

**MAKİNE ÖĞRENMESİ YÖNTEMLERİ İLE İMHA
DEĞERLENDİRME**

**KILL ASSESSMENT USING MACHINE LEARNING
METHODS**

RAHİME SEVİNÇ GÖKDUMAN

PROF. DR. CENK TOKER

Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Elektrik ve Elektronik Mühendisliği Anabilim Dalı için Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.

2022

ÖZET

MAKİNA ÖĞRENMESİ YÖNTEMLERİ İLE İMHA DEĞERLENDİRME

Rahime Sevinç GÖKDUMAN

Yüksek Lisans, ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ Bölümü

Tez Danışmanı: Prof. Dr. Cenk TOKER

Ocak 2022, 119 sayfa

İmha değerlendirme başlatılan bir angajman sonucunda ilgili tehdidin vurulup-vurulmadığı kararının verilebilmesi, başlatılan angajmanın başarılı olup olmadığının değerlendirilmesidir. Tehdit için belirtilen imha edildi - edilmedi bilgisi hem kullanıcı dikkatinin hem maliyetlerin hem de angajmanın doğru yönetilmesi için kritik önem ifade etmektedir. Bu tespit artık insan gözüyle yapılabilecek menzillerden ve büyüklüklerden çok uzaktır. İmha değerlendirmesinde, makine öğrenmesi ve derin öğrenme tekniklerinin kullanılmasıyla karar alma sürecinde kararlarının insan müdahalesinden bağımsız olarak verilmesi ve böylece hataların ve kaynak israflarının azaltılması hedeflenmektedir.

İmha değerlendirmesinin makine öğrenmesi ve derin öğrenme algoritmaları ile yapılmasına dair gerçekleştirilen bu çalışmada, gerçek dünyaya ait radar verilerine sahip olunmaması nedeniyle radar sinyallerini sentetik veri olarak üreten ve imha-ıskalama gibi fiziksel olayları simüle eden bir simülatör geliştirilmiştir. Ham radar verilerinin gürültüye ve aşırı uyumlamaya açık olması, eğitim sürecinde donanımın yetersiz kalması ve eğitim süresinin uzun sürmesi gibi dezavantajlar nedeniyle sınıflandırma algoritmalarına girdi olarak kullanımı uygun bulunmamıştır. Bu nedenle sentetik ham radar sinyallerinden öznitelik çıkarım işlemleri

yapılarak algoritmalara bu öznitelikler girdi olarak sağlanmıştır. Bu öznitelik tipleri noksan-sız öznitelik seti, noksan öznitelik seti ve belirsizlik analizi öznitelik seti [1] olmak üzere üçe ayrılmaktadır.

İmha değerlendirme işlemi Vurulma Öncesi, İskalama ve İmha olmak üzere farklı durumları belirten bu üç sınıfa ayrılarak ifade edilmiştir. İmha durumu doğası gereği anlık ya da milisaniyeler mertebesinde gerçekleşen bir olaydır. Bu nedenle sınıflara ait veri setindeki denge gözetildiğinde vurulma öncesi anına ya da ıskalama anına ait veri yoğunluğu imha anına ait veri ile kıyaslandığında çok daha fazla olmaktadır. Bu sebeple sınıf dengesini sağlayan çeşitli yöntemler araştırılarak sınıf dengesinin sağlandığı öznitelik setleri ile eğitim işlemi gerçekleştirilmiştir. Sınıf dengesini sağlamak için SMOTE, ROS, RUS, Yakındakini Atma, SMOTE-ENN, SMOTE-Tomek yöntemleri kullanılmıştır.

Dengesiz ve dengeli hale getirilen veri setleri sırası ile girdi olarak kullanılarak imha değerlendirme algoritmaları test edilmiştir. Makine öğrenmesi algoritmalarından ada artırım, gradyan artırım, karar ağacı, rastgele orman, destek vektör makineleri, çok katmanlı algılayıcı, derin öğrenme algoritmalarından ise sinir ağları ve LSTM kullanılarak doğru bir imha kararının verilmesinde performans ve doğruluk açısından hangi yöntemin daha doğru ve en iyi çalıştığı kıyaslanarak saptanmıştır.

Performans analizlerine bakıldığında üç ayrı öznitelik seti için de ROS uygulanmış öznitelik setinin girdi olduğu rastgele orman algoritmasının %99 üzerinde doğruluk oranı ile en iyi sınıflandırmayı gerçekleştirdiği sonucuna varılmıştır. Sınıf dengesinin sağlanıyor olmasının tüm sınıflandırma algoritmalarında performansı artıran bir unsur olduğu görülmüştür. Bu tez sonucunda yapay zekâ algoritmaları ile radar sinyalleri üzerinden çıkarılan öznitelikler kullanılarak başarılı bir şekilde imha değerlendirme yapılabileceği sonucuna ulaşılmıştır.

Anahtar Kelimeler: Yapay Zekâ, Derin öğrenme, Makine Öğrenmesi Algoritmaları, Vuruldu - Vurulmadı Kararı, İmha Değerlendirme, Rastgele Orman, LSTM, Karar Ağacı, Sinir Ağları, SVM, ROS, SMOTE

ABSTRACT

KILL ASSESSMENT USING MACHINE LEARNING METHODS

Rahime Sevinç GÖKDUMAN

**Master of Science, Department of ELECTRICAL AND ELECTRONICS
ENGINEERING**

Supervisor: Prof. Dr. Cenk TOKER

January 2022, 119 pages

Kill assessment is the ability to decide whether the relevant threat has been hit or not as a result of an initiated engagement, and to evaluate whether the initiated engagement has been successful or not. The destroyed-not destroyed information specified for the threat is critical for the proper management of user attention, costs and engagement. This detection is so far from the ranges and magnitudes that can be observed by the human eye today. In the kill assessment, it is aimed to make decisions independently of the human figure in the decision-making process by using machine learning and deep learning techniques. Thus, the assessment reduces errors and waste of resources.

In this study, which was carried out on the assessment of kill using machine learning and deep learning algorithms, a simulator was developed that generates radar signals as synthetic data and simulates physical events such as kill - miss, due to the lack of real-world radar data. The use of raw radar data as an input to classification algorithms has not been found appropriate due to disadvantages such as being susceptible to noise and overfitting, insufficient equipment in the training process and long training time. For this reason, these features are provided as input to the algorithms by performing feature extraction processes. These feature types are divided into three as the perfect feature set, unperfect feature set and the

ambiguity graphs feature set [1].

The kill assessment process is expressed by dividing these three classes, which indicate different situations: Pre Hit, Miss and Kill. Kill is an event that takes place in the order of instantaneous or milliseconds by its nature. For this reason, when the balance in the data set of the classes is supervised, the data density of the moment before the hit or the moment of the miss is much higher when compared to the data density of the moment of kill. For this reason, various methods that provide class balance were investigated and the training process was carried out with feature sets that provided class balance. SMOTE, ROS, RUS, Near-Miss, SMOTE-ENN, SMOTE-Tomek methods were used to provide class balance.

Kill assessment algorithms were tested by using unbalanced and balanced data sets as inputs, respectively. By comparing which method is more accurate and best in terms of performance and accuracy in making a correct kill decision by using adaboost, gradientboosting, decision tree, random forest, support vector machines, multi-layer perceptron from machine learning algorithms and neural networks and LSTM from deep learning algorithms detected.

When algorithm performances are analyzed, it was concluded that the Random Forest algorithm performed the most successful classification with a rate of 99% in all three ROS applied data sets. It has been seen that ensuring class balance is a factor that increases performance in all classification algorithms. As a result of this thesis, it has been concluded that a successful kill assessment can be made by using the features extracted from the radar signals with artificial intelligence algorithms.

Keywords: Artificial Intelligence, Deep Learning, Machine Learning Algorithms, Hit/Miss Decision, Kill Assessment, Random Forest, LSTM, Decision Tree, Neural Network, SVM, ROS, SMOTE

TEŐEKKÜR

İçinde bulunduđum bu tez sürecinde yönlendirmeleri ile beni girdiđim tüm çıkmazlardan kurtaran, yaptıđım işe saygı duymamı sađlayan, yaptıđı girdi ve yorumlarla bir tez yazıyor olmanın dıőında bana akademik bir bakıő aşıısı kazandıran, deđerli zamanımı bizden esirgemeyen her zaman yanımda olan saygıdeđer danıőmanım Prof. Dr. Cenk TOKER'e sonsuz teőekkürlerimi sunarım.

Birlikte bu yola baőladıđım her adımda birlikte karar ve emek vererek ortaya bir alıőma koyduđum hem iş hem okul hem tez arkadaőım olan Nihan UZUN'a sürecin getirdiđi çıkmazlarda yaőadıđımız zorlukları motivasyon kayıplarını birlikte aőtıđımız için teőekkür ederim.

İő hayatım boyunca bana öğrettikleri kodlama bakıő aşıısı sayesinde tez sürecinde katkılarını sürekli hissettiđim ekip liderim Soner INAR'a, mentorum Burcu ALIK'a ve tüm eğitim hayatım boyunca bende emeđi olan tüm hocalarıma teőekkür ederim.

İmkânlarının tüm sınırlarını zorlayarak beni bu günlere getiren, sonsuz bir güven ve sevgi ile bana destek veren sevgili anneme, babama, kardeőime, tezimi anısına ithaf ettiđim canım dedem Akif GÖREN'e, her zaman yanımda olan ve olacađını bildiđim her konuda olduđu gibi bu süreçte de desteđini hissettiđim sevgili eőim Mert GÖKDUMAN'a sonsuz sevgi ve teőekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET	1
ABSTRACT	3
TEŞEKKÜR	5
İÇİNDEKİLER	8
ÇİZELGELER	11
ŞEKİLLER	14
KISALTMALAR	15
1 GİRİŞ	1
2 LİTERATÜRDE İMHA DEĞERLENDİRME VE YAPAY ZEKÂ ALGORİTMA- LARI	5
2.1 İmha Değerlendirme	5
2.2 Yapay Zekâ Algoritmaları İle Sınıflandırma	9
2.2.1 Makine Öğrenmesi Algoritmaları	10
2.2.1.1 Karar Ağacı Algoritması	11
2.2.1.1.1 Sınıflandırma ve Regresyon Ağacı (CART)	12
2.2.1.2 Kolektif Öğrenme Algoritmaları	13
2.2.1.2.1 Ada Artırım Algoritması	13
2.2.1.2.2 Gradyan Artırım Algoritması	15
2.2.1.3 Rastgele Orman Algoritması	16
2.2.1.4 Destek Vektör Makineleri Algoritması	17
2.2.1.4.1 Çekirdek Fonksiyonları	18
2.2.1.4.2 Çok sınıflı sınıflandırma	19
2.2.2 Sinir Ağları Algoritmaları	19
2.2.2.1 Algılayıcı	20
2.2.2.2 Çok Katmanlı Algılayıcı	21
2.2.2.3 Derin Öğrenme Algoritmaları	22
2.2.2.3.1 Yinelenen Sinir Ağları	22
2.2.2.3.2 Uzun Kısa Süreli Bellek Algoritması	23

2.2.3	Öznitelik Çıkarımı	25
2.2.4	Sınıflandırma Ön İşlemleri	26
2.2.4.1	Veri Normalizasyonu	27
2.2.4.2	Eğitim - Doğrulama - Test Verisi	27
2.2.4.3	Hiperparametre Eniyilemesi	29
2.2.4.3.1	Izgara Arama Yöntemi	29
2.2.4.3.2	Çapraz Doğrulama	30
2.2.5	Sınıf Dengesi Sağlama Yöntemleri	30
2.2.5.1	Sentetik Azınlık Aşırı Örnekleme Yöntemi	32
2.2.5.2	Rastgele Aşırı Örnekleme Yöntemi	33
2.2.5.3	Rastgele Alt Örnekleme Yöntemi	34
2.2.5.4	Sentetik Azınlık Aşırı Örnekleme Ve Düzenlenmiş En Ya- kın Komşu Yöntemi	36
2.2.5.5	Sentetik Azınlık Aşırı Örnekleme Ve Tomek Örnekleme Yöntemi	37
2.2.5.6	Yakındakini Atma Yöntemi	38
2.2.6	Sınıflandırma Performans Analiz Metrikleri	40
3	İMHA DEĞERLENDİRME ALGORİTMASI METODOLOJİSİ	43
3.1	Fiziksel Simülatör Ve Sentetik Radar Sinyali Üretimi	43
3.1.1	İmha Senaryosu	44
3.1.2	İskalama Senaryosu	46
3.1.3	Tehdit	46
3.1.4	Mühimmat	47
3.1.5	Sentetik Radar Sinyalinin Oluşturulması	48
3.2	Öznitelik Setlerinin Oluşturulması	50
3.2.1	Sentetik Radar Sinyal Veri Seti	52
3.2.2	Noksansız Öznitelik Seti	53
3.2.3	Noksan Öznitelik Seti	57
3.2.4	Belirsizlik Analizi Öznitelik Seti	61
3.3	Verilerin Etiketlenmesi	64
3.3.1	Vurulma Öncesi Durum	64
3.3.2	İskalama Durumu	65

3.3.3	İmha Durumu	65
3.4	İmha Deęerlendirme Sınıflandırma Yöntemleri	68
3.4.1	Algoritmalarda Kullanılan Parametreler	69
4	İMHA DEęERLENDİRME ALGORİTMALARININ PERFORMANS ANALİZ- LERİ	73
4.1	Sentetik Radar Sinyal Ham Veri Seti Performansı	73
4.2	Noksansız Öznitelik Seti Performansı	75
4.3	Noksan Öznitelik Seti Performansı	77
4.4	Belirsizlik Analizi Öznitelik Seti Performansı	82
5	SONUÇLAR	86
6	EKLER	96
6.1	EK 1 - Performans Çizelgeleri	96
6.1.1	Noksansız Öznitelik Seti Performans Çizelgeleri	96
6.1.2	Noksan Öznitelik Seti Performans Çizelgeleri	104
6.1.3	Belirsizlik Analizi Öznitelik Seti Çizelgeleri	112

ÇİZELGELER

Çizelge 3.2.1	Öznitelik Çıkarımı ve Etiketleme Dosyası Örneği	51
Çizelge 4.1.1	Sentetik Ham Radar Sinyal Veri Seti Performans Değerlendirme Çizelgesi	74
Çizelge 4.2.1	Dengesiz Noksansız Öznitelik Seti ile Rastgele Orman Algoritması Performans Analizi	75
Çizelge 4.2.2	ROS Uygulanmış Noksansız Öznitelik Seti Rastgele Orman Algoritması Performans Analizi	76
Çizelge 4.3.1	Dengesiz Noksan Öznitelik Seti Rastgele Orman Algoritması Performans Analizi	78
Çizelge 4.3.2	ROS Uygulanmış Noksan Öznitelik Seti Rastgele Orman Algoritması Performans Analizi	79
Çizelge 4.4.1	Dengesiz Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Performans Analizi	83
Çizelge 4.4.2	ROS Uygulanmış Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Performans Analizi	84
Çizelge 6.1.1	Noksansız Öznitelik Seti Ada Artırım Algoritması ile Sınıflandırma Performans Analizi	96
Çizelge 6.1.2	Noksansız Öznitelik Seti Karar Ağacı Algoritması ile Sınıflandırma Performans Analizi	97
Çizelge 6.1.3	Noksansız Öznitelik Seti Gradyan Artırım Algoritması ile Sınıflandırma Performans Analizi	98
Çizelge 6.1.4	Noksansız Öznitelik Seti Rastgele Orman Algoritması ile Sınıflandırma Performans Analizi	99
Çizelge 6.1.5	Noksansız Öznitelik Seti Çok Katmanlı Algılayıcı Algoritması ile Sınıflandırma Performans Analizi	100
Çizelge 6.1.6	Noksansız Öznitelik Seti Destek Vektör Makineleri ile Sınıflandırma Performans Analizi	101

Çizelge 6.1.7	Noksansız Öznitelik Seti Sinir Ağları Algoritması ile Sınıflandırma Performans Analizi	102
Çizelge 6.1.8	Noksansız Öznitelik Seti LSTM Algoritması ile Sınıflandırma Performans Analizi	103
Çizelge 6.1.9	Noksan Öznitelik Seti Ada Artırım Algoritması Performans Analizi	104
Çizelge 6.1.10	Noksan Öznitelik Seti Karar Ağacı Algoritması ile Sınıflandırma Performans Analizi	105
Çizelge 6.1.11	Noksan Öznitelik Seti Gradyan Artırım Algoritması ile Sınıflandırma Performans Analizi	106
Çizelge 6.1.12	Noksan Öznitelik Seti Rastgele Orman Algoritması ile Sınıflandırma Performans Analizi	107
Çizelge 6.1.13	Noksan Öznitelik Seti Çok Katmanlı Algılayıcı Algoritması ile Sınıflandırma Performans Analizi	108
Çizelge 6.1.14	Noksan Öznitelik Seti Destek Vektör Makineleri Algoritması ile Sınıflandırma Performans Analizi	109
Çizelge 6.1.15	Noksan Öznitelik Seti Sinir Ağları Algoritması ile Sınıflandırma Performans Analizi	110
Çizelge 6.1.16	Noksan Öznitelik Seti LSTM Algoritması ile Sınıflandırma Performans Analizi	111
Çizelge 6.1.17	Belirsizlik Analizi Öznitelik Seti Ada Artırım Algoritması Performans Analizi	112
Çizelge 6.1.18	Belirsizlik Analizi Öznitelik Seti Karar Ağacı Algoritması ile Sınıflandırma Performans Analizi	113
Çizelge 6.1.19	Belirsizlik Analizi Öznitelik Seti Gradyan Artırım Algoritması ile Sınıflandırma Performans Analizi	114
Çizelge 6.1.20	Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması ile Sınıflandırma Performans Analizi	115
Çizelge 6.1.21	Belirsizlik Analizi Öznitelik Seti Çok Katmanlı Algılayıcı Algoritması ile Sınıflandırma Performans Analizi	116
Çizelge 6.1.22	Belirsizlik Analizi Öznitelik Seti Destek Vektör Makineleri Algoritması ile Sınıflandırma Performans Analizi	117

Çizelge 6.1.23	Belirsizlik Analizi Öznitelik Seti Sinir Ağları Algoritması ile Sınıflandırma Performans Analizi	118
Çizelge 6.1.24	Belirsizlik Analizi Öznitelik Seti LSTM Algoritması ile Sınıflandırma Performans Analizi	119

ŞEKİLLER

Şekil 1.0.1	Hisar - O Sistemi [2]	1
Şekil 2.1.1	Bir Çarpışma Testi Sırasında Yüksek Hızlı Görüntüleyici Tarafından Toplanan Görüntüler [8]	8
Şekil 2.2.1	Karar Ağacı Algoritması	11
Şekil 2.2.2	Ada Artırım Algoritması [19]	14
Şekil 2.2.3	Gradyan Artırım Algoritması [24]	16
Şekil 2.2.4	Rastgele Orman Algoritması [26]	17
Şekil 2.2.5	Doğrusal Ayrılabilen İki Sınıf [14]	18
Şekil 2.2.6	Algılayıcı	21
Şekil 2.2.7	Sinir Ağları Katmanları	22
Şekil 2.2.8	LSTM Hücresi [33]	24
Şekil 2.2.9	Dengesiz Öznitelik Seti Grafiği	31
Şekil 2.2.10	Dengesiz Normalize Edilmiş Öznitelik Seti Grafiği	32
Şekil 2.2.11	SMOTE Yöntemi Uygulanmış Veri Seti	33
Şekil 2.2.12	ROS Yöntemi Uygulanmış Veri Seti	34
Şekil 2.2.13	RUS Yöntemi Uygulanmış Veri Seti	35
Şekil 2.2.14	RUS - ROS Sınıf Dengeleme Algoritmaları [40]	35
Şekil 2.2.15	SMOTE + ENN Yöntemi Uygulanmış Veri Seti	37
Şekil 2.2.16	SMOTE - Tomek Yöntemi Uygulanmış Veri Seti	38
Şekil 2.2.17	Yakındakini Atma Yöntemi Uygulanmış Veri Seti	40
Şekil 2.2.18	Hata Dizeyi	41
Şekil 3.1.1	Dört parçacıklı İmha Senaryosu	45
Şekil 3.1.2	Üç parçacıklı İmha Senaryosu	45
Şekil 3.1.3	Parçacık Açılarının Normal Dağılımı	46
Şekil 3.1.4	Iskalama Senaryoları	47
Şekil 3.1.5	RRP-117 Radarı, Lockheed Martin [44]	48
Şekil 3.2.1	Belirsizlik Analizi [1]	51
Şekil 3.2.2	Ham Radar Sinyal Veri Senaryo Akışı	52

Şekil 3.2.3	Noksansız Veri Senaryo Akışı	53
Şekil 3.2.4	Noksansız Veri Iskalama Senaryosu Öznitelik Grafiği	54
Şekil 3.2.5	Noksansız Veri İmha Senaryosu Öznitelik Grafiği	55
Şekil 3.2.6	Noksansız Veri Seti Dengesiz Öznitelik Seti Grafiği	56
Şekil 3.2.7	Noksan Veri Senaryo Akışı	57
Şekil 3.2.8	Noksan Veri Seti Iskalama Senaryosu Öznitelik Grafiği	58
Şekil 3.2.9	Noksan Veri Seti İmha Senaryosu Öznitelik Grafiği	59
Şekil 3.2.10	Noksan Veri Seti Dengesiz Öznitelik Seti Grafiği	60
Şekil 3.2.11	Belirsizlik Analizi Öznitelik Seti Senaryo Akışı	61
Şekil 3.2.12	Belirsizlik Analizi Seti Iskalama Senaryosu Öznitelik Grafiği	62
Şekil 3.2.13	Belirsizlik Analizi Seti İmha Senaryosu Öznitelik Grafiği	63
Şekil 3.2.14	Belirsizlik Analizi Dengesiz Öznitelik Seti Grafiği	64
Şekil 3.3.1	Noksansız Veri Seti Sınıf Dağılımı	66
Şekil 3.3.2	Noksan Veri Seti Sınıf Dağılımı	66
Şekil 3.3.3	Belirsizlik Analizi Veri Seti Sınıf Dağılımı	66
Şekil 3.3.4	Sınıf Dengeleme Yöntemleri Uygulanmış Noksansız Veri Seti Sınıf Dağılımı	67
Şekil 3.4.1	İmha Değerlendirme Algoritma Akışı	68
Şekil 3.4.2	Sinir Ağı Algoritma Yapısı	71
Şekil 3.4.3	LSTM Algoritma Yapısı	72
Şekil 4.1.1	Sentetik Ham Radar Sinyal Seti Performans Analizi	73
Şekil 4.2.1	Dengesiz Noksansız Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi	75
Şekil 4.2.2	ROS Uygulanmış Noksansız Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi	76
Şekil 4.2.3	ROS Uygulanmış Noksansız Öznitelik Seti Rastgele Orman Algoritması Önem Çizelgesi	77
Şekil 4.3.1	Dengesiz Noksan Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi	78
Şekil 4.3.2	ROS Uygulanmış Noksan Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi	79

Şekil 4.3.3	ROS Uygulanmış Noksan Öznitelik Seti ile Rastgele Orman Algoritması Önem Çizelgesi	80
Şekil 4.3.4	Farklı Gürültü Değerlerinde Dengesiz Noksan Veri Seti Sınıflandırma Performansı Hata Dizeyleri	81
Şekil 4.3.5	Farklı Gürültü Değerlerinde ROS Uygulanmış Noksan Veri Seti Sınıflandırma Performansı Hata Dizeyleri	82
Şekil 4.4.1	Dengesiz Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi	83
Şekil 4.4.2	ROS Uygulanmış Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi	84
Şekil 4.4.3	ROS Uygulanmış Belirsizlik Analizi Öznitelik Seti ile Rastgele Orman Algoritması Önem Çizelgesi	85
Şekil 5.0.1	İmha Değerlendirme Algoritması	89

KISALTMALAR

CART	Sınıflandırma ve Regresyon Ağacı (Classification And Regression Trees)
DTFT	Ayrık Zamanlı Fourier Dönüşümü (Discrete Time Fourier Transform)
ENN	En Yakın Komşu Yöntemi (Edited Nearest Neighbor)
FFT	Hızlı Fourier Dönüşümü (Fast Fourier Transform)
k-NN	k-En yakın Komşu (k-Nearest Neighbor)
LSTM	Uzun Kısa Süreli Bellek (Long Short Time Memory)
ROS	Rastgele Aşırı Örnekleme (Random Over Sampler)
RUS	Rastgele Alt Örnekleme (Random Under Sampler)
STFT	Kısa Zamanlı Fourier Dönüşümü (Short Time Fourier Transform)
SMOTE	Sentetik Azınlık Aşırı Örnekleme Yöntemi (Synthetic Minority Over-Sampling Technique)
SVM	Destek Vektör Makineleri (Support Vector Machine)
YA	Yakındakini Atma (NearMiss)

1. GİRİŞ

Uzun menzilli mühimmatlar hayatımıza girdiğinden beri savaş durumunda hedef düşman uçağının mühimmat ile vurulup vurulmadığını belirlemek, insan gözünün ötesinde bir mekanizma ile çözülebilecek bir sorun olmaya başlamıştır. Bu sorun günümüzde hala önemini koruyan bir tespit konusudur.



Şekil 1.0.1: Hisar - O Sistemi [2]

Bir radar savunma sistemiyle tespit edilen tehdit unsuruna karşı başlatılan angajmanın başarılı olup olmadığının anlaşılması, yapılan savunma atışının başarı performansının bilinmesi ya da aynı tehdide yeniden bir angajman ataması gerekliliğinin tespiti gibi kararlar, hava savunma konseptinin belirlenmesi sürecinde önem arz etmektedir. Tehdidin imha edilip edilmediği tespiti ne kadar doğru yapılırsa, ikinci mühimmat tahsisi kararı da aynı şekilde doğru olarak hesaplanabilmektedir. İmha değerlendirme kararının yanlış verilmesi, kaynakların israf edilmesine ve mevcut tehdidin bertaraf edildiği yanılgısına sebep olabilmektedir. İmha değerlendirme kararının doğru verilmesinin ise gerektiği anda birden fazla angajman tahsisi kararı hızlı bir şekilde vererek kaynakların etkin ve çevik kullanımını sağlayabileceği değerlendirilmektedir.

İmha kararı değerlendirme incelendiğinde literatürde bu konu özelinde radar, elektro optik sensör ve operatör kararından yararlandığı görülmektedir. Konu özelinde kullanılan tekniği tam anlamıyla yansıtan çok fazla kaynak bulunmamaktadır.

İmha değerlendirme konusunda en temel sensörlerden biri olan radar teknolojileri incelen-

diğinde, radar sistemlerinin uzun süre büyük bir gizlilik ile yürütölmüş projeler olduđu ve genellikle askeri amaçlar için geliştirilmiş oldukları görölmektedir. Ancak günümüzde radar uygulamalarının sivil alanlarda da kendine sıkça yer bulduđu görölmektedir. Radarlar gözetleme, dost ya da düşman unsurları tespit etme, bu unsurları izleme ve tanımlama gibi amaçlar için küçük, orta ve uzun mesafelerde kara, deniz ve hava sistemlerinde rehberlik amaçlı kullanılmaktadır. Radar sistemleri yarım asırdan fazla bir süredir gündüz ya da gece fark etmeksizin hareketli veya hareketsiz nesnelerin tespitinde bahsedilen amaçlar için etkin şekilde kullanılmaktadır.

Radar sistemleri aracılığıyla tespit edilen tehdit ve müttefik unsurlardan elde edilen verilerin analiz edilmesine yönelik çalışmalar da radar sistemlerinin gelişimine paralel şekilde ilerlemektedir. Radar verilerinin belirli araçlar yardımıyla farklı formlara dönüştürölerek kritik çıkarımların yapılabilmesi, sinyal istihbaratı, elektronik istihbarat ve görüntü istihbaratı gibi faaliyetlerin de temeli oluşturmaktadır.

Radar sinyallerinin tanınması, özellikle elektronik istihbarat keşifleri alanında büyük öneme sahiptir. Gerçek zamanlı güvenilir radar sinyali tanıma ile olası tehditleri tanıma, belirleme güvenlik ve savunma açısından büyük bir önem arz etmektedir. Bu soruna çözüm olabilmesi adına elde edilen radar verilerinin, makine öğrenme yöntemleriyle tanınması ve sınıflandırılması çalışmaları günümüzde birçok alanda etkin şekilde kullanılmaktadır. Sivil uygulamalarda gemilerde, uçaklarda havacılık radarlarının tespitinde ve tanımlanmasında, ayrıca savunma ve gözetim amaçlı hava hedeflerinin tespit edilmesinde ve sınıflandırılmasında kullanılmaktadır. Radarlar hedefi belirleme konusunda başarılıdır fakat hedefi tanıma konusunda operatör desteğine ihtiyaç duymaktadırlar. Bu sınıflandırma kararı da operatörden operatöre değışeceğinden belirli bir performans kriteri oluşturulamamaktadır. Otomatik bir hedefi belirleme ve tanıma özelliğinin olması insana bağımlılığı azaltacaktır. Bu sorunu çözme amaçlı yapılan bir çalışmada [3] makine öğrenmesi algoritmalarından yapay sinir ağıları, destek vektör makineleri, rastgele orman algoritmaları incelenmektedir.

Elektronik istihbarat keşifleri dışında yaya, uçak ve kamyonlardan alınan radar sinyalleri de yapay zekâ algoritmaları aracılığıyla sınıflandırma amaçlı kullanılmaktadır [4]. [5] makalesinde ise insan düşme durumunu derin öğrenme algoritması ile sınıflandırmak için bir yaklaşımdan bahsedilmektedir. Hem zaman frekans hem de menzil verileri birlikte kullanılarak hata oranı minimize edilmesi hedeflenmektedir. Elde edilen öznitelikler üzerinden derin öğ-

renme ile sınıflandırma işlemi gerçekleştirilmektedir. Ayrıca öznitelik olarak spektrogram, menzil-Doppler grafikleri ve bu ikisinin birlikte füzyonu verilerek öğrenme işleminde sonucu nasıl etkiledikleri incelenmektedir. Radardan dönen sinyalden çıkarılan öznitelikler temel alınarak bu sınıflandırma işlemi yapılmaktadır.

Hedef tanıma ve imha durumu belirleme konuları benzer konulardır. İmha değerlendirme için sensörden tehdide ait bir örüntü elde edilmektedir, daha sonrasında tehdit ve mühim-mata ait bir örüntü elde edilerek buluşma noktalarından önce ve sonrası için yine bir örüntü elde edilerek bu anlara ait örüntülerin karşılaştırılmasından imha durumu belirlenmeye çalışılmaktadır. Yapılması planlanan imha değerlendirme algoritmasında da burada bahsedilen yaklaşıma benzer şekilde radar sinyalleri kullanılarak çıkarılan öznitelikler üzerinden makine öğrenmesi ve derin öğrenme algoritmaları ile bir durum sınıflandırma işlemi gerçekleştirilmesi hedeflenmektedir.

Makine öğrenimi, bilgisayar sistemlerine, açıkça programlanmadan, “öğrenerek” ilerlemelerini sağlamak ve performanslarını geliştirme yeteneği kazandırmak için çeşitli istatistik ve diğer teknikleri kullanan bir bilgisayar bilimi alanıdır. Temel amaç veriden bilgi elde edilmesini sağlamaktadır. Verilerin oluşturulduğu düzenlilikleri, özellikleri ve kuralları öğrenerek bir sınıflandırma ya da tahmin mekanizması ortaya koyar. Makine öğrenmesinin temel fikri, verilerden öğrenebilecek ve davranışlarını tahmin edebilecek algoritmalar bulmaktır. Temelde, bilgisayar destekli tahminler yapmak için algoritmalar oluşturmayı da amaçlayan bu bilim temelde hesaplama istatistikleriyle doğrudan ilişkilidir.

Askeri alanda radar sinyallerinin analizinde, hedef tespitinin yapılmasında ve sınıflandırılmasında, hedef performanslarına yönelik tahminlerin çıkarılmasında ve bu gibi birçok çalışmada makine öğrenmesi yöntemlerine sıkça başvurulmaktadır. [6] 5 farklı sınıftan oluşan radar hedef tanıma problemi için gerçek zamanlı yinelenen öğrenme algoritması kullanılmaktadır. Geniş bantlı radardan gönderilip hedeften yansıyan sinyallere ait yanca ve yükseliş açıları yinelenen sinir ağlarına girdi olarak kullanılmaktadır. Sensör ile ölçüm, sinyal işleme, öznitelik çıkarımı ve sınıflandırma işlemleri olmak üzere dört aşama uygulanmaktadır.

Bu tez ile radar sinyalleri ve makine öğrenmesi kullanılarak imha kararının verilebildiği yeni bir yaklaşım geliştirilmesi hedeflenmektedir. Radar sinyallerinden çıkarılan öznitelikler kullanılarak makine öğrenmesi yöntemleri ile kesin ve zamanında değerlendirme yapılarak doğru bir imha değerlendirme yapabilmek ana amaç olacaktır. Öncelikle makine öğrenme

yöntemleri analiz edilecek, daha sonra radar sinyallerinin makine öğrenme yöntemleri ile sınıflandırılmasına dair çalışmalar incelenerek, bunlarla ilgili literatür taraması yapılacaktır. Tehdit imha senaryolarına dair gerçek radar verilerine sahip olunmadığı için bir simülatör ile fiziksel ortam simüle edilerek sentetik radar sinyalleri ve üç farklı tip öznitelik seti oluşturulacaktır. Daha sonra bu oluşturulan öznitelikler kullanılarak seçilen çeşitli makine öğrenme yöntemleri ve derin öğrenme algoritmaları ile doğru bir imha kararının verilmesinde performans ve doğruluk açısından hangi yöntemin daha doğru ve optimum çalıştığı saptanacaktır. Performans analizleri sonucuna bağlı olarak hangi makine öğrenmesi metodunun imha değerlendirme algoritmasında kullanılmasının daha uygun olacağı kararı verilecektir. Birden fazla uygun metot olması durumunda yöntemler birbirleri ile karşılaştırılacaktır.

İmha değerlendirmesinde, makine öğrenmesi ve derin öğrenme tekniklerinin kullanılmasıyla karar alma sürecinde kararlarının insan figüründen bağımsız olarak verilmesi ve böylece hataların ve kaynak israflarının azaltılması hedeflenmektedir.

2. LİTERATÜRDE İMHA DEĞERLENDİRME VE YAPAY ZEKÂ ALGORİTMALARI

2.1. İmha Değerlendirme

İmha değerlendirme başlatılan bir angajman sonucunda ilgili tehdidin vurulup - vurulmadığı kararının verilebilmesi, başlatılan angajmanın başarılı olup olmadığının değerlendirilmesidir.

Günümüz teknolojisi tehdit unsurlarının kilometrelerce uzaklardan tespit ederek ortadan kaldırmak için gerekli angajmanın yapılabilmesini mümkün kılmaktadır. Tehdidin sonlandırılıp sonlandırılmadığını öğrenebilmek ve bu bilgi sonucunda yeni bir angajmana ihtiyaç olup olmadığının kararını verebilmek için imha değerlendirmesinin yapılmasına ihtiyaç duyulmaktadır. Ancak angajman sonrası tehdidin sonlandırılıp sonlandırılmadığı bilgisi artık insan gözü ile ayırt edilebilecek noktada değildir.

İmha değerlendirme kararı verebilme ve imha etme durumu arasında bir denge durumu söz konusudur. İmha olasılığının yüksek olması imha kararını belirleme kısmında çeşitli optimizasyonlara neden olabilir. Düşük imha ihtimali olan iyi bir imha kararı verebilen sistem yerine yüksek imha ihtimali olan fakat imha kararı konusunda yüksek başarıları olmayan sistemler daha çok tercih edilmektedir [7].

İmha değerlendirme kararını doğru şekilde ve tam zamanlı verebilmenin iki önemli nedeni bulunmaktadır. Bunlardan biri yeni bir mühimmat hazırlanması durumu oluştuğunda bunun için gereken reaksiyon süresine ihtiyaç duyuyor olmaktır. Eğer doğru ve tam zamanlı bir karar verilebilirse başarılı bir yeni mühimmat atama işlemi gerçekleştirilebilir. Diğer amaç ise gereksiz mühimmat kullanımını önlemektir. Eğer karar yanlış ya da bir değerlendirme bilgisi yoksa bu durum kaynak israfına sebep olabilmektedir. İmha kararı, kaynakların doğru şekilde kullanımına ve kısa bir zaman aralığı içerisinde birden fazla hedefe angajman tahsisine imkân sağlamaktadır [7].

Çeşitli sensörler tarafından sağlanan verilere bakıldığında tehdidin sahip olduğu özelliklerin vurulma öncesi, vurulma anında ve vurulma sonrası farklı niteliklerde olması imha değerlendirme için girdi sağlamaktadır. Dinamik ve dinamik olmayan niteliklerden oluşan veri kombinasyonları imha kararını daha kesin hale getirme amaçlı kullanılmaktadır. Bu bilgi

kapalı döngüye ait bir bilgi olmadığından imha kararı hiçbir zaman kesin bir bilgi olmayacaktır. Bu sebeple tehdit ile alakalı birçok bilgiye ait verinin birleşiminden oluşan bir karar algoritması oluşturuyor olmak bu doğru bilgiye ulaşım konusunda daha güvenilir bir sistem sağlamaktadır.

Karar aşamalarına ait sonuçlar hem sinyal işleme konusundaki hem de bu kararı verecek algoritmaların koştığı yapılardaki bilgisayarların gücüne de bağlı olacaktır [7].

İmha değerlendirme kararı için uygulanan yöntemlere bakıldığında bu veriler radar, insan, optik sensör, arayıcı başlık kaynaklı olabilmektedir. Tehdidin mühimmat ile buluşma noktasından önceki ve sonraki hareket dinamikleri incelenerek karar verilmesi amaçlanmaktadır. Sensörlerden alınan verilerle bu dinamiklerde meydana gelen değişim incelenerek tehdiye yönelik hasar ya da imha durumunun söz konusu olup olmadığına dair bir bilgi sunumu sağlanabileceği değerlendirilmektedir [7].

İmha değerlendirme algoritmaları için tehdit ve füzeye ait bilgi verebilecek en temel sensörlerden biri radardır. Mühimmatın tehditle buluşma zaman aralığındaki radardan gönderilen sinyalde meydana gelen değişiklik bir potansiyel vurulma ihtimali örneği olarak görülebilir. Radyo frekans izgesindeki veriler tehdit ve mühimmat hakkında birçok bilgiyi sunmaktadır. Objelerden yansıyor gelen radar sinyallerinde objelerin dinamik karakteristiklerine ait (hız, yönelim, Doppler, radar kesit alanı vb.) bilgiler yer almaktadır.

Bir çarpışma sonrası tehdit ve mühimmatın yapısında meydana gelen değişimler radar sinyallerden elde edilen objelerin dinamik karakteristiklerine yansımaktadır ve imha değerlendirme yapılmasında fayda sağlamaktadır. Mühimmat fırlatıldığında tehdiye ait radar sinyal yansımaları alımının durması yani tehdit izinin düşmesi veya çok sayıda hareket eden objeye ait Doppler izgesi elde edilmesi, imha değerlendirmesi için bir girdi sunmaktadır [7].

Patlama ile birlikte ortaya çıkan ışıksal-optiksel değişimler de yine imha değerlendirmesine girdi olabilecek bilgiler arasında yer almaktadır. Bir jet ya da roketin radarda oluşturduğu görüntü imha sonucu daha farklı bir görüntüye dönüşebilmekte ya da yok olabilmektedir. Çarpışma anından sonra tehdiye ait radar görüntüsünde imha durumu belirgin olmaktadır. İmha değerlendirmesi için sensörden tehdiye ait bir örüntü elde edilmektedir. Daha sonrasında tehdit ve mühimmata ait bir örüntü elde edilerek buluşma noktalarından önce ve sonrası için yine bir örüntü elde edilerek bu anlara ait örüntülerin karşılaştırılmasından imha durumu belirlenmeye çalışılmaktadır [8]. Bir yapay zekâ algoritması ile sensörlerden toplanan veri-

lere göre bir karar verme işlemi gerçekleştirilebilir.

Bazı radarlar diğer verilere ek olarak hedefe ait radar kesit alanı, sinyalin enerji genliği gibi değerleri de hesaplamaktadır. Tehdidin parçalanmasına ait bu değerlerdeki değişiklikler de buluşma noktası sonrası için bir girdi olabilmektedir. Radar görüntüsü ile hedefe ait şekil ve uzunluklar belirlenebilmektedir. Bu tarz radarlar ile benzer şekilde bu nitelikteki değişiklikler de birer gösterge olarak kullanılabilir. En önemli bilgi kaynaklarından biri de takip radarından elde edilen Doppler bilgisidir [7].

Eski sistemlerde kullanılan X bandı sinyal aralığında yayın yapan sürekli radarların tehditte aldıkları yansımaların insanların dinleyebileceği frekanslara çevrildiği ve ses sinyallerinden tecrübeli operatörlerin imha konusunda çıkarımda bulunduğu belirtilmektedir [7].

İmha değerlendirme konusunda radar sinyaline ait Doppler bilgisinin imha değerlendirme için kullanılması mümkündür. Bu konuda literatürde yapılan diğer çalışmalardan bahsedilecektir.

