T.C.
MİLLİ SAVUNMA ÜNİVERSİTESİ
ATATÜRK STRATEJİK ARAŞTIRMALAR VE LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK PROGRAMI

ADS-B CİHAZLARINA YAPILAN SALDIRILARIN YAPAY ÖĞRENME İLE TESPİTİ

YÜKSEK LİSANS TEZİ

İBRAHİM MERAL
19200501

TEZ DANIŞMANI: Dr. Öğr. Üyesi Elif BOZKAYA ARAS

İSTANBUL
EYLÜL 2023

T.C.
MİLLİ SAVUNMA ÜNİVERSİTESİ
ATATÜRK STRATEJİK ARAŞTIRMALAR VE LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
SİBER GÜVENLİK PROGRAMI

ADS-B CİHAZLARINA YAPILAN SALDIRILARIN YAPAY ÖĞRENME İLE TESPİTİ

YÜKSEK LİSANS TEZİ

İbrahim MERAL
19200501

Tezin Enstitüye Verildiği Tarih : 28.09.2023
Tezin Savunulduğu Tarih : 05.09.2023

Tez Oy birliği / ~~Oy çokluğu~~ ile başarılı bulunmuştur.

	Unvan Ad Soyad	İmza
Tez Danışmanı	Dr.Öğr.Üyesi Elif BOZKAYA ARAS	
Jüri Üyeleri	Dr.Öğr.Üyesi Musa MİLLİ	
	Dr.Öğr.Üyesi İrfan KÖSESOY	

İSTANBUL
EYLÜL 2023

ÖZGÜNLÜK RAPORU

Tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve ç) Sonuç kısımlarından oluşan toplam 59 sayfalık kısmına ilişkin, 18/09/2023 tarihinde Tez Danışmanı tarafından iThenticate adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan özgünlük raporuna göre, tezimin benzerlik oranı %11'dir.

Uygulanan filtrelemeler:

Kaynakça hariç

Alıntılar ~~hariç~~/dâhil

5 kelimeden daha az örtüşme içeren metin kısımları hariç

Milli Savunma Üniversitesi Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü Lisansüstü Tez Çalışması Özgünlük Raporu Alınması ve Kullanılması Uygulama Usul ve Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

İbrahim MERAL

28.09.2023

ETİK BEYAN

Milli Savunma Üniversitesi Enstitüleri Dönem Projesi ve Lisansüstü Tez Hazırlama Kılavuzu'nda yer alan kurallarına uygun olarak hazırladığım bu tez çalışmasında; tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi, tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu, tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi, kullanılan verilerde herhangi bir değişiklik yapmadığımı, bu tezde sunduğum çalışmanın özgün olduğunu, bildirir; aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Bu tezdeki düşünce, görüş, varsayım, sav veya tezler bana aittir; Milli Savunma Bakanlığı, Milli Savunma Üniversitesi ve Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü sorumlu tutulamaz.

İbrahim MERAL

28.09.2023

Sevgili aileme,

ÖNSÖZ ve TEŞEKKÜR

Öncelikle eğitim hayatım boyunca destekleriyle beni hiçbir zaman yalnız bırakmayan sevgili eşim ve kızıma, yüksek lisans eğitimim ve tez çalışmam süresince çok kıymetli destekleriyle bana yol gösterici olan tez danışmanım Dr. Öğr. Üyesi Elif BOZKAYA ARAS'a, yüksek lisans eğitimim süresince sunduğu imkanlar için MSÜ Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü sonsuz saygı ve teşekkürlerimi sunarım.

İstanbul; Eylül 2023 İbrahim MERAL

İÇİNDEKİLER

TABLOLAR LİSTESİ

Sayfa

ŞEKİLLER LİSTESİ

Sayfa

KISALTMALAR

ADS-B : Automatic Dependent Surveillance

ATCSIMTEST : ATC Simülasyon Test Yazılımı

GNSS : Global Navigation Satellite System

GPS : Global Positioning System

MHZ : MegaHertz

ACARS : Aircraft Communications Adressing Reporting System

FANS : Future Air Navigation System

HackRf : Hack Radio Frequency

LSTM : Long Short-Term Memory

MLAT : Multilateration

SVDD : Support Vector Data Description

SSR : Secondary Survelliance Radar

SVM : Support Vector Machine

RBF : Radial Basis Function

ICAO : International Civil Aviation Organisation

GNB : Gaussian Naive Bayes

MNB : Multinomial Naive Bayes

CNB : Categorical Naive Bayes

TÜRKÇE ÖZ

ADS-B Cihazlarına Yapılan Saldırıların Yapay Öğrenme ile Sınıflandırılması

İbrahim MERAL

Milli Savunma Üniversitesi, Atatürk Stratejik Araştırmalar ve Lisansüstü Eğitim Enstitüsü

İstanbul, Eylül 2023

Havayolu taşımacılığı başlangıcından itibaren hava araçlarının takibi, uçuşun emniyeti ve hava trafiğinin yönetimi için oldukça önemlidir. Hava taşıtlarının takibinde ise hava taşıtının konumumun bulunduğu alanı yöneten hava sahası işletmecileri, kurumları bulunmaktadır. Bu hava sahasını kontrol eden kurumlar hava taşıtlarını takip edebilmek için çeşitli sistemler kullanmaktadır. Bu sistemler bütününe ise Hava Trafik Yönetim Sistemleri denmektedir. Uçakları algılaması için kullanılan bir çok radar çeşidi bulunmaktadır. Bu radarların dışında hava taşıtlarının konumunu tespit etmek için ADS-B (Automatic Dependent Surveillance-Broadcast) cihazları kullanılmaktadır. ADS-B cihazları kurulumu ve maliyeti diğer radar sistemlerine göre daha ucuz olduğundan saldırganlar için daha çok tercih edilir.

Bu tez kapsamında, ADS-B verisine yapılan saldırıların sınıflandırılması için hava taşıtı simülasyon araçları kullanılmıştır. Elde edilen simülasyon yayını dinlenip Java programlama dilinde geliştirilmiş olan test yazılımı kullanılarak ADS-B verisine dönüştürülmüştür. Sonrasında bu bozulmamış veri saldırı yapılmış bir veri setine dönüştürmüştür. Bu veri seti üzerinden ise Support Vector Machines, Decision Tree ve Naive Bayes makine öğrenme teknikleri Python programlama dili kullanılarak uygulanmıştır. Yapay öğrenme teknikleri çözüm modellemesi yapılırken, test oranlarına ve öğrenme tekniklerinin kendi içerisindeki tiplerine göre farklı değerler verilerek modellenmiştir. Elde edilen sonuçlar kıyaslanarak en doğru tahminlemeyi İkili Karar Ağacı yapay öğrenme tekniğinde elde edildiği sonucuna varılmıştır.

Anahtar Sözcükler: ADS-B , Siber Güvenlik, Yapay Öğrenme, Siber Saldırı, Hava Trafik Kontrol Yönetimi, Uçuş Takip Sistemleri

Bilim Kodu : 92403

Sayfa Sayısı : XII + 47

Tez Danışmanı : Dr. Öğr. Üyesi Elif BOZKAYA ARAS

ABSTRACT

Classification of Attacks on ADS-B Devices with Artificial Learning

İbrahim MERAL

National Defense University, Atatürk Institute of Strategic Studies and Graduate Education

Istanbul, September 2023

From the beginning of air transport, the follow-up of the relevant airline vehicles in this type of transport is very important for the safety of the flight and the management of air traffic. In the follow-up of aircraft, there are airspace operators and institutions that manage the area where the aircraft is located. The institutions that control this airspace use various systems to track their aircraft. All of these systems are called Air Traffic Management Systems. There are many types of radars used to detect aircraft. Apart from radars, ADS-B devices are used to detect the position of aircraft. ADS-B devices are more preferred by attackers as they are cheaper to install and cost than a regular radar.

Within the scope of this thesis, he listened to the broadcast from Simulation Tool, which is the simulation tool for the classification of attacks on ADS-B data, and then converted the tool into a data set in which ADS-B data was attacked by using the ATCSIMTEST software developed in Java programming language. On this data set, Support Vector machines, Decision Tree and Naive Bayes machine learning techniques were applied using Python programming language. The variables used by these techniques were also given different values and the most accurate results were tried to be obtained. By making comparisons between these techniques in the study, it will also explain the most appropriate technique that can be used in the classification of attacks on the ADS-B device.

Keywords: ADS-B, Cyber Security, Artifical Learning, Cyber Attacks, Air Traffic Control Management, Flight Follow-Up Systems.

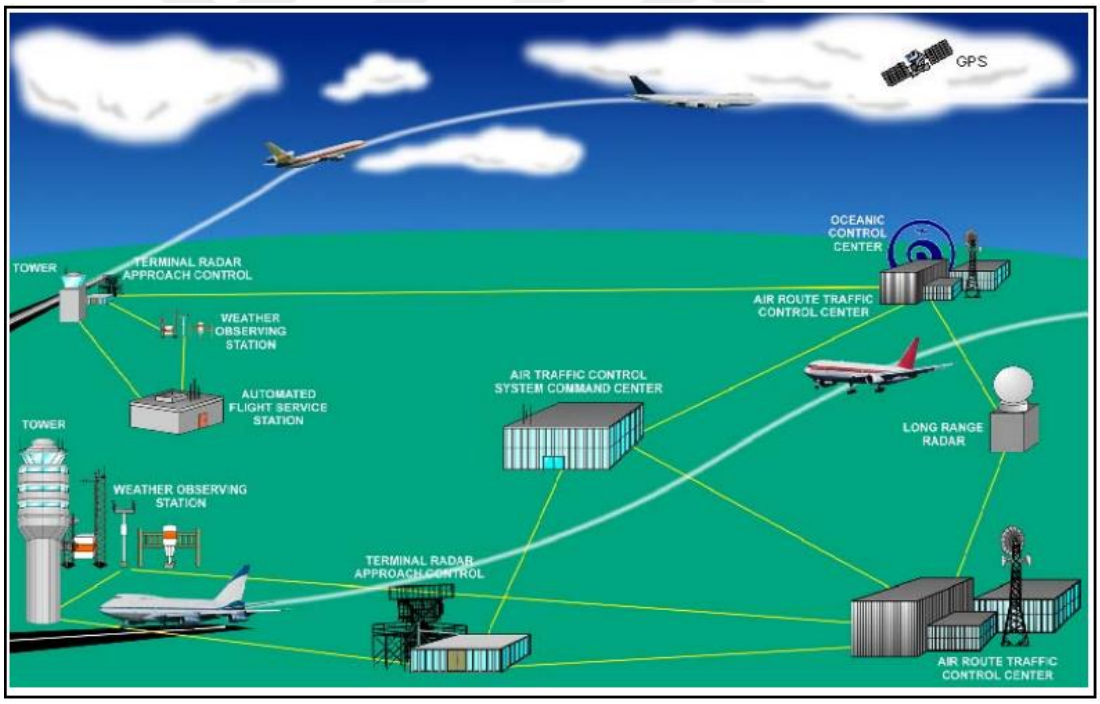
Science Code : 92403

Pages : XII + 47

Supervisor : Assist. Prof. Elif BOZKAYA ARAS, Ph.D.

1. GİRİŞ

Sivil ve askeri havacılığın hızla büyümesi ve havadaki uçağın sahip olduğu sistemlerin emniyet kritik sistemler olması nedeniyle havacılıkta kullanılan uygulamaların doğruluğu büyük öneme sahiptir. Hava trafiğinin, hava sahalarının yönetimi için kullanılan sistemler bütününe *Hava Trafik Yönetim Sistemleri* denmektedir. Şekil 1.1'de Hava Trafik Yönetim Sisteminin bir örneği gösterilmiştir (Gerardo, 2015). Şekilde görüldüğü gibi Hava Trafik Yönetim Sistemi uçağın emniyetli seyrini sağlamak maksadıyla; kule, uçağın konumunu belirleyen hava durumu radarları, seyir halinde uçuşun kontrol edildiği hava trafiği kontrol merkezi, havayolu firmaları, terminal ve yaklaşma yönetim birimlerinden oluşmaktadır.



Şekil 1.1: Hava Trafik Kontrol Yönetim Sistemi (Gerardo, 2015).

Hava trafik yönetiminde uçakların konumlarını öğrenmek için birçok sistem kullanılmaktadır. Bunlar yer bazlı radarlar veya uydu bazlı radarlar olmaktadır. Uçakların konum ve pozisyonları ile birlikte havadaki uçağın yönetimi, hareketleri ve komutları belirlenip uçağa gönderilmektedir. Uçağın konum bilgisi sürekli olarak hava

trafiğini yöneten kontrolöre iletilmekte ve kontrolörün kullandığı sistemlerde gösterilmektedir. Uçak konumlarını sağlayan radarlar ve otomatik bağımlı gözetim yayını cihazları (ADS-B; Automatic Dependent Surveillance-Broadcast) bulunmaktadır.

Radarlar, antenlerinden bir elektromanyetik dalga göndererek bunun karşı nesnelerden yansımalarını gözlemleyip çözümlerler. ADS-B cihazları, GNSS (Global Navigation Satellite System), GPS (Global Positioning System) gibi uydulardan aldıkları kendi pozisyonlarının bilgisini ADS-B verisi ile yayınlarlar. Bu sayede konum verisinin doğruluğu; yatay ve dikey uzaklık veya hava durumlarından etkilenmez. ADS-B ayrıca maliyet açısından da diğer konum bilgisi veren radarlara göre çok daha uygun maliyettedir.

Hava trafik yönetim sistemleri, uçağın kalkıştan varışa uçtan uca tüm aşamalarının yönetildiği sistemlerdir. Belirli bir havasahasında bulunan tüm uçuşların emniyetinden hava trafik yönetimi sorumludur. Bu nedenle hava trafik yönetim sistemleri; tüm sorumlu olunan uçuşların yönetildiği sistemlerdir. Uçuş emniyetini sağlayan bu hizmet ülkelerin hava seyrüsefer hizmet sağlayıcıları tarafından verilir.

Hava trafik yönetiminde uçağın konum bilgisi büyük öneme sahiptir. Hava trafik yönetim sisteminde bu bilgi, radar ve ADS-B cihazları ile elde edilmektedir. ADS-B cihazı ile ilgili detaya bir sonraki alt başlıkta yer verilmiştir.

1.1 ADS-B Cihazı Hakkında Genel Bilgi

Tez çalışmasının bu kısmında saldırı yapılan ADS-B cihazı ve bu cihaza yapılabilecek siber saldırılar detaylandırılacaktır.

ADS-B cihazının açılımı ve özellikleri:

- A: Automatic; sürekli açıktır ve operator işlemi gerektirmez.
- D: Dependent; pozisyon verisi GNSS gibi uydularla ilişkili, bağımlıdır.
- S: Surveillance; radarlarda olduğu gibi tarama özelliğine sahiptir.
- B: Broadcast; uçağın konum ve pozisyonunu sürekli yayınlar.

Hava trafik yönetimlerinde aynı hava sahası için pozisyon bilgisi veren bir sistem (radar veya ADS-B) kullanılabildiği gibi hava sahasını aynı anda birden fazla radar veya ADS-B kapsayabilir. Bu durumda her bir pozisyon sağlayıcıdan ilgili uçuşa ait

bir konum bilgisi gelecektir, bu da aynı ekranda bir uçuş için birden fazla uçuş gösterimi yapıp kontrolörün hava trafiğini yönetmesini imkansız hale getirecektir. Kontrolörün yönetim ekranında her uçuş için bir pozisyon bilgisi görüntülenmelidir, birden fazla pozisyon bilgisi geliyorsa bunları tek bir uçuş gibi kontrolörün ekranına yansıtılmalıdır. Radarlar ve ADS-B'den gelen verileri birleştirip tek bir konum bilgisi çıkarma işlemine füzyon denilmektedir. Bu füzyon işlemini yapan yazılımlara genelde Gözetim Veri İşlemcisi (SDP; Surveillance Data Processor) denmektedir.

