T.C.
YALOVA ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ÖĞRENME SINIFLANDIRICI ALGORİTMALARINDA VERİYE DAYALI KEŞİF SALDIRILARININ TESPİTİ

YÜKSEK LİSANS TEZİ

EMRE SADIKOĞLU
185105005

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi Burcu DEMİRELLİ OKKALIOĞLU

HAZİRAN 2021

T.C.
YALOVA ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

YAPAY ÖĞRENME SINIFLANDIRICI ALGORİTMALARINDA VERİYE DAYALI KEŞİF SALDIRILARININ TESPİTİ

YÜKSEK LİSANS TEZİ
EMRE SADIKOĞLU
185105005

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Tezli Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi Burcu DEMİRELLİ OKKALIOĞLU

HAZİRAN 2021

ÖNSÖZ

Bu tez çalışmasında öncelikle hiçbir zaman desteğini esirgemeyen, tecrübelerinden faydalandığım, her zaman katkıda bulunan, araştırmama ve öğrenmeme öncülük eden başta değerli danışmanım Dr. Öğr. Üyesi Burcu Demirelli Okkalıoğlu'na, değerli hocalarım Prof. Dr. Murat Gök ve Dr. Öğr. Üyesi İrfan Kösesoy'a, her zaman kararlarıma saygı duyan, maddi ve manevi her türlü desteği, motivasyonu veren aileme teşekkürlerimi sunarım.

Mayıs 2021

Emre Sadıkoğlu
Araştırma Görevlisi

İÇİNDEKİLER

KISALTMALAR

DVM **:** Destek Vektör Makineleri
k-EYK **:** k-En Yakın Komşuluğu
KA **:** Karar Ağaçları
STS **:** Saldırı Tespit Sistemi
TM **:** Tersine Mühendislik
UCI **:** University of California Irvine
EAO **:** Efektif Atak Oranı

ÇİZELGE LİSTESİ

ŞEKİL LİSTESİ

Sayfa No

YAPAY ÖĞRENME SINIFLANDIRICI ALGORİTMALARINDA VERİYE DAYALI KEŞİF SALDIRILARININ TESPİTİ

ÖZET

Günümüzde internet siteleri, kullanıcı ihtiyaçlarını karşılamak ve hayatlarını kolaylaştırmak amacıyla oldukça akıllı hale gelmiştir. Öyle ki artık internet sitelerinin arka planında yapay zeka ve makine öğrenmesi işlemleri yapabilen sistemler çalıştırılmaya başlanmıştır. Şu an makine öğrenimi ile sınıflandırma yapabilen internet sunucuları, kullanıcılara hizmet verebilmektedir. Bu sistemler tasarlandıklarında herhangi bir güvenlik endişesi yoktu ve güvenlik konusu düşünülmeden geliştirildiler. Bu nedenle bu sistemler saldırılara açıktır. Fakat günümüzde siber saldırıları göz ardı etmek imkansızdır. Bu çalışmada sınıflandırma sisteminin zaafiyetlerini görebilmek adına sisteme bir saldırgan gibi ataklar düzenleyerek sistem hakkında bilgi keşfi yaparak öncelikle kısıtlı bir miktar veri elde edildi. Daha sonra elde edilen veriyi gradyan iniş metodu ile eğittiğimiz sistemimizde daha geniş bir sahte veri kümesi haline getirerek sisteme saldırmak üzere hazırlandı. Saldırı veri kümesi ile sınıflandırma sistemine ataklar yapılarak sonuçlar elde edildi. Elde edilen sonuçlar gerçek sonuçlar ile karşılaştırıldığında, yapılan saldırılar neticesinde sistemin yanlış tarafa yönelmesini ve sınıflandırma başarımının düşüşe geçmesi sağlandı.

Anahtar Kelimeler: siber güvenlik; saldırı tespit; sahte veri üretimi; yapay zeka temelli sınıflandırıcılar.

DETECTION OF DATA-DRIVEN DISCOVERY ATTACKS ON MACHINE LEARNING CLASSIFIER ALGORITHMS

ABSTRACT

Recently, websites have become very smart in order to meet user needs and make their lives easier. Such that systems that can perform artificial intelligence and machine learning processes have started to be on websites. Currently, web servers that can classify with machine learning can serve users. When these systems were designed, there was no security concern and they were developed without considering the security issue. Therefore, these systems are vulnerable to attack. But it is impossible to ignore cyberattacks today. In this study, in order to reveal the weaknesses of the classification system, we first obtained a limited amount of exploration data about information the system by attacking the system like an adversary. Then, we prepared the obtained data to attack the system by transforming it into a larger data set in our system, which we trained with gradient descent method. The results were obtained by making attacks on the classification system with the attack dataset. When the results obtained were compared with the actual results, it was ensured that the system was directed to wrong side and the classification performance decreased as a result of the attacks.

Keywords: cyber security; attack detection; generating adversarial data; artificial intelligence classifiers.

1. GİRİŞ

Siber güvenlik, siber saldırıların meydana getirdiği risk ve güvenlik açıklarına karşı güvenliğin korunması olarak tanımlanmıştır. Günümüzde siber saldırılar, endüstri 4.0 ile ortaya çıkan nesnelerin internetinden, büyük şirketlere ait sistemlere kadar birçok alanı tehdit etmektedir [1]. Siber saldırıların hisse senedi fiyatı etkisine yönelik araştırmalara bakacak olursak, tespit edilen hedef firmaların saldırıdan sonra %1 ile %5 oranında kayıp yaşadığını gösteriyor. Ortalama bir New York Borsası şirketi için, bu büyüklükteki fiyat düşüşleri, 50 milyon ile 200 milyon dolar arasında hissedar zararına dönüşüyor. Bu firmaların 2003 yılı zarar tahminleri 13 milyar dolardan (sadece solucanlar ve virüsler) 226 milyar dolara kadar değişiyor [2].

Siber saldırılar, kişisel verilerin istenmeyen taraflarca ele geçirilmesi ve gizliliğin ihlal edilmesine de yol açabilmektedir. Bir sisteme saldırı yapıldıktan sonra edinilen farklı bilgiler birbirine entegre edilerek kullanıcı tanımlaması yapılabilir ve kullanıcıların gizliliği ihlal edilir [3]. Yakın geçmişte Facebook üzerindeki kullanıcı verilerinin farklı firmalara verilerek kişisel verilerin ihlal edildiğine şahit olduk. Geçtiğimiz günlerde ise ülkemizde bir tekel haline gelmiş, 19 milyondan fazla üyesi bulunan Yemek Sepeti şirketi siber saldırıya uğradı. Bu saldırı sonucunda kullanıcıların ad, soyad, doğum tarihi, adresleri, telefon numaraları ve maskelenmiş haldeki parola bilgisi saldırganların eline geçti. Kullanıcılar, saldırı sonrasında gelmeye başlayan bahis sitesi reklamlı elektronik postalar almaya başladığı için yakınmaktalar.

Yapay zekâ, insan ve hayvanlar tarafından sergilenen bilinç ve duyguları içeren doğal zekânın aksine makineler tarafından sergilenen zekâdır. Yapay zekâ, makinelere doğal zekâyı taklit ettirerek makinelerin doğal zekâ gibi hareket etmelerini amaçlamaktadır [4]. Makine öğrenmesi ise yapay zekânın alt dallarından biridir [5]. Makine öğrenmesi, daha çok örüntü tanıma, hesaplama, biyomedikal sistemlerde hesaplama gibi veri temelli işlemleri gerçekleştirmeyi amaçlamaktadır [6].

Geliştirildikleri dönemde güvenlik endişesi bulunmayan makine öğrenmesi yöntemleri, günümüzde siber saldırılara maruz kalmaktadır. Bu saldırılar bazen gizliliği tehdit ederken bazen de hayati derecede hasar verebilecek etkidedirler. Örneğin makine öğrenmesi yöntemleri ile cisimleri tanımlayan otonom bir araç düşünelim. Eğer bu araç bir trafik ışığını veya trafik tabelasını yanlış tanımlarsa, maddi veya can kayıplı kazalara yol açılabilir. İnsan gözü için belki birebir aynı gözüken iki

görsel, makine tarafından sınıflandırılırken farklı anlamlara gelebilir [7]. Bu gibi sonuçlara, saldırgan girdiler neden olmaktadır. Saldırgan girdi, makine öğrenmesi modelini manipüle ederek olması gerekenden farklı bir tarafa yönlendirir.

Şekil 1.1'de görüldüğü üzere birbirinden bağımsız ve insan olarak gayet kolay bir şekilde ayırt edebiliyoruz. Fakat bilgisayar, bu şekilleri görsel olarak değil piksel olarak ayırt eder ve tanır. Bu nedenle soldaki şekle saldırgan girdiler eklenip bozulduğunda bilgisayar bunu kasis tabelası olarak sınıflandırabilir. Bunun sonucunda da otonom araç kasise yaklaştığını sanırken yaya geçidine yaklaşmaktadır.