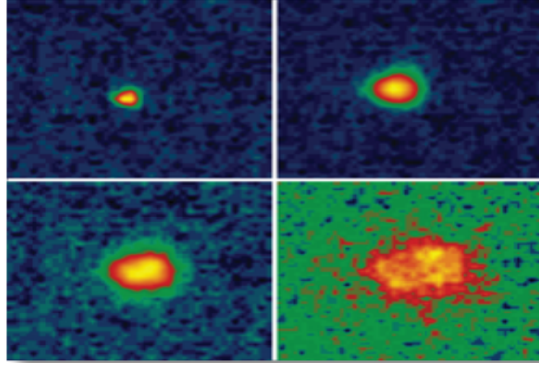
Radar sinyallerinden sinyal işleme çalışmaları ile tehdiide ait Doppler karakteristiği çıkarıldıktan sonra bu öznelik ile tehdidin imha edilip edilmediği bilgisinin yapay zekâ algoritmalarında kullanımından bahsedilmektedir [9].

Sürekli dalga radarı kullanılarak tehdit ve çarpışma sonucu oluşacak olan olası parçacıklara ait hareket ile alakalı verilerden elde edilen Doppler bilgisi kullanılarak imha değerlendirme yapan bir diğer çalışmada ise öncelikle yapay zekâ tabanlı Hopfield doğrusal durum kestirim algoritması ile hız kestirimi yapılmaktadır. Kestirim sonucu, istatistiksel karar teorisine girdi olarak kullanılmaktadır. Hipotez sonucunda hedefin imha edilip edilmediği kararı verilmektedir. Tehdidin vurulup vurulmadığına dair bilgi radyal hız bilgisi ile çıkarılabilmektedir. Vurulma anında ve sonrasında tehdiide ait Doppler bilgisinde bir değişim olmaktadır. Başarılı bir imha durumu tek hedef varken yörünge etrafında birden fazla parça oluşma durumundan anlaşılmaktadır [10].

Bir başka çalışmada imha kararı verebilmek için girdi olarak mühimmatın imha gücü, tehdit ve hedefin buluşma noktası, optik ve radar sensörlerden alınan tehdidin kütle, enerji ve tip bilgisi kullanılmaktadır. Bu bilgiler operatör ve karar vericiler tarafından kullanılmaktadır. İmha kararı verebilmek için mühimmatın fiziksel modellerinden, mühimmatın ve tehdidin yüksek hızlı sensörleri tarafından kaydedilmiş bilgilerinden ve bu bilgileri kullanarak karar

verebilen algoritmadan bahsedilmektedir [8].

İmha değerlendirilmesi kararı için mühimmata ve hedefe ait kinetik enerji, tehdidin vurulması sırasında açığa çıkan parçacıklara ait enerjiler ve enerji kayıpları gözetilerek enerji korunumundan yararlanılmaktadır. Patlama sonucunda enerjinin bir kısmı ısı olarak ortaya çıkmaktadır. Geri kalanda parçalar arasında korunacak şekilde dağılmaktadır. Hedef sıcaklıkları, sayısı ve boyutları bir parça bulutu oluşturmaktadır [8].



Şekil 2.1.1: Bir Çarpışma Testi Sırasında Yüksek Hızlı Görüntüleyici Tarafından Toplanan Görüntüler [8]

İmha değerlendirmesi yapılabilmesi için yüksek zaman çözünürlüklü çoklu spektral veriye ihtiyaç duyulmaktadır. Bu sebeple yüksek hızlı görüntüler, yüksek hızlı spektral veriler ve yüksek hızlı radyometrik veriler toplayan bir sensör geliştirilmiştir. Bu sensörlerden gelen veriler, optik imzalardan ve model doğrulamasından sorumlu fiziksel mekanizmaları tanımlamak ve imha değerlendirme için kullanılmaktadır [8].

2.2. Yapay Zekâ Algoritmaları İle Sınıflandırma

Örüntü tanıma, sınıflandırma, optimizasyon, kümeleme gibi konularda yapay zekâ algoritmaları etkin çözümler sunmaktadır.

[4] çalışmasında hedefleri otomatik olarak tespit edip yapay zekâ aracılığıyla kendi arasında sınıflandırabilen bir radar sisteminden bahsedilmektedir. Yapay zekânın eğitim ve test sürecinde yer alan veriler, radar tespit modundan elde edilen radar görüntülerinden oluşmaktadır. Yaya, uçak ve kamyonlardan alınan radar sinyalleri, hedefleri sınıflandırma amaçlı kullanılmıştır. Simülasyon sonuçları radar sistemlerinin farklı objeleri algılayıp yapay zekâ algoritmaları ile sınıflandırma konusunda başarılı olduğunu göstermektedir. Hedefleri sınıflandırma işlemi yüksek çözünürlüklü görüntülerin girdi olarak kullanıldığı yapay sinir ağları yöntemi ile gerçekleştirilmektedir [4].

Bir başka çalışmada, radar imzasının oluşturulması, gürültü eklenmesi, öznitelik çıkarımı, sinir ağını eğitmek ve hedef tanıma doğruluğunu anlamak için gerçekleştirilen test adımları yer almaktadır. Bu çalışmaya kadar yinelenen sinir ağları, kestirim ve kelime tanıma amacıyla kullanılmışken radar hedefi tanıma amaçlı kullanılmamıştır [6]. Bahse konu bu algoritmaya sağlanan girdi vektörü, radar sinyalinden oluşturulan öznitelik vektörüdür.

Doppler radar kullanılarak elde edilen insan jestlerinden yansıyan sinyallerden zaman ve frekans analizleri yapılarak elde edilen özniteliklerin makine öğrenmesi algoritmalarına girdi olarak kullanıldığı bir çalışmada ise sınıflandırıcının farklı tipteki jestleri ayırabilmekte başarılı olduğu fakat benzer el jestlerinin hareket yönünü kestirme konusunda başarılı olamadığı belirtilmektedir [11].

Benzer mantıktaki bir başka çalışmada düşük güçlü, kısa mesafeli Doppler darbe radar sistemi kullanılarak insan, köpek ve bunlar dışındaki gürültü olarak adlandırılacak cisimlerin ayrımı hedeflenmiştir. Zaman frekans analiz yöntemleri ile öznitelik çıkarım işlemi gerçekleştirilerek ve çok katmanlı yapay sinir ağları, karar ağacı ve SVM ile testler yapılarak sınıflandırma performansları incelenmektedir [12].

[9] çalışmasında hedefin vurulup vurulmadığına dair nihai bir karara varmak için sensör verilerinden elde edilen tahminleri kullanan bir yapay zekâ algoritmasından bahsedilmiştir.

[10] çalışmasında ise sinir ağları kullanılarak sistem durumu kestirim işlemi gerçekleştirilmektedir, karar mekanizmasında yapay zekâ kullanımı gelecekte yapılacak çalışmalar olarak

hedeflenmiştir. İmha değerlendirme konusunda literatürde çok fazla yapılmış çalışma bulunmadığı gibi daha alt bir küme olan yapay zekâ algoritmaları ile sınıflandırma işlemi yapılarak imha değerlendirme yapılması konusu da henüz uygulanmış bir yöntem olarak gözükmemektedir.

Bu çalışmada yapılması planlanan literatürden farklı olarak radar sinyallerinin zaman frekans yöntemleri ile analizi sonucu çıkartılan özniteliklerin farklı makine öğrenmesi algoritmaları ile eğitilerek imha, ıskalama, vurulma öncesi durumlarının sınıflandırıldığı bir imha değerlendirme algoritmasının geliştirilmesidir.

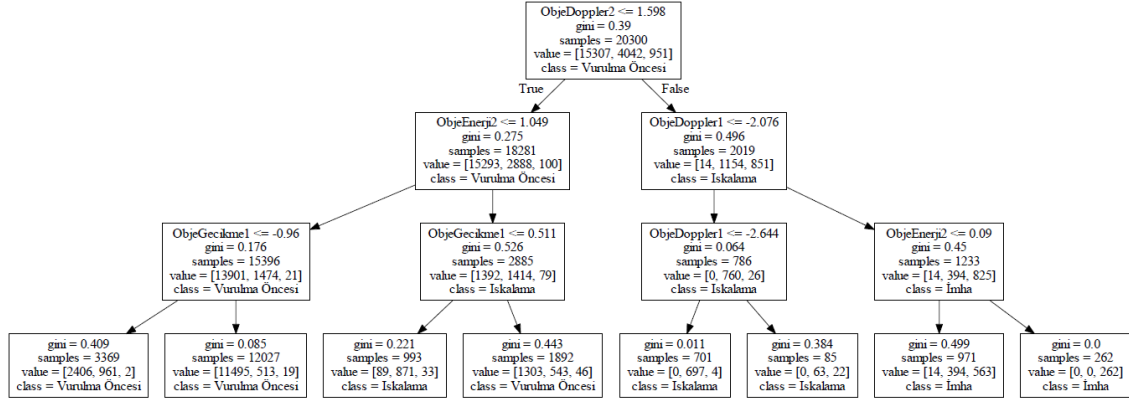
Bu bölümde makine öğrenmesi ve derin öğrenme algoritmaları, öznitelik çıkarımı, sınıflandırma ön işlemleri, sınıf dengesi sağlama yöntemleri ve sınıflandırma performans metriklerinden bahsedilecektir.

2.2.1. Makine Öğrenmesi Algoritmaları

Makine öğrenmesi algoritmaları türlerinden biri olan denetimli öğrenme algoritmaları için önceden etiketlenmiş girdi ve sonuç belirten sınıfların olması gerekmektedir. Eğitim işleminde öğrenilen bilgi ile daha sonrasında test edilen verinin hangi sınıfa ait olduğu belirlenir. Eğitim işlemi için etiketlenmiş bir eğitim veri setine ihtiyaç duyulmaktadır. Denetimli öğrenme de belirli bir uyarana istenen tepkinin ne olması gerektiği belirtilerek ağına eğitime yardımcı olmak için bir "öğretmen" kullanılmaktadır [13].

Denetimli öğrenmede, bir dizi eğitim verisinin olduğu varsayılır. Sınıflandırıcı, bu önceden bilinen bir bilgidен yararlanılarak tasarlanmaktadır. Denetimli öğrenme algoritmalarına örnekler arasında regresyon analizi, karar ağaçları, k-en yakın komşular, sinir ağları ve destek vektörü makineleri bulunmaktadır. Denetimli bir öğrenme probleminde, önceden etiketlenmiş girdi-çıkı çiftleri mevcut olmalıdır. Bir sistemin öğrenebileceği bu eğitim örnekleri verilerdeki kalıplara veya eğilimlere uyum sağlamak için öğrenme sistemi içindeki bilgileri kullanmaktadır. Eğer öğrenme başarılı olduğunda, yeni giriş verilerine maruz kalırsa eğitimli sistem kendi çıktısını yaratarak önceden öğrendiği davranışı taklit edebilmektedir. Her eğitim örneği bir çift girdi ve çıktı değerinden oluşmaktadır. Sınıflandırma probleminde de her eğitim örneği bir özellik vektörünü oluşturmaktadır. Denetimli öğrenmeyle bu özelliklere karşılık gelen çıktı sınıfı oluşturulmaktadır.

Makine öğrenmesi algoritmalarından karar ağacı algoritmaları, ada artırım ve gradyan artı-



Ŗekil 2.2.1: Karar Ađacı Algoritması

rım algoritmalarının içinde bulunduđu kolektif öğrenme algoritmaları, rastgele orman algoritması, destek vektör makineleri algoritması incelenecektir.

2.2.1.1 Karar Ađacı Algoritması

Karar ađacı algoritmasının amacı veri matrisini belirli kurallar silsilesinde homojen alt gruplara ayırmaktır. Karar ađacı algoritması böl ve yönet stratejisini uygulayan hiyerarşik bir veri yapısıdır. Parametrik olmayan kural tabanlı bir tekniktir.

Karar ađacı algoritması makine öğrenmesi algoritmaları içerisinde en basit yöntem olarak değerlendirilebilir. Karar ađacı algoritması sürekli veri seti olmayan sınıflandırmalar için daha uygundur. Veriyi özniteliklere göre belirlediđi eşik değeri temel alacak şekilde ikiye bölerek bir sınıflandırma işlemi gerçekleştirerek bir ađaç yapısı oluşturmaktadır. Bu algoritma esasında kök düğüm, orta düğümler ve yapraklardan oluşmaktadır. Her dal belli bir sonuca ulaşarak sınıflandırma işlemini ifade etmektedir.

Her bir karar düğümü, dalları etiketleyen ayrık sonuçlara sahip bir test fonksiyonu uygulamaktadır. Bir girdi verildiğinde, her düğümden bir test uygulanır ve sonuca bađlı olarak dallardan birine yönelir. Bu işlem kökten başlayarak yapraklara ulaşınca kadar devam etmektedir. Yaprakta yazan değeri vermektedir. Karmaşık bir problem basit ve seri yapılara ayrıştırılmaktadır [14].

Sınıflandırma işlemi için bir yaprak, düğüm, giriş alanından bu bölgeye düşen örneklerin aynı etiketlere sahip olduđu bir bölge tanımlanmaktadır. Karar mekanizmasını oluşturan sınırlar kökten yaprak düğüme giden yol üzerindeki dâhili düğümlerde kodlanan diskriminantlar

tarafından tanımlanmaktadır. Öğrenme sırasında verinin içerdiği problemin karmaşıklığına bağlı olarak ağaç büyür, yeni dallar ve yapraklar eklenmektedir.

Karar ağaçları, öğrenme örneğini daha küçük ve daha küçük parçalara bölen bir dizi soru ile temsil edilmektedir. Bu kurallara göre hiyerarşik yapıya sahip bir ağaç yapısı oluşturulmaktadır. Ağaçlar, öznitelik değerlerini izin verilen bir hipotezi temsil eden bir dizi karar sınıfına eşleyen bir işlevi temsil etmektedir. Tüm düğümler, seçilen bölme kriterlerine göre oluşturulan öznitelikler üzerinde kurallar içermektedir. Kurallar verileri o verinin öznitelik değerlerine göre bölen veri bölünmesini belirlemenin yolunu temsil etmektedir. Kökten yapraklara doğru giderken dallarda özniteliklerin ayrılması için sonuçlar verilmektedir. Tüm kural sonuçları dallarla temsil edilmektedir. Yapraklar ayrışmanın gerçekleştirildiği son basamaklardır [15].

Her bir düğümde verilerin nasıl ayrılacağına belirlenmesi karar ağaç algoritmaları açısından en kritik karardır. Sınıflar arası ayrımın en hatasız olacak şekilde gerçekleştirilmesi için çeşitli ayrılma ölçütleri belirlenmektedir. Bu ayrılma ölçütleri, ayrılmadan önceki ve sonraki sınıf dağılımı kayıtlarına bağlı olarak tanımlanmaktadır. Bu ayrılma ölçütlerinden biri de Eşitlik 2.2.1'de verilen gini ölçütüdür [16]. Şekil 2.2.1'de gini sabitine göre nasıl ayrıldığına dair bir örnek verilmektedir.

$$Gini = 1 - \sum_{i=1}^c (p_i)^2 \quad (2.2.1)$$

Algoritma ayırma işlemi için en iyi sonucu verecek özniteliği aramaktadır. Farklı karar ağaç yöntemleri için ayırtaç fonksiyonu da farklı olmaktadır. Ayırma kriterleri için en önemli şart düğümdeki saflığı artırmaktır.

2.2.1.1.1 Sınıflandırma ve Regresyon Ağacı (CART) Sınıflandırma ve regresyon ağaç yaklaşımı 1984 yılında geliştirilmiştir. Üç temel aşamadan oluşmaktadır. İlk olarak CART, ikili ağaçlar oluşturması ile karakterize edilmektedir. Her bir iç düğümün tam olarak iki giden kenarı vardır. Bölmeler, Gini kriterleri kullanılarak seçilmektedir. CART, karesel tahmin hatasını en aza indiren bölmeleri arar. Karar ağaçları, öğrenme örneğinin daha küçük parçalara çoklu bölünmesini gerçekleştiren bir bölme kuralına göre oluşturulmaktadır. Ağaç yapısı elde edildikten sonra yeni verilerin sınıflandırılması için kullanılmadan önce optimize edilmeleri gerekmektedir.

Maksimum ağaçların çok yüksek karmaşıklıkta olduğu ve yüzlerce seviyeden oluştuğu ortaya çıkarılmaktadır. Bu nedenle ağaç optimizasyonu, doğru ağaç boyutunu seçmeyi, önemsiz düğümleri ve hatta alt ağaçları kesmeyi gerektirmektedir. Pratikte iki budama algoritması kullanılabilir; her düğümdeki nokta sayısına göre optimizasyon veya çapraz doğrulama. Son olarak da kurulan yeni ağaç yapısından verinin sınıflandırılması sağlanmaktadır [17].

CART algoritmasının avantajları: CART algoritması, en önemli değişkenleri kendisi belirleyebilme ve önemsiz olanları ortadan kaldıracak kabiliyetine sahiptir. Aykırı değerler ile başa çıkabilmektedir. CART algoritması, en iyi bölünmeyi bulmak için tüm olası değişkenleri ve tüm olası değerleri aramaktadır. Amacı verileri maksimum homojenlikle iki parçaya bölen soruyu sormaktır. İşlem daha sonra elde edilen veri parçalarının her biri için tekrarlanmaktadır. Karar ağaç yapıları "if/else" kurallarına çevrilebildiğinden anlaşılması oldukça kolay yapılardır. Bu sebeple bazen karmaşık fakat daha doğru yöntemlerin yerine tercih edilmektedirler.

CART algoritmasının dezavantajları: Veri setindeki önemsiz sayılabilecek değişiklikler ağaç yapısındaki karmaşıklığı artırıp azaltabilmektedir. Bu sebeple CART kararsız bir ağaç yapısıdır. Ayrıca aynı anda sadece tek bir özneliğe göre ayrılmaktadır.

2.2.1.2 Kolektif Öğrenme Algoritmaları

Sınıflandırma için çoklu sınıflandırıcı yapılmasıdır. Büyük veri setlerini küçük parçalara ayırarak çoklu sınıflandırıcılar için birden fazla veri seti oluşumunu sağlamaktadır. Avantajları doğruluk oranını artırmaktadır, az ve çok sayıda veri içeren veri setleri için de çözüm sunabilmektedir. Topluluk öğrenimi yöntemi, tekli modellere kıyaslandığında daha sağlıklı sonuçlar vermektedir. Birçok farklı kolektif öğrenme tekniği bulunmaktadır. Bunlardan bazıları ortalama, ağırlıklı ortalama, yığma, torbalama, maksimum oylama, harmanlama ve yükseltme gibi tekniklerdir. Radar sınıflandırılmalarında bu yöntemlerden de yararlanılmaktadır [18].

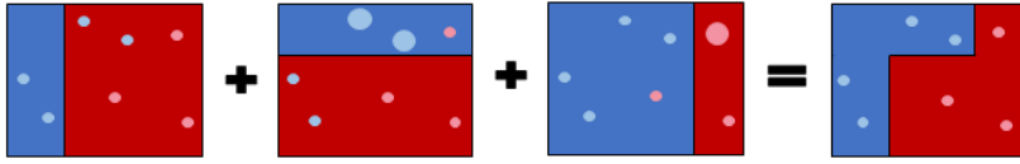
Ada artırım ve gradyan artırım gibi algoritmaları içeren ve birçok makine öğrenmesi yönteminde sıklıkla başvurulan kolektif öğrenimi yöntemi, "yükseltme" tekniğidir [19].

2.2.1.2.1 Ada Artırım Algoritması Ada Artırım algoritması, Freund ve Schapire (1996) tarafından önerilmiş olup, çeşitli zayıf sınıflandırıcıların bir araya getirilerek karma ve daha güçlü bir eğitim modeli sunan bir makine öğrenmesi algoritmasının gerçekleştirilmesi amaç-

lanmaktadır. İçerisinde kendisinden daha önce geliştirilmiş olan farklı makine öğrenmesi algoritmalarını aynı eğitim seti ile tekrar tekrar kullanmasıyla, birleştirerek ya da karşılaştırarak öğrenme konusunda en iyi sonucun elde edilmesi hedeflenmektedir. Aşırı uyumlama sorunu olmaması için basit sınıflandırıcılar tercih edilmektedir [14].

Ada Artırım algoritması arkasındaki ana fikir, birkaç temel modelin tahminlerini düşük doğrulukla veya karar ağacı gibi zayıf sınıflandırıcılarla birleştirerek sonunda yüksek doğrulukta bir model veya güçlü bir sınıflandırıcı oluşturmaktır. Artırım algoritması performansı birden çok "temel" sınıflandırıcıyı birleştirerek bir komite biçimi oluşturduğu için herhangi bir temel sınıflandırıcıdan önemli ölçüde daha iyidir [20].

Belirli bir problem üzerinden gidildiğinde artırım algoritmasının gerçek performansının, açıkça verilere ve temel sınıflandırıcıya bağlı olduğu görülmektedir. Bu yöntemde yeteri sayıda eğitim verisine sahip olunmalıdır. Ayrıca temel sınıflandırıcı da zayıf olmalı ancak çok zayıf olmamalıdır. Artırım algoritması özellikle gürültüye ve aykırı değerlere karşı hassastır.



Şekil 2.2.2: Ada Artırım Algoritması [19]

Ada Artırım algoritması bir dizi sınıflandırıcı oluşturur ve onları oylar. Daha önce oluşturulmuş sınıflandırıcılara dayalı olarak her biri bir sonraki sınıflandırıcıya girdi olarak sağlanan eğitim örneklerinin ağırlıklarını da değiştirir. Amaç, bir sonraki sınıflandırıcıda farklı girdi dağılımlarında beklenen hatayı en aza indirmektir [21].

Ada Artırım algoritması, iki sınıflı sınıflandırma problemini çözmek için başarılı bir teknik olmaktadır. İki sınıflı sınıflandırmadan çok sınıflı sınıflandırmaya geçerken çoğu algoritma, çok sınıflı sınıflandırma problemini çoklu iki sınıflı problemlere indirgemekle sınırlandırmaktadır. Burada, Ada Artırım algoritmasını birden çok iki sınıflı probleme indirgmeden doğrudan çok sınıflı duruma genişleten yeni bir algoritma geliştirilmiştir [22].

Ada Artırım yönteminde temel yaklaşım, ilk etapta veri setini bir ön bilinçle modelleyerek test ve doğrulama süreçlerinde ortaya çıkan hata sonuçlarına bağlı olarak verilerin güncellenen ağırlıklarla yeniden eğitilmesi sürecini ve en iyi sonuç alınana dek bu güncelleme iterasyonlarının sürdürülmesi faaliyetlerini içermektedir. Bu yaklaşımda yanlış sınıflandırılan

gözlemlerin ağırlık katsayıları artırılarak hatalı gözlemlerin doğru sınıflandırılmasını sağlayacak sonuçlar üretilmesi amaçlanmaktadır. İlk modelleme yapılırken tüm ağırlıklar eşit olarak başlatılmaktadır. Sonuçta sınıflandırma tahmini tüm bu farklı modellerin ağırlıklı ortalaması bulunarak hesaplanmaktadır.

Bu algoritma aşağıdaki şekilde formüle edilmiştir.

Ada Artırım 1 SAMME.R Algoritması

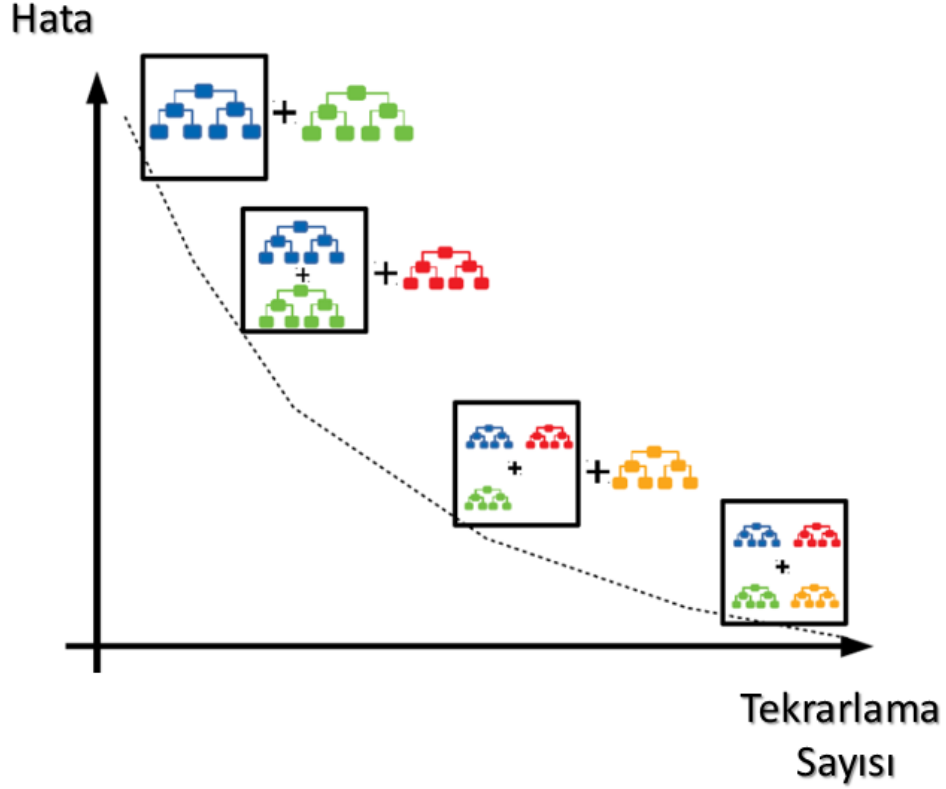
1. Gözlem ağırlıklarını ilklendirilir. $w_i = 1/N, i = 1, 2, \dots, N$
 2. $m = 1$ 'den M 'ye:
 - (a) w_i ağırlıkları ile veriyi eğitmek için bir sınıflandırıcı $T^m(x)$ seçilir.
 - (b) Ağırlıklı sınıf olasılık tahminlerini elde edilir.

$$p_k^m(x) = Prob_w(c = k|x), k = 1, \dots, K$$
 - (c) $h_k^m(x) \leftarrow (K - 1) \left(\log_{p_k^m}^m(x) - \frac{1}{K} \sum_{k'} \log_{p_{k'}^m}^m(x) \right), k = 1, 2, \dots, K.$
 - (d) $w_i \leftarrow w_i \cdot \left(\exp(-) \frac{K-1}{K} y_i^T \log_{p_k^m}^m(x_i) \right), i = 1, 2, \dots, n.$
 - (e) w_i ağırlıkları yeniden normalize edilir.
 3. Çıktı:

$$C(x) = \left[\operatorname{argmax}_k \sum_{m=1}^M h_k^m(x) \right].$$
-

2.2.1.2.2 Gradyan Artırım Algoritması Gradyan artırım tekniğinde sınıflandırıcı olarak seçilmiş karar ağaçları, Şekil 2.2.3'de verildiği gibi kendisinden bir önceki modelin hatalarının düzeltilmesine odaklı şekilde modellenmiş birçok zayıf sınıflandırıcının zincir şeklinde birleşiminden oluşmaktadır. Bir öğrenme algoritmasında hatalı sınıflandırılmış verilerin doğru sınıflandırılmasını sağlamak için, buradaki ağırlık katsayılarını güncellemek yerine yeni bir karar ağacı modelleyerek mevcut öğrencinin devamına bu modeli ekleyerek hataların azaltılması hedeflemektedir. Ayrıca çok daha küçük bir modelden başlayarak büyük bir modele doğru bir akışa sahiptir [23].

Ada artırma tekniğinde olduğu gibi başta bir öğrenci modelleyip, öğrenme sürecinin sonuna kadar o öğrenciyi güncelleyerek birçok farklı modelin ağırlıklı ortalamasının alınması prensibini taşımamaktadır. Ada artırma yöntemi ile kıyaslandığında en temel farkı budur. Gradyan artırma yöntemi özetle birçok zayıf öğrenciyi birleştirerek güçlü bir öğrenci geliştirme çabasıdır. Gradyan artırma tekniğinin en önemli özelliklerinden birisi de aşırı uyumu azaltarak algoritmanın performansını artıran bir yöntem olmasıdır. Ek olarak bu teknikte en önemli hususlardan biri de veri setinde yüksek sayıda veriye ihtiyaç duyulmaktadır [23].



Şekil 2.2.3: Gradyan Artırım Algoritması [24]

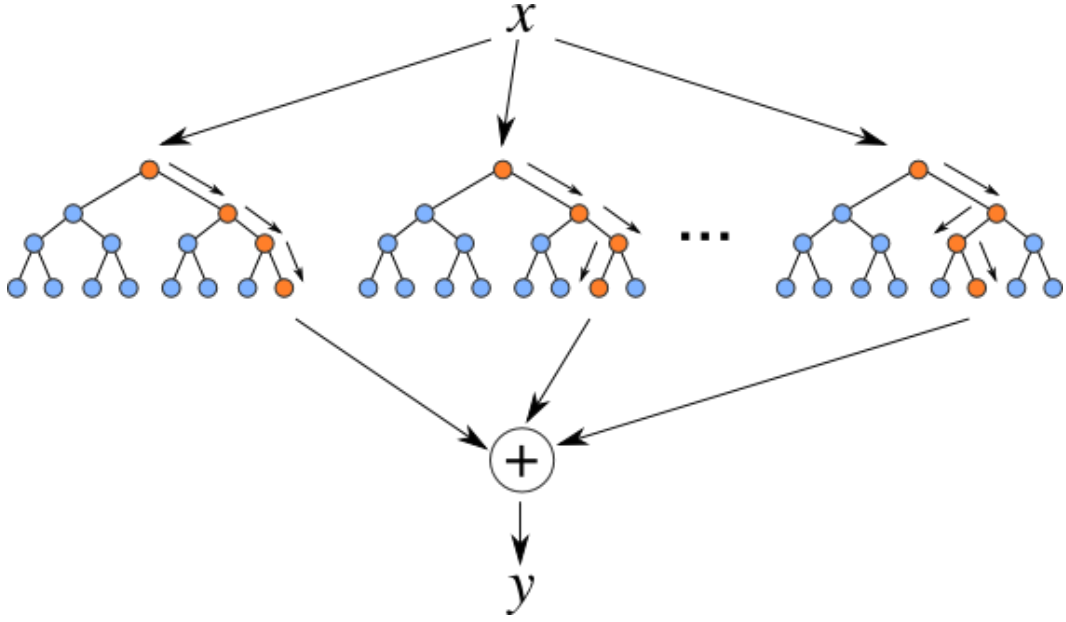
2.2.1.3 Rastgele Orman Algoritması

Rastgele ormanlar, her biri rastgele bir eğitim kümesinin alt kümesini veya girdi özelliklerinin rastgele bir alt kümesini kullanan bir değil birden çok karar ağacının eğitilip sonuçlarının her birinin bir oy hakkı olacak şekilde sınıflandırma işlemini gerçekleştirmesidir [25]. Karar ağaçlarının bir kombinasyonudur. Rastgele orman algoritmasının genelleme hatası, ormandaki tek tek ağaçlara ve bu ağaçlar aralarındaki korelasyona bağlıdır.

Rastgele Orman algoritması Leo Breiman [25] tarafından Amit ve Geman'ın (1997), yazılı karakter tanıma üzerine çok sayıda geometrik özellik tanımladığı ve her bir düğümdeki en iyi bölünme için bunların rastgele bir seçimini araştırdıkları önemli bir makaledeki çalışmalarından esinlenilerek geliştirilmiştir [25].

Karar ağaçlarına ait tahminler birleştirildiğinde genel doğruluk önemli ölçüde artırılabilir. Her bir karar ağacına ait veri seti de azalmış olduğu için öğrenme süresinin kısalıyor olması bir diğer avantajdır [14]. Çoklu bağımsız sınıflandırma ve regresyon ağacı yapılarından oluşmaktadır. Tek bir karar ağacının neden olacağı hatayı azaltmak için kullanılmaktadır.

Yapıda bulunan her bir karar ağacı eğitim setine ait bir alt küme tarafından eğitilerek bir



Şekil 2.2.4: Rastgele Orman Algoritması [26]

sınıflandırma işlemi gerçekleştirmektedir ve son sınıflandırma işlemi tüm karar ağaçları arasında en çok oyu olan çıktıya göre yapılmaktadır. Rastgele orman global optimum değere ulaşırken tek bir karar ağacı yerel optimum ile sınırlı kalabilir [14].

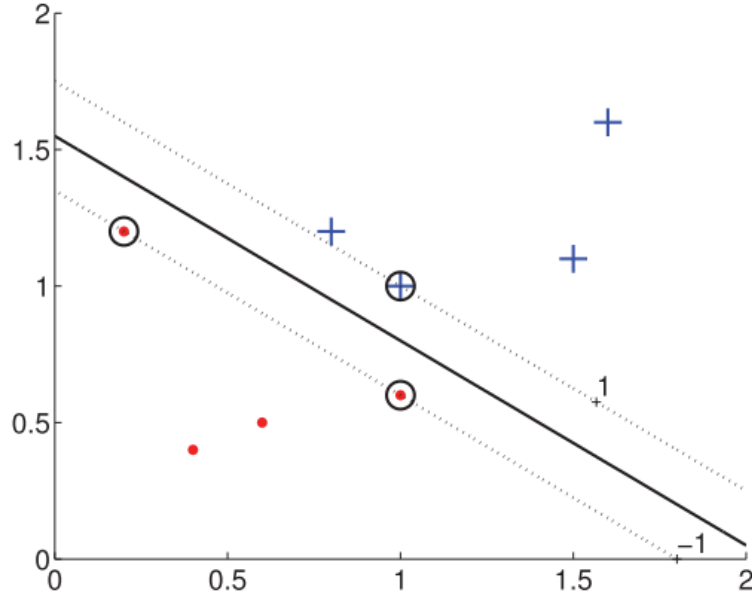
2.2.1.4 Destek Vektör Makineleri Algoritması

Destek vektör makineleri ikili sınıflandırma için çok fazla tercih edilen özellikle çekirdek fonksiyon aracılığıyla yüksek boyutlu uzaya geçiş sağlanarak doğrusal olmayan sınıfların ayırımında da kullanılan başarılı bir yöntemdir. SVM sınıflandırmayı iki sınıfı ayıran optimum hiperdüzlemi bularak optimizasyon problemine çözüm bulacak şekilde gerçekleştirmektedir [20].

Destek vektör makineleri için iki yaklaşım bulunmaktadır. Sorunun doğasına bağlı olarak doğrusal ve doğrusal olmayan çekirdekler kullanılabilir. Genel olarak, SVM'ler veri sınıflarını bir hiper düzlem kullanarak sınıflar arasındaki en büyük uzaklığa göre ayıran bir sınıflandırma işlevi bulmaktadır. Uygulamadan bağımsız olarak tüm makine öğrenme algoritmalarında ayıran temel fark, en uygun ayırma alt düzleminin hesaplanmasında yer alan matematiksel işlemlerdir. Optimal hiper düzleme yakın veri noktaları “destek vektörleri” olarak adlandırılmaktadır [20].

İlk olarak doğrusal olarak ayrılabilen iki sınıfı ayırmada kullanılmıştır. $g(x) = \mathbf{w}^T \mathbf{x} + w_0$ denklemi doğrusal bir hiperdüzlemi temsil etmektedir. İki sınıfı birbirinden ayıran bir düz-

lemde yeni bir noktanın nereye ait olduğu eşitlikte yerine konulduğunda $g(x) > 0$ için bir sınıfa $g(x) < 0$ ise diğer sınıfa ait olduğu çıkarımı elde edilmektedir [14].



Şekil 2.2.5: Doğrusal Ayrılabilen İki Sınıf [14]

Destek vektör makinelerinde destek vektör olarak seçilmiş noktalar karar sınırını belirlemektedir. Sınıflar arası ayırımı sağlayan destek vektörler, aralarındaki uzaklık maksimum olacak şekilde seçilmektedir [20].

2.2.1.4.1 Çekirdek Fonksiyonları

Doğrusal ayrılmayan sınıflar için, doğrusal olmayan bir model bulmaya çalışmak yerine, uygun şekilde seçilmiş temel fonksiyonları kullanarak çekirdek fonksiyonlar üzerinden bir dönüşüm yaparak problemi doğrusal bir probleme dönüştürüp yeni bir uzaya taşımaktadır. Çekirdek fonksiyonlar sayesinde verileri yeni bir uzaya direkt taşımadan daha kolay bir şekilde çözüme ulaştırmaktadır [14].

$$g(x) = w^T \phi(x) = \sum_t \alpha^t r^t K(\mathbf{x}^t, \mathbf{x}) \quad (2.2.2)$$

\mathbf{x} veri kümesi olmak üzere en çok tercih edilen $K(\mathbf{x}^t, \mathbf{x})$ çekirdek fonksiyonları aşağıda belirtilmiştir:

Radyal Tabanlı Çekirdek Fonksiyonu

\mathbf{x}^t merkezi, s 'de kullanıcı tarafından sağlanan yarıçapı temsil etmekte olup Eşitlik 2.2.3 ile ifade edilmiştir.

$$K(\mathbf{x}^t, \mathbf{x}) = \left(e^{-\frac{\|\mathbf{x}^t - \mathbf{x}\|^2}{2s^2}} \right) \quad (2.2.3)$$

Parabolik bir çekirdek fonksiyon yerine radyal tabanlı fonksiyon kullanılması hem işlemci üzerinde oluşacak yükü azaltmakta hem de herhangi bir uzayda veriyi doğru bir şekilde ayıracak bir hiper-uzayın olup olmadığını daha hızlı neticelendirmektedir.

Polinom Çekirdek Fonksiyonu

q dereceden bir polinom olarak $K(\mathbf{x}^t, \mathbf{x}) = (\mathbf{x}^T \mathbf{x}^t + 1)^q$ ifade edilen bir çekirdek fonksiyondur.

2.2.1.4.2 Çok sınıflı sınıflandırma

Birden çok sınıfın olduğu sınıflandırma problemlerinde destek vektör makineleri yöntemi ile sınıflandırma yapılması isteniyorsa sınıflar her biri birbirleri ile ikişerli ikişerli gruplandırılarak iki sınıflı sınıflandırma problemiymiş gibi sınıflandırma işlemi yapılmaktadır. Bu yönteme bire bir yöntemi denir. Çoklu sınıflandırma sorunu ikili sınıflandırma sorununa çevrilmektedir. Sonrasında bir oylama stratejisi uygulanmaktadır [14].

Çok sınıflı sınıflandırma yöntemlerinden biri de bir sınıfa karşı diğer tüm sınıfların olduğu yöntemdir. Sınıf sayısı kadar destek vektör makinesi modeli oluşturulmaktadır. Sınıflandırma da belli bir sınıfa ait veriler pozitif etikete sahip iken kalan diğer tüm verilere ait etiketler ise negatif olarak etiketlenmektedir. Bir verinin hangi sınıfa ait olduğu bilgisi karar fonksiyonda yerine konularak en büyük değerine sahip sınıfa sahip ise o sınıfa ait olmaktadır [27].

2.2.2. Sinir Ağları Algoritmaları

İnsan beyninden esinlenerek tasarlanan sinir ağları sınıflandırma ve regresyon işlemleri için kullanılmaktadır. Basit bir sinir ağı bir girdi, bir gizli katman, bir çıktı ve katmanlarda bulunan nöronlardan oluşmaktadır. Nöron sayısı öznitelik sayısına, sınıf sayısına ve gizli katman sayısına göre değişebilmektedir. Doğrusal olmayan aktivasyon fonksiyonları, ağırlıklar ve bayesler ile bir katmandaki tüm nöronlar birbirleri ile ilintili durumdadır. Derin öğrenmenin

de temeli olan gizli katmandaki nöron sayısının artması kimi zaman öğrenme kapasitesini artırırken kimi zaman azaltmaktadır. Bu performans temelde problemin cinsine bağlıdır. Sınıflandırma ya da öğrenme problemine bağlı olarak en optimum gizli katman sayısı belirlenmelidir [28].

Sinir ağları uygulamakta ki temel amaç, öğrenme sistemini uyarlamak için bir dizi girdi verisi ve istenen sistem yanıtlarının kullanılmasıdır [13]. Bu sistem yanıtlarının başarımlarının artırılması için ağırlıkları güncellenmiş birçok iterasyon yürütülebilmektedir. Öğrenme sisteminin çıktısı ile bilgi uzmanından istenen yanıt arasındaki hata ölçülmektedir. Hata sinyali daha sonra öğrenme sisteminin yanıtını değiştirmek için kullanılmaktadır, ağırlıkları sinir ağları için uyarlar, böylece yanıt bilgi uzmanının yanıtıyla daha yakından eşleşmektedir [13].

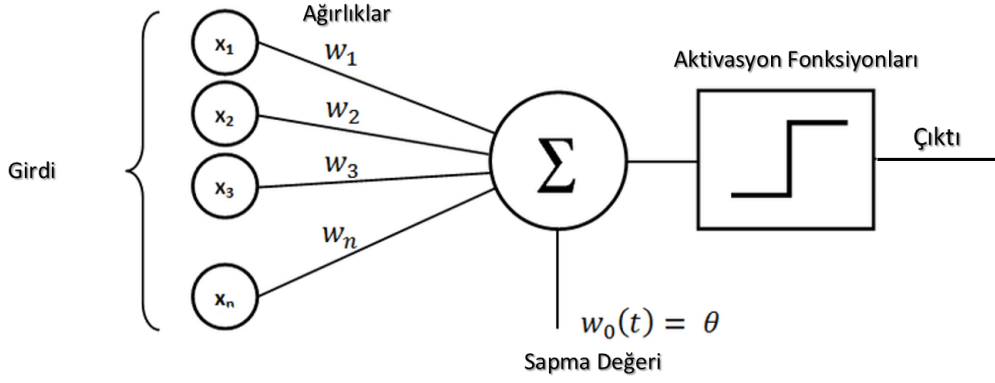
Geri yayılım algoritmaları ile her bir iterasyonda güncellenen parametreler sayesinde hata oranı azaltılmış bir eğitime işleminin elde edilmesi amaçlanmaktadır [18]. Örüntü tanıma, sınıflandırma, optimizasyon, kümeleme gibi problemlerde sinir ağları çözümleri etkin sonuçlar sunabilmektedir.