Radarların ve ADS-B'nin füzyon işlemine katılması veya tek başlarına yayın yapabilmeleri için yayınladıkları verilerin doğruluğu çok önemlidir. ADS-B sistemleri birçok avantajına rağmen temel güvenlik mekanizmaları (şifreleme veya yetkilendirme) açısından zayıftır. Bunun yanı sıra ADS-B cihazı 1090 Mhz bant aralığında açık mesaj kullandığı için dışarıdan açık kaynak yazılımların saldırısına maruz kalabilmektedir. Bu saldırılar Eavesdropping (Dinleme), Jamming Attack (Yayını bozma), message injection and manipulation (Mesajı değiştirme, bozma, silme) vb. şekilde olmaktadır.

1.2 ADS-B Mesaj Yapısı

ADS-B mesajının her biri; 8 bit başlangıç (preamble) bloğu, 5 bit bağlantı (downlink) formatı, 3 bit kapasite (capacity), 24 bit uçağın adresi (aircraft address), 56 bit ADS-B genişletilmiş mesajı, 24 bit döngüsel artıklık denetimi (CRC; Cylic Redundancy Check) olmak üzere 120 bitlik bir veri bloğundan oluşmaktadır. Tablo 2.1'de ADS-B mesajı genel yapısı gösterilmiştir (ICAO,2018).

Tablo 1.1: ADS-B Mesaj Yapısı

Başlangıç	Bağlantı Formatı	Kapasite	Uçağın Adresi	Genişletilmii Mesaj	Döngüsel Artıklık Denetimi
8 bit	5 bit	3 bit	24 bit	56 bit	24 bit

Bağlantı (Downlink) Formatı: Eğer ADS-B cihazı Mode-S transponder kullanıyor ise 17 (10001) değerini göndermektedir. Transponder bulundurmayan cihazlardan gelen veri bloğunda ise bu kısım 18 (10010) olarak gönderilmektedir (ICAO,2018).

Kapasite (Capacity): Bu kısım transponder seviyesinin bilgisini vermektedir. 3 bit olduğu için 0-7 arasında değerleri alabilmektedir. Aşağıda Tablo 2.2'de değerlerin tanımları verilmiştir (ICAO,2018).

Tablo 1.2: Kapasite Tanımları

Kapasite	**Tanım**
0	Seviye 1 transponder
1-3	Rezerve edilmiş
4	Seviye 2+ transponder yer (istasyon) için
5	Seviye 2+ transponder uçak için
6	Seviye 2+ transponder hem yer hem uçak için
7	Downlink Request'e 0 atar veya Flight Status hem uçak hem yer için 2, 3, 4 veya 5'tir

Uçağın Adresi (Aircraft Address): Uçağın tekil olan Mode-S kodunu vermektedir. Aircraft adres veya Hexedecimal kodu olarak da adlandırılmaktadır (ICAO,2018).

Genişletimiş Mesaj (Extended message): Mesaj yapısının 56 bitlik kısmıdır. Bu alanın ilk 5 biti, Type Code olarak adlandırılmaktadır. Genişletilmiş mesaj uçakla ilgili uçağın pozisyonu, uçağın hızı, transponder seviyesi gibi bilgiler vermektedir (ICAO,2018).

Döngüsel Artıklık Denetimi (Cylic Redundancy Check): Gönderilen mesajın alıcı tarafından doğrulamasını sağlayan bitlerdir (ICAO,2018).

1.3 Saldırı Tipleri

ADS-B sistemleri şifrelenmemiş mesaj içeriği bulunması, başka cihaz veya araçlar ile etkileşim halinde olması nedeniyle farklı saldırı türlerine maruz kalmaktadır. Saldırılar genel anlamda veriyi dinleme veya veriyi bozma üzerine yapılan saldırılardır. Saldırılar mesajın yapısındaki veri bloklarına göre de şekillenir. Saldırı tiplerinden bazıları aşağıda açıklanmıştır:

- *Eavesdropping (Dinleme):* Sistemin yayınladığı verileri izinsiz bir şekilde dinlemesidir. Yeterli donanıma; bilgisayar, radyo cihazı ve açık kaynak ADS-B alıcısı bulunan herhangi biri havalimanı çevresinde kendini konumlandırdığında uçakların konum bilgilerini kolaylıkla görüntüleyebilir. Bu da iniş kalkış yapan uçuş ve trafik bilgilerinin ele geçirebileceği anlamına gelmektedir.

- *Jamming Attacks (Yayını Bozma):* Son zamanlarda yayın bozma cihazlarının kullanımı ve bu cihazlara erişim artmıştır. Bu cihazlar yayını kesip kendi istediği verileri göndermektedir. Bu saldırı ile hem hava trafiğini yöneten tarafa, hem de uçak tarafına giden yayın bozulabilmektedir.

- *Ghost Aircraft Injection Attack (Hayalet Uçak Saldırısı):* Bu saldırı tipinde ADS-B sistemine sızan saldırgan havada var olmayan trafiği varmış gibi sisteme ekler ve kontrolörün hava trafiğini yönetmesini zorlaştırır. Özellikle normalde havada var olan trafiğin yakınına ekleyerek kontrolörün var olan trafiğin rotasını değiştirmesi için zorlamaktadır. Burada mesaj yapısının tamamını kullanarak hava trafiğinde kullanılan ADS-B alıcısına yeni bir uçak göndermektedir.

- *Data Replay Attack (Mesajı Yineleme Saldırısı):* Bu saldırı tipinde ise var olan trafiği ADS-B alıcısına yeniden göndererek hava trafiğinde aynı uçaktan bir tane daha varmış gibi veya uçağın konum bilgisini tekrarlı bir şekilde birden fazla gönderilmesine sebep olur. Bu da aynı trafiğin çift gelmesine ve hava trafiğinin yönetilmesini zorlaştırmaktadır. Burada saldırgan mesajın tamamını kullanır. Aynı mesajı tekrar gönderir.

- *Aircaft-Based Attacker (Uçak Tabanlı Saldırı):* Bu saldırı tipinde saldırgan havada var olan trafiğin mesaj içeriğinde değişiklik yaparak mesajın çıkış adresini değiştirir. Bu nedenle kontrolör, gerçekte mesajı gönderen uçaktan değil de farklı bir uçağın adresinden konum bilgisi geliyormuş gibi görüp yanlış kararlar verebilmekte veya görüntüleme ekranında problemler yaşayabilmektedir. Saldırgan mesajın uçak adresini bozarak ADS-B alıcılarına göndermektedir.

- *Falsa Data Attack (Sahte Veri Saldırısı):* Sisteme zarar vermek veya yetkisiz erişim elde etmek amacıyla yanlış veya yanıltıcı verilerin girilmesini içeren bir siber saldırı türüdür. Bu saldırı bazen veri enjeksiyon saldırısı olarak adlandırılır.

Saldırı, sistemin işlevselliğini manipüle etmek veya bozmak amacıyla, genellikle bir uygulama veya veritabanı aracılığıyla bir sisteme tahrif edilmiş veriler sokan bir saldırganı içerir. Veriler kasıtlı olarak yanıltıcı olabilir veya tamamen uydurma olabilir ve fiziksel veya sanal yollarla eklenebilir (Ahmed & Pathan, 2020).

Yukarıda bahsedilen saldırılara karşı bir çok çözüm sunulmuştur. Başka bir radar ile doğrulanarak verinin iletimi, ADS-B cihazından gelen verideki açık mesajın şifrelenmesi gibi çözüm yöntemleri bulunmaktadır. Bu çözümlerden biri de yapay öğrenme teknikleri ile akıllı karar verme mekanizmalarının oluşturulmasıdır. Bu tez çalışması kapsamında, ADS-B cihazlarından gelen veriler üzerinde anomali tespiti maksadıyla yapay öğrenme tekniklerine dayalı bir model önerilmiştir.

1.4 Siber Saldırılara Karşı Yapay Öğrenme

Yapay öğrenme, bir bilgisayarın veya bilgisayar kontrolündeki bir robotun çeşitli faaliyetleri öğrenebilen canlılara benzer şekilde yerine getirme kabiliyeti ve insan öğrenmesine özgü olan, algılama, öğrenme, çoğul kavramları bağlama, düşünme, sorun çözme, iletişim kurma, çıkarım yapma ve karar verme gibi yüksek bilişsel fonksiyonları veya otonom davranışları sergilemesi beklenen yapay bir işletim sistemidir (Pinustech, n.d.). Makine öğrenimi teknikleri olarak da bilinen yapay öğrenme teknikleri, bilgisayar sistemlerinin açıkça programlanmadan öğrenmesine ve tahminler veya kararlar almasına olanak tanıyan bir dizi algoritma ve istatistiksel modeldir. Bu teknikler, bilgisayarların deneyim ve verilerden öğrenmelerini ve zaman içinde performanslarını sürekli iyileştirmelerini sağlar.

Siber saldırılara karşı yapay öğrenme teknikleri kullanarak, saldırıları tespit etmek, önlemek ve müdahale etmek için etkili çözümler geliştirmek mümkündür. Makine öğrenmesi algoritmaları kullanarak, saldırı türlerini tanımlayan özellikleri öğrenmek mümkündür. Bu özellikler, saldırı türüne göre değişebilir, örneğin, kötü amaçlı yazılım genellikle ağ trafiğinde yüksek miktarda veri transferi gerçekleştirirken, bir phishing saldırısı genellikle belirli bir URL veya alan adı kullanır. Makine öğrenmesi algoritmaları, bu özellikleri belirleyerek, yeni saldırılar tespit etmek için kullanılabilir. Diğer yandan yapay öğrenme teknikleri kullanılarak normal bir şekilde çalışan ve sonuçlar üreten bir cihazın veya bilgisayarın normalin dışında anomali olarak öngörülmesi sağlanır. Bu sayede normalden farklı bir şekilde gelen bu saldırı anomalisi hesaplanarak ona göre işletilir.

1.5 Motivasyon

Havacılık tarihinde son 10 yılda meydana gelen ADS-B cihazlarına karşı yapılan saldırı örnekleri aşağıda sunulmuştur.

2013 yılında, Andrei Costin adlı bir araştırmacının Black Hat Europe güvenlik konferansında bir ADS-B saldırısını başarılı bir şekilde gösterdiği bir olay meydana gelmiştir. Costin, hava trafik kontrolörlerinin ekranlarında sanal bir uçağın görünmesine neden olan sahte ADS-B sinyalleri yayınlamak için 2.000 dolarlık yazılım tanımlı bir radyo kullanmıştır. Ayrıca sanal uçağın irtifasını ve hızını manipüle ederek bir çarpışmaya neden olmuştur (Costin & Francillon, 2012).

2015 yılında, Austin'deki Texas Üniversitesi'ndeki araştırmacılar, bir saldırganın ADS-B sinyallerini bozmak ve gökyüzünde kaosa neden olmak için küçük, ucuz bir cihazı nasıl kullanabileceğini gösteren benzer bir gösteri gerçekleştirilmiştir (Taylor et al., 2016).

2015 yılında Westjet adlı havayolu firması 7500 SSR kodunu kontrolöre ulaştırmıştır. 7500 SSR kodu uçaktan hava trafiğini yöneten kontrolöre gönderilen bir mesajdır. Bir uçak kaçırma olayına işaret eden 7500 kodunu iletmesi siber yollarla yapılan bir değişiklik olduğunu düşündürmüştür ("WestJet Says It Never Sent Hijack Alarm, Wasn't in Danger," 2015).

Bu tezde, yukarıdaki saldırılar göz önüne alınarak ADS-B cihazına karşı yapılabilecek olası saldırıları tespit etmek amacıyla yapay öğrenme tekniklerine dayalı anomali tespiti yapan bir sistem modeli önerilmiştir.

1.6 Bu Tezde Sunulan Katkılar

Bu tez kapsamında ADS-B cihazına yapılan olası saldırılardan sahte veri saldırısı (false data attack) ele alınacaktır. Aşağıda bu tezde sunulan katkılar detaylandırılmıştır.

1. ADS-B cihazına karşı olası saldırılar sınıflandırılmıştır.
2. Bu saldırıların tespit edilebilmesi maksadıyla yapay öğrenme tekniklerine dayalı bir sistem modeli önerilmiştir.
3. Sistem modelinin değerlendirilmesi maksadıyla ADS-B verilerinin bulunduğu özgün bir veri seti elde edilmiş ve üzerinde anomali tespiti yapılmıştır.
4. ADS-B cihazına sahte veri saldırısı yapılmış veri seti üzerinde yapay öğrenme tekniklerinden SVM, Naive Bayes ve İkili Karar Ağacı kullanılmıştır.

2. LİTERATÜRDEKİ ÇALIŞMALAR

Tajdini ve arkadaşları çalışmasında; ADS-B cihazının iki frekansı kullanması ile birlikte çalışmayı gerçekleştirmiştir (TajDini et al., 2021). Bu iki frekans 1090Mhz. ve 978Mhz. değerleridir. 1090 Mhz kullanılan cihaz 1090ES olarak, 978 MHz olan ise Universal Access Transceiver (UAT) olarak adlandırılmaktadır. Genellikle 18000 feet'in üstündeki uçaklar için 1090ES kullanılır. Geri kalan uçuş seviyeleri (yükseklik) için ise UAT veya 1090ES kullanılabilir. Downlink formatları kısaca verinin içerisindeki bitlerin değerlerinin ifadesidir. DF17 ADS-B cihazından gelen verinin downlink formatıdır. ADS-B yapısında bulunan Dowlink format günümüzde DF17'den farklı şekilde kullanılabilmektedir. Bunlar DF0, DF4, DF5 DF11, DF16, DF20, DF21 ve DF24 olarak da kullanılabilmektedir. Örneğin DF0 Uçak çakışmadan kaçınma sistemi (Airborne Collision Avoidance System – ACAS) hakkında bilgi sağlar. Diğer taraftan DF24 ise 56 bitlik genişletilmiş mesajı 80 bit'e çıkarır, bu nedenle de genişletilmiş uzun mesaj (extended length message (ELM)) olarak bilinir. DF22 ise yalnızca askeri kullanımlar için geçerlidir. Örnek ADS-B verisinin 56 bitlik genişletilmiş mesajının çözümlemesi yapılmıştır; ve Sniffing pasif atak ve Jamming, inject spoofing aktif atak olarak çalışmalar yapılacağı söylenmiştir. Çalışmada ADS-B verisini elde etmede kullanılan araçlar şunlardır: BladeRF, Dump1090 ve VirtualRadar. Kullanılan yöntemde gelen ADS-B verisini dinlemekte ve çözümlemekte ve sonrasında ACARS üzerinde gönderilecek olan FANS1/A mesajını oluşturmaktadır. Bunu da HackRF yazılımı ile oluşturup antene bağlanıp göndermektedir. Sistem hem dinlemeye hem de verileri manipüle edilmeye çok uygundur. Manipule edilen veriler uçak ve yerdeki kontrol merkezi için yanlış alarmlar, hatalı yönlendirme vb. sonuçları ile kaza kırıma dahi sebep olabilmektedir. Bozulmuş veya değiştirilmiş olan verilerin tespiti için LSTM sinir ağları kullanılmıştır. Toplanan veriler tahminleme ve ölçümler ile gerçek ve sahte verilerin tespitinin elde edilmesi amaçlanmıştır.

