**T.C.**
**HALİÇ ÜNİVERSİTESİ**
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**
**BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI**

# MAKİNE ÖĞRENİMİ YÖNTEMLERİ İLE AĞ TRAFİĞİNİN ANOMALİ TABANLI ANALİZİ

**YÜKSEK LİSANS TEZİ**

**Hazırlayan**
**Ahmet Yasir KALAYCI**

**Danışmanı**
**Doç. Dr. Ülviye HACIZADE**

**İSTANBUL**
**Aralık- 2023**

**T.C.**
**HALİÇ ÜNİVERSİTESİ**
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**
**BİLGİSAYAR MÜHENDİSLİĞİ TEZLİ YÜKSEK LİSANS PROGRAMI**

# MAKİNE ÖĞRENİMİ YÖNTEMLERİ İLE AĞ TRAFİĞİNİN ANOMALİ TABANLI ANALİZİ

**YÜKSEK LİSANS TEZİ**

**Hazırlayan**
**Ahmet Yasir KALAYCI**

**Danışmanı**
**Doç. Dr. Ülviye HACIZADE**

**İSTANBUL**
**Aralık- 2023**

M6- Yüksek Lisans Tez Onay Sayfası

19/12/2023

# TEZ ETİK BEYANI

Yüksek Lisans Tezi olarak sunduğum “Makine Öğrenimi Yöntemleri ile Ağ Trafiğinin Anomali Tabanlı Analizi” başlıklı bu çalışmayı baştan sona kadar danışmanım Doç. Dr. Ülviye HACIZADE’nin sorumluluğunda tamamladığımı, farklı kaynaklardan edindiğim bilgileri metin içerisinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma süresi boyunca bilimsel araştırma ve etik kurallara uygun olarak davrandığımı beyan ederim.

Ahmet Yasir KALAYCI

# ÖNSÖZ

Çalışma süreci boyunca her adımda bana rehberlik eden, engin tecrübeleriyle yol gösteren hocam ve danışmanım Sayın Doç.Dr. Ülviye HACIZADE'ye, her konuda yanımda olarak bana destek olan sevgili aileme, çalışmaya değerli vaktini ayıran tüm katılımcılara ve bu katılımcılara ulaşmamda bana yardımcı olan tüm arkadaşlarıma gönülden teşekkür eder minnettarlığımı belirtmek isterim.

**Aralık, 2023** Ahmet Yasir KALAYCI

# İÇİNDEKİLER

**Sayfa No**

# KISALTMALAR

| | | |
|---|---|---|
| **DP** | **:** | Doğru Pozitif |
| **YP** | **:** | Yanlış Pozitif |
| **ASTS** | **:** | Anomali Tabanlı Saldırı Tespit Sistemi |
| **İSTS** | **:** | İmza Tabanlı Saldırı Tespit Sistemi |
| **YPO** | **:** | Yanlış Pozitif Oran |
| **YNO** | **:** | Yanlış Negatif Oran |
| **STS** | **:** | Saldırı Tespit Sistemi |

# TABLO LİSTESİ

**Sayfa**

# ŞEKİL LİSTESİ

**Sayfa**

# ÖZET

## MAKİNE ÖĞRENİMİ YÖNTEMLERİ İLE AĞ TRAFİĞİNİN ANOMALİ TABANLI ANALİZİ

Teknoloji ve internet, günlük yaşantımızın vazgeçilmez bir parçası haline gelmesiyle birlikte internete erişim birçok sektör için hayati bir öneme sahip olmasına neden olmuştur. Bu durum aynı zamanda siber saldırıların sayısını ve etkisini artırmış sistemlerin saldırılardan korunması, üzerinde düşünülmesi gereken bir konu haline getirmiştir. Siber güvenlik, ülkelerin savunma stratejilerinde yeni bir önemli unsur olarak kabul edilmeye başlanmış ve pek çok ülke, kamu ve özel sektör, siber savunma alanında önemli adımlar atmış ve atmaya devam etmektedir. Bilgisayar ağları üzerinde meydana gelen saldırılardan korunmak ve mümkün olduğunca engellemek için, Saldırı Tespit Sistemlerinin kullanılması verilerin güvenlik altına alınmasında en önemli konulardan biri olduğu değerlendirilmiştir.

Bu kapsamda, tez çalışmasında saldırı tespit sistemleri üzerine yapılmış çalışmalar titizlikle incelenmiş ve UNSW-NB15 veri seti kullanılarak Lojistik regresyon, K-en yakın komşu algoritması, Destek Vektör Makineleri, Naive Bayes, Karar Ağacı ve Rastgele Orman Algoritması gibi makine öğrenimi yöntemleri eğitim ve test aşamalarına tabi tutulmuştur. Eğitim aşamasında kullanılan makine öğrenimi yöntemlerine normal ağ trafiği öğretilmiş ve bu bilgilerle bir referans model oluşturulmuştur. Test aşamasında ise, oluşturulan model ağ trafiğini izleyerek anormal aktiviteleri tespit etmeye çalışır. Eğer sistem, belirlenen normal davranıştan sapmalar tespit ederse, potansiyel bir saldırı veya tehdit olduğunu değerlendirir. Son olarak algoritmaların anomali trafiğin tespit edilmesindeki başarı oranları, karışıklık matrisi kullanılarak değerlendirilmiştir.

**Anahtar Kelimeler:** *K en yakın komşu algoritması, Makine öğrenimi, Saldırı tespit sistemi, Siber saldırı, Yapay zekâ*

# ABSTRACT


## ANOMALY BASED ANALYSİS OF NETWORK TRAFFİC WİTH MACHİNE LEARNİNG METHODS

As technology and the internet have become an indispensable part of our daily lives, access to the internet has become of vital importance for many sectors. This situation has also increased the number and impact of cyber attacks, making protecting systems from attacks an issue that needs to be considered. Cyber security has begun to be accepted as a new important element in the defense strategies of countries, and many countries, public and private sectors, have taken and continue to take important steps in the field of cyber defense. In order to be protected from attacks occurring on computer networks and to prevent them as much as possible, the use of Intrusion Detection Systems has been evaluated as one of the most important issues in securing data.

In this context, in the thesis study, studies on intrusion detection systems were meticulously examined and machine learning methods the UNSW-NB15 dataset Logistic regression, K-nearest neighbor algorithm, Support Vector Machines, Naive Bayes, Decision Tree and Random Forest were trained and tested has been subjected to stages. Normal network traffic was taught to the machine learning methods used in the training phase and a reference model was created with this information. In the testing phase, the created model tries to detect abnormal activities by monitoring network traffic. If the system detects deviations from the established normal behavior, it considers there to be a potential attack or threat. Finally, the success rates of the algorithms in detecting anomaly traffic were evaluated using the confusion matrix.



**Keywords:** *Artificial intelligence, Cyber attack, Intrusion detection system, K nearest neighbor algorithm, Machine learning.*

# 1. GİRİŞ

İnternet insanlar için vazgeçilmez bir kaynak haline gelmiştir. 2014 yılında yapılan çalışmalara göre dünya nüfusunun yaklaşık %40'ının internet kullandığı tespit edilmiştir ve bu rakamın gelişmiş ülkelerde %78'e kadar çıktığı bildirilmiştir. Kuzey Atlantik Antlaşması Örgütü (NATO), interneti "hükümetler için kritik bir ulusal kaynak, ulusal altyapıların hayati bir parçası ve sosyo-ekonomik büyüme ve kalkınmanın temel itici gücü" olarak tanımlamaktadır (Scholarlycommons et al., 2015). İnternet kullanımının yaygınlaşmasıyla bağlantılı olarak, kötü amaçlı kod ve yazılımların bilgisayar sistemlerini tehlikeye atarak içerdikleri bilgilere saldırdıkları ortaya çıkmıştır. Bu tür saldırılar, sadece kullanıcıların kredi kartı numaraları veya şifreleri gibi bilgilerini toplamak için değil, aynı zamanda kullanıcının izni olmadan bilgi dağıtmak için tasarlanırlar (Kolter et al.,2006 ). Kötü amaçlı yazılım, verilere ve sistemlere zarar verebilecek yazılım olarak tanımlanır (Altaher et al., 2012). Bu yazılımlar sadece bireyler için değil, aynı zamanda hem sivil hem de askeri altyapılar dahil olmak üzere (Bauer et al., 2009) değerli bilgilerini ve itibarlarını kaybetme riski altında olan kuruluşlar, şirketler ve hatta hükümetler için bir tehdittir (Vazquez, 2014). Bilgisayar ağlarının güvenliğini sağlamak için çeşitli güvenlik sistemleri geliştirilmiştir. Bu sistemler, bilgisayarların ve ağ altyapılarının siber saldırılara karşı korumasını ve saldırıların tespit edilmesi için bilgisayarların ve ağ alt yapılarında çalışmak üzere tasarlanan güvenlik sistemleridir. Ağ tabanlı saldırı tespit sistemleri, sistemlerin ağ tabanlı saldırılara karşı en etkili savunma yöntemidir. Bu sistemler çoğunlukla tüm büyük ölçekli bilgi sistem altyapılarında kullanılmaktadır (Debar et al., 2000). Temel olarak, imza tabanlı ve anomali tabanlı olmak üzere iki çeşit saldırı tespit sistemi vardır (Roesch, 1999).

İmza tabanlı saldırı tespit sistemi (İSTS), önceden bilinen saldırıların imzalarını veri tabanlarında depolayarak, analiz edilen verilerle bu imzalar arasında örüntü tanıma teknikleriyle karşılaştırılması esasına dayanmaktadır. İmzaların eşleşmesi durumunda bir alarm tetiklenir. Bu sistemler, kötüye kullanım durumlarını analiz ederken önceden tanımlanmış saldırı imza modellerine bakar ve eğer bir eşleşme tespit edilirse, saldırıyı engelleme işlemi gerçekleştirirler. Ancak, tanımlanmış saldırı imzası ile eşleşme

sağlanmıyorsa, sistem ilgili durumu saldırı olarak değerlendirmez. (Bace, 2000; Bace and Mell, 2001; McHugh et al., 2000). Öte yandan anomali tabanlı saldırı tespit sistemleri (ASTS), normal ağ trafiğini tanımlayan istatistiksel bir model oluşturur ve modelden sapan herhangi bir davranış anomali trafik olarak belirlenir. İmza tabanlı sistemlerin aksine, anomali tabanlı sistemler, sıfır gün saldırılarını tespit edebilme özelliğine sahiptir, çünkü yeni saldırılar gerçekleşir gerçekleşmez tespit edilebilir (Wang and Stolfo, 2004). Anomali ağ trafiği, kullanıcıların kendi hak ve sınırlarını aşması veya ağ bağlantılarının akışını engelleyecek kadar sınırları aşan ağ trafikleridir. Ağda oluşan anomali durumları tespit etmek mümkündür. Bu tespit sistemleri ağdan bilgi toplama veya hizmeti durdurma gibi bilinen saldırıları yakalamada kullanılır. Anomali tabanlı sistemler, imza tabanlı sistemlere göre avantajı ilk defa gerçekleşen saldırıları yakalamada daha etkilidir (Kemmerer and Vigna,2002).

Her geçen gün internetin yaygınlaşması ve bağlı cihazların sayısının hızla artması, bir dizi avantajın yanı sıra çeşitli sorunları da beraberinde getiriyor. Bu sorunların en ciddisi, siber saldırılardır. Bireylerden kurumlara ve devletlere kadar geniş bir yelpazede gerçekleştirilen siber saldırılar, maddi, itibari ve zaman kayıplarına neden olabilir. Bu kayıpları azaltmak veya ortadan kaldırmak için saldırı tespit ve önleme sistemleri kullanılır. Saldırı tespit sistemleri genellikle imza tabanlı veya anomali tabanlı olarak tasarlanır, günümüzde anomali tabanlı sistemler genellikle makine öğrenimi yöntemleri kullanılarak geliştirilir.

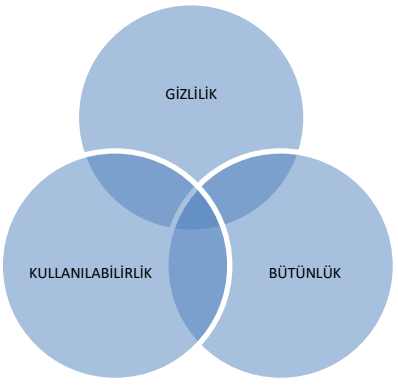
Yapılan bu tez çalışmasında, makine öğrenimi yöntemleri kullanılarak anomali tabanlı bir saldırı tespit sistemi tasarlamıştır. Bu çalışmada makine öğrenimi yöntemlerinin başarılarının tespit edilmesi için UNSW-NB15 hazır veri seti kullanılmıştır.

# 2. GENEL BİLGİLER

Makine öğrenimi, günümüz bilgi çağında önemli bir role sahip olan ağ trafiğinin anomali tabanlı analizi konusunda önemli bir araştırma alanı haline gelmiştir. Bu alandaki çalışmalar, ağ güvenliği uzmanlarının, potansiyel tehditleri tespit etme ve önleme konusundaki becerilerini artırmayı hedeflemektedir. Ağ trafiği analizi, büyük veri kümeleri üzerinde yapılan karmaşık işlemlerle gerçekleştirilir ve bu süreçte makine öğrenimi yöntemleri kullanılarak anormal davranışlar tespit edilmeye çalışılır. Bu tez çalışması, makine öğrenimi tekniklerinin ağ trafiği analizi alanındaki uygulamalarını inceleyerek, güvenlik açısından kritik öneme sahip olan anomali tespiti konusunda daha derinlemesine bir anlayış sunmayı amaçlamaktadır. Bu bağlamda, genel bilgiler bölümü, makine öğrenimi ve ağ trafiği analizi kavramlarını ele alacak ve tezin temelini oluşturan anahtar konulara odaklanacaktır.

