TÜRKİYE CUMHURİYETİ
ANKARA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

VERİ SETİNE UYGULANAN ÖN İŞLEMLERİN ANOMALİ TABANLI SALDIRI TESPİT MODELLERİNİN PERFORMANSLARI ÜZERİNDEKİ ETKİSİNİN İNCELENMESİ

Esen Gül İLGÜN

DİSİPLİNLERARASI ADLİ BİLİMLER ANABİLİM DALI
ADLİ BİLİŞİM
YÜKSEK LİSANS TEZİ

DANIŞMAN
Prof. Dr. Refik SAMET

ANKARA
2021

TÜRKİYE CUMHURİYETİ
ANKARA ÜNİVERSİTESİ
SAĞLIK BİLİMLERİ ENSTİTÜSÜ

VERİ SETİNE UYGULANAN ÖN İŞLEMLERİN ANOMALİ TABANLI SALDIRI TESPİT MODELLERİNİN PERFORMANSLARI ÜZERİNDEKİ ETKİSİNİN İNCELENMESİ

Esen Gül İLGÜN

DİSİPLİNLERARASI ADLİ BİLİMLER ANABİLİM DALI
ADLİ BİLİŞİM
YÜKSEK LİSANS TEZİ

DANIŞMAN
Prof. Dr. Refik SAMET

ANKARA 2021

Ankara Üniversitesi
Sağlık Bilimleri Enstitüsü Müdürlüğü'ne,

Yüksek Lisans tezi olarak hazırlayıp sunduğum "Veri Setine Uygulanan Ön İşlemlerin Anomali Tabanlı Saldırı Tespit Modellerinin Performansları Üzerindeki Etkisinin İncelenmesi" başlıklı tez; bilimsel ahlak ve değerlere uygun olarak tarafımdan yazılmıştır. Tezimin fikir/hipotezi tümüyle tez danışmanım ve bana aittir. Tezde yer alan deneysel çalışma/araştırma tarafımdan yapılmış olup, tüm cümleler, yorumlar bana aittir.
Yukarıda belirtilen hususların doğruluğunu beyan ederim.

Öğrencinin Adı Soyadı: Esen Gül İLGÜN
Tarih: 12.01.2022
İmza:

Ankara Üniversitesi Sağlık Bilimleri Enstitüsü
Adli Bilimler Anabilim Dalında
Esen Gül İLGÜN tarafından hazırlanan
"Veri Setine Uygulanan Ön İşlemlerin Anomali Tabanlı Saldırı Tespit Modellerinin Performansları Üzerindeki Etkisinin İncelenmesi"adlı tez çalışması aşağıdaki jüri tarafındanYÜKSEK LİSANS TEZİ olarak OY BİRLİĞİ ile kabul edilmiştir.

Tez Savunma Tarihi: 12.01.2022

Prof.Dr. Refik SAMET
Ankara Ü. Mühendislik Fakültesi
Jüri Başkanı

Dr.Öğr.Üyesi İrem ÜLKÜ
Ankara Ü. Mühendislik Fakültesi
Üye

Dr.Öğr.Üyesi Semra AYDIN
Milli Savunma Ü.Kara Harp Okulu
Üye

Tez hakkında alınan jüri kararı, Ankara Üniversitesi Sağlık Bilimleri Enstitüsü Yönetim Kurulu tarafından onaylanmıştır.

Prof.Dr. Fügen AKTAN
Sağlık Bilimleri Enstitüsü Müdürü

İÇİNDEKİLER

ÖNSÖZ

İnternet kullanıcı sayısının artışı ile dijital ortamdaki kritik öneme sahip verilerin artması, sistemlerin güvenlik açıkları, gelişen yazılım teknolojisi gibi nedenler, siber saldırganların bu verilere ulaşmak için sayısı ve çeşidi artan saldırıları ile sonuçlanmıştır. Bu saldırıların tespitinde, sadece bilinen saldırıları tanıyabilen imza tabanlı Saldırı Tespit Sistemleri (STS) yetersiz kalmakta ve bu durum, karşılaşılmamış bir saldırıyı yüksek hızda ve oranda tespit edebilen akıllı siber savunma yöntemleri geliştirmeyi zorunlu kılmaktadır. Bu çalışmada kısa sürede yüksek saldırı tespiti başarısı elde etmek için kategorik veri kodlama, ölçeklendirme, öznitelik seçimi ön işlemlerinin ayrı ayrı ve birlikte uygulandığı NSL-KDD veri setleri ile K-Nearest Neighbor (KNN), Multi Layer Perceptron (MLP), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM) makine öğrenimi algoritmaları kullanılarak saldırı tespit modelleri oluşturulmuştur. Oluşturulan modeller test edilmiş ve alınan sonuçlar, ön işlem uygulanmamış veri setleri ile oluşturulan modellerden alınan sonuçlarla karşılaştırılarak analiz edilmiştir. En başarılı sonuçlara, leave one out kategorik kodlama ve min-max ölçeklendirme ön işlemlerinin uygulandığı veri setinden öznitelik seçimi ile elde edilen öznitelik alt kümesine sahip, bahsi geçen tüm ön işlemlerin birlikte uygulandığı veri setleri ile ulaşılmıştır. XGBoost gömülü öznitelik seçim yöntemiyle seçilen öznitelikler ve LightGBM algoritması ile oluşturulan modelde, eğitim veri seti üzerinde 0,373 s sürede %96,1 saldırı tespit başarısına, hibrit öznitelik seçim yöntemi kullanılarak Mutual Information (MI) ve XGBoost öznitelik seçim yöntemlerinin ortak seçtiği öznitelikler ve XGBoost algoritması ile oluşturulan modelde test veri seti üzerinde 0,007 s sürede %98,2 saldırı tespit başarısına ulaşılmıştır. MI ve XGBoost öznitelik seçim yöntemlerinin ortak olarak seçtiği öznitelikler ve RF algoritması ile oluşturulan modelde ise eğitim veri seti üzerinde 5,1 s sürede %99,6 saldırı tespit başarısı elde edilmiştir.

Çalışma disiplinini ve dinamik eğitim anlayışını, bir öğretmen olarak örnek aldığım danışman hocam, Prof. Dr. Refik SAMET'e ve bu çalışmanın hazırlanması aşamasında her türlü desteğini benden esirgemeyen İLGÜN ailesine teşekkür ederim.

SİMGELER VE KISALTMALAR

CFS	Korelasyon Tabanlı Öznitelik Seçimi (Correlation Based Feature Selection)
DCNN	Derin Evrişimli Sinir Ağı (Deep Convolutional Neural Network)
DoS	Hizmet Reddi Saldırıları (Denial of Service)
DT	Karar Ağacı (Decision Tree)
FN	Kaçan Alarm (False Negative)
FP	Yanlış Alarm (False Positive)
GA	Genetik Algoritma (Genetic Algorithm)
KNN	K-En Yakın Komşu (K-Nearest Neighbor)
LightGBM	Hafif Gradyan Arttırma Makinesi (Light Gradient BoostingMachine)
LR	Logistic Regression
MLP	Çok Katmanlı Algılayıcı (Multi Layer Perceptron)
MI	Karşılıklı Bilgi (Mutual Information)
R2L	Uzaktan Yerel Saldırı (Remote to Local)
RF	Rastgele Orman (Random Forest)
s	Saniye
STS	Saldırı Tespit Sistemi
SVM	Destek Vektör Makineleri (Support Vector Machine)
TN	Doğru Hamle (True Negative)
TP	Doğru Alarm (True Positive)
U2R	Normal Kullanıcıdan Yönetici Hesabına (User to Root)
XGBoost	Ekstrem Gradyan Arttırma (eXtreme Gradient Boosting)

ŞEKİLLER

ÇİZELGELER

1. GİRİŞ

Donanım ve yazılım teknolojilerinin gelişmesi ile hayatı birçok alanda kolaylaştıran uygulamalar geliştirilmiş ve internet ile kesintisiz olarak hizmete sunulan bu uygulamalar, internet kullanıcı sayısını tüm dünyada dikkat çekecek oranda arttırmıştır. Güncel dijital verilerin yayınlandığı We Are Social Temmuz 2021 Raporu'na göre, Dünya genelinde internet kullanıcı sayısı yaklaşık 4,8 milyar ve bu sayı Dünya nüfusunun %61'i (We are social, 2021). İnternet kullanıcı sayısının 2022 yılına kadar altı milyara, 2030 yılına kadar ise 7,5 milyara yükseleceği tahmin ediliyor. Bu sayı 2030 yılında öngörülen 8,5 milyarlık Dünya nüfusunun %90'ına tekabül etmektedir (Cybersecurity Ventures, 2021). Bu artış, aynı zamanda dijital ortamda kritik öneme sahip bilginin artması demektir ve bu bilgilerin değeri saldırganları motive ederken; pratikte tamamen güvenilir bir sistem kurmanın zorluğu, işletim sistemlerindeki açıkları çoğunlukla ilk olarak saldırganların fark etmeleri ve kullanmaları, sistemin güvenlik düzeyi yüksek olsa bile, kullanıcı ayrıcalıklarına sahip içerideki kişiler tarafından suistimal edilmesi, erişim kontrolü düzeyinin yükseldikçe kullanıcı verimliliğinin düşmesi, tüm kripto sistemlerin kırılabilmeleri ve yazılım teknolojisinin hızlı gelişimi gibi nedenler bu verileri saldırganların açık hedefi haline getirmiştir (Sundaram, 1996).

Siber güvenlik kuruluşu Cybersecurity Ventures'ın verilerine göre 2015 yılında siber saldırıların küresel ekonomiye zararı yaklaşık üç trilyon dolar iken, bu sayının 2021 yılında altı trilyon dolara, 2025 yılında ise 10,5 trilyon dolara ulaşabileceği öngörülüyor ki bu zarar, doğal afetlerin ve yasadışı uyuşturucu ticaretinin toplamının verdiği zarardan kat be kat daha büyüktür (Cybersecurity Ventures, 2020). 2020 Küresel Riskler Raporu'nda, geçmiş yıllarda çoğunluğu özel sektör kuruluşlarını hedef alan siber saldırıların yerini, son zamanlarda enerji, sağlık, ulaşım sektörleri gibi kritik altyapıları hedef alan siber saldırıların aldığı ve bu saldırıların zararının büyüklüğünün siber güvenliğe yatırımı kaçınılmaz kıldığı belirtilmektedir (World Economic Forum, 2020). Gartner'ın son tahminine göre, dünya çapında bilgi güvenliği ve risk yönetimi

teknolojisi ile ilgili harcamaların %12,4 artarak 2021'de 150,4 milyar dolara ulaşması bekleniyor. Bu oran 2020'de %6,4 idi (Gartner, 2021).

Siber saldırıların verdiği zararların büyüklüğü, son yıllarda kullandıkları gelişmiş teknikler sayesinde, hedef aldıkları sisteme yönelik çeşitlenerek ve sızdıkları sistemin kullanıcı davranışlarını öğrenerek sisteme başarılı şekilde entegre olabilen, bu sayede güvenlik duvarları, antivürüs yazılımları gibi çekirdek modunda çalışan güvenlik yazılımlarını kolaylıkla atlatabilen yeni nesil kötü amaçlı yazılımların artışı ile yakından ilgilidir (Aslan ve Samet, 2019). Gelişmiş tekniklerin kullanıldığı siber saldırıların verdiği zararların engellenmesi aşamasında, yapay zekâ'nın bir alt dalı olan makine öğrenimi yöntemleri kullanılarak geliştirilen ve ağ trafiğini analiz ederek sıfır gün atakları denilen daha önce karşılaşılmamış saldırıların tespitinde oldukça başarılı olabilen anomali tabanlı STS'ler gittikçe daha çok önem kazanmaktadır. Anomali tabanlı STS'lerde algılama bir sınıflandırma görevidir ve bir sınıflandırma görevinde amaç makine öğrenimi algoritmasını örnek verilerle eğiterek yeni gelen örneklerin sınıfının doğru tahmin edilmesini sağlamaktır. Makine öğrenimi algoritmalarından en üst düzeyde performans, dolayısıyla sınıflandırma başarısı elde etmenin etkili bir yolu ise kullanılan veri setini en doğru şekilde ön işlemektir.