El Marady, ADS-B cihazı ile sağlanan verilerin daha doğru olması için MLAT radar verisi ile kaynaştırılması amaçlanmıştır (El Marady, 2017). ADS-B verisi dışarıdaki saldırılara açık olması nedeniyle almış olduğu verileri doğrulamak için MLAT radardan gelen veriler kullanmıştır. ADS-B hem konum hem hız bilgisi içeriyorken MLAT yalnızca pozisyon bilgisi içermektedir. Makalede iki verinin de konumu hesaplama yöntemlerine yer verilmiştir.

Wahlgren ve Thorn çalışmasında; ADS-B cihazına yapılmış olan saldırıları yapay öğrenme tekniklerinden SVM kullanarak saldırıları sınıflandırmayı amaçlamıştır (Wahlgren & Thorn, 2021). Saldırı tiplerinden sahte veri saldırısı seçilerek simülasyon yazılımı kullanılmıştır. Sonrasında elde edilen veri setine SVM yapay öğrenme tekniği kullanılarak sınıflandırma yapılmıştır.

Damis çalışmasında; mikroservis çatısı (mimarisi) altında blockchain teknolojisi kullanarak ADS-B verisinin güvenliği amaçlanmıştır (Damis et al., 2020). ADS-B'den alınan verileri işlerken her bir fonksiyonu ayrı yazılım birimlerine ayırıp ayrı servisler olarak çalıştırılmıştır. Konteynır tabanlı sanallaştırma ve yazılım kullanılmıştır. Cihaz kimlik doğrulama, veri doğrulama, mesaj kimliği doğrulama, anomali hesaplama vb. servislere ayrılmıştır. Her servis birer konteynır üzerinde çalıştırılarak güvenlikleri sağlanmıştır. Güvenlik işlemleri ayrı ayrı denetlendiği için daha doğru ve kapsamlı bir çalışma ortaya konmuş ayrıca da daha denetlenebilir ve ölçülebilir bir sistem modellenmiştir. ADS-B'ye yapılan dinleme (Eavesdropping, Jamming ve Spoofing) saldırılarına karşı çözüm olarak üretilmiştir. Veriyi dinleyecek paydaşların güvenilirliği ve ardından, verilerin güvenirliği çözümlenmeye ve hesaplanmaya çalışılmıştır.

Kocaağa ve arkadaşları çalışmasında; başlangıçta hava trafik yönetiminde kullanılan radarları ele almıştır (Kocaağa, 2017). Daha sonra ADS-B cihazına yapılan saldırılar tanımlanmıştır. Bu güvenlik açıklarını önlemek için kullanılan yöntemler açıklanmıştır. Bu yöntemler: makine öğrenmesi, frekans atlamalı yöntemler, kriptografik yöntemler, geriye dönük anahtar yayınlama yöntemi, çoklu algılama ve konumlama yöntemi, mesafe bazlı protokoller olarak sıralanmaktadır. Gelecek çalışma olarak ise ADS-B verisinin asimetrik açık anahtar şifrleme metodlarıyla gerçeklenmesi amaçlanmıştır.

Li ve arkadaşları çalışmasında; mesaj kesme, değiştirme veya slime gibi saldırılara karşı LSTM Encoder-Decoder mimarisi ile SVDD (Support Vector Data Description) sınıflandırma yöntemi kullanarak anomilerlin tespit edilmesi amaçlanmıştır (Li et al., 2021). LSTM mimarisi; recurrent neural network bazlı bir yapı oluşturmaktadır; zaman bazlı problemlerde kullanılan bu yöntem input olarak almış olduğu veriyi encoder kısmında ilgili zaman dilimlerine bölerek işledikten sonra decoder kısmında bir sonuç (output) üretir. SVDD sınıflandırma yöntemini ve algoritmasını anlatıldığı

çalışmanın, analiz ve sonuç kısımlarında, çözüm modelleri ve kullanılan yöntemler sayısal olarak karşılaştırılmıştır.

Khandker ve arkadaşları çalışmasında; OCSVM, IFOREST, LTSM ve LTSM_ENDED SVDD kullanarak karşılaştırma yapmıştır (Khandker et al., 2022). Hesaplama bileşenlerinde doğruluk verileri ile karşılaştırılarak LTSM_ENDED SVDD çok daha doğru sonucu verdiği gözlemlenmiştir.

Khan ve arkadaşları çalışmasında; ADS-B receiver'ının uçağın içerisinde bulunan EFB (Elektronik Flight Bag) cihazı ve yazılımı ile bağlı olarak değerlendirilmiştir (Khan et al., 2021). ADS-B cihazına veya kokpit içerisindeki EFB cihazına yapılan saldırılar açıklanmıştır. Araştırmacılar bu yayında makine öğrenmesi için dört saldırı veri seti kullanılmıştır: Jumping saldırısı, yanlış bilgi saldırısı (false information attack), yanlış yön saldırısı (false heading attack), yanlış squawk kodu atağı. Veri setine öncelikle ön işlem (remove missing values, removing outliers, data transformation), ardından Logistic regression, K-Nearest Neighbor ve Naive bayes makine öğrenmesi algoritmaları uygulanmaştır. Yazarlar, değerlendirme yapmış olduğu veri setindeki özelliklere göre bu algoritmalar arasında karşılaştırma yaparak KNN algoritmasının daha başarılı olduğu sonucuna ulaşmıştır.

Kacem ve arkadaşları çalışmasında; multiclass sınıflandırma yapmayı amaçlamıştır. Support Vector Machine, Decision Tree ve Random Forest makine öğrenmesi teknikleri kullanılmıştır (Kacem et al., 2021). Modellerin performansını hesaplamak için Five-fold cross-validation uygulanmıştır. Üç saldırı tipi ele alınmıştır; replay attack, ghost aircraft injection, multiple ghost aircraft injection. Kacem ve arkadaşları çalışmasında öğrenme esnasında; enlem, boylam, metre cinsinden yükseklik farkı, feet cinsinden deniz seviyesinden yükseklik, metre cinsinden aradaki mesafe, metre cinsinden kümülatif mesafe, rota açısı ve son olarak uçuş kimliği özelliklerini kullanmıştır (Kacem et al., 2021). Kullanmış olduğu makine öğrenmesi sınıflandırma modeli tasarımında orjinal veri seti saldırıya uğradıktan sonra iki ayrı veri ortaya çıkmaktadır: orjinal ve bozulmuş veri setleri sonrasında yapay öğrenme teknikleri kullanılıp performans hesaplamaları yapılmıştır. Kacem ve arkadarşları çalışmasında; üç ayrı, %5, %10, %15 oranlarında bozulmuş veri setleri kullanmıştır. Bunların üzerinden üç metrik hesaplanmıştır: siniflandirma doğruluğu (classification accuracy), hassaslık-duyarlılık (sensitivity), özgüllük (specificity). Karşılaştırmalı olarak modelde kullanılan orjinal ve değiştirilmiş veri setlerinin sınıflandırmaları

yapılmıştır. Burada elde edilen sonuçlarda Random Forest tekniğinin daha iyi olduğu ve SVM'nin ise daha az başarılı performans sergilediği belirtilmiştir.

Shang ve arkadaşları çalışmasında; kullanılan veri setini ADS-B cihazı ile toplamıştır. Elde edilen verilerin şifrelenme yöntemleri ile yeniden şifrenin çözülmesi yöntemleri arasında hata oranları karşılaştırılmıştır (Shang et al., 2020). Karşılaştırılan şifrelenme yöntemleri Manchester Encoding, Basic iteration Method (BIM) teknikleridir. Karşılarştırılan test teknikleri test verisi oranı üzerinden de karşılaştırmaya eklenmiştir. Burada elde edilen sonuç, MEA ile şifreleme ve deşifreleme yöntemlerinin daha doğru veri elde edileceğini yönündedir.

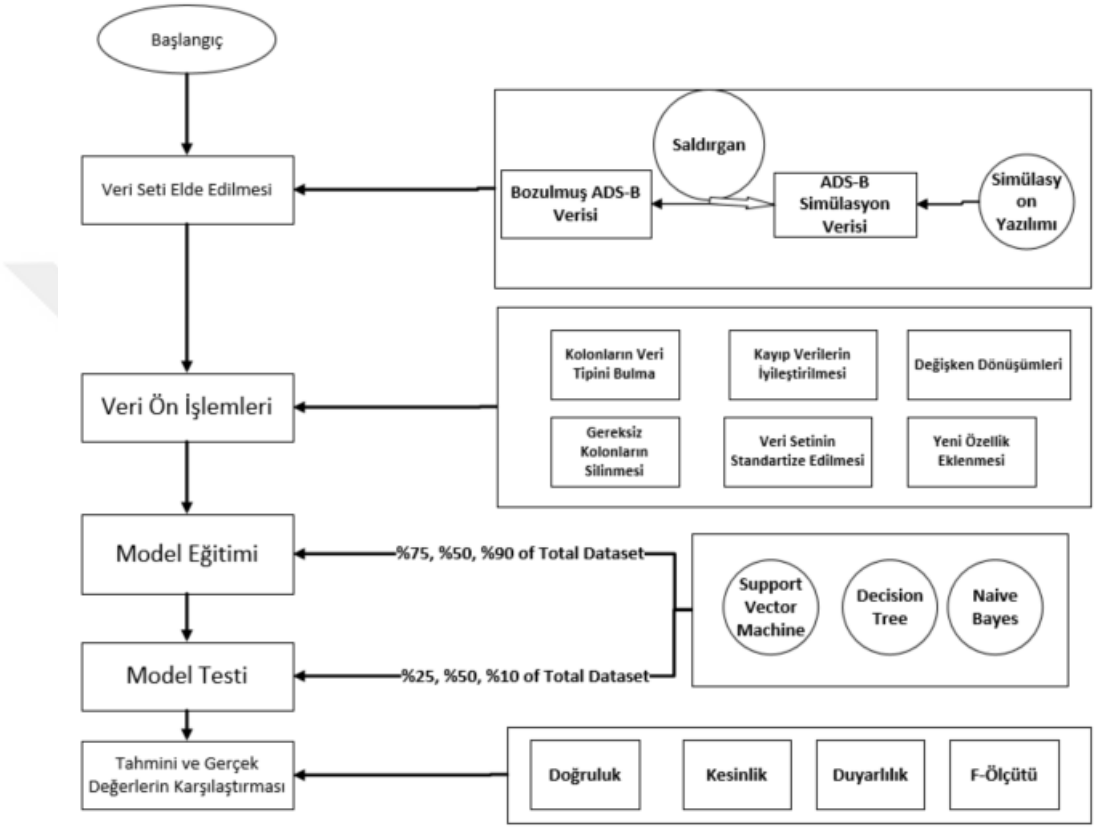
Ying ve arkadaşları kullanılacak veri setini ADS-B alıcısı ve SDR ile etmiştir. Over The Air (OTA) uygulaması ile spoofing atakları gerçekleştirilmiştir (Ying et al., 2019). Veri setini %60'ı öğrenmek, %20'si doğrulama (validation) ve %20'si de test için kullanılmıştır. 5 Model kullanılmıştır. Bunların karmaşıklığı:

- M1: 512 düğümlü bir gizli katman,
- M2: 1024 düğümlü bir gelişmiş gizli katman,
- M3: Katman başına 512 düğüm içeren iki gizli katman ve
- M4: Katman başına 512 düğüm içeren üç gizli katman.
- M5: İki gizli katman (512 ve 256 düğüm), toplu normalleştirme ve 200 (epoch) dönem içeren ince ayarlı bir model.

Metrikler ise; kesinlik, duyarlılık ve f-ölçütü. Bunlarla ilgili çıkan sonuç ise; 5 modelin karşılaştırılması sonucunda yakın değerler çıkmıştır. Ancak yapılan çalışma sonucunda 5. Modelin en uygun model olduğu gözlemlenmiştir.

3. ÖNERİLEN SİSTEM MODELİ

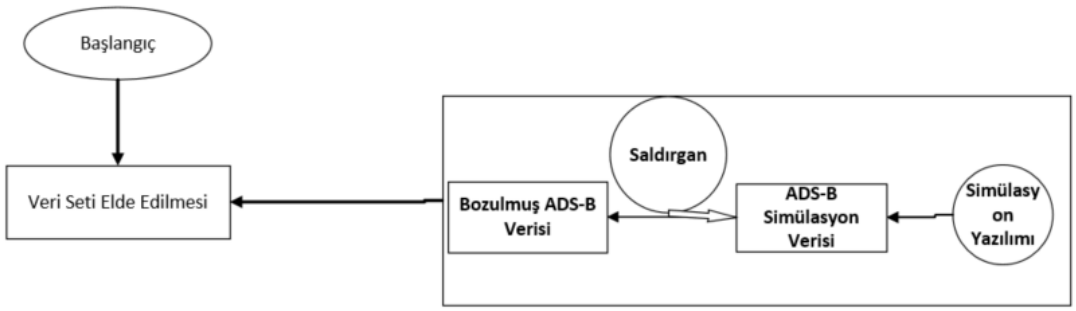
Bu bölümde ADS-B cihazlarına yapılan saldırılara karşı önerilen sistem modeli sunulmuştur. Tez kapsamında sunulan sistem modelinde; veri setinin elde edilmesi, veri setine uygulanan ön işlemler ve sonrasında kullanılan yapay öğrenme tekniklerindeki modelin eğitimi ve modelin testi aşamaları bulunmaktadır. Önerilen sistem modelin Şekil 3.1'de işlem adımları gösterilmiştir.



Şekil 3.1 : Önerilen Sistem Modeli

3.1 Veri Setinin Elde Edilmesi

Çalışmada kullanılacak veri setinin elde edilebilmesi için ilk olarak ADS-B verisi elde edilmiştir. Hava aracı simülasyon yazılımından elde edilen bu orijinal veriye daha sonra sahte veri saldırısı yapılarak manipüle edilmiş veri elde edilmiştir. Sistem modelinin bu adımı Şekil 3.2'de gösterilmektedir.



Şekil 3.2 : Veri Setinin Elde Edilme Modeli

3.1.1 ADS-B Veri Setinin Elde Edilmesi

Hava taşıtı simülasyon yazılımı tarafından yayınlanan toplam 64 uçak bulunduran yaklaşık 40 dakikalık simulasyon uçuş verileri ATCSIMTEST yazılımı tarafından dinlenmektedir. ATCSIMTEST yazılımı, hava taşıtı simülasyon nesnelerinden asterix formatındaki ADS-B verisine dönüştürmektedir. ADS-B verisi olarak yayınlanan veriyi anlık olarak csv olarak da dışarı çıkarılabilmektedir. Bu dışarı çıkarılan veri saldırı yapılmamış veri olarak kullanılabilecektir. Dışarı çıkarılan bozulmamış veri setinin örneği Tablo 3.1'de gösterilmiştir.