## 2.1. Bilgi Güvenliği

Bilgi güvenliği amacı, siber olaylarının etkisini sınırlandırarak iş sürekliliğini sağlamak ve iş zararını en aza indirmektir (Solms, 1998).Bilgi güvenliği, Şekil 2.1'de görüldüğü üzere üç başlıkta tanımlanabilir.




**Şekil 2.1.** Bilgi Güvenlik Üçlüsünün Bir Revizyonu

### 2.1.1. Gizlilik

'Gizlilik' terimi, Latince de güvenmek anlamına gelen confidere fiilinden türetilmiştir. Gizlilik, bilgi güvenliğinin temel ilkelerinden biridir. Camp (1999), gizliliğin, verilerin ve veriler tarafından temsil edilen bilgilerin korunması gerektiği fikrini ima ettiğini öne sürer; kullanımı yalnızca yetkili kişiler tarafından izin verilen amaçlarla sınırlandırılacak şekilde korunmasıdır (Camp, 1999). Benzer şekilde, Zwick ve Dholakia (2004) gizliliği, içinde açıklananlara ve onu kimin görebileceğine göre bilgi akışını kısıtlayan yetenek olarak tanımlamaktadır. (Zwick and Dholakia, 2004). Gizliliğin bu yönleri, resmi devlet belgelerine ve mevzuatına da yansır. Örneğin, ABD Yasasının 44. Başlığının 3542. Bölümünde gizlilik, "kişisel mahremiyeti ve özel bilgileri koruma araçları da dahil olmak üzere, bilgi erişiminin ve ifşasının yetkili kısıtlaması" olarak anılır.

Bilgi teknolojisinin benimsenmesinin ilk günlerinden beri gizlilik bilgi güvenliğinin merkezinde yer alıyorken, iş ihtiyaçlarında meydana gelen değişiklere bağlı olarak oluşan güvenlik kaygılarıyla birlikte bilgi güvenliğinin önemi azalmıştır. Fitzgerald (1995), bilgi gizliliğinin artık önemli bir sorun olmadığını kaydetti. Bununla birlikte, gizliliğin mahremiyet yönlerinin gelecekte, özellikle sağlık ve finans gibi hassas kişisel bilgilerin yönetimine odaklanan ana iş odaklarının olduğu sektörlerde öneminin artacağına da işaret etti ( Fitzgerald, 1995).

Mahremiyet ve güven ilişkisi, bilgi sistemleri literatüründe pek çok farklı açıdan incelenen önemli bir konudur. Katzan (2011), bir kuruluşun bilgi sistem uygulamalarına ilişkin dürüstlüğünün ve hesap verebilirliğinin, mahremiyetle ilgili endişeleri gidermek ve kullanıcı güveni oluşturmak için önemli olduğunu öne sürmüştür (Katzan,2011). Wang et al. (1998), internet müşterileri tarafından belirlenen en kritik sorunun, elektronik ticaret pazarlarında kişisel mahremiyetin kaybıyla ilgili korku ve güvensizlik olduğunu iddia etmektedir (Wang et al., 1998). Güven konusunda sosyal değişim teorisi bakış açısını benimseyen araştırmacılar, güvenin işletmelerin üzerine inşa edildiği en önemli varlık olduğunu ileri sürmektedirler (Luo, 2002 ).

### 2.1.2. Bütünlük

Etimolojik olarak sağlamlık anlamına gelen bütünlük kelimesi, Latincede dokunmak anlamına gelen tangere kelimesinden türemiştir. Bilginin yetkisi olmayan kişilerce değiştirilmemesi, yani bilginin doğru ve gerektiği şekilde muhafaza edilmesi ve korunması prensibidir. Etik, doğru ve yanlış davranış kavramlarını

sistematikleştirmeyi, savunmayı ve önermeyi içeren bir süreç olarak tanımlanmaktadır. (Fieser et al., 2006).

### 2.1.3. Kullanılabilirlik

Kullanılabilirlik kelimesi, değerli olmak anlamına gelen Latince valere'den gelir. Bilgi güvenliğinde, kullanılabilirlik terimi "bilgiye zamanında ve güvenilir erişim" anlamına gelir. Kullanılabilirlik mühendislik perspektifine göre bir sistem etkili ve verimli olduğunda kullanılabilir olarak kabul edilir.(Weir et al., 2009). Güvenlik yazılımı söz konusu olduğunda, Padayachee (2012), Whitten and Tygar'ın (1999) kullanılabilirliğin aynı zamanda tehlikeli hatalardan kaçınma ve kullanıcıları gerçekleştirmeleri gereken görevlerden güvenilir bir şekilde haberdar etme yeteneği ile ilişkili olduğunu aktarmaktadır.(Padayachee, 2012; Whitten and Tygar,1999).

## 2.2. Saldırı Tespit Sistemleri

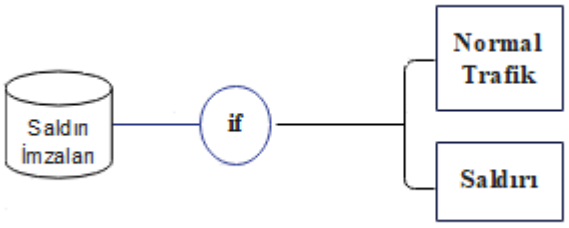
İzinsiz giriş, bilgi sistem cihazlarına yetkisiz bir şekilde erişim sağlanan her türlü aktivite olarak tanımlanabilir. Bu durum, bilgilerin gizliliğine, bütünlüğüne veya kullanılabilirliğine zarar verebilecek her türlü saldırıyı kapsar. Örneğin, sunucuların normal kullanıcılara yanıt vermemesine neden olan faaliyetler izinsiz giriş olarak kabul edilir. İzinsiz girişler sistemleri, bilgisayar sistemlerinde kötü amaçlı eylemleri gerçekleştirmek üzere tasarlanan yazılım veya donanımdan oluşan sistemlerdir (Liao et al., 2013). Saldırı tespit sistemleri, geleneksel bir güvenlik duvarının tanımlayamadığı çeşitli kötü amaçlı ağ trafiği ve bilgisayar kullanımını belirleyerek, bilgisayar sistemlerinin kullanılabilirliği, bütünlüğü veya gizliliğini tehlikeye atabilen eylemlere karşı koruma sağlar. Saldırı tespit sistemleri imza tabanlı ve anomali tabanlı olarak iki gruba ayrılabilir. İmza tabanlı saldırı tespit sistemleri, gelen trafiği önceden bilinen saldırı kalıpları ve imzalarla karşılaştırırken, anomali tabanlı saldırı tespit sistemleri ise, izinsiz giriş yapanların davranışlarının yasal bir kullanıcıdan farklı olduğu varsayımına dayanmaktadır (Khraisat et al., 2019; Stallings, 2006).

### 2.2.1. İmza Tabanlı Saldırı Tespit Sistemleri

İmza tabanlı saldırı tespit sistemlerinin temel amacı, önceden bilinen saldırıların imzaların kullanarak mevcut ağ trafiğini izlemek ve eşleşen bir imza bulunursa alarm sinyali üretmektir. İmza tabanlı saldırı tespit sistemleri genellikle önceden bilinen saldırıları algılamak için kullanılır ve doğruluk oranı yüksektir.

Ancak, yeni veya bilinmeyen saldırıları tespit etmekte başarısızdırlar. Bu durumun nedeni veri tabanında saldırı ile eşleşen imzanın olmamasından dolayı alarm sinyalinin tetiklenememesidir. İmza tabanlı saldırı tespit sistemleri için kullanılan veriler genellikle sunucuların loglarıdır. Bu loglar, daha önce kötü amaçlı yazılım olarak tanımlanmış komut dizilerini veya eylemleri içerir ve imza tabanlı saldırı tespit sistemlerinin veri tabanını oluşturmak için kullanılır. İmza tabanlı saldırı tespit sistemleri, bilgi tabanlı tespit veya kötüye kullanım tespiti olarak da adlandırılabilir. (Modi et al., 2013).

Sistem, daha önceden tespit edilen saldırıların imzalarını veri tabanında saklar ve mevcut ağ trafiklerini bu imzalarla karşılaştırır. Eşleşme bulunursa, bir alarm sinyali tetiklenir. Örneğin, if (kaynak IP adresi=hedef IP adresi) saldırı olarak etiketlenir. Bu örnekte, kaynak IP adresi ile hedef IP adresi eşleştiğinde, sistem bunu bir saldırı olarak etiketler ve alarm sinyali üretir. (Kreibich and Crowcroft, 2004). Snort ve NetSTAT gibi çok sayıda yaygın araç, imza tabanlı saldırı tespit teknolojisi kullanarak network trafiği üzerinde saldırıları tespit etmekte kullanılır. İmza tabanlı saldırı tespit sistemlerinin yapısı aşağıdaki Şekil 2.2 ‘de gösterilmektedir.




**Şekil 2.2.** İmza Tabanlı Saldırı Tespit Sistemi

İmza tabanlı saldırı tespit sistemlerine yönelik olarak imza oluşturma için, genellikle sonlu durum makineleri, biçimsel dil dizisi kalıpları veya semantik koşullar kullanılır. (Meiners et al., 2010)

Sıfır gün saldırılarının artması nedeniyle, imza tabanlı saldırı tespit sistemlerinin teknikleri giderek daha az etkili hale gelmektedir. Polimorfik kötü amaçlı yazılım varyantlarının artması, bu geleneksel paradigmanın yeterliliğini daha da azaltmaktadır. Bu sorunların çözümü olarak anomali ağ trafiğinin tespitinden ziyade normal davranışları belirleyip, normal davranışlardan sapmaları saldırı olarak değerlendiren anomali tabanlı saldırı tespit sistemlerinin kullanılması önem kazanmıştır.

### 2.2.2. Anomali Tabanlı Saldırı Tespit Sistemleri

Anomali tabanlı saldırı tespit sistemleri, bir bilgisayar sisteminin olağan davranışının bir modelidir. Makine öğrenimi, istatistiksel tabanlı veya bilgiye dayalı yöntemler kullanılarak tasarlanır. (Arivarasan and Obaidat, 2022).

Gözlemlenen ağ trafiği ile model arasındaki herhangi önemli bir sapma, saldırı olarak yorumlanarak anomali trafik şeklinde kabul edilir. Bu teknik için varsayım, kötü niyetli davranışın tipik kullanıcı davranışından farklı olduğudur. Anomali tabanlı saldırı tespit sistemleri anomali davranışları saldırı olarak sınıflandırmak için, eğitim ve test aşamalarından oluşur. Eğitim aşamasında, normal davranış modeli öğretilirken, test aşamasında sistemin yeni ve görülmemiş saldırıları tespit etme başarısı test edilir. Eğitim aşaması için kullanılan yönteme dayalı olarak makine öğrenimi temelli, bilgiye dayalı veya istatistiksel tabanlı olarak sınıflandırılabilir (Butun et al., 2014).

Anomali tabanlı saldırı tespit sistemleri, imza tabanlı saldırı tespit sistemleri gibi geleneksel güvenlik algılama sistemlerine göre önemli avantajları vardır. Bunların başında sıfır gün saldırılarının belirlenmesi yeteneği gelir. Sıfır gün saldırıları, kötü amaçlı yazılım veya saldırılar için henüz bilinmeyen imzalara sahip olur ve imza tabanlı saldırı tespit sistemleri tarafından tanınamazlar. Anomali tabanlı saldırı tespit sistemleri, anomali kullanıcı etkinliğini tanımak için imza veri tabanına ihtiyaç duymaz. Bunun yerine, normal davranışın bir modelini oluşturur ve gözlemlenen ağ trafiği ile model arasındaki herhangi bir önemli sapmayı saldırı olarak yorumlayabilir. Bu anomali tabanlı saldırı tespit sistemlerinin sıfır gün saldırılarını belirlemedeki yeteneğini gösterir.(Alazab et al. 2012)

Anomali tabanlı saldırı tespit sistemleri, incelenen davranışın olağan davranıştan farklı olduğunda bir tehlike sinyali tetiklemesi anomali tabanlı saldırı tespit sistemlerinin dahili kötü amaçlı etkinlikleri keşfetme yeteneğine de sahip olduğunu gösterir. Bir saldırgan, çalıntı bir hesapta tipik kullanıcı etkinliğinde tanımlanamayan işlemler yapmaya başlarsa bir alarm oluşur. Tablo 2.1'de imza tabanlı STS ile anomali tabanlı STS arasındaki farkları gösterilmiştir. İmza tabanlı algılama sadece imzası bilinen saldırıları tespit edebilirken, anomali tabanlı algılama sıfır gün saldırılarını da tespit edebilir. Anomali tabanlı saldırı tespit sistemlerinin bu avantajlarının yanı sıra, yüksek oranda normal trafiğin anomali trafik olarak tespit etmeye neden olabilir çünkü tespit edilen anormallikler saldırı değil yeni normal ağ trafiği olabileceğinden yanlış tespit de bulunulmasına neden olabilir. Bu durum, anomali tabanlı saldırı tespit sistemleri için bir taksonomi eksikliği olduğu

belirtilmiştir (Khraisat et al. 2019).