Bu çalışmanın amacı, makine öğrenimi yöntemi kullanılarak geliştirilen saldırı tespit modellerinin saldırı tespiti başarısında, kullanılan veri setine uygulanan ön işlemlerin etkisini analiz etmek, karşılaşılan problemlere çözüm önerileri sunmak, yüksek tespit başarısı ve hızına sahip, sıfır gün saldırıları tespiti yapabilen anomali tabanlı saldırı tespit modeli geliştirmektir. Bu çalışma, ilgili çalışmalar başlığı altında bahsedilen çalışmalarda görüleceği üzere, saldırı tespit modeli geliştirmek için kullanılan veri setine uygulanan kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemlerinin üçünün ayrı ayrı ve birlikte saldırı tespit performansına etkisini inceleyen yeterince çalışma olmadığı için, başarılı bir saldırı tespit modeli geliştirirken veri setine uygulanan bu üç ön işleme odaklanmıştır. Çalışmada veri setinin ön işlenmesi, ön işlenmiş veri seti ve makine öğrenimi algoritmalarıyla model oluşturma ve model değerlendirme aşamalarından oluşan bir metodoloji önerilmiş ve önerilen

metodolojinin uygulaması yapılmıştır. Uygulamalar sonucunda ön işlenmiş veri setleri ve makine öğrenimi algoritmaları ile oluşturulan saldırı tespit modellerinde, ön işlemlerin saldırı tespit başarısı ve hızı üzerindeki etkisi ön işlem uygulanmayan veri seti ile oluşturulan modellerle karşılaştırılarak incelenmiştir. İnceleme sonucunda; veri setine uygulanan ön işlemlerin, kullanılan makine öğrenimi algoritmasının performansına, dolayısıyla geliştirilen saldırı tespit modelinin saldırı tespit başarısı ve hızına etkisinin analizi yapılmıştır. Ayrıca kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemlerin veri seti üzerinde ayrı ayrı ve birlikte uygulanmasının, kurulan modelin performansına etkisi değerlendirilmiştir. Veri setine uygun ön işlemlerin uygulanmasının önemi belirtilmiştir.

Giriş bölümünden sonra yer alan konu başlıklarında, konu ile ilgili çalışmalar, yapılan incelemeler ve tespitlerin anlaşılırlığını kolaylaştırmak için STS geliştirme aşamalarında kullanılan kavramlar ve tanımlar, bilgi güvenliği ve siber saldırılar, STS'ler ve STS çalışmalarında kullanılan veri setleri anlatılmıştır. Çalışmanın 2. bölümünde yüksek tespit başarısı ve hızına sahip anomali tabanlı saldırı tespit modeli geliştirmek için önerilen yöntem anlatılmaktadır. 3.bölümde ise önerilen yöntemin uygulaması yapılmıştır. Sonraki bölümlerde ise tartışma ve sonuçlar yer almıştır.

1.1. İlgili Çalışmalar

Çalışmanın ilk aşamasında literatür incelendiğinde, makine öğrenimi yöntemi ile geliştirilen anomali tabanlı STS'lerde saldırı tespit başarısını ve hızını arttırmak için, veri setine uygulanan ön işlemlere odaklanan bir çok çalışma olduğu görülmüştür. Bunlardan bazıları:

Birçok STS modelinin ağ trafiğine ilişkin görüşlerini TCP/IP paket başlıklarıyla sınırladığı fakat ağ solucanı ve hizmet reddi saldırıları gibi bir çok farklı özellikteki saldırıların daha etkili tespiti için istek paketlerin derinlemesine incelenmesi gerektiği, bunun da veri boyutunu ve hesaplama maliyetini azaltmak, doğru tespit oranını arttırıp

yanlış alarm oranını azaltmak için öznitelik seçimi gibi yaygın olarak kullanılan ön işlemlerle ile mümkün olduğu, veri ön işlemenin anomali tabanlı STS'lerin başarısı üzerinde büyük bir etkiye sahip olduğu vurgulanmıştır (Davis and Clark, 2011).

Kategorik kodlama yöntemlerinin determined, algorithmic ve automatic olmak üzere üç başlık altında incelendiğinden, determined tekniklerinin (one hot encoder, leave one out encoder, hashing encoder vs.) düşük çalışma süresi ve düşük hesaplama karmaşıklığına sahip olmalarından dolayı büyük boyutlu veri setleri için uygun olduğundan bahsedilmiştir (Hancock ve Khoshgoftaar, 2020).

Anomali tabanlı STS geliştirilirken kullanılan veri setine ön işlem olarak sırasıyla, çeşitli kategorik veri kodlama yöntemleri uygulanarak en başarılı kodlama yönteminin seçildiğinden, daha sonra kullanılacak Derin Evrişimli Sinir Ağı (DCNN) algoritması için rastgele arama yöntemi kullanılarak hiper parametre değerleri bulunduğundan, uygulanan ön işlemlerin, oluşturulan modellerin saldırı tespit oranını ve hızını kayda değer ölçüde iyileştirdiğinden bahsedilmiştir (Naseer and Saleem, 2018).

Başka bir çalışmada veri setine sırasıyla, min-max ölçeklendirme, one hot encoding kategorik veri kodlama ve LightGBM algoritması ile öznitelik seçimi ön işlemleri uygulanmıştır. En başarılı sonuç olan %89, 82 doğruluk oranına, tüm ön işlemler uygulanarak elde edilen veri seti ve derin öğrenme algoritmalarından olan Autoencoder (AE) algoritması ile oluşturulan model ile ulaşılmıştır (Tang ve ark., 2020).

NSL-KDD ve ISCXIDS2012 veri setlerinde, kategorik veri kodlama ve ölçeklendirme ön işlemleri yapıldıktan sonra öznitelik seçimi için parametre düzenlemesi yapılmış yapay arı kolonisi (ABC) algoritması ve seçilen öznitelikleri değerlendirmek için AdaBoost.M2 algoritması kullanıldığından bahsedilmiştir. Söz konusu çalışmada kullanılan AdaBoost.M2'nin, standart Adaboost algoritmasından farkının, zayıf hipotezin iyiliğini ölçen sahte kayıp kavramını kullanarak çoklu sınıf

içeren veri setlerini sınıflandırmada daha başarılı olması, AdaBoost.M2'de beş yapraklı karar ağaçlarının temel öğreniciler olarak kullanılması, ilk aşamada tüm örneklerin eşit ağırlığa sahip olduğu ve her yinelemede örneğin ağırlığının değiştirilmesi ve sözde kaybın en aza indirilerek temel öğrenicilerin tahmin yeteneklerinin güçlendirilmesi olduğu belirtilmiştir. Çalışmanın sonunda %99,61 algılama, %0,01 yanlış algılama, %98,90 doğru tespit oranlarına ulaşılmıştır (Mazini ve ark., 2019).

Bilgi kazanım oranına göre öznitelik seçimi yapan, optimal öznitelik seçim algoritması önerilmiş, önerilen öznitelik seçim algoritmasının, SVM (Support Vector Machine) sınıflandırma algoritması ile kullanımının DoS (Denial of Service Attack) saldırılarında % 99,11 tespit başarısı, 2,08 s tespit süresi sonuçlarına ulaştırdığından ve yanlış alarm oranını kayda değer oranda azalttığından bahsedilmiştir (Balakrishnan ve ark., 2014).

KDD99 veri seti üzerinde Ki-Kare (Chi Square), Bilgi Kazancı (Information Gain), Kazanım Oranı (Gain Ratio), Gini Katsayısı (Gini Index), OneR, ReliefF, GA, FFS (Forward Feature Selection) ve BFS (Backward Feature Selection) olmak üzere dokuz farklı öznitelik seçim yöntemi kullanılarak elde edilen veri setleri ile oluşturulan modellerde SVM, KNN ve ELM (Extreme Learning Machines) sınıflandırıcılarının kullanıldığından bahsedilen bir çalışmada, 2,24 s ile en hızlı model kurulma süresinin, oneR ve ELM'i bir arada kullanan modele, en yüksek saldırı tespit oranının ise, KNN ve FFS yöntemlerini bir arada kullanan modele ait olduğu görülmüştür. Çalışmada yapılan tüm analizlerin sonucunda öznitelik seçim yöntemlerinin her üç sınıflandırıcı için de tespit oranını ve hızını arttırdığı görülmüştür (Kaynar ve ark., 2018).

Filtreleme öznitelik seçim yöntemi olan, korelasyon tabanlı öznitelik seçimi (CFS) ile sınıf değişkeni ile yüksek korelasyona sahip öznitelikler dışında, geriye kalan özniteliklerden en az biriyle yüksek ilişkiye sahip özniteliklerin seçiminin gereksiz ve tekrar eden öznitelik seçimine sebep olduğu ve bu sorunun Bat Algorithm (BA) ile çözülerek CFS+BA öznitelik seçim yönteminin önerildiğinden ve C4.5, RF, Passive Aggressive (PA) algoritmalarından oluşan bir topluluk sınıflandırıcı ile %99,81

doğruluk oranı, %0,08 yanlış alarm oranı sonuçlarına ulaşıldığından bahsedilmiştir (Zhou ve ark., 2020).

Gömülü öznitelik seçim yönteminin önerildiği bir çalışmada, öznitelik seçimi ve sınıflandırma için kullanılan MARS (Multivariate Adaptive Regression Splines), SVM, LPG (Learned Policy Gradient) algoritmaları ile altı adet öznitelik seçilmiş ve bu algoritmalarla sırasıyla %63,9; 99,46; 95,26 oranlarında saldırı tespiti sonuçlarına ulaşılmıştır (Sung ve Mukkamala, 2004).

Öznitelikleri, sınıf değişkeni ile ilişkilerine göre seçmeye dayalı filtreleme ve sarmalayıcı yöntemlerin aksine özniteliklerin birbirleriyle ilişkisinin de incelenmesi gerektiği ve bazı özniteliklerin birlikte çalıştıklarında daha başarılı sonuçlar doğurduğu fikrini temel alan optimizasyona dayalı öznitelik seçim yöntemlerinin en uygun öznitelik alt kümesini seçmede daha başarılı olduğundan, özellikle sıfır gün saldırılarını tanımlamak için birden fazla sınıflandırma algoritmasının tahmin sonuçlarının çeşitli tekniklerle birleştirilerek daha güçlü ve genellenebilir sonuçların üretildiği topluluk öğrenme yöntemlerinin başarısından ve geliştirilen saldırı tespit modellerinin genelleme başarısını kanıtlamak için farklı ve güncel veri setleri ile kullanılmasının öneminden bahsedilmiştir (Torabi ve ark., 2021).

Filtreleme ve sarmalayıcı yöntemleri birleştiren hibrit öznitelik seçme yönteminin önerildiği bir çalışmada, ilk aşamada CFS kullanılarak, sınıf ile en yüksek korelasyona sahip öznitelikler seçilmiş, ikinci aşamada ise GA (Genetic Algorithm) ile seçilen öznitelikleri değerlendirmek için RF algoritması kullanılarak optimum öznitelik alt kümesi oluşturulmuştur. Elde edilen optimum öznitelik alt kümesi k-çapraz doğrulama yöntemi ile değerlendirilmiş ve diğer birçok yaklaşıma göre daha başarılı sonuçlara ulaşıldığı belirtilmiştir (Kamarudin ve Watson, 2019).