Sinir ağları çözümünde her nöron kendisine verilen görevi yürütmek ve arzulanan çıktıyı vermek için tasarlanmaktadır. Nöronların içerisinde gerçekleştirilen ve çıktının belirli bir değer arasında sıkıştırılmasını amaçlayan fonksiyonlara, aktivasyon fonksiyonları denilmektedir. En yaygın aktivasyon fonksiyonları lojistik sigmoid, hiperbolik tanjant, Gauss ve doğrusal aktivasyon fonksiyonudur. Bu fonksiyonların girdi üzerinde yarattıkları etki her birine özgüdür ve çıktının başarımlarının performansının gereksinimlerine bağlı olarak tasarımcı tarafından özellikle seçilmektedir. Sınıflandırma problemleri için doğrusal olmayan aktivasyon fonksiyonları daha iyi sonuçlar üretmektedir [13].

2.2.2.1 Algılayıcı

Algılayıcı en basit tek katmanlı yapay sinir ağı modelidir. Temel olarak eğitilebilecek tek bir yapay sinir hücresinden oluşmaktadır. 1957'de Cornell Havacılık Laboratuvarı'nda Amerikalı psikolog Frank Rosenblatt tarafından ortaya atılmıştır. Warren McCulloch ve Walter Pitts'in çalışmalarından (biyolojik nöron ve öğrenme yeteneği) etkilenen Rosenblatt, bir nöron gibi davranan fiziksel bir makine üretmeyi amaçlamıştır [29].

Algılayıcı denetimli bir öğrenme algoritmasıdır. Yani ağa hem giriş hem de çıkış değerleri



Şekil 2.2.6: Algılayıcı

verilir ve sistemin öğrenmesi beklenir. Algılayıcının öğrenme algoritması “hata” üzerine dayandırılmış bir öğrenme metodolojisidir [29]. Algılayıcı sistemdeki hataları öğrene öğrene, en az hataya indirgemeyi hedeflemektedir.

Bir algılayıcı girdileri, eşik değeri, ağırlık parametreleri, ağırlıklı toplama fonksiyonu, aktivasyon fonksiyonu ve çıkış değerinden oluşmaktadır. Toplama fonksiyonunda her giriş kendi ağırlığı ile çarpılır ve bu değerler toplanır. Toplanan değer aktivasyon fonksiyonuna girdi olmaktadır ve bu işlem sonucunda oluşan çıkış, aktivasyon fonksiyonu tarafından belirlenen değerdir.

2.2.2.2 Çok Katmanlı Algılayıcı

Tek katmanlı algılayıcılarda yalnızca doğrusal olarak ayrılabilen problemler ve sınıflandırma işlemleri yapılmaktadır. Örnek olarak VE kapısı gibi bir problemin girdisine bağlı çıktıları sınıflandırmak istediğimizde tek katmanlı algılayıcılar kullanmak yeterli olacaktır. Ancak gerçek hayattaki problemlerin çoğu doğrusal ayrılabilir problemler değildir. Bu kısıt tek katmanlı algılayıcıların en önemli zayıflıklarından biridir. Doğrusal ayrılamayan problemlerin çözümü için basit ve tek katmanlı algılayıcılara başka bir ağırlık katmanı eklenerek çözüme erişmek hedeflenmiştir. Çok katmanlı algılayıcıların özü esasında bu yaklaşımdır. Eklenen yeni ağırlık katmanıyla doğrusal ikinci bir ayırıcı daha belirlenmiş olur [30]. Doğrusal tek bir ayırıcı ile ayrılamayan problemler bu iki ve daha çok doğrusal ayırıcı ile ayrılmaya çalışılmaktadır. YA DA kapısı problemi bu durum için önemli bir örnek teşkil etmektedir.

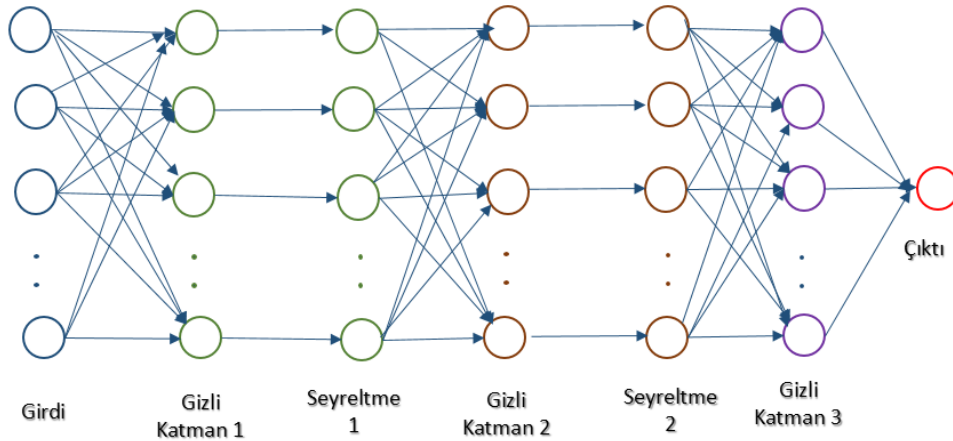
Doğrusal olmayan problemlerin çözümü için geliştirilen çok katmanlı algılayıcılarda eklenen orta ağırlık katmanlarında ne olduğunu ve tam anlamıyla hangi işlemin yapıldığını bilmek

gerekmektedir. Basit problemlerde bu orta katman için beklenen çıktılarının ne olması gerektiği belirlemek kolaydır ancak zor ve karmaşık problemlerde bunu tespit etmenin önemli zorlukları bulunmaktadır.

Karmaşık ve doğrusal olmayan problemlerin çözümünde çok katmanlı algılayıcılar kullanılırken orta katmanların ağırlıklarının öğrenmek için standart optimizasyon tekniklerinden faydalanılmaktadır. Tüm sinir ağının çıktısı üzerinden, yine çıktı cinsinden ifade edilen bir hata fonksiyonu tanımlanır. Gradyan iniş teknikleri de kullanılarak sürekli ve türevlenebilir bir hata fonksiyonu elde edilir. Ara katmanlardaki ağırlıkların tespit edilmesi çalışmalarında bu fonksiyondan faydalanılır [30].

2.2.2.3 Derin Öğrenme Algoritmaları

Çoklu yapay sinir ağlarından oluşmaktadır. Girdi katmanı, gizli katmanlar, çıktı katmanı bulunmaktadır. Çok sayıda gizli katmanlar aracılığıyla yüksek sayıda öznetelik ile öğrenme sağlanmaktadır. Aşırı öğrenme olasılığını indirgeme adına bu katmanlar arasında seyreltme yapılabilmektedir.



Şekil 2.2.7: Sinir Ağları Katmanları

2.2.2.3.1 Yinelenen Sinir Ağları Williams ve Zipser yinelenen sinir ağları algoritmalarını geliştirmişlerdir. [6] çalışmasına kadar bu algoritmalar radar hedefi tanıma amaçlı kullanılmamıştır. Bir girdi vektörü bir ağırlık matrisi ile çarpılması sonucu bir vektör elde edilmektedir. Ağırlık matrisi ise etiketlenmiş girdi vektörlerine ve gradyan inişe göre eğitilmektedir. Bu sayede istenilen sonuç ile oluşan sonuç arasındaki hata oranı minimuma indirilmektedir.

Sonuç vektöründe hem gizli düğümler hem de sonuç düğümleri bulunmaktadır. Yinelenebilir sinir ağlarının karmaşık öğrenme işlemleri en büyük dezavantajları arasındadır. Yinelenebilir sinir ağları sadece sıralı bir veri dizisine uygulanmaktadır. İstenilen ve ulaşılan çıktılar arasındaki hatalar her bir epokta kümülatif şekilde hesaplanmaktadır. Gerçek zamanlı yinelenebilir öğrenme algoritması, gradyan inişini kullanılarak istenen ve gerçek çıktı düğüm değerleri arasındaki ortalama kare hatasının en aza indirilmesi amaçlanmaktadır. [6] yinelenebilir sinir ağlarının karmaşık öğrenme işlemleri en büyük dezavantajları arasındadır.

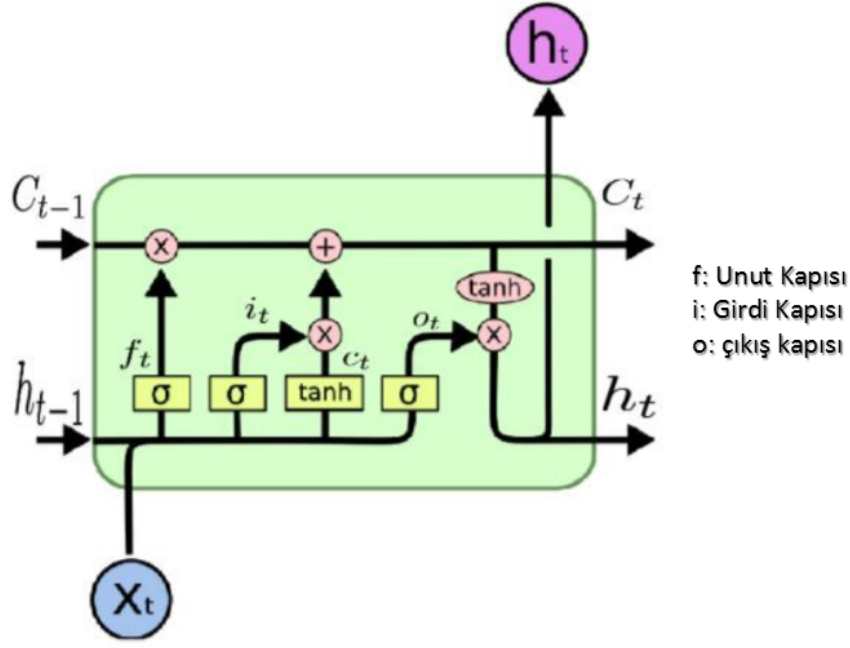
Geleneksel sinir ağları, yeni gelen verilerle öncekiler arasında bir ilişki olabileceğini saptayamazlar. Tekrarlayan sinir ağları bu sorunu gidermek üzere ortaya atılmış, bilginin kalıcı olmasını sağlayan döngüleri olan ağ yapılarıdır. Ara katmandaki bir çıktı, başka bir sinire girdi olabileceğinden nihai çıktının hata oranları her aşamada biraz daha azaltılmış olmaktadır.

2.2.2.3.2 Uzun Kısa Süreli Bellek Algoritması Uzun kısa süreli bellek ağı, bir tekrarlayan sinir ağı türüdür. Tekrarlayan sinir ağları, dizi problemleri için tasarlanmış özel bir sinir ağıdır. Tekrarlayan bağlantılar, ağa durum veya bellek ekleyerek giriş dizilerindeki gözlemlerin sıralı yapısını öğrenmesine ve kullanılmasına izin verir. LSTM ağları, bir zaman serisindeki önemli olaylar arasında bilinmeyen süreli gecikmeler olabileceğinden; zaman serisi verilerine dayalı olarak sınıflandırmak, işlemek ve tahminler yapmak için uygundur.

Uzun kısa süreli bellek ağı sabit boyutlu zaman pencereleri kullanan ileri beslemeli ağlar tarafından çözülemeyen birçok zaman serisi görevini çözmektedir. LSTM'lerin dâhili durumları vardır, girdilerdeki zamansal yapının açıkça farkındadırlar, birden fazla paralel girdi serisini ayrı ayrı modellemektedir ve her seferinde bir gözlem olmak üzere değişken uzunluklu çıktı dizileri üretmek için çeşitli uzunluktaki girdi dizilerinden geçmektedirler.

Bir tekrarlayan sinir ağı eğitiminin teknik sorunlarının, yani kaybolan gradyanların üstesinden gelir. Tekrarlayan sinir ağı eğitiminin bu hatasına çözüm bulabilmek amacıyla LSTM'ler Hochreiter ve Schmidhuber tarafından 1997'de geliştirilmiştir ve kümülatif katkılar sayesinde bugünkü halini almıştır [31].

LSTM ağı, klasik birçok katmanlı algılayıcıdan farklıdır. Birçok katmanlı algılayıcı gibi, LSTM ağı da nöron katmanlarından oluşmaktadır. Girdi verileri, bir tahmin yapmak için ağ üzerinden yayılmaktadır.



Şekil 2.2.8: LSTM Hücresi [33]

Tekrarlayan sinir ağlar gibi, LSTM'ler de tekrarlayan bağlantılara sahiptir, böylece nöronun önceki zaman adımından önceki aktivasyonlarından gelen durum, bir çıktı formüle etmek için bağlam olarak kullanılmaktadır. LSTM, diğer tekrarlayan sinir ağlarının aksine, eğitimi ve ölçeklenmesini engelleyen sorunlardan kaçınmasını sağlayan benzersiz bir formülasyona sahiptir [32].

LSTM ağının hesaplama birimine hafıza hücresi, hafıza bloğu veya kısaca hücre denilmektedir. Hesaplama birimi olarak nöron terimi, çok katmanlı algılayıcıları tanımlarken o kadar kökleşmiştir ki, genellikle LSTM bellek hücresine atıfta bulunmak için de kullanılmaktadır. LSTM hücreleri, ağırlıklardan ve kapılardan oluşmaktadır. Bir LSTM katmanı, bellek blokları olarak bilinen, tekrar tekrar bağlanan bir dizi bloktan oluşmaktadır. Bellek hücresi, giriş, çıkış ve ayrıca giriş zaman adımlarına maruz kalma yoluyla oluşturulan bir dâhili durum için ağırlık parametrelerine sahiptir [32]. LSTM, kapılar adı verilen yapılar tarafından dikkatlice düzenlenen hücre durumuna bilgi ekleme veya çıkarma yeteneğine sahiptir.

Bir LSTM ünitesi, bir hücre, bir giriş kapısı, bir çıkış kapısı ve bir unut kapısından oluşmaktadır [33]. Giriş, çıkış ve unut adlı üç kapı, isteğe bağlı olarak bilginin geçmesine izin vermeyi sağlamaktadır. Klasik yinelenen sinir ağlarındaki sıralı gelen verinin kapsamlı şekilde yorumlanmasını sağlayarak sonraki aşama için benzer bir çıktı üreten mekanizmasının

aksine; LSTM'ler sahip olduđu unutma kapısı sayesinde yeni gelen verileri rastgele zaman aralıklarıyla öğrenerek önceki verileri unutması engellenmektedir [31].

Giriş dizileriyle uzun vadeli geçici bağımlılık sorunlarının üstesinden gelmek için belleğe sahiptir. Değişken uzunlukta giriş ve çıkışlara izin vererek, giriş sıralarını ve çıkış sıralarını zaman adım adım işler.

LSTM uygulamaları, bir dizi karmaşık dizi tahmin probleminde etkileyici sonuçlar elde etmektedir. Ancak LSTM'ler tüm dizi tahmin problemleri için ideal olmayabilir. Örneğin, zaman serisi tahmininde, genellikle bir tahmin yapmakla ilgili bilgiler, geçmiş gözlemlerin küçük bir penceresi içindedir. Genellikle bir pencereli veya doğrusal modellenmiş birçok katmanlı algılayıcı daha az karmaşık ve daha uygun bir model olabilir [32].

LSTM'lerin önemli bir sınırlaması bellektir. Bir LSTM modelini, belirli bir zaman adımında çok uzun sayıda giriş tek bir gözlemi hatırlamaya zorlamak mümkündür. Bu, LSTM'lerin kötü bir kullanımudur ve birden fazla gözlemi hatırlamak için bir LSTM modeli gerektirmesi başarısız olacaktır [31].

2.2.3. Öznitelik Çıkarımı

Sınıflandırma işlemleri yapılırken karmaşıklığın azaltılması için öznitelik çıkarım yöntemleri kullanılmaktadır. Sinyal işleme sonrası uygulanan öznitelik çıkarımı ile hedeften yansıyan sinyalin daha az ve daha kritik parametrelerle ifade edilmesi hedeflenmektedir.

Öznitelik seçimi, başarılı bir sinir ağı geliştirmenin anahtarıdır. Öznitelik sayısı az ve örnek veri sayısı fazla olduğunda, eğitim sırasında her bir özelliğin sinir ağının oluşumuna katkısı farklı olabilmektedir. Öte yandan, çok sayıda özellik ve az sayıda örnek varsa, sinir ağı genellenemeyecek bir model üretebilir ve düşük performansla sonuçlanabilmektedir [13].

Özniteliklerin veya boyutların sayısı arttıkça bilgi miktarı da artar. Çok sayıda özneliğe sahip olmak tercih edilebilir gibi görünse de daha az özneliğe sahip ağa göre kötü performans göstermesi mümkündür. Ek olarak, öznitelikler eklendikçe, sinir ağının verileri ezberlemesini önlemek için daha çeşitli veriye ihtiyaç duyulmaktadır. Bu genellikle "boyutluluğun laneti" olarak adlandırılır [13].

Sinir ağına yapılan her ek girdi, modele başka bir boyut ekler. Eğitim sırasında, girdi uzayını çıktı uzayına uygun şekilde eşlemesi için sinir ağını eğitmek için girdi uzayının her parçası-

nın veri temsilcisine ihtiyaç vardır. Eşleme ilişkisini temsil etmek için alanı yeterince yoğun bir şekilde doldurmak için yeterli veri olmalıdır, bu nedenle daha yüksek boyutlu bir alanı temsil etmek için daha düşük boyutlu bir uzaydan önemli ölçüde daha fazla örnek gerekmektedir [13].

Girdilere ait boyutu ve veri sayısını artırmak, zaman, bellek ve veri dâhil olmak üzere kaynakları tüketmektedir. Neyse ki, bu boyutluluk laneti, özelliklerin uygun seçimi ve azaltılmasıyla iyileştirilebilir. Belirgin özellikleri bilmek, sinir ağı tasarımcısının, zayıf özellikleri ortadan kaldırarak girdi veri boyutlarının azaltılmasına olanak tanımaktadır [13].

Veri kümesindeki özellikler, öğrenilecek olan girdi ve çıktı uzayı arasındaki ilişkiyle ilgisi olmayan girdileri içerebilir. Özellik vektörleri, birbiriyle ilişkili, gereksiz bilgileri temsil eden özellikler içerebilmektedir. Öznitelik boyutu bu gibi durumlarda azaltılmaktadır. Bu özellik seçimi, özellik çıkarma veya ikisinin bir kombinasyonu yoluyla yapılmaktadır [13].

Doğru uygulandığında, öznitelik seçimi ve çıkarımı, öznitelik sayısını azaltırken eğitim için gerekli bilgileri korumaktadır. Bununla birlikte, özellik azaltmayı körü körüne uygulamak, düşük performansa yol açabilir ve daha az sayıda girdi üretmek için bir dizi özelliği seçme ve çıkarma süreci, uygulamaya özel alan bilgisi gerektirebilir [13].

Öznitelik çıkarım yapılmasının bir başka nedeni de özellik fazlalığını azaltmak veya ortadan kaldırmaktır. Bir hava durumu uygulamasında sıcaklık, çiy noktası ve nem gibi ölçümler kaydedilir. Nem, sıcaklık ve çiy noktası ile ilişkilidir ve matematiksel olarak ikisinden türetilmektedir. Bu nedenle, bilgi kaybı olmadan nemi hariç tutarak sahip olunan özniteliklerin sayısını üçten ikiye indirilmektedir [13].

Sınıflandırma yapılırken karmaşıklığın azaltılması için öznitelik çıkarım yöntemleri kullanılmaktadır. Gürültüyü yok etme, FFT, DTFT, STFT, Wavelet gibi sinyallere uygulanan işlemlerin sonucunda elde edilen izgeler ile öznitelik çıkarımı gerçekleştirilmektedir. Bu çıkarım sonucu elde edilen yapı kullanılan makine öğrenmesi algoritmalarına girdi sağlamaktadır [34].

2.2.4. Sınıflandırma Ön İşlemleri

Sınıflandırma, adından da anlaşılacağı gibi, bir dizi veriyi sınıflandırmayı amaçlamaktadır. Bilinmeyen bir nesneye doğru sınıf ataması için sınıflandırıcı, özellikler tarafından tanımla-

nan ve nitelikleri adı verilen ve kendi sınıflarının bilindiği nesnelere kurallar çıkarmaktır. Yapay zekâ algoritmaları ile sınıflandırma işlemi yapılmadan önce çeşitli adımlar gerçekleştirilmektedir.

2.2.4.1 Veri Normalizasyonu

İdeal olarak bir sinir ağı içinde her bir öznitelik için aynı değer aralığını içerisinde olması için normalizasyon yöntemleri uygulanmaktadır. Bu şekilde özniteliklerden birine karşı bir taraflılık oluşumu engellenmekte ve eğitim süresi kısaltılmaktadır. Çok geniş aralıkta öznitelik içeren veri setleri için normalizasyon uygulaması çok etkili olmaktadır.

En çok kullanılan öznitelik ölçeklendirme yönteminde değerler genelde [0-1] aralığına ya da [-1,1] aralığına taşınmaktadır. σ standard sapma, μ ortalama değer olmak üzere x değeri Eşitlik 2.2.4'de görüldüğü gibi normalize edilir.

$$z = \frac{x - \mu}{\sigma} \quad (2.2.4)$$

2.2.4.2 Eğitim - Doğrulama - Test Verisi

İstatistik terminolojisinde eğitim seti modelleri eğitim için kullanılır, doğrulama seti ise model seçimi ve tahmin hatasını tahmin etmek için kullanılır. Test seti, seçilen nihai modelin genelleme hatasını belirlemek için kullanılır [20].

Makine öğrenmesi alanındaki ana problemlerden biri sınıflandırmadır. Sınıflandırma sürecinde, mevcut veriler iki alt gruba ayrılmaktadır; öğrenme seti ve test seti. Öğrenme seti, kuralları türeten sınıflandırma modelini oluşturmak için test seti ise bu modelin performansını değerlendirmek için kullanılmaktadır. Her bir sınıflandırıcı, öğrenme setinden bağımsız bir test seti kullanılarak değerlendirilmektedir, dolayısıyla genelleme hatası en aza indirgenmektedir.

Ağ tasarımcısı önce problemi tanımlamaktadır, sinir ağı mimarisi seçildikten sonra veriler toplanmaktadır. Veriler ön işlemlerden geçirilip etiketlenerek sinir ağını eğitmeye hazır hale getirilmektedir. Eğitimin amacı, sinir ağının bilinmeyen verilerle performansına göre en iyi performansla sonuçlanan eğitim parametrelerini bulmaktır. Bu, ağın ne kadar iyi genellebileceğini belirtmektedir. Genelleme, sınıflandırıcının daha önce hiç sunulmadığı ancak giriş özellik uzayının kabul edilebilir sınırları içinde olan veri örnekleri üzerinde ne kadar

iyi performans gösterdiğinin bir ölçüsüdür. Genelleme, sınıflandırıcının giriş verilerini ezberleyip ezberlemediğini belirlemek için kullanılmaktadır. Bir ağ aşırı eğitilmişse, genellikle sınıflandırma görevlerindeki verileri ezberlemektedir veya tahmin görevleri için kullanıldığında verileri aşırı doldurmaktadır. Amaç, bağımsız veriler üzerinde en iyi performansa sahip konfigürasyonu bulmak olduğundan, sinir ağını eğitmek için kullanılmayan test verilerini kullanmak çok önemlidir [13].

Optimum sinir ağı konfigürasyonunu bulmak için ideal bir yaklaşım, üç bağımsız veri seti üretmek için popülasyonu üç kez rastgele örnekleme: bir eğitim seti, bir doğrulama seti ve bir test seti [13].

Eğitim seti, istenen sonucu elde etmek için sinir ağındaki ağırlıkları ayarlamak veya eğitmek için kullanılan bir dizi örnektir. Doğrulama seti (bazen makine öğrenimi sözlüğünde test seti olarak adlandırılır), en iyi sinir ağı konfigürasyonunu ve eğitim parametrelerini bulmak için kullanılan bir dizi örnektir. Örneğin, eğitim yinelemelerinin veya dönemlerinin optimal sayısını belirlemek için eğitim sırasında ağ hatasını izlemek için kullanılmaktadır. Optimum gizli nöron sayısını belirlemek için ve birden fazla eğitilmiş ağ arasında seçim yapmak için kullanılmaktadır. Doğrulama seti, eğitimi durdurmak için kullanıldığında, sinir ağı, verilere maruz kaldığı için iyimser bir şekilde önyargılıdır [13].

Test seti yalnızca tam olarak eğitilmiş sinir ağını değerlendirmek için kullanılan bir dizi örnektir. Genellikle, bağımsızlığın sağlanmasına yardımcı olmak için eğitim ve doğrulama setlerinden ayrı olarak toplanmaktadır [13].

Sinir ağı hem eğitim hem de doğrulama kümelerine karşı önyargılıdır, bu nedenle genelleme hatasını belirlemek için bağımsız test kümesi kullanılmaktadır. Test seti, ağın genelleme hatasının tarafsız bir tahmini olarak kalması için, sinir ağıları arasında seçim yapmak için asla kullanılmamalıdır [13].

Ayrıca, eğitimin durdurulması için optimum noktayı belirlemek için doğrulama seti hatası da sıklıkla izlenmektedir. Normalde doğrulama setindeki hata da ilk eğitim aşamasında azalmaktadır. Bununla birlikte, ağ veriyi fazla öğrenmeye başladığında, doğrulama seti tarafından üretilen çıktı hatası tipik olarak yükselmeye başlamaktadır [13].

Doğrulama hatasının azalıp artmaya başladığı optimum öğrenme anında eğitim durdurulur minimum doğrulama hatasında üretilen ağırlıklar sinir ağında işlem için kullanılmaktadır

[13]. Bu yaklaşım en iyi genellemeyi vermektedir. Bu noktanın ötesinde oluşturulan ağırlıkların, eğitim verilerinin kendine özgü özelliklerine uyması daha olasıdır ve iyi bir şekilde enterpolasyon veya genelleme yapmamaktadır [13].

2.2.4.3 Hiperparametre Eniyilemesi

Yapay sinir ağları ya da derin öğrenme modelinde yapılan temel işlem, modelin en iyi öğrenmeyi gerçekleştireceği parametrelerinin hesabını doğru yapabilmektir.

Hiperparametreler, bir makine öğrenmesi modelinin belirli bir veri seti ve hedefleri için özelleşmesini mümkün kılan noktalardır. Veri setinin özniteliklerine bağlı olarak belirlenen aralıklarda farklı hiperparametre analizlerini incelenerek öğrenme sürecini en doğru şekilde ortaya çıkaran hiperparametre noktaları belirlenebilmektedir. Bu belirleme sürecinde birden çok hiperparametre çıkarma ya da optimize etme yöntemi bulunmaktadır. Optimizasyonun temel amacı, maksimum doğruluk ve minimum hata ile modelin en iyi performansını sağlayan bir sınıflandırıcı vektörün çıkarımıdır. Bu çıkarım yöntemlerinden birkaçı Izgara Arama, Rastgele Arama, Çapraz Doğrulama gibi yöntemlerdir. Bu yöntemlerden biri ya da birkaçı test edilerek en iyi performansı sağlayan bir dizi hiperparametre aranabilir.

2.2.4.3.1 Izgara Arama Yöntemi

Izgara arama yöntemi, optimal hiperparametrelerin çıkarılması ve optimizasyonu sürecinde kullanılacak en doğru analiz yöntemlerinden biri olarak değerlendirilmektedir. En güncel uygulamalarda ızgara arama, çapraz doğrulama yönteminin sıklıkla tercih edildiği görülmektedir. Birçok farklı optimizasyon problemi için uygulanabilir bir yöntem olması nedeniyle en çok tercih edilen optimizasyon algoritmalarından biridir. Standart ve belirli bir “nokta tarama” mantığına göre çalışmasına rağmen çıktı olarak sunduğu hiperparametre kümesi, öğrenme performansı üzerinde genellikle pozitif etki etmektedir [35].

Bu yöntemlerden hangisinin tercih edilmesi gerekliliği, veri setinin karakteriyle ve öznitelikleriyle doğrudan ilişkilendirilmektedir. Izgara arama yöntemi, sınıflandırılması sürecinde sezgisel çözümlerin geliştirilmesinin mümkün olduğu veri setleri için en iyi yöntem olarak değerlendirilirken rastgele arama yöntemi, tahmin etmenin mümkün olmadığı hiperparametre noktalarının keşfedilmesini sağlayabilecek bir yöntem olarak değerlendirilmektedir [36]. Ancak rastgele arama yöntemi var olan her ihtimal için bir deneme yapacağından, işlem süresi çok uzun olabilmektedir.

2.2.4.3.2 Çapraz Doğrulama

Çapraz doğrulama yöntemi ise temel olarak veri seti üzerinde yapılan istatistiksel analizlerin doğruluğunu çapraz şekilde ölçmeyi amaçlayan bir yöntemdir. Veri setinin bir kısmı eğitim verileri olarak ayrılırken bir kısmı test ve doğrulama verileri olarak ayrılmaktadır. Test ve doğrulama seti ölçülen öğrenme performansı veri seti içinden tekrar tekrar belirlenen test ve doğrulama setleriyle tekrarlanarak, öğrenme performansında bir önyargının varlığı tespit edilmektedir.

2.2.5. Sınıf Dengesi Sağlama Yöntemleri

Denetimli yaklaşımlar için veriler etiketlenmektedir. Bu, sinir ağı tasarımcısının veya bir modelin toplanan her örneğe hedef değerler atamasını gerektirmektedir. Etiket zaten bir sayı değilse, sinir ağının bilgisayar aracılığıyla eğitilmesi için sayısal bir forma dönüştürülmesi gerekmektedir. Veriler belirli bir sınıfa ait olacak şekilde etiketlendiğinde bir sınıfa ait eğitim setindeki örneklerin diğer sınıftaki örneklerden çok daha fazla olduğu durumlar sınıflandırma performansını etkileyebilecek sınıf dengesizliği konusudur. Gerçek dünya verilerinde karşılaşılabilecek bu durum sonucunda algoritma azınlık sınıfı ile ilgili kavramı öğrenmekte zorluk çekmektedir ve bu durum sınıfları ayırt etme de başarısızlığa neden olmaktadır.

Dengesiz veri kümeleri, özellikle belirli bir sınıfın çok az verisi olduğunda ve azınlık sınıfının belirli önemli yönlerinin tam olarak temsil edilmemesi gibi sorunlarla karşılaşıldığında bir sorundur. Genel olarak, dengesiz veri kümeleri sorunu, bir sınıf sınırlandırılmış bir kavramı temsil ederken, diğer sınıf bu kavramın karşılığını temsil ettiğinde ortaya çıkmaktadır, böylece karşı sınıftan örnekler, pozitif sınıftan örneklerden çok daha fazla olmaktadır [37].

Dengesiz veri kümelerinde makine öğrenmesi algoritmaları yetersiz sınıflandırıcılar üretmektedir. Verilerin çoğunluğunun belirli bir sınıftan olması, bir öğrenme algoritmasının belirli bir sınıfa eğilimli olacak şekilde öğrenmesine neden olmaktadır. Bu durumda çok veri setine sahip verinin başarısının sorgulanmasına ve sınıflandırıcının veri ayırımı konusunda başarısız olmasına yol açmaktadır [37].

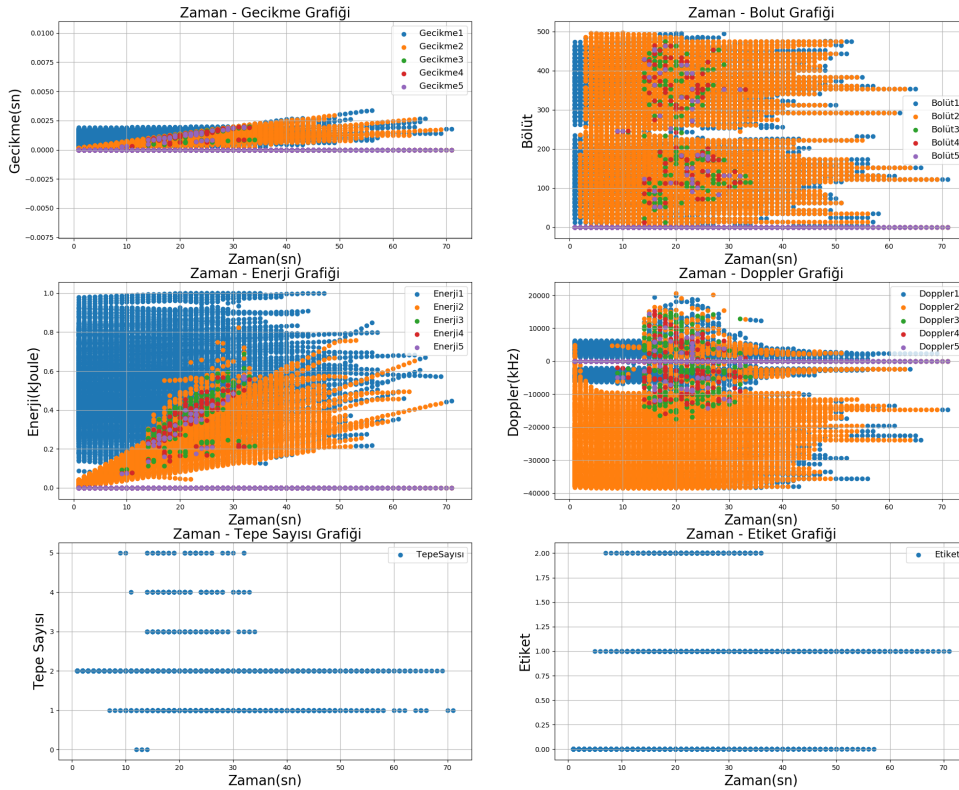
Her şeyi çoğunluk sınıfıyla etiketlenen veri seti ile yaptığı bir eğitim ve test işlemlerinde adil bir öğrenme işlemi gerçekleştirmesi zor olabilmektedir. Sınıf dengesini ayarlamak ya azınlık sınıfları çoğaltmakla ya da çoğunluk sınıflara ait verinin bir kısmını atmakla mümkün olmaktadır. Sonuç olarak, dengesiz verilerle problemler incelenirken, standart makine öğre-

nimi algoritmaları tarafından üretilen sınıflandırıcıları çıktı eşikini ayarlamadan kullanmak kritik bir hata olabilmektedir.

Veri setleri içerisinde bir ya da birkaç sınıftan çok sayıda veri varken bir ya da birkaç sınıfta çok daha az sayıda verinin varlığı durumunda ciddi bir önyargı sorunu ortaya çıkmaktadır. Bu durumlarda oraya çıkan hiperparametreler ya da sınıf ayrıcılar, bu azınlık gruplarına karşı ciddi anlamda önyargılıdır. Bu durum, sistemde ciddi bir performans eksikliğine neden olmaktadır. Sayıca yüksek olan sınıfa karşı ön yargı probleminin aşılabilmesi amacıyla “çoğunluk sınıfına karşı eksik örnekleme tekniği”, “azınlık sınıfı üzerinden aşırı örnekleme tekniği”, “maliyete duyarlı öğrenme teknikleri” ve “öznelik seçme/eleme” gibi yöntemler geliştirilmektedir [37].

Bir veri setine ait dengesiz sınıf ve veri dağılımı Şekil 2.2.9’da verilmektedir.

Noksan Öznitelik Seti Dağılımı

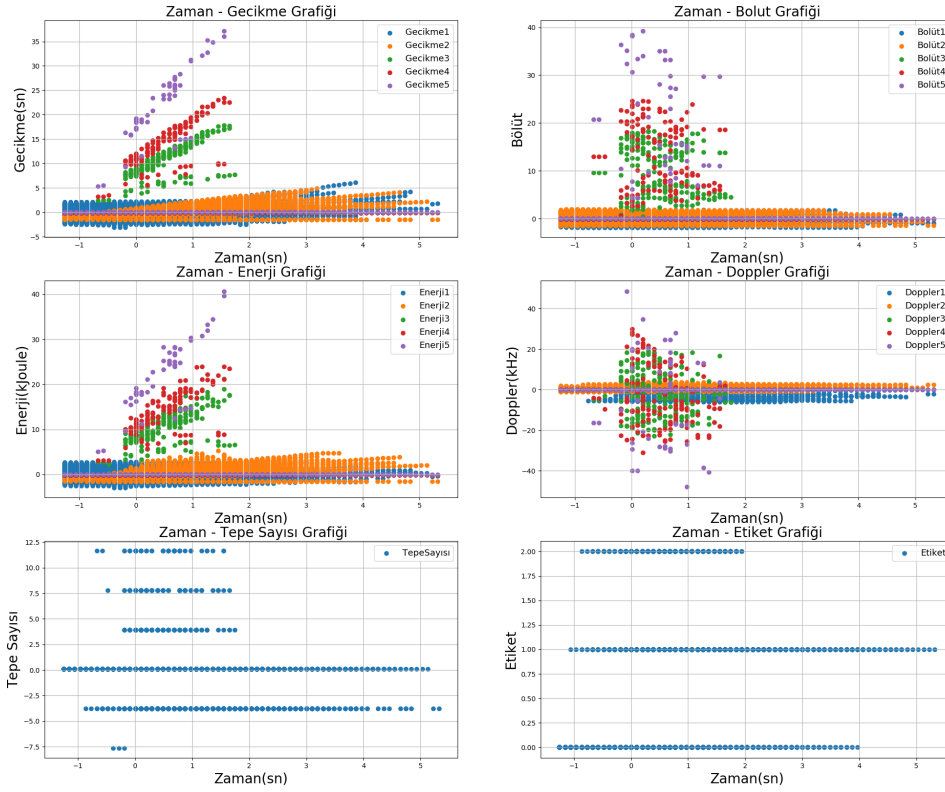


Şekil 2.2.9: Dengesiz Öznitelik Seti Grafiği

Aynı veri setinin normalize edilmiş veri dağılımı Şekil 2.2.10’da verilmektedir. Sınıf den-

gesini sağlamak için kullanılan örnekleme yöntemlerinden; sentetik azınlık aşırı örnekleme yöntemi, rastgele aşırı örnekleme yöntemi, rastgele alt örnekleme, sentetik azınlık aşırı örnekleme - düzenlenmiş en yakın komşu yöntemi, sentetik azınlık aşırı örnekleme - Tomek örnekleme yöntemi ve yakındakini atma yöntemini incelenecektir.

Sınıf Dengesizliği Veri Dağılımı



Şekil 2.2.10: Dengesiz Normalize Edilmiş Öznitelik Seti Grafiği

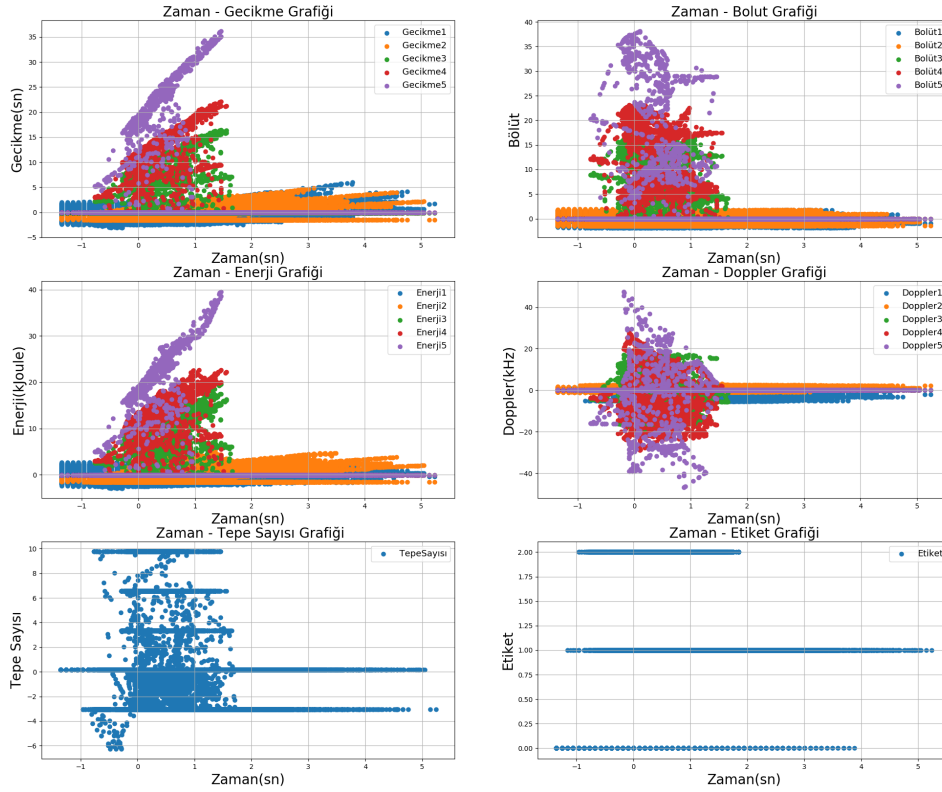
2.2.5.1 Sentetik Azınlık Aşırı Örnekleme Yöntemi

Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE) bir aşırı örnekleme yöntemidir. Ana fikri, bir arada bulunan birkaç azınlık sınıfı örneği arasında enterpolasyon yaparak yeni azınlık sınıfı örnekleri oluşturmaktır. Böylece, aşırı uyum sorunu önlenir ve azınlık sınıfı için karar sınırlarının çoğunluk sınıf alanına daha da yayılmasına neden olmaktadır [38].