ASTS ve İSTS sistemlerinin avantajları ve dezavantajları aşağıdaki tabloda gösterilmektedir.

**Tablo 2.1.** ASTS ile İSTS Karşılaştırması

| | **Avantaj** | **Dezavantaj** |
|---|---|---|
| **İSTS** | Minimum yanlış alarm, izinsiz girişleri belirlemede etkilidir. | İmzaların sık sık güncellenmesi gerekir. |
| | Bilinen saldırıları tespit etmede başarılıdır. | Sıfırıncı gün saldırılarını tespit edemez. |
| | Tasarlanması basittir. | Çok adımlı saldırıları tespit etmek için uygun değildir. |
| **ASTS** | Yeni Saldırıları tespit etmek için kullanılabilir. | Yüksek oranda normal trafiğin saldırı olarak değerlendirilmesi. |
| | İzinsiz giriş imzası oluşturmak için kullanılabilir. | Eğitime ihtiyaç duyması. |

### 2.2.2.1. Anomali tabanlı saldırı tespit sistemleri çeşitleri

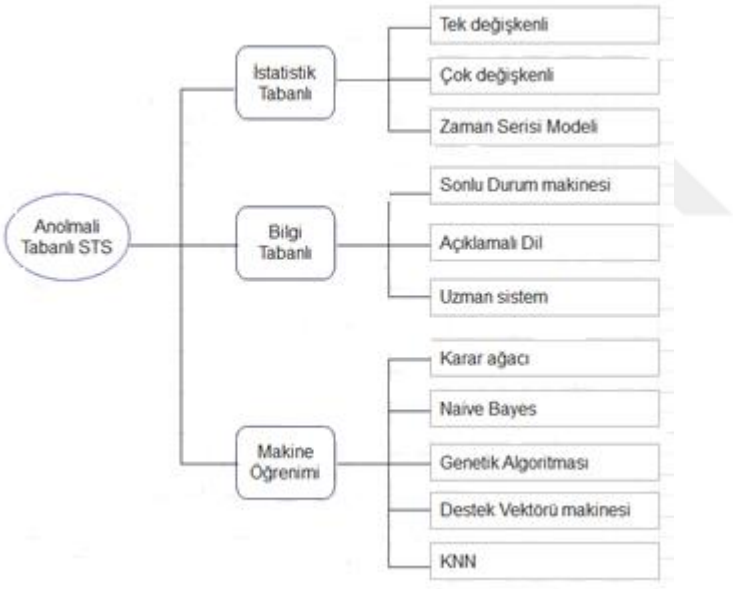
Anomali tabanlı STS'ler ağ tabanlı ve sunucu tabanlı olmak üzere iki ana kategoride incelenebilir. Sunucu tabanlı STS'ler, işletim sistemi, sunucusu günlükleri, güvenlik duvarı günlükleri, uygulama sistemi denetimleri veya veri tabanı günlükleri gibi sunucu sisteminden ve denetim kaynaklarından elde edilen verileri inceler. Bu veriler, sistemdeki anomali aktiviteleri belirlemek için kullanılır. Ağ trafiğini içermeyen içeriden saldırıları tespit etme yeteneğine sahiptir. Örneğin, bir kullanıcının yasadışı bir şekilde bir hesaba giriş yaptığı veya bir kötü amaçlı yazılımın sisteme yüklendiği gibi durumları tespit edebilir. Ağ tabanlı STS, ise ağ trafiğinden kaynaklanan verileri inceler. (Creech and Hu, 2014a)(Creech and Computers, 2013).

Ağ tabanlı STS'ler, ağ trafiğini izlemek için kullanılan bir sistemdir. Bu sistem, paket yakalama, NetFlow ve diğer ağ veri kaynakları aracılığıyla ağdan çıkarılan verileri izleyerek, harici bir tehditten başlatılabilecek harici kötü amaçlı faaliyetleri tespit etmeye çalışır. Ağ tabanlı STS'ler, harici bir tehditten başlatılabilecek saldırıları izleyebilir. Ancak, modern yüksek hızlı ağların büyük veri hacmi nedeniyle, tüm verileri denetleme yeteneği sınırlıdır (Bhuyan et al., 2014). Ağ tabanlı STS ve sunucu tabanlı STS, Şekil 2.3'te gösterilen belirli bir ağ topolojisi içinde çeşitli konumlara dağıtılarak hem harici hem de içeriden gelen saldırıları tespit etmek amacıyla kullanılabilirler.


**Şekil 2.3.** SSTS ve ASTS ağ topolojisinde dağılım örneği

**Anomali Tabanlı Saldırı Tespit Sistemlerinin Uygulama Teknikleri**

Anomali tabanlı STS yöntemleri Şekil 2.4'te görüldüğü gibi istatistik temelli, bilgi temelli ve makine öğrenimi temelli olmak üzere üç farklı tipte sınıflandırılabilir.




**Şekil 2.4.** Anomali Tabanlı Saldırı Tespit Sistemi

- İstatistik temelli yaklaşım, bir trafikteki her veri kaydını toplayıp incelemeyi ve normal kullanıcı davranışının istatistiksel bir modelini oluşturmayı içerir. Bu yöntem, veri setlerinden istatistikleri ve normal dağılımları çıkartarak, anomali verileri tespit etmeye çalışır.
- Bilgi temelli yaklaşım, protokol özellikleri, ağ trafiği örnekleri ve diğer mevcut sistem verilerinden istenen eylemleri belirlemeye çalışır. Bu yöntemler bilgiye

dayalı karar alma mekanizmalarını kullanarak, anomali trafiği tespit etmeye çalışır.

- Makine öğrenimi temelli yaklaşım, eğitim verilerinden karmaşık model eşleştirme yeteneklerini elde eder. Bu yöntemler, veri setlerinden eğitim aşamasında öğrenerek, anomali trafiğin tespit edilmesinde kullanılır.

Her yöntem kendine özgü avantajları ve dezavantajları vardır ve saldırıları tespit etmekte kullanılabilir. Ancak, hangi yöntemin kullanılması gerektiği, sistemlerin özelliklerine, veri kaynaklarına ve sistem amacına göre değişebilir.

**İstatistik Tabanlı Saldırı Tespit Sistemleri**

İstatistik tabanlı bir STS, normal davranış profilini oluşturmak için bir dağıtım modeli kullanır. Bu verilerin medyan, ortalama ve standart sapması gibi istatistiksel ölçümlerini hesaba katar. Daha sonra, düşük olasılıklı olayları algılar ve bunları izinsiz girişler olarak işaretler. İstatistiksel tabanlı ASTS, veri trafiğini denetlemek yerine paketlerin parmak izini gösteren her paketi izler. Bu teknik, normal davranıştan mevcut davranıştaki herhangi bir farklılığı belirlemek için kullanılır.

**Tek değişkenli:** Bu teknik sadece bir değişkeni izler ve o değişkenin normal davranışını modellemek için bir istatistiksel profil oluşturur. Tek değişkenli STS, her bir ölçümdeki anomali trafiği arar. Bu yöntem sadece tek değişkenli veriler için kullanılabilir.(Ye et al. 2002).

**Çok değişkenli:** Çok değişkenli modeller, iki veya daha fazla ölçüm arasındaki ilişkileri anlamak için kullanılır. Bu modeller, deneysel verileri ayrı ayrı analiz etmek yerine, ilişkili ölçümlerin kombinasyonlarını kullanarak daha iyi sınıflandırma elde etmek için değerlidir. Örneğin, Ye et al. (2002) yılında yaptıkları çalışmada, normal faaliyetlerin uzun vadeli bir profilini oluşturarak izinsiz girişleri tespit etmek için çok değişkenli bir kalite kontrol yöntemi inceleme yapmışlardır. Ancak, çok değişkenli istatistiksel modeller için temel bir zorluk, yüksek boyutlu veriler için dağılımları tahmin etmenin zor olmasıdır (Ye et al. 2002).

**Zaman serisi modeli:** Zaman serisi modelleri, zaman içinde toplanan verileri tahmin etmek için kullanılır. İzinsiz giriş tespitinde, zaman serisi verileri ağ etkinliğinde anormal davranışları belirlemek için kullanılır (Viinikka et al., 2009).

**Bilgi Tabanlı Saldırı Tespit Sistemleri**

Uzman sistem yöntemi olarak adlandırılan bu teknik, normal trafik profilini yansıtan bir bilgi tabanı oluşturulmasını gerektirir. Bu profil, standart bir profilden

farklı olan eylemleri izinsiz giriş olarak değerlendirir. Bu yöntem, diğer ASTS sınıflarından farklı olarak, normal sistem etkinliğini tanımlanırken insan bilgisi temel alınarak oluşturulur (Khraisat et al., 2019).

Bilgiye dayalı tekniklerin ana faydası, sistemde mevcut olan normal davranışlar hakkında bilgiye sahip olmasıdır. Bu sayede, yanlış pozitif alarmlar azaltılır. Bununla birlikte, dinamik olarak değişen bir bilgi işlem ortamında, bu tür bir ASTS'nin tüm normal davranışlar hakkında bilgi toplamak çok zor olduğu için, beklenen normal davranış için düzenli olarak bilgi güncellemesi gerektirir. Bu güncellemeler, zaman alıcı ve zor bir görev olabilir (Khraisat et al., 2019).

Sonlu Durum Makinesi (SDM) bir hesaplama modelidir ve yürütme akışını temsil etmek ve kontrol etmek için kullanılır. Bu model, bir saldırı tespit sistemi modeli oluşturmak için izinsiz giriş tespitinde uygulanabilir. SDM, durumlar, geçişler ve faaliyetler şeklinde temsil edilir. Durumlar geçmiş verileri kontrol eder. Örneğin, girdideki herhangi bir varyasyon tespit edilir ve tespit edilen varyasyona göre geçiş gerçekleşir (Walkinshaw et al., 2016). Bir SDM normal sistem davranışını temsil edebilir ve bu SDM 'den gözlemlenen herhangi bir sapma bir saldırı olarak kabul edilir.

Açıklama Dili, tanımlanmış bir saldırının özelliklerini belirtmek için kullanılabilecek kuralların sözdizimini tanımlar. Bu kurallar, N-dilbilgisi ve UML gibi açıklama dilleriyle oluşturulabilir. Bu diller, saldırı özelliklerini tanımlamak için kullanılan ifadelerin söz dizimi tanımlar ve bu sayede saldırı tespit sistemi kurallarını yazmak daha kolay hale gelir (Studnia et al., 2018)

Uzman Sistem, bir dizi kurala dayalı bir sistemdir. Bu kurallar genellikle bir alan uzmanı tarafından tanımlanır ve saldırıları tanımlayan bir dizi kural oluşur. Bir uzman sistemde, kurallar genellikle bir alan uzmanıyla iş birliği içinde çalışan bir mühendis tarafından manuel olarak tanımlanır. Bu kurallar, sistemde mevcut olan normal davranışları tanımlar ve anormal davranışları saldırı olarak tespit etmek için kullanılır. Uzman sistemler, özel bir alan veya uygulama için öğrenilmiş özel bilgi ve kuralları kullanarak çalışırlar. (Kim et al., 2014)

**Makine Öğrenimi tabanlı Saldırı Tespit Sistemleri**

Makine öğrenimi, bilgisayar sistemlerinin belirli bir görevi gerçekleştirmek için açıkça programlanmadığı, bunun yerine algoritmaların ve istatistiksel modellerin bilimsel olarak incelendiği bir disiplindir. Günlük olarak kullandığımız birçok uygulamada öğrenme algoritmaları ile tasarlanmıştır. Örneğin google web arama

motorunun internette arama yapmak için her kullanıldığında web sayfalarını nasıl sıralayacağını öğrenen bir öğrenme algoritmasıdır. Bu algoritmalar, veri madenciliği, görüntü işleme teknikleri, tahmine dayalı analitik ve benzeri çeşitli amaçlar için kullanılır. Makine öğreniminin temel avantajı, bir algoritmanın veri ile ne yapacağını öğrendiğinde, ilgili görevi otomatik olarak gerçekleştirebilmesidir. (Mahesh, 2018).

Makine öğrenimi yöntemine dayalı STS'nin temel odak noktası, kalıpları tespit etmek ve veri kümesine dayalı saldırı tespit sistemi oluşturmaktır. Genel olarak, makine öğrenimi yöntemleri denetimli ve denetimsiz olarak ikiye ayrılır. Denetimli yöntemlerde, eğitim verileri etiketlenmiş olarak sağlanır ve sistem sadece mevcut etiketli verileri kullanarak öğrenir. Denetimsiz yöntemlerde ise etiketlenmiş veri yoktur ve sistem kendisi kalıpları tespit etmek için verileri analiz eder.

Makine öğrenimi, geniş kapsamlı uygulamalarla bilgisayar biliminin en hızlı büyüyen alanlarından biridir. Verilerdeki anlamlı kalıpların otomatik olarak çıkarılmasını ifade eder. Makine öğrenimi araçları, programlara öğrenme ve uyum sağlama yeteneği kazandırmaktır (Shai, 2014).