Tablo 3.1: Bozulmamış ADS-B Verisinin Örneği

TimeStamp	Callsign	Mode-S	SSR Code	latitude	Longtitude	Flight Leve	Speed
2023-01-0	KLMT7	4BE2A3	2010	3095711.3	3705802.1	124800.0	433.59814
2023-01-0	SHYT3	4BF5DA	2004	3431164.0	3644750.1	145600.0	424.91446
2023-01-0	XXXT18	4BFAC6	2023	3311040.7	3724299.5	122000.0	474.49606
2023-01-0	AZAT20	4B871C	2025	2821323.8	4108767.3	121600.0	459.2125
2023-01-0	AJAT59	4BCB7E	2074	2493283.0	4239840.2	157600.0	446.9952

ADS-B verilerinde bir uçuş trafiğine ait bir kayıtta aşağıdaki bilgiler bulunmaktadır:

- *TimeStamp:* Radardan gelen verinin anlık zamanı milisaniye cinsindendir.
- *Callsign:* İlgili trafiğin telsiz çağrı kodudur. Uçuşa verilen tekil bir isimdir.
- *Mode-S:* Uçakların 24 bit uzunluğunda bireysel seçici (S= Selective) adresleme ile kodlanmasına dayanır. Her hava aracına bireysel bir kod ataması yapılır (SHGM, n.d.).
- *SSR Code:* Bir uçuşa ait 4 haneli sekizli tabanda verilen koddur. Uçak tarafından değiştirilebilir. Acil kod numaraları bulunmaktadır.
- *Latitude:* Uçağın anlık enlemidir.
- *Longtitude:* Uçağın anlık boylamıdır.
- *Speed:* Uçağın anlık hızıdır.
- *Flight Level:* Uçağın anlık uçuş seviyesi, deniz seviyesinden yüksekliğidir.

Hava taşıtı simülasyon yazılımından yayınlanan simülasyon verisini kullanan ATCSIMTEST yazılımı tam bir ADS-B verisi oluşturulabilmektedir.

3.1.2 Saldırı Yapılmış ADS-B Veri Setinin Elde Edilmesi

Tez çalışmasında saldırı yapılmış olan ADS-B verisi kullanılacağı için saldırı yapılmış hale getirmek gereklidir. Bu çalışmada ADS-B mesajında bulunan SSR Code ve hız bilgileri manipüle edilmiştir. Elde edilen veri setinde 'spoofed' etiketi ile verinin bozulup bozulmadığı da eklenmiştir. Veri setinde bulunan kayıt sayısı 23194'tür. Sonuç etiketinde bulunan anomali kayıt sayısı 4406, normal kayıt sayısı ise 18788'dir. Örnek csv çıktısı Tablo 3.2'de gösterilmiştir.

Tablo 3.2: Bozulmuş ADS-B Verisi Örneği

TimeStamp	Callsign	Mode-S	SSR Code	latitude	Longtitude	Flight Leve	Speed	isSpoofed
2023-01-02T16:5	KLMT7	4BE2A3	7600	3095711	3705802	124800	433.5981	TRUE
2023-01-02T16:5	SHYT3	4BF5DA	2004	3431164	3644750	145600	424.9145	FALSE
2023-01-02T16:5	XXXT18	4BFAC6	2023	3311041	3724300	122000	474.4961	FALSE
2023-01-02T16:5	AZAT20	4B871C	2025	2821324	4108767	121600	459.2125	FALSE

Sistem modelinde kullanılmak için orijinal ADS-B verisinden, saldırı yapılmış ADS-B verisi elde edilmesi istenmiştir. Bu işlem için kullanılan saldırıların konfigürasyonu, ATCSIMTEST yazılımı içerisinde bulunan konfigürasyon dosyasında bulunmaktadır. Bu dosya attackMode ve attackLevel değerleri ile ayarlanmaktadır. Eğer attackMode 1 ise yalnızca uçuş yüksekliğinin bozulmasını, attackMode 2 ise yalnızca uçuş SSR code değerinin yanlış gönderilmesini, eğer attackMode 3 ise hem uçuş yüksekliği hem de SSR code değerlerinin bozulmasını sağlayacaktır. Konfigürasyonda bulunan attackLevel değeri ile 100 değeri çarpılıp uçuşun yüksekliğinin feet cinsinden değeri eklenmiştir. İlgili kod parçası Ekler bölümünde EK-1'te verilmiştir.

Bu çalışmada attackMode değeri 3 olarak ayarlanarak her iki saldırı da modellenmiştir. attackLevel değeri ise 4 olarak ayarlanarak bazı trafiklerin 400 feet eklenerek uçuş seviyelerinin bozulmasına neden olunmuştur. ATCSIMTEST yazılımı Intellij geliştirici aracında Java 8 versiyonu ile geliştirilmiştir. ATCSIMTEST yazılımının bozulmamış ADS-B ve bozulmuş ADS-B verisi elde etme için kullandığı kod parçaları Ekler bölümünde EK-2'te verilmiştir.

Sahte SSR Code Saldırısı : Her trafiğin kendine özgü SSR code değeri bulunmaktadır. Bu nedenle burada yapılacak olan değişiklik hem Hava Trafiğini yöneten tarafta hem de uçak tarafında problemlere sebep olacaktır. Hava trafiğinin yönetiminde kontrolör tarafında farklı bir SSR code ile trafik verisinin gelmesi,

kontrolörün önünde bulunan yardımcı araçlarda ilgili trafiğin bilgilerinin yanlış değerlendirilmesine sebep olacaktır. Ayrıca yeni geliştirilen veri bağlantısı araçları da bunlardan etkilenecektir. SSR code değeri 8'lik tabanda değerler ile ifade edilmektedir. Ancak 7 ile başlayan SSR code değerleri alarm anlamına gelmektedir. Bunlar;

- 7500 - uçak kaçırma,
- 7600 - iletişim problemi
- 7700 - acil durum

Bu çalışmada bazı trafiklere saldırı yapılarak SSR code değerleri 7600 ile değiştirip yayınlanmıştır. Hava trafiğini yöneten tarafa yanlış bir ikaz gitmesine sebep olan bu hata kontrolörün ve karar destek yazılımlarının yanlış karar vermelerine sebep olacaktır.

Hava taşıtının ADS-B mesajında bulunan anlık uçuş yüksekliğinin bozulması trafiğin anlık uçuş seviyesine bağlı olan tüm problemleri beraberinde getirecektir. Hava trafiği yönetiminde bulunan kontrolör ile hava taşıtı arasındaki iletişimde sürekli yanlış bilgiye dayalı olarak iletişim gerçekleşecektir. Hava aracının kaza kırımına dahi sebep olabilir.

3.1.3 Veri Setine Uygulanan Ön İşlemler

Önerilen sistem modelinde veri seti üzerinde uygulanan ön işlemler aşağıda detaylandırılmıştır.

- *Verileri Anlama:* Keşif analizi, verilerin özelliklerini ve dağılımını anlamaya yardımcı olur. Verilerdeki kalıpları, eğilimleri ve aykırı değerleri belirlemeye yardımcı olur. Bu çalışmada; yeni özellik ekleme, kolonların veri tipini ve sayısını bulma gibi işlemler yapılarak verinin özellikleri öğrenilmiştir ve yapay öğrenme tekniklerinde kullanıma uygun hale getirmek için veri seti anlamlandırılmıştır.
- *Veri Temizleme ve Ön Işleme:* Keşif analizi, eksik değerler, tutarsızlıklar ve aykırı değerler gibi veri kalitesi sorunlarını belirlemeye yardımcı olur. Verilerin doğruluğunu ve eksiksizliğini sağlamak için veri temizleme ve ön işlemler yapılmalıdır. Bu çalışmada; verilerin standardize edilmesi, eksik verilerin işlenmesi gibi işlemler yapılarak verilerin yapay öğrenme tekniklerinde kullanılırken hem hata almaması hem de ortalama bir değer alarak modele katkıda bulunması amaçlanmıştır.

- *Özellik Seçimi:* Keşif analizi, sonuç değişkeni üzerinde en fazla etkiye sahip olan özellikleri belirlemeye yardımcı olur. Öngörü modellerinin doğruluğunu ve genelleştirilmesini geliştirebilen özellik seçimine izin verir. Bu aşamada ise gereksiz kolonlar silinerek önerilen sistem modeli işlemciyi zorlamaması ve daha doğru sonuçlara ulaşmamız amaçlanmıştır.

Yapay öğrenme teknikleri kullanmadan önce daha iyi performans ve daha doğru sonuçlar elde etmek için veri setleri ön işlemlerden geçmektedir. Şekil 3.3'te bu çalışma kapsamında kullanılan ön işlemler verilmiştir. Yapılan işlemlerin kodları Ekler bölümünde EK-3'te verilmiştir.

Şekil 3.3: Veri Setine Uygulanan Analiz İşlemleri.

Yeni Özellik Ekleme: Callsign bilgisi bir uçuşun tekil değeridir. Burada kullanılan değer ise karakter tipindedir. Bu nedenle modele dahil ederken bu kolondaki değerler nümerik değerlere dönüştürülüp 'CallSignNumeric' adında bir kolon oluşturumuştur.

Kolonların Veri Tipini Bulma: Bu analiz işleminde veri setinde bulunan tüm kayıtların durumu gösterilmektedir. Sistem modelinde kullanılan orijinal veri setine ait bilgiler burada gösterilmektedir. Boyut ve tipleri ile birlikte her bir kolonun bilgileri aşağıda Şekil 3.4'te verilmiştir.

```
#   Column           Non-Null Count  Dtype
---  ------           --------------  -----
0   TimeStamp        23194 non-null  object
1   Callsign         23194 non-null  object
2   CallSignNumeric  23194 non-null  int64
3   Mode-S           23194 non-null  object
4   SSR Code         23194 non-null  int64
5   latitude         23194 non-null  float64
6   Longtitude       23194 non-null  float64
7   Flight Level     23082 non-null  float64
8   Speed            23194 non-null  float64
9   isSpoofed        23194 non-null  bool
```

Şekil 3.4: Bozulmamış Verinin Özellikleri

Gereksiz Kolonların Silinmesi: Burada önerilen sistem modelinde kullanılmayacak olan kolonlar silinmektedir. Bu sayede işlemin performansı artırılıp kullanılacak verinin de alanı azaltılmış olur. Bu çalışmada trafiğin 'latitude', 'longtitude', 'TimeStamp', 'Callsign', 'Mode-S' kolonları gereksiz olduğu için silinmiştir.

Eksik Verilerin İşlenmesi: Kullanılacak olan kayıtlardaki verilerin bazıları boş kalmış olabilir. Bunları hesaplamayı bozmaması için bir değer atanmalıdır. Bu çalışmada ortalama değer atanmıştır. 'Flight Level' kolonun eksik bulunan kayıtlara 'Flight Level' kolonunun ortalama değeri atanmıştır.

Veri Setinin Standartize Edilmesi: Bu aşama ile birlikte kullanılan değerler standardize edilmiş olacaktır. StandardScaler ortalamayı kaldırır ve her özelliği/değişkeni birim varyansa göre ölçeklendirir. Bu işlem, özellik bazında bağımsız bir şekilde gerçekleştirilir. StandardScaler, ampirik ortalamanın tahminini ve her özelliğin standart sapmasını içerdiğinden (veri kümesinde varsa) aykırı değerlerden etkilenebilir (Manikanth, 2021).

Bu çalışmada veri seti standardize edilmeden önce öğrenme ve doğrulama veri setleri olarak ayrılmıştır. Sonuç etiketi olarak 'isSpoofed' etiketi olacağı belirtilmiştir.

3.2 Model Eğitimi ve Model Testi

Bu kısımda önerilen sistem modelindeki veri setinin eğitilmesi ve testinde kullanılan yapay öğrenme teknikleri ve bu teknikler için kullanılan değişkenlerin değerleri verilmiştir. Performans sonuçları 4. Bölüm'de gösterilmiştir. Bir sonraki alt başlıkta önerilen sistem modelinde kullanılan hesaplama metrikleri tanıtılmıştır. Kullanılan kod parçaları Ekler bölümündeki EK-2'de verilmiştir.

3.2.1 Hesaplamada Kullanılacak Değerler

Denetimli öğrenme algoritmalarının değerlendirmesi, karmaşıklık matrisindeki değişkenler kullanılarak yapılmaktadır. Hangi denetimli öğrenme algoritmasının daha iyi olduğuna karar vermek için hesaplanan verilerin temelini, Doğru Pozitif (True Positive), Doğru Negatif (True Negative), Yanlış Pozitif (False Positive) ve Yanlış Negatif (False Negative) olarak nitelendirilen dört adet değişken oluşturmakta ve bu dört değişkenin nasıl hesaplandığı Tablo 3.3'te gösterilmiştir (Nguyen & Armitage, 2008).

Karmaşıklık matrisindeki diyagonal hücreler doğru tespit edilen veri sayısını, diğer hücreler ise hatalı tespit sayısını göstermektedir (Deshmukh et al., 2015).

Tablo 3.3: Karmaşıklık Matrisi

Karmaşıklık Matrisi		Tahmin Edilen Sınıf	
		Negatif	**Pozitif**
Gerçek Sınıf	**Negatif**	Doğru Negatif	Yanlış Pozitif
	Pozitif	Yanlış Negatif	Doğru Pozitif

Tabloda gösterildiği gibi:

- Doğru Pozitif (DP) değeri, gerçekte A sınıfına ait olan ve A sınıfına ait olduğu tahmin edilen veri sayısını,

- Doğru Negatif (DN) değeri, gerçekte A sınıfına ait olmayan ve A sınıfına ait olmadığı tahmin edilen veri sayısını,

- Yanlış Pozitif (YP) değeri, gerçekte A sınıfına ait olmayan ancak A sınıfına ait olduğu tahmin edilen veri sayısını,

- Yanlış Negatif (YN) değeri, gerçekte A sınıfına ait olan ancak A sınıfına ait olmadığı tahmin edilen veri sayısını göstermektedir.

Karmaşıklık matrisinde karşımıza çıkan bu dört değişken kullanılarak, Doğruluk (Accuracy), Kesinlik (Precision), Duyarlılık (Recall) ve F-Ölçütü hesaplanıp denetimli öğrenme algoritmalarının performans değerlendirmesi yapılmaktadır (Yiğidim, 2012). Doğruluk, doğru olarak tahmin edilen verilerin toplam veriye oranı olarak ifade edilmektedir. Sınıflandırma algoritmasının performansını ortaya koyan önemli bir ölçüttür. Eşitlik ifadesi aşağıdaki gibidir:

$$\boldsymbol{Doğruluk} = \frac{\boldsymbol{DP+DN}}{\boldsymbol{DP+YP+DN+YN}} \tag{3.1}$$

Kesinlik, doğru olarak tahmin edilen verilerin doğru tahmin edilen toplam veri sayısına oranıdır. Eşitlik ifadesi aşağıdaki gibidir:

$$\boldsymbol{Kesinlik} = \frac{\boldsymbol{DP}}{\boldsymbol{DP+YP}} \tag{3.2}$$

Duyarlılık, doğru olarak tahmin edilen verilerin gerçekte o sınıfa ait verilerin sayısına oranıdır. Algoritmanın veriyi hangi oranla doğru tahmin ettiğini göstermektedir. Eşitlik olarak ifadesi aşağıdaki gibidir:

$$\boldsymbol{Duyarlılık} = \frac{\boldsymbol{DP}}{\boldsymbol{DP+YN}} \tag{3.3}$$

F-ölçütü, Kesinlik ve Duyarlılık verilerinin harmonik ortalaması alınarak bulunmaktadır. Bu nedenle her iki veriyi ayrı ayrı kullanmak yerine bu veri kullanılarak, denetimli öğrenme algoritmalarının kıyaslaması yapılabilir. Eşitlik olarak ifadesi aşağıda olduğu gibidir:

$$\boldsymbol{F-ölçütü} = \frac{\boldsymbol{2\ x\ Duyarlılık\ x\ Kesinlik}}{\boldsymbol{Duyarlılık+Kesinlik}} \tag{3.4}$$

3.2.2 Kullanılan Yapay Öğrenme Teknikleri

Çalışmada aşağıda belirtilen gözetimli öğrenme teknikleri kullanılmıştır.

