölçümler sinyalin daha ince detaylar vererek sınıflandırma ve tahmin süreçlerinde önemli bilgiler sunar. Bu metrikler genellikle biyolojik veriler, finansal zaman serileri ve fiziksel aktivite verileri gibi karmaşık sistemlerin anlaşılmasında yaygın olarak kullanılmaktadır.

Shannon entropi

Shannon entropi, bir veri setinin rastgeleliğini ve bilgi yoğunluğunu ölçen bir metriktir [24]. Shannon entropi, bir sinyalin belirsizlik seviyesini değerlendirmek için kullanılır. Özellikle zaman serisi analizlerinde, verinin düzenli veya kaotik davranışını anlamaya yardımcı olur.

$$H_i = - \sum_{k=1}^M p_k \log_2 p_k \quad (2.19)$$

H_i : i değerindeki pencere için Shannon Entropisi

p_k : Belirli bir eksenlerdeki olasılık dağılımlarını ifade eder.

M : Sinyaldeki toplam benzersiz sınıf sayısı

Higuchi fraktal boyutu

Higuchi fraktal boyutu (Higuchi Fractal Dimension-HFD), bir zaman serisinin fraktal yapısını ölçmek için kullanılan bir yöntemdir [25]. Bu yöntem, özellikle düzensiz, periyodik olmayan ve kararsız fiziksel veriler için doğru ve kararlı fraktal boyut değerlerini

hızlı bir şekilde hesaplamayı sağlar. Higuchi yöntemi, zaman serisini farklı ölçeklerde (k) parçalarına ayırarak her bir parçanın uzunluğunu ölçer ve fraktal boyutun bir tahminini yapmak için bu uzunlukları analiz eder [26]. Bu işlem her bir eksen için ayrı ayrı uygulanmaktadır.

$$L_k^X = \frac{1}{\lfloor \frac{N-m}{k} \rfloor} \sum_{i=1}^{\lfloor \frac{N-m}{k} \rfloor} |X_{m+ik} - X_{m+(i-1)k}| \quad (2.20)$$

L_k : Her eksende ölçek (k) ile hesaplanan segment uzunluğu.

N : Zaman serisindeki toplam veri noktalarının sayısı.

m : Zaman serisinin başlangıç noktası veya kaydırmalı pencerede kullanılan kaydırma değeri.

k : Segment boyutu.

$$HFD_{axis} = \lim_{k \rightarrow 0} \frac{\log(L_k)}{\log(\frac{1}{k})} \quad (2.21)$$

L_k : Her eksende ölçek (k) ile hesaplanan segment uzunluğu.

k : Segment boyutu

Katz fraktal boyutu

Katz fraktal boyutu, bir sinyalin karmaşıklığını veya fraktal yapısını ölçmek için kullanılan bir metriktir [27]. Bu metrik, sinyalin uzunluğu ile sinyalin başlangıç noktası ve en uzak noktası arasındaki mesafeyi kullanarak hesaplanır. Katz fraktal boyutunu hesaplamak için (Eş. 2.22)'deki formül gereklidir. Bu formül ile sinyalin toplam uzunluğu bulunur. Daha sonra (Eş. 2.23)'teki formül ile sinyalin başlangıç noktası ile en uzak noktası arasındaki Öklid mesafesinin hesaplaması yapılır. (Eş. 2.22) ve (Eş. 2.23)'te verilen formüllerden elde edilen sonuçlar (Eş. 2.24)'teki formülde kullanılır ve bu sayede Katz fraktal boyutu hesaplanır.

$$L = \sum_{j=i+1}^{i+N-1} |X_j - X_{j-1}| \quad (2.22)$$

X_j : X, Y veya Z eksenleri j noktasındaki verisi

i : Pencerenin başlangıç indeksidir.

N : Pencere boyutu

$$d = \max \sqrt{(X_j - X_i)^2}, j \in [i, i + N - 1] \quad (2.23)$$

X_i : Pencerenin ilk veri noktasıdır.

$$FD_{katz}^{axis} = \frac{\log(\frac{L}{d})}{\log(\frac{d}{\Delta})} \quad (2.24)$$

L : Sinyalin toplam uzunluğu.

d : Sinyalin başlangıç noktası ile en uzak noktası arasındaki Öklid mesafe

Δ : Zaman örnekleme aralığı.

Permütasyon entropi

Zaman serisi verilerindeki düzenlilik ve karmaşıklığı ölçmek için kullanılmaktadır. Veri noktalarının sıralama düzenine (permütasyona) dayalıdır ve karmaşık zaman serilerinin analizinde oldukça etkilidir [28].

Her pencere içindeki veri noktalarının sıralama modeli, permütasyon entropisini oluşturmaktadır.

$$PE_i^{axis} = \sum_{p=1}^m P(p) \cdot \log(P(p)) \quad (2.25)$$

$P(p)$: Pencere içindeki p permütasyonuna karşılık gelen olasılık.

m : Permütasyon sıralamasını belirlemek için kullanılan gömme boyutu

Spektral entropi

Spektral entropi, bir sinyalin güç spektrumunun dağılımını ölçmek için kullanılan bir metriktir [29]. Shannon entropisine benzer şekilde, bu metrik, sinyalin frekans bileşenlerinin enerji dağılımını analiz eder. Düzensizlik ya da rastgelelik ölçüsü olarak

kullanılabilir. Daha homojen bir enerji dağılımı daha yüksek bir entropiye, belirli bir frekansta yoğunlaşma ise daha düşük bir entropiye işaret etmektedir.

$$SE_i^{axis} = -\sum_{k=1}^t P(k) \cdot \log(P(k)) \quad (2.26)$$

$P(k)$: Frekans k 'deki normalize edilmiş güç spektrumu (spektral yoğunluk).

t : Toplam frekans bileşeni sayısı.

SVD entropi

SVD (Singular Value Decomposition-SVD) entropi, bir veri kümesinin karmaşıklığını ve bilgi içeriğini ölçmek için kullanılan bir metriktir. Bu metrik, zaman serisi verilerindeki bilgi yoğunluğunu ve düzensizliğini anlamak için özellikle kullanışlıdır. Zaman serisinin faz uzayında yeniden yapılandırılmasıyla elde edilen dinamik yapıların analizi, SVD entropi ile karmaşıklık ölçümü yapılmasına olanak tanır [30]. Bu süreçte, faz uzayında elde edilen yapılar, tekil değer ayrışımı SVD entropi yöntemiyle incelenir ve bu ayrışımın entropisi hesaplanır. Her eksen X, Y ve Z için ayrı ayrı hesaplanabilir.

$$SVDE = -\sum_{i=1}^N p_i \cdot \log(p_i) \quad (2.27)$$

p_i : Tekil değerlerin normalize edilmiş olasılıkları.

N : Pencere boyutuna göre oluşturulan tekil değerlerin toplam sayısı.

Yaklaşık entropi

Yaklaşık entropi (approximate entropy), zaman serisi verilerinin düzensizliğini ve karmaşıklığını ölçmek için kullanılan bir metriktir. Büyük yaklaşık entropi değeri, zaman serisinde fazla düzensizlik olduğunu göstermektedir. Bu metrik, kaotik veya başka türlü herhangi bir zaman serisi verilerinde kullanılabilir [31]. Her eksen X, Y, Z için ayrı ayrı hesaplanabilmektedir.

$$ApEn_i^{axis} = \phi^m(r) - \phi^{m+1}(r) \quad (2.28)$$

m : Yerleştirme boyutu.

r : Tolerans eşiği.

\emptyset : Yaklaşık entropi hesaplamasında kullanılan bir ara değişkendir. Zaman serisi içerisindeki vektörlerin benzerlik oranlarını logaritmik olarak ifade etmektedir.

Örnek entropi

Örnek entropi (sample entropy), bir zaman serisinin düzenliliğini ve öngörülebilirliğini ölçmek için kullanılan bir metriktir [32]. Yaklaşık entropi'ye benzer şekilde çalışmaktadır ancak daha doğru bir ölçüm sunmaktadır çünkü örnek entropi kendi kendine eşleşmeleri hariç tutar. Bu metrik, zaman serilerindeki kaotik davranışları ve düzensizlikleri anlamak için yaygın olarak kullanılmaktadır.

$$SampEn_i^{axis}(m, r) = -\ln\left(\frac{A_i^{axis}(m+1)}{A_i^{axis}(m)}\right) \quad (2.29)$$

m : Yerleştirme boyutu.

r : Tolerans eşiği.

A_i^{axis} : Hesaplama yapılan X, Y veya Z eksenini

$A_i^{axis}(m)$: m boyutlu vektörler arasındaki toplam benzerlik sayısı.

$A_i^{axis}(m + 1)$: $m+1$ boyutlu vektörler arasındaki toplam benzerlik sayısı.

i : Pencere başlangıç indeksi.

2.5.4. Zaman frekans alanı özellikleri

Zaman frekans alanı özellikleri, bir sinyalin hem zaman hem de frekans boyutlarındaki davranışını analiz etmek için kullanılan metriklerdir. Bu özellikler, sinyalin zamanla nasıl değiştiğini ve belirli frekans bileşenlerinin enerji dağılımını ortaya koymaktadır. Zaman frekans alanı özellikleri, sinyallerin hem geçici özelliklerini hem de genel spektral (frekans bazlı) özelliklerini anlamak için kritik öneme sahiptir.

Baskın frekans

Baskın frekans (dominant frequency), bir sinyalin frekans uzayındaki en güçlü bileşenini

(en yüksek genlik değerine sahip frekans) yani temel frekans bileşenini temsil etmektedir. Bu metrik, sinyalin temel titreşim frekansını belirlemek için kullanılmaktadır.

Baskın frekansın elde edilmesi için belirli adımların gerçekleştirilmesi gerekmektedir. Öncelikle, hızlı Fourier dönüşümü (Fast Fourier transform-FFT) uygulanır. FFT'nin formülü (Eş. 2.30)'da verilmiştir. FFT'den elde edilen sonuçlar, (Eş. 2.31)'de verilen yöntem kullanılarak frekans bileşenlerinin genliklerini hesaplamak için kullanılır. Daha sonra, (Eş. 2.32) formülü yardımıyla yalnızca pozitif frekanslar analiz edilir. Son olarak, (Eş. 2.33) formülü ile sinyalin baskın frekansı belirlenir.

$$F_k = \sum_{n=0}^{N-1} x_n e^{-j2\frac{\pi}{N}kn}, \quad k = 0, 1, \dots, N-1 \quad (2.30)$$

F_k : Frekans bileşeninin genlik spektrumu

x_n : Zaman alanındaki sinyalin n n. Örneği

N : Sinyalin örnek sayısı

$$|F_k| = \sqrt{Re(F_k)^2 + Im(F_k)^2} \quad (2.31)$$

$|F_k|$: Frekans bileşeninin genliği

$Re(F_k)$: Frekans bileşeninin gerçek kısmı

$Im(F_k)$: Frekans bileşeninin sanal kısmı

$$f = \frac{k}{N.T} \quad (2.32)$$

T : Örnekleme Periyodu ($T = \frac{1}{\text{Örnekleme Oranı}}$)

$$f_{Baskin} = \arg \max (|F_k|), \quad f > 0 \quad (2.33)$$

Karekök ortalama

Karekök Ortalama (Root Mean Square-RMS), bir sinyalin genlik değerlerinin karesinin ortalamasını alıp bunun karekökünü hesaplayarak elde edilen bir metriktir. RMS, sinyalin

enerjisi veya genel büyüklüğünü anlamak için kullanılmaktadır.

$$RMS_i^{axis} = \sqrt{\frac{1}{N} \sum_{j=i}^{i+N+1} (X_j^{axis})^2} \quad (2.34)$$

RMS_i^{axis} : X, Y veya Z eksenini için RMS değeri.

N : Pencere boyutu.

X_j^{axis} : X, Y veya Z eksenindeki veri noktası.

i : Pencere başlangıç indeksi.

Sıfır geçiş oranı

Sıfır geçiş oranı (zero crossing rate), bir sinyalin sıfır eksenini kaç kez geçtiğini ölçen bir metriktir. Sinyalin osilasyon hızını analiz etmek ve özellikle titreşimli veya periyodik sinyalleri anlamak için yaygın olarak kullanılır. Zaman serisi analizinde, her kaydırmalı pencere için sıfır geçiş sayısı ayrı ayrı hesaplanabilmektedir.

(Eş. 2.35)'teki formül kullanılarak 0'dan küçük olup olmadığına bakılır. Eğer 0'dan küçük ise doğru, 0'dan büyük ise yanlış sonucu elde edilir. Bunun sonucunda da sıfır geçiş oranı elde edilmektedir.

$$ZCR_i^{axis} = \frac{1}{N} \sum_{j=i}^{i+N+1} (X_j^{axis} \cdot X_{j-1}^{axis}) \quad (2.35)$$

ZCR_i^{axis} : X, Y veya Z eksenini için sıfır geçiş oranı.

N : Pencere boyutu.

X_j^{axis} : X, Y veya Z eksenindeki veri noktası.

2.5.5. Yineleme grafiği tabanlı özellikler

Yineleme grafiği tabanlı özellikler, zaman serisi verilerinin karmaşık ve tekrar eden desenlerini analiz etmek için kullanılan güçlü bir araçtır. Bu yöntem, zaman serisindeki veri noktalarının kendi geçmişleriyle olan ilişkilerini görselleştirerek, sinyalin dinamik yapısını ve altında yatan süreçleri ortaya çıkarır.

RP, sinyalin tekrar eden davranışlarını iki boyutlu bir grafik olarak temsil ederken, yineleme nicelik belirleme analizi (Recurrence Quantification Analysis-RQA) bu grafiğin sayısal olarak analiz edilmesine olanak tanımaktadır. RQA, sinyalin kaotik doğasını, yineleme sıklığını, kararlılığını ve düzenliliğini nicel olarak ölçen metrikler sunar. Bu metrikler, sinyalin kaotik yapısını ve karmaşıklığını anlamada önemli bilgiler sağlar ve sınıflandırma gibi veri analizi süreçlerinde yaygın olarak kullanılmaktadır.

Kontrast

Kontrast, RP tabanlı özelliklerden biridir ve sinyalin lokal yoğunluk farklarını ölçmek için kullanılmaktadır. Kontrast, yineleme grafiğindeki gri-seviyeli ortak-oluşum matrisi (Gray Level Co-occurrence Matrix, GLCM) üzerinden hesaplanır ve bir sinyalin komşu veri noktaları arasındaki farkların bir ölçüsü olarak ifade edilir. Zaman serisi analizinde kontrast, sinyalin karmaşıklığını ve dinamiklerini daha iyi anlamak için kullanılır.

$$Contrast = \sum_{i,j} P(i,j) \cdot (i - j)^2 \quad (2.36)$$

$P(i, j)$: GLCM'deki i i-inci satır ve j j-inci sütundaki olasılık değeri,

i : Matrisin satır indeksi,

j : Matrisin sütun indeksi.

Yineleme Oranı

Yineleme oranı (recurrence rate), zaman serisi sinyalinin kendini ne kadar tekrar ettiğini ölçen temel bir metriktir. RP analizinin en temel özelliklerinden biridir. Sinyalin zaman içindeki yapısal düzenini ve tekrarlayan davranışlarının yoğunluğunu ifade etmektedir.

$$RR = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N R_{i,j} \quad (2.37)$$

$R_{i,j}$: RP yineleme matrisi değeridir. Eğer iki noktada yineleme varsa $R_{i,j}=1$, yoksa $R_{i,j}=0$ 'dır.

N : Pencere içindeki veri noktalarının toplam sayısıdır.

Determinizm

Determinizm, RP analizinde sinyalin zaman içindeki düzenli ve tekrarlayan davranışlarını ölçen bir metriktir. Bu metrik, RP'deki çapraz çizgilerin uzunluğunu kullanarak hesaplanmaktadır. Determinizm, sistemin öngörülebilirliği hakkında bilgi vermektedir ve genellikle düzenli ve kaotik sinyalleri ayırt etmek için kullanılmaktadır.

$$DET = \frac{\sum_{l=l_{min}}^N l.P(l)}{\sum_{l=1}^N l.P(l)} \quad (2.38)$$

l : RP'deki bir çapraz (diagonal) çizginin uzunluğu.

$P(l)$: Uzunluğu l olan diyagonal çizgilerin sayısı.

l_{min} : Hesaplama dikkate alınacak minimum diyagonal çizgi uzunluğu

N : Toplam RP boyutu veya pencere boyutuna bağlı veri noktası sayısı.

Doğrusallık

Doğrusallık (laminarity), RP analizinde sinyalin düzensiz ama uzun süreli sabit durumu temsil eden davranışlarını ölçmek için kullanılan bir metriktir. Doğrusallık, RP üzerindeki dikey çizgilerin uzunluğunu kullanarak hesaplanır. Bu metrik, sistemdeki durağan veya düzensiz ama uzun süreli sabit fazların varlığını ve süresini analiz etmek için kullanılır.

$$LAM = \frac{\sum_{v=v_{min}}^N v.P(v)}{\sum_{v=1}^N v.P(v)} \quad (2.39)$$

v : RP'deki bir dikey çizginin uzunluğu.

$P(v)$: Uzunluğu v olan dikey çizgilerin sayısı.

v_{min} : Hesaplama dikkate alınacak minimum dikey çizgi uzunluğu

N : Toplam RP boyutu veya pencere boyutuna bağlı veri noktası sayısı.

En uzun çapraz çizgi

En uzun çapraz çizgi, RP analizinde sinyalin dinamik davranışlarını anlamak için kullanılan bir metriktir. En uzun çapraz çizgi özellikle sistemin düzenli veya kaotik özelliklerini

değerlendirmek için kullanılmaktadır.

$$L_{max} = \max(L) \quad (2.40)$$

L : RP üzerindeki çapraz çizgilerin uzunlukları.

L_{max} : RP'deki en uzun çapraz çizginin uzunluğu

Yineleme oranı tabanlı entropi

Yineleme oranı tabanlı entropi, bir sinyalin karmaşıklığını ve düzensizliğini değerlendirmek için kullanılan bir metriktir. RP üzerindeki desenlere dayalı olarak hesaplanmaktadır. RP'deki yinelenen yapıların çeşitliliğine dayanarak, sistemin öngörülebilirlik seviyesini belirlemeye yardımcı olur.

$$HR_{RR} = - \sum_l P(l) \cdot \log(P(l)) \quad (2.41)$$

HR_{RR} : RP Tabanlı Entropi

$P(l)$: Çapraz çizgi uzunluğu l için olasılık dağılımı.

l : RP üzerindeki çapraz çizgilerin uzunluğu.

Yakalama süresi

Yakalama süresi (Tapping Time-TT), bir sistemin RP üzerinde laminer bölgelerde ne kadar süre geçirdiğini ölçen bir metriktir. TT, bir sinyalin dinamik davranışını anlamak için kullanılır ve sistemin kararlılık seviyesini değerlendirmede önemli bir araçtır. Bu metrik, sistemin zamansal sürekliliğini analiz ederek ani değişimlerin ve geçişlerin belirlenmesine yardımcı olur.

$$TT = \frac{\sum_V v \cdot P(v)}{\sum_V P(v)} \quad , V \geq V_{min} \text{ olmalı} \quad (2.42)$$

$P(v)$: Uzunluğu V olan laminer çizgilerin olasılığı.

V_{min} : Minimum çizgi uzunluğu

V : Laminer çizgi uzunluğu.

Çapraz yineleme oranı

Çapraz yineleme oranı (Diagonal Recurrence Rate-DRR), RP’de oluşan çapraz çizgilerin toplam uzunluğunun, toplam yineleme noktalarına oranını ifade eden bir metriktir. Bu metrik, sinyalin zaman içindeki dinamiklerinin ne kadar düzenli olduğunu analiz etmek için kullanılmaktadır. Özellikle zaman serisi analizlerinde, sinyaldeki düzenlilik ve periyodik yapıların karakterize edilmesinde kullanılmaktadır.

$$DRR = \frac{\sum_V v.P(v)}{N_{total}} \quad , \quad V \geq V_{min} \text{ olmalı} \quad (2.43)$$

$P(v)$: Uzunluğu V olan çapraz çizgilerin sayısı.

V_{min} : Minimum çapraz çizgi uzunluğu.

N_{total} : Toplam yineleme noktalarının sayısı.