Makine öğrenimi modelleri, veri setinden davranışları tanımlamak veya tahmin etmek için bir dizi kural ve yöntemlerden oluşur. Bu modeller, öğrenme sürecinde veriye dayanarak kendilerini optimize eder ve veriye dayalı karar vermek için kullanılır. (Dua and Du 2016).

Makine öğrenimi teknikleri, anomali tabanlı saldırı tespit sistemleri alanında geniş bir şekilde uygulanmıştır. Bu teknikler arasında kümeleme, sinir ağları, ilişkilendirme kuralları, karar ağaçları, genetik algoritmalar ve en yakın komşu yöntemleri gibi çeşitli algoritmalar ve teknikler bulunmaktadır. Algoritmalar ve teknikler, izinsiz giriş veri kümelerinden bilgi keşfetmek ve sistemlerin güvenliğini sağlamak için kullanılabilir.(Kshetri and Voas 2017; Xiao et al., 2018).

Daha önceki bazı araştırmalar, anomali tabanlı saldırı tespit sistemleri oluşturmak için farklı tekniklerin kullanımını incelemiştir. Örneğin, Chebrolu ve arkadaşları, Bayes Ağları (BN) ve Sınıflandırma Regresyon Ağaçları (CRC) içeren iki özellik seçme algoritmasının performansını incelemişlerdir. Bu çalışmalar, bu yöntemlerin birleştirilmesi ile daha yüksek doğruluk elde edebileceğini göstermiştir.(Chebrolu et al., 2005).

Bajaj ve arkadaşları, Bilgi Kazanımı (IG) ve Korelasyon Niteliği değerlendirmesi gibi özellik seçme algoritmalarının bir kombinasyonunu kullanan özellik seçimi için bir teknik önermişlerdir. Bu çalışmalar seçilen özelliklerin

performansını C4.5, Naive Bayes, NB-Tree ve Multi-Layer Perceptron gibi farklı sınıflandırma algoritmaları uygulayarak test etmişlerdir (Khraisat et al., 2018; Bajaj and Arora, 2013). STS özelliklerinin önemini değerlendirmek için genetik-bulanık kural madenciliği yöntemi kullanılmıştır (Elhag et al., 2015). Thaseen ve Kumar (2013) doğruluğu artırmak ve yanlış alarm oranını azaltmak için Rastgele Ağaç modelini kullanarak imza tabanlı saldırı tespit sistemi önermişlerdir(Thaseen and Kumar, 2013). Subramanian et al. (2012), metrik verilerine göre bir model oluşturmak için karar ağacı algoritmalarını kullanarak NSL-KDD veri setini sınıflandırmayı ve karar ağacı algoritmalarının performansını incelemeyi önermişlerdir (Subramanian et al.,2012). Bu çalışmalar, farklı tekniklerin kullanımının saldırı tespit sistemleri için önemli olduğunu ve bu tekniklerin bir arada kullanılmasının daha iyi sonuçlar elde edilmektedir.

Makine öğrenimi tekniklerine dayalı olarak çeşitli saldırı tespit sistemi oluşturulmuştur. Bu tekniklerin amacı, gelişmiş doğruluk ve az insan bilgisi gerektirerek STS oluşturmaktır.

### 2.3. Gerçekleştirilen Çalışmalar

Narudin et al. (2016), MalGenome veri setini kullanarak kötü amaçlı yazılımları tespit etmek için Rasgele Orman Algoritması, J-48, Çok katmanlı algılayıcı, Naive Bayes ve k en yakın komşu algoritmalarını kullanarak değerlendirmesini açıkladı. DP, YP, kesinlik, geri çağırma ve F-measure gibi performans metrikleri, makine öğrenimi algoritmalarının performansını doğrulamak için kullanıldı. MalGenome veri setinde yapılan deneysel çalışmalarda Rasgele Orman sınıflandırması kullanılarak elde edilen doğruluk %99,99'dur. Yazar, gelecekteki çalışmalarında sonuçları iyileştirmek için özellik seçme yöntemlerinin kullanılmasını önermiştir (Narudin et al.,2016).

Belavagi and Muniyal (2016), çeşitli denetimli makine öğrenimi sınıflandırıcıları ile bir imza tabanlı saldırı tespit sistemi tasarladı. Çeşitli sınıflandırıcıların performansını kontrol etmek için NSL-KDD veri seti kullanıldı. Sonuç, Rasgele Orman sınıflandırıcının diğer sınıflandırıcılardan daha iyi performans gösterdiğini göstermektedir. En düşük Yanlış pozitif oran ve en yüksek doğru pozitif oran ile sonuçlanır ve elde edilen doğruluk %99'dur. Ancak yine de çok sınıflı sınıflandırma için kullanılabilecek sınıflandırıcılara ihtiyaç vardır (Belavagi and Muniyal, 2016).

Ashfaq et al. (2017), bulanıklığa dayalı bir yarı denetimli öğrenme yaklaşımını tanımlamıştır. Sınıflandırıcı performansını iyileştirmek için, denetimli bir öğrenme algoritmasıyla etiketlenmemiş örnekleri kullanır. Bu modelin değerlendirilmesinde NSL-KDD veri seti kullanılmıştır. Bu modelin sınırlaması, performansının yalnızca ikili sınıflandırma görevi için çalışılmasıydı (Ashfaq et al., 2017).

Yaseen et al. (2017), DVM ve EVM kullanan çok düzeyli bir hibrit izinsiz giriş tespit modelini tanımlamıştır. Değerlendirme KDD 99 veri seti üzerinde yapılmıştır. Önerilen bu modelde elde edilen doğruluk %95,75 olmuştur. Bu teknik yalnızca bilinen saldırılar için doğruluk oranı yüksektir. Yeni saldırılar için verimli sınıflandırıcılar gereklidir (Yaseen et al., 2017).

Aljumah (2017), Yapay Sinir Ağı'na (YSA) dayalı DDoS saldırılarını tespit etmek için eğitilmiş bir algoritma tanımlamıştır. Tasarlanan YSA, eski veri kümeleri ile eğitildiğinde %92, güncellenmiş veri kümeleri ile eğitildiğinde ise %98 doğruluk göstermiştir. YSA modelinin doğruluğu veri setine bağlıdır. Bu nedenle güncel ve dengeli bir veri setine ihtiyaç vardır (Aljumah 2017).

Roshan et al. (2018), Extreme Learning Machines'e (ELM) dayalı uyarlanabilir bir STS tasarımını tartıştı. Değerlendirme için NSL-KDD veri seti uygulandı. Yeni saldırıları ve bilinen saldırıları kabul edilebilir bir tespit oranı ve yanlış pozitiflerle tespit edebildiği bulundu (Roshan et al. 2018).

Literatür taraması, araştırmaların çoğunun daha eski veri kümeleri kullanılarak doğrulandığı sonucuna varılmıştır. Bu veri kümelerinde yeni saldırılar yoktur ve dengesiz ağ denetim verileri içerir. Dolayısı ile bu tez çalışmasında daha güncel olan UNSW-NB15 veri seti kullanılmıştır.

# 3. GEREÇ VE YÖNTEM

Anomali tabanlı saldırı tespiti, günümüzdeki karmaşık ve sıfır gün siber tehditlerle başa çıkabilmek için önemli bir araştırma alanı olmuştur. Bu nedenle tez çalışmasında, makine öğrenimi algoritmaları kullanılarak anomali tabanlı saldırı tespit sistemi geliştirilmiştir. Bu bölümde, gereç ve yöntemlerimizi detaylandırarak, hedeflenen başarı kriterlerine ulaşmak için kullanılan teknikler açıklanmaktadır.

## 3.1. Kullanılan Yöntemler

Anomali tabanlı saldırı tespit sistemi tasarlanırken makine öğrenimi sınıflandırıcıları kullanılmıştır. Bu sınıflandırıcılar, belirli bir veri kümesinden öğrenme süreciyle ağ trafiklerinin otomatik olarak sınıflandırılması amaçlanmıştır.

### 3.1.1. Naive Bayes

Naive Bayes, Bayes teoremine göre tanımlanan bir sınıflandırma algoritmasıdır. Bu sınıflandırıcı, verilen bir sınıf değerine ait her özelliğin olasılığının diğer özelliklerden bağımsız olduğunu varsayımına dayanır. Tahmin, her sınıfın örnek olasılıkları hesaplanarak, en yüksek olasılığa sahip sınıf değeri seçilerek elde edilebilir.

Sistem değişkenleri arasındaki istatistiksel bağımlılıkları veya nedensel ilişkileri bildiğimiz birçok durum vardır. Ancak, bu değişkenler arasındaki olasılıksal ilişkileri tam olarak ifade etmek zor olabilir. Başka bir deyişle, sistem hakkındaki ön bilgi, basitçe bazı değişkenlerin diğerlerini etkileyebileceğidir. Bir problemin rastgele değişkenleri arasındaki bu yapısal ilişkiden veya nedensel bağımlılıklardan yararlanmak için Naive Bayes adı verilen olasılıksal bir grafik modeli kullanılabilir (Tsai et al. 2009).

Naive Bayes, "Gözlenen sistem faaliyetleri göz önüne alındığında, belirli bir tür saldırının meydana gelme olasılığı nedir?" gibi soruları koşullu olasılık formüllerini uygulayarak yanıtlar. Naive Bayes, saldırılarda ve normal davranışlarda meydana gelme olasılıkları farklı olan özelliklere ele alır. Naive Bayes sınıflandırma modeli, kullanım kolaylığı ve hesaplama etkinliği nedeniyle STS'de kullanılan en yaygın modellerden biridir (Yang and Tian, 2012). Ancak, karmaşık öznitelik bağımlılıklarına

sahip KDD’99 veri setinde olduğu gibi bağımsızlık varsayımı geçerli değilse sistem iyi çalışmaz. Yapılan araştırmalar sonucunda Naive Bayes modelinin büyük veri kümeleri için doğruluğu azalttığını da ortaya koymaktadır. Başka bir çalışmada gelişmiş Yüksek Naive Bayes modelinin yüksek boyutluluk, son derece birbiriyle ilişkili nitelikler ve yüksek hızlı ağlar içeren ortamlarda STS görevlerine uygulanabileceğini göstermiştir(Koc et al. 2012).

Naive bayes algoritması 3.1’de formülüze edilmiştir.

$$P(A\backslash B) = \frac{P(B\backslash A) * P(A)}{P(B)} \quad \textbf{(3.1)}$$

Bu formüle göre A sınıfının bilinmesi durumunda B değişkeninin var olma olasılığını ve B değişkeninin bilinmesi durumunda A olma olasılığıdır. Bu formüle göre en yüksek olasılık hangi sınıfa aitse o sınıf karar sınıfıdır (Solmaz ve ark., 2014).

### 3.1.2. Lojistik Regresyon Sınıflandırıcısı

Son yıllarda derin öğrenme hızla gelişmiştir ancak derin öğrenme büyük miktarda veri seti gerektirir ve bilgisayarlar için yüksek donanım gereksinimlerine sahiptir. Bazı basit ikili sınıflandırma problemleri için, makine öğrenimi yöntemleri kullanılabilir, bu nedenle makine öğrenimi hala önemli araştırma değerine sahiptir. Bunlar arasında Lojistik regresyon, ikili sınıflandırma ve tahminler açısından geniş bir veri işleme yöntemi olarak yaygın bir şekilde kullanılmaktadır (Rongheng S. 2014 ).

Lojistik regresyon, istatistiksel sınıflandırma alanında öne çıkan bir yaklaşımdır. Özellikle ikili sınıflandırma problemlerinde yaygın olarak kullanılan bu yöntem, veri noktalarını iki farklı kategoriye ayırmayı amaçlar. Her bir girdi özelliği, formül 3.2'de belirtildiği gibi, özgül ağırlıklarla çarpılarak toplanır ve böylece bir z-değeri hesaplanır. Ardından, bu z-değeri sigmoid fonksiyonundan geçirilir ve sonuç olarak bir değer elde edilir. Eğer bu değer 0.5'ten küçükse, sonuç 0 olarak sınıflandırılırken, değer 0.5'ten büyükse 1 olarak sınıflandırılır. Bu şekilde, lojistik regresyon modeli, girdi özellikleri ile sınıf etiketleri arasındaki ilişkiyi yakalayarak veri noktalarını ilgili sınıflara doğru şekilde tahmin eder.

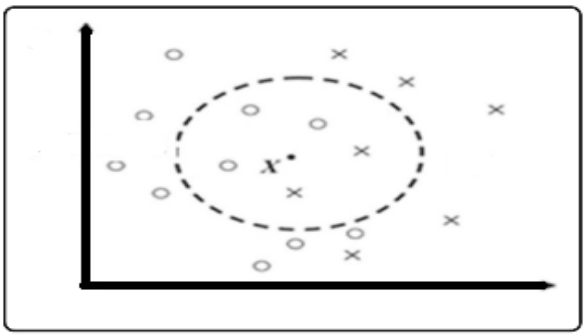
$$Z = W_0X_0 + W_1X_1 + W_2X_2 + \cdots + W_nX_n \quad \textbf{(3.2)}$$

### 3.1.3. Destek Vektör Makineleri (DVM)

Destek vektör makineleri (DVM), Vapnik (1998) tarafından önerilmiştir. DVM, doğru veya bir hiper düzlem tarafından ayırt edici bir sınıflandırıcıdır. DVM'ler, izinsiz girişin doğru olarak sınıflandırıla bilmesi için eğitim verilerini daha yüksek boyutlu bir alana eşlemek için bir çekirdek işlevi kullanır. DVM'ler genelleştirme yetenekleriyle iyi bilinirler ve özniteliklerin sayısı büyük ve veri noktalarının sayısı az olduğunda esas olarak değerlidirler. Doğrusal, polinom, Gauss Radyal Temel Fonksiyonu (RBF) veya hiperbolik tanjant gibi bir çekirdek uygulanarak farklı türde ayırıcı hiper düzlemler elde edilebilir. STS veri setlerinde, veri noktalarının doğru sınıflara ayrılmasında birçok özellik gereksizdir veya daha az etkilidir. Bu nedenle, DVM eğitimi sırasında özellik seçimi dikkate alınmalıdır. DVM, birden çok sınıfa sınıflandırmak için de kullanılabilir. Li et al. (2012) tarafından yapılan çalışmada, KDD 1999 veri setini önceden tanımlanmış sınıflara sınıflandırmak için RBF çekirdeğine sahip bir DVM sınıflandırıcısı uygulanmıştır. (Li et al. 2012).