Bu yaklaşımda, azınlık sınıfa ait veriler ya da pozitif olması beklenen sınıfa ait veriler aşırı örneklendirilerek öğrenici üzerinde bu verilerin daha etkili ve yönlendirici olması amaçlanmaktadır. Böylelikle eğitim setinde kullanılan verilerin çoğunu kapsayan sınıfa dair önyargıların

oluşmasının engellenmesi mümkün kılınmaktadır [39]. Fakat dezavantajlara yol açan olan bazı durumlar oluşabilmektedir. Bunlardan biri, doğrusal bir dağılım bulunmaması sebebiyle iki azınlık sınıfı arasında konumlanmış diğer sınıflara ait eğitim verilerinin direkt olarak azınlık sınıfına ait veriler gibi sınıflandırılmasıdır. Bu yaklaşım sistem performansının belirli zafiyetlere sahip olduğunu ifade ederek test ve doğrulama sırasında bu zafiyet seçilen veri setine bağlı olarak gözlenebilmektedir. Bu zafiyetin etkisinin azaltulabilmesi, eğitim verilerinin SMOTE algoritmalarının yürütülmesinden önce daha doğrusal ayrılması ile mümkün olabilmektedir [39].

Sentetik Azınlık Aşırı Örnekleme Yöntemi Sonucu Veri Dağılımı



Şekil 2.2.11: SMOTE Yöntemi Uygulanmış Veri Seti

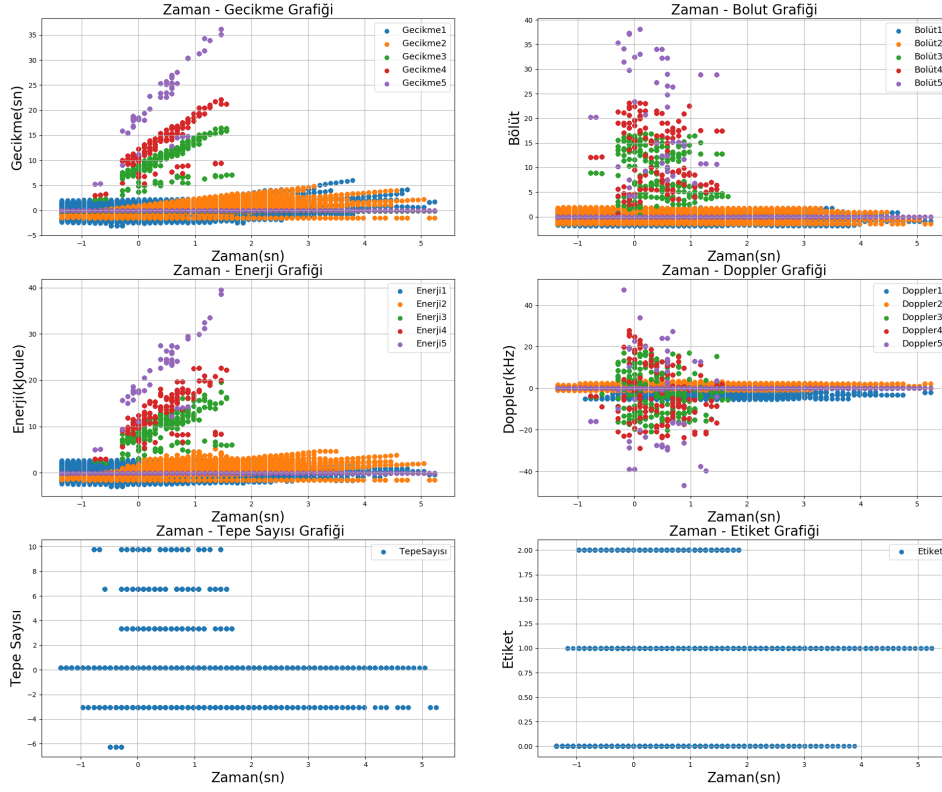
SMOTE uygulanmış veri dağılımı Şekil 2.2.11'de verilmektedir.

2.2.5.2 Rastgele Aşırı Örnekleme Yöntemi

Rastgele aşırı örnekleme, azınlık sınıfı örneklerinin rastgele kopyalanması yoluyla sınıf dağılımını dengelemeyi amaçlayan, buluşsal olmayan bir yöntemdir. Orijinal örneklerin kop-

yalanması nedeniyle aşırı uyum olasılığını artırabilme durumu söz konusu olabilmektedir [37]. Bu şekilde, örneğin, bir sembolik sınıflandırıcı, görünüşte doğru olan, ancak aslında çoğaltılmış bir örneği kapsayan kurallar oluşturabilmektedir.

Rastgele Aşırı Örnekleme Yöntemi Sonucu Veri Dağılımı



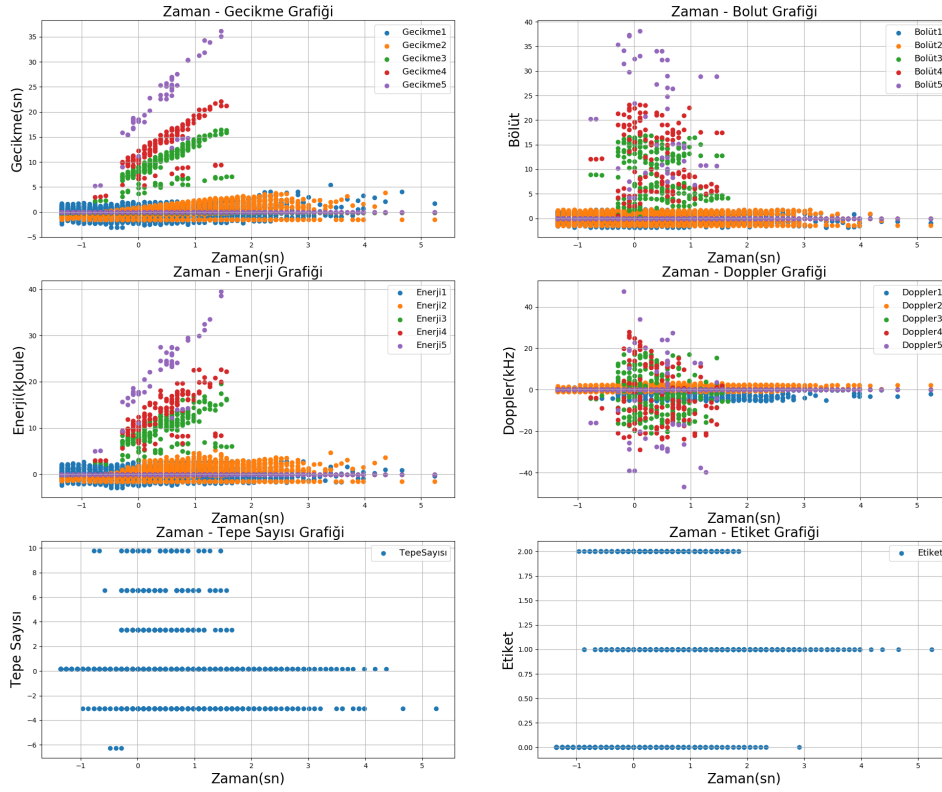
Şekil 2.2.12: ROS Yöntemi Uygulanmış Veri Seti

ROS uygulanmış veri dağılımı Şekil 2.2.12’de verilmektedir.

2.2.5.3 Rastgele Alt Örnekleme Yöntemi

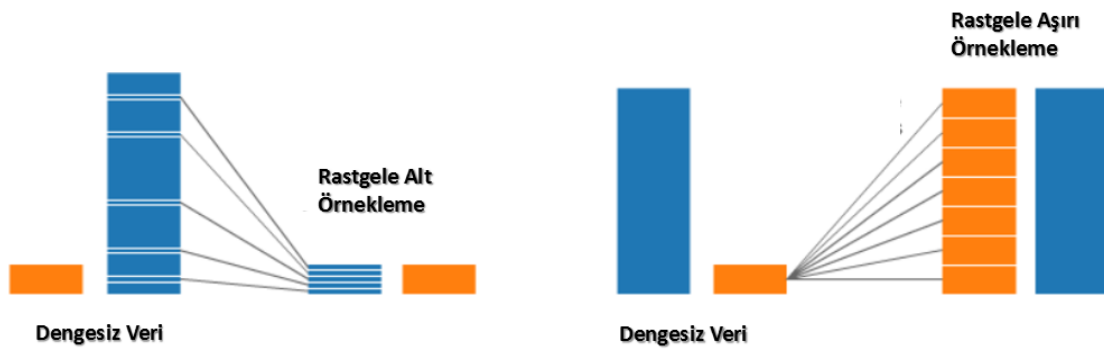
Rastgele Alt örnekleme aynı zamanda çoğunluk sınıfının rastgele ortadan kaldırılması yoluyla sınıf dağılımını dengelemeyi amaçlayan sezgisel olmayan bir yöntemdir. Rastgele eksik örnekleme en büyük dezavantajı, bu yöntemin tümevarım süreci için önemli olabilecek potansiyel olarak yararlı verileri atabilmesidir [37].

Rastgele Alt Örneklemeye Yöntemi Sonucu Veri Dağılımı



Şekil 2.2.13: RUS Yöntemi Uygulanmış Veri Seti

RUS uygulanmış veri dağılımı Şekil 2.2.13’de verilmektedir.



Şekil 2.2.14: RUS - ROS Sınıf Dengeleme Algoritmaları [40]

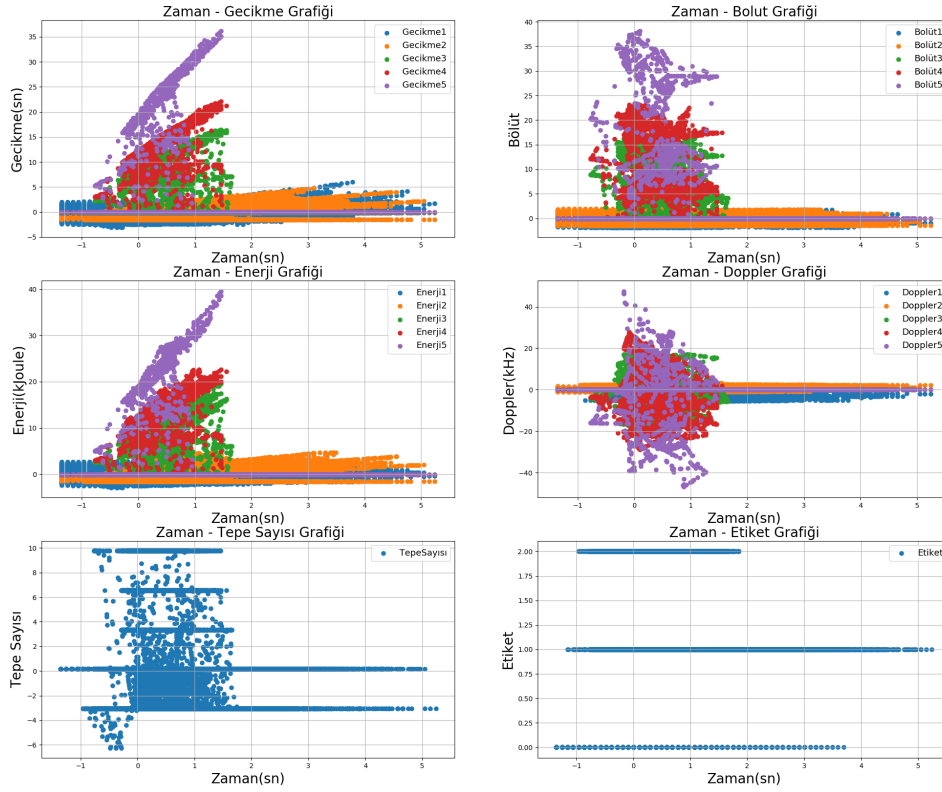
Şekil 2.2.14’de RUS ve ROS algoritmalarının orijinal veri setinden uygulama mantığı verilmektedir.

2.2.5.4 Sentetik Azınlık Aşırı Örnekleme Ve Düzenlenmiş En Yakın Komşu Yöntemi

ENN tekniği temelde k-En yakın komşu (k-NN) yaklaşımına dayanmaktadır. K-NN yönteminde hangi sınıfa ait olduğunun belirlenmesi istenen gözlemin, en yakınında olan gözlemlerin sınıfı tespit edilerek bu gözlem bu sınıfa dâhil olarak ifade edilmektedir. ENN tekniğinin farkı, eğer k-NN yöntemi ile belirlenen bir gözlemin sınıfı bahse konu k gözlemin sınıfından farklı olduğu tespit edilirse hem bu gözlem hem de bu gözlemin en yakınındaki gözlem veri setinden silinmektedir. Böylelikle yanlış öğrenmeye neden olabilecek verilerin öğrenme sürecinde etkisini azaltmak amaçlanmaktadır. ENN ifadesi “düzenlenmiş en yakın komşu” olarak belirtilmektedir. ENN yöntemi SMOTE algoritmalarıyla entegre kullanılabilir. Bu yaklaşım, SMOTE algoritmalarının azınlık sınıfın etkisini artırma yeteneği ile ENN tekniğinin verinin asıl sınıfıyla k-NN ile belirlenen sınıfı arasında bir fark olması durumunda her iki sınıftan bu verinin ve en yakınındaki diğer verinin silinmesi yeteneğini birleştirmektedir. SMOTE-ENN tekniğinde öncelikle veri setinde ki azınlık gözlemlerinden rastgele seçimler yapılmaktadır. Bu gözlemlere en yakın diğer k gözlemler belirlenmektedir. Bu mesafe birden küçük bir sayı ile çarpılarak en başta seçilen gözleme en yakın konumda yerler belirlenerek azınlık sınıfı aynı sınıfı paylaşan yapay gözlemler eklenmektedir. Bu işlem azınlık sınıfını tüm eğitim seti içerisinde ön yargıyı engelleyebileceği etkiye erişene dek sürdürülmektedir. Bu süreç tamamlandığında esasında SMOTE algoritması işlemiş olmaktadır. Bu aşamadan sonra k-NN uygulanarak her gözlemin en yakın gözlemleri ve bunların sınıfları belirlenir. Ardından ENN uygulaması başlatılarak seçilen gözlemlerin sınıfları yer aldıkları k çoklu gözlem sınıfından farklı ise hem bu seçilen gözlem hem de ona en yakın gözlem eğitim veri setinden silinmektedir. Bu işleme arzu edilen öğrenme başarısına ulaşına kadar devam edilmektedir. SMOTE ve ENN yöntemlerin bir arada kullanılmasıyla hem önyargı etkisi azaltılarak hem de doğru sınıf kontrolleri sürekli yapılarak öğrenme performansının artırılması amaçlanmaktadır [37].

SMOTE + ENN yönteminin arkasındaki motivasyon SMOTE + Tomek bağlantılarına benzer. ENN, Tomek bağlantılarından daha fazla örneği kaldırma eğilimindedir, bu nedenle daha derinlemesine bir veri temizliği sağlaması beklenmektedir. En yakın üç komşusu tarafından yanlış sınıflandırılan herhangi bir örnek eğitim kümesinden çıkarılmaktadır [38].

Sentetik Azınlık Aşırı Örnekleme Ve Düzenlenmiş En Yakın Komşu Yöntemi Sonucu Veri Dağılımı



Şekil 2.2.15: SMOTE + ENN Yöntemi Uygulanmış Veri Seti

SMOTE + ENN uygulanmış veri dağılımı Şekil 2.2.15’de verilmektedir.

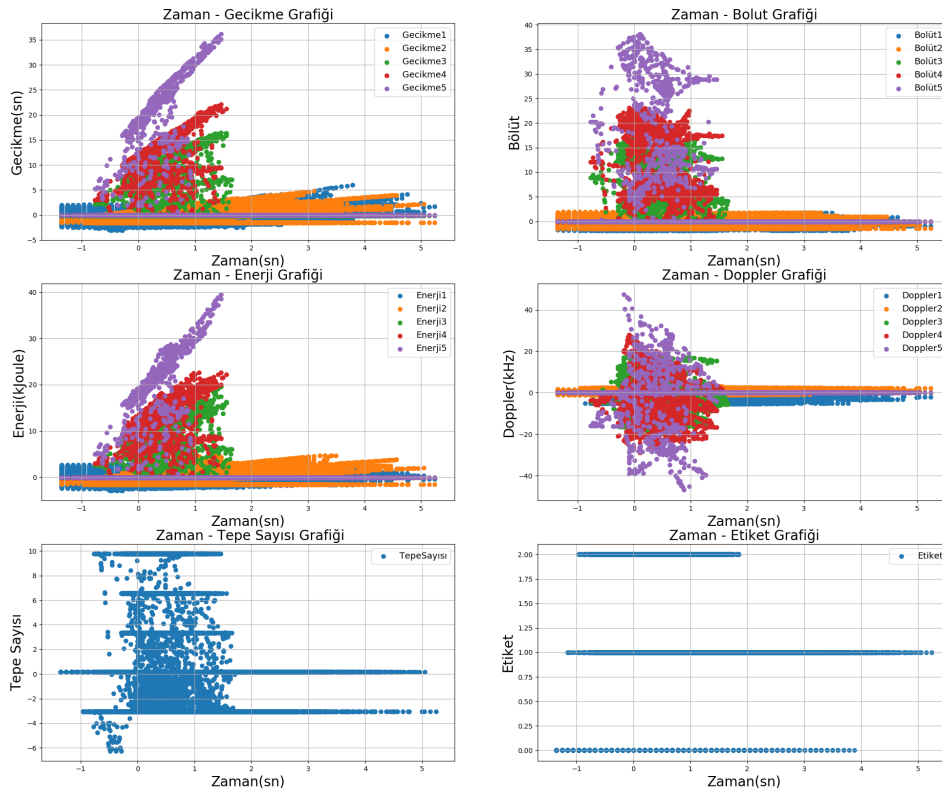
2.2.5.5 Sentetik Azınlık Aşırı Örnekleme Ve Tomek Örnekleme Yöntemi

Tomek link yöntemi esasında sıkıştırılmış en yakın komşu yönteminin farklı bir modifikasyonudur. Temel amaç, azınlık gözlemleriyle çok yakın konumlanmış çoğunluk gözlemlerini tespit ederek bunları eğitim setinden çıkarmaktır. Böylelikle sınıfların iç içe girmiş gözlemleri göz ardı edilerek sınıflandırma ya da öğrenme performansı artırılmaktadır.

SMOTE ve Tomek teknikleri bir arada kullanılabilir. Bu yaklaşım ile hem veri setinde eğitim sürecinde önyargı oluşmasının önüne geçmek hem de çoğunluk sınıfına ait gözlemlerle azınlık sınıfına ait gözlemleri birbirinden ayırarak k en yakın komşu yönteminin uygulanmasına zemin hazırlamak hedeflenmektedir. Öncelikle SMOTE tekniği uygulanarak azınlık sınıfından yapılan rastgele seçimlerin k en yakın komşularıyla mesafeleri belirlenerek bunlar birden küçük sayılarla çarpılır. En başta seçilen azınlık gözleme çok daha yakın bir

yer tespit edilmiş olur. Bu konuma azınlık sınıfına ait sentetik gözlemler eklenerek, azınlık sınıfının tesiri artırılır. Ardından güncellenmiş eğitim setinde çoğunluk gözlemlerinden rastgele seçimler yapılarak Tomek uygulaması başlatılır. Seçilen bu verinin en yakın komşusu belirlenir. Eğer belirlenen bu gözlemin azınlık sınıfına ait olduğu tespit edilirse, çoğunluk sınıfına ait bu gözlem eğitim veri setinden silinir. Bu işlem öğrenme performansı istenen seviyeye erişene dek devam eder [41].

Sentetik Azınlık Aşırı Örnekleme Ve TOMEK Örnekleme Yöntemi Sonucu Veri Dağılımı



Şekil 2.2.16: SMOTE - Tomek Yöntemi Uygulanmış Veri Seti

SMOTE + Tomek uygulanmış veri dağılımı Şekil 2.2.16’da verilmektedir.

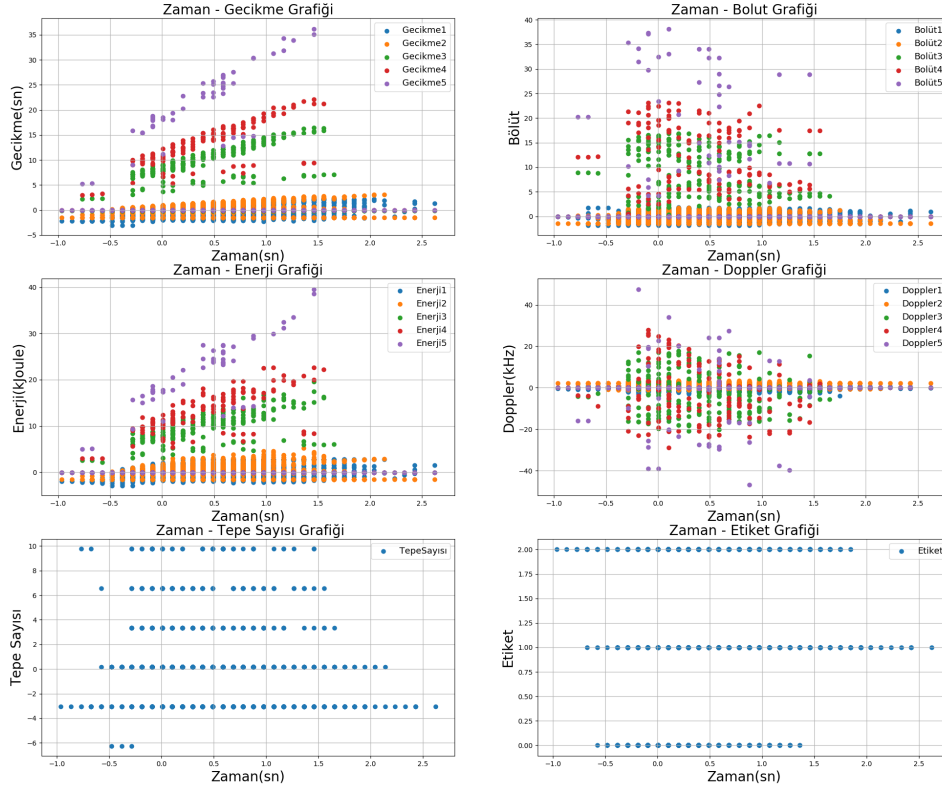
2.2.5.6 Yakındakini Atma Yöntemi

“Yakındakini Atma” algoritması olarak da ifade edilen bu yöntem temelde bir “yetersiz örnekleme tekniği” olarak adlandırılmaktadır. Çoğunluk sınıfının örneklerinin sayısının fazla olması nedeniyle, yine önyargıyı engellemek amacıyla, çoğunluk sınıfından seçilen ve azınlık örneklere en yakın noktada konumlanan rastgele örneklerin çıkarılması yaklaşımıyla ça-

lımaktadır. Yetersiz örnekleme tekniklerinin çoğunda bilgi kaybı riskini yönetmek önem arz etmektedir. Bu teknikte bilgi kaybı riskini azaltmak amacıyla en yakın komşuluk tekniklerinden faydalanılır. Böylelikle aykırı verilerin kaybolması engellenmektedir. Yakındakini atma algoritmasında öncelikle çoğunluk sınıfının tüm örneklerinin azınlık sınıfı örneklerine olan uzaklıkları euclidean uzaklık yöntemiyle hesaplanmaktadır. Çoğunluk sınıfında yer alan örneklerden, azınlık sınıfındaki örneklere en yakın konumda bulunan n adet örnek tespit edilmektedir. Tespit edilen en yakın çoğunluk sınıfı örnekleri veri setinden çıkarılmaktadır. Bu yöntemde eğer k adet azınlık sınıfı örnek mevcut olduğu düşünülürse, yöntemin sonucunda $k \times n$ adet çoğunluk sınıfı örnek çıkarılmaktadır [42].

Yakındakini atma tekniğinde azınlık sınıfı örneklerine en yakın n adet çoğunluk sınıfı örneği bulmanın üç farklı yöntemi bulunmaktadır [42]. İlk olarak azınlık sınıfının, çoğunluk sınıfına en yakın olan k örneğine olan ortalama mesafesinin en küçük olduğu çoğunluk sınıfı örnekleri seçilmektedir. İkinci olarak azınlık sınıfının, çoğunluk sınıfına en uzak olan k örneğine olan ortalama mesafesinin en küçük olduğu çoğunluk sınıfı örnekleri seçilmektedir. Son yöntem olarak ise azınlık sınıfındaki en yakın olan her örnek için belirli sayıda çoğunluk sınıfı örneğinin seçilmesini içermektedir. Bu sürüm daha çok çoğunluk sınıfı örneklerin karar sınırında yığıldığı veri seti yapısında tercih edilen bir sürüm olmaktadır.

Yakındakini Atma Yöntemi Sonucu Veri Dağılımı



Şekil 2.2.17: Yakındakini Atma Yöntemi Uygulanmış Veri Seti

Yakındakini Atma yöntemi uygulanmış veri dağılımı Şekil 2.2.17’de verilmektedir.

2.2.6. Sınıflandırma Performans Analiz Metrikleri

Hata Dizeyi

Hata dizeyleri, bir test setine uygulandığında tahmine dayalı bir modelin performansını tam olarak tanımlamanın uygun bir yol olmaktadır. Ayrıca, bir tahmine dayalı modelin performansının farklı yönlerini vurgulayabilen bir dizi farklı performans ölçütü için de temel oluşturmaktadır. Bu performans ölçütleri şu şekilde belirtilmektedir [43].

- **Gerçek Pozitif (GP)**

Doğru tahmin edilen pozitif değerlerdir. Gerçek sınıfın değeri ve tahmini sınıfın aynı olduğunu göstermektedir.

- **Gerçek Negatif (GN)**

Gerçek Sınıf	Gerçek Pozitif	Yanlış Negatif
	Yanlış Pozitif	Gerçek Negatif
		Tahmin Edilen Sınıf

Şekil 2.2.18: Hata Dizeyi

Bunlar doğru tahmin edilen negatif değerlerdir. Bu, gerçek sınıfın değeri ve tahmini sınıfın aynı olduğunu göstermektedir.

- **Yanlış Pozitif (YP)**

Bu değer gerçek sınıfınız ve tahmin edilen sınıfla çeliştiğinde ortaya çıkarmaktadır.

- **Yanlış Negatif (YN)**

Bu değer gerçek sınıfınız ve tahmin edilen sınıfla çeliştiğinde ortaya çıkarmaktadır.

Doğruluk, duyarlılık, hassasiyet ve F1 ölçüsü, doğrudan hata dizeyinden hesaplanabilen performans ölçütleri grubudur.

- **Doğruluk**

Doğruluk bir model olumlu bir tahmin yaptığında, bu tahminin ne sıklıkla doğru çıktığını belirtmektedir [43]. Doğruluk Acc olmak üzere;

$$Acc = \frac{GP + GN}{GP + YP + GN + YN} \quad (2.2.5)$$

- **Kesinlik**

Kesinlik, pozitif hedef düzeyine sahip olduğu tahmin edilen bir örneğin aslında pozitif hedef düzeyine sahip olduğundan ne kadar emin olunabileceğini belirtmektedir. Yüksek değerler daha iyi model performansı olduğunu göstermektedir [43]. Kesinlik K olmak üzere;

$$K = \frac{GP}{GP + YP} \quad (2.2.6)$$

- **Duyarlılık**

Pozitif hedef düzeyine sahip tüm örneklerin model tarafından bulunduğundan ne kadar emin olunabileceğini belirtmektedir. Yüksek değerler daha iyi model performansını

göstermektedir [43]. Duyarlılık D olmak üzere;

$$D = \frac{GP}{GP + YN} \quad (2.2.7)$$

- **F1 ölçüsü**

Kesinlik ve duyarlılık, daha basit yanlış sınıflandırma oranına yararlı bir alternatif sunan F1 ölçüsü olarak bilinen tek bir performans ölçüsüne indirgenmektedir [43]. F1 ölçüsü, kesinlik ve hatırlamanın harmonik ortalamasıdır ve şu şekilde tanımlanır: Kesinlik K , Duyarlılık D olmak üzere;

$$F1 = 2 \frac{KD}{K + D} \quad (2.2.8)$$

3. İMHA DEĞERLENDİRME ALGORİTMASI METODOLOJİSİ

Tehdidin vurulup vurulmadığına dair gerçek dünyaya ait bir veri setine sahip olunmadığı için öncelikli olarak çalışmalara radar sinyallerini sentetik veri şeklinde üreten ve çarpışma ıskalama gibi fiziksel olayları simüle eden bir simülatör geliştirilerek başlanmıştır.

Çalışmalar temel olarak iki adımda gerçekleştirilmektedir. İlk olarak oluşturulan simülatör ile tehdit, mühimmat ve imha sonucu oluşan parçacıklara ait sentetik sinyal ve fiziksel ortam verileri oluşturulmaktadır. Mühimmat aracılığıyla tehdidin ıskalandığı ve imha edildiği farklı senaryolar oluşturulmakta ve üç farklı şekilde öznitelikler elde edilerek etiketleme işlemi gerçekleştirilmektedir. Bu yöntemlerden ilkinde rastgele üretilen hız ve pozisyon verileri kullanılarak, gecikme, Doppler, bölüt, enerji ve senaryo sırasında ortamda bulunan parça sayısından tepe sayısı öznitelikleri hiçbir kısıtlama ve hata olmadığı varsayımı ile veri içeriği noksansız olacak şekilde oluşturulmaktadır. İkinci yöntemde ise bu çıkarılan öznitelikler gerçek bir uzun menzil radarı olan LockHeed Martin RRP-117 [44] referans alınarak oluşturulan ve bu radarın sahip olduğu kısıtlamalar da dâhil edilerek ve yüzde bir oranında verilere gürültü eklenerek yeni bir öznitelik seti oluşturulmaktadır. Üçüncü yöntemde ise referans alınan RRP-117 radarına [44] göre [1] tezinde üretilen mühimmat, tehdit ve parçacıklara ait daha gerçekçi sentetik sinyallerinin belirsizlik fonksiyonları kullanılarak analizinden elde edilen öznitelik seti kullanılmaktadır. Bu adımlar için MATLAB kullanılmaktadır. Senaryolar ayrı ayrı dosyalara kaydedilmektedir.

Daha sonrasında kaydedilen bu senaryolar farklı makine öğrenmesi ve derin öğrenme algoritmalarına girdi olacak şekilde sınıflandırma işlemi gerçekleştirilmektedir.

3.1. Fiziksel Simülatör Ve Sentetik Radar Sinyali Üretimi

Hava savunma sistemlerine ait gerçek imha ve ıskalama senaryolarına sahip olunamaması sebebiyle tehdit, mühimmat ve parçacıklara ait kütle, hız, senaryo boyunca izlenen yolu oluşturan, imha ve ıskalama senaryolarını ve elde edilen hız, zaman değerleri ve kabuller sonucunda sentetik radar sinyali üretimi gerçekleştiren bir simülatör oluşturulmuştur. Tehdit, mühimmat ve eğer vurulma senaryosu ise parçacıklarına ait zaman, pozisyon, hız değerleri doğrusal bir yol izledikleri kabul edilmektedir.

Makine öğrenmesi ve derin öğrenme algoritmalarının iyi bir şekilde öğrenebilmesi için çok

sayıda farklı senaryolardan oluşan veriye ihtiyaç duyulmaktadır. Bu sebeple her senaryo için farklı kütle, hız, farklı bir noktadan başlayıp farklı bir noktaya giden çarpışma sonucunda parçacık sayısı değişen birçok senaryo oluşturulmaktadır. Bu kısım rastgele sayılar kullanılarak otomatik olarak üretilecek şekilde tasarlanmıştır. Simülatörde imha ya da ıskalama senaryosu olması durumunda senaryoyu mantıken sıkıntıya sokabilecek bir durumun oluşup oluşmadığı kontrol edilmektedir. Böyle bir durum oluşması halinde senaryo iptal edilerek baştan oluşturulmaktadır. Havada tek bir tehdit ve tek bir mühimmatın bulunduğu değerlendirilmekte birden fazla mühimmat veya tehdit ile senaryolar oluşturulmamaktadır. Eğer imha senaryosu ise parçacıkların da senaryoya dâhil edildiği kabul edilmektedir.

Senaryolar da tehdit ve mühimmat için izleyeceği yola ait rasgele oluşturulan ilk ve son noktaları kullanılarak katedeceği yol üzerinden sahip olması rastgele bileşke hız atanmaktadır. Ayrıca kütleleri de referans alınan değerler üzerinden rastgele oluşturulmaktadır.

İzlenecek yol objeye ait hız V , zaman t olmak üzere $X_{son} = Vt + X_{ilk}$ formülü temel alınarak 10 ms aralıklarla doğrusal interpolasyon yapılarak örneklendirilmektedir. Senaryo boyunca yol ve rastgele bileşke hızı bilinen objelere ait uçuş süresi parametresi elde edilmektedir. Daha sonrasında uçuş süresi kullanılarak eksenler boyunca sahip olduğu hız değerleri hesaplanmaktadır.

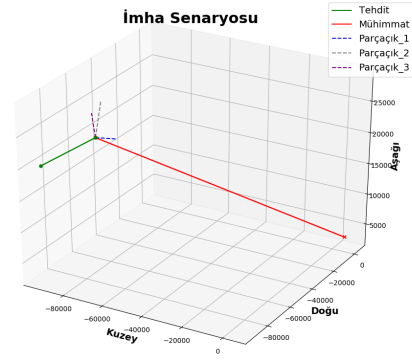
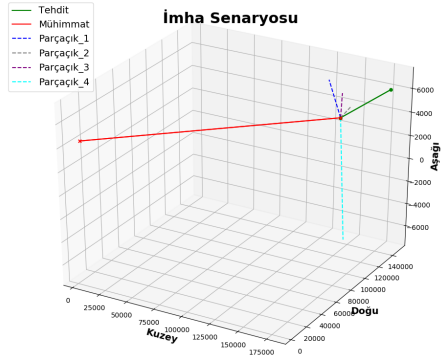
3.1.1. İmha Senaryosu

İmha senaryosunda mühimmatın ve tehdidin izlediği yolda kesişmelerini sağlayacak şekilde patlama öncesi son noktaları neredeyse aynı verilmektedir. Minimum parçacık sayısı bir, maksimum parçacık sayısı beş olacak şekilde rastgele farklı senaryolar oluşturulmaktadır. Şekil 3.1.1 ve Şekil 3.1.2’de imha durumlarına ait iki örnek senaryo verilmektedir.

İmha senaryoları kinetik enerji ve kütle korunum ilkeleri temel alınarak oluşturulmuştur. Vurulma sonrası patlama için ise oluşacak parçacıklar rastgele atanarak çarpışma anında hedef ve füzeyle ait bileşke hız vektörü, toplam kütle ve momentum korunacak şekilde izledikleri yol yine rastgele olarak belirlenmektedir.

$$KE_{ilk} = KE_{son} \quad (3.1.1)$$

$$KE_{ilk} = \frac{1}{2}M_{Mühimmat}V_{Mühimmat}^2 + \frac{1}{2}K_{Tehdit}V_{Tehdit}^2 + IE_{Mühimmat} - LE \quad (3.1.2)$$



Şekil 3.1.1: Dört parçacıklı İmha Senaryosu Şekil 3.1.2: Üç parçacıklı İmha Senaryosu

Kinetik enerji 3.1.2’de verildiği üzere mühimmat patladığında ortaya çıkan iç enerji IE ve ısıya dönüşerek kaybolan enerji LE de hesaba katılmaktadır.

Parçacıklar

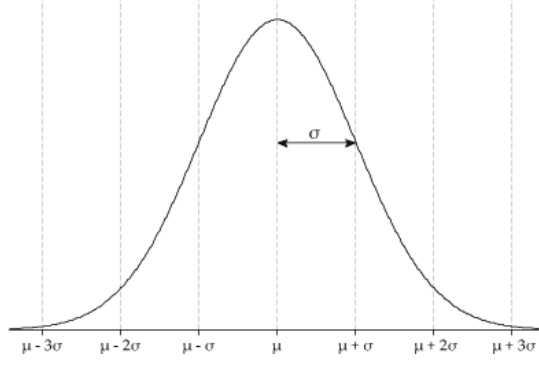
Patlama gerçekleşiyse parçacık sayısı ve parçacıklara ait mühimmat ve tehdidin sahip olduğu kütle ve hızlar ve mühimmata ait iç enerji ve patlama anında kaybolan enerji de hesaba katılır. Toplam kinetik enerji korunumu gözetilerek, mühimmat ve tehdede ait çarpışma anındaki bileşke hız vektörü korunacak şekilde parçacıklara ait hızlar ve kütleler rastgele oluşturulmaktadır. Patlama anında mühimmattan açığa çıkan iç enerji 10 MJ ve 40 MJ, kaybolan kinetik enerji senaryoya özel olacak şekilde rastgele 10 MJ ve 15 MJ aralığındadır.

Mühimmata ve tehdede ait toplam kütle korunacak şekilde parçacık sayısına göre her bir parçacığa özel rasgele kütle hesaplanır.

$$M_{Toplam} = M_{Mühimmat} + M_{Tehdit} \quad (3.1.3)$$

Patlama olduktan sonra parçacıkların belli bir süre izledikleri yol oluşturulmaktadır. Mühimmat ve tehdidin izledikleri yola ait bileşke vektör kullanılarak parçacıkların çarpışma sonrası ne şekilde dağılacakları Şekil 3.1.3’de belirtildiği gibi gaussian dağılımına göre hesaplanır.

Çarpışma anındaki mühimmat ve tehdede ait vektörleri kullanarak bileşke vektörün açısı hesaplanır, bileşke vektör korunacak şekilde patlama sonrasında oluşacak parçacıkların izleyecekleri yönün rastgele açıları oluşturulmaktadır. Bileşke vektöre ait hesaplanan kuzey-doğu eksenindeki izdüşüm vektörünün doğu eksenine ile yaptığı açı ve aşağı-doğu eksenindeki izdüşüm



Şekil 3.1.3: Parçacık Açılarının Normal Dağılımı

şüm vektörünün doğu eksenini ile yaptığı açı hesaplanır. Bu açılar ayrı ayrı normal dağılımın ortalama değerleri olacak şekilde, normal dağılımın varyans değeri ise 50° belirlenerek $+/- 3\text{varyans}$ aralığında parçacıklara ait saçılma açıları rastgele oluşturulmaktadır. Patlama sırasında buldukları ilk nokta bilinen parçacıklara ait, saçılma açısı da hesaplandıktan sonra $y = mx + b$ eğim formülü temel alınarak m hesaplanan hareket yönü açıları, x buldukları ilk nokta b hareket ettikleri mesafe bilgileri ile parçacıklara ait hareket boyunca buldukları noktalar oluşturulabilmektedir.

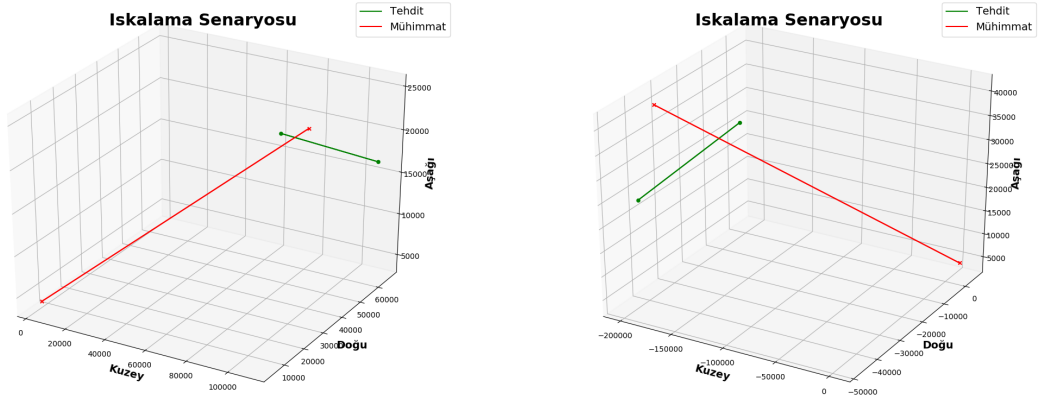
3.1.2. Iskalama Senaryosu

Iskalama senaryoları iki temel senaryodan oluşmaktadır. İki senaryoda temelde senaryo boyunca tehdit ve mühimmatın birbirine yaklaşması sonrasında mühimmatın hedefi iskalaması sonucu birbirlerinden uzaklaşmalarını simüle etmektedir. Bu senaryoya ek olarak mühimmeta ya da tehdiide ait izin düşürülerek devam eden bir simülasyon senaryosu da gerçekleştirilmektedir. Patlama olmadığı için fiziksel durumlarında (hız, kütle vb.) da değişim olmamaktadır. Şekil 3.1.4’de iskalama senaryosuna ait iki örnek senaryo verilmektedir.

Iskalama senaryosu mühimmat tehdidin yakınından geçip uzaklaşacak ya da bir süre sonra mühimmat ya da tehdiide karşı iz kaybedilecek şekilde oluşturulmaktadır.

3.1.3. Tehdit

Her bir döngü sırasında tehdit için rastgele atanan ilk ve son noktaların arası doğrusal olacak şekilde ekstrapolasyon yapılarak izledikleri yörünge belirlenmektedir. Tehdit manevralı hareket gerçekleştirilmemektedir. Savaş uçakları referans alınarak tehdiide ait hız 300 metre/saniye ile 500 metre/saniye arasında bir değer olacak şekilde rasgele olarak hesaplanmaktadır. Tehdiide ait ağırlık her senaryo özelinde rastgele olacak şekilde 8000 kg ile



Şekil 3.1.4: Iskalama Senaryoları

12000 kg aralığında oluşturulmaktadır [45].

Tehdidin kuzeyde ve doğuda radar ve atıcı platform referanslı olacak şekilde ilk görüldükleri nokta -35000 ile -200000 metre aralığında ya da 35000 ile 200000 metre aralığında olmaktadır. İlk noktaya bağlı olarak tehdiide ait son kuzey ve doğu noktası o noktanın 10000-40000 metre aralığında daha ilerisinde ya da gerisinde oluşabilmektedir. Tehdiide ait irtifa ise ilk olarak 4000 ile 30000 km arasındadır ve senaryo sonunda ilk doğu noktasının -3000 ile 10000 metre aralığında ilerlemiş vaziyette veriler üretilmektedir.