V : Çapraz çizgi uzunluğu.

Dikey yineleme oranı

Dikey yineleme oranı (Vertical Recurrence Rate-VRR), RP’deki dikey çizgilerin yoğunluğunu ölçen bir metriktir. Bu metrik, sinyalin zaman içinde durağanlık veya sabit kalma davranışlarını analiz etmek için kullanılmaktadır.

$$VRR = \frac{\sum_V v.P(v)}{N.(N-1)} \quad (2.44)$$

$P(v)$: Uzunluğu V olan dikey çizgilerin sayısı.

V : Uzunluğu V olan dikey çizgiler.

N : Yineleme grafiğinin boyutu.

Ortalama çapraz çizgi uzunluğu

Ortalama çapraz çizgi uzunluğu, RP üzerinde oluşan çapraz çizgilerin ortalama uzunluğunu ifade eder. Bu metrik, sinyalin zamansal kararlılığını ve düzenliliğini ölçmek için kullanılmaktadır. Uzun çapraz çizgiler, sinyaldeki düzenli ve tekrar eden süreçlerin bir göstergesidir. Daha kısa çizgiler ise daha düzensiz veya kaotik bir yapı hakkında bize bilgi

vermektedir.

$$L_{mean} = \frac{\sum_{v=v_{min}}^{V_{max}} v.P(v)}{\sum_{v=v_{min}}^{V_{max}} P(v)} \quad (2.45)$$

$P(v)$: Uzunluğu V olan çapraz çizgilerin sayısı.

V : Çapraz çizgilerin uzunluğu.

v_{min} : En kısa çapraz çizgi uzunluğu.

v_{max} : En uzun çapraz çizgi uzunluğu.

İlk yineleme zamanı

İlk yineleme zamanı (First Recurrence Rate-FRT), bir yineleme grafiğinde sinyalin ilk kez kendine benzer bir duruma dönmesi için geçen zaman adımını ifade etmektedir. FRT, sinyalin dinamik yapısını anlamak ve zamansal bağımlılıkları incelemek için kullanılmaktadır.

$$FRT = \frac{1}{M} \sum_{m=1}^M \tau_m \quad (2.46)$$

τ_m : m değerine göre veri noktası için ilk yineleme zamanı (ilk benzerlik).

M : Veri setindeki toplam gözlem sayısı.

Eğilim

Eğilim (trend), bir zaman serisinin genel yönelimini veya uzun vadeli değişimini tanımlayan bir ölçümdür. Özellikle RP tabanlı analizlerde sinyalin zaman içinde artış, azalış veya sabit kalma eğilimini belirlemek için kullanılır. Bu analiz geçici dalgalanmaları göz ardı ederek temel hareket modelini ortaya çıkarmaya yardımcı olur. Ayrıca, sistemdeki yapısal değişiklikleri tespit etmek ve geleceğe yönelik tahminlerde bulunmak için de kritik bir rol oynar.

$$Trend = a_{axis} \quad (2.47)$$

$$y = ax + b \quad (2.48)$$

x : Zaman eksenini,

y : Sinyal deęerleri,

a : Doğrusal regresyon katsayısı,

b : Doğrunun y eksenini kestięi nokta

2.6. Sınıf Dengesi

Bu çalışmada, veri setindeki sınıflar arasında belirgin bir dengesizlik olduęu tespit edilmiştir. Örneęin, ayakta dinlenme (RES) davranışı 150,130 örnekle veri setindeki en büyük sınıfı oluştururken, saldırma (ATT) gibi davranışlar yalnızca 366 örnek içermektedir. Benzer şekilde, yalama (LCK), idrar yapma (URI) ve çiftleşme (BMN) gibi davranışların her biri toplamda 2,000'den az örneęe sahiptir. Bazı davranışların tespit edilmesi için gerekli veri sayısı çok az kalmış ve bu da model performansını olumsuz yönde etkilemektedir. Modeller, azınlık sınıfları doğru bir şekilde öğrenmekte güçlük çekmekte ve bu sınıfları genellikle yanlış sınıflandırmaktadır.

Bu dengesizlięin sınıflandırma performansı üzerindeki olumsuz etkisini azaltmak amacıyla, literatürde [33] bu veri setinde bulunan, LCK, yatma pozisyonunda dinlenme (Resting Lying Position-REL), URI, ATT, kaçma (Escape-ESC) ve BMN sınıfları 'Dięerleri' başlığı altında birleştirilmiştir. Bu işlem sonucunda davranış sayısı sekize indirgenmiştir. Bu birleşim, veri setindeki azınlık sınıfların temsiliyetini artırarak modelin genelleme yeteneęini ve sınıflandırma doğruluęunu da artırmıştır.

Bu çalışmada [33]'de yapılan çalışmadan farklı olarak birleştirilmiş olan veri setindeki azınlık sınıflarının çıkarılmasına karar verilmiştir. Azınlık sınıflarının çıkarılması, bu sınıfların yeterince temsil edilmemesi nedeniyle modelin genelleme yeteneęini artırmayı hedeflemiştir. Alternatif olarak veri artırma yöntemleri (örneęin Sentetik Azınlık Aşırı Örnekleme Teknięi (Synthetic Minority Over-Sampling Technique-SMOTE) veya veri çoęaltma) tercih edilebilirdi. Ancak bu yöntem mevcut veri setinde bulunan su içme DRN ve Yemlikte yem yeme (Feeding in Stanchion-FES) davranışı üzerinde uygulanmış ve olumlu sonuçlar vermiştir. Bununla birlikte benzer bir yaklaşım dięer azınlık sınıflarında kullanıldığında, veri setinin yapısının tamamen bozulabileceęinden çekinilmiş ve bu

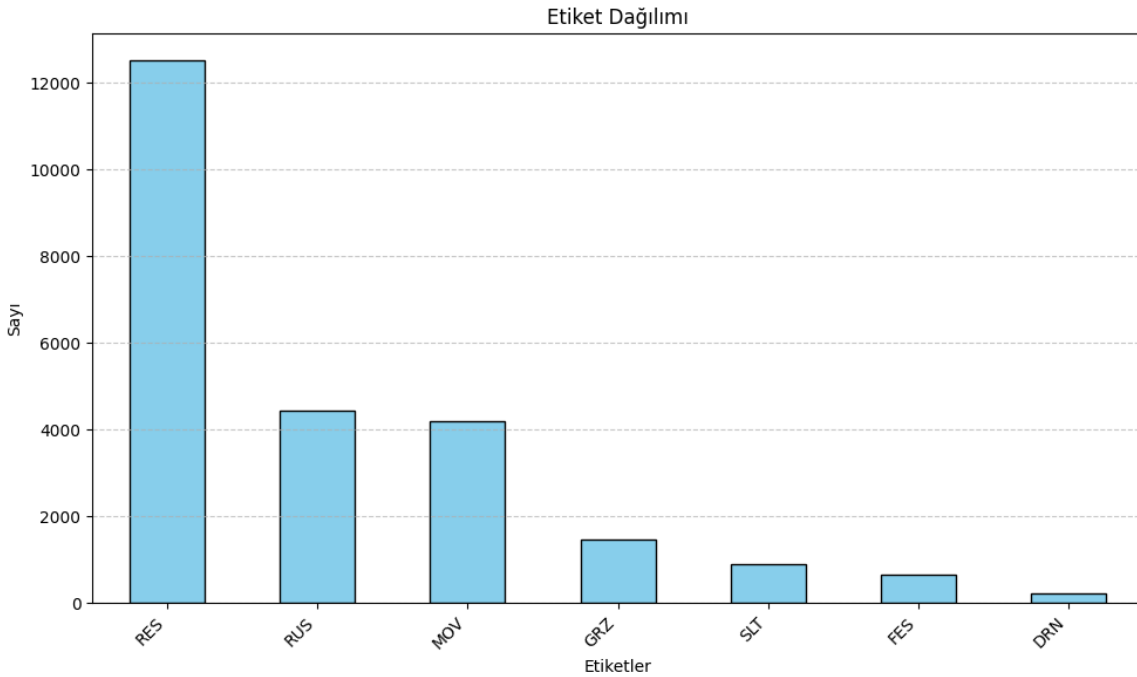
durumdan kaçınılmıştır.

2.6.1. Azınlık sınıfların çıkarılması

Bu çalışmada, her biri farklı bir ineği temsil eden 6 ayrı csv dosyası önce tek bir dosyada birleştirilmiştir. Bu birleştirme işlemi, tüm verilerin ortak bir yapıda analiz edilmesine olanak sağlamış ve modelleme sürecini kolaylaştırmıştır. Daha sonra 2 adet yöntem izlenmiştir.

İlk yöntem olarak, belirli azınlık sınıflar veri setinden tamamen çıkarılmıştır. Bu yaklaşım, özellikle bu sınıfların veri setindeki sayısının düşük olması nedeniyle modellerin genelleme performansını olumsuz etkileme riskini azaltmak amacıyla benimsenmiştir. Ancak bu işlem, özellik çıkarma sürecinden sonra gerçekleştirilmiştir. Böylece, özellik çıkarması aşamasında veriler bozulmadan korunmuştur.

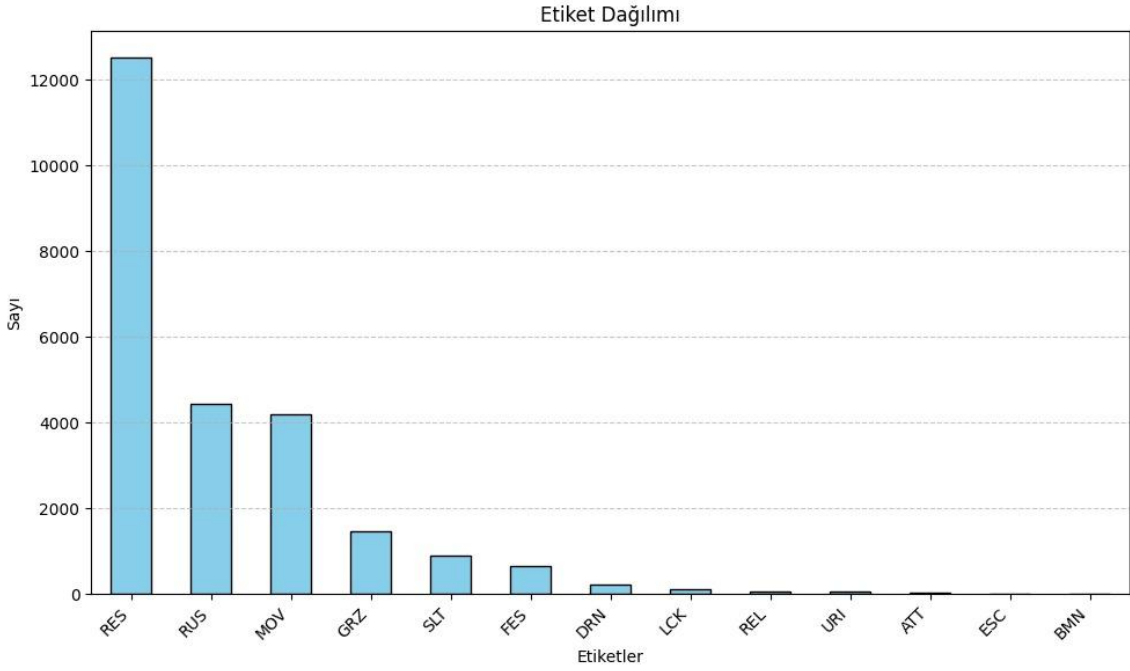
ATT, BMN, ESC, URI, REL ve LCK azınlık sınıfları çıkarılmıştır. Bu işlemden sonra veri setindeki değişim Şekil 2.4.'de gösterilmiştir.



Şekil 2.4. Veri setinden azınlık sınıfları çıkarılmasından sonra etiket dağılımları

İkinci yöntem olarak, azınlık sınıfları veri setinden çıkarmadan da bir analiz

gerçekleştirilmiştir. Bu yaklaşım veri setindeki tüm sınıfların modelleme sürecine dahil edilmesi ve her bir sınıfın temsil ettiği davranışın analize katkıda bulunmasını sağlamak amacıyla benimsenmiştir. Tüm azınlık sınıfları içeren veri seti modellerin farklı davranış kategorilerini ayırt etme performansını değerlendirmek için kullanılmıştır. Bu işlem modelin genelleme yeteneği ve azınlık sınıfları tanımlama kapasitesini ölçmek açısından önemli bir adım olmuştur.



Şekil 2.5. Veri setindeki ham etiket dağılımları

Bu yaklaşımla elde edilen sonuçlar azınlık sınıfların veri setinde tutulmasının modelin doğruluğu, karmaşıklığı ve performansı üzerindeki etkilerini göstermektedir. Veri setinde bulunan sınıfların dağılımı Şekil 2.5.'te gösterilmiştir.

2.7. Sentetik Azınlık Aşırı Örnekleme Tekniği

Makine öğrenimi modellerinin etkin ve doğru bir şekilde eğitilmesi dengeli bir veri seti gerektirir. Çoğu gerçek dünya veri setinde olduğu gibi bu çalışmada kullanılan veri setinde de sınıflar arasında ciddi bir dengesizlik tespit edilmiştir. Örneğin bazı davranış kategorileri oldukça fazla sayıda örneğe sahipken, bazıları çok az temsil edilmiştir. Bu dengesizlik, sınıflandırma modellerinin azınlık sınıfları öğrenmesini zorlaştırmakta ve bu sınıfları genellikle yanlış sınıflandırmasına neden olmaktadır.

Bu problemi çözmek ve veri setindeki azınlık sınıfının temsiliyetini artırmak için SMOTE yöntemi uygulanmıştır. SMOTE, azınlık sınıflarındaki veri örneklerinin yakın komşularını dikkate alarak ara noktalar üretir ve böylece sentetik veri örnekleri oluşturur. Bu yöntem azınlık sınıflarının yalnızca kopyalanması yerine, mevcut verilerden türetilmiş yeni veriler oluşturularak daha dengeli bir veri seti elde edilmesini sağlar.

Bu çalışmada az sayıda örneğe sahip olan DRN ve FES üzerinde SMOTE yöntemi kullanılmıştır.

Azınlık sınıfındaki veri örneklerinin özellik uzayındaki yakın komşuları belirlenmiştir. Bu komşular arasındaki doğrular üzerinde ara noktalar oluşturularak sentetik veri örnekleri üretilmiştir. Yeni oluşturulan sentetik veri örnekleri, azınlık sınıfının temsiliyetini artırmış ve veri setini daha dengeli hale getirmiştir.

SMOTE aşırı öğrenme riskini azaltmaktadır çünkü veri setine aynı örneklerin kopyalarını eklemek yerine yeni örnekler üretmektedir. SMOTE, dengeli bir veri seti sağlaması ile modelin genelleme kapasitesini artırmaktadır.

2.8. Sınıflama

Bu çalışmada ivmeölçer verilerinden hayvan davranışlarını sınıflandırmak amacıyla hem geleneksel makine öğrenimi algoritmaları hem de derin öğrenme yöntemlerinden biri olan LSTM kullanılmıştır. Sınıflandırma aşaması veri analiz sürecinin en kritik adımlarından biridir ve kullanılan yöntemlerin performansı, elde edilen sonuçların doğruluğunu ve genelleme kapasitesini doğrudan etkiler. Bu kapsamda geleneksel makine öğrenimi sınıflayıcılarının yanı sıra, ardışık verilerin özelliklerini daha iyi öğrenebilme kapasitesine sahip olan LSTM ağı da değerlendirilmiştir.

Bu çalışmada sınıflandırma modellerine daha anlamlı ve etkili özellikler sunmak amacıyla bir özellik seçimi aşaması uygulanmıştır. Özellik seçimi, modelin gereksiz veya az etkili özelliklerden arındırılması ve yalnızca sınıflandırmaya katkı sağlayan özelliklerin belirlenmesi hedeflenmiştir. Bu amaçla bir önem eşik değeri belirlenmesi gerekmektedir. Bu eşik değerinin altında kalan özellikler veri setinden çıkarılmakta, böylece daha optimize edilmiş ve daha hızlı çalışan sınıflandırma modelleri elde edilmesi amaçlanmaktadır.

Özellik seçimi veri setinin genelleme performansını artırmak, hesaplama maliyetlerini düşürmek ve aşırı öğrenme riskini en aza indirmek için etkili bir yöntem olarak uygulanmıştır.

Bu çalışmada kullanılan sınıflandırma algoritmaları, farklı veri yapıları ve problemler için optimize edilmiş çeşitli yaklaşımları içermektedir.

2.8.1. Ağaç tabanlı sınıflayıcılar

Karar Ağacı (Decision Tree-DT) sınıflayıcı, sınıflayıcı problemleri için kullanılan bir karar ağacı modelidir. Bu model eğitim veri setinden bir ağaç yapısı oluşturarak sınıflandırma işlemi yapar. DT kök düğümden başlayarak veri kümesini özelliklerine göre dallara ayırır ve yaprak düğümlerde sınıfları temsil eder. Her dal belirli bir özelliğe dayalı olarak veri kümesini alt gruplara böler ve bu işlem veriler tamamen sınıflanana kadar devam eder.

RF sınıflayıcısı, birden fazla karar ağacının bir araya getirilmesiyle oluşturulan güçlü bir topluluk öğrenme yöntemidir [34]. Bu algoritma sınıflandırma problemlerinde yüksek doğruluk ve genelleme kapasitesi sunarak makine öğreniminde sıklıkla tercih edilmektedir. RF, her bir karar ağacının bağımsız olarak eğitildiği ve tahminlerin çoğunluk oylaması ya da regresyon problemlerinde ortalama alınarak birleştirildiği bir yapıya sahiptir [34]. Bu yöntem modelin hem aşırı öğrenme riskini azaltmak hem de veri setindeki çeşitliliği artırmak için rastgele örnekleme ve özellik seçimi mekanizmalarını kullanır. RF'nin çalışma prensibi üç ana adımdan oluşur. İlk olarak eğitim veri setinden rastgele örnekler seçilir ve her bir karar ağacı bu alt veri kümesi üzerinde eğitilir. Bu süreç, önyükleme toplama (bootstrap aggregating) yöntemiyle gerçekleştirilir ve ağaçlar arasında çeşitlilik sağlar. İkinci olarak her bir düğüm bölünmesinde tüm özellikler yerine rastgele bir alt küme kullanılır. Bu özellik seçimi yöntemi modelin genelleme performansını artırır. Son olarak tüm ağaçların tahminleri birleştirilerek nihai tahmin oluşturulur. Bu süreçte sınıflandırma problemlerinde çoğunluk oylaması, regresyon problemlerinde ise ortalama alınır. RF sınıflayıcısı genelleme kapasitesi, doğruluk ve esneklik açısından birçok avantaj sunmaktadır. Özellikle yüksek boyutlu veri setlerinde, rastgele örnekleme ve rastgele özellik seçimi mekanizmaları sayesinde daha dengeli ve doğru tahminler yapabilir.

ET, karar ağaçlarına dayalı bir topluluk öğrenme yöntemidir. Bu yöntem, veri setinden

rastgele alt örnekler olarak ve her düğümde bölünmeleri rastgele seçerek birden fazla karar ağacı oluşturur [35]. ET, RF algoritmasına benzer, ancak temel farkı, düğüm bölünmelerinin tamamen rastgele seçilmesidir. Bu algoritma, yüksek boyutlu ve gürültülü veri setlerinde etkili bir şekilde çalışır. Her bir ağacın bağımsız olarak eğitilmesi, modelin hesaplama sürecini paralel bir şekilde gerçekleştirebilmesini sağlar. Bununla birlikte, ET, RF ile karşılaştırıldığında daha hızlı bir eğitim sürecine sahiptir çünkü bölünme noktalarını seçmek için ek bir optimizasyon adımı gerekmez. Bu yöntem, özellikle büyük veri setlerinde doğruluk, genelleme kapasitesi ve hesaplama maliyeti arasında iyi bir denge sağlar.

2.8.2. Doğrusal yöntemli sınıflayıcılar

Linear Ridge, doğrusal regresyonu L2 düzenleme tekniğiyle birleştirerek, modelin ağırlıklarını sınırlandırır ve bu sayede aşırı öğrenme riskini azaltır. Özellikle çoklu doğrusal bağımlılıkların olduğu veri setlerinde etkili sonuçlar sunar.