### 3.1.4. K En Yakın Komşu Algoritması

K-en yakın komşu (k-NN), örnekleri sınıflandırmak için en basit ve geleneksel parametrik olmayan tekniklerden biridir (Bishop 1995; Manocha and Girolami, 2007). K-NN sınıflandırıcı oluşturma sürecinde K önemli bir parametredir. Farklı K değerleri farklı performanslara neden olacaktır. Eğer K oldukça büyük seçilirse tahmin için kullanılan komşuların sayısı artmasından dolayı zamanı ve tahminin doğruluğunu etkileyecektir. Bu tekniklerin fikri, etiketlenmemiş bir veri örneğini k en yakın komşularının sınıfına etiketlemektir. Şekil 3.1'de k = 5 olan bir K-En yakın komşu algoritması göstermektedir. X noktası, sınıflandırılması gereken etiketlenmemiş verinin bir örneğini temsil eder. X'in en yakın beş komşusu arasında saldırı sınıfından üç ve normal sınıftan iki veri vardır. Çoğunluk, X'in saldırı sınıfına atanmasını sağlar. k-NN, çoğu STS'de iyi bir sınıflandırma performansı sağladığından, diğer tüm sınıflandırıcılar için bir kıyaslama noktası olarak uygun bir şekilde uygulanabilir(Lin et al. 2012.) .



**Şekil 3.1.** K-NN Sınıflandırma Örneği

### 3.1.5. Karar Ağacı

Bir karar ağacı, mevcut kararın sonraki kararın alınmasına yardımcı olduğu bir dizi karar yoluyla bir örneği sınıflandırılmasıdır. Böyle bir karar dizisi bir ağaç yapısında temsil edilir. Bir örneğin sınıflandırılması, kök düğümden, her bir uç yaprak düğümün bir sınıflandırma kategorisini temsil ettiği uygun bir uç yaprak düğümüne doğru ilerler. Örneklerin öznitelikleri her düğüme atanır ve her dalın değeri özniteliklere karşılık gelir (Mitchell, 1997).

Karar ağacı oluşturmak için iyi bilinen algoritma CART'tır (Breiman et al. 1984). CART, karar ağacı oluşturmada kullanılan bir algoritmadır. İkili ağaç yapısından oluşur. Yani ana düğümden iki yavru düğüm oluşur. Homojen bir ağaç elde edilmeyi amaçlar. CART algoritması, veri setini daha homojen alt kümeler halinde bölmek için Gini katsayısını kullanır. Gini katsayısı, bir veri kümesinin karışıklığını veya belirsizliğini ölçer bu değer ne kadar düşükse, o kadar iyi bir bölünme olduğunu ifade eder. Her adımda, algoritma en küçük Gini katsayısına sahip bir özelliği seçer ve bu özelliği kullanarak veriyi böler.

Bu yaklaşım, özellikle sınıflandırma problemlerinde, verilerdeki sınıflar arasındaki ayrımı en iyi şekilde belirlemek için etkilidir. Her bir düğüm, yalnızca iki dalı olan bir ikili ağaç yapısını sürdürür. Bu nedenle, nihai karar ağacı yapısı, kısa, öz ve anlaşılır bir formda olur.

Gini katsayısının hesaplanması için 3.3'te verilen formül kullanılır:

$$Gini = 1 - \sum_{i=1}^{C} (Pi)^2 \qquad \textbf{(3.3)}$$

Burada, Pi her sınıfın veri kümesindeki oranını ifade eder. Bu formül, Gini katsayısını hesaplamak için kullanılır ve 0 ile 1 arasında bir değere sahiptir. 0, tam bir

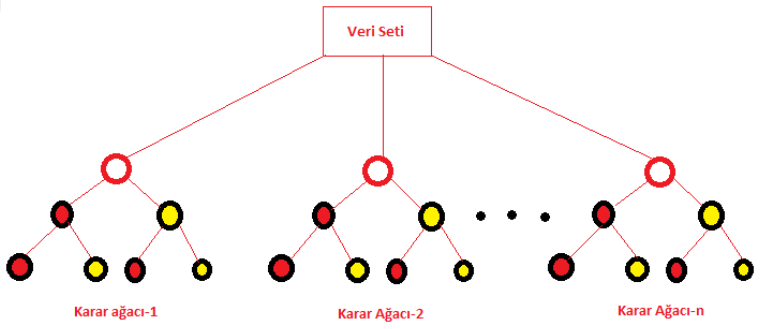
düzeni ifade ederken, 1 tam bir karışıklığı ifade eder.

### 3.1.6. Rastgele Orman Algoritması

Rastgele Orman (Random Forest), bağımsız ve aynı şekilde dağıtılmış rastgele vektörlere sahip bir dizi ağaç yapısından oluşan bir sınıflandırma yöntemidir. Her ağaç, giriş verilerinde bulunan her sınıf için bir oy verir ve en popüler sınıfı seçer. Bu yöntem, genellikle sınıflandırma problemlerini çözmek için kullanılır (Reis vd., 2018). Rastgele Ormanların genelleme hatasını hesaplamak için iki önemli parametre vardır. İlk parametre, her bir ağacın bağımsız olarak eğitildiği ve tahminlerde bulunduğu bağımsız rastgele vektörlerin oluşturulma şeklidir. İkinci parametre ise eğitim ve test verileri kullanılarak her bir ağacın oluşturulmasıdır.

Bu yöntemin genelleme hatasını değerlendirmek için kullanılan üst sınır, sınıflandırıcıların bireysel doğruluğu ve birbirleri arasındaki bağımlılığıyla ilişkilidir. Bu bağımlılık, farklı ağaçların benzer tahminlerde bulunmasına neden olabilir.

Aşağıdaki Şekil 3.2'de, Rastgele Orman yönteminin akış şemasını göstermektedir. Bu yöntem, çeşitli bağımsız ağaçların tahminlerini bir araya getirerek daha güvenilir sonuçlar elde etmek için kullanılır (Özgode Yigin ve diğerleri, 2020).




**Şekil 3.2.** Rastgele Orman Algoritma Örneği

## 3.2. Kullanılan Veri Seti

Avustralya Siber Güvenlik Merkezi'nin siber güvenlik araştırma ekibi, KDDCup 99 ve NSL-KDD veri kümelerinde bulunan sorunları çözmek için UNSW-NB15 adlı yeni bir veri setini kullanıma sunmuştur. Bu veriler, yeni saldırılar ve ortak güvenlik açıkları (CVE) içeren bir havuza sahip olan IXIA Perfect Storm aracı kullanılarak canlı bir ağ trafiğinin normal ve saldırı davranışlarını içeren hibrit bir şekilde oluşturulmuştur. IXIA trafik oluşturma aracında, bir sunucunun normal

etkinlikleri ürettiği, diğer sunucu ise ağda kötü niyetli etkinlikleri ürettiği iki sunucu kullanıldı. Bu veri seti, Tablo 3.1'de sunulan dokuz farklı saldırı türü ve Tablo 3.2'de gösterilen 43 özellikten meydana gelmektedir. Söz konusu veri seti, farklı saldırı türlerine ait örnek verilerin, bu 43 özellik üzerinden temsil edildiği bir yapıya sahiptir. Her bir saldırı türü, belirli özelliklerin birleşiminden oluşan ve sınıf etiketleri olarak kullanılabilen nitelikli kategorik bilgiler içermektedir.

Bu çalışmada, tez projesinin bir parçası olarak geliştirilen anomali tabanlı saldırı tespit sisteminin eğitim ve test aşamaları için toplamda 175,342 eğitim verisi ve 82,333 test verisi kullanılmıştır.

**Tablo 3.1.** Saldırı Türleri

| SIRA NO. | SÜTUN ADI |
|---|---|
| **1** | Fuzzer Saldırıları |
| **2** | Analiz Saldırıları |
| **3** | Arka Kapı(Backdoor) Saldırılıarı |
| **4** | Hizmet Reddi (Denial of Service-DoS) Saldırıları |
| **5** | Exploit (İstismar) Saldırıları |
| **6** | Generic (Genel) Saldırılar |
| **7** | Reconnaissance (Keşif) Saldırıları |
| **8** | Shellcode (Kabuk Kodu) Saldırıları |
| **9** | Worm (Solucan) Saldırıları |

**Tablo 3.2.** Veri Seti Kolon Adı ve Açıklamaları

| SIRA NO. | SÜTÜN ADI | AÇIKLAMA |
|---|---|---|
| **1** | attack_cat | Saldırı Adı |
| **2** | dur | Toplam süre |
| **3** | proto | işlem protokolü |
| **4** | service | http, ftp, ssh, dns … |
| **5** | state | Durum ve bağımlı protokol |
| **6** | spkts | Kaynaktan hedefe paket sayısı |
| **7** | dpkts | Hedeften Kaynağa Paket Sayısı |
| **8** | sbytes | Kaynaktan hedef byte |
| **9** | dbytes | Hedeften Kaynağa byte |
| **10** | rate | Oran |
| **11** | sttl | Kaynaktan Hedefe Geçen Zaman |
| **12** | dttl | Hedeften Kaynağa Geçen Zaman |
| **13** | sload | Kaynak bit/saniye |
| **14** | dload | Hedef bit/saniye |
| **15** | sloss | Kaynak Paketlerinin Durumu |
| **16** | dloss | Hedef Paketlerinin Durumu |
| **17** | sinpkt | Paketler arası kaynak varış süresi |
| **18** | dinpkt | Paketler arası hedef varış süresi |

**Tablo 3.2.** Veri Seti Kolon Adı ve Açıklamaları (devam)

| | | |
|---|---|---|
| **19** | sjit | Kaynak Gecikmesi |
| **20** | djit | Hedef Gecikmesi |
| **21** | swin | Kaynak TCP Protokolü Gösterim Süresi |
| **22** | stcpb | Kaynak TCP Protokolü Sıra Numarası |
| **23** | dtrcpb | Hedef TCP Protokolü Sıra Numarası |
| **24** | dwin | Hedef TCP Protokolü Gösterim Süresi |
| **25** | tcprtt | TCP Protokolünün "synack" ve "acckdat" değerlerinin toplamı. |
| **26** | synack | TCP Protokolünün SYN ve SYN ACK paketleri arasındaki süre. |
| **27** | ackdat | TCP Protokolünün SYN_ACK ve ACK paketleri arasındaki süre |
| **28** | smean | Kaynak Tarafından İletilen Akış Paketi Boyutunun Ortalaması |
| **29** | dmean | Hedef Tara fından İletilen Akış Paketi Boyutunun Ortalaması |
| **30** | trans_depth | HTTP İsteğinin Veri Derinliği |
| **31** | response_body_len | http hizmetinden aktarılan verilerin içerik boyutu. |
| **32** | ct_srv_src | 100 bağlantıda aynı hizmeti ve kaynak adresini içeren bağlantı sayısı |
| **33** | ct_state_ttl | Kaynaktan Hedefe İletim Süresi |
| **34** | ct_dst_ltm | Son 100 bağlantıda aynı hedef adresine bağlantı sayısı. |
| **35** | ct_src_dport_ltm | Son 100 bağlantıda aynı kaynak adres ve hedef port bağlantı sayısı. |
| **36** | ct_dst_sport_ltm | Son 100 bağlantıda aynı Hedef adres ve Kaynak port bağlantı sayısı. |
| **37** | ct_dst_src_ltm | Son 100 bağlantıda aynı kaynak ve hedef adresine ait bağlantı sayısı. |
| **38** | is_ftp_login | Ftp oturumuna kullanıcı Adı ve şifresi ile erişiliyorsa 1 değilse 0. |
| **39** | ct_ftp_cmd | Ftp oturumunda komut akış sayısı |
| **40** | ct_flw_http_mthd | Http hizmetinde Al ve Gönder gibi yöntemlere sahip akış sayısı. |
| **41** | ct_src_ltm | Son 100 bağlantıda aynı kaynak adresin bağlantı sayısı. |
| **42** | ct_srv_dst | Son 100 bağlantıda aynı hizmeti ve hedef adresi içeren bağlantı sayısı. |
| **43** | is_sm_ips_ports | Kaynak ve hedefin IP adresleri ve Port numaraları eşitse, bu değişken 1 değerini alır, aksi takdirde 0 |
| **44** | Label | Eğitim veri setinde normal trafik sayısı 47912, anormal trafik sayısı 127430, Test veri setinde normal trafik sayısı 244, anormal trafik sayısı 82089 |

### 3.3. Geliştirme Ortamı

Python programlama dili, bilimsel bilgi işleme için en popüler dillerden biridir. Yüksek düzeyde etkileşimli doğası ve olgunlaşan bilimsel kütüphaneler ekosistemi sayesinde, algoritmik geliştirme ve keşifsel veri analizi için çekici bir seçimdir (Dubois, 2007; Milmann ve Avaizis, 2011).