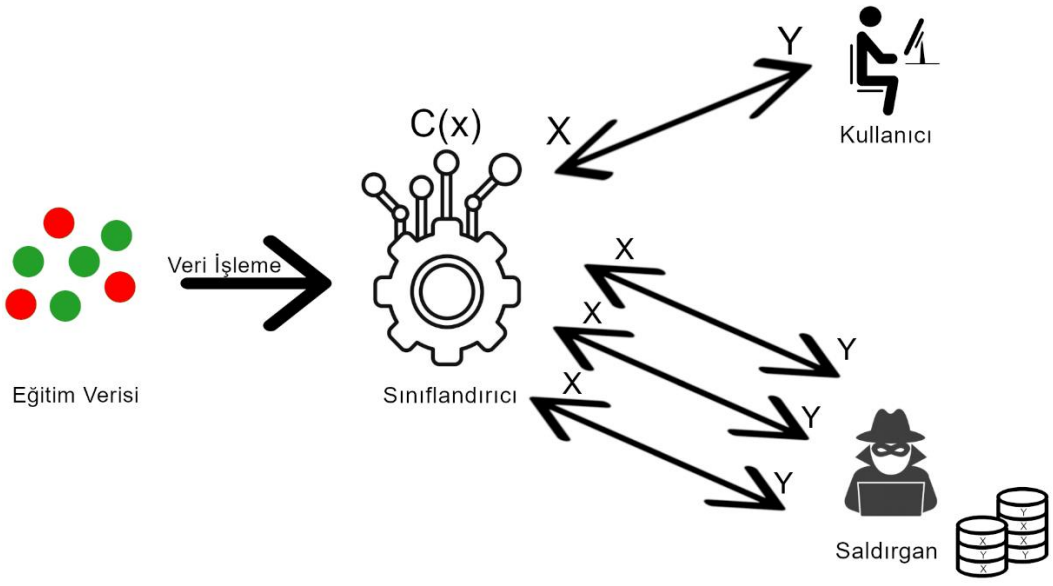
Şekil 1.1 Yaya Geçidi ve Kasis Trafik Levhaları

Makine öğrenmesi modelini bu gibi saldırılara karşı korumak adına çalışmalar yapılmıştır. Bu çalışmalarda Saldırı Tespit Sistemi (STS), spam gibi istenmeyen girdilerin tespiti [8], sistemin genel durumundaki anormal değerlerin tespiti [9], sisteme gönderilen girdilerin sınırlandırılması [11] gibi yöntemler kullanılmıştır. Spam gibi istenmeyen elektronik postaları filtrelemek adına çeşitli sistemler geliştirilmiştir. Bu sistemlerin amacı, kullanıcılara gönderilen elektronik postaları sınıflandırmak ve istenmeyen bu elektronik postaların kullanıcının gelen kutusuna düşmesini engellemektir. Filtreleme için genellikle yazım denetiminden faydalanılmaktadır. Fakat kötü niyetli saldırganlar bu metin tabanlı denetimi de kolaylıkla aşabilmektedirler. Örneğin "BEDAVA KUPON" kelimelerini içeren bir elektronik postanın filtrelendiğini düşünürsek saldırgan kişi buradaki O harfini sıfır rakamı ile değiştirir ve "BEDAVA KUP0N" yazarak bu filtreleme sistemini aşabilir. Şekil 1.2'de yer alan iki görsele bakarsak ilk bakışta hiçbir farkı yok gibi düşünebiliriz. Fakat sağdaki resimde "KUPON" kelimesinde O harfi yerine 0 rakamı yazılmıştır. İlk bakışta insan gözünün ayırt edemediği bu gibi benzer alfa-numerik karakterler

saldırganlar tarafından filtreleme sistemlerini aşmak amacıyla sıklıkla kullanılmaktadır.

Şekil 1.2 Tanıtım E-Posta İçerikleri

Şekil 1.3'te görüleceği üzere normal kullanıcılar, kara-kutu sınıflandırıcıya genelde bir adet sorgu gönderir ve sınıflandırıcıdan dönen değeri alır. Fakat saldırganlar ise kara-kutu sınıflandırıcıya birden çok sorgu gönderir, gönderdiği sorguları ve sınıflandırıcının döndürdüğü değerleri bir yerde saldırı için kullanmak amacıyla depolar, asıl hedefi sistemin işleyiş mekanizmasını çözmektir.



Şekil 1.3 Saldırgan ve Normal Kullanıcının Sorgu Şekilleri

1.1 Tezin Amacı

Bu tez çalışmasının amacı, makine öğrenmesi temelli sınıflandırıcıların saldırılara karşı zafiyetlerini ortaya çıkarılmasıdır. Bu bağlamda, kara kutu makine öğrenmesi modellerinin zafiyetlerinin aşılması amacıyla sahte verilerin üretilmesine dayanan bir

atak yöntemi geliştirilmiştir. Geliştirilen atak yöntemi 6 farklı veri seti üzerinde ve 3 farklı makine öğrenmesi sınıflandırıcısı üzerinde test edilmiştir.

1.2 Literatür Özeti

Laskov ve Kloft [9], makine öğrenmesi sınıflandırıcılara karşı yapılan saldırıları sayısal veriler üzerinden tespit etmek için bir sistem geliştirmişler. Bu sisteme göre öğrenme ve saldırı süreçleri aksiyomatik olarak biçimlendirilir, saldırganın kısıtları belirlenir, optimal saldırı politikası araştırılır ve optimal politika kapsamında saldırganın saldırıdan kazancı sınırlandırılır. Geliştirilen bu sistemde sınıflandırıcıya yeni bir vektör gönderildiğinde ağırlık merkezi yeniden hesaplanır ve ağırlık merkezinin değişimi gözlenir. Eğer ağırlık merkezi değişmiş olursa, bu anormal bir durumu gösterir ve sistemin anomali tespiti yapması ile sonuçlanır.

R.P. Lippmann ve R.K. Cunningham [10], anahtar kelime temelli bir anomali tespit sistemi önermişlerdir. Telnet oturum dökümlerinden elde edilen anahtar kelimelerin hangi sıklıkla meydana geldiklerini ölçmüş ve ardıl saldırı olasılıklarını tespit etmişlerdir. İlk sinir ağı, anahtar kelime temelli sistemden oluşurken, ikinci sinir ağı ise saldırı olarak işaretlenen girdiler üzerinde çalışmaktadır. Bu iki yapay sinir ağından oluşan sistem günde yaklaşık 1 yanlış alarm ile %80 oranında bir algılama sağlamıştır.

Huang, Joseph vd. [8], makine öğrenmesi modeli karşısındaki tehditler için saldırganı ve saldırıyı tanımlayan bir taksonomi önermişlerdir. Eğitim ve test verilerinin alt ve üst sınırlarını araştırarak ve makine öğrenmesi algoritmasının zafiyetlerini keşfederek saldırganın etkinliğini tartışmışlardır. Saldırılar için iki farklı saldırı tanımı yapmışlardır. İlki nedensel (causative) atak; saldırganın eğitim verilerini etkilediği ataklar, diğeri ise keşifsel (exploratory) atak; STS ve spam gibi sistemin tespitinden kaçınarak kör noktalardan faydalanılan ataklardır.

Ilyas, Engstrom vd. [11], yapay sinir ağları temelli kara-kutu sınıflandırıcılar üzerinde saldırıların etkilerini test etmişlerdir. Bu çalışmada üç farklı tehdit modeli tanımlanmıştır. İlki sorgu limitli ayar; sınıflandırıcı belirli sayıda sorguya izin verir ve bu da saldırgan için maliyet oluşturur, ikincisi kısmi bilgi ayarı; saldırgan en üst k adet sınıf etiketine ve bunların olasılık değerlerine erişebilir, üçüncü sadece etiket ayarı; saldırgan olasılık veya skor değerlerine erişemez, sadece etiket değerlerine ve etiketlerin sıralanmış k adedine erişebilir.

D. Lowd ve C. Meek [12], TM temelli bir yaklaşım önermişlerdir. Yöntemin adı Saldırgan Sınıflandırıcı Tersine Mühendislik (ACRE) olarak geçmektedir. Bu yöntem, saldırganın tersine mühendislik teknikleri aracılığı ile kötü amaçlı örnekleri elde edebileceğini belirtmektedir. Örnek bir senaryoda saldırgan, spam filtresini aşmak için bazı spam mesajlarını gizlemek ister ve spam filtresinde hangi kelimelerin bulunduğunu tespit etmeye çalışır. Sisteme gönderdiği sorgular neticesinde spam filtresindeki kelimeleri tespit eder ve spam filtresine takılmayan saldırgan ve istenmeyen mesajlar üretebilir.

Tygar, Barreno vd. [13], kara-kutu sınıflandırıcılara karşı yapılan saldırılar ve tehditleri göstermek adına veri keşfine dayalı yöntem önermişlerdir. Saldırıları tanımlamak adına bir kategorizasyon yapmışlardır. Buna göre saldırıları etki, özgüllük ve güvenlik ihlali ana başlıkları altında toplamışlardır. Etki alanını, nedensel saldırı ve keşifsel saldırı olarak iki kısma ayırmışlar. Nedensel saldırıların amacı yanlış öğrenmeye yöneltmek, keşifsel saldırıların amacı ise sistemin tespitinden kaçınmaktır. Özgüllük alanını, hedeflenen ve ayrım göstermeyen saldırılar olarak iki kısma ayırmışlar. Hedeflenen saldırılar, yalnızca belirli örnekleri etkilemeyi hedeflerken, ayrım göstermeyen saldırılar, örneğin ne olduğunu önemsemeden saldırırlar. Güvenlik ihlali alanını ise bütünlük ve uygunluk alt başlıklarına ayırmışlar. Bütünlük, yanlış pozitif değerlerinin artışını tanımlarken uygunluk, hizmet reddi (DoS) saldırılarını tanımlamaktadır.

Sethi ve Kandartzic [14], saldırıların etkisini göstermek adına Anchor Points(AP) ve Tersine Mühendislik (TM) yöntemlerini uygulamışlardır. AP yöntemi, tek bir seed örneği üzerinden keşif ve saldırı yaparken, TM yöntemi, iki adet farklı sınıfa ait seed örneğini kullanarak ana sınıflandırıcının kopyasını oluşturur ve bu kopya sınıflandırıcı üzerinden saldırılar gerçekleştirir.