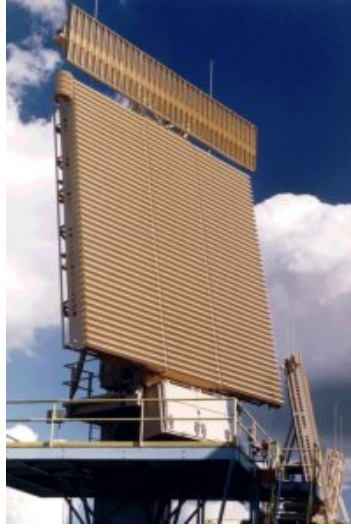
3.1.4. Mühimmat

Radar ve savunma platformu (0, 0, 0) pozisyonunda kabul edilmektedir. Mühimmatın savunma platformundan atıldığı var sayılmaktadır. Mühimmat manevralı hareket gerçekleştirmemektedir. Mühimmatın atıldığı andan itibaren ilk noktası kuzey, doğu ve aşağı eksen takımında 3000 ile 4500 metre arasında rastgele değer almaktadır. İmha senaryolarında mühimmat ile tehdiide ait son nokta aynı verilmektedir. Mühimmatın radar tarafından ilk görülme anı (mühimmatın fırlatılma anı) tehdiide ait toplam uçuş süresinden mühimmatın havada kalma süresi çıkarılarak hesaplanmaktadır. Bu kabule uygun olması için mühimmatın hep tehdiiden daha sonra görüldüğü varsayılmaktadır. Gerçeğe yakın verilerle çalışılırken kör menzil değeri de hesaba katılarak mühimmata ait bir ilk nokta atanmaktadır.

S400 ve S300 füzeleri referans alınarak oluşturulan mühimmata ait hız 1800 metre/saniye ile 4800 metre/saniye arasında bir değer olacak şekilde rasgele olarak hesaplanmaktadır.

Mühimmata ait ağırlık her senaryo özelinde rastgele olacak şekilde 1000 kg ile 2000 kg aralığında oluşturulmaktadır [46] [47].

3.1.5. Sentetik Radar Sinyalinin Oluşturulması



Şekil 3.1.5: RRP-117 Radarı, Lockheed Martin [44]

Zaman, hız ve pozisyon, kütle, verileri, obje hareket yönü ve radar arasındaki açı bilinen objelere ait RKA, gecikme, bölüt ve Doppler parametreleri hesaplanmaktadır. Bu değerler kullanılarak referans alınan radar özellikleri ile birlikte objeden yansıdığı varsayılan sentetik sinyal üretilebilmektedir.

Radarda gönderilen sinyalin $s(t)$, sinyalin gecikmesi τ , Doppler frekansı f_d , c ışık hızı, r menzil olarak alındığında hedeften yansıyan $r(t)$ sinyali;

$$r(t) = s(t - \tau)e^{j\pi f_d t} \quad (3.1.4)$$

$$\tau = \frac{2r}{c} \quad (3.1.5)$$

$$f_d = \frac{2f_c V \cos(\theta)}{c} \quad (3.1.6)$$

r radar platformu ile obje arasındaki mesafeyi, c ışık hızını, f_c radarın çalışma frekansını, θ ise radar ile havadaki objenin ilerleme yönünün yaptığı açıyı temsil etmektedir.

Simüle radarın hava resmini iki saniye de bir tam tur dönerek çıkardığı kabul edilmektedir. Radar hava resmini yarıda 360 dereceyi 3.6 derecelik hüzmeye genişliğinde taradığı için yan-

cada 100 bölüte ve yükselişte ise 2.7 derecelik bir hüzme ile 10 bölüte yayın yaptığı kabulü üzerine toplamda 1000 farklı bölüte sahiptir. Yükselişte 27 derecelik bir alanı tarayabilmektedir.

Radar sinyalinin örnekleme frekansı F_s olmak üzere 2×10^6 , darbe genişliği $2 \times 10^{-4} F_s$, darbe yinleme aralığı $2 \times 10^{-3} F_s$, radara ait taşıyıcı frekansı 1.2×10^9 , Antenin kazancı 25 dB'dir. Radarın çalışma frekansı 1.2 GHz'dir. Radarın attığı darbenin darbe genişliği 200 us, darbe tekrarlama aralığı ise 2 ms'dir. Radar sisteminde faz dizili anten bulunmaktadır.

Radar her bir bölüte yayın yapmakta fakat sadece obje bulunan bölütlerden yansıma alabilmektedir. Mühimmat ile tehdit bir senaryonun ilk aşamalarında farklı bölütlerde yer almakta, fakat bu iki unsur birbirlerine yaklaştıkları durumda zamanla aynı bölüt içerisinde yer almaya başlamaktadırlar.

Bir imha durumu söz konusu olduğunda bu bölüt veya bu bölüte yakın yerlerde parçacık kümesinin oluşumunun tespiti beklenmektedir. Iskalama durumlarında ise farklı bölütlerde belirmeleri, herhangi birine ait tepenin kayboluyor olması fakat diğerinin fiziksel özelliklerini (hız, kütle vb.) koruyor olması ya da fiziksel özelliklerini koruyacak şekilde devam ediyor olmaları beklenmektedir.

Simüle veri üzerinden enerji özniteliği çıkarılırken yansıyan objenin hareket yönü ile yapılan açı ve objenin kütlesi kullanılarak üretilen radar kesit alanı bilgisi elde edilmektedir. Daha sonra RKA ve objeden yansıyan sinyalin gecikme bilgisinden elde edilen mesafe kullanılarak simüle enerji bilgisi Eşitlik 3.1.8 şeklinde simüle edilmiştir.

$$RKA_{Obje} = \frac{Kütle_{Obje} \sin(\theta)_{Obje}}{10000} + 0.5 \quad (3.1.7)$$

$$Enerji_{Obje} = \frac{RKA_{Obje}}{Gecikme_{Obje}^2} 10^6 \quad (3.1.8)$$

RKA, bir nesnenin radar tarafından ne kadar algılanabilir olduğunun bir ölçüsüdür. RKA özelinde literatürde temel alınabilecek modeller bulunmaktadır. Bu modellerden biri olan Swerling bir hedefin RKA dalgalanmalarını istatistiksel bir bakış açısıyla tanımlayan dört hedef modelidir. Bu model Peter Swerling tarafından, dalgalanma kayıplarının radar menzili üzerindeki etkilerinin belirlenmesi amacıyla oluşturulmuştur. Swerling 1, Swerling 2, Swerling 3 ve Swerling olarak adlandırılan dört Swerling modeli vardır. 5. Beşinci Swer-

ling modeli, sabit bir RCS'ye sahip bir hedef olarak tanımlanır. Bu bir kürenin ideal RCS'si olduğundan, oryantasyon açısı ve zaman üzerinde sabit dönen model bir kürenin temsilcisi olmaktadır [48]. Bir hedefin mikro dalgalanmalarını istatistiksel asimetri olarak tanımlayan Rician hedef modeli de literatürde geçen RKA modellerindendir [49]. Bu tür RKA hedef modelleri, daha gerçekçi bir yaklaşım elde etmek adına kullanılabilir.

3.2. Öznitelik Setlerinin Oluşturulması

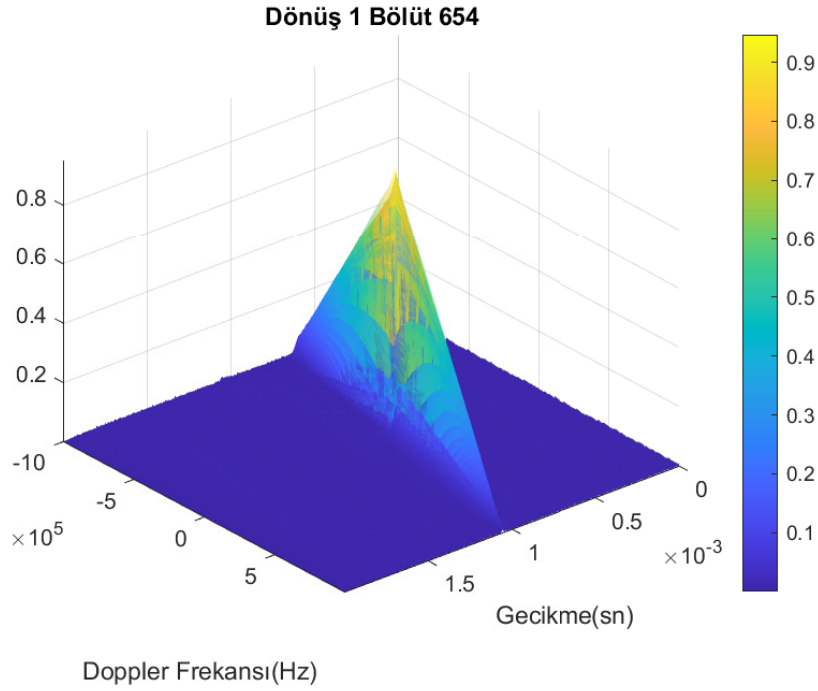
Ham radar sinyallerini sınıflandırma amacıyla eğitimde kullanıyor olmanın çeşitli dezavantajları bulunmaktadır. Verinin sınıflandırmaya çalışılan konu dışında gürültüye ve aşırı uyumlamaya sebebiyet verebilecek olması, eğitim için kullanılan donanımın öğrenmede yetersiz kalması, çok daha güçlü donanımlara ihtiyacın olması ve eğitim süresinin çok daha uzun sürmesi gibi sebepler sıralanabilir. Dolayısıyla öznitelik çıkarımı yapılarak bu veri üzerinden öğrenme işlemi yapılması hedeflenmektedir.

Veriden küçük önemli bir set çıkarılarak ilgili modele girdi olarak sağlanır ve modelin karmaşıklığı azaltılarak daha verimli bir şekilde öğrenmesi sağlanmaktadır. Ayrıca, öznitelik çıkarma işlemi, hangi veri özelliklerini izlemek istediğimizi ve hangi veri özelliklerinin önemli olduğunu belirlememize olanak tanımaktadır. Öznitelik çıkarma işlemi ile makine öğrenimi algoritmasının, ham girdi verilerinden kalıplar bulmak yerine seçilen metriklerdeki kalıpları kullanarak tahminlerini oluşturmasını sağlamaktadır.

Radar tarafından tespit edilen her objenin farklı bir radar imzası bulunduğundan her bir obje farklı öznitelik vektörleri ile ifade edilebilmektedir. Bu öznitelik vektörleri genel olarak frekans zaman grafiklerinden oluşmaktadır. Bu grafiklerden bir objenin varlığını yansıtan tepe sayıları, her bir tepenin hangi bölütle ortaya çıktığı bilgisi, tepenin oluşmasına neden olan objeye ait sinyalin gecikme süresi, objenin hızından ve hareket yönünden elde edilen Doppler değeri, tepeye ait enerji gibi öznitelik değerleri çıkarılabilmektedir. Elektromanyetik sinyal göndericiden gönderildikten sonra karşıda bulunan bir objeye çarpıp geri yansımaktadır. Yansıyan sinyal farklı bir frekans değeriyle yansımaktadır ve frekansta meydana gelen bu kayma Doppler frekansı olarak adlandırılmaktadır. Doppler frekansı sinyalin yansıdığı objenin hızına, radar sinyali ve sinyalin yansıdığı obje hareket yönü arasındaki açığa, ışık hızına bağlı olarak değişmektedir.

Öznitelik seti belirsizlik analizi [1] temel alınarak analiz sonucundan ayırt edici özellikler

çıkarılmaktadır. Şekil 3.2.1’de bir tam tur radar dönüşüne ait belirsizlik analiz grafiği verilmektedir.



Şekil 3.2.1: Belirsizlik Analizi [1]

Bu analiz sonucu oluşturulabilecek öznitelikler belirli bir eşik üzerinde en yüksek enerjiye sahip olan maksimum 5 tepe ve bu tepelere ait öznitelikleri içermektedir. Bu öznitelikler bulunan bölüt, Doppler, gecikme, enerji olmak üzere dört temel alandır. Her radarın bir tam turu için en çok 5 tepe olabilecek şekilde bölüt, Doppler, gecikme, enerji alanı ve 1 tepe sayısı ile birlikte toplamda 21 öznitelik bulunmaktadır. Bu 21 özniteliğe ek olarak her senaryo dönüşü için Vurulma öncesi, Iskalama, İmha olmak üzere bir etiketleme işlemi gerçekleştirilmektedir. 500’ü imha senaryosu 500’ü ıskalama senaryosu olmak üzere üç ayrı öznitelik seti özelinde ayrı ayrı oluşturulan bu 1000 senaryo makine öğrenmesi ve derin öğrenme algoritmalarına girdi olmaktadır.

Çizelge 3.2.1: Öznitelik Çıkarımı ve Etiketleme Dosyası Örneği

TepeSayısı	ObjeBölüt1	ObjeGecikme1	ObjeDoppler1	ObjeEnerji1	ObjeBölüt2	ObjeGecikme2	ObjeEnerji5	Etiket
2	63	0.00143757	-2607.634156	0.449637334	63	0.00143383	0	0
2	63	0.001441916	-2607.732973	0.450723869	63	0.001462277	0	1
.
.
2	123	0.001473533	-3715.256854	0.744376992	123	0.001473281	0	0
3	123	0.00151285	-13423.87838	0.698284609	123	0.001502323	0	2

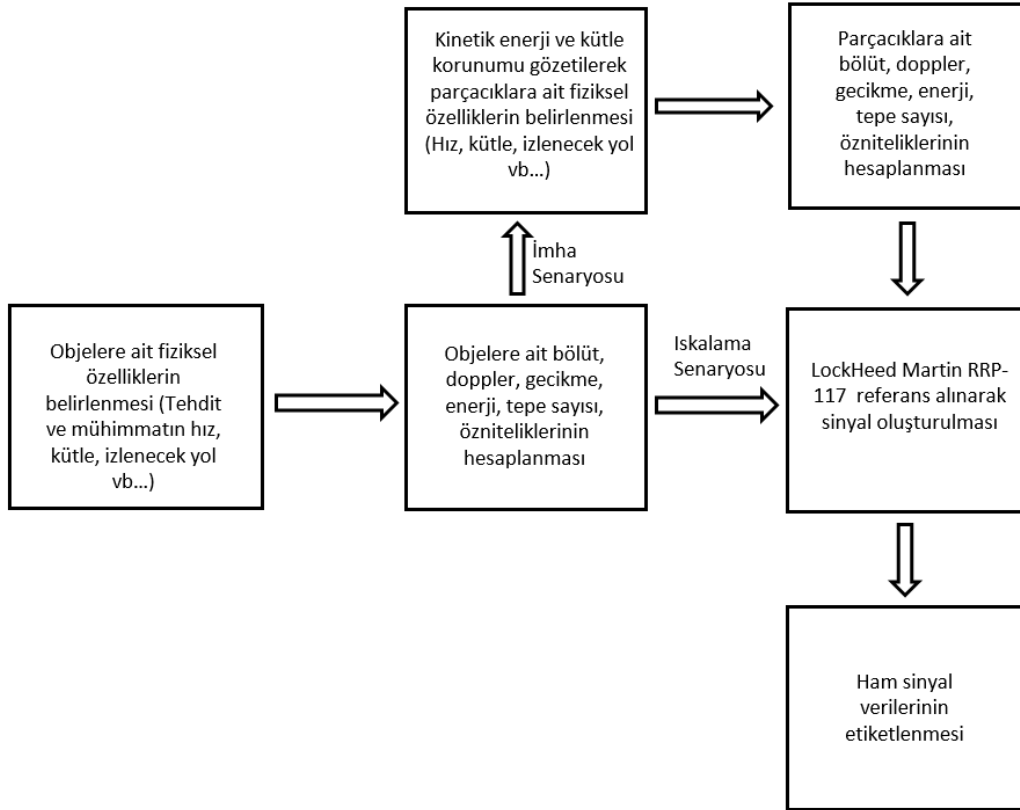
Öznitelik setleri oluşturulurken yüksek enerjili tepelere ait bölüt, gecikme, Doppler, enerji

değerleri en önce olacak şekilde büyükten küçüğe enerjisi belirli bir eşik değerinin üzerindeki maksimum 5 tepe için öznitelik çıkarım işlemi gerçekleştirilmektedir. O anda belirlenen eşik değerinin altında kaldığı için elde olmayan öznitelikler 0 olacak şekilde verilmektedir.

Sınıflandırma işlemleri için biri ham veri diğerleri noksansız, noksan ve belirsizlik analizi öznitelik seti olmak üzere 4 farklı veri seti yapısı kullanılmıştır.

3.2.1. Sentetik Radar Sinyal Veri Seti

Sentetik olarak üretilen radar sinyal verileri ile öğrenme işlemleri gerçekleştirilmiştir. Simüle radar sinyalleri simülatör rasgele üretilen hız, kütle, pozisyon verileri kullanılarak hesaplanan gecikme, Doppler, radar kesit alanı verilerinin kullanılması ile Lockheed Martin RRP-117 radar [44] özellikleri referans alınarak üretilmektedir. Şekil 3.2.2'de radar sinyal veri setinin oluşturulma akışı verilmektedir.

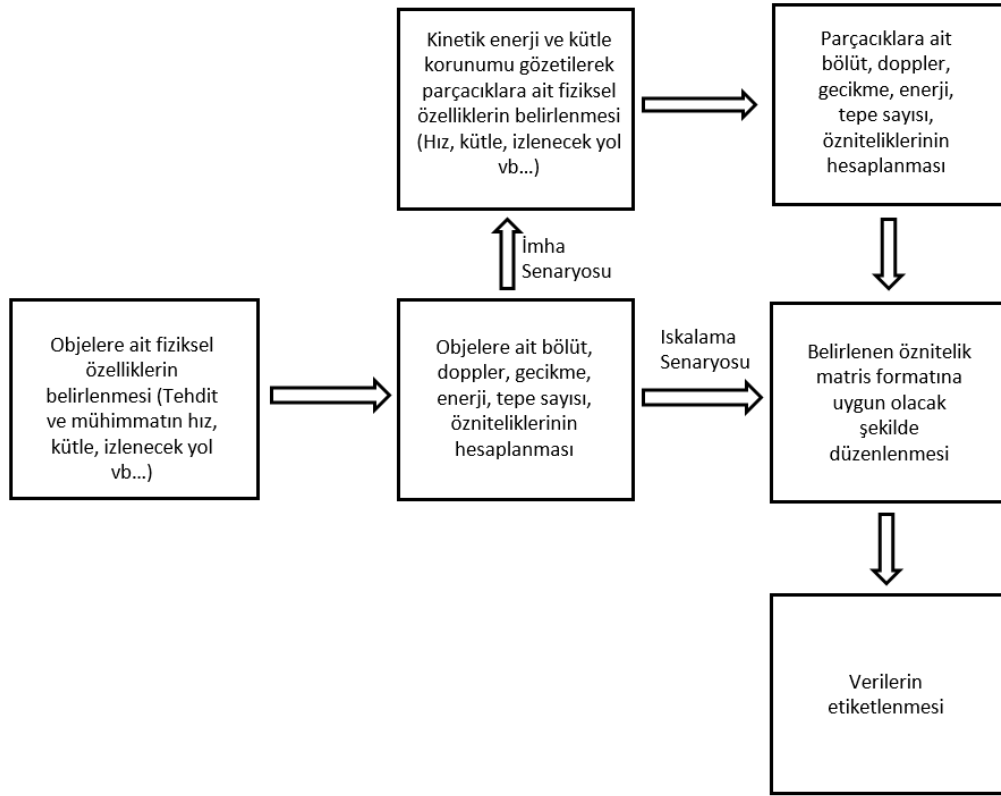


Şekil 3.2.2: Ham Radar Sinyal Veri Senaryo Akışı

3.2.2. Noksansız Öznitelik Seti

Simülâtör üzerinden objeye ait rasgele üretilen hız, kütle, pozisyon verileri kullanılarak hesaplanan bölüt, gecikme, Doppler, enerji öznitelikleri ve her bir dönüşte radar tarafından görülen obje sayısı kullanılarak oluşturulan öznitelik vektör setleri kullanılmaktadır.

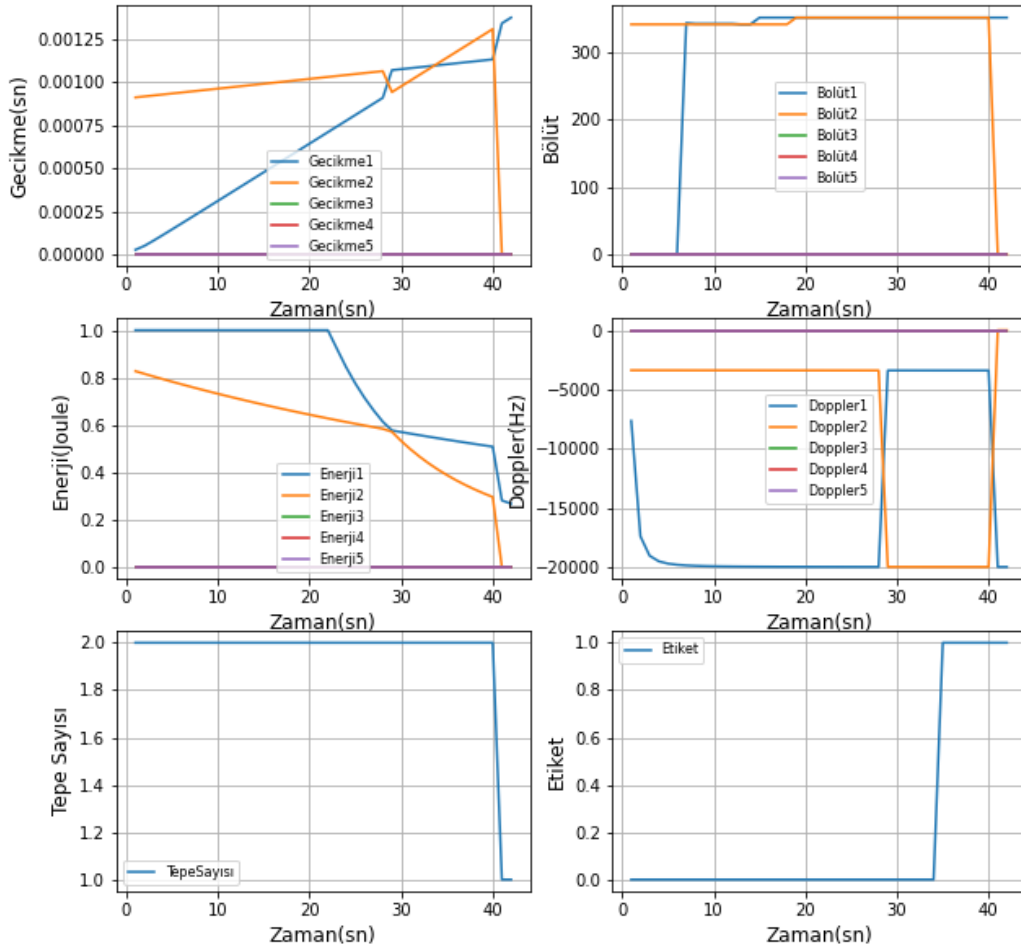
Bu öznitelik setinde gürültü ve gerçek dünyadaki radar ve analiz kısıtlamaları yok sayılmaktadır. Şekil 3.2.3’de noksansız veri setinin oluşturulma akışı verilmektedir.



Şekil 3.2.3: Noksansız Veri Senaryo Akışı

Şekil 3.2.4’de de ıskalama senaryosuna ait bu örnekte gecikme grafikleri incelendiğinde iki obje 28. saniyeye kadar birbirlerine yaklaşırken daha sonrasında birlikte hareket ettiği ve 40. saniye civarı bir tanesine ait radar izinin düştüğü gözlenmektedir. Grafikler incelendiğinde kalan objeye ait belirgin bir değişim olmadığı için bir imha durumu gerçekleşmediği yorumu yapılabilmektedir.

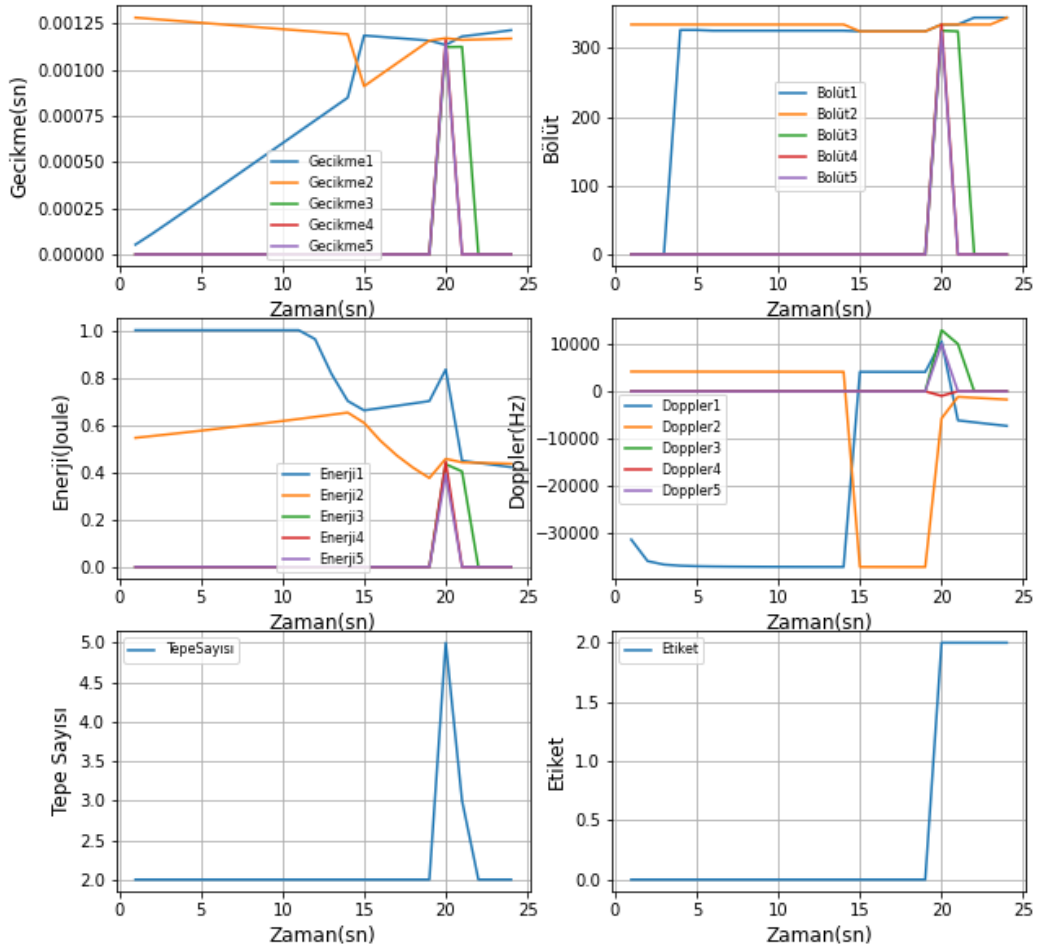
Iskalama Senaryosu Öznitelik Grafiği



Şekil 3.2.4: Noksansız Veri Iskalama Senaryosu Öznitelik Grafiği

Şekil 3.2.5’de de imha senaryosuna ait bu örnekte gecikme grafikleri incelendiğinde iki obje birbirlerine doğru hareket etmiş 20. saniye civarı tepe sayısında ve özniteliklerde belirgin değişimlerin olduğu görülmektedir. Bir imha durumunun söz konusu olduğu kanısına varılmaktadır.

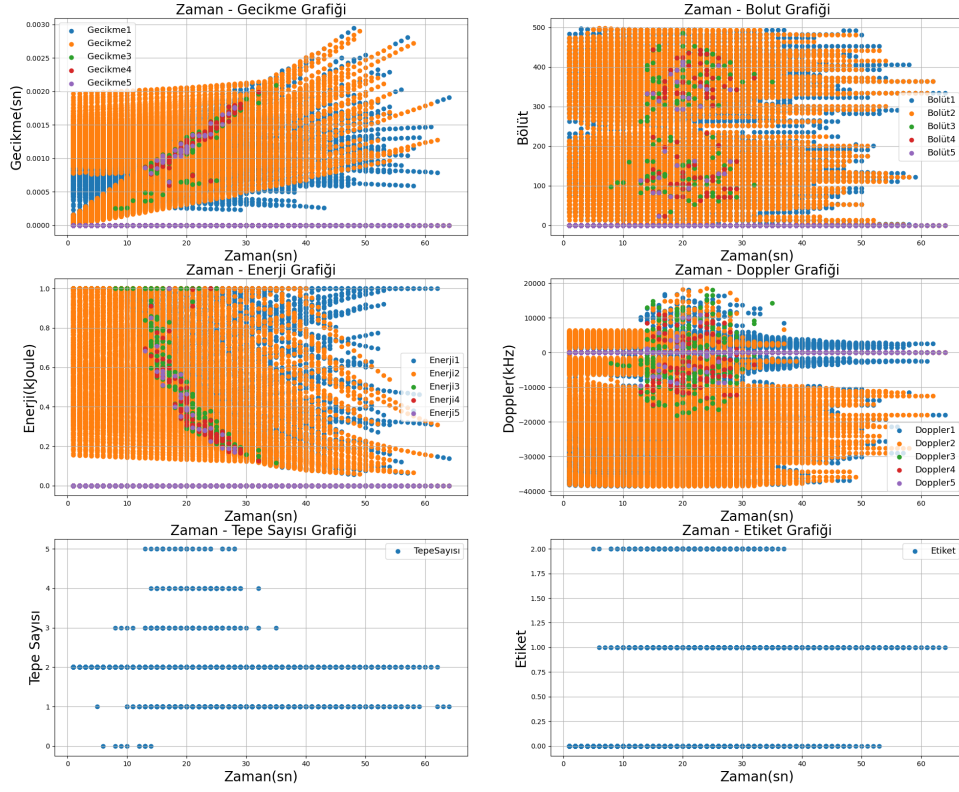
İmha Senaryosu Öznitelik Grafiği



Şekil 3.2.5: Noksansız Veri İmha Senaryosu Öznitelik Grafiği

Noksansız veri seti ile üretilen 1000 senaryo makine öğrenmesi ve derin öğrenme algoritmaları için eğitim ve test işlemlerinde kullanılmaktadır.

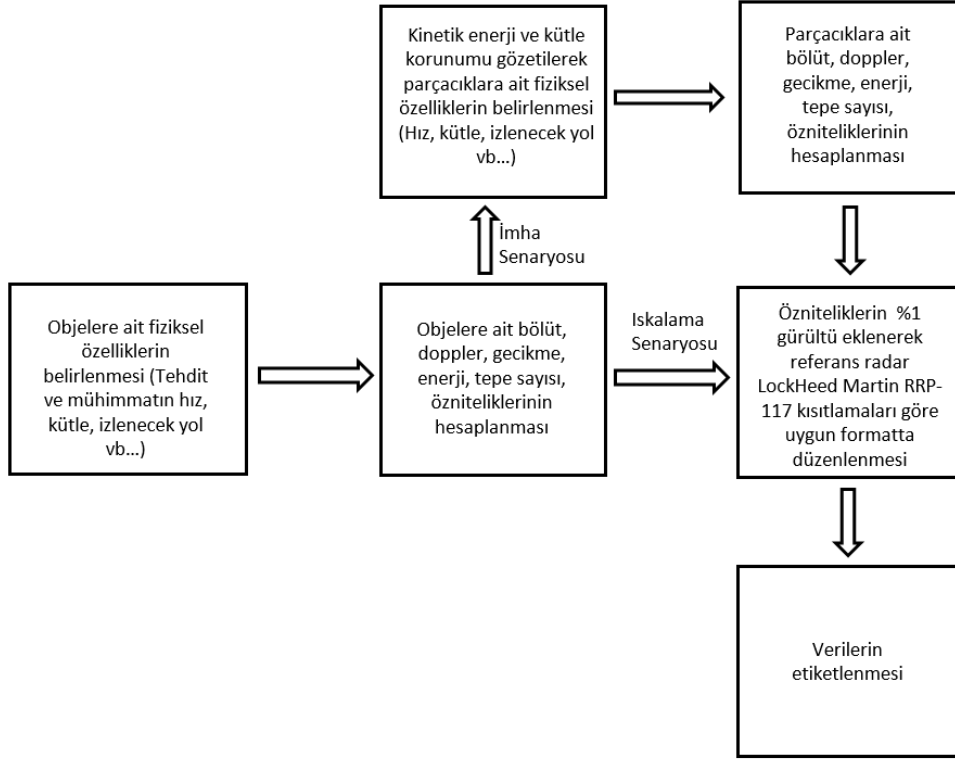
Noksan Öznitelik Seti Dağılımı



Şekil 3.2.6: Noksansız Veri Seti Dengesiz Öznitelik Seti Grafiği

Şekil 3.2.6’da sınıflandırma algoritmalarına sağlanan dengesiz veri setinin dağılımı görülmektedir.

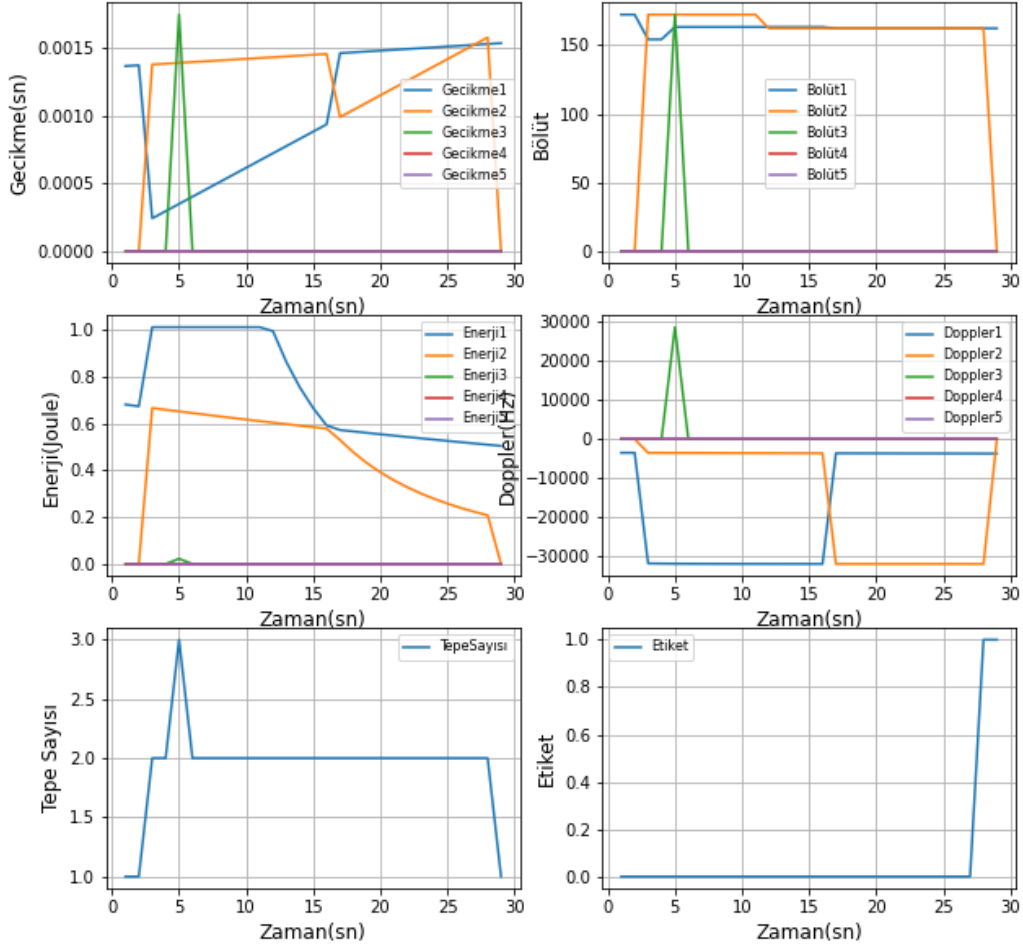
3.2.3. Noksan Öznitelik Seti



Şekil 3.2.7: Noksan Veri Senaryo Akışı

Bu veri setinde noksansız veri seti gerçek hayattaki referans alınan RRP-117 radarına [44] ve analiz kısıtlamalarına uygun olacak şekilde güncellenmiştir. Verilere yüzde bir olacak şekilde gürültü eklenmiştir. Gecikme çözünürlüğü 1 mikrosaniye, Doppler çözünürlüğü 5000 Hz, mesafe çözünürlüğü 150 metre, kör menzil 30 km olacak şekilde sentetik öz nitelik üretimi gerçekleştirilmektedir. Şekil 3.2.7’de noksan veri setinin oluşturulma akışı verilmektedir.

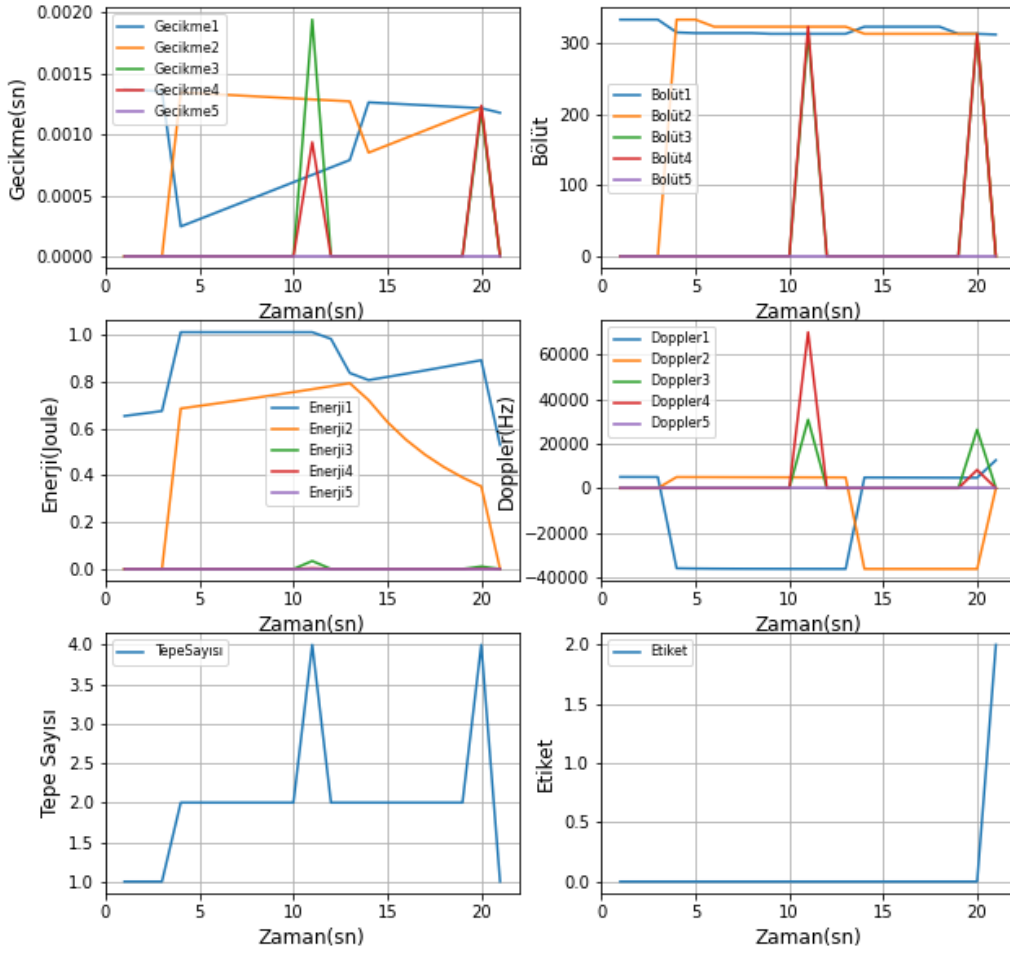
Iskalama Senaryosu Öznitelik Grafiği



Şekil 3.2.8: Noksan Veri Seti Iskalama Senaryosu Öznitelik Grafiği

Şekil 3.2.8’de de senaryo sırasında oluşan gürültülü öznitelikler görülebilmektedir. Iskalama senaryosuna ait bu örnekte iki obje birlikte görülürken 5. saniye civarı hayalet tepe ortaya çıktığı ve imha durumu olabileceği gibi görüldüğü fakat devamında daha önce elimizde bulunan özniteliklere ait bir değişim söz konusu olmadığı sonucuna varılmaktadır. 27. saniyeye kadar mühimmatın hedefe yaklaştığı fakat daha sonrasında mühimmata ait radar izinin kaybedildiği yorumu yapılabilir. Doppler değeri büyük olana ait iz düştüğü için mühimmata ait izin düşüp tehdite ait izin devam ettiği görülmektedir. Tehdite ait özniteliklerde belirgin bir değişim olmadığı için bir imha durumu gerçekleşmediği yorumu yapılmaktadır.