Python, son yıllarda bilgi güvenliği ve siber güvenlik alanında popüler bir dil haline gelmiştir. Saldırı tespit sistemleri, siber tehditleri tespit ederek, koruma

mekanizmalarını devreye sokmak ve ağ güvenliğini artırmak amacıyla geliştirilir. Python, bu tür sistemlerin geliştirilmesinde kullanım kolaylığı, geniş kütüphane desteği sağlamaktadır. Bu tez çalışmasında bu nedenlerle Python programlama dili tercih edilmiştir.

Python programlama dilini kullanarak bir uygulama geliştirme ortamı olarak Google Colab seçilmiştir. Google Colab, araştırmacıların ve veri bilimcilerin ücretsiz olarak bulut tabanlı bir ortamda Python programlama dili kullanarak çeşitli projeleri gerçekleştirebildiği, popüler bir platformdur.

Google Colab, Google tarafından sunulan bir hizmet olup, özellikle yapay zekâ, veri analitiği ve makine öğrenmesi gibi veri yoğun işlemler için kullanılır. Colab, Jupyter notebook tabanlı bir platformdur ve Python programlama diliyle uyumludur.

Google Colab, birçok popüler Python kütüphanesini önceden yükleyerek kullanıcıların projelerini hızla başlatmalarına yardımcı olur. Numpy, Pandas, Matplotlib, TensorFlow, PyTorch gibi kütüphaneleri içeren birçok önceden yüklenmiş kütüphaneye sahiptir.

### 3.3.1. Kullanılan Kütüphaneler

Bu çalışmada, Python programlama dilinde geliştirilmiş ve veri analizi ile makine öğrenimi uygulamalarını desteklemek amacıyla Tablo 3.3'te gösterilen kütüphaneler kullanılmıştır. Veri ön işleme aşamalarında "NumPy" ve "Pandas" kütüphaneleri, veri setinin işlenmesi ve düzenlenmesinde önemli bir rol oynamıştır.

Veri görselleştirmesi, veri setinin anlaşılması ve model sonuçlarının yorumlanması açısından önemlidir. "Matplotlib.pyplot" kütüphanesi, grafikler ve görseller oluşturmak için kullanılmıştır. Ayrıca, etiketleri sayısal değerlere dönüştürmek ve kategorik verileri işlemek amacıyla "LabelEncoder" kullanılmıştır.

Makine öğrenimi aşamalarında ise farklı algoritmaların uygulanması ve karşılaştırılması hedeflenmiştir. Sınıflandırma algoritmaları olarak "Lojistik regresyon", "Rastgele orman", "K-en yakın komşu", "Destek vektör makineleri", "Naive Bayes" ve "Karar ağacı" tercih edilmiştir. Bu algoritmalar, veri setinin öğrenilmesi ve sınıflandırılması için kullanılmıştır.

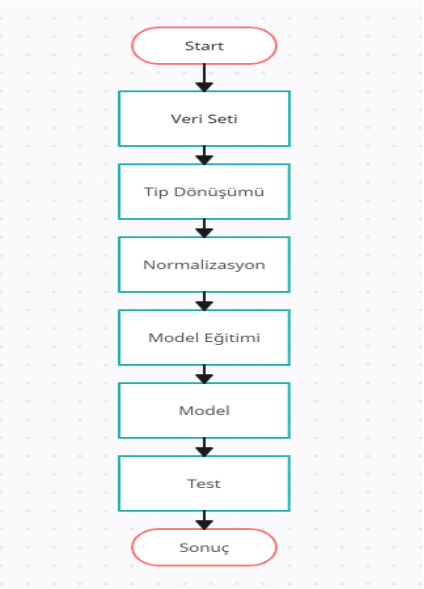
Sonuç olarak, bu çalışmada kullanılan kütüphaneler, veri analizi ve makine öğrenimi işlemlerini kolaylaştırmış ve modelin performansını değerlendirmek amacıyla çeşitli araçlar sağlamıştır. Kütüphanelerin seçimi ve kullanımı, projenin başarılı bir şekilde yürütülmesi için temel bir rol oynamıştır.

**Tablo 3.3.** Kullanılan Kütüphaneler

| SIRA NUMARASI | KÜTÜPHANE ADI |
|---|---|
| 1 | numpy |
| 2 | pandas |
| 3 | matplotlib.pyplot |
| 4 | LabelEncoder |
| 5 | LogisticRegression |
| 6 | RandomForestClassifier |
| 7 | KNeighborsClassifier |
| 8 | confusion_matrix |
| 9 | SVC |
| 10 | GaussianNB |
| 11 | DecisionTreeClassifier |

### 3.4. Saldırı Tespit Sisteminin Gerçekleştirilmesi

Bu çalışmanın amacı, siber güvenlik alanında etkili bir saldırı tespit sistemi tasarımını sunmaktadır. Söz konusu sistem, çeşitli saldırı türlerini otomatik olarak tespit ederek, ağ güvenliğini artırmayı amaçlamaktadır. Anomali tabanlı saldırı tespit sistemi gerçekleştirilirken Şekil 3.3'te ki akış diyagramına göre gerçekleştirilmiştir. Makine öğrenimi projelerinde, veri ön işleme aşaması, elde edilecek sonuçların doğruluğunu artırmak amacıyla büyük bir önem taşır. Bu aşama, veri kümesinin nitelik ve nicelik yönünden düzeltilmesi ve hazırlanması sürecini ifade eder. İlk olarak, veri temizleme adımıyla eksik veya hatalı veriler tespit edilerek düzeltilir. Ardından veri dönüşümü ile verileri makine öğrenimi modeline uygun hale getirilir. Sonrasında, normalizasyon adımında aykırı değerlerin tespit edilmesi ve düzeltilmesi adımı izlenir. Aykırı değerler, genellikle diğer verilere göre anlamlı şekilde sapkın değerlere sahip verilerdir ve modelin hatalı öğrenmesine yol açabilirler.




**Şekil 3.3.** Çalışmanın Akış Diyagramı

Veri seti, makine öğrenimi yöntemleri kullanılarak gerçekleştirilen anomali tabanlı saldırı tespit sisteminin eğitim aşamasına geçilmeden önce, veri ön işleme ve normalizasyon aşamalarına tabi tutulmuştur.

Bu aşamada, veri ön işleme süreci, veri setinin analizine dayanarak gerçekleştirilmiştir. Bu süreç, veri setindeki kategorik (nümerik olmayan) özelliklerin sayısal bir biçime dönüştürülmesini içermiştir. Özellikle, "object" tipindeki sütunlar, makine öğrenimi algoritmalarının etkili bir şekilde çalışabilmesi için sayısal "integer" tiplerine dönüştürülmüştür. Veri ön işleme adımı çerçevesinde, Tablo 3.4'te görüldüğü üzere, özelliklerin veri setindeki tipi "object" olan sütunlar, Tablo 3.5'te gösterildiği üzere "integer" türüne dönüştürülmüştür.

Bu dönüşüm, veri setinin homojenliğini artırmış ve verilerin daha tutarlı ve karşılaştırılabilir bir yapıya getirilmesini sağlamıştır.

**Tablo 3.4.** Veri Seti Ön İşleme Adımı Öncesi

| | id | dur | proto | service | state | spkts | dpkts | sbytes | dbytes | rate | ... | ct_dst_sport_ltm | ct_dst_src_ltm | is_ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0.121478 | tcp | - | FIN | 6 | 4 | 258 | 172 | 74.087490 | ... | 1 | 1 | |
| **1** | 2 | 0.649902 | tcp | - | FIN | 14 | 38 | 734 | 42014 | 78.473372 | ... | 1 | 2 | |
| **2** | 3 | 1.623129 | tcp | - | FIN | 8 | 16 | 364 | 13186 | 14.170161 | ... | 1 | 3 | |
| **3** | 4 | 1.681642 | tcp | ftp | FIN | 12 | 12 | 628 | 770 | 13.677108 | ... | 1 | 3 | |
| **4** | 5 | 0.449454 | tcp | - | FIN | 10 | 6 | 534 | 268 | 33.373826 | ... | 1 | 40 | |

5 rows × 45 columns

**Tablo 3.5.** Veri Seti Ön İşleme Adımı Sonrası

| | dur | proto | service | state | spkts | dpkts | sbytes | dbytes | rate | sttl | ... | ct_src_dport_ltm | ct_dst_sport_ltm |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.121478 | 113 | 0 | 2 | 6 | 4 | 258 | 172 | 74.087490 | 252 | ... | 1 | 1 |
| 1 | 0.649902 | 113 | 0 | 2 | 14 | 38 | 734 | 42014 | 78.473372 | 62 | ... | 1 | 1 |
| 2 | 1.623129 | 113 | 0 | 2 | 8 | 16 | 364 | 13186 | 14.170161 | 62 | ... | 1 | 1 |
| 3 | 1.681642 | 113 | 3 | 2 | 12 | 12 | 628 | 770 | 13.677108 | 62 | ... | 1 | 1 |
| 4 | 0.449454 | 113 | 0 | 2 | 10 | 6 | 534 | 268 | 33.373826 | 254 | ... | 2 | 1 |

5 rows × 40 columns

Normalizasyon aşamasında, eğitim ve test veri setlerinin değerleri [0-1] aralığına dönüştürülmüştür, bu işlem Şekil 3.4'te gösterilen kod parçacığı kullanılarak gerçekleştirilmiştir. Normalizasyon işlemi sonucu Tablo 3.6'da gösterilmiştir.

**Tablo 3.6.** Normalizasyon Adımı Sonrası

| | dur | proto | service | state | spkts | dpkts | sbytes | dbytes | \ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.002025 | 0.856061 | 0.00 | 0.25 | 0.000520 | 0.000364 | 0.000018 | 0.000012 | |
| 1 | 0.010832 | 0.856061 | 0.00 | 0.25 | 0.001352 | 0.003463 | 0.000054 | 0.002867 | |
| 2 | 0.027052 | 0.856061 | 0.00 | 0.25 | 0.000728 | 0.001458 | 0.000026 | 0.000900 | |
| 3 | 0.028027 | 0.856061 | 0.25 | 0.25 | 0.001144 | 0.001093 | 0.000046 | 0.000053 | |
| 4 | 0.007491 | 0.856061 | 0.00 | 0.25 | 0.000936 | 0.000547 | 0.000039 | 0.000018 | |

Veri normalizasyonu, verilerin farklı ölçeklerde olmaları durumunda makine öğrenimi algoritmalarının daha iyi performans göstermesini sağlamak amacıyla uygulanan önemli bir adımdır. Bu aşama, veri setinin tüm değerlerini [0-1] aralığına sıkıştırarak, her bir özelliğin birbiriyle daha uyumlu bir şekilde karşılaştırılabilir hale getirilmesini amaçlar.

```
x_tr=(x_tr-np.min(x_tr))/(np.max(x_tr)-np.min(x_tr))
x_te=(x_te-np.min(x_te))/(np.max(x_te)-np.min(x_te))
```

**Şekil 3.4.** Normalizasyon Kod Parçacığı

Veri setinin önişleme ve normalizasyon adımları tamamlandıktan sonra, makine öğrenimi algoritmalarının saldırıları tespit edebilme başarılarını ölçmek amacıyla eğitim aşamasına geçildi. Bu aşamada, algoritmaların öğrenme süreci gerçekleştirildi. Ardından, başarılarının değerlendirilmesi ve ölçülmesi için bir test aşaması gerçekleştirildi. Test aşaması, algoritmaların eğitim verisi dışındaki verilerle nasıl başa çıktığını değerlendirmek için önemlidir.

### 3.5. Gerçekleştirilen Sistemin Performans Ölçümü

Bazıları birden çok adla bilinen STS için birçok sınıflandırma metriği vardır. Şekil 3.5'te bir STS'nin performansını değerlendirmek için kullanılabilecek iki sınıflı bir sınıflandırıcı için karışıklık matrisini göstermektedir. Matrisin her satırı, tahmin edilen bir sınıftaki örnekleri temsil ederken, her sütün gerçek bir sınıftaki örnekleri temsil eder.