Biggio, Fumera ve Roli [15], makine öğrenmesi sınıflandırıcısı ile saldırgan arasında çok kısıtlı bir miktar verinin paylaşıldığını belirtmişlerdir. Saldırgan sadece sınıflandırıcıya bir örnek vektör sorgusu yapar ve sorgunun cevabı olarak ikilik tabanda bir sayısal değer elde eder (0 veya 1). Ayrıca saldırgan çok fazla sorgulama yaptığında ise sistem tarafından engellenir. Araştırmacılar, saldırgan ile makine öğrenmesi sınıflandırıcı arasındaki bilgileri daha iyi anlatmak adına 3 başlıktan oluşan bir saldırgan model formu oluşturmuşlar. İlk başlık olan bilgi (knowledge), saldırganın

verinin boyutu, sınırı veya içeriği hakkında bilgi sahibi olduğunu söyler. Örneğin bir spam filtresi düşündüğümüzde, veriler muhtemelen sözlükte yer alan kelimelerden oluşmaktadır ve saldırgan da bunu bilmektedir. İkinci başlık olan amaçlar (goals), saldırganın amaçlarını kapsar. Saldırganın amaçları, sağlam bir saldırı, yanlış hesapların artışı ve sistem tespitinden kaçınmaktır. Üçüncü ve son başlık olan kaynaklar (resources), saldırganın sadece bir istemci olarak makine öğrenmesi sınıflandırıcısına erişebildiğini, sistemin izin verdiği ölçüde sorgular yapabildiğini ve çıktılar alabildiğini belirtir. Saldırgan ancak sistemin tespitinden kaçınabildiği sürece sorgu yapar ve çıktı alabilir.

M.Barreno, J.D. Tygar vd. [15], saldırganların makine öğrenmesi sistemlerinden faydalanabileceğini ve makine öğrenmesi sitemlerinde güvenlik açıkları olduğunu belirtmişlerdir. Makine öğrenmesi sistemlerine yönelik saldırıları tanımlayan ve analiz eden bir taksonomi önerisinde bulunmuşlardır. Önerilen sistem, saldırgan ve savunmacı taraf için ne kadar maliyet gerektiğini tanımlamaktadır. Geliştirilen sistem, popüler bir istatistiksel spam filtresi olan SpamBayes'e [15] yönelik saldırılara karşı nasıl rehberlik edeceği gösterilmiştir.

J. Cannady [16], kötü amaçlı kullanımı tespit etmek adına yapay sinir ağı tabanlı bir yaklaşım önermiştir. Kullanılan yapay sinir ağı modeli, çok kategorili bir sınıflandırıcıdır. Uygulama için bir RealSecureTM [17] ağ monitörü tarafından oluşturulan veriler kullanılmıştır. Monitörden 3000'i simüle ataklardan olmak üzere toplamda 10000 aktivite elde edilmiştir. Saldırılardan elde edilen sonuçlara göre geliştirilen sistem, saldırgan ve kötüye kullanımı tespit etme safhasında %93 gibi bir başarım sağlamıştır.

1.3 Hipotez

Kara-kutu makine öğrenmesi sınıflandırıcıları, sahte girdiler aracılığı ile ataklar yapılarak yanlış tahmin yapılması sağlanabilir. Ayrıca sınıflandırma başarımları da düşürülebilir. Bu bağlamda mevcut atak yöntemlerinin yanı sıra yeni atak yöntemleri de geliştirilebilir. Bu hipotezi kanıtlamak adına yeni bir atak yöntemi geliştirdik. Geliştirdiğimiz bu atak yöntemini 6 farklı veri seti üzerinde test ederek hipotezimizin doğruluğunu teyit etmiş olduk.

1.4 Tez Organizasyonu

Tez organizasyonu 5 bölümden oluşmaktadır:

Bölüm 1'de; giriş, tezin amacı ve literatür özeti, Bölüm 2'de; materyal ve yöntem, veri setleri, destek vektör makineleri, karar ağaçları ve k-en yakın komşuluğu yöntemleri, Bölüm 3'te; geliştirilen atak yöntemi ve geliştirilen atak yönteminin başarı etkenleri, Bölüm 4'te; bulgular, deneysel kurulum, başarım metrikleri ve deneysel sonuçlar, Bölüm 5'te; sonuçlar ve gelecek vizyonu yer almaktadır.

2. MATERYAL VE YÖNTEM

Bu bölümde tezde geliştirilen atak yöntemi ve yönteme ait sonuçların elde edilmesi sırasında kullanılan yöntemler açıklanmıştır. Yöntemler haricinde deneysel çalışmalar sırasında kullanılan veri setlerine ait detaylı bilgiler verilmiştir.

2.1 Veri Setleri

Geliştirilen atak yöntemini test etmek için 6 farklı veri seti kullanılmıştır. Veri setlerinin öznitelik ve örnek sayıları Çizelge 2.1'de görülmektedir. Bu çalışmada kullanılan veri setleri University of California Irvine (UCI) makine öğrenmesi veri havuzundan elde edilmiştir.

Çizelge 2.1 : Veri setleri

Veri Seti	Öznitelik Sayısı	Örnek Sayısı
Diabetes	9	768
QSAR	42	1055
Credit	62	1000
Cancer	11	699
Spambase	58	4600
Sonar	61	208

Diabetes [18] veri seti, Pima yerlileri arasından en az 25 yaşında olan kadınların bazı tanısal bilgilerini içermektedir. Öznitelikler arasında gebelik sayısı, VKİ (vücut kitle indeksi), insülin düzeyi, yaş vb. yer almaktadır. Öznitelik sayısı 8'dir. 9. öznitelik ise kişinin diyabet hastası olup olmadığını göstermektedir.

QSAR [19] veri seti, biyolojik bozunma ile ilgili bilgiler içermektedir. Öznitelikleri arasında ağır atom sayısı, oksijen atom sayısı, elektronegatiflik vb. bilgiler yer almaktadır. Öznitelik sayısı 41'dir. 42. öznitelik ise biyolojik bozunmak olup olmadığını göstermektedir.

Credit [20] veri seti, bir müşterinin kredi durumu ile ilgili bilgiler içermektedir. Öznitelikler arasında cinsiyet, medeni durum, iş durumu, araba sahipliği, kredi geçmişi

vb. bilgiler yer almaktadır. Öznitelik sayısı 61'dir. 62. öznitelik ise kredi durumunun iyi veya kötü olduğunu göstermektedir.

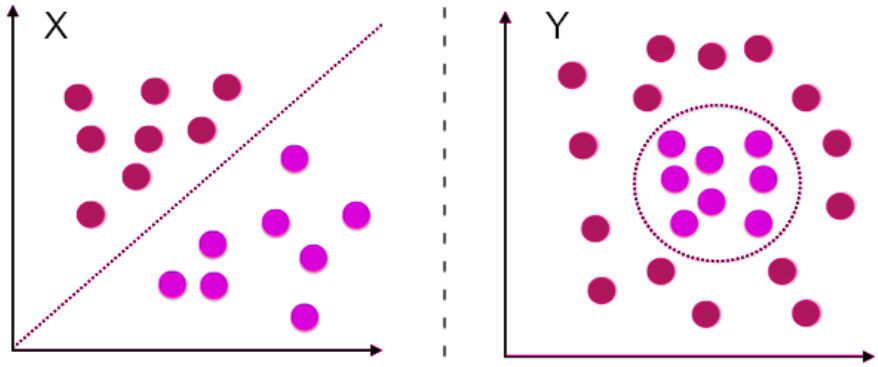
Cancer [21] veri seti, Wisconsin Üniversite Hastanesi'ndeki meme kanseri hastalarının bilgilerini içermektedir. Öznitelikleri arasında hücre boyutu, epitel hücre boyutu, kitle kalınlığı vb. bilgiler yer almaktadır. Öznitelik sayısı 10'dur. 11. öznitelik ise kitlenin iyi veya kötü huylu olduğunu göstermektedir.

Spambase [22] veri seti, istenmeyen e-posta türü olan spam e-postaları tanımlayan bilgiler içermektedir. Öznitelikleri arasında kelime uzunluğu, karakter sıklığı, büyük harf sayısı vb. bilgiler yer almaktadır. Öznitelik sayısı 57'dir. 58. öznitelik ise e-postanın spam olup olmadığını göstermektedir.

Sonar [23] veri seti, Alled Signal Havacılık ve Uzay Teknoloji Merkezi tarafından geliştirilmiştir. Öznitelikleri arasında sonar sinyali, frekans, örüntüler vb. bilgiler yer almaktadır. Öznitelik sayısı 60'tır. 61. öznitelik ise sonar sinyalinin mayından mı yoksa bir kayadan mı geldiğini göstermektedir.

2.2 Destek Vektör Makineleri

Destek vektör makineleri (DVM) [24], 1992 yılında temelleri Vapnik [25] tarafından atılan makine öğrenmesi temelli sınıflandırıcı yöntemlerden biridir. Büyük verilerin anlamlı hale getirilmesini sağlayan DVM, matematiksel formülü aracılığıyla örnek uzaydaki verileri sınıflandırmayı sağlar. DVM, genel olarak iki bölüme ayrılır [26]: Biri doğrusal (linear) DVM, diğeri doğrusal olmayan (non-linear) DVM'dir. Doğrusal DVM, örnek uzaydaki verileri matematiksel formül ile oluşturduğu bir düz çizgi (hyperline) ile birbirinden ayırır. Eğer veriler doğrusal bir çizgi ile birbirinden ayrılamıyorsa bu durumda doğrusal olmayan DVM kullanılır. Şekil 2.1'de iki farklı veri kümesine ait örneklerin destek vektörleri ile birbirinden nasıl ayrılabildiği görülmektedir:



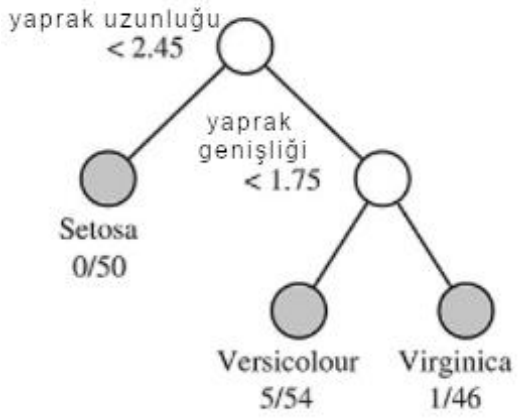
Şekil 2.1 Destek Vektör Makineleri ile Verilerin Ayrılması

Şekilde, X ile gösterilen veri kümesi doğrusal olarak ayrılabiliyorken, Y ile gösterilen veri kümesi doğrusal bir ayırıcı ile ayrılamamaktadır.