İmha Senaryosu Öznitelik Grafiği

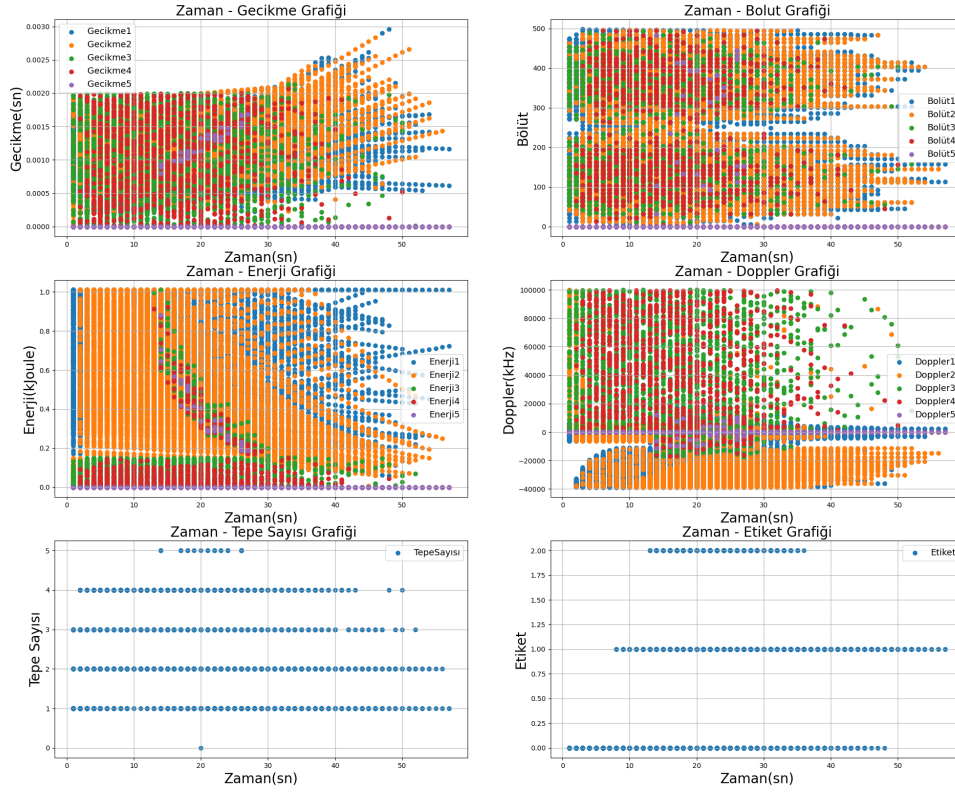


Şekil 3.2.9: Noksan Veri Seti İmha Senaryosu Öznitelik Grafiği

Şekil 3.2.9’da da senaryo sırasında oluşan gürültülü öznitelikler görülebilmektedir. İmha senaryosuna ait bu örnekte iki obje birlikte görülürken 11. saniye civarı tepe sayısındaki değişim nedeniyle bir imha durumu söz konusu gibi durmaktadır fakat aralarındaki gecikme farkı dikkate alındığında ve sonrasında objelere ait özniteliklerde değişim olmadığı göeüldüğünden anlık bir gürültü olduğu yorumu yapılabilmektedir. 20. saniye ve sonrasında bakıldığında önce parçacık sayısının arttığı sonrasında bir azalma olduğu ayrıca objelere ait enerji ve Doppler grafikleri değişiminden bir imha durumu gerçekleştiği yorumu yapılmaktadır.

Noksan veri setine ile üretilen 1000 senaryo makine öğrenmesi ve derin öğrenme algoritmaları için eğitim ve test işlemlerinde kullanılmıştır. Bu senaryoların hepsine ait öznitelik

dağılımını Şekil 3.2.10'da verilmektedir.

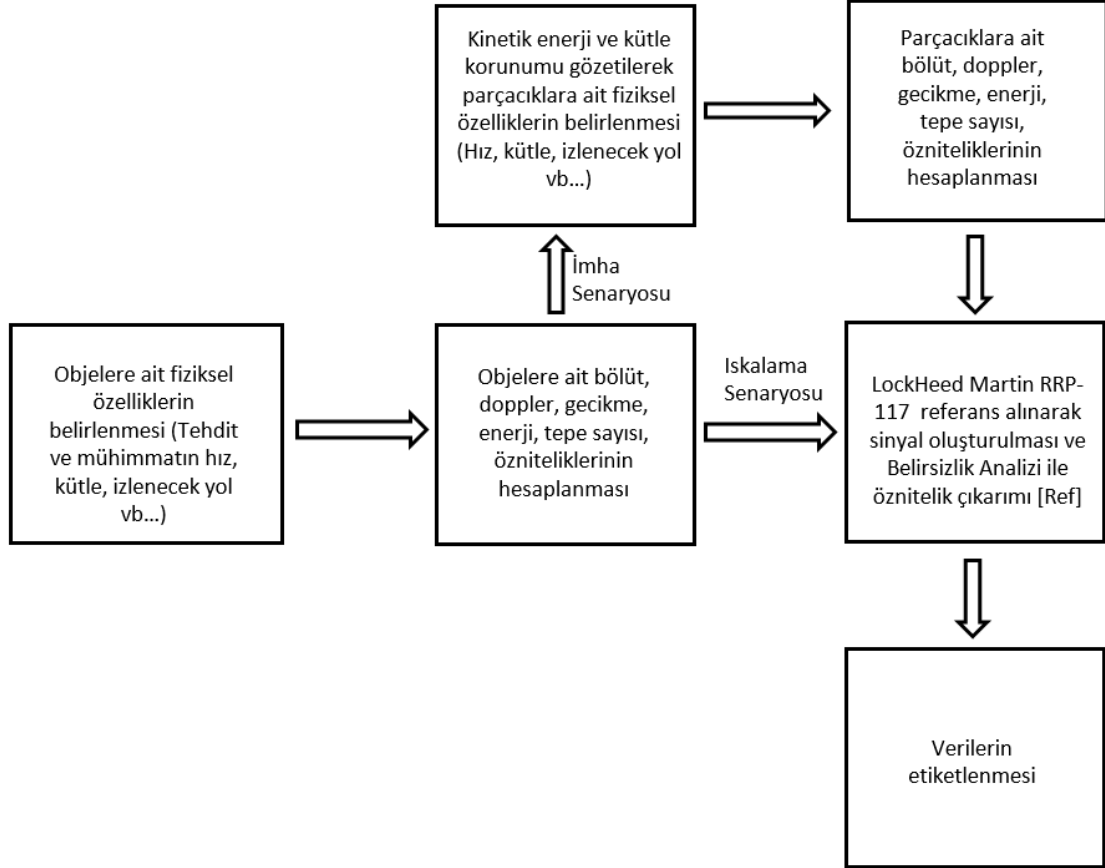


Şekil 3.2.10: Noksan Veri Seti Dengersiz Öznitelik Seti Grafiği

Şekil 3.2.6'da verilen noksansız öznitelik seti ile karşılaştırıldığında Şekil 3.2.10'daki veri setinin gürültülü olduğu görülebilmektedir.

3.2.4. Belirsizlik Analizi Öznitelik Seti

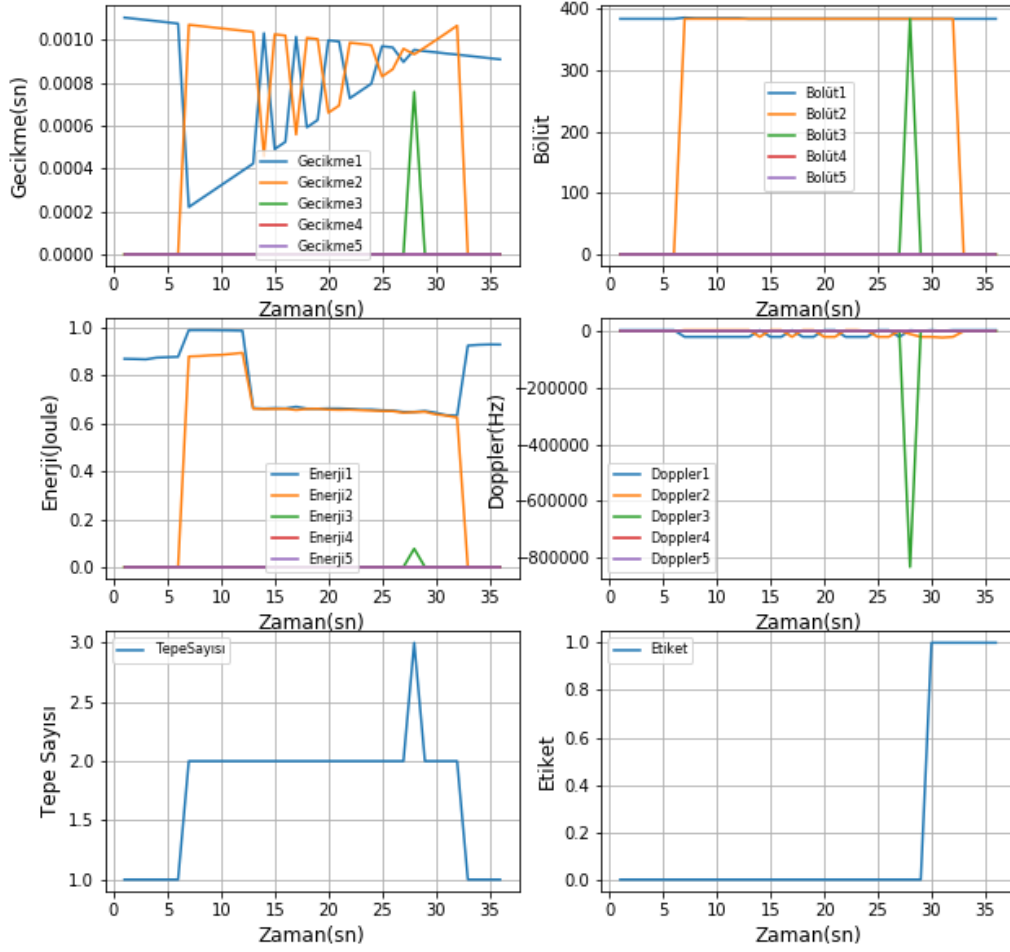
Analiz sonucu elde edilen veri setinde ise simülasyon aracılığı ile oluşturulan sentetik radar sinyalleri kullanılarak "Zaman Frekans Analiz Yöntemleri ile İmha Değerlendirme" [1] tez çalışması sonucu çıkarılan öznitelik seti kullanılmıştır. Şekil 3.2.11'de belirsizlik analizi veri setinin oluşturulma akışı verilmektedir.



Şekil 3.2.11: Belirsizlik Analizi Öznitelik Seti Senaryo Akışı

Şekil 3.2.12'de de senaryo sırasında oluşan gürültülü öznelikler görülebilmektedir. Iskalama senaryosuna ait bu örnekte iki obje birlikte görülürken 33. saniye civarı bir tanesine ait radar izinin kaybedildiği görülmektedir. Daha önce bulunan objelerden birine ait gecikme, enerji, bölüt gibi özneliklerde belirgin bir değişim olmadığı için bir imha durumu gerçekleştirmediği yorumu yapılabilmektedir.

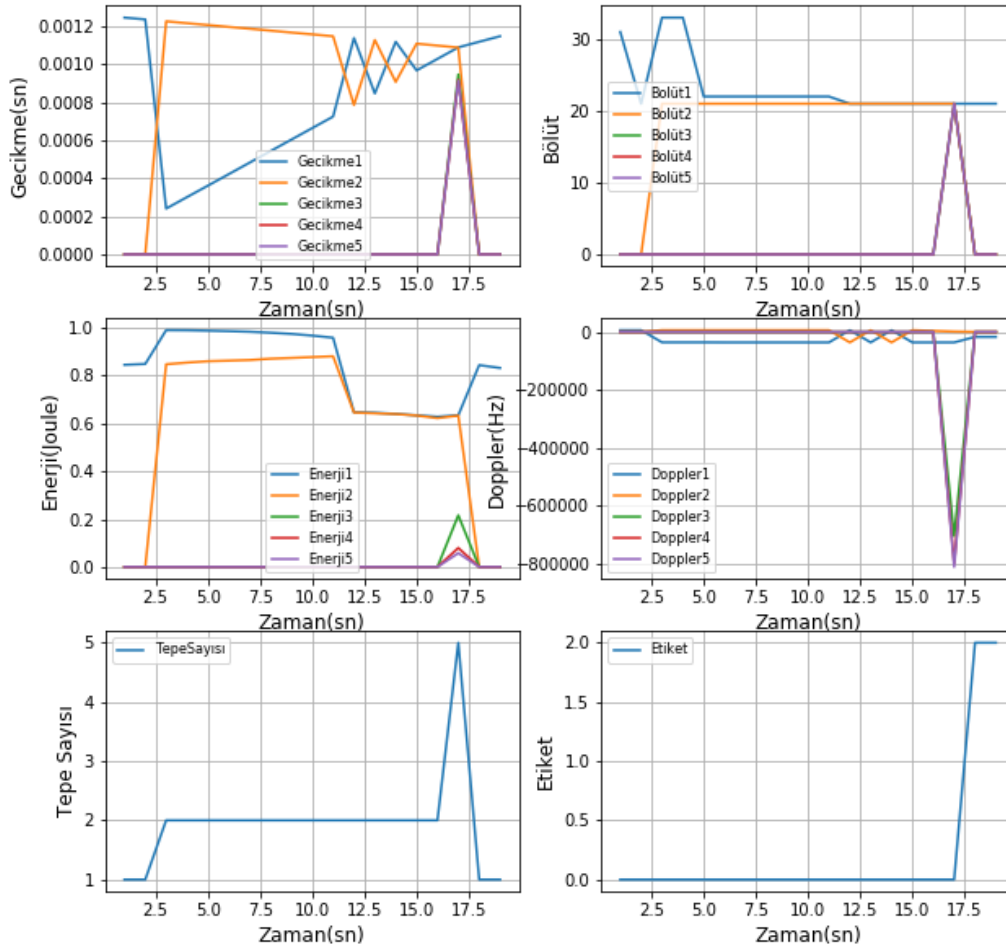
Iskalama Senaryosu Öznitelik Grafiği



Şekil 3.2.12: Belirsizlik Analizi Seti Iskalama Senaryosu Öznitelik Grafiği

Şekil 3.2.13’de de senaryo sırasında oluşan gürültülü öznitelikler görülebilmektedir. İmha senaryosuna ait bu örnekte iki obje birlikte görülürken 17. saniye civarı obje sayısında belirgin bir artış görülmektedir. Ayrıca bu andan itibaren elde edilen öznitelikler diğerlerinden farklı bir karakteristiğe sahip olduğu için imha gerçekleştiği yorumu yapılabilmektedir.

İmha Senaryosu Öznitelik Grafiği



Şekil 3.2.13: Belirsizlik Analizi Seti İmha Senaryosu Öznitelik Grafiği

Belirsizlik analizi ile üretilen 1000 senaryo makine öğrenmesi ve derin öğrenme algoritmaları için eğitim ve test işlemlerinde kullanılmaktadır. Bu senaryoların hepsine ait öznitelik dağılımı Şekil 3.2.14’de verilmektedir.

Belirsizlik Analizi Öznitelik Seti Dağılımı



Şekil 3.2.14: Belirsizlik Analizi Dengesiz Öznitelik Seti Grafiği

3.3. Verilerin Etiketlenmesi

Sınıf etiketleri imha değerlendirme algoritması için üçe ayrılmıştır.

- Vurulma Öncesi Durum (0)
- Iskalama Durumu (1)
- İmha Durumu (2)

3.3.1. Vurulma Öncesi Durum

Mühimmat fırlatıldıktan sonra bir ıskalama ya da imha anına kadar geçen süreye ait öznelikler bu sınıfa ait olacak şekilde etiketlenmektedirler.

3.3.2. Iskalama Durumu

Mühimmat ile tehdidin arasındaki mesafenin en az olduğu andan itibaren senaryo sonuna kadar mühimmat ile hedefin arasındaki mesafenin artmaya devam ettiği ya da herhangi birisine ait izin düştüğü senaryoya ait verilerden çıkarılan öznitelikler bu sınıfa ait olacak şekilde etiketlenmektedirler.

3.3.3. İmha Durumu

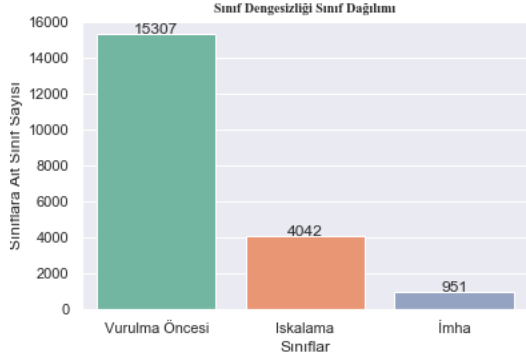
Mühimmat ile tehdidin çarpışması sonrasında oluşan parçacıklara ait verilerden çıkarılan öznitelikler bu sınıfa ait olacak şekilde etiketlenmektedirler.

Etiketleme işlemi için iki yöntem uygulanmaktadır.

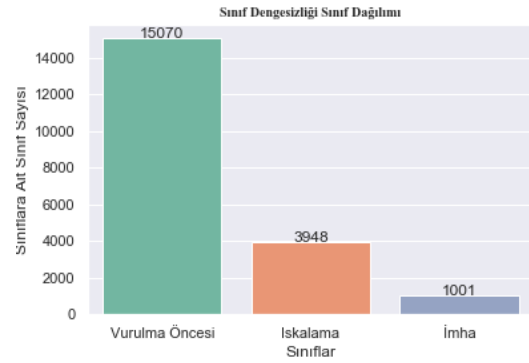
İlk seçenekte makine öğrenmesi tabanlı algoritmalar için radara ait iki saniyelik bir tam dönüş anı etiketlenmektedir. Bir dönüş için öznitelik çıkarım analizi sonucu elde edilen grafikten çıkarılabilecek tepe sayısı maksimum beş olacak şekilde kabul edilmektedir. Bu sebeple görülmesi beklenen her bir tepeye ait enerji, Doppler, gecikme değerleri ve tepenin görüldüğü bölüt ayrıca öznitelikleri oluşturmaktadır. Totalde bir tepeye ait dört adet öznitelik çıkarımı gerçekleştirilebilmektedir. Maksimum beş tepe görülebileceği varsayımından 20 adet özniteliğe ek olarak toplam tepe sayısı da öznitelik olarak kabul edildiğinde 21 adet özniteliğe indirgenebilmektedir. Özetlendiğinde, her bir tur radar dönüşü için objelerden yansıyan sinyallerden 21'lik bir vektör şeklinde öznitelik çıkarımı gerçekleştirilmektedir. Bu vektörü oluşturan hava resmine ait bir sınıflandırma etiketi belirlenmektedir.

İkinci uygulamada ise uzun kısa süreli bellek algoritmasının kullanılabilmesi için geçmişe ait tam turların da dâhil edildiği bir öznitelik vektörü etiketlenmiştir. Bulunulan ana ait radar turundan iki tur öncesi radar turları da dâhil edilerek geçmişe dayalı sıralı bir öznitelik elde edilmektedir ve bu dizi etiketlenmektedir.

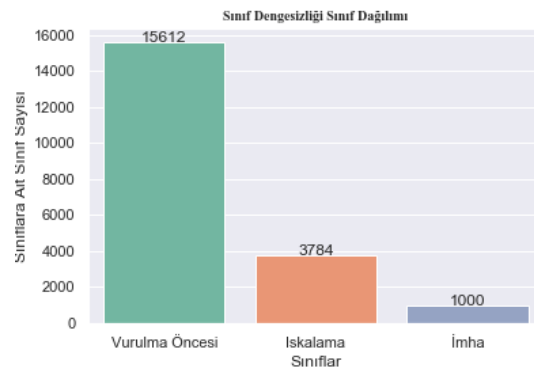
Noksansız öznitelik seti, noksan öznitelik seti ve belirsizlik analizi öznitelik seti olmak üzere toplamda üç ayrı veri setinin her biri için 500 adet ıskalama, 500 adet imha senaryosu olmak üzere toplamda 1000 adet senaryo oluşturulmaktadır. Etiketleme sonucu elde edilen sınıflar arasında bir sınıf dengesizliği problemi olduğu görülmektedir. Kullanılan üç veri setine ait sınıf dağılımı Şekil 3.3.3'de verilmektedir.



Şekil 3.3.1: Noksansız Veri Seti Sınıf Dağılımı



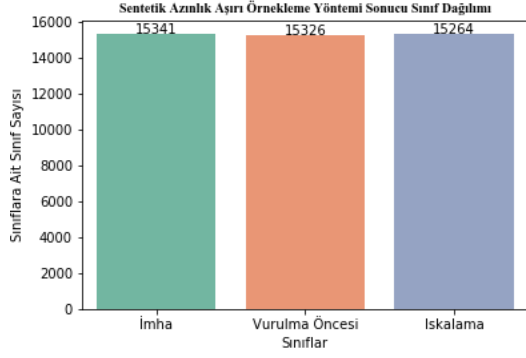
Şekil 3.3.2: Noksan Veri Seti Sınıf Dağılımı



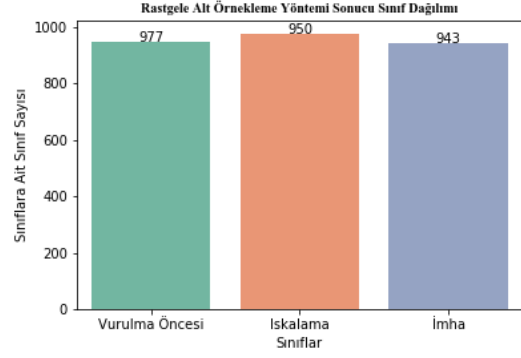
Şekil 3.3.3: Belirsizlik Analizi Veri Seti Sınıf Dağılımı

Verilerde bulunan sınıf dengesizliğinin eğitimlerde algoritmaların bir sınıfa meyilli olmasına neden olup öğrenmeyi olumsuz etkileyebileceği değerlendirilerek noksansız öznitelik seti, noksan öznitelik seti ve belirsizlik analizi öznitelik seti veri kümelerine ayrı ayrı sınıf dengesini sağlayacak yöntemler uygulanmaktadır. Bu yöntemler; SMOTE yöntemi, ROS yöntemi, RUS yöntemi, SMOTE-ENN yöntemi, SMOTE-Tomek örnekleme yöntemi ve yakındakini atma yöntemi.

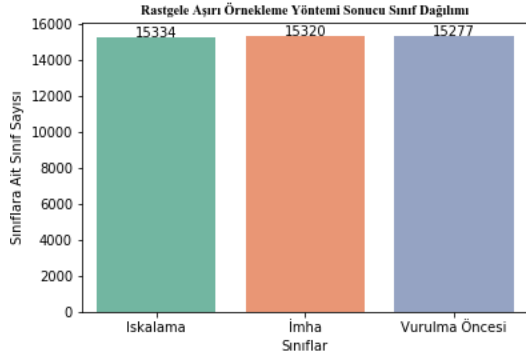
Şekil 3.3.4'de sınıf dengeleme yöntemleri sonrası imha vurulma öncesi ve ıskalamaya yönelik sınıf dağılım durumları verilmektedir. Dengeleme sonrası neredeyse her sınıfa ait veri dağılımının eşitlendiği ve eğitim sırasında herhangi bir sınıfa meyilli olma durumunun ortadan kalktığı görülmektedir.



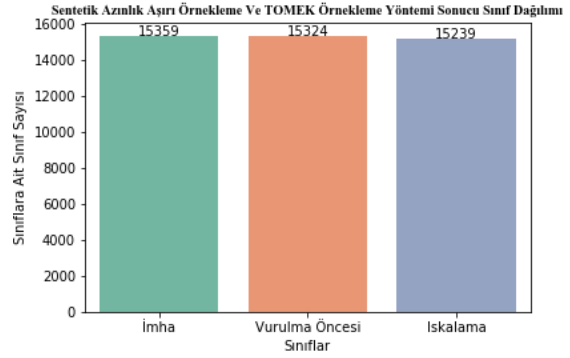
(a) SMOTE Yöntemi



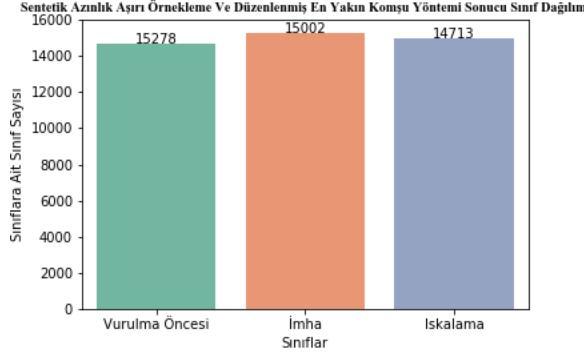
(b) RUS Yöntemi



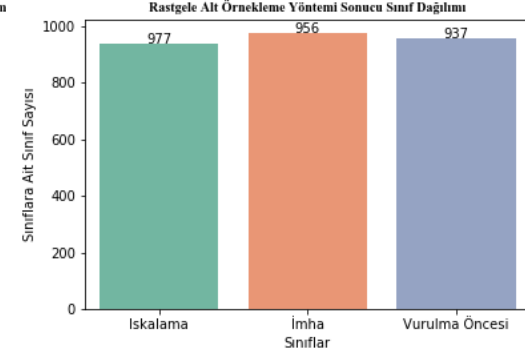
(c) ROS Yöntemi



(d) SMOTE - Tomek Yöntemi



(e) SMOTE - ENN Yöntemi

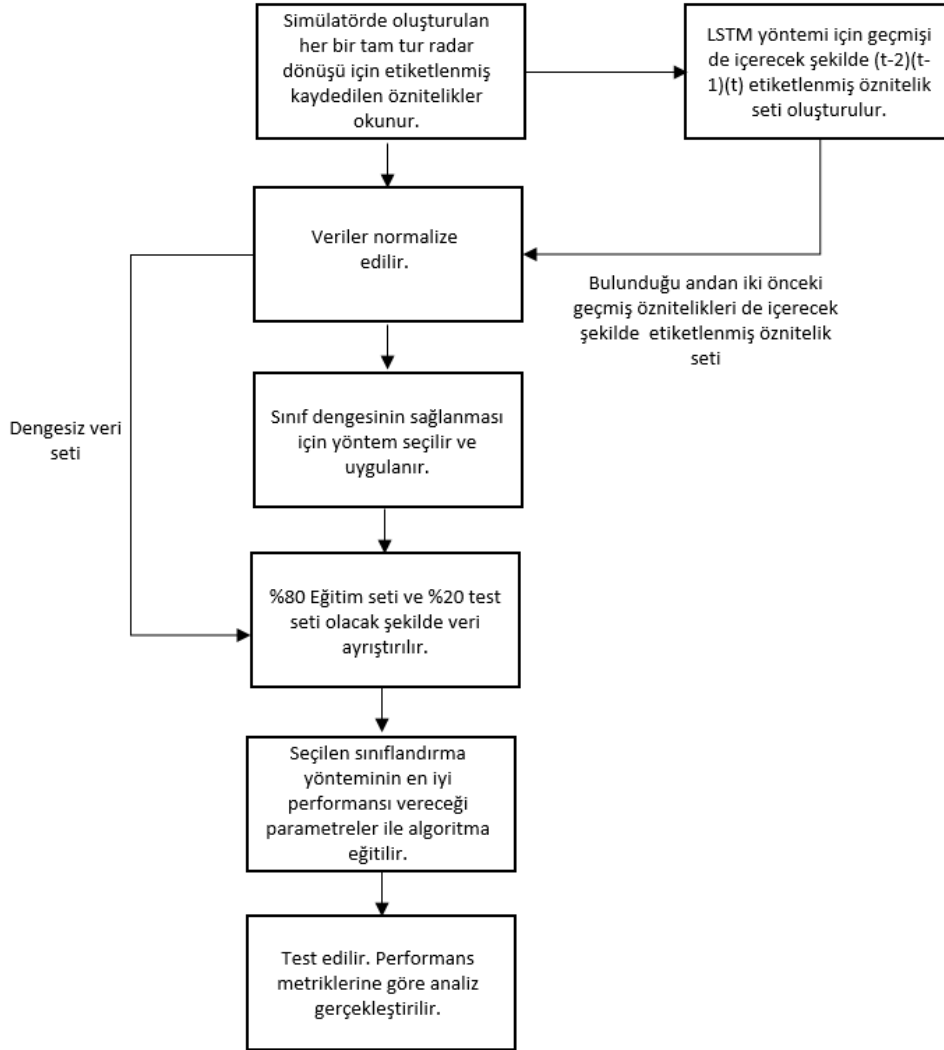


(f) YA Yöntemi

Şekil 3.3.4: Sınıf Dengeleme Yöntemleri Uygulanmış Noksansız Veri Seti Sınıf Dağılımı

Sınıf dengeleme yöntemlerinin her biri ayrı ayrı uygulanarak oluşturulan veri setleri eğitim ve test veri setine ayrılmaktadır. Makine öğrenmesi algoritmaları ile eğitim ve sınıflandırma işlemleri gerçekleştirilmekte ve performans değerlendirilmesi yapılmaktadır.

3.4. İmha Değerlendirme Sınıflandırma Yöntemleri



Şekil 3.4.1: İmha Değerlendirme Algoritma Akışı

İmha, iskalama ve vurulma öncesi durumlarının doğru şekilde sınıflandırılması için Şekil 3.4.1’de görüleceği üzere öncelikle simülâtör yardımı ile oluşturulan sentetik öznitelikler ve belirsizlik analizi sonucu oluşturulmuş olan öznitelikler ayrı veri setleri olacak şekilde bir dosyaya kaydedilmektedir. Makine öğrenmesi algoritmalarında her özniteliğin eşit şekilde etkiliyor olması için veriler normalize edilir. Bu aşamada kullanılan yöntem olan standardizasyon, ortalama değerini 0, standart sapmanın ise 1 değerini aldığı, dağılımın normale yaklaştığı bir metottür. Daha sonrasında sınıf dengesinin sağlanması planlanıyorsa öznitelik seti uygun denge sağlama algoritması seçilip bu işlem gerçekleştirilmektedir. Eğer dengesiz veri seti ile sınıflandırma yapılacaksa bu adım atlanmaktadır. Sonrasında elde edilen öznitelik setinin %80’i eğitim %20 i ise test verisi olarak ayrıştırılmaktadır. Öznitelik seti seçilen

sınıflandırma yöntemine göre daha önce hiperparametre değerleri belirlenmiş en iyi performans ölçütünün alındığı algoritmaya girdi olarak beslenmektedir.

Makine öğrenmesi algoritmalarından performans hangi parametrelere ile daha iyi olduğunun belirlenmesi için ızgara arama yöntemi kullanılmaktadır. Bu parametreler, bir sınıflandırıcının performansı üzerinde büyük bir etkiye sahiptir ve kötü seçilirse, fazla uydurma ile ilgili sorunlara yol açabilmektedir. Izgara arama yönteminin kullanılmasının nedeni kısmen yalın olması, anlaşılması kolay bir yöntem olması ve iyi bir performans göstermesidir.

Makine öğrenmesi algoritmalarından artırım algoritması, karar ağacı, rasgele orman, destek vektör makineleri algoritmaları kullanılmaktadır. Derin öğrenme algoritmalarından ise sinir ağı algoritması ve LSTM algoritması kullanılmaktadır. Bu algoritmalar sonucu belirlenen performans metrikleri üzerinden hangisinde daha iyi sınıflandırma sonucu elde edildiği sonucuna varılacaktır.

Tüm bu adımlarda Spyder programı aracılığıyla kullanılan kütüphaneler şu şekildedir: Numpy kütüphanesi matematiksel işlemler için, Pandas kütüphanesi veri analizi ve ön işlemler için, Sckit-Learn, Tensorflow, Keras gibi kütüphaneler sınıflandırma işlemleri için kullanılmaktadır. Ayrıca Matplotlib ise veri görselleştirme adımlarında kullanılmaktadır.

Genel olarak sınıflandırma için yapılan aşamalar öncelikle senaryolara ait veri setinin okunması, veri setinin sınıf dengesini sağlamak adına işlemlerden geçirilmesi, öznitelik ölçeklendirmenin sağlanması için verinin normalize edilmesi, akabinde eğitim, test ve doğrulama amaçlı kısımlara bölünmesi, eğitim seti ile algoritmanın eğitilmesi ve doğrulama veri seti ile geliştirilmesi eğitilen algoritmanın test edilmesi ve tahmin edilen sınıflar ile gerçek sonuçların kıyaslanarak algoritma performansının ölçülmesi adımlarından oluşmaktadır.

3.4.1. Algoritmalarda Kullanılan Parametreler

Makine öğrenmesi algoritmalarında ızgara arama yöntemi ile hiperparametreler arasında performansın en iyi olduğu parametre kombinasyonu bulunmaktadır. Bu parametreler şu şekildedir:

- **Ada Artırım Yöntemi**

Ada artırım yönteminde birden çok iki sınıflı probleme indirgmeden doğrudan çok sınıflı durumda ele alan Sklearn kütüphanesinde yer alan SAMME.R algoritması kul-

lanılmaktadır. Temel sınıflandırıcı olarak karar ağacı kullanılmaktadır. Maksimum tahmin edici sayısı 50 olarak seçilmektedir. Her yükseltme yinelemede her sınıflandırıcıya uygulanan ağırlık değeri ise 1 olarak belirlenmektedir.

- **Karar Ağacı Yöntemi**

Karar ağacı yöntemi için Sklearn kütüphanesi aracılığıyla CART algoritması kullanılmaktadır. Minimum maliyet-karmaşıklık budaması yapılmaktadır. Gini kriterine göre sınıflandırma yapılmaktadır.

- **Rastgele Orman Yöntemi**

Rastgele orman yönteminde karar ağacı algoritmasını temel tahmin edici sınıf olarak kullanmakta benzer şekilde Sklearn kütüphanesi üzerinden öznelikler eğitilmektedir. Daha yüksek performans elde edebilmek adına maksimum tahmin edici sayısı 100 olarak seçilmektedir.

- **Destek Vektör Makineleri Yöntemi**

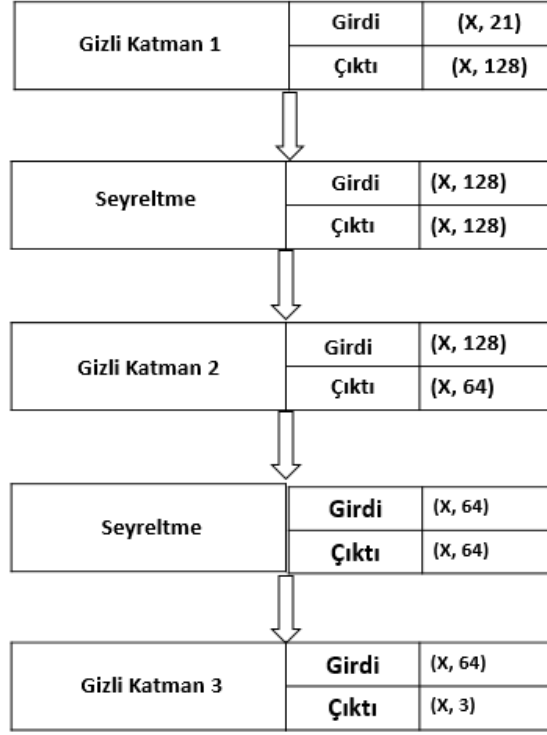
Destek vektör makineleri yönteminde ise benzer şekilde Sklearn kütüphanesi aracılığıyla kullanılan bu yöntemde libsvm algoritması temel alınmıştır. Izgara yöntemi sonucu en iyi performansın elde edildiği çekirdek tipi 'rbf' olarak seçilmektedir. Düzenleme parametresi 100, rbf için çekirdek katsayısı 1 olarak seçilmektedir.

- **Gradyan Artırım Yöntemi**

Gradyan artırım yönteminde ise benzer şekilde karar ağacı algoritmasını temel tahmin edici sınıf olarak kullanmakta benzer şekilde Sklearn kütüphanesi aracılığıyla kullanılmaktadır. Çekirdek tipi seçimi ve diğer parametreler için ızgara yöntemi sonucu maksimum tahmin edici sayısı 100 olarak seçilmektedir. Her yükseltme yinelemede her sınıflandırıcıya uygulanan ağırlık değeri ise 1 olarak belirlenmektedir.

- **Çok Katmanlı Algılayıcı Yöntemi**

Çok katmanlı algılayıcı yönteminde Sklearn kütüphanesi aracılığıyla kullanılan bu algorithmada gizli nöron sayısı 15, aktivasyon fonksiyonu relu, ağırlık optimizasyonu için çözücü algoritma 'lbfgs', L2 (düzenleme terimi) parametresi 0.0001 olarak seçilmektedir.



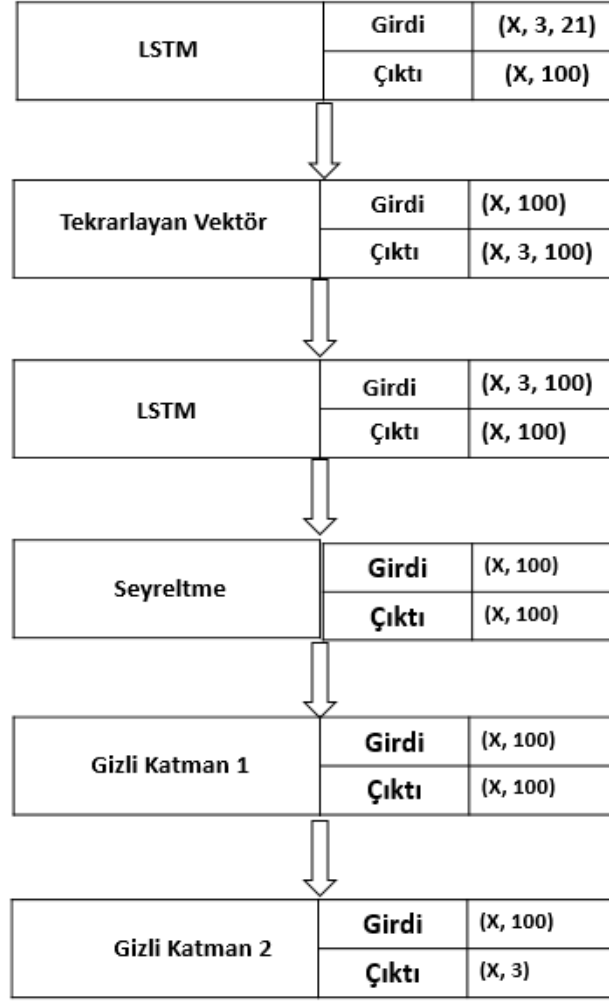
Şekil 3.4.2: Sinir Ağı Algoritma Yapısı

- **Sinir Ağları Yöntemi**

Sinir ağları yönteminde ise giriş katmanını, gizli katmanlar, bu katmanlarda bulunan nöron sayıları ayarlanarak ayrıca katmanlar arasındaki nöronlara fazla uyumlamayı azaltmak için seyreltme yapılmaktadır. Ağ yapısı Şekil 3.4.2’de görüleceği üzere 3 gizli katman ve seyreltme katmanlarından oluşmaktadır. Gizli katman 1’de ve Gizli katman 2’de tanh aktivasyon fonksiyonu ve Gizli katman 3’de ise softmax aktivasyon fonksiyonu kullanılmıştır. Eğitim setinin %20 si doğrulama verisi olacak şekilde 50 epok öğrenme işlemi gerçekleştirilmektedir.

- **LSTM Yöntemi**

LSTM yönteminde ise tekrarlayan LSTM yapısı kullanılmıştır. Ağ yapısı Şekil 3.4.3 görüleceği üzere LSTM, tekrarlayan vektör, tekrar bir LSTM, seyreltme ve gizli katman aşamalarından oluşmaktadır.



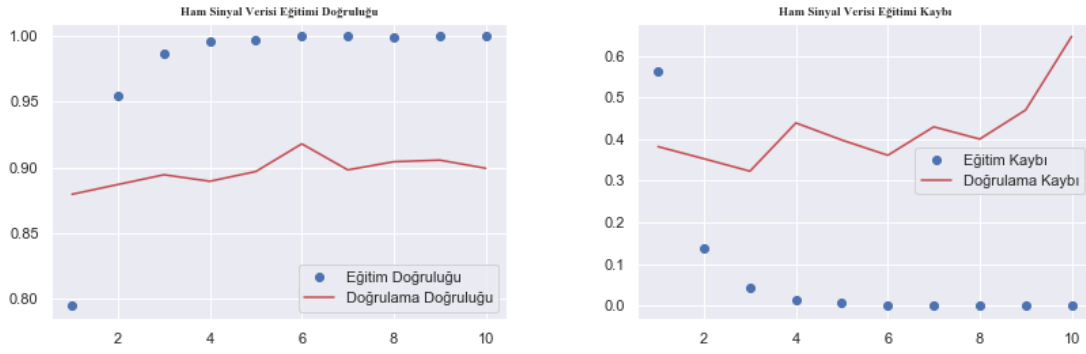
Şekil 3.4.3: LSTM Algoritma Yapısı

Gizli katman 1’de Relu aktivasyon fonksiyonu ve gizli katman 2’de ise softmax aktivasyon fonksiyonu kullanılmıştır. Eğitim setinin %20 si doğrulama verisi olacak şekilde 30 epok öğrenme işlemi gerçekleştirilmiştir.

4. İMHA DEĞERLENDİRME ALGORİTMALARININ PERFORMANS ANALİZLERİ

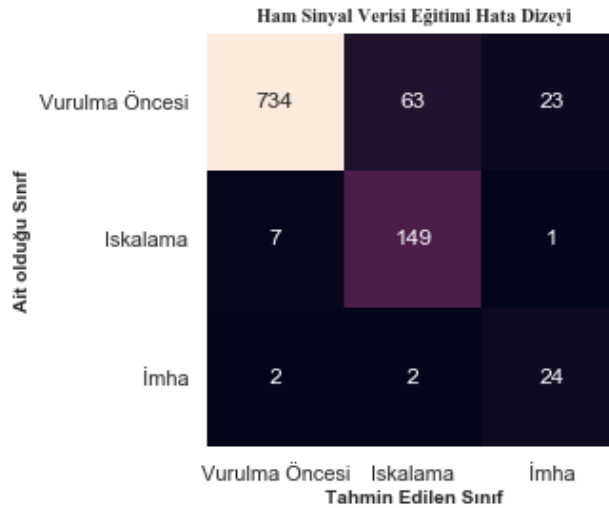
4.1. Sentetik Radar Sinyal Ham Veri Seti Performansı

Direkt ham veri seti kullanıldığında donanım gücü yetersizliği nedeniyle 200 senaryo ile öğrenme işlemi gerçekleştirilmektedir. LSTM yöntemi kullanılarak radar sinyali ham veri seti performansı aşağıdaki şekildedir. Doğrusal olmayan karmaşık bir ağ olan derin sinir ağı birçok gizli katmandan oluşmaktadır. Kendi başına veriyi nasıl çözmesi gerektiğini öğrenen ağa direkt ham veri seti verilmektedir. Fakat donanım sınırlı olduğunda gerçekçi bir uygulama için çok pahalı olma eğilimindedir.