LDA, sınıflar arasındaki ayrımı modellemek için kullanılan bir yöntemdir. Sınıflar arasındaki sınırları doğrusal olarak ayırmaya çalışır ve bu doğrusal sınırlar, sınıfların kovaryans matrislerinin eşit olduğu varsayımıyla belirlenir. Bu yöntem, özellikle sınıflar arasında doğrusal ayrım yapılabilen veri setlerinde etkili bir araçtır.

Lojistik Regresyon (Logistic Regression-LR), doğrusal regresyonun sınıflandırma problemlerine uyarlanmış bir versiyonudur ve özellikle iki sınıflı (binary) sınıflandırma problemlerinde yaygın olarak kullanılır [36]. Bu yöntem bir olayın olasılığını tahmin etmek için sigmoid fonksiyonu kullanır. Sigmoid fonksiyonu, giriş değerlerini 0 ile 1 arasında normalize ederek, sınıfların olasılık dağılımını belirler. LR bir karar sınırı oluşturmak için özelliklerin ağırlıklı toplamını kullanır ve bu sınır doğrusal bir hiper düzlemle ifade edilir [36].

Lasso LR, LR'nin düzenlenmiş bir versiyonudur ve özellikle özellik seçimi yapılması gereken durumlarda tercih edilir [37]. Bu yöntemde, L1 düzenleme terimi kullanılır [37]. L1 düzenleme, gereksiz veya düşük öneme sahip özelliklerin ağırlıklarını sıfıra indirerek modelin sadeliğini artırır ve aşırı öğrenme riskini azaltır. Bu özellik yüksek boyutlu ve gürültülü veri setlerinde özellikle faydalıdır. Lasso LR, modele sadece önemli özellikleri

dahil ederek hem hesaplama yükünü azaltır hem de modelin genelleme performansını artırır. L1 düzenleme, bazı özellikleri tamamen dışlayarak modelin doğruluğunu etkileyebilir.

Esnek Ağ Sınıflayıcısı (Elastic Net-ENC), Lasso ve Ridge regresyon yöntemlerinin bir kombinasyonudur ve özellikle çoklu doğrusal bağıntının (multicollinearity) olduğu ve yüksek boyutlu veri setlerinde etkili bir sınıflandırma yöntemidir. Bu algoritma hem L1 (Lasso) hem de L2 (Ridge) düzenleme terimlerini birleştirerek, bu iki yöntemin avantajlarını bir araya getirir [38]. L1 düzenleme, gereksiz özelliklerin ağırlıklarını sıfıra indirerek özellik seçimi yapmaktadır. L2 düzenleme ise önemli özelliklerin ağırlıklarını küçülterek modelin genelleme kapasitesini artırır. Esnek Ağ, veri setindeki özelliklerin sayısının gözlem sayısından fazla olduğu durumlarda veya özelliklerin birbirine yüksek derecede bağımlı olduğu (kolineer) veri setlerinde daha iyi performans gösterir. ENC, yalnızca Lasso'nun sıfıra eşit ağırlık problemiyle başa çıkmakla kalmaz. Aynı zamanda Ridge'in aşırı bağımlılığı minimize eden özelliğinden faydalanır. Bu sayede hem değişken seçimi hem de modelin dengeleme kapasitesi sağlanmış olur [38].

Doğrusal SVC, verileri doğrusal bir sınır ile ayırmayı hedefleyen bir sınıflayıcıdır. Bu yaklaşım ile yüksek boyutlu ve büyük veri setlerinde hızlı bir şekilde sonuç vermektedir ve özellikle sınıflar arasında doğrusal bir ayırım olduğu durumlarda etkili sonuçlar sunmaktadır. Ayrıca, diğer SVC modellerinden farklı olarak hesaplama maliyeti daha düşük olup daha az kaynak tüketir. Ancak doğrusal olmayan sınırların gerekli olduğu durumlarda performansı sınırlı olabilir.

Pasif Agresif Sınıflayıcı (Passive Aggressive Classifier), çevrimiçi öğrenme yöntemlerinden biridir ve özellikle büyük veri setlerinde hızlı sınıflandırma yapmak için tasarlanmıştır [39]. Bu algoritma, modelin yalnızca yanlış sınıflandırdığı veya marjin kriterini karşılamayan örnekler üzerinde güncellendiği pasif ve agresif bir strateji benimser [39]. Pasif durumda, eğer bir veri noktası doğru sınıflandırılmışsa, model parametreleri değiştirilmez. Ancak agresif durumda, yanlış sınıflandırılmış bir veri noktası ile karşılaşıldığında, model parametreleri bu hatayı telafi edecek şekilde güncellenir. Bu algoritma, maksimum marjin sınıflandırması ile veri uyumunu optimize etmeye çalışır. Pasif Agresif Sınıflayıcı, özellikle doğrusal modellerin yeterli olduğu ve büyük veri setlerinde hızlı işlem gerektiren durumlarda avantajlıdır. Ancak, doğrusal olmayan veri

yapılarında performansı sınırlı kalabilir.

Perceptron Sınıflayıcısı, doğrusal olarak ayrılabilen veri setlerini sınıflandırmak için kullanılan temel bir makine öğrenimi algoritmasıdır [40]. Algoritma, giriş verileri ile ağırlık vektörleri arasındaki iç çarpımı hesaplar ve bu sonucu bir aktivasyon fonksiyonu kullanarak sınıflandırma yapar. Perceptron'un temel amacı iki sınıf arasında maksimum marjın sağlayan bir hiper düzlem oluşturmaktır. Eğitim sürecinde, yanlış sınıflandırılmış veri noktaları tespit edilir ve ağırlık vektörleri bu hataları telafi edecek şekilde güncellenir [40]. Güncelleme öğrenme oranı n ile yanlış sınıflandırılmış veri noktalarının çarpılması esasına dayanır. Basitliği ve hızlı çalışması nedeniyle küçük ve doğrusal olarak ayrılabilir veri setleri için etkili bir yöntemdir. Doğrusal olmayan veri yapılarında performansı sınırlı olup, bu tür durumlar için daha karmaşık algoritmalar gereklidir. Perceptron Sınıflayıcı, modern makine öğrenimi algoritmalarının temelini oluşturan önemli bir adımdır.

2.8.3. Topluluk yöntemleri sınıflayıcılar

Bagging sınıflayıcısı, birden fazla zayıf sınıflayıcıyı birleştirerek daha güçlü bir topluluk modeli oluşturan ve makine öğreniminde yaygın olarak kullanılan bir yöntemdir. Bu yaklaşım eğitim veri setinden rastgele örneklemeler alarak her bir alt kümede bağımsız sınıflayıcılar eğitir ve tahminlerini çoğunluk oylaması veya regresyon ile birleştirir [41]. Bagging yöntemi, özellikle aşırı öğrenme riskini azaltmak ve genelleme kapasitesini artırmak amacıyla kullanılır. Yöntem, yüksek varyansa sahip modellerde performansı artırırken veri setindeki gürültüye karşı da dayanıklıdır. Bagging sınıflayıcısı, paralel çalışabilirliği sayesinde hesaplama maliyetlerini kontrol edebilir; ancak birçok modelin eğitilmesi, yine de ekstra hesaplama gücü gerektirir.

AdaBoost (Adaptive Boosting) algoritması, birden fazla zayıf sınıflayıcıyı bir araya getirerek daha güçlü bir topluluk modeli oluşturan ve sınıflandırma problemlerinde yaygın olarak kullanılan bir yöntemdir. AdaBoost, her iterasyonda yanlış sınıflandırılan veri noktalarına daha fazla ağırlık vererek modelin genelleme performansını artırmayı amaçlar [42]. Bu algoritma genellikle DT sınıflayıcısı gibi zayıf sınıflayıcılar ile kullanılır ve bu sınıflayıcıların doğruluğunu artırır. AdaBoost'un çalışma prensibi, başlangıçta eğitim verilerindeki her örneğe eşit ağırlık verilmesiyle başlar. İlk sınıflayıcı eğitildikten sonra yanlış sınıflandırılan örneklere daha fazla ağırlık atanır ve bu süreç iteratif olarak

tekrarlanır. Nihai model, her bir sınıflayıcının hata oranına göre ağırlıklandırılmış tahminlerini birleştirir. AdaBoost, hesaplama süreci yüksek iterasyon sayısı nedeniyle zaman alıcı olabilir.

Gradyan Artırma (Gradient Boosting-GB) sınıflayıcı, zayıf sınıflayıcıların ardışık olarak eğitilmesiyle güçlü bir sınıflandırma modeli oluşturan bir topluluk öğrenme yöntemidir [43]. Bu algoritma her bir zayıf sınıflayıcının, önceki sınıflayıcının yaptığı hataları düzeltmesi prensibine dayanır ve bu süreç kayıp fonksiyonunun gradyanı kullanılarak optimize edilir [43]. GB, genellikle karar ağaçlarını zayıf sınıflayıcı olarak kullanır ve her iterasyonda bu ağaçlar, modelin genelleme performansını artırmak amacıyla hataları minimize edecek şekilde eğitilir. Her iterasyonda elde edilen sınıflayıcı, toplam modele ağırlıklı olarak eklenir ve bu sayede modelin doğruluğu artar. GB, özellikle büyük ve karmaşık veri setlerinde yüksek doğruluk sağlamasıyla öne çıkar. Ayrıca, diferansiyellenebilir herhangi bir kayıp fonksiyonunu optimize edebilmesi, bu algoritmayı esnek ve farklı problemler için uyarlanabilir hale getirir. Hesaplama maliyetinin yüksek olması ve parametrelerin dikkatlice ayarlanması gerekmesi gibi dezavantajları bulunmaktadır.

HGBC (Hist Gradient Boosting Sınıflayıcı), büyük veri setlerinde hızlı ve etkili sınıflandırma yapmak için optimize edilmiş bir GB algoritmasıdır. HGBC, geleneksel GB algoritmalarından farklı olarak verileri belirli sayıda bölmelere ayırarak histogramlar üzerinden işlemler gerçekleştirir. Bu yaklaşım hesaplama maliyetini azaltırken aynı zamanda bellek kullanımını optimize eder. Her iterasyonda önceki modelin hataları histogramlarla temsil edilen verilere göre düzeltilir ve bu süreç, ardışık olarak ağaçlar oluşturularak devam eder. HGBC büyük veri setlerinde yüksek performans elde edilirken, düşük bellek tüketimiyle dikkat çeker.

Voting Hard Sınıflayıcısı, birden fazla sınıflayıcıyı birleştirerek daha güçlü bir topluluk öğrenme yöntemi sunan bir sınıflayıcıdır. Bu yöntem farklı modellerin sınıf tahminlerini değerlendirir ve çoğunluk oylaması prensibi ile nihai sınıfı belirler. Her bir sınıflayıcının eşit ağırlık taşıdığı bu yaklaşım, bireysel sınıflayıcıların hata oranlarının birbirinden bağımsız olduğu durumlarda daha doğru sonuçlar verir [44]. Özellikle doğrusal ve doğrusal olmayan sınıflayıcıların bir arada kullanıldığı durumlarda etkili performans sunar. Ancak, tüm sınıflayıcıların eşit ağırlığa sahip olması, zayıf modellerin nihai kararı olumsuz

etkileyebileceği anlamına gelir.

Voting Soft Sınıflayıcısı, sınıflandırma işlemini gerçekleştirirken sınıflayıcıların olasılık tahminlerini dikkate alan bir topluluk öğrenme yöntemidir. Bu algoritma, her bir sınıflayıcının tahmin ettiği sınıf olasılıklarını toplar ve en yüksek toplam olasılığa sahip sınıfı seçer. Sınıflayıcıların ağırlıkları, belirli bir modelin güvenilirliğini artırmak için özelleştirilebilir [44]. Bu yöntem, sınıflayıcıların farklı özelliklerini dikkate alarak daha dengeli bir performans sunar. Ancak, olasılık tahminleri gerektirdiğinden, tüm sınıflayıcılar için bu bilgilerin erişilebilir olması gerekir. Bu iki sınıflayıcı, topluluk öğrenme yöntemleri arasında esneklik ve performans açısından geniş bir uygulama alanına sahiptir. Voting Soft Sınıflayıcısı genellikle daha iyi genelleme kapasitesi sunarken, Voting Hard Sınıflayıcısı daha basit ve hızlı bir çözüm sağlar.

2.8.4. Çekirdek tabanlı sınıflayıcılar

Destek Vektör Sınıflayıcısı (Support Vector Classifier-SVC), veriler arasındaki sınırları belirlemek ve sınıfları birbirinden ayırmak için maksimum marjine sahip bir hiper düzlem oluşturmayı hedefleyen bir sınıflayıcıdır [45]. Doğrusal olmayan sınırlar için kullanılır ve farklı çekirdek fonksiyonlarıyla (kernels) esnek bir şekilde çalışabilir. Bu yaklaşım, karmaşık veri yapıları için güçlü bir model sunar ve genelleme kapasitesi yüksektir.

Kuadratik Kernel SVC, ikinci dereceden bir polinomsal çekirdek kullanarak sınıflandırma yapar. Bu yöntem doğrusal olmayan ve daha karmaşık ilişkilerin bulunduğu veri setlerinde etkili sonuçlar verir. Veriler arasındaki sınırları, ikinci dereceden bir polinomla modellediği için, sınıf ayrımları belirgin hale gelir. Kuadratik Kernel SVC, özellikle sınıflar arasındaki doğrusal olmayan ilişkilerde daha iyi bir performans sergiler.

RBF (Radial Basis Function) Kernel SVC, doğrusal olmayan veriler için en yaygın kullanılan çekirdeklerden biridir [45]. Bu yöntemde veri noktalarının çevresindeki etkiler bir Gauss fonksiyonu ile modellenir. Bu yöntem ile sınıf sınırları oldukça esnek bir şekilde çizilebilir ve karmaşık veri yapılarına uyum sağlanır. RBF Kernel SVC, genellikle varsayılan parametrelerle etkili çalışır ancak C ve γ gibi hiper parametrelerin dikkatlice optimize edilmesi gerekir.

2.8.5. Bayes yöntemli sınıflayıcılar

Gaussian Naive Bayes (Gaussian NB), NB sınıflayıcının sürekli özelliklere sahip veri setleri için özelleştirilmiş bir varyasyonudur. Temel olarak NB algoritması sınıflar arasındaki ilişkiyi Bayes Teoremi'ne dayanarak olasılık hesaplamaları ile kurar ve sınıflar arasındaki ayrımı basit bir olasılık modeliyle gerçekleştirir. Gaussian NB, sürekli özelliklere sahip veri noktalarını modellemek için Gauss (Normal) Dağılımı varsayımını kullanır. Gaussian NB'nin temel prensibi, her bir özellik için sınıf koşullu olasılıkları, Gauss Dağılımı parametreleri (ortalama ve standart sapma) ile hesaplamaktır [46].

$$P(x_i|C_k) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}} \quad (2.49)$$

$P(x_i|C_k)$: x_i özelliğinin C_k sınıfına ait olma koşullu olasılığıdır.

μ : C_k sınıfına ait x_i özelliğinin ortalamasıdır.

σ : C_k sınıfına ait x_i özelliğinin standart sapmasıdır.

x_i : Verinin i 'deki değeri.

C_k : Verinin k 'deki sınıfı.

Gaussian NB genellikle düşük hesaplama maliyeti ve hızlı çalışması nedeniyle tercih edilir. Ancak Gaussian Dağılımı varsayımı, özelliklerin gerçekte Gauss Dağılımına uymadığı durumlarda model performansını olumsuz etkileyebilir. Gaussian NB özellikle sürekli ve sayısal özelliklerin baskın olduğu veri setlerinde başarılıdır.

Bernoulli Naive Bayes (Bernoulli NB), ikili (binary) veya sıklıkla sıfır ve bir değerlerine sahip veriler için özelleştirilmiş bir NB varyasyonudur. Bu algoritma özelliklerin varlığı veya yokluğu üzerinden çalışır ve her bir özellik için ilgili sınıfta özelliğin ortaya çıkma olasılığını hesaplar. Bernoulli NB'nin temel farkı veri noktasındaki özelliklerin belirli bir sınıf için var olup olmadığını değerlendirerek bir olasılık modeli oluşturmasıdır [47].

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k)^{x_i} (1 - P(x_i|C_k))^{(1-x_i)} \quad (2.50)$$

$P(X|C_k)$: X veri noktasının C_k sınıfına ait olma koşullu olasılığıdır.

x_i : X veri noktasının i 'deki değeridir. Sadece 0 (yok) veya 1 (var) değerlerini alır.

$P(x_i|C_k)$: C_k sınıfında x_i özelliğinin var olma olasılığıdır.

n : Veri noktalarındaki toplam özellik sayısıdır.

Bernoulli NB genellikle metin sınıflandırma (örneğin spam e-posta tespiti) gibi durumlarda etkilidir. Bu tür problemler genellikle kelime varlığını veya yokluğunu temsil eden ikili özelliklere sahiptir. Model yalnızca özelliklerin varlığı veya yokluğunu dikkate aldığı için, sürekli veya daha karmaşık veri türleriyle çalışmada sınırlamalara sahiptir.

2.8.6. Yakınlık tabanlı sınıflayıcılar

KNN sınıflayıcısı, KNN algoritmasına dayanan ve makine öğreniminde yaygın olarak kullanılan bir sınıflandırma yöntemidir [48]. Bu algoritma bir veri noktasını sınıflandırmak için eğitim veri setindeki en yakın K komşusunu dikkate alır ve sınıflandırmayı bu komşuların çoğunluk sınıfına göre yapar [48]. Mesafeleri ölçmek için genellikle Öklid uzaklığı gibi bir metrik kullanılır, ancak Manhattan, Minkowski gibi alternatif metrikler de tercih edilebilir. K parametresi dikkate alınacak komşu sayısını belirler ve bu parametrenin doğru seçimi, modelin performansını doğrudan etkiler. Küçük K değerleri modelin gürültüye duyarlılığını artırırken, büyük K değerleri genelleme kapasitesini artırabilir. Algoritmanın en önemli avantajları arasında basitliği, farklı mesafe metrikleri ile uyulanabilir olması ve küçük veri setlerinde etkili sonuçlar vermesi bulunur.

Yarıçap Tabanlı Komşuluk Sınıflayıcı (Radius Neighbors-RBC), bir veri noktasını sınıflandırmak için belirli bir yarıçap (r) içinde bulunan komşuları dikkate alan denetimli bir makine öğrenimi algoritmasıdır [49]. Bu algoritma geleneksel KNN algoritmasından farklı olarak sabit bir K sayısı yerine, bir veri noktasına belirli bir mesafe aralığında bulunan tüm komşuları göz önünde bulundurur [49]. Komşuların sınıfları arasında çoğunluk oylaması yapılarak tahmin gerçekleştirilir. Özellikle yoğun veri bölgelerinde daha fazla komşuya ulaşma imkânı sunarken, seyrek veri bölgelerinde daha az komşu ile sınıflandırma yapılmasına olanak tanır. RBC genellikle Öklid veya Minkowski mesafesi kullanarak veri noktaları arasındaki benzerlikleri hesaplar. Bu yöntem veri setindeki yerel yoğunluğu dikkate alarak sınıflandırma yapar. Eğer bir veri noktasının yarıçap içinde yeterli komşusu yoksa genellikle sınıfı belirsiz olarak işaretlenir veya varsayılan bir sınıf atanır. Özellikle sınıflar arasındaki sınırların belirsiz olduğu veri setlerinde etkili bir çözüm sunar. Ancak,

yarıçap (r) parametresinin dikkatlice seçilmesi gerekir; küçük bir r değeri aşırı detaylara duyarlılığı artırabilirken, büyük bir r değeri genelleme kapasitesini etkileyebilir.

En Yakın Merkez Sınıflayıcı (Nearest Centroid Classifier-NCC), her sınıfın merkez noktasını hesaplayarak, yeni bir veri noktasını en yakın merkeze atayan basit ama etkili bir sınıflandırma algoritmasıdır [50]. Sınıflandırma işlemi veri noktalarının özellik uzayındaki öklid uzaklığı gibi bir mesafe metriği kullanılarak her sınıfın merkezi ile olan uzaklıklarına dayanır. Sınıf merkezi her sınıfa ait veri noktalarının ortalaması alınarak belirlenir ve bu merkez sınıfı temsil eder. NCC, genellikle az sayıda özellik ve sınırlı miktarda eğitim verisi bulunan durumlarda tercih edilir. Bu algoritma, doğrusal sınırlara dayalı sınıflandırma yapar ve yüksek boyutlu veri setlerinde hızlı bir şekilde sonuç verebilir. Öne çıkan avantajlarından biri, kolay uygulanabilirliği ve düşük hesaplama maliyetidir. Ancak, sınıfların düzgün şekilde ayrılmadığı durumlarda veya veri setinin karmaşık yapılar içerdiği senaryolarda doğruluğu düşebilir.