2.3 Karar Ağaçları

Karar Ağaçları (Decision Trees), sınıflandırma ve regresyon analizi yapabilen makine öğrenmesi yöntemlerinden biridir [27]. İlk regresyon ağacı algoritmasını 1963 yılında Morgan ve Sonquist [28] geliştirmişlerdir. Karar ağaçları (KA) yöntemi kök, düğümler ve yapraklardan oluşan bir şematik gösterime sahiptir. En üstte bulunan düğüm köktür. Kökün altında yer alan ve kendisinden sonra düğüm veya yaprak bulunan elemanlar ise düğümdür. En altta yer alan ve kendisinden sonra herhangi bir düğüm veya yaprak olmayan eleman ise yapraktır. Kök düğüm seçilirken alt kümenin saflık değerini belirten gini indeks değeri hesaplanır [29]. Değeri en yüksek olan düğüm kök düğüm seçilir. Kök düğümden sonraki düğümlerin seçiminde ise düzensizlik (entropi) hesaplaması yapılır ve en büyük düzensizlik değerine sahip olan düğüm üste yazılır. Bu işlem ağaç tamamlanıncaya kadar devam eder.

Şekil 2.2'de örnek bir KA yapısı görülmektedir. Şekilde yaprak uzunluğu kök düğümü, yaprak genişliği ara düğümü ve diğer gri renktekiler ise yaprakları temsil etmektedir.



Şekil 2.2 Örnek Karar Ağacı

Denk. 2.1'de kök düğümün seçiminde etkili olan gini indeks değerinin hesaplama formülü [29] yer almaktadır:

$$\varphi(t) = 1 - \sum_j p^2(j|t) \quad \textbf{(2.1)}$$

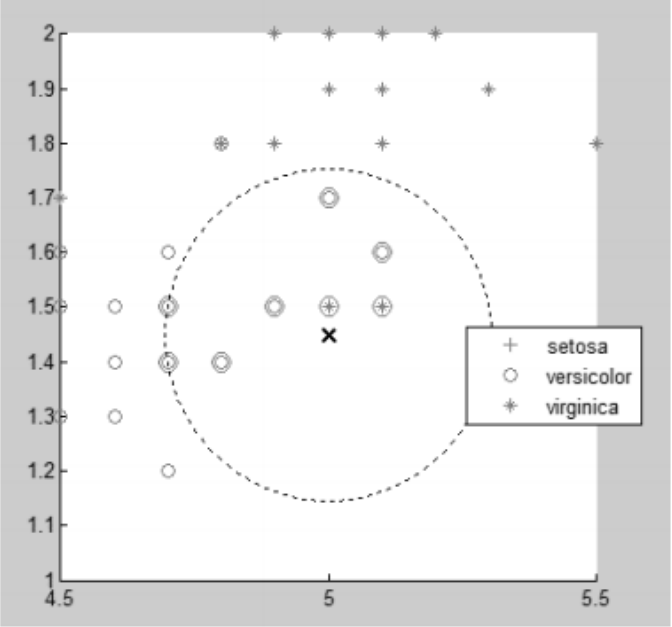
Denk. 2.2'de kök düğümden sonraki düğümlerin seçiminde etkili olan düzensizlik değerinin hesaplama formülü [29] yer almaktadır:

$$\varphi(t) = -\sum_j p(j|t) log p(j|t) \quad \textbf{(2.2)}$$

2.4 k-En Yakın Komşuluğu

k-en yakın komşuluğu (k-EYK) algoritması 1967 yılında T. M. Cover ve P. E. Hart [30] tarafından geliştirilmiştir. Gözetimli öğrenme algoritmalarından olan k-EYK, verileri örnek uzaydaki uzaklıklarına göre sınıflandırır. Her bir veri noktasının sınıf etiketi belirlenirken kendisine en yakın olan *k* adet komşusuna bakılır ve hangi sınıfa ait komşu sayısı fazla ise veri o sınıfa dâhil edilir.

Şekil 2.3'te k=8 olduğu durumda k-EYK algoritmasında sınıflandırma için hangi komşuların dikkate alındığı görülmektedir [31]:



Şekil 2.3 k-En Yakın Komşuluğu Örneği

2.5 Gradyan İniş Yöntemi

Gradyan iniş yöntemi ilk olarak 1847 yılında Cauchy [32] tarafından keşfedilmiştir. Matematik ve fizik alanından çalışmalar yapan Cauchy, astronomide gök cisimlerinin yörüngeleri ile hesaplamalar yaparken altı bilinmeyenli denklemlere çözüm olarak gradyan iniş yöntemini ortaya çıkarmıştır. Her hesaplama sonrasında daha düşük sonuçlara yakınsamayı hedeflemiş ve ulaşmıştır. Bu şekilde optimum değere ulaşıncaya kadar işlemleri tekrar etmiş ve optimum değeri elde etmiştir.

İteratif olarak çalışan gradyan iniş yönteminde, her iterasyonda bir önceki iterasyonda gidilen noktaya çok yakın bir nokta ele alınır ve hesaplama yapılır. Eğer şartlar sağlanıyorsa algoritma çalışmayı durdurur. Aksi halde şartları sağlayana kadar çalışmasına devam eder. Makine öğrenmesi ve yapay zeka alanında, optimizasyon problemlerinde gradyan iniş yöntemi sıklıkla kullanılmaktadır. Özellikle minimum maliyet ve en kısa yoldan çözüme ulaşmak gibi problemlerde bu optimizasyon yöntemine başvurulmaktadır.

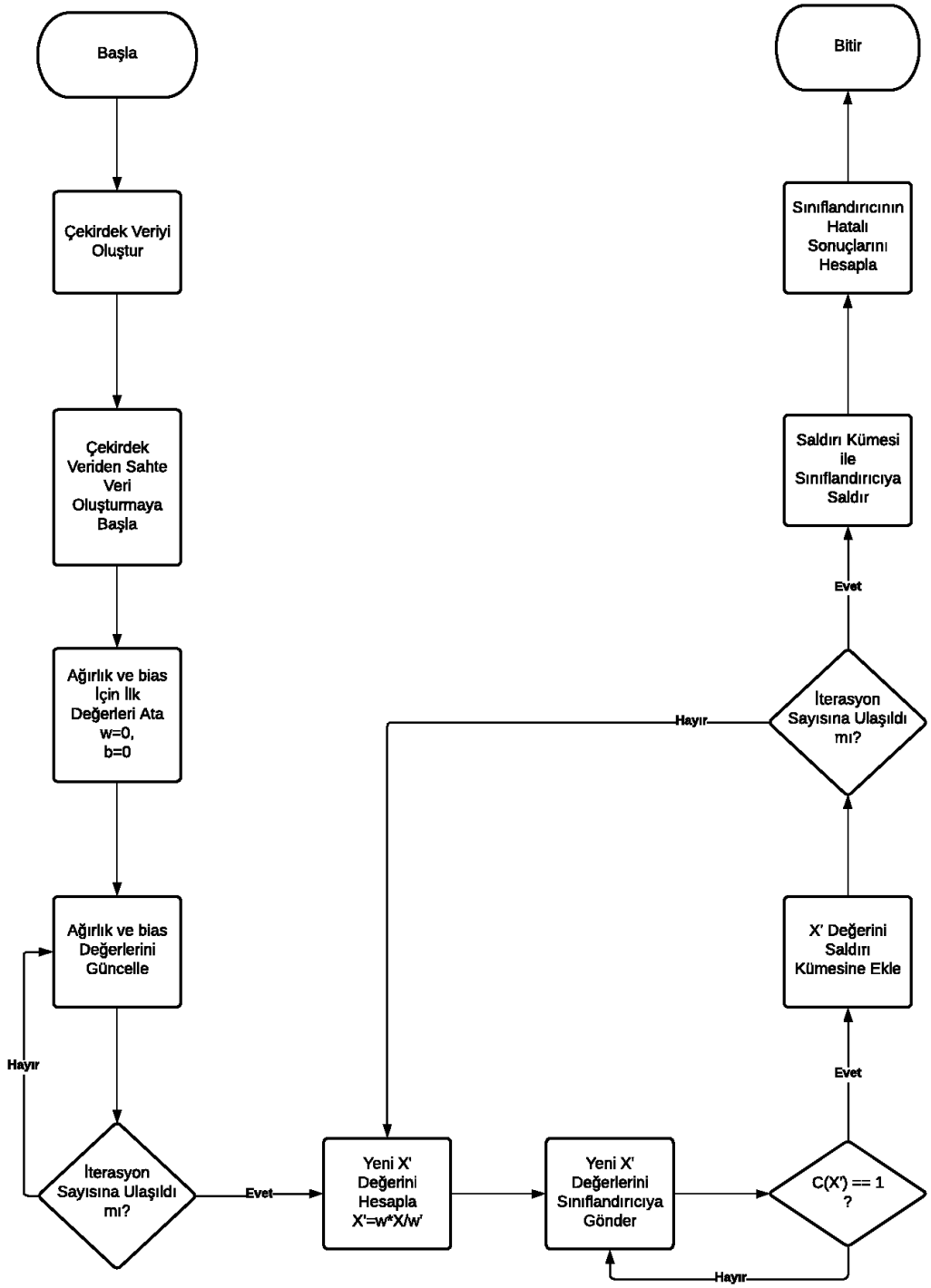
Bu çalışmada ise sahte veri üretimi aşamasında gradyan iniş yöntemi kullanılmıştır. Başlangıçta algoritmaya bir adet çekirdek verisi verilir. Algoritma ağırlık değerlerini güncelleyerek bu veri örneğine yakın bir noktaya gider, eğer bu noktanın sınıf etiket değeri de çekirdek verisi ile aynıysa bir sonraki iterasyonda bu noktayı ve ağırlıkları

esas alarak çalışmaya devam eder. Bu şekilde durdurma kriterini sağlayana kadar devam eder ve sahte verilerden oluşan bir veri kümesi oluşturur.