(a) Doğruluk

(b) Kayıp



(c) Hata Dizeyi

Şekil 4.1.1: Sentetik Ham Radar Sinyal Seti Performans Analizi

Hata dizeyi incelendiğinde dengesiz veri seti dolayısıyla algoritmanın meyilli öğrendiği vu-

rulma öncesi sınıf ile imha ve ıskalama sınıflarını karıştırma olasılığının oldukça yüksek olduğu görülmektedir.

Çizelge 4.1.1: Sentetik Ham Radar Sinyal Veri Seti Performans Değerlendirme Çizelgesi

	Performans Ölçütü	Kesinlik	Duyarlılık	F1-Ölçütü
Sınıf Tipi	Vurulma Öncesi	0.90	0.99	0.94
	Iskalama	0.95	0.70	0.80
	İmha	0.86	0.50	0.63

Şekil 4.1.1 ve Çizelge 4.1.1’de verilen sonuçlar incelendiğinde sentetik ham radar sinyal setinin LSTM yöntemi ile eğitimi sonucu doğruluk oranı % 90’dır fakat uzun öğrenme süreleri ve daha çeşitli veri seti ile öğrenme işlemi için donanım yetersizliği nedenlerinden dolayı direkt ham sinyal verisini kullanmak yerine öznitelik çıkarımı yapıp önem teşkil eden özellikleri elde edip algoritmaları bu verilerle eğitmenin daha iyi sonuç vereceği öngörülmektedir.

İmha değerlendirme algoritmaları performansı ölçmek için üç farklı yöntem ile üretilmiş öznitelikleri içeren veriler çeşitli makine öğrenmesi ve derin öğrenme algoritmaları ile eğitilmektedir. Ada artırım, karar ağacı, granyan artırım, rastgele orman, çok katmanlı algılayıcı, SVM olmak üzere çeşitli makine öğrenmesi algoritmaları ve derin öğrenme algoritmalarından temel ileri beslemeli yapay sinir ağı algoritması ve LSTM algoritması denenerek hangisinin daha iyi sonuç verdiği bulunacaktır.

İmha doğası gereği anlık ya da milisaniyeler mertebesinde gerçekleşen bir olaydır. Bu nedenle veri setindeki denge gözetildiğinde vurulma öncesi anına ya da ıskalama anına ait veri yoğunluğu imha anına ait veri ile kıyaslandığında çok daha fazla olmaktadır. Bu sebeple veri dengesini sağlayan çeşitli yöntemler araştırılarak veri dengesinin sağlandığı veri setleri ile eğitim işlemi gerçekleştirilecektir. Veri dengesini sağlamak için SMOTE, ROS, RUS, Yakındaki Atma, SMOTE-ENN, SMOTE-Tomek yöntemleri kullanılmıştır.

Oluşturulan üç veri setinin (noksansız öznitelik seti, noksan öznitelik seti, belirsizlik analizi öznitelik seti) sınıf dengesi sağlanmamış halleri ve sınıf dengesinin sağlandığı yeni veri setleri ayrı ayrı belirtilen makine öğrenmesi algoritmaları ile eğitilmektedir. Hangi veri seti dengeleme yöntemi ve hangi algoritma ile en iyi imha değerlendirme performansının elde edildiği çıkarımı aşağıda verilmektedir.

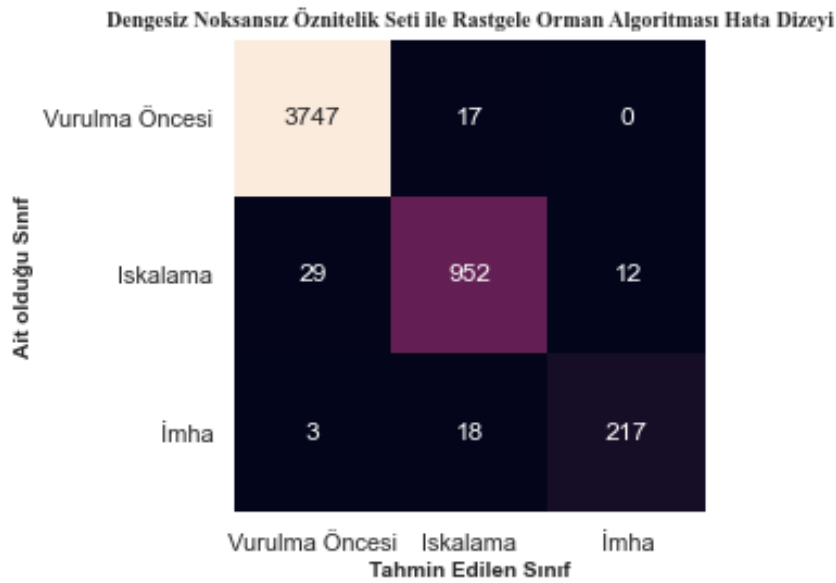
4.2. Noksansız Öznitelik Seti Performansı

Noksansız yani gürültü ya da hayalet obje içermeyen ve her şeyin mükemmel şekilde ölçülebildiği kabulü yapılan veriden çıkarılan öznitelikler ile eğitilen ve test edilen algoritmalara dair performans metrikleri EKLER bölümünde yer alan "Noksansız Öznitelik Seti Performans Çizelgeleri" verilmektedir.

Şekil 4.2.1 ve Çizelge 4.2.1 incelendiğinde sınıf dengesi algoritmaları uygulanmadan dengesiz veri seti ile en iyi çözüme ulaşan sınıflandırma algoritması rastgele orman algoritmasıdır.

Çizelge 4.2.1: Dengesiz Noksansız Öznitelik Seti ile Rastgele Orman Algoritması Performans Analizi

Rastgele Orman Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.984	Vurulma Öncesi	0.99	1.00	0.99
		Iskalama	0.96	0.96	0.96
		İmha	0.95	0.91	0.93



Şekil 4.2.1: Dengesiz Noksansız Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Sınıf dengesi algoritmaları uygulanarak sınıflar arası daha dengeli bir noksansız öznitelik

seti ile çalışıldığında en iyi sonuç veren sınıflandırma sonucunun rastgele aşırı örnekleme algoritması sonucu elde edilen öznelik seti ile eğitilen rastgele orman algoritması ile elde edildiği Çizelge 4.2.2’de verilmektedir.

ROS Uygulanmış Noksansız Öznelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Ait olduğu Sınıf	Vurulma Öncesi	3733	28	0
	Iskalama	3	3772	3
	İmha	0	0	3815
		Tahmin Edilen Sınıf		
		Vurulma Öncesi	Iskalama	İmha

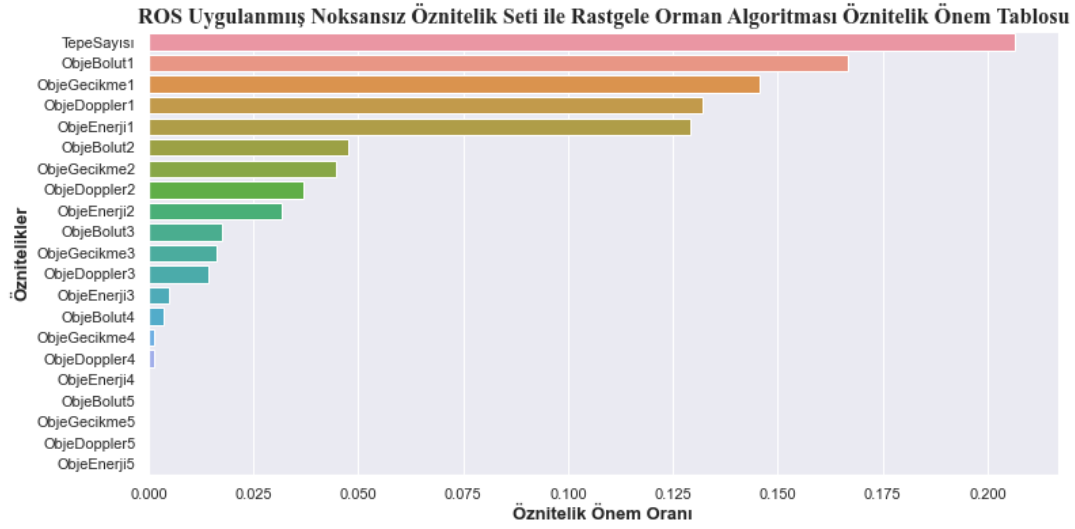
Şekil 4.2.2: ROS Uygulanmış Noksansız Öznelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Şekil 4.2.2’de görüleceği üzere Iskalama sınıfına ait olan 3 adet veri Vurulma Öncesi, 3 adet veri ise İmha sınıfına ait olarak bilinmiş, Vurulma Öncesi sınıfına ait olan 28 veri ise Iskalama olarak tahmin edilmiş. İmha sınıfı hatasız sınıflandırılabilmiştir.

Çizelge 4.2.2: ROS Uygulanmış Noksansız Öznelik Seti Rastgele Orman Algoritması Performans Analizi

Rastgele Orman					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
ROS	0.997	Vurulma Öncesi	1.00	0.99	1.00
		Iskalama	0.99	1.00	1.00
		İmha	1.00	1.00	1.00

ROS uygulanmış öznelik seti ile eğitilen rastgele orman algoritmasının eğitim sırasında öznelikleri kullanım önemi Şekil 4.2.3’de görülmektedir.



Şekil 4.2.3: ROS Uygulanmış Noksansız Öznitelik Seti Rastgele Orman Algoritması Önem Çizelgesi

Şekil 4.2.3 incelendiğinde tepe sayısı özneliğinin sınıflandırma konusunda en etkileyen nitelik olduğu görülmektedir. En yüksek enerjili objenin bulunduğu Bölüt değeri de bir sonraki etkili özneliktir. ObjEnerji4, ObjBolut5, ObjGecikme5, ObjDoppler5, ObjEnerji5 gibi öznelikler yani beşinci en düşük enerjiye sahip tepeye ait özneliklerin neredeyse sınıflandırmaya hiçbir katkısı olmamaktadır.

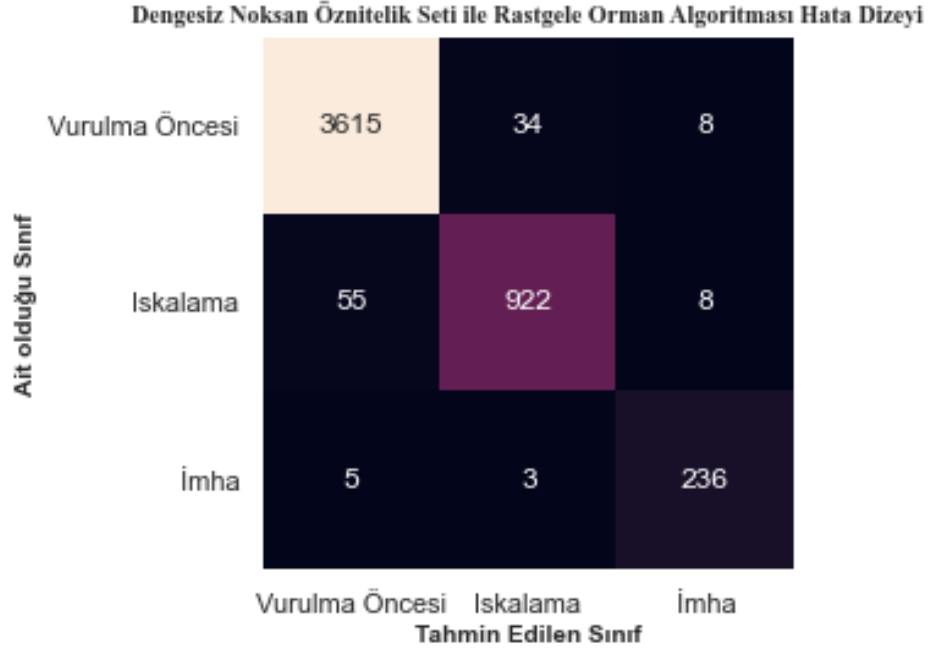
4.3. Noksan Öznitelik Seti Performansı

Noksan öznelikler ile eğitilen ve test edilen algoritmalara dair performans metrikleri Ekler bölümünde yer alan "Noksan Öznitelik Seti Performans Çizelgeleri" verilmektedir.

Şekil 4.3.1 ve Çizelge 4.3.1 incelendiğinde sınıf dengesi algoritmaları uygulanmadan dengesiz veri seti ile en iyi çözüme ulaşan sınıflandırma algoritması rastgele orman algoritmasıdır.

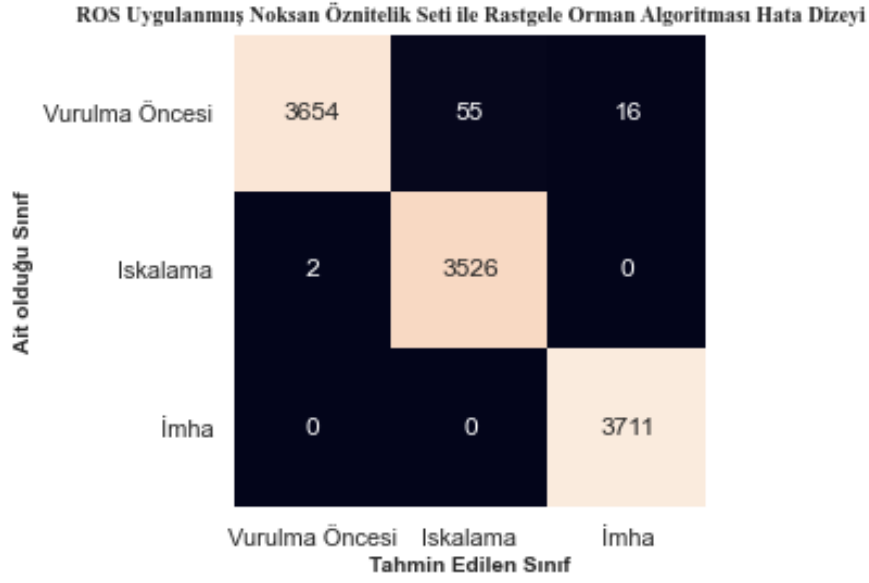
Çizelge 4.3.1: Dengesiz Noksan Öznitelik Seti Rastgele Orman Algoritması Performans Analizi

Rastgele Orman Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.977	Vurulma Öncesi	0.98	0.99	0.99
		İskalama	0.96	0.94	0.95
		İmha	0.94	0.97	0.95



Şekil 4.3.1: Dengesiz Noksan Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi

Sınıf dengesi algoritmaları uygulanarak sınıflar arası daha dengeli bir noksan öznitelik seti ile çalışıldığında en iyi sonuç veren sınıflandırma sonucunun rastgele aşırı örnekleme algoritması sonucu elde edilen öznitelik seti ile eğitilen rastgele orman algoritması ile elde edildiği Çizelge 4.3.2’de görülmektedir.



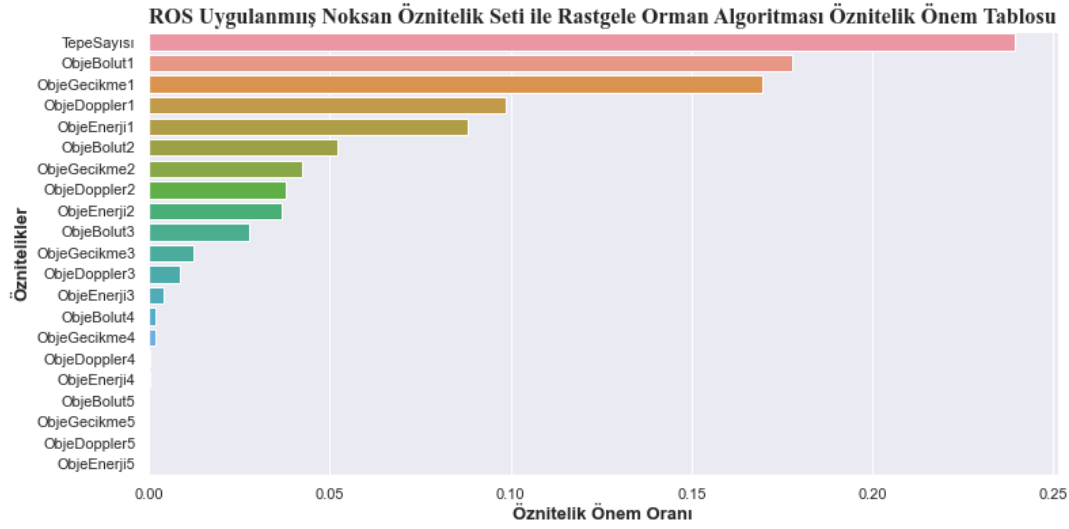
Şekil 4.3.2: ROS Uygulanmış Noksan Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi

Şekil 4.3.2’de görüleceği üzere Iskalama sınıfına ait olan 2 adet veri Vurulma Öncesi olarak bilinmiş, Vurulma Öncesi sınıfına ait olan 55 veri Iskalama olarak, 16 veri ise imha olarak tahmin edilmiş. İmha sınıfı hatasız sınıflandırılmaktadır.

Çizelge 4.3.2: ROS Uygulanmış Noksan Öznitelik Seti Rastgele Orman Algoritması Performans Analizi

Rastgele Orman Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
ROS	0.993	Vurulma Öncesi	1.00	0.98	0.99
		Iskalama	0.98	1.00	0.99
		İmha	1.00	1.00	1.00

ROS uygulanmış noksan öznitelik seti ile eğitilen rastgele orman algoritmasının eğitim sırasında öznitelikleri kullanım önemi Şekil 4.3.3’de görülmektedir.



Şekil 4.3.3: ROS Uygulanmış Noksan Öznitelik Seti ile Rastgele Orman Algoritması Önem Çizelgesi

Şekil 4.3.3 incelendiğinde tepe sayısı özneliğinin sınıflandırma konusunda en etkileyen nitelik olduğu görülmektedir. En yüksek enerjili objenin bulunduğu Bölüt değeri de bir sonraki etkili öznelik. Objebolut5, Objegecikme5, Objedoppler5, Objenerji5 gibi öznelikler yani beşinci en düşük enerjiye sahip tepeye ait öznelikler neredeyse sınıflandırmaya hiçbir katkısı olmamaktadır.

Şekil 4.3.4’de dengesiz noksan öznelik setinin farklı gürültü oranları için performans karşılaştırılması gerçekleştirilmiştir. 3 dB gürültü uygulandığı durumda imha ve ıskalama sınıflarındaki hataların çok fazla olduğu ve vurulma öncesi sınıf ile karıştığı çıkarımı yapılmaktadır. Yine benzer şekilde 5 dB gürültü oranında da algoritma performansının kötüye gittiği görülmektedir. 10 dB gürültü oranı da performansı kötü etkilemesine rağmen 3 dB ve 5 dB gürültü oranlarına kıyasla daha tolere edilebileceği yorumu yapılabilmektedir.

Dengesiz Noksan Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Ait olduğu Sınıf	Vurulma Öncesi	3628	30	10
	Iskalama	73	903	2
	İmha	7	1	256
		Vurulma Öncesi	Iskalama	İmha
		Tahmin Edilen Sınıf		

(a) Gürültüsüz Öznitelik Seti

Dengesiz 10 dB Gürültü İçeren Noksan Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Ait olduğu Sınıf	Vurulma Öncesi	3525	117	26
	Iskalama	315	634	29
	İmha	84	23	157
		Vurulma Öncesi	Iskalama	İmha
		Tahmin Edilen Sınıf		

(b) 10 dB Gürültü İçeren Öznitelik Seti

Dengesiz 5 dB Gürültü İçeren Noksan Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Ait olduğu Sınıf	Vurulma Öncesi	3491	147	30
	Iskalama	436	506	36
	İmha	97	39	128
		Vurulma Öncesi	Iskalama	İmha
		Tahmin Edilen Sınıf		

(c) 5 dB Gürültü İçeren Öznitelik Seti

Dengesiz 3 dB Gürültü İçeren Noksan Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Ait olduğu Sınıf	Vurulma Öncesi	3503	132	33
	Iskalama	540	413	25
	İmha	123	37	104
		Vurulma Öncesi	Iskalama	İmha
		Tahmin Edilen Sınıf		

(d) 3 dB Gürültü İçeren Öznitelik Seti

Şekil 4.3.4: Farklı Gürültü Değerlerinde Dengesiz Noksan Veri Seti Sınıflandırma Performansı Hata Dizeleri

Şekil 4.3.5’de rastgele aşırı örnekleme uygulanmış noksan öznitelik setinin farklı gürültü oranları için performans karşılaştırılması gerçekleştirilmiştir. Gürültüsüz noksan veri seti ile gürültü ekledikçe hatalı sınıflandırmanın arttığı görülmektedir.

Şekil 4.3.4 ve Şekil 4.3.5 beraber incelendiğinde sınıf dengelemenin gürültü ile baş etmede etkili olduğu görülüp doğru bir sınıflandırmaya pozitif bir etkisi olduğu sonucuna varılabilir.

ROS Uygulanmış Noksan Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Ait olduğu Sınıf	Vurulma Öncesi	3628	55	5
	Iskalama	4	3695	0
	İmha	0	0	3742
		Tahmin Edilen Sınıf		
		Vurulma Öncesi	Iskalama	İmha

(a) Gürültüsüz Öznitelik Seti

ROS Uygulanmış 5 dB Gürültü İçeren Noksan Öznitelik Seti Rastgele Orman ile Sınıflandırma

Ait olduğu Sınıf	Vurulma Öncesi	2864	620	204
	Iskalama	331	2842	526
	İmha	96	256	3390
		Tahmin Edilen Sınıf		
		Vurulma Öncesi	Iskalama	İmha

(b) 5 dB Gürültü İçeren Öznitelik Seti

ROS Uygulanmış 3 dB Gürültü İçeren Noksan Öznitelik Seti Rastgele Orman ile Sınıflandırma

Ait olduğu Sınıf	Vurulma Öncesi	2685	772	231
	Iskalama	375	2730	594
	İmha	132	317	3293
		Tahmin Edilen Sınıf		
		Vurulma Öncesi	Iskalama	İmha

(c) 3 dB Gürültü İçeren Öznitelik Seti

ROS Uygulanmış 1 dB Gürültü İçeren Noksan Öznitelik Seti Rastgele Orman ile Sınıflandırma

Ait olduğu Sınıf	Vurulma Öncesi	2557	896	235
	Iskalama	465	2548	686
	İmha	180	341	3221
		Tahmin Edilen Sınıf		
		Vurulma Öncesi	Iskalama	İmha

(d) 1 dB Gürültü İçeren Öznitelik Seti

Şekil 4.3.5: Farklı Gürültü Değerlerinde ROS Uygulanmış Noksan Veri Seti Sınıflandırma Performansı Hata Dizeyi

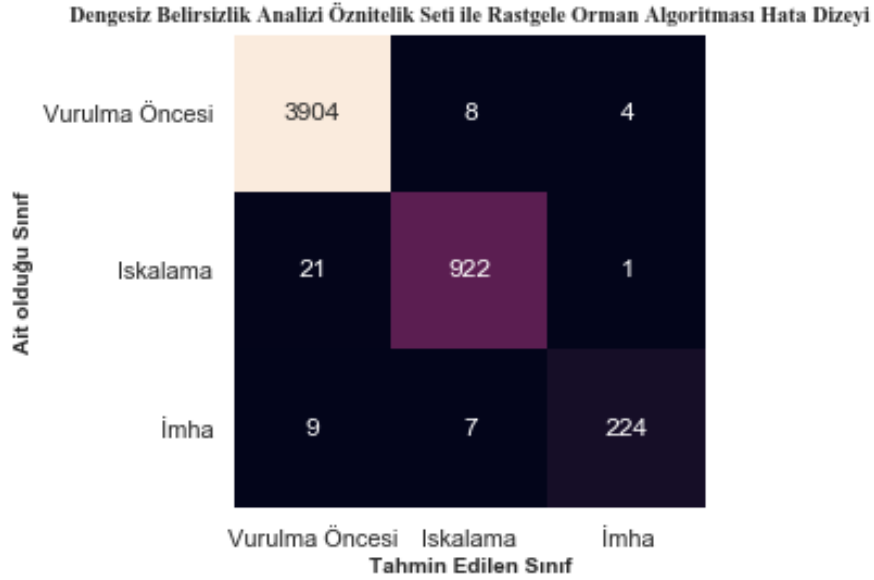
4.4. Belirsizlik Analizi Öznitelik Seti Performansı

Belirsizlik analizi sonucu elde edilen öznitelikler ile eğitilen ve test edilen algoritmalara dair performans metrikleri EKLER bölümünde yer alan "Belirsizlik Analizi Öznitelik Seti Çizelgeleri" verilmektedir.

Şekil 4.4.1 ve Çizelge 4.4.1 incelendiğinde sınıf dengesi algoritmaları uygulanmadan dengesiz veri seti ile en iyi çözüme ulaşan sınıflandırma algoritması rastgele orman algoritmasıdır.

Çizelge 4.4.1: Dengesiz Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Performans Analizi

Rastgele Orman					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.990	Vurulma Öncesi	0.98	0.99	0.99
		Iskalama	0.98	0.98	0.98
		İmha	0.98	0.93	0.96



Şekil 4.4.1: Dengesiz Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi

Sınıf dengesi algoritmaları uygulanarak sınıflar arası daha dengeli bir belirsizlik analizi öznitelik seti ile çalışıldığında en iyi sonuç veren sınıflandırma sonucunun rastgele aşırı örnekleme algoritması sonucu elde edilen öznitelik seti ile eğitilen rastgele orman algoritması ile elde edildiği görülmektedir.

ROS Uygulanmış Belirsizlik Analizi Öznitelik Seti ile Rastgele Orman Algoritması Hata Dizeyi

Ait olduğu Sınıf	Vurulma Öncesi	3888	15	10
	Iskalama	0	3907	24
	İmha	0	0	3873
		Tahmin Edilen Sınıf		
		Vurulma Öncesi	Iskalama	İmha

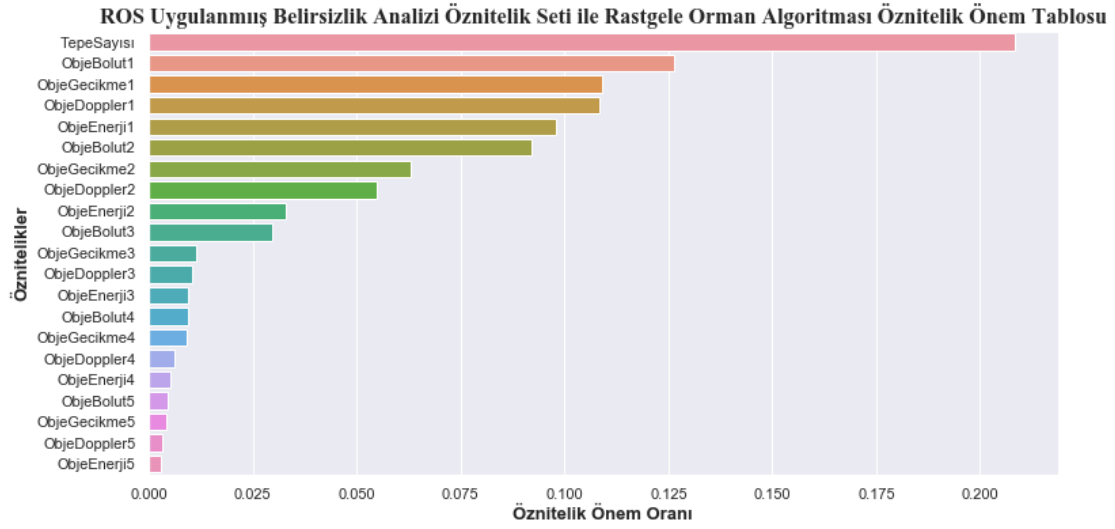
Şekil 4.4.2: ROS Uygulanmış Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Hata Dizeyi

Şekil 4.4.2’de görüleceği üzere Iskalama sınıfına ait olan 24 adet veri İmha olarak bilinmiş, Vurulma Öncesi sınıfına ait olan 15 veri Iskalama olarak, 10 veri ise imha olarak tahmin edilmiş. İmha sınıfı hatasız sınıflandırılmaktadır.

Çizelge 4.4.2: ROS Uygulanmış Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması Performans Analizi

Rastgele Orman Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
ROS	0.996	Vurulma Öncesi	1.00	0.99	1.00
		Iskalama	1.00	0.99	1.00
		İmha	0.99	1.00	1.00

ROS uygulanmış belirsizlik analizi öznitelik seti ile eğitilen rastgele orman algoritmasının eğitim sırasında öznitelikleri kullanım önemi Şekil 4.4.3’de görülmektedir.



Şekil 4.4.3: ROS Uygulanmış Belirsizlik Analizi Öznitelik Seti ile Rastgele Orman Algoritması Önem Çizelgesi

Şekil 4.4.3 incelendiğinde tepe sayısı özneliğinin sınıflandırma konusunda en etkileyen nitelik olduğu görülmektedir. En yüksek enerjili objenin bulunduğu bölüt değeri de bir sonraki etkili özneliktir. Tüm özneliklerin sınıflandırma sonucunu etkilediği görülebilmektedir. Gürültü arttıkça tüm öznelik alanlarının sınıflandırma konusunda etkili olduğu görülmektedir.

5. SONUÇLAR

Hava savunma sistemleri için bir hava tehdidi için yapılan angajman sonucunda imha edildi - edilmedi durumunun doğruluğu kritik öneme sahiptir. Bu tespit artık insan gözüyle yapılabilecek menzillerden ve büyüklüklerden çok uzaktır. Literatürde imha değerlendirme kararı için detay ve teknik yönden veri içeren çok fazla yöntem olmaması ile birlikte olan yöntemlere bakıldığında bu veriler radar, insan, optik sensör, arayıcı başlık kaynaklı olabilmektedir. Tehdit ve mühimmat hareketi incelenerek angajman sonrası vurulma öncesi ve sonrasındaki hareketlerine dair öngörüler yapılabilmektedir. Gerçekleşen hareketlere ait farklılıklar ve durumlar belirtilen sensörler tarafından gözlemlenerek yorumlanabilmektedir. Bu tespitler, hedefte bir parlamanın anlaşılması, tek hedeften radar yansıması alınmasına rağmen angajman sonrası birçok düzensiz parçadan yansıyan radar sinyallerinin alındığının anlaşılması, enerji korunumu yarasından da faydalanılarak tehdidin hız ve enerji verilerinde değişiklikler görülmesi, tehdide ait radar izinin düşmesi, mühimmat ve tehdit hareketinde değişiklik olmaması gibi tespitler yorumlanarak imha değerlendirme yapılabilmektedir. İmha kararı değerlendirmesinin yanlış verilmesi, kaynakların israf edilmesine ve mevcut tehdidin sürmesine sebep olabilmektedir. İmha kararı değerlendirmesinin doğru verilmesinin ise gerektiği anda birden fazla angajman tahsisi kararını hızlı bir şekilde vererek kaynakların verimli bir şekilde kullanımını sağlayabileceği değerlendirilmektedir.

Bu tezde ise literatürdekinden farklı olarak daha otomatize bir şekilde radar sinyallerinden çıkarılan öznitelikler ile makine öğrenmesi ve derin öğrenme tekniklerinin kullanılmasıyla tehdit ve mühimmata ait hareketlerin sınıflandırılması kararlaştırılmıştır. İmha değerlendirme kararının insan müdahalesinden bağımsız olarak verilmesi ve böylece hataların ve kaynak israflarının azaltılması hedeflenmiştir.

Hava savunma sistemlerinin imha ve ıskalama anlarına ait gerçek hayata dair veriye sahip olunmaması nedeniyle "Zaman Frekans Analiz Yöntemleri ile İmha Değerlendirme" [1] tezinde de bahsi geçen radar sinyallerini sentetik veri olarak üreten ve imha/ıskalama gibi fiziksel olayları simüle eden bir simülatör ortak çalışmalar sonucunda oluşturulmuştur. Analizler bu simülatör tarafından oluşturulan veri seti üzerinde yürütülmüştür. Simülatörde radar sinyallerinin oluşumu konusunda Lock Heed Martin firmasının RRP-117 radarı [44], tehdide ait kütle ve hız konusunda F16 savaş uçağı, mühimmata ait özelliklerde ise S400 ve S300 fü-

zeleri [46] [47] referans olarak alınmıştır. Senaryolar tehdit ve mühimmatın hızları, izlediği yollar, imha sonrası açığa çıkan enerji, ısıya dönüşen enerji, parçacık sayısı gibi durumlar her senaryo özelinde rastgele olarak oluşturulmuştur. Bu senaryolar belirlenirken tehdidin ve mühimmatın manevra yapmadığı doğrusal hareket ettiği senaryolar kurgulanmıştır.

Ham radar sinyalleri ile sınıflandırma işlemi yapılmayarak öznitelik çıkarım işlemlerinin yapılması ve öznitelik verileri kullanılarak öğrenme ve sınıflandırma yapılması kararlaştırılmıştır. Çünkü radar sinyal verilerinin çok fazla bilgi içermesi ve bu verilerin aşırı uyumlamaya açık olması, eğitim sürecinde donanımın yetersiz kalması ve eğitim süresinin uzun sürmesine gibi dezavantajlar mevcuttur. Öznitelik çıkarımları yapılarak öğrenme işlemi bu veriler üzerinden sürdürülmüştür. Fiziksel ortamı ve radar sinyallerini simüle eden bu simülatör sonucunda üç farklı şekilde oluşturulan öznitelik setleri 500'ü imha 500'ü ıskalama senaryosu olmak üzere toplamda her birinden 1000 senaryo olacak şekilde makine öğrenmesi ve derin öğrenme algoritmalarına ayrı ayrı girdi olarak sağlanmıştır. Bu öznitelikler Vurulma Öncesi, İskalama ve İmha olmak üzere üç farklı sınıftan birine ait olacak şekilde etiketlenmiştir.

Bu öznitelik setlerinden noksan öznitelik seti ve noksansız öznitelik setinin oluşturulma amacı asıl gerçek hayattaki özniteliklere daha yakın olarak gürültü içerecek şekilde elde edilen ve analizi gerçekleştirilen Belirsizlik Analizi öznitelik setinin [1] kullanımı öncesinde yapay zekâ algoritmalarının başka veri setleri ile eğitilmesini ve performans değerlendirmesinin yapılmasını sağlamaktır. Böylece gürültüsüz (noksansız öznitelik seti), gürültü içeren (noksan öznitelik seti) ve gerçeğe yakın (belirsizlik analizi öznitelik seti) üç ayrı veri seti ile sınıflandırma algoritmalarının performansı ölçülmüş olmaktadır. Sınıflandırmanın doğruluğu, gerçek hayatta sensörlerin tespit kabiliyetiyle doğrudan ilişkiliyken bu çalışmada simülasyonda belirlenen senaryolardan çıkarılan özniteliklerin içeriğiyle doğrudan ilişkilidir.

İmha değerlendirme doğası gereği anlık ya da milisaniyeler mertebesinde gerçekleşen bir olaydır. Bu nedenle etiketlenen senaryolarda sınıflara ait veri setindeki denge gözetildiğinde vurulma öncesi anına ya da ıskalama anına ait veri yoğunluğu imha anına ait veri ile kıyaslandığında çok daha fazla olmaktadır. Bu sebeple sınıf dengesini sağlayan çeşitli yöntemler araştırılarak sınıf dengesinin sağlandığı öznitelik setleri ile eğitim işlemi gerçekleştirilmiştir. Sınıf dengesini sağlamak için SMOTE, ROS, RUS, Yakındakini Atma, SMOTE-ENN, SMOTE-Tomek yöntemleri kullanılmıştır.

Dengesiz ve dengeli hale getirilen öznitelik setleri sırası ile girdi olarak kullanılarak imha

değerlendirme algoritmaları test edilmiştir. Makine öğrenmesi algoritmalarından ada artırımı, gradyan artırımı, karar ağacı, rastgele orman, destek vektör makineleri, çok katmanlı algılayıcı, derin öğrenme algoritmalarından ise sinir ağları ve LSTM kullanılmıştır. Izgara arama yöntemi ile sınıflandırma algoritmalarının en yüksek performans ile çalışmasını sağlayan hiperparametre eniyileme işlemleri yapılmıştır.

Noksansız, noksan ve belirsizlik analizi öznitelik setlerine ayrı ayrı olmak üzere 6 tane (SMOTE, ROS, RUS, Yakındakini Atma, SMOTE-ENN, SMOTE-TOMEK) sınıf dengeleme yöntemi uygulanmıştır. Dengesiz veri seti hali ile birlikte her öznitelik seti toplamda 7 farklı veri tipine sahip olabilmektedir. Noksansız, noksan ve belirsizlik analizi öznitelik setleri hep beraber düşünüldüğünde toplamda 21 adet farklı veri setine sahip olunmaktadır. 8 farklı sınıflandırma algoritması ile her veri seti tipi ayrı ayrı sınıflandırılarak toplamda $21 \times 8 = 168$ sınıflandırma işlemi gerçekleştirilmiştir. Bu çalışmaya ait detaylar EKLER bölümünde verilmektedir.

Performans metrikleri kıyaslandığında Noksansız dengesiz öznitelik seti özelinde en iyi iki yöntem %98.4 ile rastgele orman ve %98.9 LSTM yöntemi olarak görülmektedir. Sınıf dengesi yöntemlerinden de en iyi iki sonucun %99.7 ile rastgele aşırı örnekleme uygulanmış veri setinin rastgele orman ve %99.6 ile gradyan artırımı ile sınıflandırılması sonucu elde edildiği görülmektedir.

Noksan dengesiz öznitelik seti özelinde en iyi iki yöntem %97.9 ile LSTM ve %97.7 ile rastgele orman yöntemi olarak görülmektedir. Sınıf dengesi yöntemlerinden de en iyi ilk sonucun %99.3 ile rastgele aşırı örnekleme uygulanmış veri setinin rastgele orman yöntemi ile sınıflandırılması sonucu elde edildiği görülmektedir. En iyi ikinci sonucun %98.5 ile rastgele aşırı öğrenme uygulanmış veri setinin gradyan artırımı yöntemi ile sınıflandırılması sonucu elde edildiği görülmektedir. %98.2 ile rastgele aşırı öğrenme uygulanmış veri setinin SVM ile sınıflandırılması da benzer bir sonuç vermektedir. Bu çalışmalara ek olarak noksan öznitelik setine ait hem dengesiz hem de rastgele aşırı örneklenmiş veri setleri farklı değerlerde gürültülere maruz bırakılmış ve daha sonrasında rastgele orman algoritması ile performans analizleri gerçekleştirilmiştir. Sınıf dengeleme yöntemlerinin gürültü ile baş etme konusunda da sınıflandırmaya olumlu bir etkisi olduğu sonucuna varılmaktadır.

Belirsizlik analizi dengesiz öznitelik seti özelinde en iyi iki yöntem %99.0 ile rastgele orman ve %98.0 ile LSTM yöntemi olarak görülmektedir. Sınıf dengesi yöntemlerinden de en iyi

ilk sonucun %99.6 ile rastgele aşırı örnekleme uygulanmış veri setinin rastgele orman yöntemi ile sınıflandırılması sonucu elde edildiği görülmektedir. En iyi ikinci sonucun %98.8 ile SMOTE-ENN uygulanmış veri setinin rastgele orman yöntemi ile sınıflandırılması sonucu elde edildiği görülmektedir. Ayrıca %98.3 başarı oranı ile SMOTE-ENN uygulanmış veri setinin LSTM yöntemi ile sınıflandırılmasının da benzer başarı oranında sonuç verdiği görülmektedir.

Tüm veri setleri incelenip genel bir sonuca varılmak istenildiğinde sınıf dengesi yöntemlerinin karar ağacı ve çok katmanlı algılayıcı algoritmaları dışında sınıflandırma performansına olumlu bir etkisinin olduğu görülmektedir. Her veri setinde de sınıf dengesinin sağlanıyor olması sınıflandırma performansını artırmaktadır. Performans analizlerine bakıldığında üç ayrı öznitelik seti için de ROS uygulanmış öznitelik setinin girdi olduğu rastgele orman algoritmasının %99 üzerinde doğruluk oranı ile en iyi sınıflandırmayı gerçekleştirdiği sonucuna varılmaktadır. Her öznitelik seti için Ada Artırım yönteminin uygun olmadığı başarı oranı en düşük algoritma olduğu sonucuna varılmaktadır. LSTM, SVM, gradyan artırımı gibi yöntemlerin de başarı olarak rastgele orman algoritmasına yakın olduğu görülmüştür. Sinir ağları ve LSTM yöntemi kıyaslandığında LSTM'in daha başarılı sonuç verdiği görülmektedir. Özniteliklerin geçmişi de içerecek şekilde sıralı bir dizi oluşturularak verilmesinin performansı artırdığı yorumu yapılabilmektedir. Fakat LSTM, SVM algoritmaları ile rastgele orman algoritması kıyaslandığında rastgele orman algoritmasının daha hızlı oluşu, gürültülü veri ile daha başarılı sonuç vermesi ve gürültü oranları farklı her öznitelik tipinde iyi sonuç veriyor olması gibi sebeplerden imha değerlendirme algoritmalarında kullanımı daha uygun olacağı sonucuna varılmaktadır.