İkili Karar Ağacı: İkili karar ağacı, verileri sınıflandırmak için kullanılan bir dizi kuralın grafiksel bir temsilidir. İkili karar ağacı, bir durdurma kriterine ulaşılana kadar belirli bir özelliğin değerine dayalı olarak veriyi yinelemeli olarak alt kümelere bölerek oluşturulur.

Naive Bayes: Naive Bayes, Bayes teoremine dayanan olasılıksal bir algoritmadır. Girdi özellikleri verilen her sınıfın olasılığını hesaplar ve en yüksek olasılığa sahip sınıfı seçer.

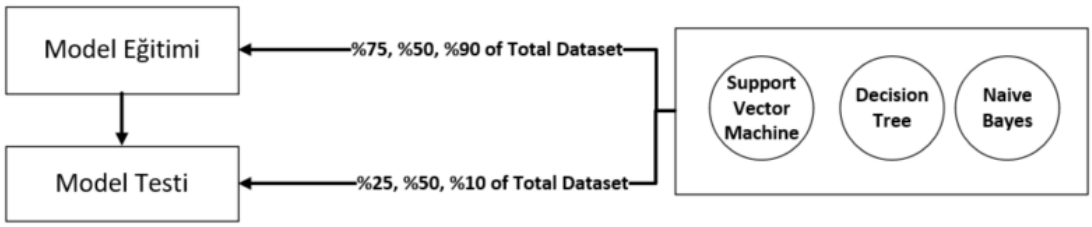
Support Vector Machines: Sınıflar arasındaki marjı maksimize ederek sınıflar arasındaki en iyi sınırı bulan bir sınıflandırma algoritmasıdır. Verileri, doğrusal bir sınırın bulunabileceği daha yüksek boyutlu bir alana dönüştürerek çalışır.

Bu bölümde sistem modelinde kullanılan Support Vector Machines, Naive Bayes ve İkili Karar Ağaçları tekniklerinin uygulamalarına yer verilmiştir. Tüm teknikler yukarıda kullanılan ön işlemlerden geçmiştir. Standart ön işlem aşamaları uygulanmıştır.

Veri analizi yapıldıktan sonrasında ön işlemlerden geçmiş olan veri seti, öğrenme ve doğrulama veri setleri olarak bölünmüştür. Bu çalışmada 3 farklı oranda ayırma yapılarak teknikler uygulanmıştır.

- Veri setinin %50'si modelin eğitimi %50'si modelin testi,
- Veri setinin %75'i modelin eğitimi %25'i modelin testi,
- Veri setinin %90'ı modelin eğitimi %10'u modelin testi

için kullanılarak teknikler uygulanmıştır. Şekil 3.5'te modelin bu adımı gösterilmiştir.



Şekil 3.5: Modelin Eğitim ve Test Adımı

3.2.2.1 Support Vector Machine (SVM) Tekniği Uygulaması

Support Vector Machine, sınıflandırma ve regresyon analizi için yaygın olarak kullanılan bir makine öğrenme algoritmasıdır. Veri kümesindeki sınıfları maksimum düzeyde ayıran yüksek boyutlu bir uzayda hiperdüzlemi bularak çalışır.

Basit bir ifadeyle; SVM, sınıflar arasındaki marjı en üst düzeye çıkaracak şekilde sınıfları ayıran en iyi çizgiyi veya sınırı bulmaya çalışır. Kenar boşluğu, hiperdüzlem ile her sınıftan en yakın veri noktaları arasındaki mesafedir. SVM'nin amacı, daha sağlam ve genelleştirilebilir bir model ile sonuçlanan en büyük marja sahip hiperdüzlemi bulmaktır (Berwick, n.d.).

SVM, orijinal veri kümesini bir çekirdek işlevi kullanarak daha yüksek boyutlu bir alana dönüştürerek çalışır. Bu, algoritmanın orijinal özellik uzayında doğrusal olarak ayrılamaz durumda olsalar bile sınıfları ayıran bir hiperdüzlem bulmasını sağlar. SVM, doğrusal, polinom ve radyal temel işlev (RBF) çekirdekleri dahil olmak üzere çeşitli çekirdek işlevlerine sahiptir. Bu çekirdek tipleri verilerin koordinatlarını daha yüksek boyutlu bir uzayda hesaplamadan orijinal özellik uzayında işlem yapmamızı sağlar. Bu çalışmada doğrusal ve radial temel işlevi çekirdek tipleri kullanılmıştır.

SVM tekniğinde C parametresinin de önemli bir rolü bulunmaktadır. C, SVM'nin ihlaller konusunda ne kadar ciddi olduğunu belirler. C 0 ise, ceza süresi ortadan kalktığı için SVM ihlalleri hiç umursamaz. C çok büyükse, küçük ihlaller amaç fonksiyonunda büyük bir artışa yol açacaktır (*Support Vector Machine Explained-Theory, Implementation, and Visualization*, n.d.).

Tekniği uygularken C değişkenini {0.1, 0.01, 0.001} değerleri kernel tipi 'RBF' ve 'Linear' iken test edilmiştir. Performans Değerlendirmesi bölümünde tablo halinde kıyaslaması yapılmıştır.

C değeri 0.1 olarak RBF ve Linear kernel tipleri için veri setlerinin test oranları üzerinden bir sistem modeli kullanılmıştır. Kullanılan bu sistem modelinin sonuçlarına Performans Değerlendirmesi bölümünde yer verilmiştir.

3.2.2.2 Naive Bayes Tekniği Uygulaması

Naive Bayes, Bayes Teoremine dayalı istatistiksel bir sınıflandırma tekniğidir. En basit denetimli öğrenme algoritmalarından biridir. Naive Bayes sınıflandırıcısı hızlı, doğru ve güvenilir bir algoritmadır. Naive Bayes sınıflandırıcıları, büyük veri kümelerinde yüksek doğruluk ve hıza sahiptir.

Kelime anlamı "saf" olan bu tekniğin bir veri kümesinin özelliklerinin, gerçekte ilişkili olsalar bile, birbirinden bağımsız olduğu varsayımını yapar.

Naive Bayes sınıflandırmasında algoritma, özelliklerin değerlerine dayalı olarak her bir sınıfın olasılık dağılımını öğrenmek için eğitim verilerini kullanır. Ardından, bir dizi özellik değerine sahip yeni bir örnek verildiğinde, algoritma, öğrenilen

dağılımlara dayanarak örneğin her bir sınıfa ait olma olasılığını hesaplar. En yüksek olasılığa sahip sınıf daha sonra örneğe atanır.

Naive Bayes'in basitliği, hızı ve yüksek boyutlu veri kümelerindeki etkinliği gibi çeşitli avantajları vardır. Doğal dil işleme ve metin sınıflandırma görevlerinin yanı sıra spam filtreleme ve duyarlılık analizinde yaygın olarak kullanılmaktadır (*Naive Bayes Classifier Tutorial*, n.d.) .

$$\boldsymbol{P(h|D)} = \frac{\boldsymbol{P(D|h)P(h)}}{\mathbf{P(D)}} \qquad (3.5)$$

Yukarıda verilen eşitlikte: (Shoba R., 2019)

- P(h): h hipotezinin doğru olma olasılığı (verilerden bağımsız olarak). Bu, h'nin ön olasılığı olarak bilinir.
- P(D): Verilerin olasılığı (hipotezden bağımsız olarak). Bu ön olasılık olarak bilinir.
- P(h|D): D verisi verildiğinde h hipotezinin olasılığı. Bu, arka (sonrası) olasılık olarak bilinir.
- P(D|h): h hipotezinin doğru olması durumunda D verisinin olasılığı. Bu, arka (sonrası) olasılık olarak bilinir.

Bu çalışma kapsamında Gaussian Naive Bayes, Categorical (Kategorik) Naive Bayes ve Multinomial Naive Bayes teknikleri uygulanmıştır.

Gauss Naive Bayes, Multinomial Naive Bayes ve Kategorik Naive Bayes, Naive Bayes algoritmasının sınıflandırma problemlerinde yaygın olarak kullanılan üç türüdür. Bu üç değişken arasındaki temel fark, işleyebilecekleri veri türünde ve varsaydıkları olasılık dağılımında yatmaktadır.

Gaussian Naive Bayes (GNB): Gaussian Naive Bayes, sürekli veriler için kullanılır ve özelliklerin bir Gauss dağılımını (yani normal dağılımı) takip ettiğini varsayar. Naive Bayes'in bu varyantı, özelliklerin kelime frekanslarının olduğu metin sınıflandırma problemlerinde sıklıkla kullanılır.

Multinomial Naive Bayes (MNB): Multinomial Naive Bayes, kelime sayıları veya belge frekansları gibi ayrık veriler için kullanılır. Özniteliklerin çok terimli bir dağılımdan üretildiğini varsayar. Naive Bayes'in bu varyantı, metin sınıflandırma problemlerinde yaygın olarak kullanılır ("Naive Bayes Classifiers," 2017).

Categorical Naive Bayes (CNB): Categorical Naive Bayes, özelliklerin ikili olduğu (yani, 0 veya 1 gibi yalnızca iki değer alabildikleri) Multinomial Naive Bayes'in özel bir durumudur. Özelliklerin bir Bernoulli dağılımından üretildiğini varsayar. Naive

Bayes'in bu varyantı genellikle spam filtrelemede, duyarlılık analizinde ve diğer ikili sınıflandırma problemlerinde kullanılır.

Çalışma esnasında GNB, MNB ve CNB algoritma tiplerinin test oranı üzerinden model gerçeklenmiştir. Yapay öğrenme tekniklerinin karşılaştırmalı tablosuna ve kıyaslanıp sonuçlarına Performans Değerlendirmesi bölümünde yer verilmiştir.

3.2.2.3 İkili Karar Ağacı Tekniği Uygulaması

İkili karar ağacı algoritmasında veri seti eğitilerek ikili karar ağacı oluşturulmaktadır. Sonrasında bu ikili karar ağacı kullanılarak test verisindeki sınıflandırılacak verinin sınıfı belirlenir. İkili karar ağacı oluşturulurken öncelikle kök düğüm belirlenir. Kök düğüm belirlenirken örnekleri en iyi ayıran özellik seçilir. Sonrasında bu işlem yaprak düğümlerde tekrarlanarak ağacın yapısı belirlenir (Aksu & Dogan, 2019).

İkili karar ağacı, makine öğrenimi ve veri madenciliğinde kullanılan bir tür denetimli öğrenme algoritmasıdır. Ağacın her bir iç düğümünün bir öznitelik üzerindeki testi temsil ettiği, her dalın testin sonucunu temsil ettiği ve her yaprak düğümün bir kararı veya sonucu temsil ettiğini gösteren karar verme sürecinin grafiksel bir temsilidir.

İkili karar ağaçları, amacın niteliklerine dayalı olarak yeni bir örneğin sınıf etiketini tahmin etmek olduğu sınıflandırma görevleri için veya amacın sürekli bir hedef değişkeni tahmin etmek olduğu regresyon görevleri için kullanılabilir. İkili karar ağaçları, basitlikleri, yorumlanabilirlikleri ve çok çeşitli veri setlerinde iyi performans göstermeleri nedeniyle finans, sağlık, pazarlama ve mühendislik gibi çeşitli alanlarda yaygın olarak kullanılmaktadır.

Bu çalışmada ikili karar ağacı tekniği test oranları farklı değerlerde kullanılarak ayrı ayrı sonuçlar elde edilmiştir. Elde edilen bu sonuçlara Performans Değerlendirmesi bölümünde yer verilmiştir.

4. PERFORMANS DEĞERLENDİRMESİ

Tezin bu bölümünde yapılan çalışmaların ve hesaplamaların değerlendirilmesi verilecektir.

Tezde kullanılan veri setinin elde edilmesi için ATCTRSIM hava trafik kontrol yazılımı simülasyon ortamı kullanılmıştır. Bu simülasyon ortamında 64 adet hava aracı kullanılmış ve 39 dakika 2 saniye boyunca çalıştırılmıştır. Simülasyon yazılımından elde edilen simülasyon nesneleri ATCSIMTEST yazılımı aracılığı ile saldırı yapılmış csv formatında ADS-B veri setine dönüştürülmüştür. Elde edilmiş olan ADS-B veri setinde 23194 kayıt bulunmaktadır. Bu kayıtların 4406 tanesi bozulmuş veri, 18788 tanesi ise bozulmamış veri olarak etiketlenmiştir. Kullanılan veri setinde veri oluşturma zamanı olan TimeStamp, uçağın tekil olmasını sağlayan Callsign, uçak ve kontrolörün haberleşmesi için kullanılan Mode-S, hava trafiğinde kullanılan SSR Code, Enlem, Boylam, Uçuş Seviyesi, hız ve saldırı yapılmış etiketi isSpoofed bulunmaktadır. Kullanılan veri setinde yapay öğrenme tekniklerinde kullanılırken ilk olarak ön işlemler gerçekleştirilmiştir. Bu işlemlerin yapıldığı örnek kod parçası Ekler bölümündeki EK-3'te verilmiştir.

Bu çalışma kapsamında kullanılan yapay öğrenme teknikleri doğruluk, kesinlik, duyarlılık ve F-ölçütleri kıyaslanarak değerlendirilmiştir. Öncelikli olarak kullanılan yapay öğrenme tekniklerinin sonuçları değerlendirilmiştir. Daha sonra en iyi test oranı sabit tutularak yapay öğrenme teknikleri karşılaştırılarak değerlendirilmiştir. Bu adımın modeldeki yeri Şekil 4.1'de verilmiştir.

Şekil 4.1 Tahmin ve Karşılaştırma Adımı

Support Vector Machine yapay öğrenme tekniğinde C değerinin Kernel bazında farklı değerler ile uygulanması sonucunda skor tablosu Tablo 4.1'de gösterilmiştir.

Tablo 4.1: C Değerinin SVM'de Kernel Bazında Karşılaştırılması

	Kernel					
	RBF			Linear		
C	Doğruluk	Kesinlik	Duyarlılık	Doğruluk	Kesinlik	Duyarlılık
0.1	0.907	1	0.5	0.895	0.861	0.511
0.01	0.895	0.862	0.511	0.895	0.861	0.511
0.001	0.887	0.85	0.465	0.895	0.861	0.511

Yukarıdaki tabloda da görüldüğü gibi kernel tiplerinde 'RBF' veya 'Linear' olması kullanmış olduğumuz veride C parametresi kullanılan tekniğin ihlallere karşı hassasiyetini belirlemektedir. C parametresi büyüdükçe duyarlılık oranının da kullanılan veriler doğrultusunda arttığı görülmektedir. C parametresi 0.1 olduğunda duyarlılık oranının çok yüksek olduğu ve doğruluk değerinin de en yüksek değerlere sahip olduğu görülmüştür.

Naive Bayes yapay öğrenme tekniğinde C değerinin 0.1 olarak çalıştırılması daha doğru sonuçlar verdiği için bu değer sabit tutularak Kernel bazında RBF ve Linear değişkenlerini farklı test oranları ile uygulanmıştır. Tablo 4.2'de bu çalışmanın skor tablosu gösterilmiştir.