**Şekil 3.5.** Karışıklık Matrisi

Kesinlik: Modelin doğru pozitiflerin tüm pozitif tahminlerine oranını temsil eder. Kesinlik hesaplamak için 3.4'te verilen formül kullanılır.

$$Kesinlik = \frac{DP}{DP+YP} \tag{3.4}$$

Duyarlılık: Sadece pozitif değerlerden doğru sınıflandırılanların oranını verir. Duyarlılık hesaplamak için 3.5'te verilen formül kullanılır.

$$Duyarlılık = \frac{DP}{DP+YN} \tag{3.5}$$

Yanlış Pozitif Oranı (YPO): Yanlış bir şekilde saldırı olarak sınıflandırılan normal vaka sayısının toplam normal vaka sayısına oranı olarak hesaplanır. Yanlış Pozitif Oranı hesaplamak için 3.6'da verilen formül kullanılır.

$$YPO = \frac{YP}{YP+DN} \tag{3.6}$$

Yanlış Negatif Oranı (YNO): Yanlış negatif oran, anomali trafiği tespit edemediği ve onu normal olarak sınıflandırdığı anlamına gelir. Yanlış negatif oran 3.7'de verilen formülle hesaplanır.

$$YNO = \frac{YN}{YN + DP} \quad \textbf{(3.7)}$$

Doğruluk: STS'nin normal veya anomali trafik davranışını tespit etmede ne kadar doğru olduğunu ölçer. Tüm bu doğru tahmin edilen örneklerin tüm örneklere yüzdesi olarak hesaplanır. Doğruluk 3.8'de verilen formülle hesaplanır.

$$Doğruluk = \frac{DP + DN}{DP + DN + YP + YN} \quad \textbf{(3.8)}$$

## 4. BULGULAR

Makine öğrenimi algoritmalarının test aşamasındaki performanslarını Tablo 4.7'de özetlendi. Bu tabloda, algoritmaların saldırıları tespit etme yetenekleri karışıklık matrisleri kullanılarak gösterildi. Doğruluk, kesinlik, duyarlılık, Yanlış Pozitif Oran ve Yanlış Negatif Oran gibi metrikler aracılığıyla algoritmaların performansları detaylı bir şekilde değerlendirildi.

Tablo 4.1'de lojistik regresyon algoritmasının karışıklık matrisi gösterilmektedir.

**Tablo 4.1.** Lojistik Regresyon Karışıklık Matrisi

| | | **Gerçek Değer** | |
|---|---|---|---|
| | | **Pozitif** | **Negatif** |
| **Tahmin** | **Pozitif** | 20051 | 16949 |
| | **Negatif** | 305 | 45027 |

Tablo 4.2'de Rastgele Orman Algoritması karışıklık matrisi gösterilmektedir.

**Tablo 4.2.** Rastgele Orman Algoritması Karışıklık Matrisi

| | | **Gerçek Değer** | |
|---|---|---|---|
| | | **Pozitif** | **Negatif** |
| **Tahmin** | **Pozitif** | 37000 | 0 |
| | **Negatif** | 2 | 45330 |

Tablo 4.3'te K En Yakın Komşu Algoritması karışıklık matrisi gösterilmektedir.

**Tablo 4.3.** K En Yakın Komşu Algoritması Karışıklık Matrisi

| | | Gerçek Değer | |
|---|---|---|---|
| | | **Pozitif** | **Negatif** |
| **Tahmin** | **Pozitif** | 36588 | 412 |
| | **Negatif** | 70 | 45262 |

Tablo 4.4'te Destek Vektör Makineleri karışıklık matrisi gösterilmektedir.

**Tablo 4.4.** Destek Vektör Makineleri Karışıklık Matrisi

| | | Gerçek Değer | |
|---|---|---|---|
| | | **Pozitif** | **Negatif** |
| **Tahmin** | **Pozitif** | 32056 | 4944 |
| | **Negatif** | 193 | 45139 |

Tablo 4.5'te Naive Bayes karışıklık matrisi gösterilmektedir.

**Tablo 4.5.** Naive Bayes Karışıklık Matrisi

| | | Gerçek Değer | |
|---|---|---|---|
| | | **Pozitif** | **Negatif** |
| **Tahmin** | **Pozitif** | 32992 | 4008 |
| | **Negatif** | 1 | 45331 |

Tablo 4.6'da Karar Ağacı karışıklık matrisi gösterilmektedir.

**Tablo 4.6.** Karar Ağacı Karışıklık Matrisi

| | | Gerçek Değer | |
|---|---|---|---|
| | | **Pozitif** | **Negatif** |
| **Tahmin** | **Pozitif** | 36997 | 3 |
| | **Negatif** | 4 | 45328 |

Modelin sınıflandırma sonuçlarına dayanarak, altı farklı sınıflandırıcı modelinin performansını değerlendirilmiştir. Tablo 4.7' de ve Şekil 4.1'te görüldüğü

gibi en yüksek başarı oranı Rastgele Orman Algoritması ve Karar Ağacı vermiştir. En düşük oranı ise Lojistik regresyon vermiştir.

Modellerin başarım oranını artırmak için UNSW-NB15 veri kümesinde

- Object olan kolon tiplerini integer tipine dönüştürülmüştür.
- Özellik seçiminde etkisi olmayan gürültülü veriler kaldırılmıştır.
- Verilerin 0 ile 1 arasına normalize edilmiştir.

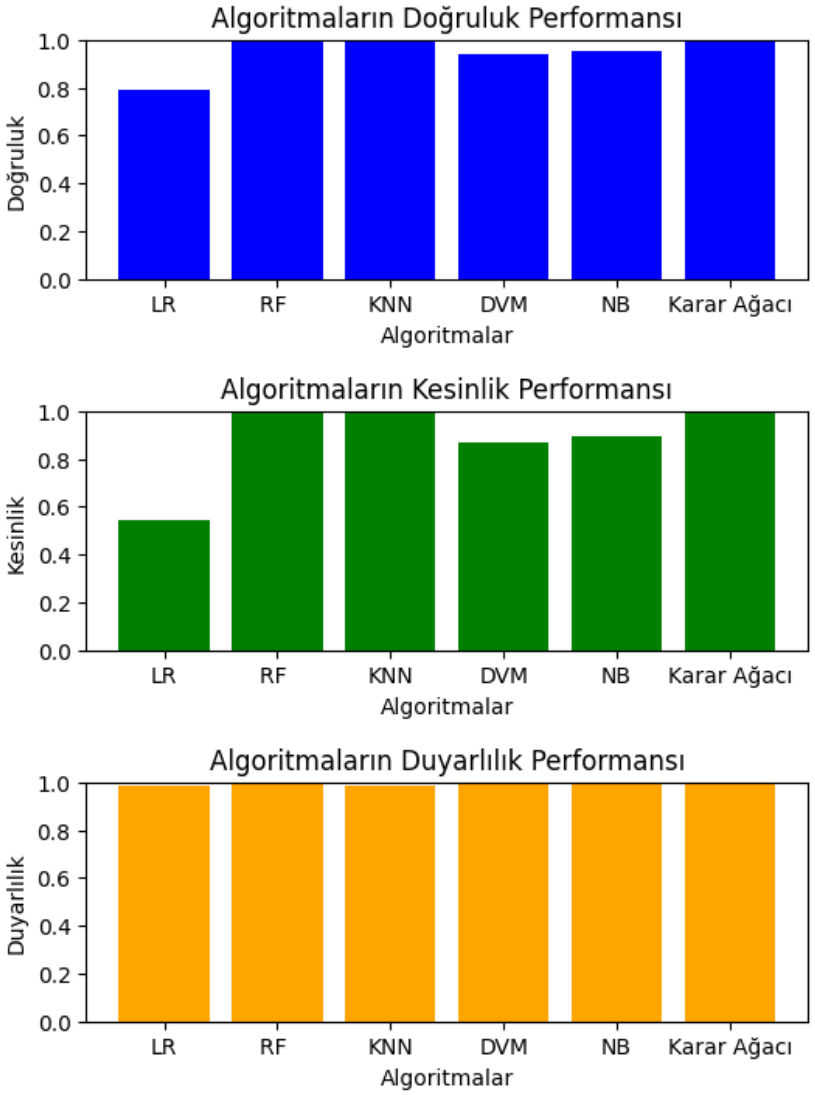
Bu işlemlerin sonucunda altı sınıflandırıcı içinde başarım oranı artmıştır. Bu durum makine öğreniminde veri ön işlemenin en önemli adım olduğunu göstermektedir.

Genel olarak, DP oranı ve kesinlik değerleri, saldırı tespit sistemi için önemli performans parametreleridir, ancak başka bir açıdan en ciddi performans parametreleri, YP oran ve YN oranıdır. Saldırı tespit sistemi çalışmalarında, bu iki parametreyi mümkün olduğunca azaltmayı amaçlamaktadır. Elde edilen sonuçları gösteren Tablo 4.7'de, YPO (Yanlış Pozitif Oran) ve YNO (Yanlış Negatif Oran) performans parametreleri sunulmuştur. Lojistik Regresyon YPO değeri, diğer sınıflandırıcılar arasında en yüksek yanlış pozitif oranı (%27.34) elde ettiği ve en yüksek normal davranışı saldırı olarak yanlış değerlendirdiği anlamına gelirken, en düşük YPO sonucu Rastgele Orman Algoritması ve Karar Ağacı sınıflandırıcısı vermiştir.

YN oranı Lojistik regresyon sınıflandırıcısı, en yüksek yanlış negatif oranı (%1.49) elde ederek en fazla saldırı trafiğini normal trafik olarak yanlış değerlendirdiği görülmüştür. Bu durumun aksine, Naive Bayes ve Rastgele Orman Algoritması en düşük YNO sonuçlarını elde etmiştir.

**Tablo 4.7.** Genel Sonuçlar

| Sınıflandırıcı | Doğruluk | Kesinlik | Duyarlılık | YPO | YNO |
|---|---|---|---|---|---|
| Lojistik Regresyon | %79.04 | % 54.19 | % 98.50 | % 27.34 | %1.49 |
| Rastgele Orman Algoritması | % 99.99 | % 100 | % 99.99 | % 0.0 | % 0.0 |
| K-en yakın komşu algoritması | % 99.41 | % 98.88 | % 98.80 | % 0.9 | % 0.1 |
| DVM Sınıflandırıcısı | % 93.76 | %86.63 | % 99.40 | % 9.87 | % 0.59 |
| Naive Bayes Sınıflandırıcısı | % 95.13 | % 89.16 | % 99.99 | % 8.12 | % 0.0 |
| Karar Ağacı Sınıflandırıcısı | % 99.99 | % 99.99 | % 99.98 | % 0.0 | %0.01 |




**Şekil 4.1.** Algoritmaların Performans Grafiği

# 5. TARTIŞMA

Bu çalışmada, makine öğrenimi yöntemlerinin kullanıldığı anomali saldırı tespiti sistemleri (ASTS) konusu incelenmiş ve mevcut literatürdeki bilgiler ışığında bir değerlendirme yapılmıştır. ASTS'ler, ağ güvenliğini sağlamak amacıyla geliştirilen etkili araçlardır ve makine öğrenimi, bu sistemlerin performansını artırmak için yaygın olarak kullanılan bir yaklaşımdır.

Makine öğrenimi yöntemleri, ASTS'lerin anomali ağ trafiğini tespit etmede ve saldırılara karşı önlemler alma yeteneğini artırmada büyük bir potansiyele sahiptir. Bu çalışmada incelenen literatür taramaları, farklı makine öğrenimi algoritmalarının ASTS'lerde başarıyla uygulanabileceğini göstermiştir.

Ancak, makine öğrenimi yöntemlerinin ASTS'lerde kullanımıyla ilgili bazı zorluklar ve tartışmalar da mevcuttur. Birincisi, ASTS'lerin eğitim verilerinin temsil edilmesi ve etiketlenmesi sürecindeki zorluklardır. Genellikle, normal ve anomali ağ trafiği örneklerinin etiketlenmesi zaman alıcı ve maliyetli olabilir. Ayrıca, ASTS'lerin öğrenme sürecinde kullanılan veri setlerinin doğru temsilini sağlamak önemlidir, çünkü yanlış veya eksik veri setleri ASTS'nin performansını etkileyebilir. Diğer bir tartışma noktası, ASTS'lerin yanlış alarm oranı ile ilgilidir. Makine öğrenimi algoritmaları, öğrenme sürecinde kullanılan verilere dayalı olarak karar verirler ve bu da yanlış pozitif sonuçların ortaya çıkmasına neden olabilir. Yüksek yanlış alarm oranı, ağ yöneticilerini gereksiz tepkilere yönlendirebilir, kaynakların ve zamanın boşa harcanmasına yol açabilir. Bu nedenle, makine öğrenimi yöntemlerinin ASTS'lerde kullanılmasıyla birlikte, yanlış alarm oranının düşük tutulması için çözümler araştırılmalıdır.

Ayrıca, makine öğrenimi modellerinin ASTS'lerdeki uygulanabilirliği ve performansı, kullanılan özellik seçimi ve model hiperparametrelerine de bağlıdır. Her saldırı türü ve ağ yapısı farklı olduğu için, ASTS'lerin özellik seçimi ve model yapılandırması dikkatlice yapılmalıdır. Bu, ASTS'lerin genellemesini ve saldırıları etkin bir şekilde tespit etmesini sağlayabilir.

Sonuç olarak, bu çalışma, makine öğrenimi yöntemlerinin ASTS'lerdeki potansiyelini vurgulamış ve mevcut literatürdeki bulgulara dayanarak değerlendirme

yapmıştır. Makine öğrenimi, AŞTİ'lerin etkinliğini artırmak ve ağ güvenliğini sağlamak için önemli bir araçtır. Ancak, veri setinin güncelliği, yanlış alarm oranının düşük tutulması ve özellik seçimi gibi zorluklar hala dikkate alınması gereken önemli hususlardır. Gelecekteki çalışmalar, ASTS'lerin performansını daha da geliştirmek ve güvenlik açıklarını minimize etmek için bu zorluklar üzerinde odaklanmalıdır.