Anomali tabanlı STS geliştirilirken kullanılan veri setinden en uygun öznitelikleri seçmek için Gri Kurt Optimizasyonu (GWO) ve Parçacık Sürü Optimizasyonu (PSO) algoritmaları ile seçim işlemi 30 kez tekrarlanarak iki algoritmanın ortak olarak

seçtikleri ve en sık tekrarlanan öznitelikleri kullanan hibrit öznitelik seçim modelinden bahseden bir çalışmada, seçilen özniteliklerden oluşan veri seti üzerinde Naive Bayesian (NB) ve Yapay Sinir Ağları (ANN) algoritmaları eğitilmiştir. PSO-GWO-NB ile %87,7 doğru tespit oranına, %8,8 yanlış pozitif oranına ulaşılmıştır (Mohammad, 2021).

Dört aşamadan oluşan bir metodolojinin önerildiği bir çalışmada, ilk aşamada KDD99, UNSW-NB15, ADFA-LD gibi iyi bilinen veri setleri analiz edilmiş ve ortak öznitelikler belirlenmiştir. Bu ortak öznitelikler önerilen saldırı tespit tekniğinin imza tabanlı model aşamasında, içerik öznitelikleri ise anomali tabanlı model aşamasında kullanılmıştır. İkinci aşamada, iki adımdan oluşan öznitelik seçimi işleminin ilk adımında, eşik değeri bir denklem ile hesaplanmış, eşik değerin altında kalan öznitelikler elenmiş, ikinci adımda ise, bir öznitelik çok sayıda farklı değerden oluşacağı ve aynı değer, hem saldırı hem normal durumlarını karşılayabileceği için öznitelik eşik değerinin üzerinde kalsa bile, içerik analiz edildiğinde saldırı durumunu tespit etmede önemli bir öznitelik olmayabileceği için, özniteliğin atılıp atılmayacağına Feature Selection Approach (FSAP) algoritması ile karar verilmiştir. Önerilen metodolojinin üçüncü aşamasında, imza tabanlı ve anomali tabanlı saldırı tespit modellerinin bir arada kullanıldığı hibrit bir model oluşturulmuştur. Son aşamada ise sınıflandırma algoritmaları ve ön işlemden geçmiş veri setleri ile oluşturulan modeller test veri seti ile test edilmiş %99,65 ve %99,17 doğruluk oranları elde edilmiştir (Özkan Okay ve ark., 2021).

Başka bir çalışmada, filtreleme ve sarmal öznitelik seçim yöntemlerinin bir arada kullanıldığı hibrit bir öznitelik seçim modelinden bahsedilmiştir. Önerilen modelde ilk aşamada filtreleme öznitelik seçim yöntemi olan karşılıklı bilgi (MI) kullanılarak sınıf değişkeni ile en alakalı öznitelikler seçilmiş, çalışmanın ikinci aşamasında ise kullanılan sarmal tabanlı öznitelik seçim yöntemi olan ardışık ileri yönde seçim (SFS) ile ilk aşamada seçilen öznitelikler arasından seçim yapılmış ve hesaplama maliyeti düşürülmüştür. Böylece hem sarıcı yaklaşımların yüksek doğruluğu hem de filtre yaklaşımlarının hız konusundaki verimliliği kullanılmıştır. Sınıflandırma algoritması olarak LS-SVM'nin kullanıldığı çalışmada, KDD99 veri setinden önerilen hibrit

öznitelik seçim yöntemiyle seçilen altı öznitelikten oluşan veri seti ile, %98,90 saldırı tespit oranı ve %0,521 yanlış pozitif oranına ulaşılmıştır (Ambusaidi ve ark., 2014).

Öznitelik seçim algoritmalarının dezavantajlarını ortadan kaldırmak için topluluk öznitelik seçim yöntemlerinin başarısı vurgulanmış, filtreleme, sarmal ve gömülü öznitelik seçim yöntemlerinin örnekleri olan Principal Component Analysis (PCA), GA ve C4.5 tekniklerinin seçtiği özniteliklerin farklı birleşim ve kesişim kombinasyonlarından oluşan veri setleri ile SVM algoritmasının eğitildiğinden, PCA ve GA tekniklerin ortak seçtiği öznitelikler ile en başarılı sonuca ulaşıldığından bahsedilmiştir (Chen ve ark., 2020).

Hibrit öznitelik seçimi ve iki seviyeli sınıflandırıcı topluluğuna dayalı STS geliştirilen bir çalışmada, PSO ve ACO'dan oluşan hibrit bir öznitelik seçim yöntemi kullanımından, özniteliklerin REP Tree (RT) algoritması kullanılarak sınıflandırma performansına göre seçildiğinden, daha sonra RF ve Bagging algoritmalarından oluşan iki seviyeli sınıflandırıcı topluluğu ile modeller oluşturulduğundan, 10 kat çapraz doğrulama tekniği ile değerlendirilen modellerle %85,5 doğruluk, %86,8 duyarlılık ve %88 saldırı tespit oranlarına ulaşıldığından bahsedilmiştir (Tama ve ark., 2019).

Doğru özniteliklerin seçilmesinde, kullanılan veri setindeki örneklerin son derece etkili olduğundan ve geleneksel öznitelik seçimi algoritmalarının değişken boyutlu veri setleri için yeterli olmadığından, bu sorunun çevrimiçi öznitelik seçimi algoritmaları ile çözülebileceğinden bahsedilmiştir (Song, 2016).

Bu çalışmanın ilk aşamasında yapılan literatür taraması sonucunda, geçmiş yıllarda yapılan çalışmaların çoğunda, anomali tabanlı STS geliştirmek için kullanılan veri setlerine uygulanan ön işlemlerde, kategorik veriler sayısallaştırılırken label encoding, ölçeklendirme işlemi için min-max, öznitelik seçimi için ise filtreleme ya da sarmal yöntemlerin kullanımına odaklanıldığı ve makine öğrenimi algoritmalarında parametre ayarı yapılmadığı görülmüştür; fakat son yıllarda yapılan çalışmalarda, kullanılan veri setine en uygun yöntemin seçilebilmesi adına her ön işlem için birden

çok yöntemin denenerek, saldırı tespit başarısı üzerindeki etkilerinin karşılaştırıldığı ve öznitelik seçimi ön işlemi için hibrit yöntemlerin kullanımının, model değerlendirme ve iyileştirme aşamasında parametre ayarının saldırı tespit başarısını arttırdığı görülmüştür. Yine de, kullanılan veri setlerine uygulanan kategorik veri kodlama, ölçeklendirme, öznitelik seçimi ön işlemlerine aynı çalışmada odaklanan ve bu ön işlemlerin saldırı tespit performansı üzerindeki etkisini ayrıntılı olarak inceleyen yeterince çalışma olmadığı fark edilmiştir. Bu nedenle bu çalışmada, bahsedilen ön işlemlere odaklanılarak ve kullanılan veri setine en uygun ön işlemin seçilebilmesi için çok sayıda yöntem ve teknik denenerek, makine öğrenimi yöntemiyle geliştirilen saldırı tespit modelinin başarısının arttırılması hedeflenmiştir.

1.2. Kavramlar ve Tanımlar

- **Bilgisayar Ağı:** İki ya da daha çok bilgisayarın bir iletişim protokolüne göre birbirleri ile veri paylaşımı yapmak için, bir iletişim aracı üzerinden kurdukları bağlantıya denir.
- **Bilgi Güvenliği:** Dijital ortamlardaki bilgilerin gizliliğinin ve yetkisiz erişimlerden korunması, bütünlüğünün bozulmamasını sağlamak için, güvenli bir bilgi işleme platformu oluşturma yönündeki çabalardır.
- **Saldırı:** Bir sisteme yetkisiz erişim sağlayarak, sistemdeki bilgileri değiştirerek, sistemin güvenilmez veya kullanılamaz hale gelmesine neden olan girişimdir (Anderson, 1980).
- **Siber Saldırı:** Bir ağ kaynağının bütünlüğünü, kullanılabilirliğini ve gizliliğini tehdit eden, bilgisayar sistemlerinin güvenlik mekanizmalarını atlatmaya çalışan eylemlerdir.
- **Saldırı Tespiti:** Bilginin gizliliğini, bütünlüğünü ve kullanılabilirliğini tehlikeye atabilecek, bilgisayarın ve ağın güvenlik mekanizmalarını atlatma girişimlerinin analiz ve tespit edilmesi sürecidir.

- **Makine Öğrenimi (Machine Learning) :** İnsanların öğrenme tarzını, veri örneklerini ve algoritmaları kullanarak taklit eden, doğruluğunu aşamalı olarak arttıran bir bilgisayar bilimi dalıdır.
- **Model:** Eğitim veri örnekleri ile eğitilen makine öğrenimi algoritmaları kullanılarak, eğitim verisinde yer alan ya da almayan veriler hakkında tahmin yürüten yapay zekâdır.
- **Aşırı Öğrenme (Overfitting):** Modelin, eğitim verileri üzerindeki gürültüyü ve ayrıntıları, modelin yeni veriler üzerinde performansını düşürecek ölçüde iyi öğrenmesi, ezberlemesidir.
- **Parametre:** Modele dâhil olan ve değeri verilerden tahmin edilen bir yapılandırma değişkenidir. Modelin tahmin gücünü arttırmak veya modeli eğitmeyi kolaylaştırmak için kullanılır.
- **Hiper parametreler:** Modelden maksimum performans alınmasını sağlayan parametre kombinasyonuna denir.
- **Bağlantı Kaydı:** Bir veri setinin yer aldığı tabloda her bir satırdaki bilgiye denir.
- **Öznitelik (Features/Attributes) :** Veri setindeki her bir bağlantıya ait bağımsız değişkenler, özelliklerdir.
- **Sınıf Etiketi (Labels) :** Veri setindeki bağlantı kaydının sınıfını belirleyen öznitelik, bağımlı değişkendir. Sınıf etiketi, sınıf değişkeni ya da hedef değişken olarak isimlendirilir.
- **Eğitim Veri Seti (Training Dataset):** Makine öğrenimi algoritmalarının tahminde bulunması için eğitilmesinde kullanılan veri örneği.
- **Test Veri Seti (Testing Dataset) :** Eğitim veri seti ve makine öğrenimi algoritmaları kullanılarak oluşturulan modelin başarısının, çeşitli değerlendirme metrikleri ile test edilmesi için kullanılan veri örneği.
- **Doğrulama Veri Seti (Validation Dataset):** Modelin parametre ayarı için kullanılan veri örneği.

1.3. Bilgi Güvenliği ve Siber Saldırılar

Veri, birbiriyle bağlantısı henüz kurulmamış, dijital ortamlarda taşınan bit dizeleri olarak tanımlanabilir (Canbek ve Sağıroğlu, 2006). Bilgi ise işlenmiş veridir, yorumlanabilir ve bir fikir ya da görüş belirtir. Türk Standartları Enstitüsü TSE tarafından yayınlanan TS ISO/IEC 27001:2017 Bilgi Güvenliği Yönetim Sistemi Standardı, bilgi güvenliğini üç başlık altında inceler:

- Gizlilik: Bilgilerin yetkisiz erişime karşı korunması.
- Bütünlük: Bilgilerin eksiksiz, tam, tutarlı ve doğru olması.
- Kullanılabilirlik: Bilgilerin, yetkili kişilerce ihtiyaç duyulduğunda erişilebilir olması.