Tablo 4.2: Veri Setinde Test Verisinin Oranının Etkisi

Test Oranı	Kernel					
	RBF			Linear		
	Doğruluk	Kesinlik	Duyarlılık	Doğruluk	Kesinlik	Duyarlılık
0.1	0.907	1	0.5	0.895	0.861	0.511
0.25	0.891	0.872	0.5	0.891	0.872	0.5
0.5	0.894	0.874	0.52	0.894	0.874	0.52

Yukarıdaki Tablo 4.2'de gösterilen sonuçlarda test oranı her iki Kernel tipinde de kullanılan verinin eğitim oranı daha yüksek, test oranı daha düşük olduğunda daha iyi sonuçlar elde edildiği gözlemlenmiştir. Veri setinin test edilen kısmının tüm veri setine oranı 0.1 olduğunda doğruluk oranı en yüksek sonuca ulaşmıştır. Bu kullanılan test oranında elde edilen anomalilerde bir doğrusal ayrım olmadığından Kernel tipinin RBF olması daha doğru sonuçlar elde edilmesini sağlamıştır.

Tablo 4.3: Naive Bayes Algoritmaları Test Oranı ile Karşılaştırması

Type	Test Oranı								
	0.1			0.25			0.5		
	Doğruluk	Kesinlik	Duyarlılık	Doğruluk	Kesinlik	Duyarlılık	Doğruluk	Kesinlik	Duyarlılık
GNB	0.895	0.862	0.511	0.891	0.872	0.5	0.894	0.875	0.522
MNB	0.895	0.862	0.512	0.891	0.872	0.5	0.894	0.875	0.522
CNB	0.887	1	0.384	0.877	1	0.351	0.846	1	0.191

Tablo 4.3'de Gaussian, Multinomial ve Categorical Naive Bayes tekniklerinin test oranı ile beraber karşılaştırılmaları verilmiştir. Naive Bayes algoritmasında eğer özellikler sayılabilir değerler bulundurmakta ise MNB tekniği daha iyi sonuçlar vermektedir. CNB tekniğinde ise özelliklerin kategorilere ayrılmış olması gerekmektedir. Kullanılan veri setinde özelliklerin nümerik değerler içermesi nedeniyle tabloda da görüldüğü üzere MNB tekniğinin CNB tekniğine karşı doğruluk oranında üstün olduğu görülmektedir. MNB tekniğinde kullanılan test veri setinin tüm

veri setine oranı 0.1 olduğunda Doğruluk ve Kesinlik sonuçlarında daha iyi sonuç elde edildiği görülmüştür.

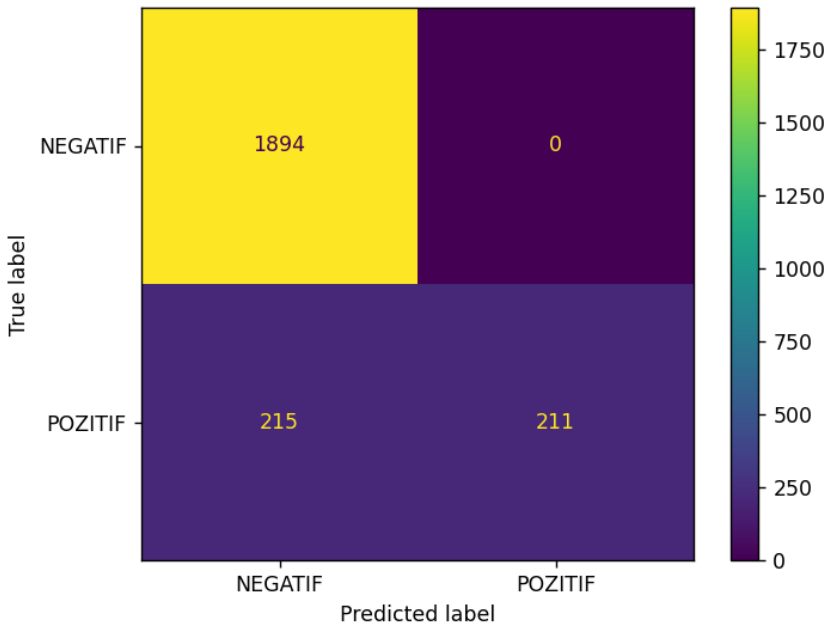
İkili Karar Ağacı yapay öğrenme tekniği test oranları bazında çalıştırılması sonucunda skor tablosu Tablo 4.4'de gösterilmiştir.

Tablo 4.4: İkili Karar Ağacı Test Oranı Tablosu

Test Oranı	Test Oranı		
	Doğruluk	Kesinlik	Duyarlılık
0.1	0.9983	1	0.9906
0.25	0.9978	1	0.9881
0.5	0.9983	1	0.9901

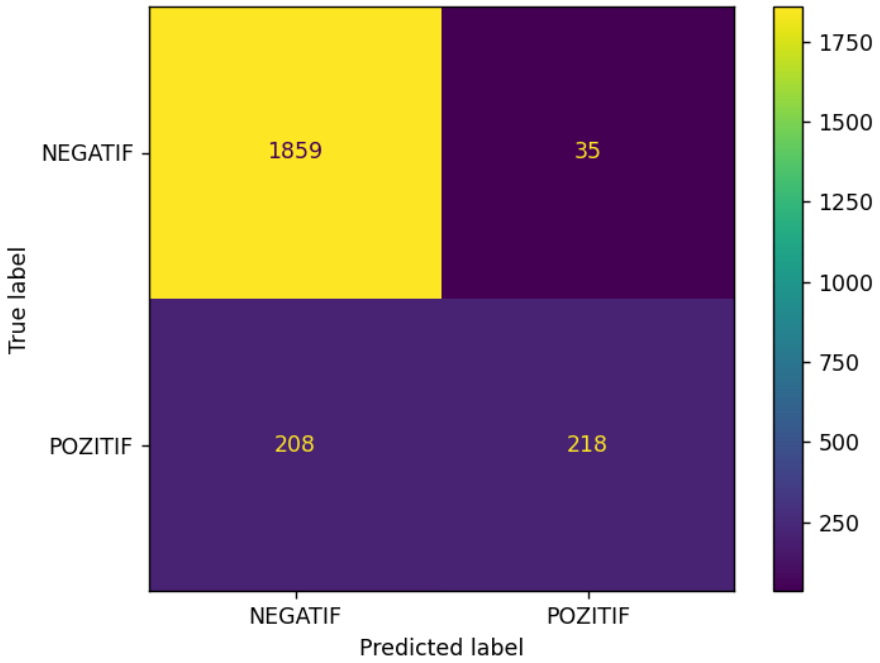
Yukarıdaki Tablo 4.4'te görüldüğü gibi İkili karar ağacı tekniğinde kullanılan test veri setinin tüm veri setine oranı 0.1 olduğunda daha iyi sonuç verdiği gözlemlenmiştir.

Tablo 4.1'de görüldüğü üzere C değeri 0.1 olduğunda en iyi sonucu vermiştir. Bu nedenle de test oranları kernel tipi üzerinden çalıştırılırken C değeri 0.1 sabiti kullanılarak hesaplanmıştır. Tablo 4.2'te görüldüğü gibi test oranı azaltılıp veri setinin eğitilme oranı artırılınca doğruluk oranı artacaktır. Ancak görüldüğü gibi yanlış pozitif hiç tespit edememiş yanlış değerleri tamamen hatalı tespit etmiştir. Burada görülen test oranı 0.1 ve kernel tipi de RBF olduğunda en iyi sonucu verdiği görülmüştür. Karmaşıklık matrisi olarak tüm hesaplamaların sonuçları yerine en iyi modelin; test oranı 0.1 ve C değeri 0.1 ve Kernel tipi RBF yapılan modelin veri seti üzerindeki sonucu Şekil 4.2'de gösterilmiştir.



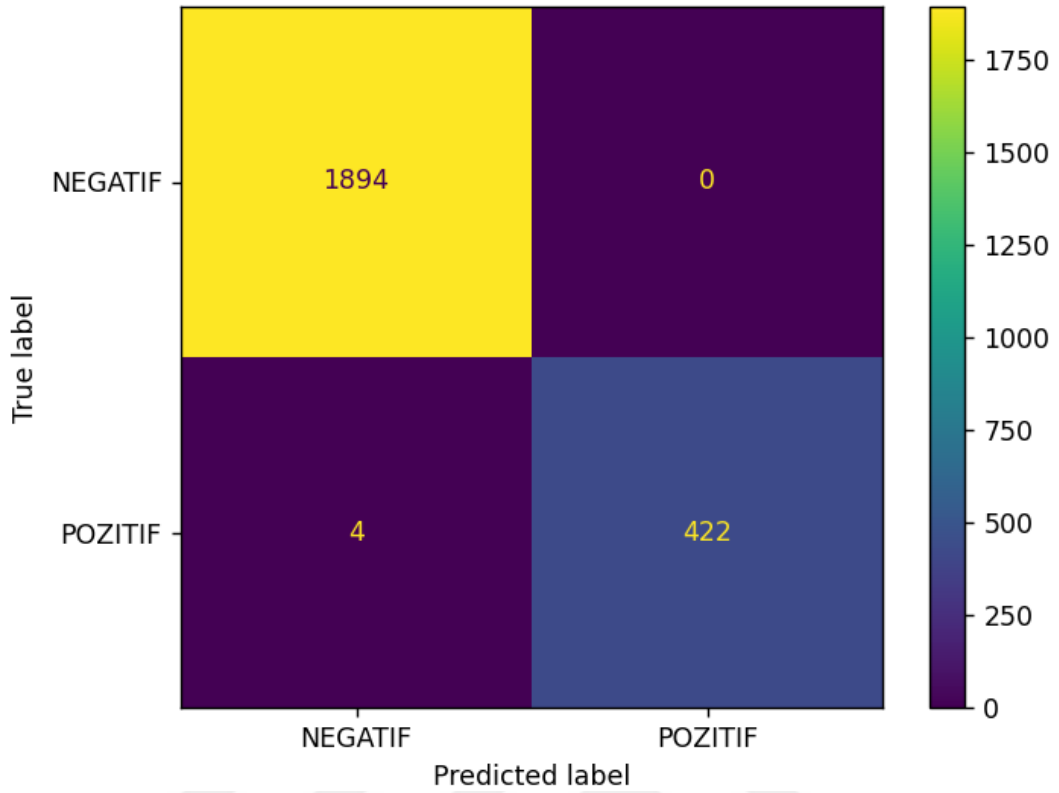
Şekil 4.2: SVM Karmaşıklık Matrisi

Tablo 4.3'te gösterilen Naive Bayes yapay öğrenme teknikleri ve test oranı değerlendirmesinde görüldüğü üzere en iyi sonuçları Multinomial Naive Bayes tekniği ve test oranı 0.1 olan model vermiştir. Bu modelin de karmaşıklık matrisi olarak en iyi modelin sonucu Şekil 4.3'te gösterilmiştir.



Şekil 4.3: Multinomial Naive Bayes Karmaşıklık Matrisi

Tablo 4.4'te görüldüğü gibi İkili Karar Ağacı tekniği 0.1 oranında test veri seti kullanılınca daha doğru sonuçlar elde edilmiştir. Burada elde edilen karmaşıklık matrisi de aşağıdaki Şekil 4.4'te gösterilmiştir.



Şekil 4.4 : İkili Karar Ağacı Karmaşıklık Matrisi

Yukarıda Support Vector Machine, Naive Bayes ve Karar ağacı yapay öğrenme tekniklerinin değerlendirmeleri bulunmaktadır. Tüm değerlendirmelerdeki test oranı ortak olarak 0.1 en iyi sonucu vermiştir. Karşılaştırma, test oranını 0.1 olarak kullanarak yapılmıştır. Bu bölümde karşılaştırma yapılırken en iyi sonuçları veren modellerin değerlendirmesinde; doğruluk, kesinlik, duyarlılık ve F-Ölçütü kullanılmıştır.

Kıyaslaması yapılan yapay öğrenme tekniklerinin karşılaştırma sonuçları Tablo 4.5'te verilmiştir.

SVM				Naive Bayes				Karar Ağacı			
Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü	Doğruluk	Kesinlik	Duyarlılık	F-Ölçütü
0.907	1	0.5	0.662	0.895	0.862	0.512	0.642	0.998	1	0.996	0.995

Tablo 4.5: Yapay Öğrenme Tekniklerinin Sonuçlarının Karşılaştırması

Tablo 4.5'te görüldüğü gibi kullanılan veri seti üzerinde test oranı 0.1 olduğunda İkili karar ağacı yapay öğrenme tekniğinin tüm skorlarda üstünlük sağladığı görülmüştür. En kötü sonuç ise Naive Bayes yapay öğrenme tekniğinde elde edilmiştir. Yapay

öğrenme tekniklerini en iyi sonuçtan en kötü sonuca doğru sıralayacak olursak; İkili Karar Ağacı, SVM ardından Naive Bayes olarak sıralanır. Naive Bayes daha çok özellik barındıran veri setlerinde daha iyi tahmin yeteneğine sahiptir. Bu çalışmada SSR kodu ve uçuş seviyesi değerleri bozulmuştur. Uçuş seviyesi değeri her uçakta farklı değerlerde olduğu için buraya yapılan aynı seviyede bozulma dahil SVM ve Naive Bayes için uygun olmamasına sebep olmuştur.

4.1 Çapraz Doğrulama Tekniği

Elde edilen sonuçları doğrulamak için farklı test oranlarına ek olarak çapraz doğrulama (cross validation) tekniği uygulanmıştır. Çapraz doğrulama tekniği; algoritmaların performansını değerlendirmek ve modelin genelleme yeteneğini ölçmek için kullanılan bir tekniktir. Çapraz doğrulama tekniğinde veri iki ana bölümden oluşur: bunlar eğitim seti ve test seti. Veri kümesi, k alt kümeye bölünür. Her bir alt küme, sırayla test seti olarak kullanılır geri kalan kümeler eğitim seti olarak birleştirilir. Her bir alt kümenin sonuçları ele alınır.

Kullanmış olduğumuz modelde çapraz doğrulama tekniği olarak 5 katmanlı çapraz doğrulama tekniği kullanılmıştır. Kullanılan bu teknik ile birlikte modelin genelleme yeteneği daha iyi ölçülecektir.

Kullanmış olduğumuz üç ayrı yapay öğrenme tekniğinin her birinin en iyi sonuçları veren metrikleri üzerinden 5-Katmanlı çapraz doğrulama teknikleri uygulanmıştır. Bu uygulanan doğrulama tekniği ile beraber her bir ayrılan kümenin doğrulama oranları Tablo 4.6'da gösterilmiştir.