2.8.7. Diğer sınıflayıcılar

SGD sınıflayıcı (Stochastic Gradient Descent), büyük veri setleri üzerinde hızlı sonuçlar elde etmek amacıyla her adımda küçük veri grupları üzerinden öğrenim yapan bir algoritmadır. Bu yöntem hesaplama maliyetini düşürmesi ve çevrim içi öğrenim kapasitesiyle geniş veri setleri için etkili bir çözüm sunar.

Gauss Süreci sınıflayıcı (Gaussian Process Classifier-GPC), olasılıksal bir sınıflandırma algoritmasıdır ve temelinde Gauss süreci adı verilen bir olasılık modeli yatar. GPC, verilen veri noktalarını temsil eden bir olasılık dağılımı oluşturur ve bu dağılımı kullanarak yeni bir veri noktasının hangi sınıfa ait olabileceğini tahmin eder. Bu yöntem özellikle sınıflandırma problemlerinde karar sınırlarının doğrusal olmadığı durumlarda etkili bir çözüm sunar.

QDA, LDA'nın bir geliştirilmiş versiyonudur. LDA'nın aksine, sınıfların kovaryans matrislerinin eşit olduğunu varsaymaz ve bu nedenle sınıflar arasında kuadratik (doğrusal olmayan) sınırlar çizer. Bu özellik, QDA'nın sınıfların daha karmaşık ve doğrusal olmayan dağılımlarını modelleme kapasitesini artırır. Özellikle sınıflar arasındaki dağılımın doğrusal olmadığı durumlarda, QDA etkili bir sınıflandırıcı olarak öne çıkar.

LSTM (Long Short-Term Memory), ardışık veri yapılarını işlemek ve uzun vadeli bağımlılıkları modellemek için geliştirilmiş bir yapay sinir ağı türüdür. Geleneksel Yinelemeli Sinir Ağları'ndan (RNN) farklı olarak, uzun süreli bilgilerin unutulmasını önlemek için tasarlanmıştır. LSTM'nin temel yapı taşı, giriş kapısı, unutma kapısı ve çıkış kapısı olmak üzere üç ana kapıdan oluşur [51]. Bu kapılar, bilgi akışını kontrol ederek hangi bilgilerin hatırlanması, unutulması ve çıktı olarak verilmesi gerektiğini belirler. Model, her zaman adımında yalnızca mevcut girişleri değil, geçmişteki bilgileri de dikkate alarak tahmin yapar. Bu, uzun süreli bağımlılıkların öğrenilmesini sağlar. Eğitim sırasında genellikle Adam optimizasyon algoritması ve "categorical_crossentropy" kayıp fonksiyonu tercih edilmektedir. LSTM'nin başlıca avantajları arasında, patlayan veya sönümlenen gradyan sorunlarının üstesinden gelmesi, ardışık verilerdeki uzun vadeli ilişkileri etkili bir şekilde öğrenmesi ve geniş bir uygulama alanına sahip olması yer alır [51]. Bununla birlikte, eğitim süreci zaman alıcı olabilir ve yüksek hesaplama kaynakları gerektirebilir. Özellikle büyük veri setlerinde performansı artırmak için hiper parametre optimizasyonu dikkatle yapılmalıdır.

GPC, RBC, Lasso LR, Quadratic Kernel SVC ve Pasif Agresif Sınıflayıcısı'nın, büyük baş hayvanların ivmeölçer verilerinin sınıflandırılmasında daha önce literatürde kullanılmadığı görülmektedir. Önerilen çalışmada kullanılan sınıflayıcılar, bu alandaki uygulamaları açısından bir yenilik sunmaktadır.

GPC, DT sınıflayıcısı, KNN sınıflayıcısı, RBC, NCC, LR, Lasso LR, ENC, Kuadratik Kernel SVC, RBF Kernel SVC, Linear SVC, Pasif Agresif sınıflayıcısı, Perceptron, Voting Hard sınıflayıcısı ve Voting Soft sınıflayıcısı aynı veri setini kullanan çalışmalarda sınıflayıcılarda kullanılmamıştır. Literatürde bu veri setini kullanan çalışmalardan farklı sınıflayıcılar kullanarak bir yenilik sunmaktadır.



3. UYGULAMALAR VE ELDE EDİLEN SONUÇLAR

Bu bölümde, 3. bölümde açıklanan tekniklerin ve yöntemlerin pratik olarak nasıl uygulandığı detaylandırılmaktadır. İlk olarak, veri ön işleme adımları ve sınıflandırma modelleri için kullanılan yazılımlar ve araçlar açıklanmıştır. Daha sonra, geleneksel makine öğrenimi algoritmaları ve LSTM tabanlı derin öğrenme modelinin eğitim ve test süreçleri ele alınmıştır. Bu süreçte kullanılan veri setinin bölünmesi, model eğitimi ve değerlendirme aşamaları adım adım açıklanmış, elde edilen sonuçlar Doğruluk (Accuracy), Makro Kesinlik (Precision / PPV- Positive Predictive Value), Makro Duyarlılık (Recall / Sensitivity), Makro F1 Skoru (F1 Score) ve ortalama FRR (False Rejection Rate - FRR) ile kıyaslanarak sunulmuştur. Ayrıca her bir davranış için yanlış reddetme oranı (False Rejection Rate - FRR) hesaplanmıştır. FRR, bir sınıfa ait verilerin yanlış şekilde reddedilme oranını temsil eder ve özellikle azınlık sınıfların öğrenilme performansını anlamak için kritik bir metriktir. Çapraz doğrulama doğruluğu, her bir katlamadaki doğrulukların ortalaması alınarak hesaplanmaktadır. Doğruluk metriği, modelin genel performansını yansıtmak için kullanılır. Makro metrikler, her sınıfın performansını eşit şekilde değerlendirir. Büyük sınıfların performansının, küçük sınıfların performansını gölgelememesini sağlar.

$$\text{Doğruluk} = \frac{\text{Doğru Pozitif} + \text{Yanlış Pozitif}}{\text{Toplam Örnek Sayısı}} \quad (3.1)$$

$$\text{Kesinlik} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Pozitif}} \quad (3.2)$$

$$\text{Duyarlılık} = \frac{\text{Doğru Pozitif}}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \quad (3.3)$$

$$F1 = 2 \cdot \frac{\text{Kesinlik} \cdot \text{Duyarlılık}}{\text{Kesinlik} + \text{Duyarlılık}} \quad (3.4)$$

$$\text{Makro Kesinlik} = \frac{1}{N} \sum_{i=1}^N \text{Kesinlik}_i \quad (3.5)$$

$$\text{Makro Duyarlılık} = \frac{1}{N} \sum_{i=1}^N \text{Duyarlılık}_i \quad (3.6)$$

$$\text{Makro } F1 \text{ Skoru} = \frac{1}{N} \sum_{i=1}^N F1_i \quad (3.7)$$

$$\text{CV Doğruluğu} = \frac{1}{N} \sum_{i=1}^N \text{Doğruluk}_i \quad (3.8)$$

$$\text{FRR} = \frac{\text{Yanlış Negatif}}{\text{Doğru Pozitif} + \text{Yanlış Negatif}} \quad (3.9)$$

$$\text{Ortalama FRRR} = \frac{1}{N} \sum_{i=1}^N \text{FRR}_i \quad (3.10)$$

Bu performans ölçümleri ile her bir modelin güçlü ve zayıf yönleri tartışılmıştır. Son olarak, modellerin sonuçlarının hayvan davranışlarının sınıflandırılması açısından pratik uygulanabilirliği değerlendirilmiştir.

3.1. Kullanılan Donanım ve Yazılım Altyapısı

Bu çalışmada kullanılan sistem altyapısı ve yazılım araçları, veri işleme, model eğitimi ve performans değerlendirme süreçlerinin verimli bir şekilde gerçekleştirilmesini sağlamıştır. Donanım ve yazılım bileşenleri aşağıda detaylandırılmıştır.

Donanım Özellikleri

İşlemci (CPU): AMD Ryzen 5 5600G ile Radeon Graphics, 3.90 GHz

(RAM): 16 GB DDR4, 3600 MHz.

Depolama: 512 GB NVMe SSD.

Ekran Kartı (GPU): Radeon Graphics (entegre)

İşletim Sistemi: Windows 11 Pro, 64 bit.

Yazılım Özellikleri

Python Sürümü: 3.11.4.

Kullanılan IDE/Çalışma Ortamı: Visual Studio Code

Ek araçlar: MATLAB

Python Kütüphaneleri ve Versiyonları:

NumPy: 1.25.0

Pandas: 2.1.3

Scikit-learn: 1.4.0

TensorFlow: 2.13.0

Matplotlib: 3.7.2

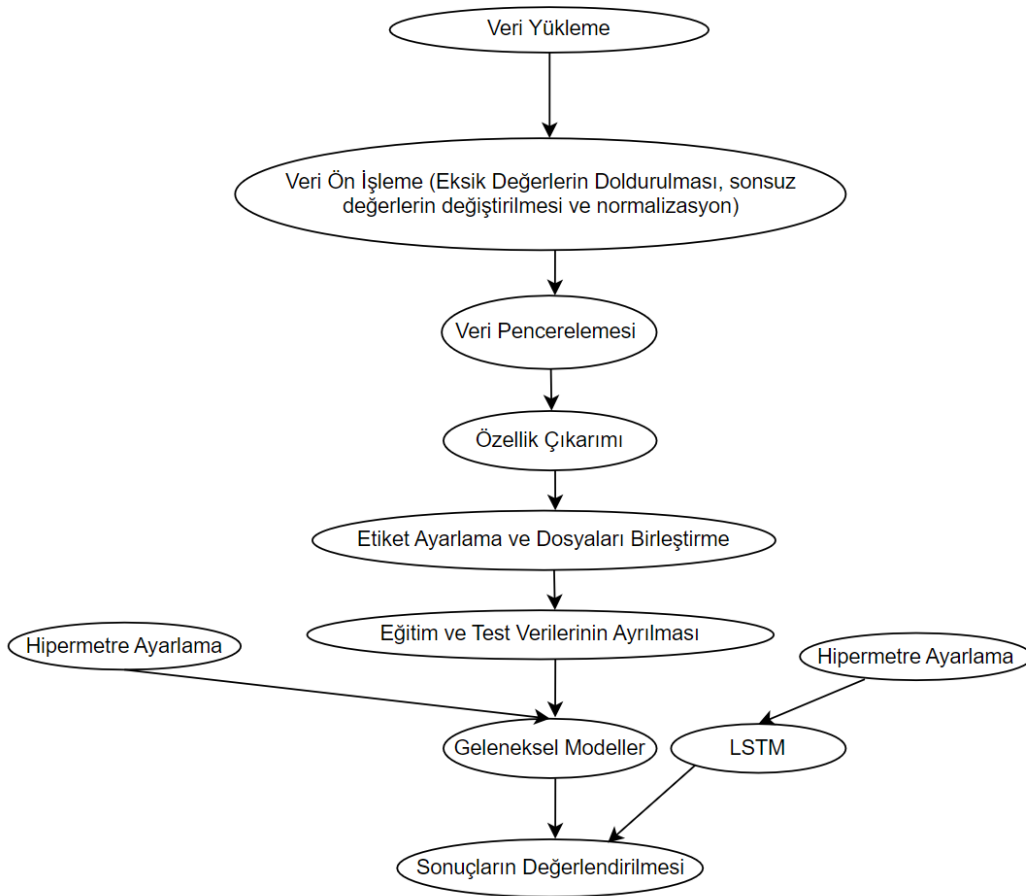
Seaborn: 0.12.2

AntroPy: 0.3.1

Imbalanced-learn: 0.11.0

3.2. Algoritmanın İlerleyişi

Bu alt bölümde, veri ön işleme, özellik çıkarımı, model eğitimi ve sonuç değerlendirme aşamalarından oluşan algoritmanın işleyişi detaylı olarak açıklanmaktadır. Algoritmanın her bir adımı, çalışma sırasında kullanılan araçlar ve yöntemlerle birlikte sunulmuş, sistemin iş akışı Şekil 3.1.'de gösterilmiştir. Bu adımların sırasıyla yönetilmesi, doğru sonuçlara ulaşmak ve sistemin performansını artırmak için kritik öneme sahiptir.



Şekil 3.1. Çalışmanın uygulama süreci

3.3. Özellik Seçimi Süreci

Bu çalışma kapsamında, özellik seçimi için ağaç tabanlı algoritmalarından biri olan ET sınıflayıcı modeli kullanılmıştır. Bu model, özelliklerin hedef değişken üzerindeki etkisini değerlendiren bir özellik önemi (feature importance) metriği sağlar. Bu metrik sayesinde, her bir özelliğin sınıflandırma performansına olan katkısı sayısal olarak ifade edilebilir hale gelmiştir. Bu sayede etkisi çok az olan özellikler kolayca belirlenebilecektir.

Bu çalışmada, toplamda 109 özellik ile başlanılmıştır. Ancak, ET modelinin hesapladığı özellik önemi skorlarına göre daha az etkili olan özellikler elenmiştir. Bu işlemin sonucunda, özellikler arasından en anlamlı katkıyı sağlayanlar belirlenmiştir.

Çizelge 3.1. ET algoritması ile yapılan özellik seçimi işlemi sonrasında elde edilen özelliklerin önemi değerlerini sunmaktadır. Bu tablo, sınıflandırma modeline dahil edilen özelliklerin etkisini gözler önüne sermektedir. Seçim işlemi sonucunda, hedef doğruluk oranına ulaşılmasında kritik öneme sahip özelliklerin belirlenmesi sağlanmıştır.

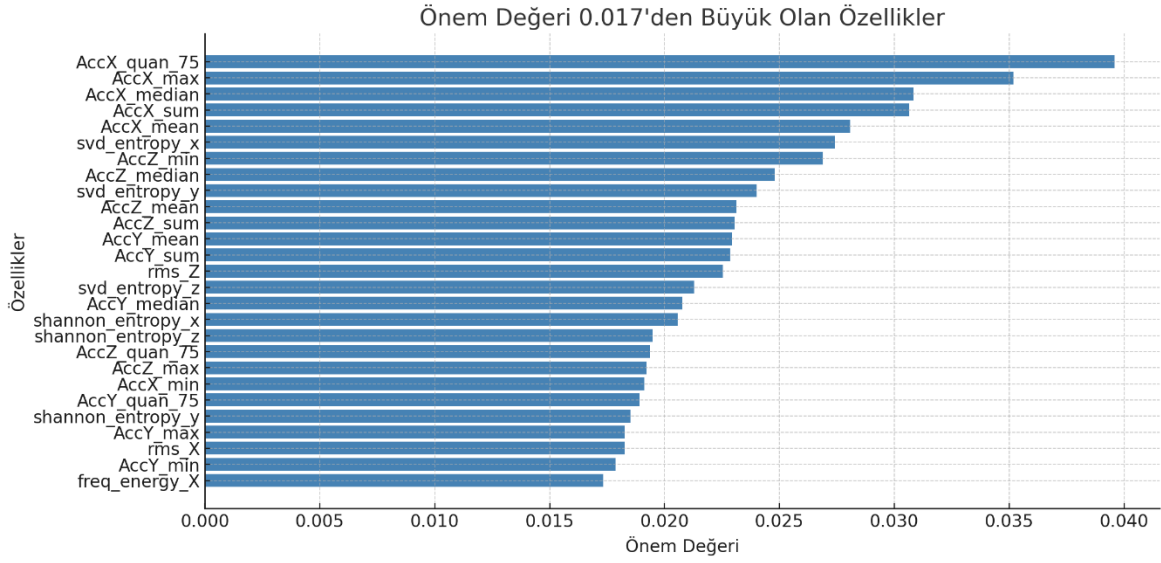
Çizelge 3.1. çalışmanın ilerleyen kısımlarında referans alınmıştır ve model performansı üzerindeki etkileri detaylı bir şekilde analiz edilmiştir. Bu analiz sonucunda, daha az özellik kullanılarak hızlı ve doğru sınıflandırma yapılabileceği gösterilmiştir.

Çizelge 3.1. Özellik seçimi işlemi sonrası belirlenen önem değerleri

Sıra	Özellik	Önem	Sıra	Özellik	Önem
1	AccX_quan_75	0,039577	55	spectral_entropy_z	0,003832
2	AccX_max	0,035193	56	zcr_X	0,003706
3	AccX_median	0,030817	57	perm_entropy_z	0,003616
4	AccX_sum	0,030639	58	higuchi_x	0,003516
5	AccX_mean	0,028079	59	AccX_TT	0,003352
6	svd_entropy_x	0,0274	60	AccZ_contrast	0,003352
7	AccZ_min	0,026884	61	dominant_freq_Y	0,003305
8	AccZ_median	0,024788	62	AccZ_VRR	0,003212
8	svd_entropy_y	0,024013	63	AccX_DRR	0,002987
9	AccZ_mean	0,023126	64	AccY_VRR	0,002982
10	AccZ_sum	0,023054	65	dominant_freq_Z	0,002907
11	AccY_mean	0,022938	66	katz_x	0,002856
12	AccY_sum	0,022855	67	katz_z	0,002826
13	rms_Z	0,02254	68	higuchi_z	0,002819

Çizelge 3.1. (devam) Özellik seçimi işlemi sonrası belirlenen önem değerleri

14	svd_entropy_z	0,021296	69	AccY_contrast	0,00279
15	AccY_median	0,020768	70	AccY_RR	0,002789
16	shannon_entropy_x	0,020566	71	AccZ_TT	0,002677
17	shannon_entropy_z	0,019469	72	AccZ_correlation	0,002583
18	AccZ_quan_75	0,019374	73	AccY_kurt	0,002551
19	AccZ_max	0,019201	74	AccY_correlation	0,00249
20	AccX_min	0,019131	75	AccY_TT	0,002487
21	AccY_quan_75	0,018923	76	AccX_skew	0,00248
22	shannon_entropy_y	0,018519	77	AccX_Lmean	0,00241
23	AccY_max	0,018265	78	higuchi_y	0,002274
24	rms_X	0,01826	79	AccX_DET	0,002193
25	AccY_min	0,017858	80	AccX_ENTR	0,002185
26	freq_energy_X	0,017328	81	AccZ_DRR	0,002126
27	rms_Y	0,016869	82	AccY_LAM	0,002064
28	freq_energy_Z	0,015055	83	AccX_LAM	0,001969
29	AccX_mad	0,014603	84	katz_y	0,001961
30	MovementVariation	0,013397	85	AccZ_ENTR	0,00183
31	VectorMagnitude	0,012872	86	AccY_skew	0,001786
32	AccZ_mad	0,012658	87	zcr_Z	0,001701
33	Energy	0,012615	88	AccX_kurt	0,001607
34	AccY_std	0,010962	89	AccY_Lmean	0,001596
35	freq_energy_Y	0,010891	90	AccY_ENTR	0,001577
36	AccX_std	0,010623	91	app_entropy_z	0,001554
37	AccZ_std	0,01048	92	AccZ_Lmean	0,001548
38	SignalMagnitudeArea	0,010207	93	AccZ_kurt	0,001514
39	AccY_mad	0,009709	94	AccZ_skew	0,001496
40	spectral_entropy_y	0,009545	95	AccZ_LAM	0,001492
41	perm_entropy_x	0,009497	96	app_entropy_x	0,001481
42	spectral_entropy_x	0,008609	97	AccY_DRR	0,001472
43	perm_entropy_y	0,007572	98	sample_entropy_x	0,001373
44	AccX_var	0,00593	99	AccZ_DET	0,001369
45	euclidean_dist	0,005706	100	app_entropy_y	0,001338
46	AccZ_var	0,005676	101	AccY_DET	0,001325
47	AccX_RR	0,00552	102	AccZ_Trend	0,001258
48	AccX_contrast	0,005306	103	sample_entropy_z	0,00125
49	AccY_var	0,005204	104	AccX_Trend	0,001232
50	dominant_freq_X	0,005053	105	AccY_Trend	0,001201
51	AccX_correlation	0,004836	106	sample_entropy_y	0,001193
52	zcr_Y	0,004607	107	AccX_FirstRecurrenceTime	0,000742
53	AccX_VRR	0,004478	108	AccZ_FirstRecurrenceTime	0,000123
54	AccZ_RR	0,004183	109	AccY_FirstRecurrenceTime	0,000122



Şekil 3.2. Seçilen özelliklerin gösterimi

3.4. Model Eğitim Süreci

Bu alt bölümde, geleneksel makine öğrenimi algoritmaları ve LSTM tabanlı derin öğrenme modelinin eğitim süreci detaylı bir şekilde açıklanmıştır. Model eğitim süreci, veri setinin hazırlanmasından, modellerin eğitilmesine ve performans metriklerinin değerlendirilmesine kadar olan adımları kapsamaktadır.