3. GELİŞTİRİLEN ATAK YÖNTEMİ

Makine öğrenmesi sınıflandırıcı modellerinin saldırılar karşısında zafiyetlerini göstermek için yeni bir atak yöntemi geliştirilmiştir.

Şekil 3.1 geliştirilen atak yönteminin nasıl uygulandığını adım adım gösteren uygulama akış diyagramını içermektedir:



Şekil 3.1 Geliştirilen Yöntemin Akış Diyagramı

Çekirdek (seed), algoritmanın ihtiyaç duyduğu ilk kurulum verisidir. Bu veri sayesinde diğer ihtiyaç duyulan veriler elde edilecektir. Bu veri sınıflandırma probleminde bir X vektörü olmaktadır. Başlangıç için bir adet çekirdek verisi, algoritmanın çalışması için yeterli olmaktadır. Çekirdek verisi rastgele üretilir veya var olan verilerden elde edilebilir. Denk. 3.1'deki formül çekirdek verisinin tanımını göstermektedir:

$$çekirdek = \{rastgele\_seç(X) | C(X) = 1\} \quad \textbf{(3.1)}$$

Keşif (explore) verisi, çekirdek veri kullanılarak oluşturulan X vektörleri kümesidir. Denk. 3.2'deki formül keşif verisinin tanımını göstermektedir:

$$keşif = \{X' | X' = Gradyan\text{İ}niş(çekirdek)\} \quad \textbf{(3.2)}$$

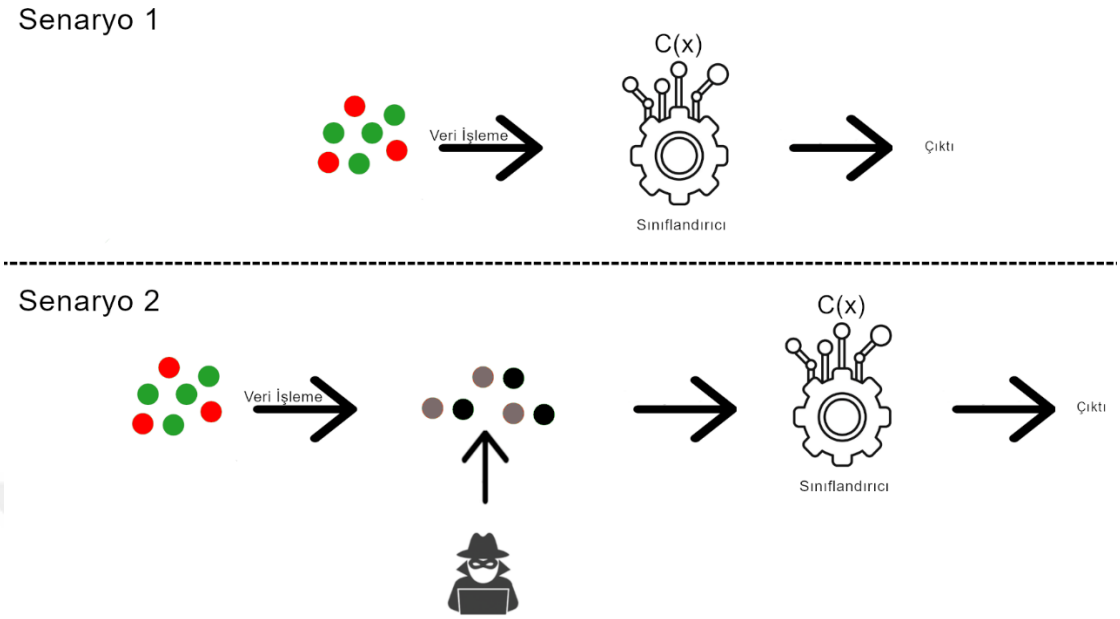
Atak (exploit) verisi ise yöntemin keşif verilerini kullanarak, makine öğrenmesi sınıflandırıcısını sömürmek üzere oluşturduğu yeni X (X') vektörleri kümesidir. Denk. 3.3'teki formül yapılan atağın tanımını göstermektedir:

$$atak: C(X') \quad \textbf{(3.3)}$$

Geliştirilen atak yönteminde algoritma, çekirdek verisinin oluşturulması ile başlamaktadır. Çekirdek veri oluşturulduktan sonra bu veri kullanılarak sınıflandırıcı sisteme saldırmak üzere yeni sahte veriler (X') oluşturulacaktır. Yeni sahte verilerin oluşturulması safhasında ise Gradyan İniş algoritması tabanlı bir optimizasyon yöntemi kullanılmıştır. Bu yöntem ile çekirdek veri kullanılarak çok sayıda yeni sahte veri hızlıca oluşturulabilmektedir. Daha sonra oluşturulan bu sahte veriler, bir veri kümesi haline getirilmektedir. Atak için hazır hale gelen veri kümesi, sınıflandırıcı sisteme gönderilir ve sınıflandırıcı sistemden sonuç değerleri elde edilir. Bu sonuçlardan yanlış olan değerler, sınıflandırıcı sistemin ne derece yönlendirildiğini göstermektedir.

Şekil 3.2'de makine öğrenmesi sınıflandırıcısının sınıflandırma senaryoları görülmektedir. Şekilde "Senaryo 1", normal veriler ile standart sınıflandırmayı gösterirken "Senaryo 2" ise saldırganın müdahalesi ile değiştirilen sahte verilerin

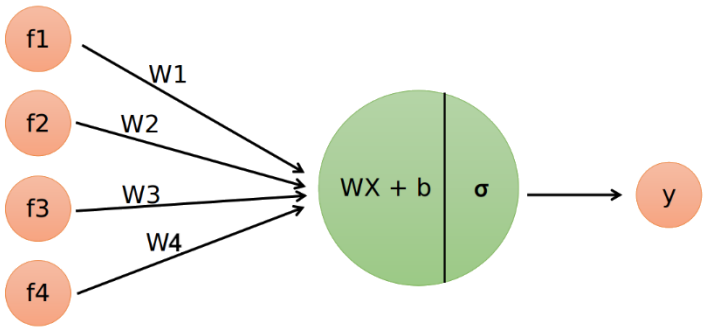
kullanıldığı saldırgan sınıflandırmayı yani geliştirilen atak yönteminin uygulanma biçimini göstermektedir.



Şekil 3.2 Sınıflandırma Senaryoları

Geliştirilen atak yönteminin uygulanmasını adım adım incelersek, ilk olarak kullanılacak veri seti yüklenir. Daha sonra veri setindeki sınıf etiketlerine sahip sütun ayrılır. Eğer bu sütundaki sayısal değerler 0 ve 1'den oluşmuyorsa, öncelikle sınıf etiketleri 0 ve 1'lere dönüştürülür. Bu işlemden sonra kullanılan sınıflandırma yöntemine göre sınıflandırma işlemi yapılır ve elde edilen doğruluk değerleri tutulur. Sınıflandırma işlemi bittikten sonra esas yöntemin çalışması başlar. Burada öncelikle çekirdek verisi oluşturma aşaması yer alır. Çekirdek veri oluşturulurken bir adet sınıf değeri 1 olan veri rastgele oluşturulur veya var olanlar arasından seçilir. Çekirdek veri oluşturulduktan sonra bu veri gradyan iniş metodu kullanılarak yeni sahte verilerin oluşturulması sağlanır. Sahte veri kümesi oluşturulduktan sonra bu veri kümesinin tüm vektörleri sınıflandırıcıya gönderilir ve sınıflandırıcıdan çıktılar alınır. Sınıflandırıcının verdiği çıktılara göre sınıflandırma doğruluğu tekrar hesaplanır ve ilk hesaplanan doğruluk değerleri ile karşılaştırılır. Sahte veriler oluşturulduktan sonra sınıflandırıcıdan gelen çıktılara göre 0 sınıf etiketine sahip olan vektörlerin sayısının tüm sahte verilerin sayısına oranı ise efektif atak oranını (EAO) vermektedir.

Şekil 3.3'te sahte veri üretilirken kullanılan ve ağırlıkları güncelleyen fonksiyonun şeması görülmektedir. Ağırlık hesaplanırken bu şekilde yer alan işlemler kullanılmaktadır:



Şekil 3.3 Ağırlık Fonksiyonu Şeması

Şekil 3.3'te f fonksiyonunun hesaplanması Denk. 3.4'teki formülde gösterildiği şekilde yapılmaktadır:

$$f = W_{(i)}X_{(i)} + bias \quad \textbf{(3.4)}$$

Y çıkışının hesaplanması da Denk. 3.5'teki formüle göre yapılmaktadır:

$$y = \sigma(WX + b) \quad \textbf{(3.5)}$$

Geliştirilen atak yönteminde sezgisel fonksiyon olarak sigmoid [33] fonksiyonu kullanıldığı için yukarıdaki formül Denk. 3.6'daki şekilde açılmaktadır:

$$x = \frac{1}{1 + e^{-(WX+b)}} \quad \textbf{(3.6)}$$

Geliştirilen atak yönteminde sahte girdiler Denk. 3.7'deki formüle göre üretilmektedir:

$$W_{eski}X_{eski} = X_{yeni}W_{yeni}$$
$$X_{yeni}{}^{-1}W_{eski}X_{eski} = X_{yeni}W_{yeni}W_{yeni}{}^{-1} \quad \textbf{(3.7)}$$
$$X_{yeni}\mathrm{I} = X_{yeni}{}^{-1}W_{eski}X_{eski}$$

Formülde *Xeski*, uygulama başlangıcında çekirdek (seed) verisini temsil etmektedir. W_{eski}, ilk ağırlık değeri, W_{yeni} güncellenen ağırlık değeri ve $W_{yeni}{}^{-1}$, güncellenen ağırlık değerinin devriğidir. X_{yeni} ise yeni üretilecek sahte veridir.