Bilginin güvende olduğundan söz edebilmemiz için bu üç unsurun da sağlanması gerekir. Çalışmamızda söz konusu bilgilerin saklanma ortamı bilgisayar ve ağ olduğundan bilgi güvenliği ağın güvenliğine bağlıdır. Ağ güvenliği ise, ağdaki bilgilerin güvenli depolanmasını, taşınmasını ve uygulanmasını kapsar (Bernaschi ve ark, 1999).

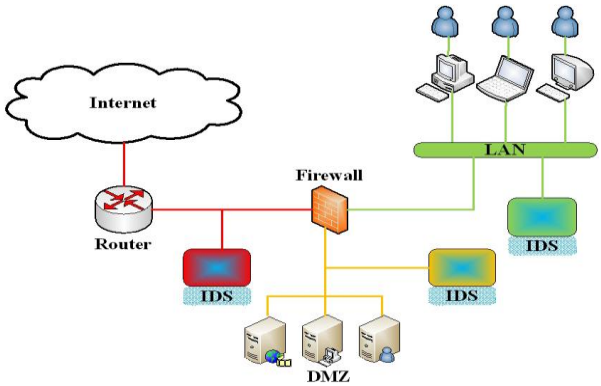
Siber saldırılar, bir ağ kaynağının bütünlüğünü, kullanılabilirliğini ve gizliliğini tehdit eden, bilgisayar sistemlerinin güvenlik mekanizmalarını atlatmaya çalışan eylemlerdir (Singh ve Silakari, 2009). Bir siber saldırının örnek anatomisi, aşağıda yer alan adımlardan oluşur (Bou-Harb ve Debbabi, 2014):

1) Siber tarama
2) Numaralandırma
3) İzinsiz giriş girişimi
4) Ayrıcalık yükselmesi
5) Kötü amaçlı eylemlerin gerçekleştirilmesi
6) Kötü amaçlı yazılımların sisteme dağıtılması
7) Adli delillerin silinmesi ve sistemden çıkış

Birçok siber saldırı türü, bu basit olay dizisini takip eder. Bir saldırı modeline uyan bir dizi olay görülürse, saldırının modele göre devam edeceği varsayılabilir, böylece saldırganın bir sonraki adımı tahmin edilebilir ve benzer davranışları sergileyen bu saldırıları tespit edebilecek STS'ler geliştirilebilir; fakat sıfır gün saldırıları, saldırganlarca yeni keşfedilen siber saldırılardır. Yararlanılan güvenlik açığı henüz bilinmediği için etkilenen yazılım yamalanamaz ve imza tabanlı STS'ler saldırıyı tespit edemez. Sıfır gün saldırılarının tespit edilebilmesi için daha önce karşılaşılmış ya da suni olarak üretilmiş saldırı verileri ile eğitilmiş makine öğrenimi algoritmaları kullanılarak saldırı tespit modelleri geliştirilmektedir.

1.4. Saldırı Tespit Sistemleri (STS)

STS, bir bilgisayar sisteminde veya ağ ortamında bilgi güvenliğini tehlikeye atacak her türlü faaliyetin analiz ve tespit sürecini otomatikleştiren yazılım ve de donanım ürünleridir (Bace ve Mell, 2011). STS'ler bilgisayar sistemindeki veya ağdaki cihazlara gelen ve giden tüm trafiği izlemeleri için, Yerel Alan Ağı (LAN) ve Savunmasız Bölge (DMZ) arasına ya da internet alanına yerleştirilirler. Şekil 1.1'de STS konumlandırılmış bir ağ gösterilmektedir.



Şekil 1.1. Saldırı tespit sisteminin genel yapısı (Song, 2016).

Saldırıları analiz etme stratejisine göre STS'ler, imza tabanlı (kötüye kullanım) ve anomali (davranış) tabanlı olmak üzere iki gruba ayrılır.

1.4.1. İmza Tabanlı STS

Bu yaklaşımda, STS'nin daha önce karşılaştığı saldırı verileri kullanılarak bir veri tabanı oluşturulur ve yeni gelen hizmet isteği, bu veri tabanındaki herhangi bir örnek ile eşleşirse saldırı olarak tespit edilir. İmza tabanlı STS'lerde dikkat edilmesi gereken noktalar, daha önce karşılaşılmış saldırıları tespit etmek ve önceden tanımlanmış eylemlerle yanıt vermek için kullanılan bir dizi kural olan imzaların, saldırıların değişmesine paralel olarak konfigüre edilmesi gerektiği ve de yanlış alarm verme olasılığını arttırdığı için çok yüksek boyutta veri tabanı kullanmaktan kaçınılması gerektiğidir.

İmza tabanlı STS'lerin avantajları: Tespit hızlarının yüksek olması, veri tabanı boyutu ayarlandığı sürece yanlış alarm üretimi sayısının az olması ve sistem yöneticisi tarafından hangi saldırı tipinin alarma neden olacağının kolaylıkla anlaşılabilir, böylece imzaların kolaylıkla güncellenebilir olmasıdır.

İmza tabanlı STS'lerin dezavantajları: Sıfır gün saldırıları adı verilen henüz karşılaşılmamış saldırıları tanımlamakta başarısız olmalarıdır.

1.4.2. Anomali Tabanlı STS

Anomali tabanlı STS'ler, normal çalışma süresi boyunca toplanan geçmiş verilerden, ağ trafiği içerisindeki anormal ve normal davranışları temsil eden dinamik profiller çıkarırlar ve yeni gelen hizmet isteğini, oluşturulan normal davranış profilinden sapıyor ise saldırı olarak tespit ederler. Anomali tabanlı STS'lerde kullanılan yöntemler istatistiksel, veri madenciliği, makine öğrenimi ve bilgi tabanlı yöntemlerdir. Bu çalışmada anomali tabanlı STS geliştirilirken, makine öğrenimi yöntemi kullanılacaktır.

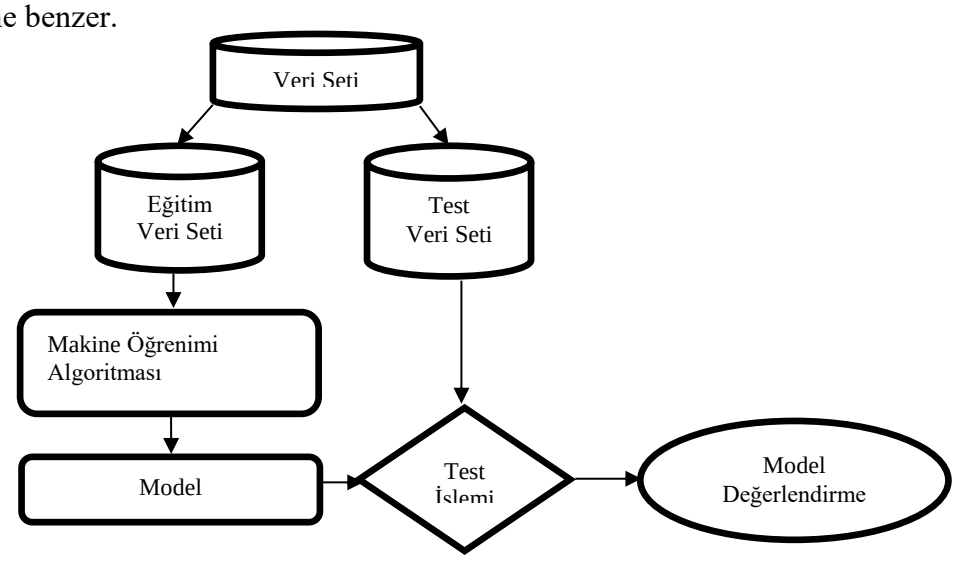
Makine öğrenimi tabanlı tespit yöntemi: Makine öğrenimi, bir programın veya sistemin belirli bir görev grubundaki performanslarını zaman içinde öğrenme ve iyileştirme yeteneği olarak tanımlanabilir. Makine öğrenimi teknikleri, önceki sonuçlara dayalı olarak performansını arttıran bir sistem oluşturmaya odaklanır ve yeni edinilen bilgilere dayanarak yürütme stratejilerini değiştirme yeteneğine sahiptir (Bou Harb ve ark, 2014).

Anomali tabanlı STS'lerin avantajları: Sıfır gün saldırılarını tespit edebilme yetenekleri ve ticari alanda yaygın olarak kullanılan imza tabanlı STS'lerde imzaları tanımlamakta kullanılacak bilgiler üretmeleridir.

Anomali tabanlı STS'lerin dezavantajları: Normal kullanıcı ve sistem davranışı kalıpları çok hızlı değişiklik gösterebileceği için çok sayıda yanlış alarm üretmeleri, başarılı, dinamik profiller oluşturabilmek için kapsamlı veri setleri gerektirmelerine rağmen, bu veri setlerinin oluşturulmasının oldukça maliyetli ve zahmetli olmasıdır.

1.5. Anomali Tabanlı STS Geliştirme Süreci

Anomali tabanlı STS geliştirme süreci en basit hali ile temel makine öğrenimi sürecine benzer.



Şekil 1.2. Temel makine öğrenimi süreci.

Şekil 1.2'de görüldüğü gibi anomali tabanlı STS modeli geliştirme sürecinde de, kullanılan veri seti eğitim ve test veri seti olarak ayrılır. Eğitim veri seti ile makine öğrenimi algoritması eğitilir ve model oluşturulur. Modelin yeni veriyi tahmin başarısı test veri seti ile değerlendirilir.

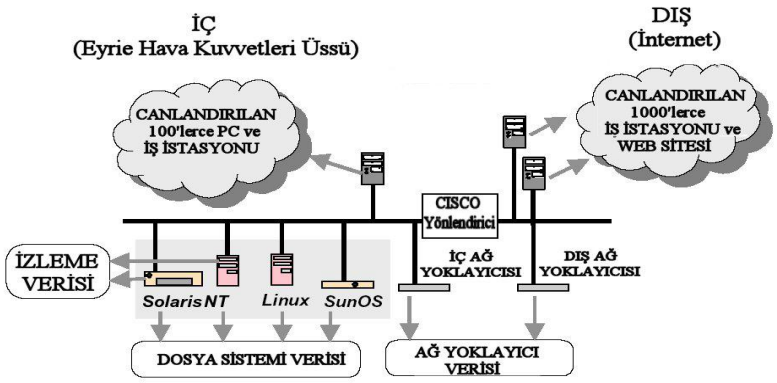
1.6. STS Çalışmalarında Kullanılan Veri Setleri

STS veri seti, STS tasarlanırken kullanılan makine öğrenimi algoritmasının verileri tanımasında yani eğitiminde, eğitim sonucu oluşturulan modelin test edilmesinde ve modelin iyileştirilmesi için parametre ayarı yapılmasında kullanılan, saldırı ve normal veri örneklerinden oluşan settir. Bir veri seti oluşturmak zahmetli ve oldukça maliyetlidir. Bu yüzden araştırmalarda kullanılmak üzere çeşitli simülasyon ortamlarında veri setleri oluşturulmuştur. Bir STS veri setinde her satır bir bağlantı kaydını, her bir sütun ise o bağlantı kaydına ait öznitelikleri içerir. Öznitelikler ise girdi değişkenler ve çıktı değişken olarak iki kısımda incelenir. Girdi değişkenler veri örneğine ait öznitelikleri içeren bağımsız değişkenler, çıktı değişken ise veri örneğinin sınıfını (normal, anormal) belirleyen bağımlı değişkendir. Bu çalışmada girdi değişken için öznitelik, çıktı değişken için sınıf değişkeni terimleri kullanılacaktır. Bu bölümde literatürdeki çalışmalarda isimleri sıkça geçen DARPA, KDD99, NSL-KDD veri setlerinden bahsedilecektir.