Model eğitime başlamadan önce veri seti, bölüm 2’de bahsedilen adımlardan geçirilmiştir. Veri setinde “train_test_split” fonksiyonu kullanılarak %80 eğitim ve %20 test oranında bölünmüştür. Eğitim ve test veri setlerine ayrılan bu oran, modellerin genelleme performansını değerlendirmek açısından kritik öneme sahiptir. Veri setindeki sınıf dengesizlikleri, özellikle azınlık sınıfların öğrenme performansını artırmak amacıyla SMOTE yöntemiyle düzeltilmiştir. Bu işlem sonucunda, az örnek içeren DRN ve FES sayısı artırılarak modellerin bu sınıfları daha iyi öğrenmesi sağlanmıştır. Özellikle gradyan tabanlı algoritmalar için özelliklerin aynı ölçeğe getirilmesi model performansını olumlu yönde etkilemiştir.

Model eğitim sürecinde Çizelge 3.1.’de verilen 109 adet özelliğin hepsi kullanılmamıştır. Özellik çıkarımı yöntemi olarak seçilen ET sınıflayıcısı ile bir eşik değeri belirlenmiştir. Bu eşik değerinin üzerindeki özellikler ele alınmıştır. Eşik değeri olarak 0,017 seçilmiş ve böylece 27 adet özellik kullanılmıştır. Bu seçilen özellikler Şekil 3.2.’de detaylı olarak

gösterilmiştir. Çizelge 3.2.'de bu özelliklerin hangi eksenlerde daha çok olduğu gösterilmiştir. Buna göre model sınıflandırma başarısına en çok katkı veren eksen X eksenidir. Bu çalışmada seçilen özelliklerin her biri modelin genelleme yeteneğine olan katkısını artırmak amacıyla belirlenmiştir. Çizelge 3.3.'te seçilen özelliklerin ortalama ve standart sapma değerleri detaylı bir şekilde verilmiştir.

Çapraz Doğrulama (Cross-Validation) olarak $cv=10$ seçilmiştir. Yani 10 katlı çapraz doğrulama gerçekleştirilmiştir. Bu yöntem, veri setini 10 eşit parçaya bölerek her parçada modelin performansını değerlendirmektedir. Böylece, modelin genelleştirme başarısını anlamak için güvenilir bir metrik sağlanmıştır. Hiper parametre optimizasyonu için Grid Search yöntemi uygulanmıştır. Parametreler, her model için ayrı ayrı optimize edilmiştir. Grid Search, modelin farklı parametre kombinasyonlarını deneyerek en iyi sonucu veren parametreleri seçmiştir. Ama modelin aşırı öğrenme riskinden dolayı Grid Search parametreleri belirlenirken maksimum değerleri kullanmaktan kaçınılmış, hatta Grid Search parametrelerinin ortanca değerleri maksimum olarak belirlenmiştir. Kullanılan Grid Search parametreleri EKLER kısmına eklenmiştir.

Bu çalışmada, ET Sınıflayıcı, GB ve SVC gibi popüler algoritmalar kullanılmıştır. Modeller, scikit-learn kütüphaneleriyle eğitilmiş; bu algoritmalar, eğitim süresi açısından oldukça hızlı sonuçlar sunmuş ve her bir model, karmaşıklığına bağlı olarak saniyeler veya birkaç dakika içinde tamamlanmıştır. Doğrusal Modeller, özellikle sınıflandırma ve düzenleme teknikleriyle aşırı öğrenmeyi önlemede etkilidir. Bu çalışmada LR, Lasso LR ve Elastik Ağ LR gibi modeller değerlendirilmiştir.

Karar ağacı tabanlı algoritmalar, yüksek yorumlana bilirlik ve genelleme kapasitesi ile ön plana çıkar. Bu çalışmada DT sınıflayıcısı, Extra Tree sınıflayıcısı ve RF sınıflayıcısı modelleri değerlendirilmiştir. RF, özellikle aşırı öğrenmeyi önlemesi ve güçlü performansı ile dikkat çekmiştir. Extra Tree sınıflayıcısı ise geçmiş çalışmalarda elde edildiği üzere yüksek doğruluk oranı sağlamıştır. KNN, farklı veri yapılarında etkili bir komşuluk tabanlı algoritma olarak bu çalışmada değerlendirilmiştir.

Topluluk yöntemleri, sınıflandırma doğruluğunu artırma amacıyla bu çalışmada geniş bir şekilde değerlendirilmiştir. GB, Hist GB Sınıflayıcı (HGBC) ve Voting sınıflayıcı gibi yöntemler birden fazla modelin sonuçlarını bir araya getirerek performansı artırmıştır.

Bu çalışmada, sadece RP ile çıkarılan özellikler kullanılarak model eğitimi de gerçekleştirilmiştir. RP tabanlı özellikler modele dahil edilerek, bu özelliklerin sınıflandırma performansına etkisi analiz edilmiştir. RP tabanlı özellikler için de yine aynı sınıflandırmalar, aynı çapraz doğrulama kullanılmış olup sadece özellikler kısmında farklılık oluşmuştur.

Çizelge 3.2. Kullanılan özelliklerin eksenlerle ilişkisi

Eksen	Özellik Sayısı	Oran (%)
X Ekseni	10	38,46
Y Ekseni	9	34,62
Z Ekseni	7	26,92

3.5. Performans Değerlendirme ve Karşılaştırma

Bu çalışmada kullanılan özellikler, sınıflandırma modellerinin performansını artırmak ve veri setindeki davranışları daha iyi temsil etmek amacıyla özenle seçilmiştir. Özellikler, her bir ekseninde hızlanma, karmaşıklık, bilgi yoğunluğu ve davranış çeşitliliğini yansıtan metriklerden oluşmaktadır. Özelliklerin ortalama ve standart sapma değerleri Çizelge 3.3.'te verilmiştir. Ortalama (mean), medyan (median), maksimum (max), RMS ve toplam değerler (sum) gibi özellikler, hareketin genel yoğunluğunu ve büyüklüğünü ölçmek için kullanılmaktadır. Örneğin, max değerleri ani ve yüksek yoğunluklu hareketleri tespit etmekte etkili olup, bu tür hareketler genellikle aktif davranışlarla ilişkilidir. Bu hareketlere koşma ve otlama örnek olarak verilebilir. Benzer şekilde, toplam ve ortalama değerleri hareketin genel seviyesini anlamak için kullanılmıştır. Bu özellikler hem kısa süreli hareketlerin hem de uzun süreli durgun davranışların analizinde önemli rol oynar.

Minimum değerler, düşük hareketlilik içeren davranışların tespitinde önemlidir. Bu değerler, hareketin en alt seviyesini ölçerek düşük enerji gerektiren davranışların sınıflandırılmasında kritik bir rol oynar. Entropi değerleri, hareketlerin rastgelelik düzeyini ölçmek için kullanılmıştır. Özellikle, Shannon Entropisi hareketlerin düzenini ve bilgi içeriğini analiz ederken, SVD Entropisi hareketlerin dinamik yapısını ve karmaşıklığını anlamak için önemlidir. Daha yüksek entropi değerleri, daha rastgele ve karmaşık davranışları temsil ederken, düşük entropi değerleri daha düzenli davranışlarla ilişkilendirilebilir. Frekans enerjisi, hareketlerin tekrarlanma paternini analiz etmek için

kullanılmıştır. Bu özellik, özellikle sürekli hareketlerin analizinde etkilidir. Sürekli hareketlere geviş getirme, yürüme örnek olarak verilebilir.

Çizelge 3.3. Kullanılan özelliklerin ortalaması ve standart sapması

Özellik	Ortalama	Standart Sapma
AccX_quan_75	-0,04428	0,10366
AccX_max	-0,02379	0,121743
AccX_median	-0,05423	0,100609
AccX_sum	-1,34341	2,51543
AccX_mean	-0,05374	0,100617
svd_entropy_x	0,565618	0,439892
AccZ_min	-0,17026	0,205561
AccZ_median	-0,14183	0,202158
svd_entropy_y	0,441484	0,364292
AccZ_mean	-0,14157	0,202086
AccZ_sum	-3,5393	5,052138
AccY_mean	-0,05276	0,083767
AccY_sum	-1,31892	2,09417
rms_Z	0,21466	0,127226
svd_entropy_z	0,308867	0,33781
AccY_median	-0,05337	0,084511
shannon_entropy_x	3,924045	0,71185
shannon_entropy_z	3,964084	0,646183
AccZ_quan_75	-0,13113	0,20529
AccZ_max	-0,11052	0,217714
AccX_min	-0,08147	0,107047
AccY_quan_75	-0,0406	0,088861
shannon_entropy_y	4,024608	0,634343
AccY_max	-0,01644	0,115559
rms_X	0,092402	0,073171
AccY_min	-0,08476	0,105357
freq_energy_X	8,682394	10,2293

Şekil 3.8.'de bu modellerin karışıklık matrisleri verilmiştir. Bu durumda kullanılan dört farklı modelin doğruluk (%), CV doğruluk, Macro Precision (PPV), Macro Recall (Sensitivity) ve Macro F1 Score değerleri incelenmiştir. Ancak, sonuçlar incelendiğinde Macro Precision, Macro Recall ve Macro F1 Score değerlerinin nispeten düşük olduğu görülmektedir. Özellikle, azınlık sınıfların düşük örnek sayısı nedeniyle bu metriklerdeki düşük performans, sınıflandırma modellerinin azınlık sınıfları doğru bir şekilde öğrenemediğini göstermektedir. Bu nedenle, veri setindeki dengesizlikleri gidermek ve modellerin performansını artırmak amacıyla özellik çıkarımı işleminden sonra bazı azınlık sınıfları tamamen çıkarılmıştır. Daha sonra model eğitimi öncesinde 2 adet azınlık sınıfı

(DRN ve FES) SMOTE yöntemi ile arttırılmıştır. Bu işlemler, azınlık sınıfların model üzerindeki etkisini daha iyi analiz etmek ve genel model performansını arttırmak için yapılmıştır.

Çizelge 3.4. Azınlık sınıfları çıkarılmadan sınıflama sonuçları

Sınıflama Modeli	Doğruluk (%)	CV Doğruluk	Macro Precision (PPV)	Macro Recall (Sensitivity)	Macro F1 Score
ET	95,67	95,42	88,40	68,05	72,65
HGBC	93,93	93,78	75,72	64,71	67,17
KNN	93,38	93,08	74,34	70,97	70,67
SGD	94,46	94,24	82,80	64,91	68,86

Çizelge 3.5.'te farklı sınıflandırma modellerinin performans karşılaştırması sunulmaktadır. Çizelge 3.4.'te bulunan performans değerleri ile Çizelge 3.5.'teki değerler karşılaştırıldığında dengeli bir veri setinin sınıflandırma modellerinin performansında ne kadar önemli bir rol oynadığını bir kez daha ortaya koymaktadır. Ayrıca çizelge içerisinde bulunan bu metrikler, modellerin genel performansını anlamak ve karşılaştırmak için kritik bir rol oynamaktadır.

ET, 98,18% doğruluk oranı, %96,73 Precision ve %97,24 Recall değerleriyle en yüksek performansı göstermiştir. Bu model, diğer yöntemlere göre daha dengeli ve üstün bir performans sunmuştur. ET modeli, karar ağaçlarının rastgele ayrımlar ve büyük sayıda ağaç ile birleştirilmesi sayesinde güçlü genelleme kapasitesine sahiptir. Ensemble (topluluk) yöntemi sayesinde, aşırı öğrenmeyi (overfitting) azaltarak hem eğitim hem de test verisinde yüksek doğruluk sağlar. Bağımsız olarak oluşturulan ağaçlardan gelen bilgi birleşerek modelin hata oranını azaltır. Veri setindeki tüm eksenlerde iyi genelleme yapabildiği için, FRR düşük kalmıştır. ET modeli, özellikle DRN ve FES gibi davranışları hatasız sınıflandırarak (%100 doğruluk, %0FRR) bu kategorilerde en yüksek doğruluğu sergilemiştir. Ayrıca SLT ile (%0,99 doğruluk ve %0,58 FRR), RES ile (%0,98 doğruluk ve %2,3 FRR), RUS ile (%0,99 doğruluk ve %3,05 FRR), ve GRZ ile (%0,96 doğruluk ve %4,3 FRR) elde edilmiştir. Bu sınıflarda da çok düşük hata oranı elde edilmiştir. Bununla birlikte, MOV davranışında gözlemlenen (%0,91 doğruluk ve %8,9 FRR) modelin bu sınıfı ayırt etmede diğerlerine kıyasla daha fazla zorlandığını göstermektedir. ET sınıflayıcısının

%10'un altında yanlış reddetme oranlarına sahip olması, modelin başarılı bir şekilde genelleme yaptığını ortaya koymaktadır. ET sınıflayıcısının %10'un altında yanlış reddetme oranlarına sahip olması, modelin başarılı bir şekilde genelleme yaptığını ortaya koymaktadır.

RF ve Bagging sınıflayıcı, birden fazla karar ağacını kullanarak çeşitlendirilmiş tahminler yapar. Bu modeller, aşırı öğrenmeyi önlemek için veri alt kümeleri üzerinde çalışır ve düşük varyanslı, kararlı tahminler üretir. HGBC, aşamalı olarak hataları düzelten bir yapıdadır ve özellikle karmaşık karar sınırlarını öğrenmede başarılıdır. Voting Soft sınıflayıcı ise, birden fazla güçlü modelin olasılık tabanlı tahminlerini birleştirerek daha dengeli ve güvenilir sonuçlar sunar. Bu modeller, sınıflar arasındaki doğrusal olmayan ayrımları daha iyi öğrenebildiğinden, yüksek doğruluk ve F1 skoru elde etmişlerdir. Özellikle dengesiz veri setlerinde daha iyi genelleme yaparak azınlık sınıflarda da başarılı olmuşlardır.

LSTM modeli, zaman serisi verilerindeki uzun vadeli bağımlılıkları öğrenme kapasitesi ile öne çıkmıştır. Model, %92,70 doğruluk oranı, %90,60 Precision ve %90,01 Recall değerleriyle, özellikle hareket dinamiklerini anlamada başarılı olmuştur. Ancak, LSTM modeli, eğitim süresi ve karmaşıklık açısından geleneksel sınıflayıcılara göre daha yüksek bir maliyetle çalışmıştır. Performansın optimizasyonu için daha karmaşık özellik setleri veya hiper parametre ayarlarının iyileştirilmesi gerekliliği öne çıkmaktadır. Yine de, LSTM'nin zaman serisi verilerinde ardışık ilişkileri modelleme yeteneği, uzun vadeli bağımlılıkların analiz edilmesi gereken durumlarda bir avantaj sağlamaktadır.

Daha düşük performans gösteren modeller arasında Bernoulli NB, ve Ada Boost Sınıflayıcı gibi modeller bulunmaktadır. Bu modellerin düşük doğruluk ve F1 skorları, verinin doğası gereği etkisiz kaldıklarını göstermektedir. Bu algoritmaların düşük performansı, özellikle veri setinin karmaşık yapısından ve sınıflar arasındaki ayrımların doğrusal olmamasından kaynaklanmaktadır. Bernoulli NB, özellikle ikili özelliklere duyarlı bir model olduğundan, verinin çok boyutlu ve sürekli yapısında sınırlı kalmıştır. Naive Bayes modelleri, değişkenler arasındaki bağımsızlık varsayımıyla çalışır. Ancak bu çalışmadaki veri seti, eksenler arasında yüksek korelasyona sahiptir. Bu korelasyon, Naive Bayes'in yaptığı bağımsızlık varsayımını ihlal ettiği için model zayıf sonuç vermiştir. AdaBoost modeli, aşırı öğrenmeye (overfitting) yatkındır ve aykırı değerlere çok duyarlıdır. Her yanlış tahmin edilen veriye daha fazla ağırlık verdiği için, veri setindeki küçük sapmalar bile modelin

genel başarısını düşürebilir. Özellikle dengesiz veri setlerinde zayıf öğrenicileri kullandıktan, azınlık sınıflarda başarısız olmuştur.

NCC, sınıflar arasında net bir merkez ayırımının olmadığı durumlarda zayıf performans göstermiştir ve bu da özellikle veri noktalarının yoğun olduğu karmaşık sınıflarda yanlış sınıflandırmalara yol açmıştır. Ayrıca, NCC modeli aykırı değerlere duyarlı olduğu için, az sayıda yanlış konumlandırılmış veri noktası bile sınıfların merkezlerini kaydırarak genel doğruluğu düşürebilir. Linear Ridge ise, doğrusal ilişkiler üzerine kurulu bir model olduğu için doğrusal olmayan sınıflandırma sınırlarında düşük başarı sergilemiştir.

Çizelge 3.5. Sınıflandırma modellerinin test doğruluğu performans karşılaştırması

Sınıflama Modeli	Doğruluk (%)	CV Doğruluk (%)	Precision (PPV) (%)	Recall (Sensitivity) (%)	F1 Score (%)	Ortalama FRR (%)
Linear Ridge	76,45	76,49	66,67	54,47	55,66	45,52
Doğrusal Diskriminant Analizi	80,19	80,11	74,60	77,11	75,74	22,89
Kuadratik Diskriminant Analizi	83,98	86,24	80,67	86,13	82,39	12,67
SGD Sınıflayıcısı	87,33	86,92	85,92	86,23	85,90	13,62
GPC (Gauss Süreci Sınıflayıcı)	95,49	95,12	93,51	94,27	93,85	7,23
Gaussian NB	71,40	71,76	68,04	75,38	70,06	24,61
Bernoulli NB	61,38	61,45	66,81	69,55	63,52	30,44
Karar Ağacı Sınıflayıcısı	96,48	96,08	94,33	94,45	94,37	5,46
Bagging Sınıflayıcısı	98,00	97,91	96,72	97,14	96,92	2,86
ET Sınıflayıcısı	98,18	97,96	96,73	97,24	96,98	2,75
Ada Boost Sınıflayıcısı	70,34	69,34	75,66	71,14	72,05	28,86
Gradyan Artırma Sınıflayıcısı	97,50	97,45	96,04	95,96	96,00	4,03
HGBC	98,02	97,73	96,76	97,23	96,99	3,00
Rastgele Orman Sınıflayıcısı	97,89	97,70	96,44	96,93	96,67	3,26
KNN Sınıflayıcı	97,48	97,49	95,84	96,60	96,20	3,40
RBC	72,06	71,30	58,68	43,93	43,22	56,06
NCC	46,20	47,01	40,23	58,36	42,69	41,64
Lojistik Regresyon	87,69	87,35	85,78	86,48	86,09	12,88

Çizelge 3.5. (devam) Sınıflandırma modellerinin test doğruluğu performans karşılaştırması

Lasso Lojistik Regresyon	87,63	87,30	85,86	86,83	86,27	13,14
ENC (Esnek Ağ Sınıflayıcısı)	87,81	87,39	85,77	86,79	86,22	13,20
SVC	96,78	96,84	95,03	95,69	95,34	4,30
Kuadritik Kernel SVC	95,22	95,27	93,27	93,89	93,55	5,59
RBF Kernel SVC	96,78	96,84	95,03	95,70	95,35	4,25
Linear SVC	87,24	87,06	85,22	86,48	85,76	12,49
Pasif Agresif Sınıflayıcı	84,36	81,98	83,70	80,95	81,82	17,95
Perceptron	84,20	83,60	81,33	82,57	81,24	17,43
Voting Hard Sınıflayıcı	96,28	96,15	94,63	93,16	93,85	6,49
Voting Soft Sınıflayıcı	97,50	97,52	95,92	96,34	96,11	3,65
LSTM	92,70	93,02	90,60	90,01	90,22	7,89

Çizelge 3.6.'da farklı sınıflayıcılar için hesaplanan FRR değerleri sunulmuştur. Genel olarak azınlık sınıflar (örneğin DRN ve FES) için yanlış reddetme oranlarının daha yüksek olduğu gözlemlenmiştir. Bu durum, bu sınıfların model tarafından yeterince öğrenilemediğini veya diğer sınıflarla ayırım yapmada zorlandığını göstermektedir.