Aşağıdaki denklemler, Gradyan İniş metoduna ait parametrelerin nasıl hesaplanacağını göstermektedir.

Denklem 3.8'de x, tahmin edilen değeri temsil ederken y, gerçek değeri temsil etmektedir:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Maliyet(h_\theta(x^{(i)}), y^{(i)}) \quad \textbf{(3.8)}$$

Denklem 3.9'da gerçek değerin 0 ve 1 olduğu durumlarda maliyetin nasıl hesaplanacağı gösterilmektedir:

$$Maliyet(h_\theta(x), y) = \begin{cases} -log(h_\theta(x)), \Rightarrow y = 1 \\ -log(1 - h_\theta(x)), \Rightarrow y = 0 \end{cases} \quad \textbf{(3.9)}$$

Denklem 3.10'da ise ilk iki denklemin açılmış hali gösterilmektedir:

$$J(\theta) = -\left[\sum_{i=1}^{m} y^{(i)} log h_\theta\left(x^{(i)}\right) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))\right] \quad \textbf{(3.10)}$$

Denklem 3.11, sezgisel fonksiyon olarak kullanılan sigmoid [33] fonksiyonunun nasıl hesaplandığını göstermektedir:

$$Sigmoid: h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}} \quad \textbf{(3.11)}$$

Denklem 3.12'de α öğrenme oranını (Learning Rate) göstermektedir:

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \quad \textbf{(3.12)}$$

Denklem 3.13'te bir sonraki adımda değerlerin nasıl güncelleneceğini gösteren formülün açık halini göstermektedir:

$$\theta_j \leftarrow \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta\left(x^{(i)}\right) - y^{(i)}) x_j{}^{(i)} \quad \textbf{(3.13)}$$

3.1 Geliştirilen Atak Yönteminin Başarı Etkenleri

- **Sorgu sayısının az olması:**

  Geliştirilen atak yönteminde kara kutu sınıflandırıcıdan bilgi edinmek için az sayıda sorguya gereksinim duyulmaktadır. EAO'ya sahip olduğu için kara listeye düşme oranı da o kadar azalmaktadır.

- **Yüksek performans ile çalışması:**

  Hem sorgu sayısının az olması hem de algoritmanın hızlı çalışması sebebiyle geliştirilen atak yöntemi benzer diğer yöntemlere göre yüksek performans ile çalışmaktadır.

- **Sınıflandırıcıdan bağımsız olması:**

  Geliştirilen atak yöntemi, sınıflandırıcıdan bağımsızdır. Diğer bir ifade ile kara kutu sınıflandırıcısının hangi algoritmayı kullandığı ile ilgilenmez. Bu da geliştirilen atak yönteminin tüm sınıflandırıcılara adapte olabileceğini göstermektedir.

- **Düşük veri kapasitesi ile yüksek miktarda sahte veri üretebilmesi:**

  Geliştirilen atak yönteminde, az miktardaki sorgu sayısı ile sınırlı ve düşük miktarda keşif verisi elde edilmektedir. Fakat kullanılan algoritma sayesinde bu düşük sayıdaki veri kullanılarak yüksek miktarda sahte veri üretilebilmektedir. Bu da saldırı aşamasında bir avantaj arz etmektedir.

4. BULGULAR

4.1 Deneysel Kurulum

Gerçekleştirilen atak yönteminin test aşamasındaki deneylerde öncelikle kara-kutu sınıflandırıcının doğruluk değerleri olduğu gibi bir bozulmaya uğramadan hesaplanmıştır. Daha sonra aynı kara-kutu sınıflandırıcının doğruluk değerleri geliştirilen atak yöntemi uygulandıktan sonra hesaplanmıştır. Doğruluk değerleri hesaplanırken tesadüfi çok yüksek veya çok düşük sonuçlardan kaçınmak adına 10 kat çapraz doğrulama test tekniği kullanılmıştır. Bu test tekniği sayesinde tüm veriler hem test hem eğitim setinde kullanulmış olmaktadır.

Sınıflandırıcıların performansları hesaplanırken eğitim verisi oranı %80 ve test verisi oranı %20 olarak seçilmiştir.

Veri setleri arasında sınıf etiket değeri 0 ve 1 olmayan veri setleri bulunmaktadır. Bunların sınıf etiketleri 1 ve 2 farklı sayılardan ya da M ve R gibi harflerden oluşmaktaydı. Deney aşamasında bu sınıf etiketleri, karışıklıkları önlemek adına 0 ve 1 olarak değiştirilmiştir.

Oluşturulan algoritmalar, Python 3 kullanılarak Windows işletim sisteminin yüklü olduğu 8 GB bellekli 3.0 GHz hız frekansına sahip bir bilgisayarda gerçekleştirilmiştir.

Deneyler sırasında Python Sci-Kit Learn kütüphanelerinden faydalanılmıştır.

4.2 Başarım Metrikleri

Geliştirilen atak yönteminin başarısını ölçebilmek için Çizelge 4.1'de yer alan karmaşıklık matrisinden (confusion matrix) [33] elde edilen doğruluk değeri kullanılmıştır:

Çizelge 4.1 : Karmaşıklık matrisi

	Öngörülen Sınıf		
Gerçek Sınıf		**Pozitif**	**Negatif**
	Pozitif	Doğru Pozitif (DP)	Yanlış Negatif (YN)
	Negatif	Yanlış Pozitif (YP)	Doğru Negatif (DN)

DP, gerçekte pozitif olan ve deneylerde de pozitif olduğu tahmin edilen değerlerin sayısıdır.
DN, gerçekte negatif olan ve deneylerde de negatif olduğu tahmin edilen değerlerin sayısıdır.
YP, gerçekte negatif olan ama deneylerde pozitif olduğu tahmin edilen değerlerin sayısıdır.

YN ise gerçekte pozitif olan ama deneylerde negatif olduğu tahmin edilen değerlerin sayısıdır.

Geliştirilen yöntemin uygulanması aşamasında kara-kutu sınıflandırıcıların doğruluk değerleri hesaplanırken Denk. 4.1'de yer alan formülden faydalanılmıştır [34]:

$$Doğruluk = \frac{DP + DN}{DP + YP + DN + YN} \quad \textbf{(4.1)}$$

Denk. 4.1'da doğruluk değeri, 0 ile 1 aralığında yer almaktadır.

Deneylerde, kara-kutu sınıflandırıcının doğruluk değerinin saldırı sonrasında ne oranda değiştiğini görebilmek adına da Denk. 17'deki formül kullanılmıştır:

$$Değişim = \frac{(Doğruluk - (Doğruluk'))}{Doğruluk} \quad \textbf{(4.2)}$$

Denk. 4.2'de Doğruluk, kara-kutu sınıflandırıcının ilk doğruluk değerini temsil etmektedir. Doğruluk' ise kara-kutu sınıflandırıcının saldırı sonrasındaki doğruluk değerini temsil etmektedir.

Geliştirilen atak yönteminin başarısını göstermek için kullandığımız metriklerden biri de EAO'dur. EAO, yöntemin makine öğrenmesi sınıflandırıcısı üzerindeki etkinliğini göstermektedir. Denk. 18, EAO değerinin nasıl hesaplanacağını göstermektedir:

$$EAO = \frac{C(x) = 1 \wedge x \in saldırı\_verisi}{|saldırı\_verisi|} \quad \textbf{(4.3)}$$

Denk. 4.3'te pay kısmı, saldırı verileri arasında yer alan ve sınıf etiketi 1 olan x örneklerinin sayısını, payda kısmı ise saldırı verilerinin tümünün sayısını ifade etmektedir.

4.3 Deneysel Sonuçlar

Geliştirilen atak yönteminin başarısını ölçebilmek adına 6 farklı veri seti kullandık. Bu veri setlerini 3 farklı makine öğrenmesi sınıflandırıcısı üzerinde test ettik. Elde ettiğimiz sonuçlar aşağıda yer alan Çizelge 4.2'den Çizelge 4.7'ye kadar olan çizelgelerde yer almaktadır.

Çizelgelerde yer alan ilk Doğruluk kısmı, sınıflandırıcının saldırı öncesi vermiş olduğu sınıflandırma doğruluğunu gösterir. Son Doğruluk kısmı ise geliştirilen atak yönteminin sınıflandırıcıya atak yapması sonucunda sınıflandırıcının vermiş olduğu sınıflandırma doğruluğunu göstermektedir:

Çizelge 4.2 : DVM sınıflandırıcısı için doğruluk değerleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Saldırı Öncesi Doğruluk	0,78	0,85	0,76	0,97	0,92	0,69
Saldırı Sonrası Doğruluk	0,74	0,77	0,76	0,95	0,92	0,66
Düşüş (%)	5,13	9,41	1,04	2,06	0,65	4,34

Çizelge 4.2'ye bakıldığında en yüksek sınıf doğruluğu düşüşü QSAR veri seti üzerinde %9,41 olarak tespit edilmişken, en düşük sınıf doğruluğu düşüşü ise Credit veri seti üzerinde %1,04 olarak tespit edilmiştir.

Çizelge 4.3 : Karar ağaçları sınıflandırıcısı için doğruluk değerleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Saldırı Öncesi Doğruluk	0,69	0,79	0,72	0,89	0,91	0,73
Saldırı Sonrası Doğruluk	0,66	0,79	0,74	0,86	0,84	0,72
Düşüş (%)	4,34	0,75	-	3,27	7,69	1,37

Çizelge 4.3'e bakıldığında en yüksek sınıf doğruluğu düşüşü Spambase veri seti üzerinde %7,69 olarak tespit edilmiştir. Bir önceki makine öğrenmesi yönteminde en düşük sınıf doğruluğu düşüşü Credit veri seti üzerinde elde edilmişken, KA yönteminde Credit veri seti kullanıldığında sınıf doğruluğu düşüşü gerçekleşmemiştir.