1.6.1. 1999 DARPA Veri Seti

DARPA 1999 veri seti, 1998'de geliştirilen veri setine yeni saldırı verilerinin eklenmesi ile oluşturulmuştur. MIT Lincoln laboratuvarlarında yedi hafta boyunca toplanan verilerden oluşan DARPA veri seti hazırlanırken, saldırıların hedefi olan iç ağ ve saldırıların yapıldığı dış ağ olmak üzere iki ağ kurulmuştur. İç ve dış ağdaki trafik üreteçleri kullanılarak yüzlerce sunucu ile internet kullanıcılarının davranışları benzetim yapılarak, ağa birbirine karıştırılmış saldırı verileri ve normal veriler

gönderilmiş, ağ izlenmiştir. Son olarak veriler toplanıp analiz edilmiştir (Tanrıkulu, 2009). Şekil 1.3'te 1999 DARPA ağ simülasyonu gösterilmektedir.



Şekil 1.3. 1999 DARPA ağ simülasyonu (Aydın ve Örencik, 2005).

1.6.2. KDD99 Veri Seti

KDD99 veri seti, DARPA veri setinin bir takım ön işlemlerden geçirilerek sadeleştirilmiş bir versiyonudur. Bu şekilde eğitim ve test işlemleri daha hızlı gerçekleştirilebilmektedir. KDD99 veri setinde eğitim verisi, yedi hafta boyunca ağ üzerinden toplanan, her biri 100 bayt olan beş milyon civarı bağlantı kaydından oluşur. Test verisi ise iki hafta boyunca ağ üzerinden toplanan iki milyon civarı bağlantı kaydından oluşur. Her bağlantı saldırı ya da normal olarak sınıflandırılır. Bir saldırı 39 saldırı türünden oluşan dört kategoriye ayrılır. Yedi haftalık eğitim veri seti 22 saldırı türü içerirken, test veri seti, eğitim setinde bulunmayan 17 saldırı türü içerir. Bir saldırı aşağıdaki dört kategoriden birinde yer alır:

1) Hizmet Reddi Saldırısı (DoS): Sisteme cevap verebileceğinden daha çok istek yollayarak, sistemin hizmet vermesini engellemek amacı ile yapılan saldırılardır.

2) Uzaktan Yerel Saldırı (R2L): Saldırganın ağ üzerinden paketler göndererek sisteme erişim kazanmak için yaptığı saldırılardır.

3) Kullanıcıdan Köke Saldırı (U2R): Normal kullanıcı yetkilerine sahip iken sistemin güvenlik açıklarından faydalanarak yönetici yetkilerine sahip olmak için yapılan saldırılardır.

4) Bilgi Toplama Saldırısı (Probe): Saldırganın yapacağı saldırıyı belirleyebilmek için, bir bilgisayar sistemi veya ağ ile ilgili bilgi toplamak için yaptığı saldırılardır.

KDD99 veri setindeki saldırı sınıfları ve türleri Çizelge 1.1'de, saldırı sınıflarının eğitim ve test veri setlerindeki yüzdelik dağılımları ise Çizelge 1.2'de gösterilmektedir.

Çizelge 1.1. KDD99 veri setindeki saldırı sınıfları ve türleri.

Saldırı Sınıfı	Açıklama	Saldırı Türü
Hizmet Engelleme (Denial of Service-Do)	Servisleri hizmet veremez duruma getirme	Back,land,neptune,pod,smurf,teardrop.
Uzaktan Yerel Saldırı (Remote to Local- R2L)	Sistemdeki kullanıcı yetkilerini ele geçirme	Ftp write, guess passwd,imap,multihop, phf,spy,warezclient,warezmaster.
Kullanıcıdan Köke Saldırı(User to Root-U2R)	Normal kullanıcı hesabından root hesabına geçme	Buffer overflow,perl,loadmodule,rootkit.
Bilgi Tarama(Probing)	Bilgisayar veya ağ bilgisini elde etmek	İpsweep,nmap,portsweep,satan

Çizelge 1.2. KDD99 veri setinde saldırı sınıflarının eğitim ve test veri setindeki yüzdelik dağılımları (Devi ve Abualkibash, 2019).

Sınıf	Eğitim Veri Seti	Yüzdelik	Test Veri Seti	Yüzdelik
Normal	812.814	75.611%	60.593	19.481%
DoS	247.267	23.002%	229.853	73.901%
R2L	999	0.093%	16.189	5.205%
U2R	52	0.005%	228	0.073%
Probe	13.860	1.289%	4.166	1.339%

1.6.3. Çalışmamızda Kullanılacak NSL-KDD Veri Seti

KDD99 veri seti üzerinde makine öğrenimi algoritmalarının daha yüksek performansta çalışabilmeleri için, tekrar eden bağlantı kayıtları silinerek, veri boyutu düşürülmüş ve NSL-KDD veri seti elde edilmiştir. Bu şekilde veri setinin, özellikle hesaplama maliyetinden dolayı parçalara bölünmesi zorunluluğu da ortadan kaldırılmıştır (Özgür ve Erdem, 2018). KDD99 eğitim veri seti 4898431 bağlantı

kaydı, NSL-KDD eğitim veri seti ise 125972 bağlantı kaydı içermektedir. Her bağlantı kaydı dokuzu temel, 32'si türetilmiş 41 tane özniteliğe ve bağlantının normal ya da anormal olduğunu belirleyen bir de sınıf değişkenine sahiptir. Bu 41 öznitelik aşağıdaki gibi dört kategoriye ayrılarak ifade edilmiştir (Song, 2016).

1) Temel öznitelikler: Bir TCP/IP bağlantısından çıkarılan tüm öznitelikleri içerir.

2) İçerik öznitelikleri: Saldırı olarak nitelendirilebilecek bir davranışı belirleyen özniteliklerdir. Örneğin başarısız oturum açma girişimlerinin sayısı.

3) Zamana bağlı trafik öznitelikleri: Son iki saniye içerisinde aynı sunucuya ve aynı servise yapılan bağlantıların özniteliklerini içerir.

4) Bağlantı tabanlı trafik öznitelikleri: İki saniyeden uzun süren ve zamana bağlı trafik öznitelikleri ile tespit edilemeyen saldırılar için, aynı sunucuya ve aynı servise yapılan bağlantılar incelenirken, bir zaman aralığı yerine her 100 bağlantıdan oluşan bir aralık kullanılarak elde edilen öznitelikleri içerir.

41 özniteliğin dört kategoriye ayrılarak öznitelik numarası, öznitelik adı, tanımı ve değer türü olarak gösterimi Çizelge 1.3'te gösterildiği gibidir.

Çizelge 1.3. NSL-KDD veri seti öznitelikleri (Stolfo ve ark., 2000).

Temel Öznitelikler			
Öznitelik no	**Öznitelik adı**	**Tanımı**	**Değer türü**
1	duration	Bağlantı süresi	Sürekli
2	protocol_type	Bağlantılarda kullanılan protokol türü (udp,tcp,vs.)	Ayrık
3	service	Hedef ağın kullandığı servis (ftp vs.)	Ayrık
4	flag	Bağlantının normal ya da hatalı olma durumu	Ayrık
5	src_bytes	Tek bağlantı ile kaynaktan hedefe aktarılan byte sayısı	Sürekli
6	dst_bytes	Tek bağlantı ile hedeften kaynağa aktarılan byte sayısı	Sürekli
7	land	Kaynak ve hedef IP ve portların aynı olup olmaması durumu (1 ya da 0)	Ayrık
8	wrong_fragment	Bağlantıdaki yanlış parça sayısı	Sürekli
9	urgent	Bağlantıdaki acil paket sayısı	Sürekli

Çizelge 1.3. Devam.

İçerik Öznitelikleri			
Öznitelik no	**Öznitelik adı**	**Tanımı**	**Değer türü**
10	Hot	Bir sistem dizini girme, programlar oluşturma ve çalıştırma sayısı	Sürekli
11	num_failed_logins	Başarısız giriş denemelerinin sayısı	Sürekli
12	logged_in	Sisteme girişin başarılı olup olmama durumu (1 ya da 0)	Ayrık
13	num_compromised	Uzlaşılmış koşul sayısı	Sürekli
14	root_shell	Root shell olup olmama durumu (1 ya da 0)	Ayrık
15	su_attempted	Yönetici hesabı ele geçirme durumu (1 ya da 0)	Ayrık
16	num_root	Bağlantıda root erişim sayısı	Sürekli
17	num_file_creations	Bağlantıda dosya oluşturma işlemleri sayısı	Sürekli
18	num_shells	Sistem kabuğuna erişim sayısı	Sürekli
19	num_access_files	Erişim kontrolü dosyaları üzerindeki işlem sayısı	Sürekli
20	num_outbound_cmds	Bir ftp oturumunda giden komut sayısı	Sürekli
21	is_hot_login	Yönetici olarak oturum açıp açmama durumu (1 ya da 0)	Ayrık
22	is_guest_login	Misafir olarak oturum açıp açmama durumu (1 ya da 0)	Ayrık

Zaman Bağlı Trafik Öznitelikleri			
Öznitelik no	**Öznitelik adı**	**Tanımı**	**Değer türü**
23	count	Son iki saniyede aynı sunucuya yapılan birbirinin aynı olan bağlantı sayısı	Sürekli
Not: aşağıdaki öznitelikler aynı ana bilgisayar bağlantılarına ilişkindir.			
24	srv_count	Son iki saniyede aynı servise yapılan birbirinin aynı olan bağlantı sayısı	Sürekli
25	serror_rate	"SYN" hata bağlantılarının yüzdesi	Sürekli
26	rerror_rate	"REJ" hata bağlantılarının yüzdesi	Sürekli
27	same_srv_rate	Aynı servise olan bağlantıların yüzdesi	Sürekli
28	diff_srv_rate	Farklı servislere olan bağlantıların yüzdesi	Sürekli
Not: Aşağıdaki öznitelikler bu aynı hizmet bağlantılarına ilişkindir.			
29	srv_serror_rate	"SYN" hata bağlantılarının yüzdesi	Sürekli
30	srv_rerror_rate	"REJ" hata bağlantılarının yüzdesi	Sürekli
31	srv_diff_host_rate	Farklı servislere olan bağlantıların yüzdesi	Sürekli

Bağlantı Tabanlı Trafik Öznitelikler			
Öznitelik no	**Öznitelik adı**	**Tanımı**	**Değer türü**
32	dst_host_count	Aynı hedef ana bilgisayar IP adresine sahip olan bağlantı sayısı	Sürekli
33	dst_host_srv_count	Aynı port numarasına sahip olan bağlantı sayısı	Sürekli
34	dst_host_same_srv_rate	Dst_host_count'ta toplanan bağlantılar arasında aynı hizmete ait olan bağlantıların yüzdesi	Sürekli
35	dst_host_diff_srv_rate	Dst_host_count'ta toplanan bağlantılar arasında farklı servislere ait olan bağlantıların yüzdesi	Sürekli
36	dst_host_same_src_port_rate	Dst_host_srv_count'ta toplanan bağlantılar arasında aynı kaynak bağlantı noktasına olan bağlantıların yüzdesi	Sürekli
37	dst_host_srv_diff_host_rate	Dst_host_srv_count'ta toplanan bağlantılar arasında farklı hedef makinelere yapılan bağlantıların yüzdesi	Sürekli
38	dst_host_serror_rate	Dst_host_count içinde toplanan bağlantılar arasında bayrak s0,s1,s2 veya s3 işaretini etkinleştiren bağlantıların yüzdesi	Sürekli

Çizelge 1.3. Devam.