	Katman 1	Katman 2	Katman 3	Katman 4	Katman 5
İkili Karar Ağacı	0.99849105	0.99784436	0.99849105	0.99913775	0.99870634
RBF Support Vector Machine	0.80987282	0.81008838	0.81008838	0.81008838	0.81004743
Multinomial Naive Bayes	0.89523604	0.90019401	0.90472085	0.89911619	0.89241052

Tablo 4.6: Yapay Öğrenme Tekniklerinin 5-Katmanlı Çapraz Doğrulama Tekniği ile Karşılaştırması

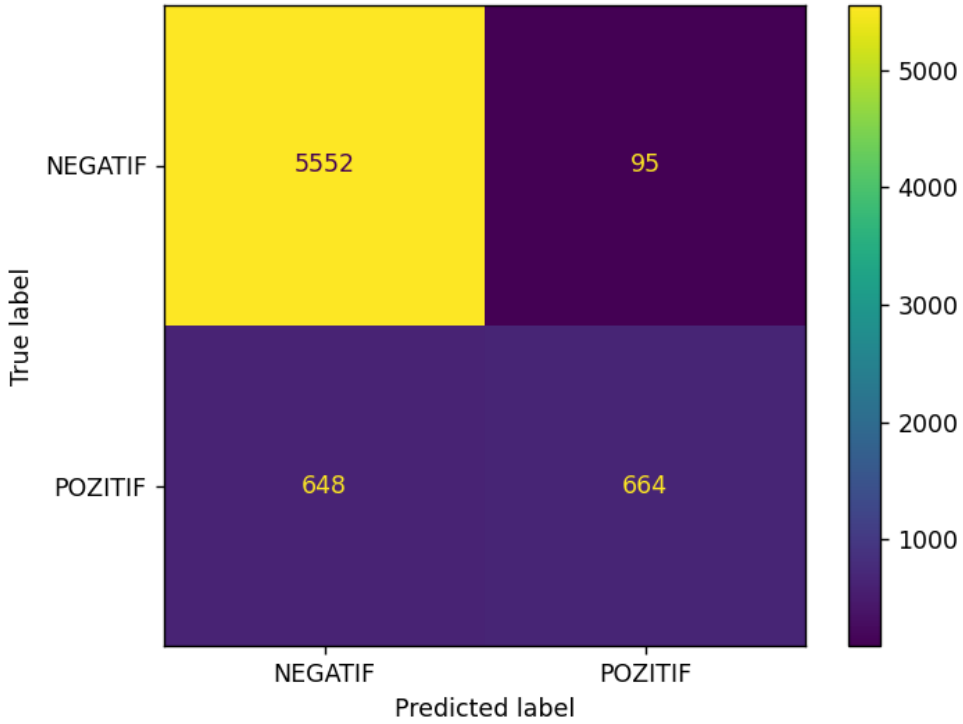
Yukarıdaki tabloda görülen sonuçlarda İkili Karar Ağacı ve Multinomial Naive Bayes tekniklerinin; doğrulama teknikleri ile 5 katmanlı çapraz doğrulama tekniklerinde her katman için alınan doğruluk skorlarının yakın olması, bu modellerin genel olarak tutarlı ve güvenilir bir performansa sahip olduğunu göstermektedir. Bu, modelin farklı eğitim ve test veri kümeleri üzerinde benzer performans sergilediği anlamına gelir. Tutarlı sonuçlar elde etmek, modelin overfitting (aşırı uyum) veya underfitting (yetersiz uyum) gibi sorunlardan kaçındığını gösterir ve daha iyi genelleme yapma olasılığını artırır. Eğer çapraz doğrulama katmanları arasında sonuçlar arasında büyük

farklılıklar olsaydı, bu durum modelin verilere aşırı uyum sağlayabileceğini veya farklı veri kümelerine uyarlanamayacağını gösterecekti. RBF Support Vector Machine yapay öğrenme tekniğinde ise elde edilen doğruluk oranı ile 5 katmanlı çapraz öğrenme tekniğinde elde edilen sonuçlar arasında farklar oluştuğu gözlemlenmiştir. RBF Support Vector Machine modelinin bu veri seti için diğer yapay öğrenme teknikleri arasında daha az genellebilir ve daha az güvenilir modeli olduğu görülmektedir.

4.2 Literatürdeki Çalışma ile Karşılaştırılması

Elde edilen sonuçları literatürdeki bir çalışma ile karşılaştırmak için Wahlgren ve Thorn çalışması ele alınmıştır. Wahlgren ve Thorn çalışmasında ADS-B verisinden gelen verilere saldırı simülasyonu üzerinden bir yazılımı kullanarak veri setini elde etmiştir (Wahlgren & Thorn,2021). Araştırmacıların ADS-B verisine saldırı simülasyonunda elde ettiği veri setinde False Heading, False Information, False Squawk, Jumping, Authenticaiton saldırılarının doğru yanlış etiketleri bulunmaktadır. Yapmış oldukları modelde %30 oranında test verisi, %70 oranında eğitim verisi olarak veriyi ayırmışlardır. Kullandıkları SVM yapay öğrenme tekniğini de Python'da gerçekleştirmişlerdir. Çalışma sonucunda tüm etiketlerin doğru tahminleme ortalaması, SVM kullanarak %91.4 doğru tahminleme yapmışlardır. Diğer taraftan dikkat çeken kötü sonuç ise False Heading etiketinde %61 gibi düşük doğrulukta bir tahmin gerçekleşmiştir. Tez kapsamında önerilen sistem modelinden elde ettiğimiz sonuçta ise İkili Karar Ağacı yapay öğrenme tekniği %10 test oranı %90 eğitim oranı ile en iyi sonucu vermiş olup doğruluk bazında %99.8 başarı elde etmiştir.

Tez çalışmasında, Wahlgren ve Thorn'un yaptığı çalışma ile kıyaslama yapmak maksadıyla aynı eğitim ve test oranı olan %30 test verisi, %70 eğitim verisi İkili Karar Ağacı yapay öğrenme tekniğinde kullanılarak aşağıdaki Şekil 4.5 Karmaşıklık Matrisi elde edilerek doğruluk skoru %89 olarak hesaplanmıştır.



Şekil 4.5: İkili Karar Ağacı 0.3 Test Oranlı Karmaşıklık Matrisi

Wahlgren ve Thorn'un çalışması ile bu tezde önerilen sistem modelin karşılaştırması aşağıdaki Tablo 4.7'da verilmiştir.

Tablo 4.7 : Wahlgren & Thorn ile Tez Çalışmasının Kıyaslanması

Karşılaştırma Tablosu	**Wahlgren & Thorn**	**Tez Çalışması**
Veri Setini Elde Etme Yöntemi	Hava Trafik Saldırı Simülasyon Yazılımı	Hava Trafik Simülasyon Yazılımından Gelen veri ADS-B verisine Dönüştürme ve Sonrasında Veriyi Bozma
Veri Ön İşlemleri	• Gereksiz Kolonların Çıkarılması • Eksik Verilerin İşlenmesi	• Gereksiz Kolonların Çıkarılması • Eksik Verilerin İşlenmesi • Veri Setinin Standartize Edilmesi

	• Veri Setinin Standartize Edilmesi	
Toplam İşlenen Mesaj Sayısı	Yaklaşık 16000	23194
Kullanılan Modellerin Eğitim Ve Test Oranları	Test Oranı %30 Eğitim Oranı %70	En iyi Sonucu Veren Model: Test Oranı %10 Eğitim Oranı %90 Diğer Kullanılan Oranlar: Test Oranı %25, %30 ve %50 olarak da çalışmalar yapılmıştır.
Kullanılan Yapay Öğrenme Teknikleri	Support Vector Machine Tekniği	En iyi Sonucu Veren Model: İkili Karar Ağacı Tekniği Diğer Kullanılan Modeller: • Support Vector Machine (RBF ve Linear) • Naive Bayes (MNB, GNB ve CNB)
Saldırı Türleri	• False Squawk • False Information • False Heading • Jumping • Authentication	• False Squawk (SSR Code) • False Flight Level (Sahte Uçuş Seviyesi Saldırısı)
Etiket Sayısı	Her bir saldırı için ayrı etiket: 5	Tüm saldırılar için bir etiket: 1
Doğruluk oranları Tahmini	Tüm etiketlerin Ortalaması **%91.4**	En iyi alınan sonuç: **%99.8** Aynı test oranı ile en iyi sonuç alınan model kullanıldığında alınan sonuç: **%89.2**

Tablo 4.7'da görüldüğü üzere çalışmalarda kullanılan veri setleri benzer yöntemlerle elde edilmiştir. Yapılan saldırılar sonucunda elde edilen veriler ve etiketler farklılık göstermektedir. Saldırı sonucu oluşan etiketlemelerin ve kullanılan yapay öğrenme tekniğinin de doğruluk oranlarına ve seçimine etki ettiği tespit edilmiştir.

Çoklu etiketi bulunan veri setlerinde öğrenme tekniği olarak SVM kullanıldığında daha iyi sonuçlar elde edildiği gözlemlenmiştir. Ancak bu bazı etiketlerde (ham veride boş değerleri çok olan) çok düşük doğruluk skorlarına da sebep olabilmektedir. False Information tahminindeki %61'lik düşük oranda doğruluk skoru vermesi gibi. Bu tez çalışmasında birden fazla saldırı yapılmış olsa da ortak bir şekilde değerlendirilerek tek kolonda bozulup bozulmadığı etiketlenmiştir. Bu nedenle tez çalışmasında İkili Karar Ağacı yapay öğrenme tekniği ile önerilmiş sistem modeli, ADS-B cihazına yapılan sahte verisi saldırılarında daha iyi sonuçlar verdiği tespit edilmiştir.

5. SONUÇLAR

Havacılığın hızla büyümesiyle, yoğunlaşan hava trafiğinin yönetimi de çok büyük önem kazanmaktadır. Hava trafiğinin yönetiminde kullanılan yardımcı yazılımlar hava taşıtının konumunu kontrolörlere yayınlamaktadır. Hava taşıtlarının konumları ile birlikte kontrolörler hava trafiğini yönetebilmektedir. Hava taşıtının konum bilgisi radar ve ADS-B cihazları üzerinden gelen veriler ile kontrolörün ekranına yansıtılmaktadır.

ADS-B cihazları radarlara oranla finansal maliyeti daha ucuz ve kurulum maliyeti de daha düşüktür. Bu nedenle ADS-B cihazının kendisi veya ADS-B cihazı üzerinden hava trafiği dışarıdan gelebilecek saldırılara diğer radarlara göre daha açıktır. Bu saldırılar hava trafiğini önemli ölçüde etkileyebilecek, kontrolörün yanlış davranmasına hatta uçak kaza kırımlarına sebep olabilecek saldırılardır. Bu tez kapsamında ADS-B üzerinden hava trafiğini yöneten kontrolöre gönderilen sahte veri saldırıları ele alınıp anomali tespiti gerçekleştirilmiştir.

Bu çalışmada ADS-B cihazlarına karşı yapılan saldırıların çözümlerinde yapay öğrenme teknikleri kullanılarak çözüm modeli sunulmuştur. Öncelikle modelde kullanılmak üzere Hava taşıtı simülasyon yazılımından elde edilen veriye sahte veri saldırıları yapılmıştır. Bu sahte verileri saldırıları ADS-B verisinde bulunan SSR code ve Uçuş Seviyesi alanlarına yapılmıştır. Bu sayede hem orjinal hem de saldırılmış veri setleri elde edilmiştir. Tez çalışmasında kendine özgü bir veri seti kullanılarak önemli bir veri seti kazandırılmıştır. Elde edilen bu veride gereksiz kolonların silinmesi, boş alanların doldurulması, standartizasyon gibi bazı ön işlemler uygulanarak yapay öğrenme tekniklerinin kullanımına hazır hale getirilmiştir. SVM, Naive Bayes ve Karar Ağacı yapay öğrenme teknikleri uygulanmıştır. Kullanılan yapay öğrenme teknikleri farklı test oranları üzerinden modellenerek en uygun test oranlarının elde edilmesi sağlanmıştır. Yapay öğrenme tekniklerinin kendi içerisinde farklı test oranları ve değişkenleri ile elde edilen sonuçların performans değerlendirmeleri tartışılmıştır.

Performans değerlendirmesi kapsamında yapay öğrenme tekniklerinden elde edilen modellerin sonuçlarını değerlendirmek üzere Doğruluk, Kesinlik, Duyarlılık ve F-Ölçütü skorları kullanılmıştır. Çalışma kapsamında Support Vector Machine, Naive Bayes ve İkili Karar Ağacı yapay öğrenme teknikleri, %10, %25, %50 test verisi oranları verilerek kullanılmıştır. Bu çalışmada en iyi %10 test verisi oranı ile İkili Karar Ağacı yapay öğrenme tekniğiyle %99,8 oranında başarı elde edilmiştir.

Kullanılan yapay öğrenme tekniklerinin modellerinin tutarlı ve güvenilebilir olmasını ölçmek ve karşılaştırmak için 5 katmanlı çapraz doğrulama tekniği uygulanmıştır. Bu doğrulama tekniği sonrasında RBF Support Vector Machine modelinin kullanılan veri seti için daha az tutarlı olduğu, İkili Karar Ağacı ve Multinomial Naive Bayes yapay öğrenme tekniklerinin daha tutarlı ve güvenilir modeller olduğu elde edilmiştir.

KAYNAKÇA

Ahmed, M., & Pathan, A.-S. K. (2020). False data injection attack (FDIA): An overview and new metrics for fair evaluation of its countermeasure. *Complex Adaptive Systems Modeling*, *8*(1), 4. https://doi.org/10.1186/s40294-020-00070-w

Aksu, G., & Dogan, N. (2019). Comparison of Decision Trees Used in Data Mining= Veri madenciliginde kullanilan karar agaçlarinin karsilastirilmasi. *Pegem Journal of Education and Instruction*, *9*(4), 1183–1208.

Berwick, R. (n.d.). *An Idiot's guide to Support vector machines (SVMs)*.

Costin, A., & Francillon, A. (2012). Ghost in the Air (Traffic): On insecurity of ADS-B protocol and practical attacks on ADS-B devices. *Black Hat USA*, *1*, 1–12.

Damis, H. A., Shehada, D., Fachkha, C., Gawanmeh, A., & Al-Karaki, J. N. (2020). A microservices architecture for ADS-B data security using blockchain. *2020 3rd International Conference on Signal Processing and Information Security (ICSPIS)*, 1–4.

Deshmukh, D. H., Ghorpade, T., & Padiya, P. (2015). Improving classification using preprocessing and machine learning algorithms on NSL-KDD dataset. *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, 1–6.

El Marady, A. A. W. (2017). Enhancing accuracy and security of ADS-B via MLAT assisted-flight information system. *2017 12th International Conference on Computer Engineering and Systems (ICCES)*, 182–187.

Gerardo W. Flintsch, A. A. W. (2017). Integrated Modeling of Air Traffic, Aviation Weather, and Communication Systems. Blacksburg, Virginia.

International Civil Aviation Organisation.(2018). *ADS-B Opeational Manuel*. Asia-Pacific Office.

Kacem, T., Kaya, A., Keceli, A. S., Catal, C., Wijsekera, D., & Costa, P. (2021). ADS-B Attack Classification using Machine Learning Techniques. *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 7–12.

Khan, S., Thorn, J., Wahlgren, A., & Gurtov, A. (2021). Intrusion detection in automatic dependent surveillance-broadcast (ADS-B) with machine learning. *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, 1–10.

Khandker, S., Turtiainen, H., Costin, A., & Hämäläinen, T. (2022). On the (In) Security of 1090ES and UAT978 Mobile Cockpit Information Systems–An Attacker Perspective on the Availability of ADS-B Safety-and Mission-Critical Systems. *IEEE Access*, *10*, 37718–37730.

Kocaağa, E. (2017). *Otomatik bağımlı gözetim-yayını güvenlik analizi* [PhD Thesis]. Bilişim Enstitüsü.

Li, N., Lin, L., & Li, F. (2021). ADS-B Anomaly Data Detection Using SVDD-based LSTM Encoder-Decoder Algorithm. *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, 1295–1300.