Özellikle, Bernoulli NB, Naive Bayes ve AdaBoost gibi algoritmaların FRR oranlarının oldukça yüksek olduğu görülmüştür. Bunun sebebi, bu algoritmaların genellikle doğrusal olmayan ayrımları öğrenmede yetersiz kalması ve verinin dağılımına duyarlı olmalarıdır. Buna karşılık, ET, Bagging, HGBC ve Voting Soft gibi sınıflayıcılar daha düşük FRR oranları ile güçlü performans sergilemiştir. Bu algoritmalar daha iyi genelleme yaparak sınıfların doğru şekilde tespit edilmesini sağlamış ve yanlış reddetme oranlarını minimize etmiştir.

Modellerin sınıflandırma yetenekleri, verinin özellikleri, sınıf dengesi ve hiper parametre optimizasyonu gibi birçok faktörden etkilenmiştir. Şekil 3.3., 3.4., 3.5., 3.6. ve 3.7.'de karışıklık matrisleri verilmiştir.

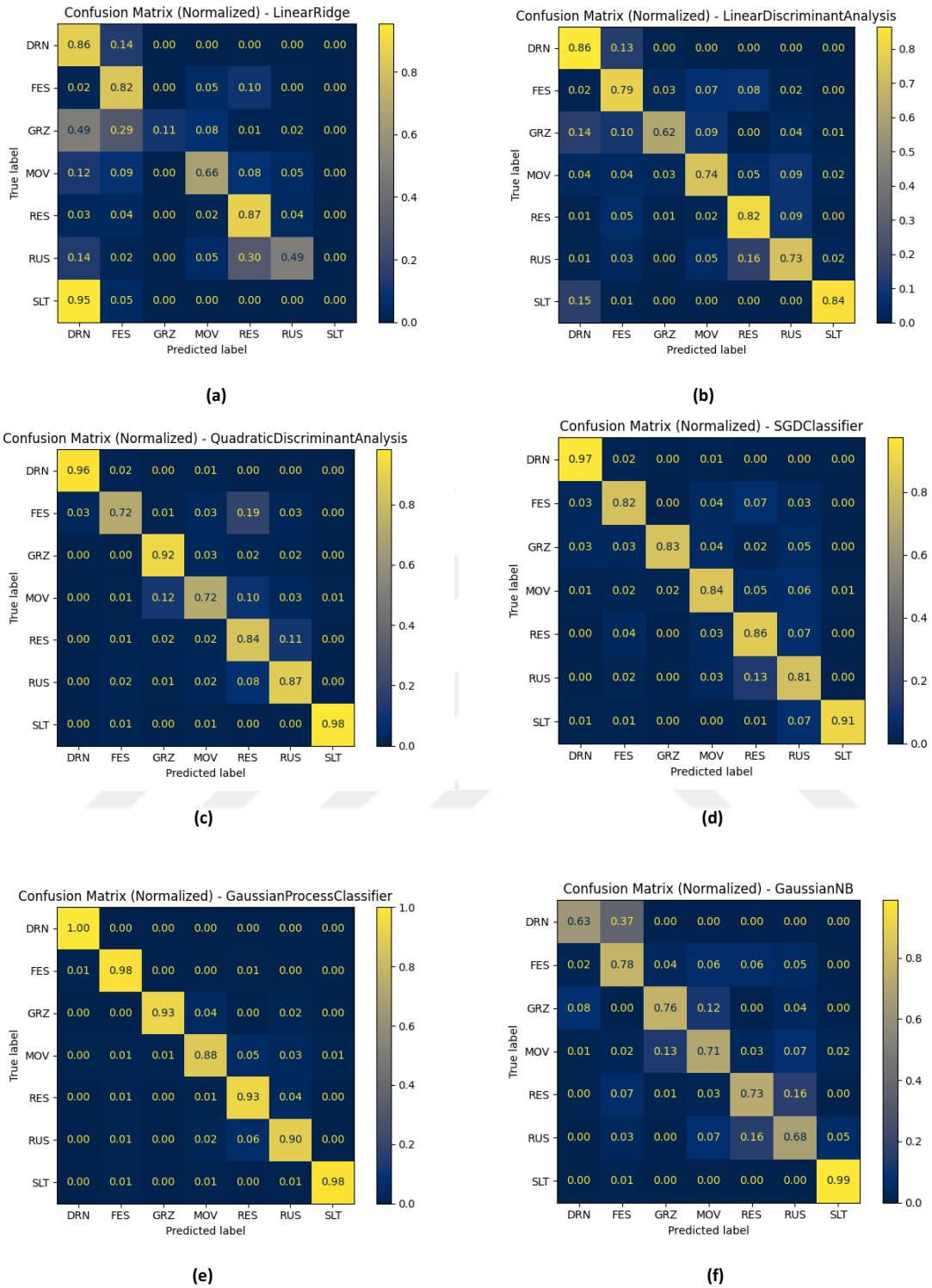
Çizelge 3.6. Sınıfların yanlış reddetme oranları (FRR, %)

Sınıflayıcı	DRN FRR	FES FRR	GRZ FRR	MOV FRR	RES FRR	RUS FRR	SLT FRR
Linear Ridge	13,85	17,78	88,88	34,36	13,12	50,67	99,99
Doğrusal Diskriminant Analizi	13,61	21,23	38,00	26,32	18,12	27,03	15,88
Kuadratik Discriminant	2,12	15,46	8,24	32,75	16,14	12,22	1,76
SGD Sınıflayıcısı	1,85	16,61	18,64	21,49	14,16	18,44	4,11
GPC (Gauss Süreci Sınıflayıcı)	0	0,75	12,24	14,81	8,73	12,11	2,88
Gaussian NB	37,05	22,44	23,66	28,62	27,00	32,35	1,17
Bernoulli NB	49,24	18,58	25,80	37,93	59,84	16,97	4,70
Karar Ağacı	0,11	0,44	10,03	12,30	4,41	9,16	2,35
Bagging Sınıflayıcısı	0	0,12	3,5	8,5	2,7	4,5	0,58
ET Sınıflayıcısı	0	0	4,3	8,9	2,3	3,05	0,58
Ada Boost Sınıflayıcısı	23,44	34,40	51,97	17,58	31,57	38,91	4,12
Gradyan Artırma Sınıflayıcısı	0	0,28	8,24	9,31	3,14	6,11	1,18
HGBC	0,04	0	4,3	8,5	3,01	3,95	1,17
Rastgele Orman	0	0,08	5,01	9,43	3,09	4,64	0,59
KNN Sınıflayıcı	0	0	3,94	10,69	4,08	4,52	0,57
RBC	2,12	16,01	99,99	93,10	18,11	63,12	99,99
NCC	99,88	31,91	31,54	34,36	41,23	51,36	1,18
Lojistik Regresyon	1,90	14,49	16,13	20,11	15,19	18,21	4,12
Lasso Lojistik Regresyon	2,36	16,18	17,92	19,20	14,07	19,34	2,94
ENC (Esnek Ağ Sınıflayıcısı)	1,97	14,81	16,49	19,77	15,19	18,89	5,29
SVC	0	0,6	5,73	10,11	5,24	7,24	1,17
Kuadratik Kernel SVC	0	1,12	6,81	13,22	8,08	7,58	2,35
RBF Kernel SVC	0	6,02	5,73	10,11	5,24	7,24	1,15
Linear SVC	2,01	16,74	17,92	17,12	13,33	18,55	1,76
Pasif Agresif Sınıflayıcı	1,32	20,63	20,43	16,90	12,05	30,77	23,52
Perceptron	1,43	20,35	10,75	16,55	25,79	15,38	31,76
Voting Hard Sınıflayıcı	0	0,52	16,13	9,42	4,62	10,63	4,12
Voting Soft Sınıflayıcı	0	0,08	5,73	9,65	3,54	5,99	0,58
LSTM	4,54	20,14	4,59	10,37	6,50	6,40	2,66

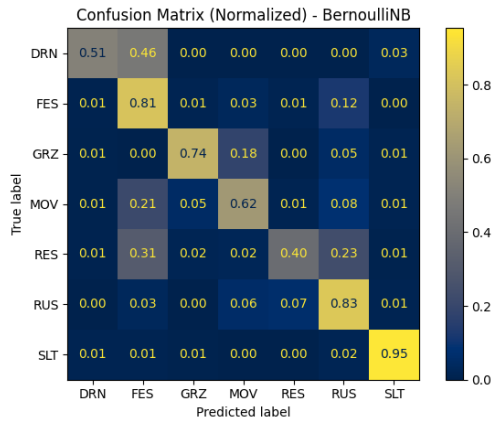
Özellik seçimi sürecinde yalnızca RP tabanlı özellikler kullanılarak da bir model eğitilmiş ve elde edilen sonuçlar incelenmiştir. RP tabanlı özelliklerin hareketin tekrar eden dinamiklerini yansıttığı göz önünde bulundurularak, bu özelliklerin sınıflandırma başarısına olan katkısı analiz edilmiştir. Çizelge 3.7.'de verilen sonuçlara göre yalnızca RP özellikleriyle eğitilen modellerin bazılarının doğruluğunun yüksek çıktığı gözlemlenmiştir. Bu modelde de en yüksek sonuç ET, 85,73% doğruluk oranı, 70,08% Precision ve %63,50 Recall değerleriyle en yüksek performansı göstermiştir. Bu bulgu, RP tabanlı özelliklerin, zaman serisi verilerinde belirli desenleri ve dinamikleri başarıyla yakalayarak sınıflandırma performansına önemli katkı sağlayabileceğini göstermektedir. Ancak, RP özellikleri ile elde edilen doğruluk değerlerinin, diğer geleneksel özellikler ile karşılaştırıldığında düşük bir sonuç verdiği de gözlemlenmektedir. Bu durumun temel nedeni, RP analizinin zamansal tekrarları daha geniş zaman pencerelerinde daha iyi yakalayabilmesi ve 1 saniyelik pencerenin bu tür analizler için kısıtlı bir bağlam sunmamasıdır.

Çizelge 3.7. Yineleme grafiği özellikleri kullanılarak elde edilen sınıflama performansları

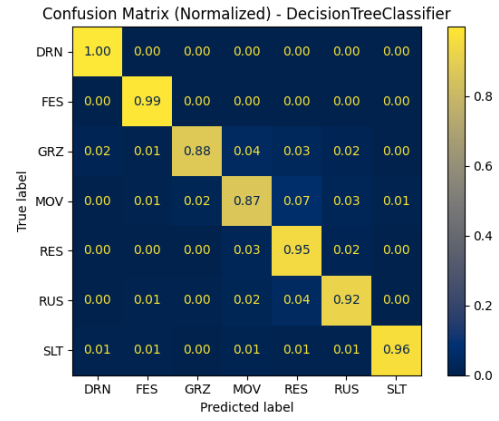
Sınıflama Modeli	Doğruluk (%)	CV Doğruluk (%)	Precision (PPV) (%)	Recall (Sensitivity) (%)	F1 Score (%)	Ortalama FRR (%)
Linear Ridge	57,15	56,87	61,36	35,55	34,82	64,44
Doğrusal Diskriminant Analizi	57,47	57,12	52,98	38,96	40,73	61,04
Kuadratik Diskriminant Analizi	49,60	49,32	41,60	43,14	37,54	56,86
SGD Sınıflayıcısı	54,43	54,80	46,65	40,90	41,69	59,92
GPC (Gauss Süreci Sınıflayıcı)	78,23	77,56	58,12	54,87	55,43	37,64
Gaussian NB	36,03	36,24	33,45	40,07	28,42	59,93
Bernoulli NB	43,98	43,84	36,47	36,70	33,90	63,30
Karar Ağacı	76,02	74,89	57,48	56,30	56,70	43,69
Bagging Sınıflayıcısı	85,33	84,90	70,08	64,65	66,02	35,34
ET Sınıflayıcısı	85,73	84,84	72,74	63,50	65,03	36,49
Ada Boost Sınıflayıcısı	68,85	67,90	57,55	55,89	56,07	44,10
Gradyan Artırma Sınıflayıcısı	83,82	83,06	69,01	66,17	67,12	33,82
HGBC	84,91	84,62	68,84	65,56	66,60	34,43
Rastgele Orman	84,44	83,72	71,47	62,96	64,54	37,04
KNN Sınıflayıcı	73,97	72,60	56,91	50,34	51,04	49,66
RBC	39,55	38,50	25,87	21,55	17,44	78,45
NCC	25,55	26,13	34,79	38,32	23,98	61,68
Lojistik Regresyon	59,34	58,55	54,55	43,15	45,31	56,85
Lasso Lojistik Regresyon	58,45	57,84	56,13	39,51	41,17	60,48
ENC (Esnek Ağ Sınıflayıcısı)	59,21	58,44	54,70	42,73	44,80	57,27
SVC	77,04	75,59	67,42	57,85	60,23	42,15
Kuadratik Kernel SVC	67,57	66,77	61,75	48,61	51,04	51,39
RBF Kernel SVC	77,03	75,58	67,40	57,82	60,20	42,24
Linear SVC	57,15	56,87	61,36	35,55	34,82	64,45
Pasif Agresif Sınıflayıcı	49,04	48,73	38,32	33,97	34,13	66,03
Perceptron	41,85	43,84	37,96	36,21	33,50	63,79
Voting Hard Sınıflayıcı	74,73	73,94	65,18	54,42	54,79	45,84
Voting Soft Sınıflayıcı	84,09	83,18	70,80	63,79	65,73	36,21
LSTM	70,52	70,48	52,50	41,85	42,24	58,14



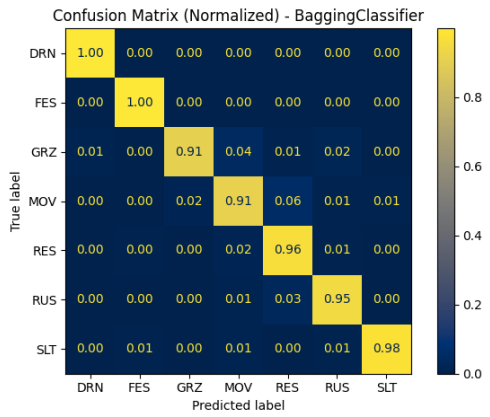
Şekil 3.3. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (a) Linear Ridge (b) LDA (c) QDA (d) SGD Classifier (e) GPC (f) Gaussian Navie Bayes



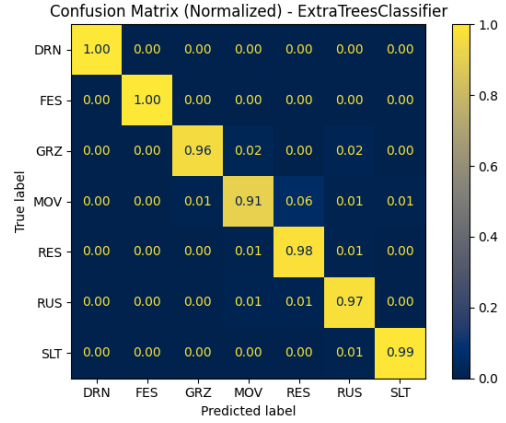
(g)



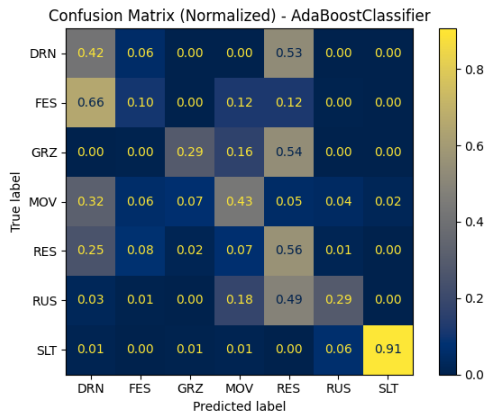
(h)



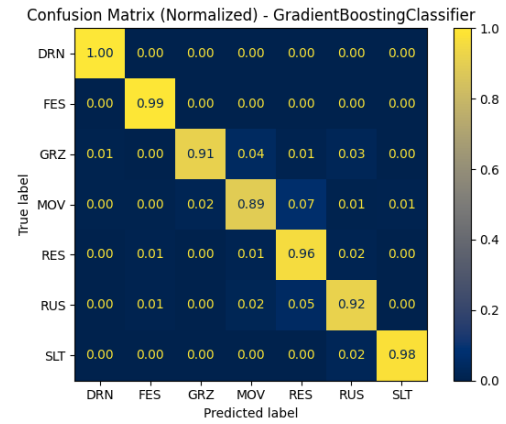
(i)



(i)



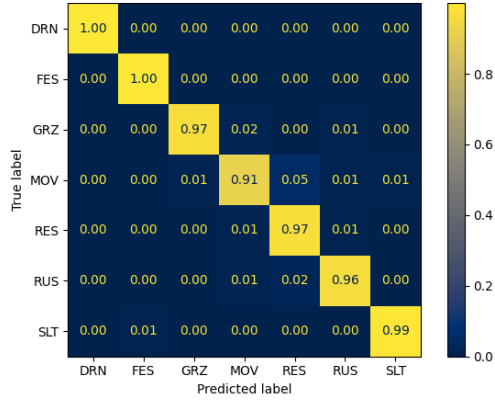
(j)



(k)

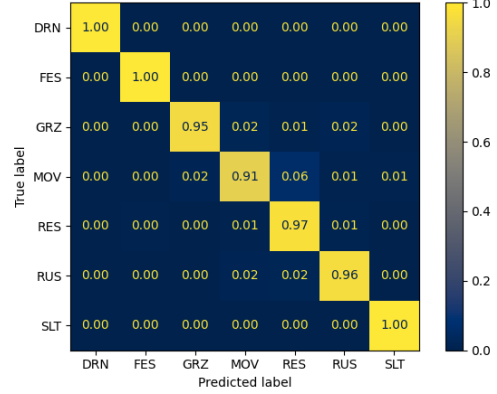
Şekil 3.4. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (g) Bernoulli Navie Bayes (h) Karar Ağacı Sınıflayıcısı (i) Bagging Classifier (i) ET Sınıflayıcısı (j) Ada Boost Classifier (k) Gradyan Artırma Sınıflayıcısı

Confusion Matrix (Normalized) - HistGradientBoostingClassifier



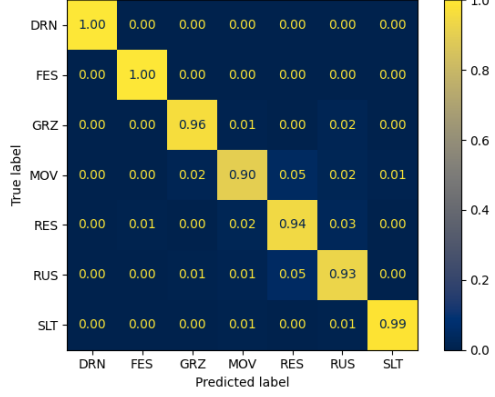
(l)

Confusion Matrix (Normalized) - RandomForestClassifier



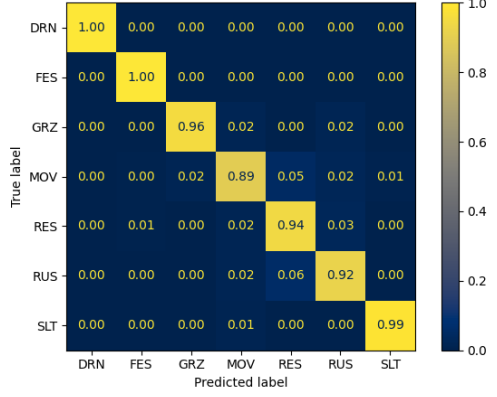
(m)