39	dst_host_srv_serror_rate	Dst_host_srv_count içinde toplanan bağlantılar arasında bayrak s0, s1, s2 veya s3 işaretini etkinleştiren bağlantıların yüzdesi	Sürekli
40	dst_host_rerror_rate	Dst_host_count içinde toplanan bağlantılar arasında, REJ bayrağını etkinleştiren bağlantıların yüzdesi	Sürekli
41	dst_host_srv_rerror_rate	Dst_host_srv_count içinde toplanan bağlantılar arasında, REJ bayrağını etkinleştiren bağlantıların yüzdesi	Sürekli
42	**Class**	**Bağlantı kaydının sınıf değişkeni (normal ya da anormal)**	**Ayrık**

NSL-KDD veri seti Çizelge 1.4' te görüldüğü gibi sekiz bölümden oluşur.

Çizelge 1.4. NSL-KDD veri seti içerisindeki dosyaların listesi ve açıklamaları.

Sıra no:	**Dosya adı**	**Tanımı**
1	KDDTrain+.arff	arff formatında ikili (normal,anormal) etiketlendirilmiş NSL-KDD tam eğitim seti
2	KDDTrain+.txt	txt formatında saldırı sınıfları (dos,u2r,r2l,probe) ile etiketlendirilmiş NSL-KDD tam eğitim seti
3	KDDTrain+_20Percent.arff	KDDTrain+.arff dosyasının %20'lik bir alt kümesi
4	KDDTrain+_20Percent.txt	KDDTrain+.txtdosyasının %20'lik bir alt kümesi
5	KDDTest+.arff	arff formatında ikili etiketlendirilmiş NSL-KDD tam test seti
6	KDDTest+.txt	txt formatında saldırı sınıfları(dos,u2r,r2l,probe) ile etiketlendirilmiş NSL-KDD tam test seti
7	KDDTest-21.arff	KDDTest+.arff dosyasının bir alt kümesi. Eğitim ve test veri setlerinde bulunmayan saldırı türleri (processtable, mscan, snmpguess, snmpgetattack, saint,apache2, HTptunnel, back, mailbomb) içerir.
8	KDDTest-21.txt	KDDTest+.txt dosyasının bir alt kümesi. Eğitim ve test veri setlerinde bulunmayan saldırı türleri (processtable, mscan, snmpguess, snmpgetattack, saint, apache2, HTptunnel, back, mailbomb) içerir.

NSL-KDD veri setindeki dört saldırı sınıfı ve normal verilerin eğitim ve test veri setlerindeki dağılım yüzdeleri Çizelge 1.5'te gösterildiği gibidir.

Çizelge 1.5. NSL-KDD veri setinde saldırı sınıfları ve normal bağlantı örneklerinin eğitim ve test veri setlerindeki yüzdelik dağılımları (Devi ve Abualkibash, 2019).

Sınıf	**Eğitim veri seti**	**Yüzde**	**Test veri seti**	**Yüzde**
Normal	67,342	53,458%	9,710	43,075%
Dos	45,927	36,458%	7,457	33,080%
Probe	11,656	9,253%	2,421	10,740%
R2L	995	0,790%	2,754	12,217%
U2R	52	0,041%	200	0,887%
Total	125,972	100%	22,542	100%

Bu bölümde, anomali tabanlı STS geliştirirken veri setine uygulanan ön işlemlere odaklanan literatürdeki çalışmalar incelenmiş ve kullanılan veri setlerine uygulanan ön işlemlere aynı anda odaklanan yeterince çalışma olmadığı görüldüğü için; kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemlerinin üçüne de odaklanarak, kurulan saldırı tespit modelinin başarısını arttırmayı hedefleyen bir çalışma yapılmak istendiğinden bahsedilmiştir. Bölümün diğer kısımlarında ise, yapılan incelemeler ve tespitlerin anlaşılırlığını kolaylaştırmak için STS çalışmalarında kullanılan genel teorik bilgiler verilmiştir.

2.GEREÇ VE YÖNTEM

Bu tez çalışmasında yazılım materyali olarak, Numpy, Scipy, StatsModels, Pandas, Matplotlib, nltk ve scikit-learn gibi birçok veri bilimi kütüphanelerinin olması ve bu sayede veri analizinde son derece başarılı olması sebebiyle tüm işlemler Python dili kullanılarak gerçekleştirilmiştir. Python kodların yazımı ve çalıştırılması için ise, açık kaynaklı bir IDE olan Spyder kullanılmıştır. Donanım materyali olarak da, aşağıdaki özelliklere sahip dizüstü bilgisayar kullanılmıştır.

- İşlemci: Intel Core i5-1135G7 (8M Cache, 2.40 GHz, up to 4.20 GHz)
- RAM Bellek: DDR IV 8GB (8GBx1, 3200MHz)
- Disk Kapasitesi: 512 GB NVMe PCIe Gen3x4 SSD
- İşletim Sistemi: Windows 10 Professional 64 Bit

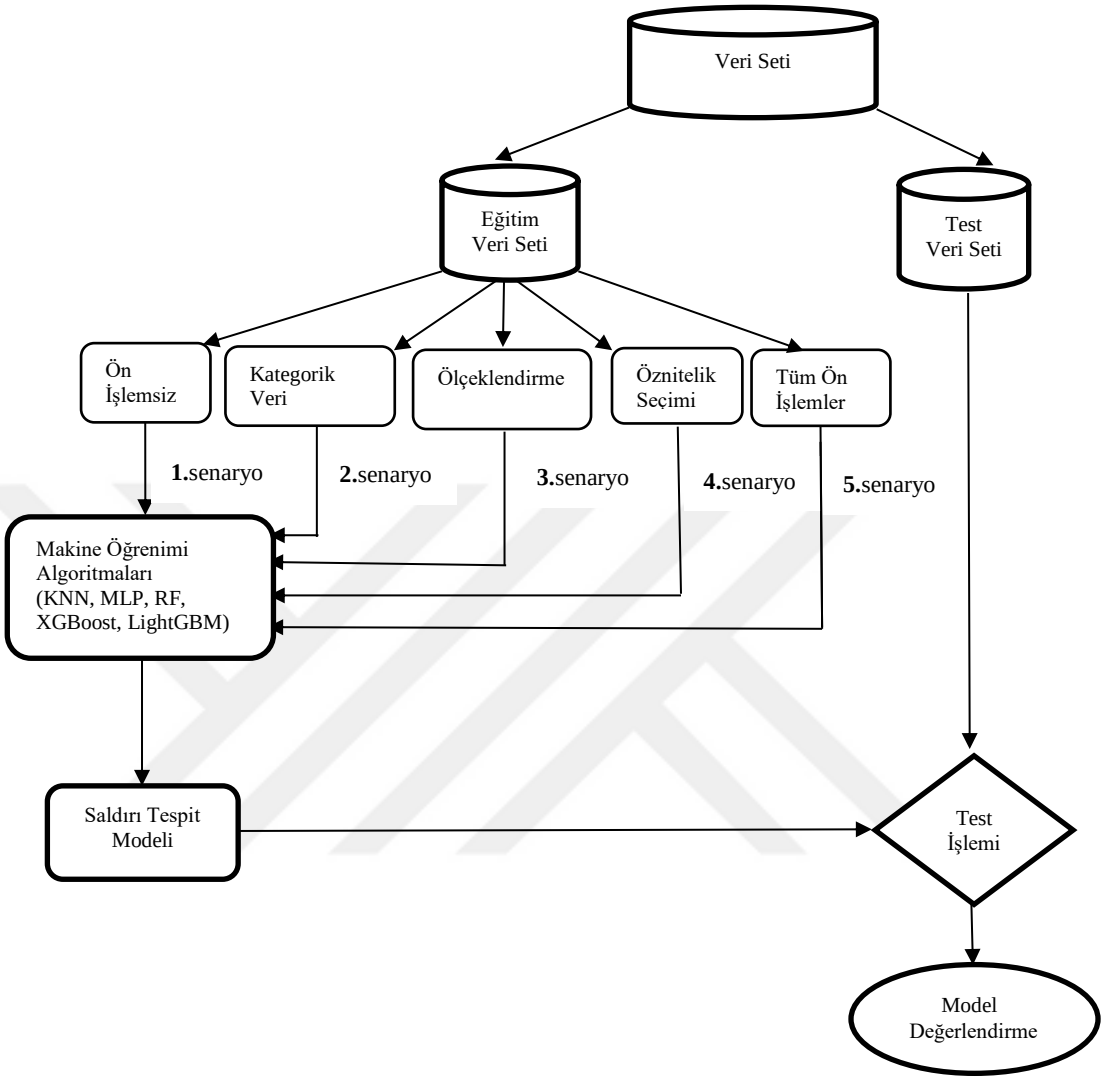
Tez çalışmasında, anomali tabanlı STS modellerinde kullanılan veri setleri üzerinde yapılan ön işlemlerin, STS modeli üzerindeki başarısını incelemek ve de saldırı tespit başarısı ve hızı yüksek anomali tabanı bir saldırı tespit modeli geliştirme hedefini gerçekleştirmek için, makine öğrenimi yöntemi kullanılmıştır. Çalışmada anomali tabanlı saldırı tespit modeli geliştirilirken üç aşamalı bir metodoloji önerilmiştir:

1) Veri Setinin Ön İşlenmesi,

2) Ön İşlenmiş Veri Setleri ve Makine Öğrenimi Algoritmaları ile Saldırı Tespit Modelleri Oluşturulması,

3) Modellerin Değerlendirilmesi.

Önerilen metodolojinin akış şeması Şekil 2.1'de görüldüğü gibidir.



Şekil 2.1. Önerilen yöntemin akış şeması.

Çalışmada önerilen metodoloji beş farklı senaryo ile uygulanmıştır. Şekil 2.1'de gösterildiği gibi ilk senaryoda eğitim veri setine ön işlem uygulanmamıştır. İkinci senaryoda eğitim veri setine kategorik veri kodlama ön işlemi; üçüncü senaryoda, eğitim veri setine ölçeklendirme ön işlemi; dördüncü senaryoda eğitim veri setine öznitelik seçimi ön işlemi uygulanmıştır. Beşinci ve son senaryoda ise, eğitim veri setine kategorik veri kodlama ve ölçeklendirme ön işlemleri uygulandıktan sonra elde edilen veri setinden öznitelik seçimi yapılmıştır. Her senaryo sonucu elde edilen veri setleri ve KNN, MLP, RF, XGBoost, LightGBM algoritmaları ile saldırı tespit modelleri oluşturulmuştur. Oluşturulan modellerin başarısı, eğitim setinden farklı

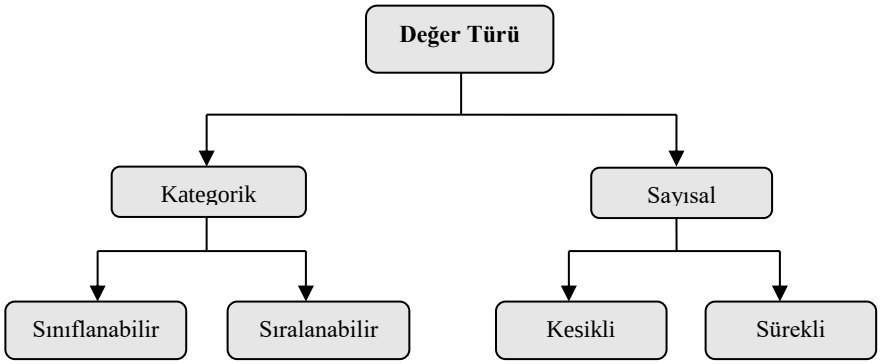
sayıda ve türde saldırı kayıtlarına sahip olan test veri seti kullanılarak değerlendirilmiştir.