Şekil 5.0.1: İmha Değerlendirme Algoritması

Bu tez çalışması sonucunda hedeflendiği şekilde yapay zekâ algoritmaları ile radar sinyalleri üzerinden çıkarılan öznitelikler kullanılarak başarılı bir şekilde imha değerlendirme yapıla-

bileceđi gör÷lmektedir. Bu tez alıřması sonunda varılan kanı Őekil 5.0.1 gör÷leceđi üzere belirtilen akıřa göre öncelikle imha, ıskalama, vurulma öncesi gibi anlara ait radar sinyallerinin elde edilerek, belirsizlik analizi yöntemi ile öznitelikleri ıkarılmalıdır. Daha sonrasında sınıf dengesinin sađlanması için rastgele aşırı örnekleme algoritması uygulanmalıdır. Veri dengesi sađlandıktan sonra rastgele orman algoritması ile eđitim ve sınıflandırma işlemleri gerçekleştirilmelidir.

Gelecekteki alıřmalarda manevra yapan birden fazla tehdit ve mühimmat ile simülasyonların gerçekleştirilmesi, gerçek radar verileri ile alıřılması, algoritmanın dođruluđunun teyit edilmesi, bu özniteliklere ek olarak füze telemetresinden elde edilen özniteliklerin de kullanımını gibi alıřmalar planlanabilir.

KAYNAKLAR

- [1] N. Uzun. Zaman Frekans Analiz Yöntemleri İle İmha Değerlendirme. Yüksek Lisans Tezi, Hacettepe Üniversitesi, Ankara, **2022**.
- [2] ASELSAN A.Ş., HİSAR-O Orta İrtifa Hava Savunma Füze Sistemi, <https://www.aselsan.com.tr/tr/cozumlerimiz/hava-ve-fuze-savunma-sistemleri/hava-ve-fuze-savunma-sistemleri/hisaro-orta-irtifa-hava-savunma-fuze-sistemi>, **(Erişim Tarihi: 12 Ekim 2021)**.
- [3] I. Jordanov, N. Petrov, and A. Petrozziello. Supervised Radar Signal Classification. International Joint Conference on Neural Networks (IJCNN), pages 1464–1471, **2016**.
- [4] J. C. Briones, B. Flores, and R. Cruz-Cano. Multi-Mode Radar Target Detection and Recognition Using Neural Networks. International Journal of Advanced Robotic Systems, 9(5):177, **2012**.
- [5] B. Jokanovic and M. Amin. Fall Detection Using Deep Learning in Range-Doppler Radars. IEEE Transactions on Aerospace and Electronic Systems, 54(1):180–189, **2018**.
- [6] E. T. Kouba and K. Usaf. Recurrent Neural Networks For Radar Target Identification. Ph.D. thesis, Air University, Faculty of the School of Engineering of the Air Force Institute of Technology, **1992**.
- [7] J. E. Gray, J. Richard, and P. Brastrom. The Problem of Kill Assessment: A Challenge for Radar Tracking Systems. Proceedings of the 2001 IEEE Radar Conference, pages 332–337, **2001**.
- [8] R. E. Erlandson, J. C. Taylor, C. H. Michaelis, J. L. Edwards, R. C. Brown, P. K. Swaminathan, C. K. Kumar, C. B. Hargis, A. C. Goldberg, E. M. Klatt, G. L. O’Marr. Development of Kill Assessment Technology for Space-Based Applications. Johns Hopkins Apl Technical Digest, Vol. 29, **2010**.
- [9] J. E. Gray and P. Brastrom. The Problem of Kill Assessment Part II The Doppler Effect and Exo-Atmospheric Kill Assessment. Proceedings of the 2002 IEEE Radar Conference, pages 85–91, **2002**.

- [10] S. K. Chappell, A. T. Alouani, T. R. Rice, and J. E. Gray. Partial Determination Of Particle Motion Using Artificial Neural Networks. [Proceedings 1992] The First IEEE Conference on Control Applications, pages 559–564, **1992**.
- [11] A. A. Pramudita, Lukas, and Edwar. Time and Frequency Domain Feature Extraction Method of Doppler Radar for Hand Gesture Based Human to Machine Interface. Progress In Electromagnetics Research C, 98:83–96, **2020**.
- [12] V. Kizhakkal. Pulsed Radar Target Recognition Based On Micro-Doppler Signatures Using Wavelet Analysis. Yüksek Lisans Tezi, The Ohio State University, Columbus, Ohio, **2013**.
- [13] K. L. Priddy and P. E. Keller. Artificial Neural Networks, volume TT68. SPIE Press, Bellingham, Washington USA, **2005**.
- [14] E. Alpaydm. Introduction To Machine Learning. The MIT Press, Cambridge Massachusetts London, England, third edition, **2014**.
- [15] J. Kozak. Decision Tree and Ensemble Learning Based on Ant Colony Optimization, volume 781 of Studies in Computational Intelligence. Springer International Publishing, Warsaw, Poland, **2019**.
- [16] S. Singh and P. Gupta. Comparative Study ID3, CART And C4.5 Decision Tree Algorithm: A Survey. International Journal of Advanced Information Science and Technology (IJAIST), 27, **2014**.
- [17] R. Timofeev. Classification and Regression Trees (CART) Theory and Applications. Yüksek Lisans Tezi, CASE - Center of Applied Statistics and Economics, Humboldt University, Berlin, **2004**.
- [18] P. Lang, X. Fu, M. Martorella, J. Dong, R. Qin, X. Meng, and M. Xie. A Comprehensive Survey of Machine Learning Applied to Radar Signal Processing. arXiv: Signal Processing, **2020**.
- [19] J. Nikulski. The Ultimate Guide to AdaBoost, random forests and XGBoost, <https://towardsdatascience.com/the-ultimate-guide-to-adaboost-random-forests-and-xgboost-7f9327061c4f>, (**Eriřim Tarihi: 7 Aralık 2021**).

- [20] C. M. Bishop. Pattern Recognition And Machine Learning. Springer, New York, second edition, **2006**.
- [21] E. Bauer and R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. Kluwer Academic Publishers. Manufactured in The Netherlands., 36(1/2):105–139, **1999**.
- [22] T. Hastie, S. Rosset, J. Zhu, and H. Zou. Multi-class AdaBoost. Statistics and Its Interface, 2(3):349–360, **2006**.
- [23] O. González-Recio, J. A. Jiménez-Montero, and R. Alenda. The Gradient Boosting Algorithm And Random Boosting For Genome-Assisted Evaluation In Large Data Sets. Journal of Dairy Science, 96:614–624, **2013**.
- [24] A. Pal, <https://medium.com/swlh/gradient-boosting-trees-for-classification-a-beginners-guide-596b594a14ea>, (**Erişim Tarihi: 8 Aralık 2021**).
- [25] L. Breiman. The Random Subspace Method For Constructing Decision Forests. Kluwer Academic Publishers. Manufactured in The Netherlands., 45(1):5–32, **2001**.
- [26] J.Singh , Random Forest: Pros and Cons, <https://medium.datadriveninvestor.com/random-forest-pros-and-cons-c1c42fb64f04>, (**Erişim Tarihi: 8 Aralık 2021**).
- [27] C.-W. Hsu and C.-J. Lin. A Comparison Of Methods For Multiclass Support Vector Machines. IEEE Transactions on Neural Networks, 13(2):415–425, **2002**.
- [28] D. J. C. MacKay. Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge UK and New York, **2003**.
- [29] F. Rosenblatt. The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain. Psychological Review, 65(6):386–408, **1958**.
- [30] L. Noriega. Multilayer Perceptron Tutorial. School of Computing Staffordshire University, **2005**.
- [31] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. Neural Computation, 9(8):1735–1780, **1997**.

- [32] J. Brownlee, Long Short-Term Memory Networks With Python - Machine Learning Mastery: Develop Sequence Prediction Models With Deep Learning. Jason Brownlee, first edition, **2017**.
- [33] Anonim, Long Short Term Memory, Wikipedia, https://tr.wikipedia.org/wiki/Long_short-term_memory, (**Erişim Tarihi: 8 Aralık 2021**).
- [34] M. Rupp. Artificial Intelligence: A Short Overview of Classification Techniques for RADAR Targets through Neural Networks (1/2), **2020**.
- [35] J. Bergstra and Y. Bengio. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research 13, The Journal of Machine Learning Research Volume 133/1/2012:281–305, **2012**.
- [36] P. M. Lerman. Fitting Segmented Regression Models by Grid Search. Journal of the Royal Statistical Society. Series C (Applied Statistics), 29(1):77, **1980**.
- [37] H. He and Y. Ma. Imbalanced Learning, Foundations, Algorithms, And Applications. John Wiley & Sons Inc, Hoboken New Jersey, **2013**.
- [38] G. Batista, R. C. Prati, and M. C. Monard. A Study Of The Behavior Of Several Methods For Balancing Machine Learning Training Data. ACM SIGKDD Explorations Newsletter, 6(1):20–29, **2004**.
- [39] J. Wang, M. Xu, H. Wang, and J. Zhang. Classification of Imbalanced Data by Using the SMOTE Algorithm and Locally Linear Embedding. ICSP2006 Proceedings, Signal Processing, 2006 8th International Conference, 3, **2006**.
- [40] Anonim, Resampling Strategies For Imbalanced Datasets, <https://www.kaggle.com/rafjaa/resampling-strategies-for-imbalanced-datasets>, (**Erişim Tarihi: 27 Aralık 2021**).
- [41] M. Zeng, B. Zou, F. Wei, X. Liu, and L. Wang. Effective Prediction Of Three Common Diseases By Combining SMOTE With Tomek Links technique for imbalanced medical data. IEEE International Conference of Online Analysis and Computing Science (ICOACS), **2016**.

- [42] Brownlee J. Undersampling Algorithms for Imbalanced Classification, <https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/>, **(Eriřim Tarihi: 10 Ekim 2021)**.
- [43] M. Vakili, M. Ghamsari, and M. Rezaei. Performance Analysis And Comparison of Machine and Deep Learning Algorithms for IoT Data Classification. arXiv: Computer Science, Mathematics, **2020**.
- [44] C. Wolff, Lockheed Martin RRP-117, Seek Igloo, <https://www.radartutorial.eu/19.kartei/02.surv/karte007.tr.html>, **(Eriřim Tarihi: 6 Aralık 2021)**.
- [45] The F-16 Fighting Falcon, <https://www.lockheedmartin.com/en-us/products/f-16.html>, **(Eriřim Tarihi: 15 Kasım 2021)**.
- [46] Anonim, S-400 Missile System, https://en.wikipedia.org/wiki/S-400_missile_system, **(Eriřim Tarihi: 9 Kasım 2021)**.
- [47] Anonim, S-300 Missile System, https://en.wikipedia.org/wiki/S-300_missile_system, **(Eriřim Tarihi: 27 Ekim 2021)**.
- [48] P. Swerling. Probability of detection for fluctuating targets. IEEE Transactions on Information Theory, 6(2):269–308, **1960**.
- [49] X. Song, W. D. Blair, P. Willett, and S. Zhou. Dominant-plus-Rayleigh Models for RCS: Swerling III/IV versus Rician. IEEE Transactions on Aerospace and Electronic Systems, 49(3):2058–2064, **2013**.

6. EKLER

6.1. EK 1 - Performans Çizelgeleri

6.1.1. Noksansız Öznitelik Seti Performans Çizelgeleri

Çizelge 6.1.1: Noksansız Öznitelik Seti Ada Artırım Algoritması ile Sınıflandırma Performans Analizi

Ada Artırım Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.785	Vurulma Öncesi	0.88	0.88	0.88
		Iskalama	0.69	0.42	0.53
		İmha	0.30	0.76	0.43
SMOTE	0.717	Vurulma Öncesi	0.94	0.62	0.75
		Iskalama	0.65	0.717	0.74
		İmha	0.66	0.68	0.67
ROS	0.703	Vurulma Öncesi	0.81	0.61	0.7
		Iskalama	0.70	0.75	0.73
		İmha	0.64	0.75	0.69
RUS	0.684	Vurulma Öncesi	0.86	0.49	0.63
		Iskalama	0.72	0.79	0.75
		İmha	0.57	0.78	0.66
YA	0.759	Vurulma Öncesi	0.73	0.84	0.78
		Iskalama	0.69	0.53	0.60
		İmha	0.84	0.92	0.88
SMOTE-ENN	0.760	Vurulma Öncesi	0.95	0.78	0.85
		Iskalama	0.65	0.86	0.74
		İmha	0.75	0.65	0.70
SMOTE-TOMEK	0.589	Vurulma Öncesi	0.64	0.44	0.52
		Iskalama	0.62	0.64	0.63
		İmha	0.53	0.68	0.60

Çizelge 6.1.2: Noksansız Öznitelik Seti Karar Ağacı Algoritması ile Sınıflandırma Performans Analizi

Karar Ağacı Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.960	Vurulma Öncesi	0.98	0.99	0.98
		Iskalama	0.93	0.89	0.91
		İmha	0.84	0.77	0.80
SMOTE	0.945	Vurulma Öncesi	0.98	0.94	0.96
		Iskalama	0.91	0.94	0.92
		İmha	0.96	0.96	0.96
ROS	0.966	Vurulma Öncesi	0.99	0.96	0.98
		Iskalama	0.95	0.95	0.95
		İmha	0.96	0.99	0.97
RUS	0.928	Vurulma Öncesi	0.96	0.94	0.95
		Iskalama	0.92	0.90	0.91
		İmha	0.91	0.95	0.93
YA	0.888	Vurulma Öncesi	0.92	0.90	0.91
		Iskalama	0.85	0.83	0.84
		İmha	0.90	0.95	0.92
SMOTE-ENN	0.95	Vurulma Öncesi	0.99	0.94	0.96
		Iskalama	0.91	0.95	0.93
		İmha	0.95	0.97	0.96
SMOTE-TOMEK	0.947	Vurulma Öncesi	0.98	0.94	0.96
		Iskalama	0.91	0.93	0.92
		İmha	0.95	0.97	0.96

Çizelge 6.1.3: Noksansız Öznitelik Seti Gradyan Artırım Algoritması ile Sınıflandırma Performans Analizi

Grandyan Artırım Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.953	Vurulma Öncesi	0.98	0.99	0.99
		Iskalama	0.93	0.86	0.89
		İmha	0.63	0.74	0.68
SMOTE	0.979	Vurulma Öncesi	1.00	0.99	0.99
		Iskalama	0.98	0.96	0.97
		İmha	0.96	0.99	0.98
ROS	0.996	Vurulma Öncesi	1.00	0.99	0.99
		Iskalama	0.99	1.00	0.99
		İmha	1.00	1.00	1.00
RUS	0.955	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.95	0.93	0.94
		İmha	0.93	0.97	0.95
YA	0.947	Vurulma Öncesi	0.95	0.96	0.95
		Iskalama	0.92	0.93	0.92
		İmha	0.97	0.95	0.96
SMOTE-ENN	0.982	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.98	0.97	0.97
		İmha	0.97	0.98	0.98
SMOTE-TOMEK	0.976	Vurulma Öncesi	1.00	0.99	0.99
		Iskalama	0.97	0.95	0.96
		İmha	0.96	0.98	0.97

Çizelge 6.1.4: Noksansız Öznitelik Seti Rastgele Orman Algoritması ile Sınıflandırma Performans Analizi

Rastgele Orman Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.984	Vurulma Öncesi	0.99	1.00	0.99
		Iskalama	0.96	0.96	0.96
		İmha	0.95	0.91	0.93
SMOTE	0.986	Vurulma Öncesi	1.00	0.99	0.99
		Iskalama	0.98	0.97	0.98
		İmha	0.98	0.99	0.98
ROS	0.997	Vurulma Öncesi	1.00	0.99	1.00
		Iskalama	0.99	1.00	1.00
		İmha	1.00	1.00	1.00
RUS	0.967	Vurulma Öncesi	1.00	0.97	0.98
		Iskalama	0.96	0.95	0.95
		İmha	0.95	0.98	0.96
YA	0.956	Vurulma Öncesi	0.95	0.96	0.96
		Iskalama	0.95	0.92	0.94
		İmha	0.97	0.99	0.98
SMOTE-ENN	0.990	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.99	0.98	0.99
		İmha	0.98	1.00	0.99
SMOTE-TOMEK	0.986	Vurulma Öncesi	1.00	0.99	1.00
		Iskalama	0.99	0.97	0.98
		İmha	0.98	0.99	0.99

Çizelge 6.1.5: Noksansız Öznitelik Seti Çok Katmanlı Algılayıcı Algoritması ile Sınıflandırma Performans Analizi

Çok Katmanlı Algılayıcı					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.982	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.96	0.95	0.96
		İmha	0.80	0.85	0.83
SMOTE	0.966	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.98	0.92	0.95
		İmha	0.93	0.98	0.95
ROS	0.959	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.98	0.89	0.94
		İmha	0.90	0.99	0.94
RUS	0.955	Vurulma Öncesi	1.00	0.99	0.99
		Iskalama	0.97	0.90	0.94
		İmha	0.89	0.98	0.93
YA	0.904	Vurulma Öncesi	0.99	1.00	0.99
		Iskalama	0.92	0.79	0.85
		İmha	0.81	0.93	0.86
SMOTE-ENN	0.966	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.98	0.92	0.95
		İmha	0.92	0.98	0.95
SMOTE-TOMEK	0.963	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.98	0.91	0.94
		İmha	0.92	0.99	0.95

Çizelge 6.1.6: Noksansız Öznitelik Seti Destek Vektör Makineleri ile Sınıflandırma Performans Analizi

Destek Vektör Makineleri Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.984	Vurulma Öncesi	0.99	1.00	1.00
		Iskalama	0.96	0.96	0.96
		İmha	0.92	0.89	0.91
SMOTE	0.982	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.98	0.96	0.97
		İmha	0.97	0.99	0.98
ROS	0.990	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.99	0.98	0.99
		İmha	0.98	0.99	0.99
RUS	0.952	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.95	0.92	0.93
		İmha	0.91	0.98	0.95
YA	0.942	Vurulma Öncesi	0.97	0.99	0.98
		Iskalama	0.91	0.92	0.91
		İmha	0.94	0.92	0.93
SMOTE-ENN	0.984	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.99	0.96	0.98
		İmha	0.97	0.99	0.98
SMOTE-TOMEK	0.981	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.99	0.96	0.97
		İmha	0.96	0.99	0.97

Çizelge 6.1.7: Noksansız Öznitelik Seti Sinir Ağları Algoritması ile Sınıflandırma Performans Analizi

Sinir Ağları Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.979	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.94	0.95	0.95
		İmha	0.81	0.83	0.82
SMOTE	0.969	Vurulma Öncesi	1.00	0.99	1.00
		Iskalama	0.97	0.94	0.95
		İmha	0.94	0.98	0.96
ROS	0.962	Vurulma Öncesi	1.00	0.99	0.99
		Iskalama	0.98	0.90	0.94
		İmha	0.91	0.99	0.95
RUS	0.939	Vurulma Öncesi	0.99	0.98	0.99
		Iskalama	0.95	0.87	0.91
		İmha	0.87	0.97	0.92
YA	0.899	Vurulma Öncesi	1.00	0.96	0.98
		Iskalama	0.84	0.87	0.86
		İmha	0.86	0.87	0.87
SMOTE-ENN	0.972	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.99	0.93	0.96
		İmha	0.94	0.99	0.96
SMOTE-TOMEK	0.966	Vurulma Öncesi	1.00	1.00	1.00
		Iskalama	0.98	0.91	0.95
		İmha	0.92	0.99	0.95

Çizelge 6.1.8: Noksansız Öznitelik Seti LSTM Algoritması ile Sınıflandırma Performans Analizi

LSTM Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.984	Vurulma Öncesi	0.99	1.00	0.99
		Iskalama	0.98	0.95	0.96
		İmha	0.90	0.98	0.94
SMOTE	0.988	Vurulma Öncesi	1.00	0.99	1.00
		Iskalama	0.99	0.97	0.98
		İmha	0.97	1.00	0.99
ROS	0.989	Vurulma Öncesi	0.99	1.00	0.99
		Iskalama	0.99	0.98	0.98
		İmha	0.99	1.00	0.99
RUS	0.967	Vurulma Öncesi	0.99	0.98	0.98
		Iskalama	0.96	0.94	0.95
		İmha	0.95	0.98	0.96
YA	0.948	Vurulma Öncesi	0.98	0.98	0.98
		Iskalama	0.97	0.87	0.92
		İmha	0.89	1.00	0.94
SMOTE-ENN	0.989	Vurulma Öncesi	0.99	1.00	0.99
		Iskalama	1.00	0.97	0.98
		İmha	0.98	1.00	0.99
SMOTE-TOMEK	0.988	Vurulma Öncesi	0.99	1.00	0.99
		Iskalama	0.96	0.96	0.96
		İmha	0.95	0.91	0.93

6.1.2. Noksan Öznitelik Seti Performans Çizelgeleri

Çizelge 6.1.9: Noksan Öznitelik Seti Ada Artırım Algoritması Performans Analizi

Ada Artırım Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.769	Vurulma Öncesi	0.83	0.87	0.85
		Iskalama	0.44	0.37	0.4
		İmha	0.89	0.86	0.88
SMOTE	0.788	Vurulma Öncesi	0.78	0.66	0.71
		Iskalama	0.69	0.73	0.71
		İmha	0.90	0.98	0.94
ROS	0.736	Vurulma Öncesi	0.64	0.67	0.66
		Iskalama	0.62	0.53	0.57
		İmha	0.91	0.99	0.95
RUS	0.776	Vurulma Öncesi	0.75	0.66	0.70
		Iskalama	0.66	0.70	0.68
		İmha	0.92	0.97	0.94
YA	0.500	Vurulma Öncesi	0.44	0.45	0.44
		Iskalama	0.32	0.35	0.33
		İmha	0.85	0.73	0.79
SMOTE-ENN	0.864	Vurulma Öncesi	0.87	0.78	0.82
		Iskalama	0.81	0.82	0.82
		İmha	0.91	0.99	0.95
SMOTE-TOMEK	0.787	Vurulma Öncesi	0.72	0.72	0.72
		Iskalama	0.72	0.66	0.69
		İmha	0.90	0.98	0.94

Çizelge 6.1.10: Noksan Öznitelik Seti Karar Ağacı Algoritması ile Sınıflandırma Performans Analizi

Karar Ağacı Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.950	Vurulma Öncesi	0.97	0.97	0.97
		Iskalama	0.89	0.91	0.90
		İmha	0.91	0.77	0.83
SMOTE	0.947	Vurulma Öncesi	0.99	0.94	0.96
		Iskalama	0.95	0.92	0.93
		İmha	0.91	0.98	0.95
ROS	0.949	Vurulma Öncesi	0.99	0.94	0.96
		Iskalama	0.92	0.94	0.93
		İmha	0.94	0.97	0.95
RUS	0.941	Vurulma Öncesi	0.96	0.93	0.94
		Iskalama	0.93	0.91	0.92
		İmha	0.93	0.99	0.96
YA	0.899	Vurulma Öncesi	0.94	0.92	0.93
		Iskalama	0.88	0.84	0.86
		İmha	0.89	0.95	0.92
SMOTE-ENN	0.950	Vurulma Öncesi	0.99	0.93	0.96
		Iskalama	0.95	0.93	0.94
		İmha	0.92	0.98	0.95
SMOTE-TOMEK	0.948	Vurulma Öncesi	0.98	0.93	0.96
		Iskalama	0.94	0.93	0.94
		İmha	0.92	0.98	0.95

Çizelge 6.1.11: Noksan Öznitelik Seti Gradyan Artırım Algoritması ile Sınıflandırma Performans Analizi

Grandyan Artırım Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.894	Vurulma Öncesi	0.97	0.93	0.957
		Iskalama	0.88	0.79	0.84
		İmha	0.40	0.82	0.54
SMOTE	0.962	Vurulma Öncesi	0.98	0.96	0.97
		Iskalama	0.96	0.95	0.95
		İmha	0.95	0.98	0.96
ROS	0.985	Vurulma Öncesi	1.00	0.96	0.98
		Iskalama	1.00	0.96	0.98
		İmha	0.99	1.00	0.99
RUS	0.941	Vurulma Öncesi	0.97	0.91	0.94
		Iskalama	0.92	0.93	0.93
		İmha	0.93	0.98	0.96
YA	0.937	Vurulma Öncesi	0.93	0.94	0.94
		Iskalama	0.90	0.93	0.91
		İmha	0.98	0.95	0.96
SMOTE-ENN	0.968	Vurulma Öncesi	0.99	0.96	0.98
		Iskalama	0.97	0.96	0.96
		İmha	0.95	0.98	0.97
SMOTE-TOMEK	0.963	Vurulma Öncesi	0.98	0.95	0.97
		Iskalama	0.95	0.96	0.96
		İmha	0.96	0.98	0.97

Çizelge 6.1.12: Noksan Öznitelik Seti Rastgele Orman Algoritması ile Sınıflandırma Performans Analizi

Rastgele Orman Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.977	Vurulma Öncesi	0.98	0.99	0.99
		Iskalama	0.96	0.94	0.95
		İmha	0.94	0.97	0.95
SMOTE	0.978	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.98	0.97	0.98
		İmha	0.97	1.00	0.98
ROS	0.993	Vurulma Öncesi	1.00	0.98	0.99
		Iskalama	0.98	1.00	0.99
		İmha	1.00	1.00	1.00
RUS	0.957	Vurulma Öncesi	0.98	0.93	0.96
		Iskalama	0.94	0.95	0.95
		İmha	0.95	0.99	0.97
YA	0.952	Vurulma Öncesi	0.96	0.94	0.95
		Iskalama	0.94	0.94	0.94
		İmha	0.96	0.98	0.97
SMOTE-ENN	0.983	Vurulma Öncesi	1.00	0.97	0.99
		Iskalama	0.99	0.98	0.98
		İmha	0.96	1.00	0.98
SMOTE-TOMEK	0.982	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.98	0.98	0.98
		İmha	0.97	1.00	0.98

Çizelge 6.1.13: Noksan Öznitelik Seti Çok Katmanlı Algılayıcı Algoritması ile Sınıflandırma Performans Analizi

Çok Katmanlı Algılayıcı Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.958	Vurulma Öncesi	0.98	0.98	0.98
		Iskalama	0.96	0.89	0.92
		İmha	0.72	0.89	0.80
SMOTE	0.953	Vurulma Öncesi	1.00	0.95	0.97
		Iskalama	0.97	0.92	0.94
		İmha	0.90	0.99	0.95
ROS	0.953	Vurulma Öncesi	1.00	0.95	0.97
		Iskalama	0.97	0.91	0.94
		İmha	0.90	1.00	0.95
RUS	0.945	Vurulma Öncesi	0.95	0.96	0.95
		Iskalama	0.98	0.88	0.93
		İmha	0.91	0.99	0.95
YA	0.894	Vurulma Öncesi	0.93	0.93	0.93
		Iskalama	0.92	0.80	0.85
		İmha	0.85	0.96	0.90
SMOTE-ENN	0.957	Vurulma Öncesi	0.99	0.95	0.97
		Iskalama	0.98	0.93	0.95
		İmha	0.91	1.00	0.95
SMOTE-TOMEK	0.955	Vurulma Öncesi	0.99	0.94	0.97
		Iskalama	0.96	0.93	0.95
		İmha	0.92	0.99	0.95

Çizelge 6.1.14: Noksan Öznitelik Seti Destek Vektör Makineleri Algoritması ile Sınıflandırma Performans Analizi

Destek Vektör Makineleri Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.948	Vurulma Öncesi	0.95	0.98	0.97
		Iskalama	0.94	0.87	0.90
		İmha	0.89	0.76	0.82
SMOTE	0.961	Vurulma Öncesi	0.96	0.96	0.96
		Iskalama	0.97	0.95	0.96
		İmha	0.95	0.98	0.96
ROS	0.982	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.97	0.98	0.98
		İmha	0.98	1.00	0.99
RUS	0.911	Vurulma Öncesi	0.97	0.89	0.93
		Iskalama	0.94	0.87	0.90
		İmha	0.84	0.97	0.90
YA	0.927	Vurulma Öncesi	0.93	0.95	0.94
		Iskalama	0.91	0.91	0.91
		İmha	0.95	0.93	0.94
SMOTE-ENN	0.964	Vurulma Öncesi	0.97	0.96	0.96
		Iskalama	0.97	0.96	0.96
		İmha	0.95	0.98	0.96
SMOTE-TOMEK	0.960	Vurulma Öncesi	0.95	0.96	0.96
		Iskalama	0.97	0.95	0.96
		İmha	0.96	0.97	0.96

Çizelge 6.1.15: Noksan Öznitelik Seti Sinir Ağları Algoritması ile Sınıflandırma Performans Analizi

Sinir Ağları Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.959	Vurulma Öncesi	0.98	0.97	0.98
		Iskalama	0.92	0.94	0.93
		İmha	0.81	0.84	0.82
SMOTE	0.951	Vurulma Öncesi	0.99	0.95	0.97
		Iskalama	0.98	0.90	0.94
		İmha	0.91	1.00	0.95
ROS	0.952	Vurulma Öncesi	1.00	0.94	0.97
		Iskalama	0.97	0.91	0.94
		İmha	0.91	1.00	0.95
RUS	0.938	Vurulma Öncesi	0.96	0.94	0.95
		Iskalama	0.95	0.89	0.92
		İmha	0.91	0.98	0.94
YA	0.899	Vurulma Öncesi	0.94	0.92	0.93
		Iskalama	0.93	0.81	0.87
		İmha	0.84	0.98	0.90
SMOTE-ENN	0.959	Vurulma Öncesi	0.99	0.95	0.97
		Iskalama	0.98	0.93	0.95
		İmha	0.91	1.00	0.95
SMOTE-TOMEK	0.956	Vurulma Öncesi	0.99	0.95	0.97
		Iskalama	0.97	0.92	0.95
		İmha	0.91	1.00	0.95

Çizelge 6.1.16: Noksan Öznitelik Seti LSTM Algoritması ile Sınıflandırma Performans Analizi

LSTM Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.979	Vurulma Öncesi	0.98	0.99	0.99
		Iskalama	0.98	0.94	0.96
		İmha	0.94	0.90	0.92
SMOTE	0.976	Vurulma Öncesi	0.98	0.97	0.97
		Iskalama	0.99	0.96	0.97
		İmha	0.96	1.00	0.98
ROS	0.974	Vurulma Öncesi	0.98	0.97	0.98
		Iskalama	0.99	0.95	0.97
		İmha	0.95	1.00	0.98
RUS	0.957	Vurulma Öncesi	0.95	0.97	0.96
		Iskalama	1.00	0.89	0.94
		İmha	0.93	1.00	0.96
YA	0.967	Vurulma Öncesi	0.92	0.97	0.95
		Iskalama	0.97	0.85	0.91
		İmha	0.91	0.98	0.94
SMOTE-ENN	0.980	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.99	0.97	0.98
		İmha	0.97	1.00	0.98
SMOTE-TOMEK	0.974	Vurulma Öncesi	0.99	0.95	0.97
		Iskalama	0.98	0.97	0.97
		İmha	0.96	1.00	0.98

6.1.3. Belirsizlik Analizi Öznitelik Seti Çizelgeleri

Çizelge 6.1.17: Belirsizlik Analizi Öznitelik Seti Ada Artırım Algoritması Performans Analizi

Ada Artırım Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.783	Vurulma Öncesi	0.83	0.91	0.87
		Iskalama	0.48	0.3	0.37
		İmha	0.63	0.57	0.6
SMOTE	0.759	Vurulma Öncesi	0.74	0.7	0.72
		Iskalama	0.71	0.68	0.69
		İmha	0.81	0.9	0.86
ROS	0.748	Vurulma Öncesi	0.71	0.69	0.7
		Iskalama	0.7	0.66	0.68
		İmha	0.83	0.9	0.86
RUS	0.716	Vurulma Öncesi	0.69	0.67	0.68
		Iskalama	0.68	0.62	0.65
		İmha	0.78	0.89	0.83
YA	0.717	Vurulma Öncesi	0.61	0.69	0.65
		Iskalama	0.67	0.58	0.62
		İmha	0.88	0.88	0.88
SMOTE-ENN	0.796	Vurulma Öncesi	0.78	0.76	0.77
		Iskalama	0.77	0.72	0.75
		İmha	0.83	0.91	0.87
SMOTE-TOMEK	0.766	Vurulma Öncesi	0.73	0.74	0.74
		Iskalama	0.73	0.65	0.69
		İmha	0.82	0.91	0.86

Çizelge 6.1.18: Belirsizlik Analizi Öznitelik Seti Karar Ağacı Algoritması ile Sınıflandırma Performans Analizi

Karar Ağacı Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.952	Vurulma Öncesi	0.96	0.99	0.97
		Iskalama	0.95	0.86	0.9
		İmha	0.79	0.78	0.78
SMOTE	0.919	Vurulma Öncesi	0.95	0.9	0.92
		Iskalama	0.91	0.9	0.9
		İmha	0.9	0.96	0.93
ROS	0.942	Vurulma Öncesi	0.97	0.89	0.93
		Iskalama	0.91	0.94	0.93
		İmha	0.95	0.99	0.97
RUS	0.875	Vurulma Öncesi	0.92	0.83	0.88
		Iskalama	0.87	0.85	0.86
		İmha	0.84	0.96	0.89
YA	0.891	Vurulma Öncesi	0.81	0.93	0.87
		Iskalama	0.93	0.8	0.86
		İmha	0.94	0.95	0.94
SMOTE-ENN	0.925	Vurulma Öncesi	0.98	0.89	0.93
		Iskalama	0.92	0.91	0.91
		İmha	0.89	0.97	0.93
SMOTE-TOMEK	0.918	Vurulma Öncesi	0.94	0.89	0.92
		Iskalama	0.91	0.9	0.9
		İmha	0.91	0.96	0.94

Çizelge 6.1.19: Belirsizlik Analizi Öznitelik Seti Gradyan Artırım Algoritması ile Sınıflandırma Performans Analizi

Grandyan Artırım Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.949	Vurulma Öncesi	0.95	0.99	0.97
		Iskalama	0.93	0.82	0.87
		İmha	0.90	0.82	0.86
SMOTE	0.964	Vurulma Öncesi	0.98	0.96	0.97
		Iskalama	0.96	0.95	0.95
		İmha	0.95	0.98	0.97
ROS	0.984	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.98	0.99	0.98
		İmha	0.98	1.00	0.99
RUS	0.899	Vurulma Öncesi	0.95	0.85	0.89
		Iskalama	0.88	0.90	0.89
		İmha	0.88	0.96	0.92
YA	0.976	Vurulma Öncesi	0.97	0.98	0.98
		Iskalama	0.97	0.97	0.97
		İmha	0.98	0.98	0.98
SMOTE-ENN	0.964	Vurulma Öncesi	0.98	0.96	0.97
		Iskalama	0.96	0.95	0.95
		İmha	0.95	0.98	0.97
SMOTE-TOMEK	0.964	Vurulma Öncesi	0.98	0.96	0.97
		Iskalama	0.96	0.95	0.95
		İmha	0.95	0.98	0.97

Çizelge 6.1.20: Belirsizlik Analizi Öznitelik Seti Rastgele Orman Algoritması ile Sınıflandırma Performans Analizi

Rastgele Orman Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.990	Vurulma Öncesi	0.98	1.00	0.99
		Iskalama	0.98	0.98	0.98
		İmha	0.98	0.93	0.96
SMOTE	0.987	Vurulma Öncesi	0.99	0.99	0.99
		Iskalama	0.99	0.97	0.98
		İmha	0.97	0.99	0.98
ROS	0.996	Vurulma Öncesi	1.00	0.99	1.00
		Iskalama	1.00	0.99	1.00
		İmha	0.99	1.00	1.00
RUS	0.923	Vurulma Öncesi	0.95	0.88	0.91
		Iskalama	0.92	0.91	0.92
		İmha	0.90	1.00	0.94
YA	0.977	Vurulma Öncesi	0.99	0.97	0.98
		Iskalama	0.97	0.98	0.97
		İmha	0.98	0.98	0.98
SMOTE-ENN	0.988	Vurulma Öncesi	1.00	0.99	0.99
		Iskalama	0.99	0.98	0.99
		İmha	0.98	1.00	0.99
SMOTE-TOMEK	0.987	Vurulma Öncesi	0.99	0.98	0.99
		Iskalama	0.99	0.98	0.99
		İmha	0.98	1.00	0.99

Çizelge 6.1.21: Belirsizlik Analizi Öznitelik Seti Çok Katmanlı Algılayıcı Algoritması ile Sınıflandırma Performans Analizi

Çok Katmanlı Algılayıcı Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.934	Vurulma Öncesi	0.95	0.98	0.97
		Iskalama	0.89	0.85	0.87
		İmha	0.74	0.53	0.62
SMOTE	0.901	Vurulma Öncesi	0.97	0.87	0.92
		Iskalama	0.90	0.91	0.90
		İmha	0.86	0.95	0.90
ROS	0.909	Vurulma Öncesi	0.98	0.87	0.92
		Iskalama	0.91	0.89	0.90
		İmha	0.85	0.96	0.91
RUS	0.860	Vurulma Öncesi	0.96	0.83	0.89
		Iskalama	0.90	0.85	0.87
		İmha	0.74	0.91	0.82
YA	0.911	Vurulma Öncesi	0.93	0.93	0.93
		Iskalama	0.94	0.90	0.92
		İmha	0.87	0.91	0.89
SMOTE-ENN	0.917	Vurulma Öncesi	0.98	0.89	0.93
		Iskalama	0.93	0.90	0.91
		İmha	0.86	0.96	0.91
SMOTE-TOMEK	0.906	Vurulma Öncesi	0.98	0.86	0.92
		Iskalama	0.90	0.90	0.90
		İmha	0.86	0.96	0.90

Çizelge 6.1.22: Belirsizlik Analizi Öznitelik Seti Destek Vektör Makineleri Algoritması ile Sınıflandırma Performans Analizi

Destek Vektör Makineleri Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.962	Vurulma Öncesi	0.96	0.99	0.98
		Iskalama	0.98	0.87	0.92
		İmha	0.93	0.80	0.86
SMOTE	0.956	Vurulma Öncesi	0.99	0.93	0.96
		Iskalama	0.96	0.96	0.96
		İmha	0.92	0.99	0.95
ROS	0.975	Vurulma Öncesi	0.99	0.96	0.98
		Iskalama	0.98	0.97	0.98
		İmha	0.95	1.00	0.97
RUS	0.872	Vurulma Öncesi	0.94	0.79	0.86
		Iskalama	0.85	0.90	0.87
		İmha	0.84	0.94	0.89
YA	0.933	Vurulma Öncesi	0.95	0.95	0.95
		Iskalama	0.92	0.95	0.93
		İmha	0.93	0.90	0.91
SMOTE-ENN	0.959	Vurulma Öncesi	0.99	0.94	0.96
		Iskalama	0.97	0.96	0.96
		İmha	0.93	0.98	0.95
SMOTE-TOMEK	0.954	Vurulma Öncesi	0.99	0.92	0.96
		Iskalama	0.95	0.96	0.96
		İmha	0.92	0.98	0.95

Çizelge 6.1.23: Belirsizlik Analizi Öznitelik Seti Sinir Ağları Algoritması ile Sınıflandırma Performans Analizi

Sinir Ağları Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.942	Vurulma Öncesi	0.95	0.99	0.97
		Iskalama	0.95	0.85	0.90
		İmha	0.67	0.57	0.62
SMOTE	0.931	Vurulma Öncesi	0.99	0.90	0.94
		Iskalama	0.96	0.91	0.93
		İmha	0.86	0.98	0.92
ROS	0.939	Vurulma Öncesi	0.99	0.92	0.95
		Iskalama	0.97	0.91	0.94
		İmha	0.87	0.99	0.93
RUS	0.835	Vurulma Öncesi	0.94	0.81	0.87
		Iskalama	0.83	0.82	0.82
		İmha	0.75	0.89	0.81
YA	0.884	Vurulma Öncesi	0.87	0.94	0.90
		Iskalama	0.93	0.86	0.89
		İmha	0.86	0.85	0.86
SMOTE-ENN	0.935	Vurulma Öncesi	0.99	0.90	0.95
		Iskalama	0.97	0.91	0.94
		İmha	0.86	0.99	0.92
SMOTE-TOMEK	0.925	Vurulma Öncesi	0.99	0.89	0.94
		Iskalama	0.94	0.91	0.93
		İmha	0.86	0.97	0.91

Çizelge 6.1.24: Belirsizlik Analizi Öznitelik Seti LSTM Algoritması ile Sınıflandırma Performans Analizi

LSTM Algoritması					
Sınıf Denge Türü	Performans Ölçütü				
	Doğruluk	Veri Tipi	Kesinlik	Duyarlılık	F1-Ölçütü
Dengesiz	0.980	Vurulma Öncesi	0.98	1.00	0.99
		Iskalama	0.98	0.95	0.96
		İmha	0.92	0.85	0.88
SMOTE	0.973	Vurulma Öncesi	0.95	0.99	0.97
		Iskalama	0.99	0.98	0.98
		İmha	0.99	0.96	0.97
ROS	0.977	Vurulma Öncesi	0.97	0.97	0.97
		Iskalama	0.99	0.98	0.98
		İmha	0.97	0.98	0.98
RUS	0.904	Vurulma Öncesi	0.98	0.86	0.91
		Iskalama	0.95	0.89	0.92
		İmha	0.8	0.98	0.88
YA	0.955	Vurulma Öncesi	0.96	0.97	0.97
		Iskalama	0.96	0.94	0.95
		İmha	0.94	0.95	0.95
SMOTE-ENN	0.983	Vurulma Öncesi	0.98	0.98	0.98
		Iskalama	1.00	0.98	0.99
		İmha	0.98	0.99	0.98
SMOTE-TOMEK	0.981	Vurulma Öncesi	0.98	0.98	0.98
		Iskalama	0.99	0.98	0.98
		İmha	0.98	0.99	0.98