# 6. SONUÇ

Bilgi teknolojisinin hızlı gelişimi ile bilgisayar ağları endüstri, iş dünyası ve insan yaşamının çeşitli alanlarında yaygın olarak kullanılmaktadır. Bu nedenle, güvenilir ağlar oluşturmak BT yöneticileri için çok önemli bir görevdir. Öte yandan, bilgi teknolojisinin hızlı gelişimi, çok zor bir görev olan güvenilir ağlar oluşturmak için çeşitli zorluklar ortaya çıkardı. Bilgisayar ağlarının kullanılabilirliğini, bütünlüğünü ve gizliliğini tehdit eden birçok saldırı türü vardır. Bu saldırıların engellenmesi için birçok saldırı tespit sistemi yapılmış ve yapılmaya devam etmektedir. Genel olarak, anomali tabanlı ve imza tabanlı iki tür STS vardır. Makine öğrenimi yöntemlerine dayalı anomali tabanlı saldırı tespit sistemleri anomali davranışları tespit etmek ve potansiyel saldırıları belirlemek için kullanılan bir güvenlik mekanizmasıdır. Bu tür saldırı tespit sistemi, yeni saldırı türlerini tespit etme yeteneğine sahip olduğundan yaygın olarak kullanılır. Ancak başka bir açıdan bakıldığında, en büyük yanlış pozitif alarm değerlerini kaydeder, bu da saldırı paketleri olarak kabul edilen çok sayıda normal paket olduğu anlamına gelir. Bu nedenle tasarlanan saldırı tespit sistemlerinin güncel veri setleri kullanılması ve sürekli veri seti güncellenerek modelin eğitilmesi gerekmektedir.

## KAYNAKLAR


**Alazab A., Hobbs M., Abawajy J., Alazab M.** (2012) Using feature selection for intrusion detection system, in 2012 international symposium on communications and information technologies (ISCIT), pp: 296–301.

**Alcaraz C.** (2018), Cloud-assisted dynamic resilience for cyber-physical control systems. IEEE Wirel Commun 25(1):76–82

**Ashfaq RAR., Wang XZ., Huang, JZ., Abbas, H., He YL.** (2017). Fuzziness based semisupervised learning approach for intrusion detection system. Information Sciences, 378, 484-497.

**Aljumah A.** (2017), Detection of Distributed Denial of Service Attacks Using Artificial Neural Networks. International Journal of Advanced Computer Science and Applications (IJACSA), 8(8).

**Altaher A., Almomani A. and Ramadass S.** (2012) Application of Adaptive Neuro-Fuzzy Inference System for Information Secuirty.

**Annachhatre C., H. Austin T., Stamp M.,** (2015), Hidden Markov models for malware classification. Journal of Computer Virology and Hacking Techniques. vol. 11, no. 2, pp. 59–73.

**Arivarasan K., Obaidat M.** (2022) Intrusion Detection System using Aggregation of Machine Learning Algorithms Proceedings of the 2022 International Conference on Computer, Information and Telecommunication Systems, CITS 2022.

**Al-Yaseen WL., Othman ZA., Nazri MZA.** (2017). Multilevel hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. Expert Systems İth. Applications, 67, 296-303.

**Bace R. and Mell P.** (2001) NIST Special Publication on Intrusion Detection Systems.

**Bace RG.** (2000) Intrusion Detection.

**Bajaj K., Arora A.** (2013) Dimension reduction in intrusion detection features using discriminative machine learning approach. IJCSI International Journal of Computer Science Issues PP:10(4):324–328

**Bauer JM. and Van Eeten MJG.** (2009) Cybersecurity: Stakeholder Incentives, Externalities, and Policy Options.

**Benassi P.** (1999). TRUSTe: an online privacy seal program. Communications of the ACM Volume 42 Issue 2, Feb. 1999, 56 – 59. Zucker, L. G. (1986). Production of trust: Institutional sources of economic structure, 1840–1920. Research in Organizational Behavior, Vol 8, 1986, 53-111).

**Belavagi MC., Muniyal B.** (2016), Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Computer Science, 89, 117-123.

**Bhuyan MH, Bhattacharyya DK, Kalita JK** (2014) Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials 16(1):303–336

**Bishop CM.** (1995), Neural networks for pattern recognition. England Oxford University.

**Breiman L., Friedman JH., Olshen R A., Stone P J.** (1984), Classification and regressing trees. California: Wadsworth International Group

**Butun I., Morgera SD., Sankar R.** (2014) A survey of intrusion detection systems in wireless sensor networks. IEEE Communications Surveys & Tutorials 16(1):266–282.

**Camp, L. J.** (1999). Web security and privacy: An American perspective. The Information Society, 15(4), 249-256.

**Castillo NM., Lee J., Zahra FT. and Wagner DA.** (2015) MOOCS for Development: Trends, Challenges and Opportunities.

**Chebrolu S., Abraham A., Thomas JP**., Feature deduction and ensemble design of intrusion detection systems. Computers & Security, vol. 24, no. 4, pp:295–307.

**Debar H., Dacier M. and Wespi A.** (2000) A Revised Taxonomy for Intrusion-Detection Systems.

**Dua S., Du X.,** (2016), Data mining and machine learning in cybersecurity. CRC press.

**Elhag S., Fernández A., Bawakid A., Alshomrani S., Herrera F.,** (2015) On the combination of genetic fuzzy systems and pairwise learning for improving detection rates on intrusion detection systems. Expert Syst Appl, vol. 42, no. 1, pp:193–202.

**El-Bakry HM. and Mastorikas N.** (2008) A Real-Time Intrusion Detection Algorithm for Network Security.

**Fitzgerald, K. J.** (1995). Information security baselines. Information Management & Computer Security, 3(2), 8-12.

**Fieser, James, Ethics,** The Internet Encyclopedia of Philosophy (2006), at www.iep.utm.edu/ (accessed on 30 December 21, 2014)

**Katzan Jr, H.** (2011). On the privacy of cloud computing. International Journal of Management & Information Systems,14(2)

**Kemmerer RA. and Vigna G.** (2002) Intrusion Detection: A Brief History and Overview.

**Khraisat A., Gondal I., Vamplew P., Kamruzzaman J.** (2019) Survey of intrusion detection systems: techniques, datasets and challenges. Cybersecurity, 2:1.

**Khraisat A., Gondal I., Vamplew P.** (2018) An anomaly intrusion detection system using C5 decision tree classifier. In: Trends and applications in knowledge discovery and data mining. Springer International Publishing, Cham, pp: 149–155

**Kim G., Lee S., Kim S.,** (2014) A novel hybrid intrusion detection method integrating anomaly detection with misuse detection. Expert Syst Appl, vol. 41, no. 4, Part 2, pp: 1690–1700

**Koc L., Mazzuchi T. A, Sarkani S.,** (2012) A network intrusion detection system based on a hidden Naïve Bayes multiclass classifier. Expert Syst Appl, vol. 39, no. 18, pp. 13492–13500.

**Kolter JZ. and Maloof MA.** (2006). Learning to Detect and Classify Malicious Executables in the Wild.

**Kreibich C., Crowcroft J.** (2004) Honeycomb: creating intrusion detection signatures using honeypots. SIGCOMM Comput Commun Rev 34(1):51–56.

**Kshetri N., Voas J.** (2017), Hacking power grids: a current problem. Computer 50(12):91–95

**K.J. Milmann and M. Avaizis,** editors. Scientific Python, volume 11 of Computing in Science & Engineering. IEEE/AIP, March 2011.

**Liao HJ., Richard Lin CH., Lin YC., and Tung KY,** (2013) Intrusion detection system: a comprehensive review. J Netw Comput Appl, vol. 36, no. 1, pp. 16–24.

**Lin WC., Ke SW., Tsai CF.,** (2015) CANN: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl-Based Syst, vol. 78, no. Supplement C, pp:13–21.

**Li Y., Xia J., Zhang S., Yan J., Ai X., Dai K.,** (2012), An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst Appl, vol. 39, no. 1, pp: 424–430.

**Luo, X.** (2002). Trust production and privacy concerns on the Internet: A framework based on relationship marketing and social exchange theory. Industrial Marketing Management, 31(2), 111-118.

**Mahesh B.,** (2018), Machine Learning Algorithms. International Journal of Science and Research, Volume 9 Issue 1

**Manocha S., and Girolami MA.** (2007), An empirical analysis of the probabilistic Knearest neighbour classifier. Pattern Recognition Letters, 28, 1818–1824.

**McHugh J., Christie A. and Allen J.** (2000) The Role of Intrusion Detection Systems.

**Meiners C. R., Patel J., Norige E., Torng E., Liu AX.** (2010) Fast regular expression matching using small TCAMs for network intrusion detection and prevention systems. Presented at the Proceedings of the 19th USENIX conference on security, Washington, DC.

**Mitchell T.** (1997). Machine learning. New york: McGraw Hill

**Modi C., Patel D., Borisaniya B., Patel H., Patel A., Rajarajan M.** (2013) A survey of intrusion detection techniques in Cloud. J Netw Comput Appl. vol. 36, no. 1, pp: 42–57.

**Narudin FA., Feizollah A., Anuar NB., Gani A.** (2016), Evaluation of machine learning classifiers for mobile malware detection. Soft Computing, 20(1), 343-357.

**Ozgode Yigin, B., Algin, O., & Saygili, G.** (2020). Comparison of morphometric parameters in prediction of hydrocephalus using random forests. Computers in Biology and Medicine, 116, 103547.

**Padayachee, K**. (2012). Taxonomy of compliant information security behavior. Computers & Security, 31(5), 673-680.

**P.F. Dubois,** (2017) editor. Python: Batteries Included, volume 9 of Computing in Science & Engineering. IEEE/AIP, May 2007.

**Reis, I., Baron, D., & Shahaf, S.** (2018). Probabilistic Random Forest: A Machine Learning Algorithm for Noisy Data Sets. The Astronomical Journal, 157(1), 16.

**Roesch M.** (1999) Snort: Lightweight Intrusion Detection for Networks.

**Roshan, S., Miche, Y., Akusok, A., & Lendasse, A.** (2018). Adaptive and online network intrusion detection system using clustering and Extreme Learning Machines. Journal of the Franklin Institute, 355(4), 1752-1779.

**Rongheng S.,** (2014) Applied Mathematical Statistics (3rd Edition) ,CA: Science Press.

**Shai SS. Shai BD.** (2014), Understanding Machine Learning From Theory to Algorithms. Cambridge Universtiy Press

**Shen C., Liu C., Tan H., Wang Z., Xu D., Su X.** (2018), Hybrid-augmented device fingerprinting for intrusion detection in industrial control system networks. IEEE Wirel Commun 25(6):26–31

**Solmaz R., Günay M., Alkan A.,** (2014), Fonksiyonel Tiroit Hastalığı Tanısında Naive Bayes Sınıflandırıcının Kullanılması.

**Solms RV.** (1998) Information security management (3): the Code of Practice for Information Security Management (BS 7799). Information Management & Computer Security,

6/5; 224–225.

**Stallings, W.** (2006). Cryptography and network security principles and practices. USA: Prentice Hall.

**Studnia I., Alata E., Nicomette V., Kaâniche M., Laarouchi Y.** (2018) A language-based intrusion detection approach for automotive embedded networks. Int J Embed Syst 10(1):1–12

**Subramanian S, Srinivasan VB, Ramasa C** (2012) Study on classification algorithms for network intrusion systems. Journal of Communication and Computer 9(11):1242–1246

**Thaseen S., Kumar CA.,** (2013) An analysis of supervised tree based classifiers for intrusion detection system in 2013 international conference on pattern recognition, informatics and Mobile engineering, 2013, pp: 294–299

**Tsai CF. , Hsu YF., Lin CY., Lin WY.,** (2009), Intrusion detection by machine learning. Expert Systems with Applications

**Vazquez C.** (2014) Auditing Using Vulnerability Tools to Identify Today's Threats to Business Performance.

**Viinikka J., Debar H., Mé L., Lehikoinen A., Tarvainen M.,** (2009) Processing intrusion detection alert aggregates with time series modeling. Information Fusion, vol. 10, no. 4, pp: 312–324

**Yang X., Tian YL.**,(2012), EigenJoints-based action recognition using NaïveBayes-nearest-neighbor in 2012 IEEE computer society conference on computer vision and pattern recognition workshops, pp. 14–19

**Ye N., Emran SM., Chen Q., Vilbert S.** (2002) Multivariate statistical analysis of audit trails for host-based intrusion detection. IEEE Trans Comput 51(7):810–820

**Zwick, D., & Dholakia, N.** (2004). Whose identity is it anyway? Consumer representation in the age of database marketing. Journal of Macromarketing, 24(1), 31-43.

**Walkinshaw N., Taylor R., Derrick J.,** (2016) Inferring extended finite state machine models from software executions. Empirical Software Engineering, journal article vol. 21, no. 3, pp: 811–853.

**Wang H., Lee M., and Wang C.** (1998), Consumer Privacy Concerns about Internet Marketing. Communications of the ACM, Volume 41, Number 3, 63-70.

**Wang K. and. Stolfo SJ.** (2004). Anomalous Payload-Based Network Intrusion Detection

**Weir, C. S., Douglas, G., Carruthers, M. and Jack, M.** (2009). User perceptions of security, convenience and usability for ebanking authentication tokens, Computers & Security,

28, 1-2, pp. 47-62

**Whitten, A., & Tygar, J. D.** (1999). Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In Usenix Security (August)