2.1. Veri Setinin Ön İşlenmesi

Eğitim için kullanılan veri setindeki verilerin dengesiz dağılması, kategorik değerli veriler, her bir bağlantı örneğine ait öznitelik sayısı gibi nedenler makine öğrenimi algoritmalarının sınıflandırma performansını düşürebilir. Önerilen yöntemin birinci aşamasında, çalışmanın ikinci aşamasında kullanılacak makine öğrenimi algoritmalarında, maksimum düzeyde sınıflandırma performansına ulaşabilmek için, NSL-KDD veri setine sırasıyla kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemleri uygulanmıştır.

2.1.1. NSL-KDD Veri Setinde Kategorik Verilerin Sayısallaştırılması

Veri setlerinde kullanılan en yaygın iki değer türü sayısal ve kategoriktir. Sayısal değer türüne sahip veriler aralarında matematiksel ya da mantıksal ilişki kurulabilen verilerdir. Kategorik değer türüne sahip veriler ise, aralarında matematiksel ya da mantıksal bir ilişki bulunmayan, dolayısıyla ölçülemeyen ve üzerinde sayısal işlem yapılamayan verilerdir. Kategorik veri türü sınıflandırılabilir (nominal) ve sıralanabilir (ordinal) olmak üzere ikiye ayrılır. Sıralanabilir veri türleri kendi içerisinde hiyerarşik bir yapı bulundurur. Sınıflandırılabilir veri türlerinde ise bu şekilde bir yapı söz konusu değildir.



Şekil 2.2. Değer türleri.

NSL-KDD veri setinde her bir bağlantı kaydına ait 42 özniteliğin üç tanesi kategorik, altı tanesi ikili (binary), 33 tanesi sayısal (23 tanesi kesikli ve 10 tanesi sürekli) değerlerden oluşur (Çizelge 2.1). Ancak çoğu makine öğrenimi algoritması kategorik değerleri aralarında matematiksel ya da mantıksal bir ilişki bulunmadığı için işleyemez. Bu yüzden ön işleme adımında kategorik değerler sayısal değerlere dönüştürülmelidir. Kategorik değerlerin makine öğrenimi algoritmalarının işleyebileceği sayısal değerlere dönüştürülmesi işlemine kategorik veri kodlama denir.

Çizelge 2.1. NSL-KDD veri setindeki özniteliklerin değer türleri.

Değer Türü	**Öznitelik ve Numarası**
Kategorik	protocol_type(2), service(3), flag(4)
İkili	land(7), logged_in(12), root_shell(14), su_attempted(15), is_host_login(21), is_guest_login(22)
Sayısal	duration(1), src_bytes(5), dst_bytes(6), wrong_fragment(8), urgent(9), hot(10), num_failed_logins(11), num_compromised(13), num_root(16), num_file_creations(17), num_shells(18), num_access_files(19), num_outbound_cmds(20), count(23) srv_count(24), serror_rate(25), srv_serror_rate(26), rerror_rate(27), srv_rerror_rate(28), same_srv_rate(29) diff_srv_rate(30), srv_diff_host_rate(31), dst_host_count(32), dst_host_srv_count(33), dst_host_same_srv_rate(34), dst_host_diff_srv_rate(35), dst_host_same_src_port_rate(36), dst_host_srv_diff_host_rate(37), dst_host_serror_rate(38), dst_host_srv_serror_rate(39), dst_host_rerror_rate(40), dst_host_srv_rerror_rate(41)

2.1.1.1. Kategorik Veri Kodlama Teknikleri

Veri setlerindeki kategorik verileri sayısallaştırmak için çeşitli kodlama teknikleri vardır. Kodlama tekniğini seçerken, veri setindeki kategorik öznitelik sayısı, her bir özniteliğin değişken sayısı, dolayısıyla eğitim ve test süresinin kurulacak model için önemi, veri setiyle yapılacak işlemlerde bilgi kaybının sonucu ne denli etkileyeceği, kurulacak modelin aşırı öğrenmeye ne denli dirençli olması gerektiği gibi nedenler etkilidir.

Label Encoding (Etiket Kodlama): Kategorik veriler arasında bir sıralama ilişkisi olduğunda kullanılır. Her kategorik değere benzersiz bir tam sayı değeri atanır. Öznitelik sayısını değiştirmez. Veriler arasında herhangi bir sıralama ilişkisi yoksa label encoding'i kullanmak uygun değildir. Çünkü değerlerin neye göre atandığı belirsizdir ve makine öğrenimi algoritması kurulmaması gereken bir bağlantı kurmaya çalışacak, bu da yanlış çıkarımlara yol açacaktır.

One Hot Encoding (Tek Sıcak Kodlama): Kategorik veriler arasında herhangi bir ilişki söz konusu olmadığında kullanılır. Kategorik verinin her bir değişkeninin, o kategorinin varlığını veya yokluğunu temsil eden bir vektörle, 0 veya 1 ile eşlendiği tekniktir. Bu teknik ile kategorik verinin değişken sayısı kadar kukla değişken oluşturulur (kukla değişken tuzağından kaçınmak için bir sütun kaldırılır), dolayısıyla öznitelik sayısı artar. Uygulanmasının basit olması ve kategorik verinin değişken sayısı az olduğunda başarılı olması tekniğin avantajlarıdır. Çok değişkenli kategorik verilere sahip veri setlerinde, değişken sayısı kadar sütun yani öznitelik eklenmesi ile kaynak tüketiminin artması, eğitim ve test süresinin uzaması, kullanılan algoritmanın verimliliğinin düşmesi, boyutluluk lanetine yol açması nedenleri tekniğin dezavantajlarıdır. Başka bir dezavantajı ise, one hot encoding ile kodlanmış iki kategorik değişken arasındaki öklid mesafesinin, bize yalnızca iki değişkenin aynı veya farklı olduğunu söylemesi ama benzerlikler ile ilgili yeterince bilgi vermemesidir (Hancock ve Khoshgoftaar, 2020).

Target Encoding (Hedef Kodlama): Bu teknikte veri setindeki örnekler, kategorik verinin değişkenlerine göre gruplandırılır, her bir grupta sınıf değişkenindeki durumun gerçekleşme sayısı bulunur. Ardından o kategorik özniteliğin aynı grubunda yer alan kayıtları için sınıf değişkenlerinin ortalaması hesaplanır. Son olarak kategorik verilerin olduğu sütun, hesaplama sonucu elde edilen sayısal değerlerden oluşan sütun ile değiştirilir. Örneğin bu çalışmada, veri setinde protocol_type isimli öznitelik udp, tcp, icmp olmak üzere üç farklı değişkene sahip kategorik bir veridir. Bu kategorik değere sahip özniteliği target encoding tekniği ile kodlarsak, tcp kategorik değeri için 0,521964, udp kategorik değeri için, 0,82932, icmp kategorik değeri için 0,157882 sayısal değerleri hesaplanır ve protocol_type isimli kategorik özniteliğin üç değişkeni, bu üç sayısal değer ile temsil edilir. Kodlama sonucu öznitelik sayısının değişmemesi tekniğin önemli bir avantajıdır; fakat baz alınan sınıf değişkeni olduğu için, makine öğrenimi algoritmasının target encoding uygulanmış öznitelik ile diğer öznitelikler arasındaki ilişkiyi öğrenmesi zorlaşır, bu da modelin kodlanmış bilgiyi ayıklama performansını olumsuz etkileyebilir. Ayrıca teknik uygulanırken, kategorik özniteliğin değişkenleri gruplandırıldığında aynı grupta yer alan değişkenlerin her biri aynı sayısal değerle değiştirildiği için, model gördüğü kodlanmış değerlere aşırı uyma eğiliminde olabilir. Bu durum aşırı öğrenmeye (overfitting) yol açar.

Leave One Out Encoding (Birini Dışarıda Bırakarak Kodlama): Zhang tarafından keşfedilip kullanılmış yeni bir tekniktir (Zhang, 2015). Leave one out encoding, target encoding'de olduğu gibi söz konusu kategorik özniteliğin aynı değişkenini içeren tüm kayıtları için sınıf değişkenlerinin ortalamasını hesaplar. Target encoding'den tek farkı, hesaplamaya değerlendirilen kaydı dâhil etmemesidir. Bu da aşırı öğrenmeyi azaltır. Aşırı öğrenmeyi önlemenin başka bir yolu da, hesaplanan ortalamaya küçük bir rastgele gürültü değeri eklemektir (Hancock ve Khoshgoftaar, 2020). Ayrıca model, aynı grupta yer alan kategorik değişkenlerin her biri için aynı sayısal değere değil, bir aralığa da maruz kaldığından, daha iyi genellemeyi öğrenir. Bu teknik öznitelik sayısını değiştirmediği ve aşırı öğrenmeyi target encoding tekniğine kıyasla azalttığı için özellikle yüksek değişken sayısına sahip kategorik değerli öznitelikler için kullanışlıdır.

Weights Of Evidence Encoding: Bu teknik bir kategorik özniteliğin tahmin gücünü, sınıf değişkeni ile ilişkilendirerek hesaplar. Kredi puanlama dünyasında geliştiği için, genellikle iyi ve kötü müşterilerin ayrımının bir ölçüsü olarak tanımlanır. Kötü müşteriler bir krediyi ödeyemeyen müşterileri, iyi müşteriler krediyi geri ödeyebilen müşterileri ifade eder. Bu teknikte her kategorik değişken, ln[p(1) /p(0)] ile değiştirilir. Burada p(1), iyi sınıf değişkeni olma olasılığı, p(0) ise kötü sınıf değişkeni olma olasılığıdır. Örnek verilecek olunursa, bu çalışmada kullanılacak veri setinde protocol_type kategorik özniteliğinin her değişkeni için hedef değişkenin normal olma durumu p(1), saldırı olma durumu ise p(0)'dır.