Çizelge 4.4 : k-EYK sınıflandırıcısı için doğruluk değerleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Saldırı Öncesi Doğruluk	0,76	0,85	0,76	0,97	0,89	0,75
Saldırı Sonrası Doğruluk	0,73	0,85	0,86	0,96	0,83	0,84
Düşüş (%)	3,95	0,23	-	1,03	6,74	10,6

Çizelge 4.4'e bakıldığında en yüksek sınıf doğruluğu düşüşü Sonar veri seti üzerinde %10,6 olarak tespit edilmiştir.

Geliştirilen yöntemin hedef sistem üzerindeki etkisini ölçmek için EAO değerleri ölçülmüştür. Bu değer, hedef sisteme saldırı için üretilen sahte veriler arasında yer alan ve sınıf etiketi değeri bir olan örneklemlerin sayısını vermektedir. Aşağıda yer alan Çizelge 4.5, Çizelge 4.6 ve Çizelge 4.7 ise geliştirilen atak yönteminin 3 farklı sınıflandırıcı üzerindeki EAO değerlerini göstermektedir:

Çizelge 4.5 : DVM sınıflandırıcısı için efektif atak oranı değerleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
EAO	28,8	22,5	23,7	4,2	7,95	33,7

Çizelge 4.5'e bakıldığında en yüksek EAO değerine 33,7 ile Sonar veri setinde ulaşılmıştır.

Çizelge 4.6 : Karar ağaçları sınıflandırıcısı için efektif atak oranı değerleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
EAO	33,9	20,7	25	14,4	16,5	24,3

Çizelge 4.6'ya bakıldığında en yüksek EAO değerine 33,9 ile Diabetes veri setinde ulaşılmıştır.

Çizelge 4.7 : k-EYK sınıflandırıcısı için efektif atak oranı değerleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
EAO	26,2	14,9	6,98	3,54	19,2	15,5

Çizelge 4.7'ye bakıldığında en yüksek EAO değerine 26,2 ile Diabetes veri setinde ulaşılmıştır.

Atak yöntemlerinin sınıf doğruluklarını düşürmesi elbette beklenen bir şeydir. Fakat gelişirilen yöntemde amaç sınıf doğruluğunu düşürmekten ziyade sınıflandırıcıya yanlış tahmin yaptırmak adına sahte veri üretebilmektir. Bu nedenle doğru bir değerlendirme ve çıkarım yapabilmek için atak yönteminin üretebildiği sahte veri oranları dikkate alınmalıdır. Bunun için geliştirilen atak yöntemi ve literatürde yer alan TM yöntemi karşılaştırılmıştır. Çizelge 4.8, Çizelge 4.9 ve Çizelge 4.10'da geliştirilen yöntemin ve TM yönteminin ürettiği sahte veri oranları görülmektedir. Geliştirilen yöntemin uygulaması aşamasında sınıf etiketi değeri bir olan örnekler sahte veri olarak varsayılmaktadır.

Çizelge 4.8 : DVM sınıflandırıcısı için üretilen sahte veri oranları.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Geliştirilen Yöntem ile Üretilen Sahte Veri Oranı	0,59	0,82	0,75	0,94	0,82	0,63
TM Yöntemi ile Üretilen Sahte Veri Oranı	0,28	0,18	0,22	0,44	0,37	0,1

Çizelge 4.8'e bakıldığında geliştirilen yöntem DVM sınıflandırıcısı üzerinde, Cancer veri setinde 0,94 oranında sahte veri üretirken Diabetes veri setinde ise 0,59 oranında sahte veri üretmiştir. Geliştirilen yöntem, TM yöntemine göre daha yüksek sahte veri üretim oranlarına sahiptir ve bu da geliştirilen yöntemin hedef makine öğrenmesi sınıflandırıcısını daha etkin biçimde yanıltabileceğini göstermektedir.

Çizelge 4.9 : Karar ağaçları sınıflandırıcısı için üretilen sahte veri oranları.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Geliştirilen Yöntem ile Üretilen Sahte Veri Oranı	0,56	0,75	0,69	0,95	0,79	0,88
TM Yöntemi ile Üretilen Sahte Veri Oranı	0,46	0,69	0,69	0,39	0,4	0,18

Çizelge 4.9'a bakıldığında geliştirilen yöntem KA sınıflandırıcısı üzerinde, Sonar veri setinde 0,88 oranında sahte veri üretirken Diabetes veri setinde ise 0,56 oranında sahte veri üretmiştir. Yine geliştirilen yöntem, TM yöntemine göre daha yüksek oranda sahte veri üretebilmiştir.

Çizelge 4.10 : k-EYK sınıflandırıcısı için üretilen sahte veri oranları.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Geliştirilen Yöntem ile Üretilen Sahte Veri Oranı	0,49	0,85	0,88	0,93	0,72	0,56
TM Yöntemi ile Üretilen Sahte Veri Oranı	0,31	0,58	0,75	0,5	0,41	0,33

Çizelge 4.10'a bakıldığında geliştirilen yöntem k-EYK sınıflandırıcısı üzerinde, Cancer veri setinde 0,93 oranında sahte veri üretirken Diabetes veri setinde ise 0,49 oranında sahte veri üretmiştir. TM yöntemi k-EYK sınıflandırıcısı üzerinde yapılan deneylerde geliştirilen yönteme yakın oranda sahte veri üretebilmiş fakat yine geliştirilen yöntem daha yüksek oranda sahte veri üretebilmiştir.

Geliştirilen yöntemin diğer bir performans ölçütü de çalışma performansıdır. Yöntemin hızlı bir şekilde sahte veri üretip saldırıyı tamamlaması hem zamandan hem de kullanılan donanıma veya sunucuya yüklenen işleri hafifletecektir. Bu nedenle geliştirilen yöntemin ve TM yönteminin çalışma süreleri ölçülerek karşılaştırmalar yapılmıştır. Çizelge 4.11, Çizelge 4.12 ve Çizelge 4.13'te ise geliştirdiğimiz yöntemin çalışma performansları gösterilmektedir:

Çizelge 4.11 : DVM sınıflandırıcısı için çalışma performans süreleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Geliştirilen Atak Yönteminin Çalışma Süresi (ms)	2980	4191	3412	3633	5002	4012
TM Yönteminin Çalışma Süresi (ms)	27430	31780	31272	25871	61213	29124

Çizelge 4.11'de geliştirilen yöntem, DVM sınıflandırıcısı üzerinde Diabetes veri setinde 2980 ms'de çalışmasını tamamlamış iken TM yöntemi aynı veri seti üzerinde 27430 ms'de çalışmasını tamamlamıştır.

Çizelge 4.12 : Karar ağaçları sınıflandırıcısı için çalışma performans süreleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Geliştirilen Atak Yönteminin Çalışma Süresi (ms)	2071	4004	2412	2522	4651	3145
TM Yönteminin Çalışma Süresi (ms)	25823	31457	28164	26599	52703	27951

Çizelge 4.12'de geliştirilen yöntem, KA sınıflandırıcısı üzerinde Diabetes veri setinde 2071 ms'de çalışmasını tamamlamış iken TM yöntemi aynı veri seti üzerinde 25823 ms'de çalışmasını tamamlamıştır.

Çizelge 4.13 : k-EYK sınıflandırıcısı için çalışma performans süreleri.

	Veri Seti					
	Diabetes	**QSAR**	**Credit**	**Cancer**	**Spambase**	**Sonar**
Geliştirilen Atak Yönteminin Çalışma Süresi (ms)	3882	5806	5072	5004	6862	4471
TM Yönteminin Çalışma Süresi (ms)	40888	44602	41250	38302	68412	41822

Çizelge 4.13'te geliştirilen yöntem k-EYK sınıflandırıcısı üzerinde Diabetes veri setinde 3882 ms'de çalışmasını tamamlamış iken TM yöntemi aynı veri seti üzerinde 40888 ms'de çalışmasını tamamlamıştır.

5. SONUÇLAR

Bu tez çalışmasında kara-kutu makine öğrenmesi sınıflandırıcıların saldırılar karşısındaki zafiyetlerini göstermek adına yeni bir atak yöntemi geliştirilmiştir. Geliştirilen atak yöntemi, 6 farklı veri seti ve 3 farklı makine öğrenmesi yöntemi üzerinde test edilmiştir. Geliştirilen yöntem üç aşamadan oluşmaktadır. İlk aşama çekirdek verisinin oluşturulması sağlanmaktadır. İkinci aşamada bu çekirdek verisi, gradyan iniş metodu kullanılarak çoğaltılır ve keşif verisi oluşturulmaktadır. Üçüncü aşamada ise bu keşif verisi ile kara-kutu sınıflandırıcıya atak yapılmaktadır. Geliştirilen atak yöntemi, destek vektör makineleri, karar ağaçları ve k-en yakın komşuluğu algoritması tabanlı sınıflandırıcılar üzerinde 6 farklı veri seti kullanılarak test edilmiştir. Elde edilen test sonuçlarına göre en yüksek sınıf doğruluğu düşüşü, %9,41 ile destek vektör makineleri algoritmasının kullanıldığı sınıflandırıcı ve QSAR veri setinde sağlanmıştır. Fakat yöntemimizde amaç sınıf doğruluğunu düşürmekten ziyade sınıflandırıcıya yanlış tahmin yaptırmak adına sahte veri üretmektir. Bu nedenle Çizelge 4.8, Çizelge 4.9 ve Çizelge 4.10'da bulunan üretilen sahte veri oranlarına bakmamız gerekmektedir. En yüksek sahte veri üretim oranı karar ağaçları algoritmasının kullanıldığı sınıflandırıcı ve Cancer veri seti üzerinde 0,95 olarak tespit edilmiştir. En düşük sahte veri üretim oranı ise k-en yakın komşuluğu algoritmasının kullanıldığı sınıflandırıcı ve Diabetes veri seti üzerinde 0,49 olarak tespit edilmiştir. Literatürde yer alan diğer bir yöntem olan TM yöntemine göre, geliştirilen atak yöntemi oldukça fazla sahte veri üretme başarısına ulaşmıştır. EAO sonuçlarına bakıldığında ise en yüksek değere 33,9 ile karar ağaçları algoritmasının kullanıldığı sınıflandırıcı ve Diabetes veri setinde ulaşılmıştır. En düşük EAO değeri ise 3,54 ile k-en yakın komşuluğu algoritmasının kullanıldığı sınıflandırıcı ve Cancer veri setinde elde edilmiştir. Geliştirilen atak yöntemi algoritmasının çalışma performansına bakıldığında ise en düşük çalışma süresi karar ağaçları algoritmasının kullanıldığı sınıflandırıcı ve Diabetes veri seti üzerinde 207 ms olarak ölçülmüştür. En yüksek çalışma süresi ise k-en yakın komşuluğu algoritmasının kullanıldığı sınıflandırıcı ve Spambase veri seti üzerinde 686 ms olarak ölçülmüştür. Geliştirilen atak yöntemi, TM yönteminin çalışma zamanına oranla oldukça iyi performans göstermiştir.