Manikanth. (2021, March 19). What is the use of data standardization and where do we use it in machine learning. *Analytics Vidhya*. https://medium.com/analytics-vidhya/what-is-the-use-of-data-standardization-and-where-do-we-use-it-in-machine-learning-97b71a294e24, [Erişim Tarihi: 23.03.2023]

Naive Bayes Classifier Tutorial: With Python Scikit-learn. (n.d.). Retrieved April 2, 2023, from https://www.datacamp.com/tutorial/naive-bayes-scikit-learn, [Erişim Tarihi: 31.03.2023]

Naive Bayes Classifiers. (2017, March 3). *GeeksforGeeks*. https://www.geeksforgeeks.org/naive-bayes-classifiers/ , [Erişim Tarihi: 30.03.2023]

Nguyen, T. T., & Armitage, G. (2008). A survey of techniques for internet traffic classification using machine learning. *IEEE Communications Surveys & Tutorials*, *10*(4), 56–76.

Pinustech. (n.d.). *Yapay Öğrenme (Machine Learning) Tam Olarak Nedir?* Retrieved April 2, 2023, from https://www.smartmind.com.tr/yapay-ogrenme-machine-learning-tam-olarak-nedir-i-927 [Erişim Tarihi: 28.03.2023]

Shang, F., Wang, B., Li, T., Tian, J., Cao, K., & Guo, R. (2020). Adversarial examples on deep-learning-based ADS-B spoofing detection. *IEEE Wireless Communications Letters*, *9*(10), 1734–1737.

SHGM. (n.d.). *Mode-S Tahsis İşlemleri | Sivil Havacılık Genel Müdürlüğü*. Retrieved April 2, 2023, from http://172.16.10.52:81/tr/hava-araci-islemleri/2233-mode-s-tahsis-islemleri [Erişim Tarihi: 27.03.2023]

Shoba R., Kenta N., Christian S., Micheal G., Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics (Amsterdam: ELSEVIER Yayınları, 2019), 405, books.google.com.tr [Erişim Tarihi: 29.03.2023]

Support Vector Machine Explained-Theory, Implementation, and Visualization. (n.d.). Retrieved April 2, 2023, from https://www.linkedin.com/pulse/support-vector-machine-explained-theoryvisualization-zixuan-zhang, [Erişim Tarihi : 29.03.2023]

TajDini, M., Sokolov, V., & Skladannyi, P. (2021). Performing Sniffing and Spoofing Attack Against ADS-B and Mode S using Software Define Radio. *2021 IEEE International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*, 1–5.

Taylor, H., Blount-Hill, B.-H., & Jennings, C. R. (2016). *Unmanned Arial Systems (UAS) in the Local Public Safety Environment: A Review*.

Wahlgren, A., & Thorn, J. (2021). *Detecting ADS-B spoofing attacks: Using collected and simulated data*.

WestJet Says It Never Sent Hijack Alarm, Wasn't in Danger. (2015, January 11). *Bloomberg.Com*. https://www.bloomberg.com/news/articles/2015-01-10/westjet-hijack-signal-called-false-alarm, [Erişim Tarihi: 02.04.2023]

Yiğidim, H. A. (2012). *Makine Öğrenme Algoritmalarını Kullanarak Ağ Trafiğinin Sınıflandırılması* [Master's Thesis]. TOBB Ekonomi ve Teknoloji Üniversitesi Fen Bilimleri Enstitüsü.

Ying, X., Mazer, J., Bernieri, G., Conti, M., Bushnell, L., & Poovendran, R. (2019). Detecting ADS-B spoofing attacks using deep neural networks. *2019 IEEE Conference on Communications and Network Security (CNS)*, 187–195.

EKLER

EK-1: Ön İşlem Kodu
EK-2: SVM, Naive Bayes ve Decision Tree yapay öğrenme Uygulama Kodu
EK-3: Yapay Öğrenme Teknikleri Skor ve Değerlendirme Kodu
EK-4: ATCSIMTEST Yazılımı Veriyi Dışarı Çıkarma Kodu
EK-5: ATCSIMTEST Saldırı kodları

EK-1: *Saldırı Kodları*

```
private void spoofer(int attackMode, int attackLevel, AsterixOutDoor cat021) {
        switch (attackMode){
            case 1:
                spoofingFL(cat021,attackLevel);
            break;
            case 2:
                spoofingSSRCode(cat021);
                break;
            case 3:
                spoofingFL(cat021,attackLevel);
                spoofingSSRCode(cat021);
            default:
                break;
        }
    }

    private void spoofingSSRCode(AsterixOutDoor cat021) {
        int rand_num = rand.nextInt(10);
        if (cat021 != null){
        switch (rand_num){
            case 2:
                cat021.setMode3AReply_Mode3ACode(7600);
                isSpoofed = true;
                break;
            default:
                break;

            }
        }
    }

    private void spoofingFL(AsterixOutDoor cat021, int i) {
        int rand_num = rand.nextInt(10);
        if (rand_num == 1) {
            if (cat021 != null) {
                cat021.setFlightLevel(cat021.getFlightLevel(true) + (100 * i));
                isSpoofed = true;
            }
        }
    }
```

EK-2: ATCSIMTEST Yazılımı Veriyi Dışarı Çıkarma Kodu

```
// Konfigürasyon Değişkenlerinin tanımlandığı satırlar
private int attack_mode;
private int attack_level;

//Konfigürasyon Değişkenlerine konfigürasyon dosyasındaki değişkenlerin atandığı satırlar
attack_mode = Integer.parseInt(prop.getProperty("attack_Mode"));
attack_level = Integer.parseInt(prop.getProperty("attack_Level"));

//Dosyaların oluşturulacağı adresin oluşturulduğu satırlar
public String excelFileSpoofed =
exportPath.resolve("export_spoofed_"+LocalDateTime.now()+".xlsx").toFile().getPath();
public String excelFile =
exportPath.resolve("export_"+LocalDateTime.now()+".xlsx").toFile().getPath();
//Verilerin dışarı çıkartıldığı dosyanın oluşturulması metodu
private void exportData(Path exportPath) {
        List<String> headers = new ArrayList<>();
        headers.add("TimeStamp");
        headers.add("CallSign");
        headers.add("Mode-S");
        headers.add("SSR Code");
        headers.add("latitude");
        headers.add("longitude");
        headers.add("altitude");
        headers.add("speed");


        exportPath.toFile().mkdirs();
        Workbook workbook = new HSSFWorkbook();
        Sheet spreadSheet = workbook.createSheet("ADS-B Data");
        Workbook workbookSpoofed = new HSSFWorkbook();
        Sheet spreadSheetSpoofed = workbookSpoofed.createSheet("ADS-B Data Spoofed");
        Row row;
        Row rowSpoofed;
        int rowId = 0;
        row = spreadSheet.createRow(1);
        AtomicInteger cellId = new AtomicInteger();
        headers.forEach(header -> {
            Cell cell = row.createCell(cellId.getAndIncrement());
            cell.setCellValue(header);
        });

        rowSpoofed = spreadSheetSpoofed.createRow(1);
        AtomicInteger cellIdSpoofed = new AtomicInteger();
        headers.forEach(header -> {
            Cell cell = rowSpoofed.createCell(cellIdSpoofed.getAndIncrement());
            cell.setCellValue(header);
        });

        try {

            FileOutputStream out = new FileOutputStream(new File(excelFile));
            workbook.write(out);
            out.close();
        }catch (IOException e){
            e.printStackTrace();
        }
```

EK-2- Devamı.

```
try {

            FileOutputStream out = new FileOutputStream(new File(excelFileSpoofed));
            workbookSpoofed.write(out);
            out.close();
        }catch (IOException e){
            e.printStackTrace();
        }

    }

//Dışarı çıkarılan dosyaya verilerin bozulması ve bozulmuş halde gönderilmesi
/**
     * Cat 21 verisi ADS-B verisini ve Simülasyon verisi olan pAtcTrack ile birlikte
bozulmuş veri elde edilecektir.
     * @param cat021
     * @param pAtcTrack
     * @param attackMode trafiğin bozulması için gerekli olan modlar
     * @param attackLevel trafiğin uçuş seviyesinin bozulma oranı.
     * @throws FileNotFoundException
     */
    public void exportData(AsterixOutDoor cat021, HlaATCTrack pAtcTrack, int attackMode, int
attackLevel) throws FileNotFoundException {
        String  existingFilePath = TInitManager.getInstance().excelFile;// Orjinal ADS-B
verisinin dosyasının gönderileceği dosya yolu
        String  existingSpoofedFilePath =
TInitManager.getInstance().excelFileSpoofed;//Bozulmuş ADS-B verisinin dosyasının
gönderileceği dosya yolu
        File xlsxFile = new File(existingFilePath);
        File xlsxFileSpoofed = new File(existingSpoofedFilePath);

        Map<String, Object> ADS-BDatasSpoofed = new HashMap<>();
        Map<String, Object> ADS-BDatas = new HashMap<>();

                //Normal Veriyi bir map'e aktarıyor.
                ADS-BDatas.put(String.valueOf(rowId),new
Object[]{LocalDateTime.now().toString(),
pAtcTrack.getCallsign(),pAtcTrack.getHlaATCTrackAttributes().getModeS(),
                        cat021.getMode3Areply(),
pAtcTrack.getSpatial().getDeadReckoningAlgorithmAAlternatives().getSpatialFVW().getWorldLoca
tion().y,

pAtcTrack.getSpatial().getDeadReckoningAlgorithmAAlternatives().getSpatialFVW().getWorldLoca
tion().x, cat021.getFlightLevel(true),
                        pAtcTrack.getHlaATCTrackAttributes().getGroundSpeed()});

        //spoofer for data according to attack mode and level
        //konfigürasyon değelerine göre ADS-B verisini bozacak olan metoda yönlendiriyor.
        spoofer(attackMode, attackLevel, cat021);

                //Spoofed Data
                //Bozulmuş veriyi map'e aktarıyor.
                ADS-BDatasSpoofed.put(String.valueOf(rowId),new
Object[]{LocalDateTime.now().toString(),
pAtcTrack.getCallsign(),pAtcTrack.getHlaATCTrackAttributes().getModeS(),

cat021.getMode3Areply(),pAtcTrack.getSpatial().getDeadReckoningAlgorithmAAlternatives().getS
patialFVW().getWorldLocation().y,
```

EK-2- Devamı.

```
pAtcTrack.getSpatial().getDeadReckoningAlgorithmAAlternatives().getSpatialFVW().getWorldLoca
tion().x, cat021.getFlightLevel(true),
                        pAtcTrack.getHlaATCTrackAttributes().getGroundSpeed(), isSpoofed });

        isSpoofed = false;

        /*
        Bu metodun geri kalan kısmında dosyayı yazdırma işlemi bulunmaktadır.
         */
        Object[] objects = (Object[]) ADS-BDatas.get(String.valueOf(rowId));
        Object[] objectsSpoofed = (Object[]) ADS-BDatasSpoofed.get(String.valueOf(rowId));
        rowId++;
        try {
            FileInputStream inputStream = new FileInputStream(xlsxFile);
            Workbook workbook = WorkbookFactory.create(inputStream);
            Sheet sheet = workbook.getSheetAt(0);
            Row row ;
            int cellId = 0;
            row = sheet.createRow(rowId);
            if (ADS-BDatas != null){
                for (Object obj : objects){
                    Cell cell = row.createCell(cellId++);
                    cell.setCellValue(obj.toString());
                }
            }
            inputStream.close();
            FileOutputStream os = new FileOutputStream(xlsxFile);
            workbook.write(os);

        } catch (InvalidFormatException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        if (attackMode > 0) {
            try {
                FileInputStream inputStream = new FileInputStream(xlsxFileSpoofed);
                Workbook workbook = WorkbookFactory.create(inputStream);
                Sheet sheet = workbook.getSheetAt(0);
                Row row;
                int cellId = 0;
                row = sheet.createRow(rowId);
                if (ADS-BDatasSpoofed != null) {
                    for (Object obj : objectsSpoofed) {
                        Cell cell = row.createCell(cellId++);
                        cell.setCellValue(obj.toString());
                    }
                }
                inputStream.close();
                FileOutputStream os = new FileOutputStream(xlsxFileSpoofed);
                workbook.write(os);
            } catch (InvalidFormatException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }

  }
```

EK-3: Ön İşlem Kodu

```
import pandas as pd
import matplotlib.pyplot as plt


from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

df = pd.read_csv("ADS-B_data_w_miss_ch_call_last.csv", header=0)
df.head()
print("Original Data")
df.info()


# remove unnecessary columns
del df["Longtitude"]
del df["latitude"]
del df["TimeStamp"]
del df["Callsign"]
del df["Mode-S"]

#Missing value treat
df["Flight Level"].mean()
df["Flight Level"].fillna(value = df["Flight Level"].mean(), inplace = True)
print("After Remove Unnecessary columns and Missing value treat")
df.info()

#X and Y split
X = df.loc[:, df.columns != ("isSpoofed")]
type(X)
print("X Data")
print(X.head())
print(X.shape)
y = df["isSpoofed"]
type(y)
print("y data")
print(y.head())
print(y.shape)

#Test and Train Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print("X_train data")
print(X_train.head())
print(X_train.shape)
print("X_test data")
print(X_test.head())
print(X_test.shape)
#Standardizing
sc = StandardScaler().fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
print("After Standadizing X_train_std")
print(X_train_std)
print("After Standadizing X_test_std")
print(X_test_std)
```

EK-4 : SVM, Naive Bayes ve Decision Tree yapay öğrenme Uygulama Kodu

```
#Training SVM
from sklearn import svm
clf_svm_1 = svm.SVC(kernel='rbf', C=0.1)
clf_svm_1.fit(X_train_std, y_train)
#predict values using trained model
y_train_pred = clf_svm_1.predict(X_train_std)
y_test_pred = clf_svm_1.predict(X_test_std)
print("y Test Pred")
print(y_test_pred)

#Training Naive Bayes
#Import Gaussian Naive Bayes model
from sklearn.naive_bayes import GaussianNB, MultinomialNB, CategoricalNB

#Create a Gaussian Classifier
gnb = MultinomialNB()

#Train the model using the training sets
gnb.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = gnb.predict(X_test)

#Training Decision Tree
from sklearn import tree
clf = tree.DecisionTreeClassifier()
clf.fit(X_train_std, y_train)
# tree.plot_tree(clf)
y_test_pred = clf.predict(X_test_std)
```

EK-5: Yapay Öğrenme Teknikleri Skor ve Değerlendirme Kodu

```
print("confusion matrix")
cm = confusion_matrix(y_test, y_test_pred, labels=clf.classes_)
print(cm)
print("Accuracy Score tree")
print(accuracy_score(y_test,y_test_pred))
print("Precision Score")
print(precision_score(y_test,y_test_pred, pos_label="POZITIF"))
print("recall")
print(recall_score(y_test, y_test_pred, pos_label="POZITIF"))
print("f1 SCORE")
print(f1_score(y_test, y_test_pred, pos_label="POZITIF"))
print("Confusion Matrix display")

#Display graphics
disp = ConfusionMatrixDisplay(confusion_matrix = cm, display_labels=clf.classes_)
disp.plot()
plt.show()

display = PrecisionRecallDisplay.from_estimator(clf_svm_1, X_test, y_test, name="Ortalama")
display.plot()
plt.show()
```

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : MERAL, İbrahim
Uyruğu : T.C.

Eğitim

Derece	**Üniversite ve Bölüm**	**Mezuniyet tarihi**
Lisans	Ankara Üniversitesi Bilgisayar Mühendisliği	2015

Yabancı Dil

İngilizce