Confusion Matrix (Normalized) - KNeighborsClassifier



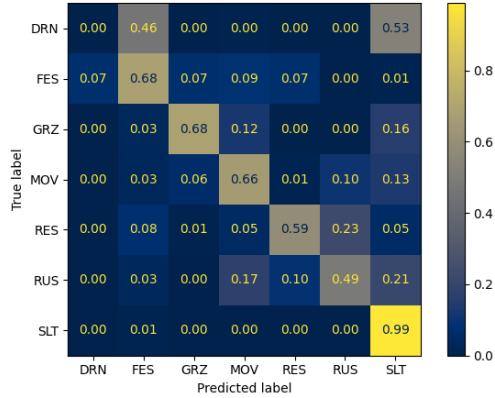
(n)

Confusion Matrix (Normalized) - RadiusNeighborsClassifier



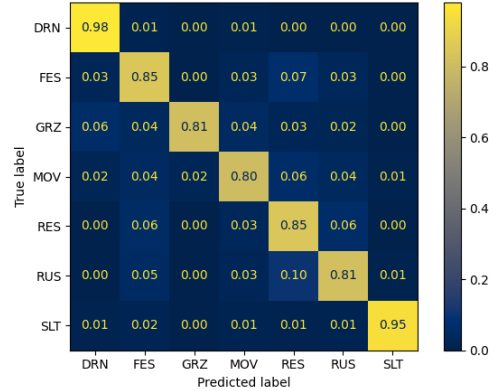
(o)

Confusion Matrix (Normalized) - NearestCentroid



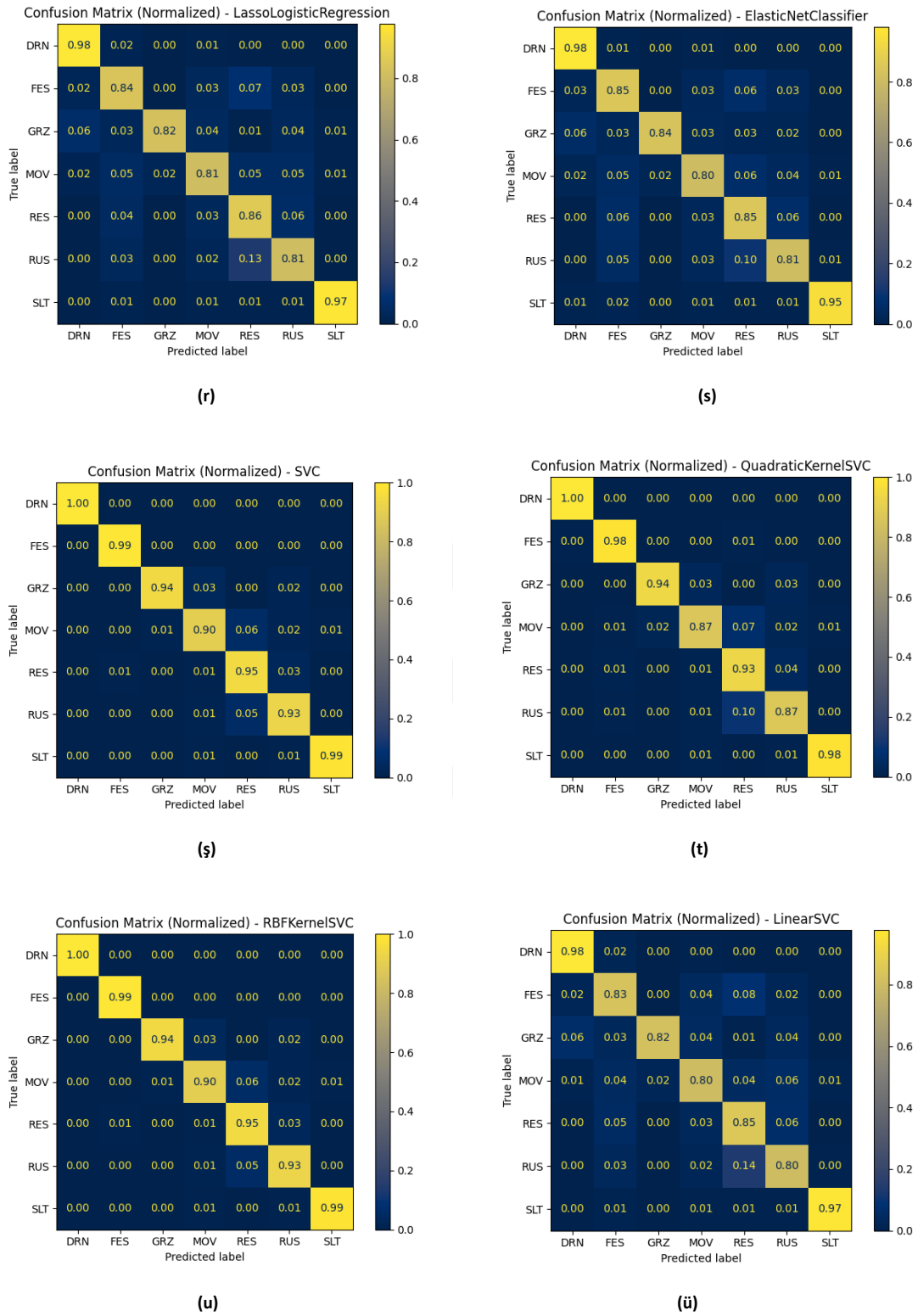
(ö)

Confusion Matrix (Normalized) - LogisticRegression

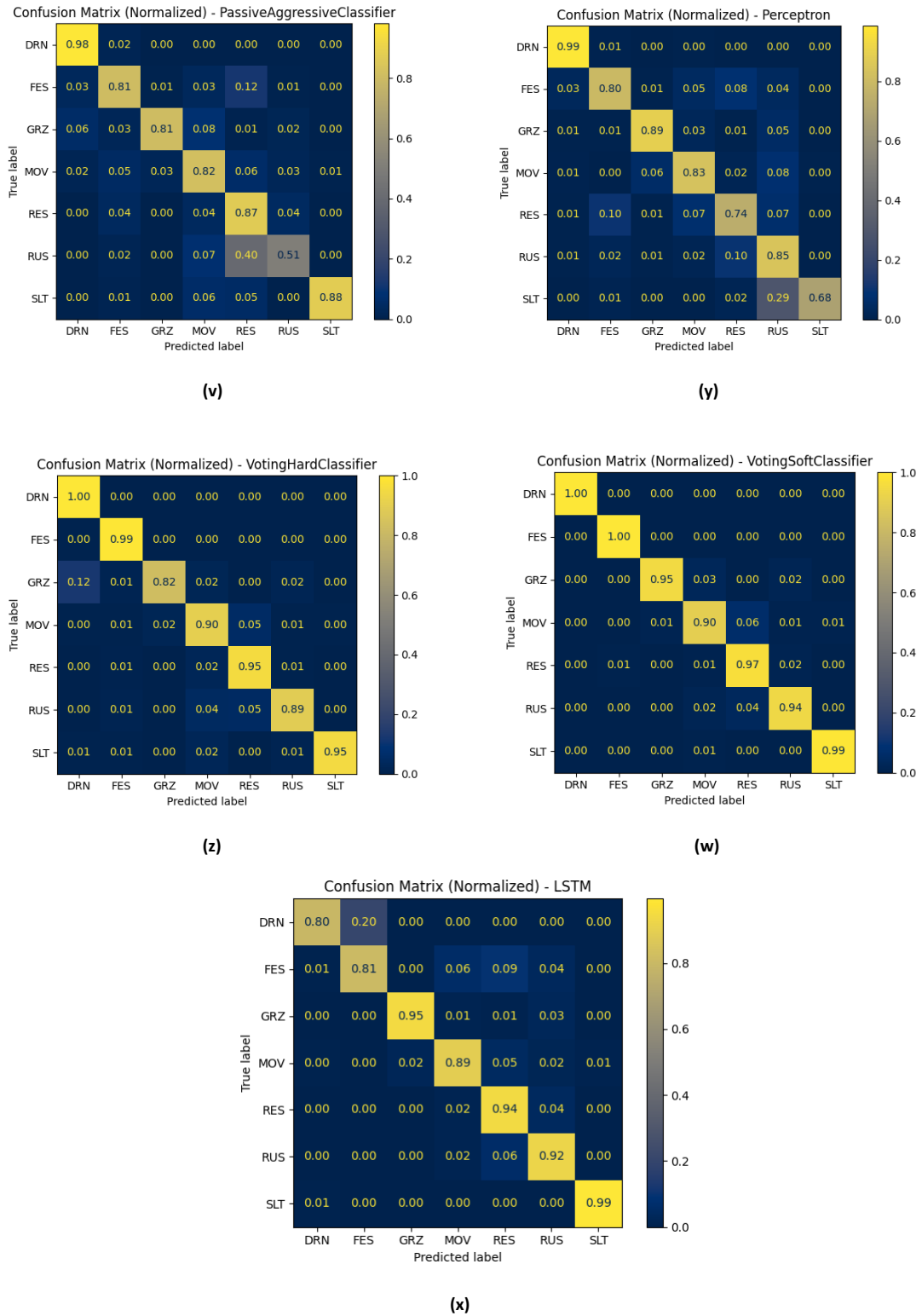


(p)

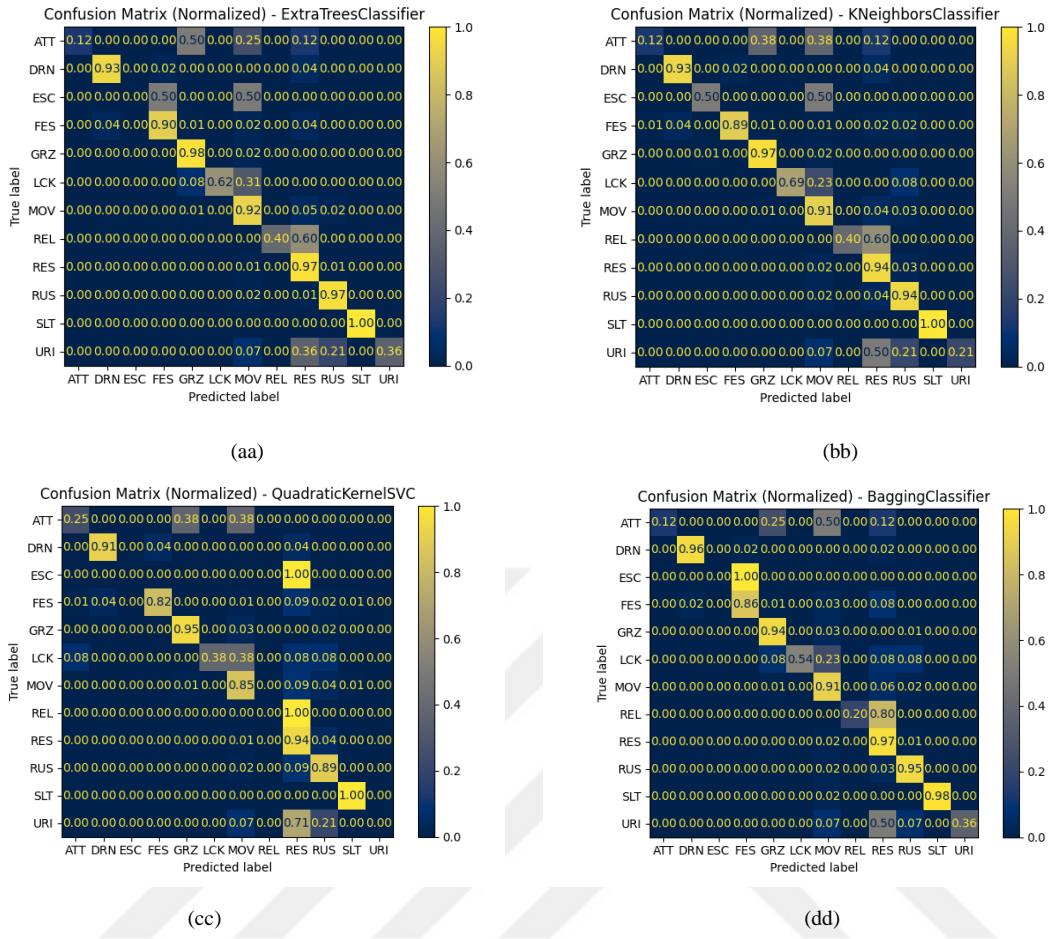
Şekil 3.5. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (l) HGBC (m) Rastgele Orman Sınıflayıcısı (n) KNN Sınıflayıcı (o) RBC (ö) NCC (p) Lojistik Regresyon



Şekil 3.6. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (r) Lasso Lojistik Regresyon (s) ENC (ş) SVC (t) Kuadratik Kernel SVC (u) RBF Kernel SVC (ü) Linear SVC



Şekil 3.7. Kullanılan makine öğrenmesi modelleri için karışıklık matrisleri (v) Pasif Agresif Sınıflayıcı (y) Perceptron (z) Voting Hard Classifier (w) Voting Soft Classifier (x) LSTM



Şekil 3.8. Tüm sınıfların kullanıldığı makine öğrenmesi modelleri için karışıklık matrisleri (aa) ET Sınıflayıcısı (bb) KNN Sınıflayıcısı (cc) Kuadritic Kernel SVC (dd) Bagging Sınıflayıcısı

4. TARTIŞMA

Bu çalışmada kullanılan özellikler, literatürde aynı veri seti ile yapılan çalışmalarla karşılaştırıldığında önemli farklılıklar göstermektedir. Literatürde genellikle zaman alanı ve frekans alanı özellikler tercih edilmiştir. Ancak, bu çalışmada hem zamansal hem de frekans tabanlı analizlerin ötesine geçilerek Yineleme Grafiği tabanlı özellikler, entropi özellikleri (örneğin, Shannon ve SVD Entropi) ve fraktal analiz özellikleri dahil edilmiştir. Özellikle, SVD Entropi ve Shannon Entropi özellikleri, veri setinin karmaşıklığını ve dinamik yapısını daha iyi modellemek için seçilmiştir. Çizelge 3.1.'de görüldüğü gibi SVD entropi ve Shannon Entropi yüksek önem oranı sunmuştur.

Çizelge 4.1.'de literatürde yapılan çalışmaların bazıları verilmiştir. Balasso, Marchesini, Ughelini, Serva ve Andrighetto (2021), süt sığırlarının davranış ve postür tespiti konusunda önemli katkılar sunmaktadır [6]. Çalışmada 32 adet özellik kullanılmış ve XGB, RF, SVM ve KNN gibi modeller test edilmiştir. KNN ile %97,4, RF ile 98,8%, SVM ile %97,6 ve XGB ile %99,2 doğruluk oranları elde ederek bu modeller ile postür tespitin yüksek doğrulukta yapılabildiğini göstermiştir.

Li ve Chai'nin yaptıkları çalışmada [11], bu tez çalışmasında kullanılan veri seti ve toplam 44 özellik kullanılmıştır ve yine ET modeli ile bu tez çalışmasında elde edilene (%98,18) yakın bir doğruluk (%97,2) elde edilmiştir. Çalışmanın detaylarında belirtilen özellikler ve kullanılan yöntemler, modelin başarılı bir şekilde davranışları sınıflandırmasını sağlamıştır.

Martono, Sawado, Nonaka, Terada ve Ohwada'nın (2023) yayınlanan çalışmasında 15 özellik kullanılmıştır [7]. Bu çalışmada farklı pencere boyutlarında 4 farklı model değerlendirilmiştir. RF, XGB, DT ve LR ile modeller eğitilmiş ve Karar Ağacı ile %87 doğruluk oranı elde edilmiştir.

Pavlovic ve diğerlerinin 2022'deki çalışmasında, 42 adet özellik çıkarılmış daha sonra bu özellikler Geriye Dönük Özelliğin Ortadan Kaldırılması (Backward Feature Elimination - BFE) yöntemi kullanılarak 7 özelliğe düşürülmüştür [8]. Model, LDA algoritmasıyla eğitilmiştir. LDA ile %83 doğruluk oranı elde edilmiştir. 7 özellik kullanıldığında işlem süresi 1.83 ± 1.00 ms, tahmin süresi ise 0.05 ± 0.01 ms olarak belirtilmiştir. Sonuç olarak,

çalıřma veri boyutunu azaltarak özellik ıkarmayı optimize etmiř ve dūřuk gū tūketimi ile sınırlı sistemlerde uygulanabilir bir özüm sunmuřtur.

Riaboff ve diđerleri yaptıkları alıřmada 67 farklı özellik ıkarımı yapmıř, ancak yüksek korelasyon gösteren bazı özellikler elenmiřtir [9]. Sonu olarak, 61 özellik analizde kullanılmıřtır. alıřma, sūt sıđırlarının yatarak dinlenme, ayakta dinlenme, yatarak geviř getirme, ayakta geviř getirme, otlama ve yürüme gibi 6 temel davranıřını tahmin etmek iin metodolojik bir ereve geliřtirmeyi hedeflemiřtir. En iyi sonular XGB algoritması ile elde edilmiř ve %98 dođruluk sunmuřtur.

Wu ve diđerleri 2022'deki alıřmalarında LSTM ve Bidirectional LSTM gibi derin öğrenme yöntemleri ile %94,9 dođruluk oranı elde etmiřlerdir [10].

Li ve diđerlerinin 2021 yılına ait alıřmasında kullanılan veri seti ile bu tez alıřmasında kullanılan veri seti aynıdır, CNN tabanlı bir sınıflayıcı ile analiz gerekleřtirilmiřtir ve ayrıca veri artırma yöntemleri kullanılmıřtır [12]. alıřmanın en yüksek dođruluk oranı %94,43 olarak elde edilmiřtir. Bu sonu, veri artırma tekniklerinin sıđır davranıř sınıflandırmasındaki etkinliđini göstermektedir.

Bartels ve diđerlerinin 2022 tarihli alıřmalarındaki veri seti de bu tez alıřmasında kullanılan veri seti aynıdır [13]. alıřmada dūřuk gū tüketimli gömülü sistemler MCU ve FPGA'ler üzerinde uygulanabilir RNN modelleri geliřtirilmiřtir. Bu modelin en yüksek dođruluk oranı %95,7 olarak elde edilmiřtir. Bu alıřma, dūřuk maliyetli hayvan davranıř analizinde hafif ve dūřuk gū tüketimli modellerin etkili bir řekilde uygulanabileceđini göstermektedir.

Russel ve Selvaraj'ın 2024 yılı alıřmalarında biri bu tez alıřmasındaki olmak üzere 3 farklı veri seti kullanılmıř ve ivmeöler verileri üzerinden derin öğrenme tabanlı bir analiz gerekleřtirilmiřtir [52]. Kullanılan model, 23 katmanlı CNN tabanlı bir derin öğrenme modeli olup, veri setleri iin modelin dođruluk oranları sırasıyla %96,72, %87,15 ve %98,7 olarak rapor edilmiřtir. Bu sonular, farklı veri setlerinde modelin genelleme bařarısını ve aynı veri seti üzerindeki performansını göstermektedir.