Gelecekte yapılacak olan çalışmalarda, geliştirilen atak yöntemi farklı sınıflandırıcılar ile farklı veri setleri üzerinde test edilebilir. Geliştirilen atak yöntemi, çevrimiçi bir sunucuda yer alan kara-kutu sınıflandırıcıda kullanılabilir. Ayrıca geliştirilen atak yöntemini ve literatürde yer alan diğer atak yöntemlerini etkisiz hale getirebilecek atak savunma sistemleri geliştirilebilir.

KAYNAKLAR

[1] S. Sharma Ve T. Dhote, Cybersecurıty – Vulnerability Assessment Of Attacks, Challenges And Defence Strategıes in Industry 4.0 Ecosystem, C. 10, S. 8, 2021.

[2] B. Cashell, W. D. Jackson, M. Jickling, ve B. Webel, The Economic Impact of Cyber-Attacks, s. 45.

[3] E. Toch vd., The Privacy Implications of Cyber Security Systems: A Technological Survey, ACM Comput. Surv., c. 51, sy 2, ss. 1-27, Haz. 2018, doi: 10.1145/3172869.

[4] S. J. Russell ve P. Norvig, Artificial intelligence: a modern approach. Englewood Cliffs, N.J: Prentice Hall, 1995.

[5] J. Heaton, Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618, Genet. Program. Evolvable Mach., c. 19, sy 1-2, ss. 305-307, Haz. 2018, doi: 10.1007/s10710-017-9314-z.

[6] I. El Naqa ve M. J. Murphy, What Is Machine Learning?, Machine Learning in Radiation Oncology, I. El Naqa, R. Li, ve M. J. Murphy, Ed. Cham: Springer International Publishing, 2015, ss. 3-11.

[7] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, ve A. Swami, Practical Black-Box Attacks against Machine Learning, Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, Abu Dhabi United Arab Emirates, Nis. 2017, ss. 506-519, doi: 10.1145/3052973.3053009.

[8] L. Huang, A. D. Joseph, B. Nelson, B. I. P. Rubinstein, ve J. D. Tygar, Adversarial machine learning, s. 15.

[9] P. Laskov ve M. Kloft, A framework for quantitative security analysis of machine learning, Proceedings of the 2nd ACM workshop on Security and artificial intelligence - AISec ’09, Chicago, Illinois, USA, 2009, s. 1, doi: 10.1145/1654988.1654990.

[10] R. P. Lippmann ve R.K. Cunningham, Improving intrusion detection performance using keyword selection and neural networks, Computer networks 34.4 (2000): 597-603.

[11] A. Ilyas, L. Engstrom, A. Athalye, ve J. Lin, Black-box Adversarial Attacks with Limited Queries and Information, s. 10.

[12] Lowd, Daniel, ve Christopher Meek, Adversarial learning, Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. 2005.

[13] Barreno, Marco, vd., The security of machine learning, Machine Learning 81.2 (2010): 121-148.

[14] T. S. Sethi ve M. Kantardzic, Data driven exploratory attacks on black box classifiers in adversarial domains, Neurocomputing, c. 289, ss. 129-143, May. 2018, doi: 10.1016/j.neucom.2018.02.007.

[15] B. Biggio, G. Fumera, ve F. Roli, Pattern Recognition Systems Under Attack: Design Issues and Research Challenges, Int. J. Pattern Recognit. Artif. Intell., c. 28, sy 07, s. 1460002, Kas. 2014, doi: 10.1142/S0218001414600027.

[16] M. Barreno, B. Nelson, R. Sears, A. D. Joseph, ve J. D. Tygar, Can machine learning be secure?, Proceedings of the 2006 ACM Symposium on Information, computer and communications security - ASIACCS '06, Taipei, Taiwan, 2006, s. 16, doi: 10.1145/1128817.1128824.

[17] Cannady, James, Artificial neural networks for misuse detection, Proceedings of the 1998 National Information Systems Security Conference (NISSC'98). 1998.

[18] RealSecureTM, ISS – Internet Security Systems, Dökümantasyon, http://www.iss.net/support/documentation/

[18] UCI Machine Learning Repository, Pima Indians Diabetes Dataset. [Çevrimiçi]. Erişim adresi: http://archive.ics.uci.edu/ml/datasets/Pima+Indians+Diabetes.

[19] UCI Machine Learning Repository, QSAR biodegradation Data Set. 2013, [Çevrimiçi]. Erişim adresi: http://archive.ics.uci.edu/ml/datasets/QSAR+biodegradation.

[20] UCI Machine Learning Repository, Statlog (German Credit Data) Data Set. 1994, [Çevrimiçi]. Erişim adresi: http://archive.ics.uci.edu/ml/datasets/statlog+(german+credit+data).

[21] UCI Machine Learning Repository, Breast Cancer Wisconsin (Original) Data Set. 1992, [Çevrimiçi]. Erişim adresi: http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original).

[22] UCI Machine Learning Repository, Spambase Data Set. 1999, [Çevrimiçi]. Erişim adresi: http://archive.ics.uci.edu/ml/datasets/spambase.

[23] UCI Machine Learning Repository, Connectionist Bench (Sonar, Mines vs. Rocks) Data Set. [Çevrimiçi]. Erişim adresi: http://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks).

[24] Hearst, M. A., Dumais, S. T., Osman, E., Platt, J., ve B, Scholkopf., Support vector machines, 1998.

[25] Vapnik, VN, Boser, BE, ve Guyon, IM, A training algorithm for optimal margin classifiers, 1992.

[26] Hastie, T., Tibshirani, R., ve Friedman, J., The Elements of Statistical Learning, s. 764, 2009.

[27] Yan-yan SONG ve Ying LU, Decision tree methods: applications for classification and prediction, 2015, doi: 10.11919/j.issn.1002-0829.215044.

[28] J. N. Morgan ve J. A. Sonquist, Some Results From a Non- Symmetrıcal Branchıng Process That Looks For Interaction Effects, s. 14.

[29] W.-Y. Loh, Fifty Years of Classification and Regression Trees: Fifty Years of Classification and Regression Trees, Int. Stat. Rev., c. 82, sy 3, ss. 329-348, Ara. 2014, doi: 10.1111/insr.12016.

[30] T. Cover ve P. Hart, Nearest neighbor pattern classification, IEEE Trans. Inf. Theory, c. 13, sy 1, ss. 21-27, Oca. 1967, doi: 10.1109/TIT.1967.1053964.

[31] K. Chomboon, P. Chujai, P. Teerarassammee, K. Kerdprasop, ve N. Kerdprasop, An Empirical Study of Distance Metrics for k-Nearest Neighbor Algorithm, The Proceedings of the 2nd International Conference on Industrial Application Engineering 2015, 2015, ss. 280-285, doi: 10.12792/iciae2015.051.

[32] A. Cauchy, Methode generale pour la resolution des systemes d'equations simultanees, C. R. Acad. Sci. Paris, 25:536–538, 1847.

[33] P. de B. Harrington, Sigmoid transfer functions in backpropagation neural networks, Anal. Chem., c. 65, sy 15, ss. 2167-2168, Ağu. 1993, doi: 10.1021/ac00063a042.

[34] A. Tharwat, Classification assessment methods, Appl. Comput. Inform., c. 17, sy 1, ss. 168-192, Tem. 2020, doi: 10.1016/j.aci.2018.08.003.

ÖZGEÇMİŞ

Ad Soyad: Emre SADIKOĞLU

Mesleki Deneyim ve Ödüller:

Yalova Üniversitesi Bilgisayar Mühendisliği Bölümü Araştırma Görevlisi (2020 - Hâlen)

Yayın ve Patent Listesi:

Sadıkoğlu, E , Demirelli Okkalıoğlu, B . (2020). Çok Kriterli Ürün-Tabanlı İşbirlikçi Filtrelemede Ağırlıklandırma Yöntemlerini Kullanarak Tahmin Performansının Arttırılması. Avrupa Bilim ve Teknoloji Dergisi, Ejosat Özel Sayı 2020 (HORA) , 110-121 . DOI: 10.31590/ejosat.779171

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

[1] Sadıkoğlu E, Demirelli Okkalıoğlu B., Kösesoy İ, Revealing Vulnerability of Different Machine Learning Models Using Reverse Engineering Method, ICONSEC (International Conference on Cyber Security and Digital Forensics), 4 Haziran 2021, Yalova, TÜRKİYE