Çizelge 4.1. Literatürde yapılan çalışmalar

Kaynak No	Özellik Sayısı	Veri Seti	Veri Türü	Sınıflayıcı Sayısı	En İyi Sınıflayıcı	Doğruluk (%)
6	32	12 Kırmızı-İtalyan inek	İvmeölçer (5 Hz)	4	XGB	%99,2
11	44	6 Japon Sığır (Aynı veri seti)	İvmeölçer (25 Hz)	15	ET	%97,2
7	15	9 Japon Sığır	İvmeölçer (14 Hz)	4	DT	%82
8	7	18 Limuzin Melez Sığır	İvmeölçer (10 Hz)	3	LDA	%83
9	61	86 Holstein Sığırı	İvmeölçer (59.5 Hz)	4	XGB	%98
10	Belirtilmemiş	12 Sığır	Ataletsel Ölçüm Ünitesi (10Hz)	2	LSTM	%94,9
12	Belirtilmemiş	6 Japon Sığır (Aynı veri seti)	İvmeölçer (25 Hz)	1	CNN	%94,43
13	3	6 Japon Sığır (Aynı veri seti)	İvmeölçer (25 Hz)	1	RNN	%95,7
52	Belirtilmemiş	3 farklı veri seti (biri bu çalışmadaki ile aynı)	İvmeölçer	1	23 katmanlı CNN	%96,72 %87,15 %98,70
Bu Çalışma	27	6 Japon Sığır	İvmeölçer (25 Hz)	29	ET	%98,19

Bu çalışmada elde edilen sonuçlar, önceki literatürle karşılaştırılarak değerlendirilmiş ve benzerlikler, paralellikler ve farklılıklar tartışılmıştır. Bu çalışmada toplamda 109 adet özellik çıkarılmıştır fakat özellik seçimi ile 27 adet özellik seçilmiş ve modeller üzerinde kullanılmıştır. Literatürde, ilk defa kullanılan Quadratic Kernel SVC ile %95,02 doğruluk oranı elde edilmiştir. Çalışmada kullanılan modellerden ET, %98,18 gibi yüksek bir doğruluk oranıyla en iyi sonucu sağlamıştır. Bu sonuç bu veri setini kullanarak yapılan çalışmaların üzerinde bir sonuçtur ve bunu daha az özellik kullanarak başarmıştır. Ayrıca bu çalışmada toplamda 29 adet sınıflayıcı modeli kullanılmış ve bunlardan 16 tanesi %90,0 üzerinde doğruluk oranı vermiştir. Çalışmada kullanılan modellerin başarısı, veri setindeki önemli özelliklerin doğru bir şekilde çıkarılmasına ve modele entegre edilmesine bağlanabilir. LSTM modeli ile %92,70 test doğruluğu elde edilmiştir. Bu durum, LSTM modelinin daha karmaşık özellik setlerine ihtiyaç duyabileceğini, özellik seçimi için farklı

bir modelin seçilmesi gerektiğini veya hiper parametre optimizasyonunun daha ileri düzeyde yapılması gerektiğini gösterebilir. Ayrıca yine bu çalışmada RP tabanlı toplamda 36 adet özellik kullanılarak 29 adet sınıflayıcı modeli kullanılmış ve bu sınıflayıcılardan da yüksek doğruluk oranları elde edilmiştir. Bu modelde de en yüksek sonuç ET, 85,73% doğruluk oranı, 70,08% Precision ve %63,50 Recall değerleriyle en yüksek performansı göstermiştir. Bu özellikleri kullanarak elde edilen modellerde, kullanılan 29 adet sınıflayıcının 6'sı %80,0'in üzerinde doğruluk oranı yakalamıştır.

Bu sonuçlar, hayvancılıkta modern teknolojilerin entegrasyonunun hayvan refahı ve verimlilik açısından önemini bir kez daha göstermektedir.



5. SONUÇ VE ÖNERİLER

Bu çalışmada, büyükbaş hayvanların davranışlarını izlemek ve sınıflandırmak amacıyla ivmeölçer sensörlerinden elde edilen veriler kullanılarak çeşitli geleneksel sınıflayıcılar ve derin öğrenme modeli uygulanmıştır. Veri setinde yer alan sınıf dengesizlikler, az örnek içeren davranışların özellik çıkarımı işleminden sonra çıkarılmış ve ayrıca SMOTE ile 2 adet azınlık sınıfının artırılması gerçekleştirilmiştir. Bu sayede model performansının artırılması sağlanmıştır. Ayrıca, bu çalışma ile aynı veri setini kullanan önceki çalışmalara göre daha az özellik kullanılarak yüksek doğruluk oranları sağlanmıştır. Performans değerlendirmeleri sonucunda, ET modeli %98,18 ile en yüksek doğruluğa ulaşırken, LSTM modeli zaman serisi verilerindeki uzun vadeli bağımlılıkları öğrenme kapasitesiyle %92,70 doğruluk oranı elde etmiştir.

Elde edilen sonuçlar, literatürde yer alan benzer çalışmalarla uyumluluk göstermektedir. Özellikle ivmeölçerlerin hayvan davranışlarını bireysel ve sürü düzeyinde analiz etme kapasitesine dair mevcut bulgular, bu çalışmada da desteklenmiştir. Bununla birlikte, özellik seçimi yöntemi kullanılarak yüksek doğruluk oranlarının elde edilmesi, çalışmayı aynı veri setini kullanan diğer araştırmalardan farklılaştırmaktadır. Ayrıca bu çalışmada, veri seti üzerinde yineleme grafiği (RP) tabanlı özellikler çıkarılmış ve sadece bu özellikler kullanılarak da 29 adet sınıflayıcı modellenmiştir. Literatürde, bu veri seti üzerinde RP tabanlı özelliklerin çıkarıldığı bir çalışmaya rastlanmamıştır. RP tabanlı özellikler kullanılarak eğitilen modellerde en yüksek doğruluk ET modeli ile elde edilmiştir. ET modeli ile %85,73 oranında doğruluğa ulaşılmıştır. LSTM modeli ile ise %70,57 oranında doğruluk sağlanmıştır.

RP tabanlı özelliklerin tek başına kullanıldığı model, tüm özellikleri içeren modele kıyasla daha düşük performans göstermiştir. Bu çalışmada pencere süresi 1 saniye ile sınırlandırılmıştır ve RP bazlı analizlerin verimli şekilde çalışması için pencere büyüklüğü yeterli olmamış olabilir. Özellikle hareketlerin belirgin bir şekilde tekrarlandığı örüntülerin RP tabanlı analizle daha iyi tespit edilebilmesi için pencere süresinin artırılması önerilebilir.

27 adet özellik kullanılarak elde edilen %98,18 oranındaki doğruluk literatürdeki diğer çalışmalar ile karşılaştırılabilecek bir düzeydedir ve bu oran aynı zamanda kullanılan veri

setinin sınıflandırma modelleri açısından zengin bilgi taşıdığını göstermektedir. Bu sonuç, özellik seçimi ve modelleme süreçlerinde yapılan optimizasyonların başarısını ortaya koymaktadır.

Gelecekte yapılacak çalışmalarda büyükbaş hayvanların davranış kategorilerini içeren daha geniş veri setleriyle yapılacak çalışmalar, modelin genelleme kapasitesini artırabilir. İvmeölçer tabanlı sistemlerin gerçek zamanlı izleme ve müdahale mekanizmalarına entegrasyonu, hayvancılık sektöründe verimlilik ve sağlık yönetimi açısından önemli ilerlemeler sağlayabilir. Hiper parametre optimizasyonu, modelin eğitim süresini önemli ölçüde uzattığı için, özellikle zaman kısıtlamalarının bulunduğu durumlarda kullanımı önerilmemektedir. Son olarak, biyolojik, çevresel ve ekonomik faktörlerin bir araya getirildiği çapraz disiplin çalışmaları, hayvan davranışlarının izlenmesine yönelik daha kapsamlı modellerin geliştirilmesine katkı sunabilir.

Sonuç olarak, bu çalışmanın bulguları, hayvan sağlığı ve refahının artırılması ile tarım ve hayvancılık sektöründeki teknolojik dönüşümde önemli bir rol oynamaktadır. İvmeölçer tabanlı sistemlerin ve makine öğrenimi modellerinin kullanımına dair bu çalışma, gelecekte yapılacak araştırmalar için güçlü bir temel sunmakta ve sürdürülebilir hayvancılık hedeflerine ulaşılmasına katkı sağlamaktadır.

KAYNAKLAR

1. Berckmans, D. (2017). General introduction to precision livestock farming. *Animal Frontiers*, 7(1), 6–11.
2. Helwatkar, A., Riordan, D., ve Walsh, J. (2014). Sensor technology for animal health monitoring. *International Journal on Smart Sensing and Intelligent Systems*, 7(5), 1–6.
3. Sveberg, G., Refsdal, A. O., Erhard, H. W., Kommisrud, E., Aldrin, M., Tvette, I. F., Buckley, F., Waldmann, A., ve Ropstad, E. (2011). Behavior of lactating Holstein-Friesian cows during spontaneous cycles of estrus. *Journal of Dairy Science*, 94(3), 1289–1301.
4. Yaylak, E. (2008). Süt sığırlarında topallık ve topallığın bazı özelliklere etkisi. *Hayvansal Üretim*, 49(1), 1-7.
5. von Keyserlingk, M. A. G., Weary, D. M., Keyserlingk, V., ve Weary, M. A. G. (2010). Feeding behaviour of dairy cattle: Measures and applications. *Canadian Journal of Animal Science*, 90(3), 303–309.
6. Balasso, P., Marchesini, G., Ughelini, N., Serva, L., ve Andrighetto, I. (2021). Machine learning to detect posture and behavior in dairy cows: Information from an accelerometer on the animal's left flank. *Animals*, 11(10), 1-13.
7. Martono, N. P., Sawado, R., Nonaka, I., Terada, F., ve Ohwada, H. (2023, 15-16 Kasım). *Automated cattle behavior classification using wearable sensors and machine learning approach*. 19th Principle and Practice of Data and Knowledge Acquisition Workshop (PKAW 2023)'da sunuldu, Jakarta, 58–69.
8. Pavlovic, D., Czerkawski, M., Davison, C., Marko, O., Michie, C., Atkinson, R., Crnojevic, V., Andonovic, I., Rajovic, V., Kvascev, G., ve Tachtatzis, C. (2022). Behavioural Classification of Cattle Using Neck-Mounted Accelerometer-Equipped Collars. *Sensors*, 22(6), 1-18.
9. Riaboff, L., Poggi, S., Madouasse, A., Couvreur, S., Aubin, S., Bédère, N., Goumand, E., Chauvin, A., ve Plantier, G. (2020). Development of a methodological framework for a robust prediction of the main behaviours of dairy cows using a combination of machine learning algorithms on accelerometer data. *Computers and Electronics in Agriculture*, 169(1), 1-16.
10. Wu, Y., Liu, M., Peng, Z., Liu, M., Wang, M., ve Peng, Y. (2022). Recognising cattle behaviour with deep residual bidirectional LSTM model using a wearable movement monitoring collar. *Agriculture (Switzerland)*, 12(8), 1-13.
11. Li, G., ve Chai, L. (2023). AnimalAccML: An open-source graphical user interface for automated behavior analytics of individual animals using triaxial accelerometers and machine learning. *Computers and Electronics in Agriculture*, 209(Part C), 1-13.

12. Li, C., Tokgoz, K., Fukawa, M., Bartels, J., Ohashi, T., Takeda, K. I., ve Ito, H. (2021). Data augmentation for inertial sensor data in CNNs for cattle behavior classification. *IEEE Sensors Letters*, 5(11), 1-4.
13. Bartels, J., Tokgoz, K. K., Sihan, A., Fukawa, M., Otsubo, S., Li, C., Rachi, I., Takeda, K. I., Minati, L., ve Ito, H. (2022). TinyCowNet: Memory- and power-minimized RNNs implementable on tiny edge devices for lifelong cow behavior distribution estimation. *IEEE Access*, 10(1), 32706–32727.
14. Arun Faisal, I., Waluyo Purboyo, T., ve Siswo Raharjo Ansori, A. (2019). A review of accelerometer sensor and gyroscope sensor in IMU sensors on motion capture. *Journal of Engineering and Applied Sciences*, 15(3), 826–829.
15. Grahn, E. (2017). *Evaluation of MEMS accelerometer and gyroscope for orientation tracking nutrunner functionality*. Master's thesis, KTH Royal Institute of Technology, Stockholm.
16. İnternet: Ito, H., Takeda, K., Tokgoz, K. K., Minati, L., Fukawa, M., Li, C., Bartels, J., ve Rachi, I. (2022). Japanese black beef cow behavior classification dataset. Web: <https://zenodo.org/record/5848608>. Son Erişim Tarihi: 10 Ocak 2025.
17. Polatgil, M. (2023). Veri ölçekleme ve eksik veri tamamlama yöntemlerinin makine öğrenmesi yöntemlerinin başarısına etkisinin incelenmesi. *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 11(1), 78–88.
18. Gopal Krishna Patro, S., ve Kumar Sahu, K. (2015). Normalization: A preprocessing stage. *International Advanced Research Journal in Science, Engineering and Technology*, 2(3), 20-22.
19. Bagnall, A., Lines, J., Bostrom, A., Large, J., ve Keogh, E. (2017). The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3), 606-660.
20. Krishnan, N. C., ve Cook, D. J. (2014). Activity recognition on streaming sensor data. *Pervasive and Mobile Computing*, 10(Part B), 138–154.
21. Banos, O., Galvez, J. M., Damas, M., Pomares, H., ve Rojas, I. (2014). Window size impact in human activity recognition. *Sensors (Switzerland)*, 14(4), 6474–6499.
22. McKinney, W. (2010, 1-3 Temmuz). *Data structures for statistical computing in python*. Proceedings of the 9th Python in Science Conference (SciPy 2010)'da sunuldu, Austin, 56-61.
23. Zhang, Z. (2016). Missing data imputation: Focusing on single imputation. *Annals of Translational Medicine*, 4(1), 1-7.
24. Bahadır, O., ve Türkmençalıkoğlu, H. (2022). Bilgi kuramında shannon entropisi ve uygulamaları. *European Journal of Science and Technology*, 32(1), 491-497.

25. Higuchi, T. (1988). Approach to an irregular time series on the basis of the fractal theory. *Physica D: Nonlinear Phenomena*, 31(2), 277–283.
26. Wanliss, J. A., ve Wanliss, G. E. (2022). Efficient calculation of fractal properties via the Higuchi method. *Nonlinear Dynamic*, 109(4), 2893-2904.
27. Katz, M. J. (1988). Fractals and the analysis of waveforms. *Computers in Biology and Medicine*, 18(3), 145–156.
28. Bandt, C., ve Pompe, B. (2002). Permutation entropy: A natural complexity measure for time series. *Physical Review Letters*, 88(17), 1-5.
29. Pan, Y. N., Chen, J., ve Li, X. L. (2009). Spectral entropy: A complementary index for rolling element bearing performance degradation assessment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(5), 1223–1231.
30. Raubitzek, S., ve Neubauer, T. (2021). Combining measures of signal complexity and machine learning for time series analysis: A review. *Entropy*, 23(12), 1–26.
31. Pincus, S. M. (1991). Approximate entropy as a measure of system complexity. *Proceedings of the National Academy of Sciences of the United States of America*, 88(6), 2297–2301.
32. Delgado-Bonal, A., ve Marshak, A. (2019). Approximate entropy and sample entropy: A comprehensive tutorial. *Entropy*, 21(6), 1-37.
33. Pütün, A., ve Yılmaz, D. (2024, 21-22 Eylül). *Classification of cattle behavior leveraging accelerometer data and machine learning*. 8th International Artificial Intelligence and Data Processing Symposium’da sunuldu, Malatya.
34. Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32.
35. Geurts, P., Ernst, D., ve Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, 63(1), 3–42.
36. Hosmer, D. W., Lemeshow, S., ve Sturdivant, R. X. (2013). *Applied logistic regression* (3rd ed.). Hoboken, NJ: John Wiley and Sons, 510.
37. Pereira, J. M., Basto, M., ve Silva, A. F. da. (2016). The logistic lasso and ridge regression in predicting corporate failure. *Procedia Economics and Finance*, 39(1), 634–641.
38. Huang, Y., Schell, C., Huber, T. B., Şimşek, A. N., Hersch, N., Merkel, R., Gompper, G., ve Sabass, B. (2019). Traction force microscopy with optimized regularization and automated Bayesian parameter selection for comparing cells. *Scientific Reports*, 9(1), 1-10.

39. Kudari, J. M., Varsha, V., Monica, B. G., ve Archana, R. (2020). Fake news detection using passive aggressive and TF-IDF vectorizer. *International Research Journal of Engineering and Technology*, 7(9), 1601–1603.
40. Freund, Y., ve Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3), 277-296.
41. Dietterich, T. G. (2000). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2), 139-157.
42. Hastie, T., Rosset, S., Zhu, J., ve Zou, H. (2009). Multi-class AdaBoost. *Statistics and Its Interface*, 2(3), 349–360.
43. Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5), 1189–1232.
44. İnternet: Scikit-learn. sklearn. ensemble.VotingClassifier. Web: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.VotingClassifier.html>. Erişim tarihi: 7 Ocak 2025.
45. Cortes, C., Vapnik, V., ve Saitta, L. (1995). Support-Vector Networks. *Machine Learning*, 20(3), 273–297.
46. John, G. H., ve Langley, P. (1995, 18-20 Ağustos). *Estimating continuous distributions in Bayesian classifiers*. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence'da sunuldu, Montreal, 338–345.
47. McCallum, A., ve Nigam, K. (1998, 26-27 Temmuz). *A Comparison of Event Models for Naive Bayes Text Classification*. Proceedings of the AAAI-98 Workshop on Learning for Text Categorization'da sunuldu, Madison.
48. İnternet: Scikit-learn. KNeighborsClassifier: k-Nearest Neighbors Classifier. URL: <https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>. Erişim tarihi: 7 Ocak 2025.
49. İnternet: Scikit-learn. RadiusNeighborsClassifier: Radius-Based Nearest Neighbors Classifier. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.RadiusNeighborsClassifier.html>. Erişim tarihi: 7 Ocak 2025.
50. İnternet: Scikit-learn. NearestCentroid: Centroid-Based Classifier. URL: <https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid.html>. Erişim tarihi: 7 Ocak 2025.
51. Hochreiter, S., and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780.

52. Russel, A., ve Selvaraj, R. (2024). Deep Learning for Animal Behavior Classification: A Multi-Dataset Study Using Triaxial Accelerometer Data. *Journal of Animal Behavior Science*, 45(3), 102-118.





EKLER

EK-1. Optimizasyonda kullanılan parametreler

Çizelge 1.1. Grid Search parametreleri

Sınıflayıcı	Parametre 1	Parametre 2	Parametre 3	Parametre 4	Parametre 5
AdaBoost	algorithm SAMME	learning_rate 1	n_estimators 50	-	-
Bagging	bootstrap False	bootstrap_features: False	max_features 0.5	max_samples 1	n_estimators 300
Bernoulli NB	alpha 0.1	binarize 0.5	fit_prior True	-	-
Karar Ağacı	criterion entropy	max_depth 50	max_features nan	min_samples_leaf : 1	min_samples_split: 2
ENC (Esnek Ağ Sınıflayıcısı)	C 1	l1_ratio 0.7	-	-	-
ET	max_depth 30	min_samples_leaf: 1	min_samples_split: 2	n_estimators 300	-
Gaussian NB	var_smoothing: 1e-09	-	-	-	-
GPC	kernel RBF	n_restarts_optimizer: 0	-	-	-
Gradyan Artırma Sınıflayıcısı	learning_rate 0.1	max_depth 5	min_samples_split: 2	n_estimators 100	subsample 1
HGBC	learning_rate 0.1	max_depth 20	max_iter 200	-	-
KNN Sınıflayıcı	n_neighbors 3	p 1	weights distance	-	-
Lasso Lojistik Regresyon	C 10	-	-	-	-
Doğrusal Diskriminant Analizi	shrinkage nan	solver svd	-	-	-
Linear Ridge	alpha 10	solver auto	-	-	-
Linear SVC	C 1	loss hinge	max_iter 2000	-	-
Lojistik Regresyon	C 1	max_iter 200	penalty l1	solver saga	-
LSTM	batch_size 64	dropout_rate 0.2	learning_rate 0.01	units 100	-
NCC	metric euclidean	shrink_threshold nan	-	-	-

EK-1. (devam) Optimizasyonda kullanılan parametreler

Çizelge 1.1. (devam) Grid Search parametreleri

Pasif Agresif Sınıflayıcı	C 0.1	loss hinge	-	-	-
Perceptron	alpha 0.0001	penalty l1	-	-	-
Kuadritik Diskriminant Analizi	reg_param 0.01	store_covariance: 1	tol 0.0001	-	-
Kuadritik Kernel SVC	C 10	coef0 1	degree 2	gamma auto	max_iter 2000
RBC	radius 10	weights distance	-	-	-
Rastgele Orman Sınıflayıcı	max_depth 20	n_estimators 200	-	-	-
RBF Kernel SVC	C 10	gamma scale	max_iter 2000	-	-
SGD Sınıflayıcısı	alpha 0.0001	penalty l2	-	-	-
SVC	C 10	kernel rbf	-	-	-
Voting Hard Sınıflayıcı	voting 'dt'	estimators DecisionTreeC lassifier max_depth:10	n_jobs -1	-	-
Voting Soft Sınıflayıcı	Voting soft	Estimators Rf RandomForest Classifier n_estimators: 100	Weights svc SVC probability: True	-	-



Gazili olmak ayrıcalıktır