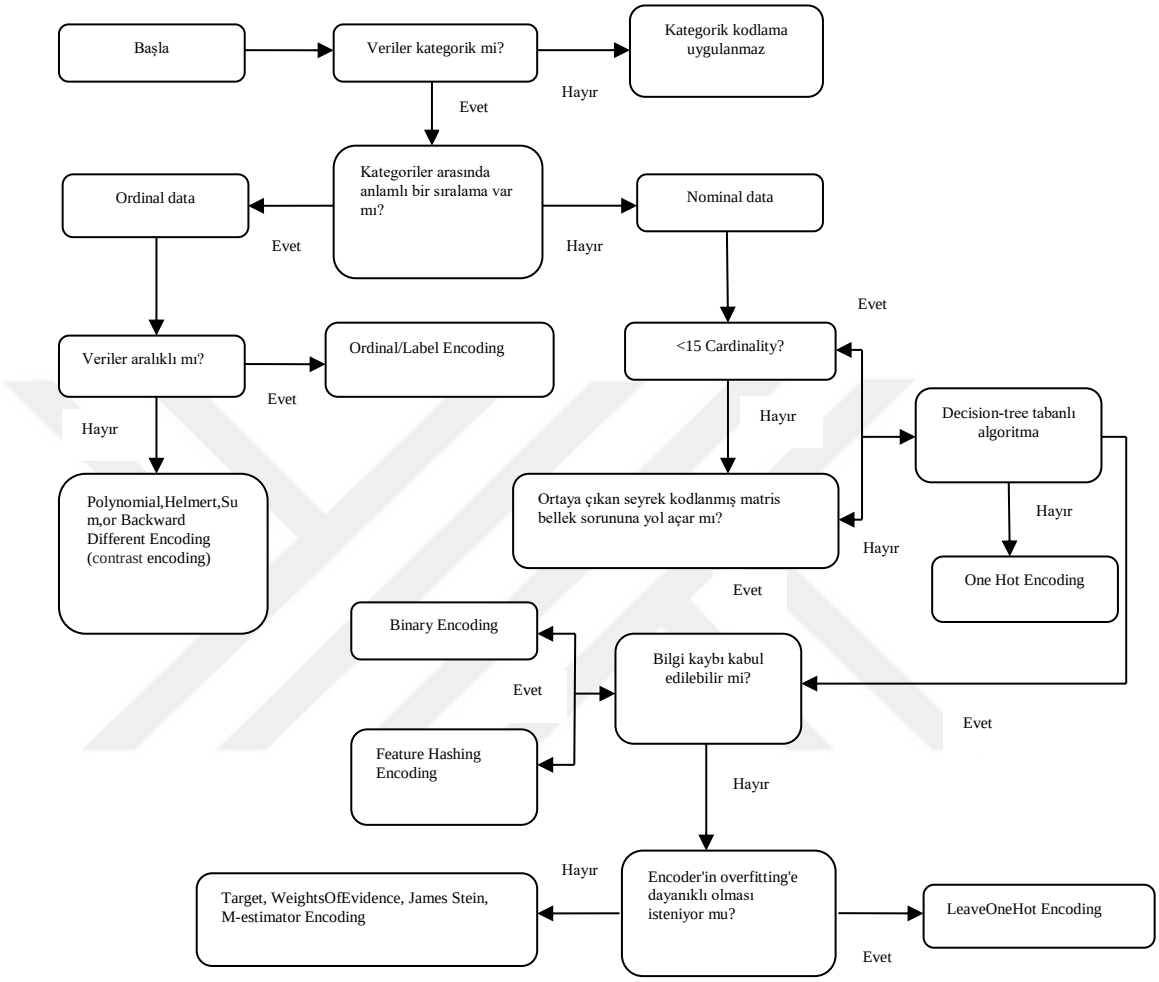
James Stein Encoding: James Stein Encoding tekniğinin Target encoding tekniğinden tek farkı, kategorik öznitelik için aynı değeri içeren tüm kayıtlarda sınıf değişkenlerinin ortalamasını hesaplarken sonucu, tahmini iyileştirmek için ortalama değere doğru yaklaştırmasıdır.

CatBoost Encoding: James Stein Encoding'den, tek farkı kategorik değere sahip özniteliklerde, her bağlantı için sınıf değişkeninin gerçekleşme olasılığını yalnızca kendisinden önceki satırlardaki örnekleri dikkate alarak hesaplamasıdır. Bu işlem aşırı öğrenmeyi arttırabilir.

M-Estimate Encoding: Target Encoding'in basitleştirilmiş bir türüdür. Tek parametresi olam M parametresi ne kadar yüksek girilirse kategorik özniteliğin değişkeni için hesaplanan değer, ortalamaya o kadar yaklaşır. M için önerilen değerler 1 ile 100 arasındadır.

Target, LeaveOneOut, WeightOfEvidence, James Stein ve M-estimator Encoding teknikleri, kategorik verileri sayısal verilere dönüştürürken sınıf değişkenindeki bilgileri kullanır ve kodlama sonucunda öznitelik sayısı aynı kalır. Bilgi kaybı diğer yöntemlere göre daha azdır. Aralarında herhangi bir ilişki olmayan değişkenlere ve yüksek kardinaliteye sahip kategorik değerli verileri bulunduran veri setleri ile iyi sonuçlar verebilirler. Bu çalışmada NSL-KDD veri setindeki kategorik

verileri sayısal verilere dönüştürmek için anlatılan tüm teknikler kullanılacak ve başarı durumları kıyaslanacaktır. Kullanılacak veri seti ve çözülmek istenen problem için uygun kodlama tekniği belirlenirken Şekil 2.3'teki akış şeması fikir verebilir.



Şekil 2.3. Kategorik veri kodlama tekniklerini seçim kriterleri (Alteryx, 2019).

2.1.2. Veri Ölçeklendirme

Veri setlerinde bazı verilerin değerleri arasında çok büyük farklılıklar olabilir ve bu durum verileri karşılaştırırken, değeri büyük olan verinin gerçekçi olamayacak kadar baskın çıkmasına neden olur, dolayısıyla doğru bir karşılaştırma yapılmasını engeller. Veri ölçeklendirme, farklı aralıklardaki veri değerlerinin ortak bir ölçekte eşleştirilmesidir. Bu sayede söz konusu veriler daha objektif karşılaştırılabilir.

Ölçeklendirme işlemi algoritmaların başarı oranlarını arttırmakla beraber eğitim sürelerini de kısaltabilir. Ölçeklendirme sürecinde sağlamlık ve verimlilik özellikleri korunmalıdır. Sağlamlık, aykırı ya da uç değerlerin veri örneklerinin ölçeklendirme sürecini etkilememesidir. Verimlilik ise, ölçeklendirme işlemi uygulanmış veride bilgi kaybının olmaması, karakteristikliğin korunmasıdır (Song ve ark., 2017). Sınıflandırma problemine, veri setine, makine öğrenimi algoritmasına bağlı olarak, kullanılacak ölçeklendirme yöntemi değişiklik gösterebilir. Uygun yöntemi seçmek modelin sınıflandırma başarısı üzerinde etkilidir. Bu çalışmada NSL-KDD veri setine uygulanacak ön işlemlerin veri ölçeklendirme basamağında, literatürdeki STS çalışmalarında da yaygın kullanılan min-max ölçeklendirme yöntemi kullanılacaktır.

Min-Max: Bu yöntemde en küçük değere sahip veri 0, en büyük değere sahip veri 1 olacak şekilde, diğer bütün veri değerleri bu 0-1 aralığına yayılarak ölçeklendirme yapılır. Min-Max yöntemi için denklem 1 kullanılır.

$$x' = x_i - x_{min} / x_{max} - x_{min} \quad \textbf{(1)}$$

x'= Ölçeklendirilmiş veri

x_i=Girdi değeri

x_{min}=Girdi seti içinde yer alan en küçük değer

x_{max}=Girdi seti içinde yer alan en büyük değerdir.

2.1.3. Öznitelik Seçimi

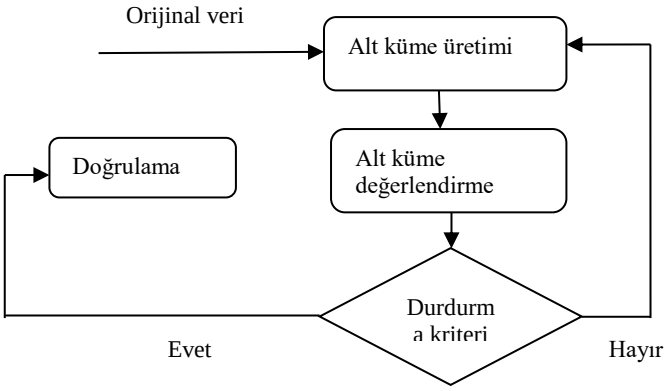
Bir veri setinin yer aldığı tabloda her bir satırdaki bilgiye kayıt denir. Her kayıt için tanımlanmış sütunda yer alan her bir veriye ise öznitelik, kaydın normal mi saldırı mı olduğunu gösteren veriye ise sınıf değişkeni denir. STS çalışmalarında oluşturulan saldırı tespit modelleri, öznitelik verilerini kullanarak sınıf değişkenini tahmin etmeyi öğrenirler.

N adet özniteliği bulunan bir veri seti için hesaplanacak öznitelik alt kümesi sayısı 2^N'dir. Örneğin, bu çalışmada kullanaılacak veri setinde 41 öznitelik, bir tane de sınıf değişkeni vardır. 41 öznitelik için oluşturulacak öznitelik alt küme sayısı

2^{41}=2.199.023.255.552'dir. Bu durumda en uygun öznitelik alt kümesini seçmek için tüm öznitelik alt kümelerini denemek, zaman, hesaplama maliyeti ve kullanılan algoritmanın performansı açısından ciddi bir problemdir. Bu yüzden veri setini temsil etmede yetersiz, algoritmanın sınıflandırma performansını düşürecek öznitelikler veri setinden çıkarılmalıdır. Öznitelik seçimi, veri setindeki özniteliklerden veri setini en iyi temsil edecek özniteliklerin seçilmesidir. STS'lerde olduğu gibi sınıflandırma problemlerinde öznitelik seçimi:

- Makine öğrenimi algoritmasının eğitim ve test süresini kısaltır.
- Modelin karmaşıklığını azaltır ve yorumlamayı kolaylaştırır.
- En doğru öznitelik alt kümesi seçimi ile modelin sınıflandırma başarısını arttırır.
- Aşırı öğrenmeyi azaltır.

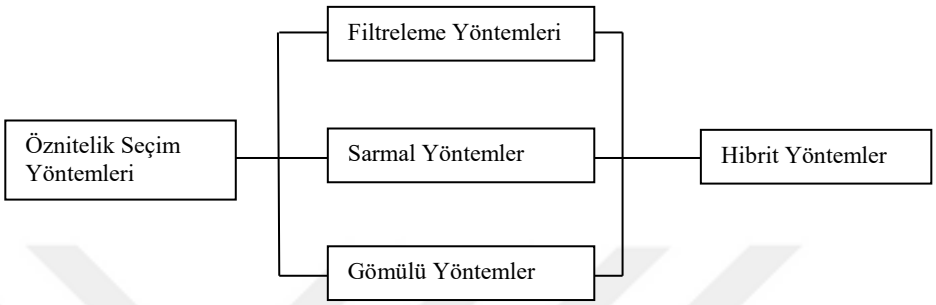
Öznitelik seçim süreci arama yönü, arama stratejisi, değerlendirme işlevi ve durma noktası olmak üzere dört bileşenden oluşur. Önce orijinal veri setinden öznitelik alt kümeleri üretilir, sonra bu alt kümelerdeki öznitelikler kullanılan öznitelik seçim yöntemine göre değerlendirilir, ilgili özelliğin seçilip seçilmeyeceğine karar verilir. Seçilmesine karar verilen öznitelik ilgili alt kümeye dâhil edilir ve durdurma kriteri sağlanıncaya kadar süreç tekrarlanır (Şekil 2.4).



Şekil 2.4. Öznitelik seçim sürecinin temel adımları (Bouaguel, 2015).

2.1.3.1 Öznitelik Seçim Yöntemleri

Bir veri setinde en büyük sınıflandırma yapma gücüne sahip öznitelikleri tanımlarken dört ana model kullanılır: filtreleme yöntemleri, sarmal yöntemler, gömülü ve hibrit yöntemler (Şekil 2.5).



Şekil 2.5. Öznitelik seçim yöntemleri.

2.1.3.1.1. Filtreleme Yöntemleri

Filtreleme yöntemleri makine öğrenimi algoritmalarından bağımsızdır. Özniteliklerin, sınıf değişkeni ile korelasyonları çeşitli istatistiksel ölçütlere göre hesaplanır ve her özelliğe bir puan atanarak bir sıralama yapılır. Yüksek puanlı öznitelikler seçilir, düşük puanlı öznitelikler elenir (Li ve ark., 2021).

Filtreleme Yöntemlerinin Avantajları: Öznitelik seçimi hızlı, kaynak tüketimi ve hesaplama maliyeti azdır, kurulacak modelden bağımsız olduğu için istenilen herhangi bir makine öğrenimi algoritmasında kullanılabilir.

Filtreleme Yöntemlerinin Dezavantajları: Öznitelikler seçilirken sadece sınıf değişkeni ile olan ilişkinin dikkate alınması, özniteliklerin birbirleriyle olan ilişkilerinin sınıflandırma üzerindeki etkisinin yok sayılmasına neden olur, bu da aşırı öğrenme riskini arttırır. Filtreleme yöntemlerinin sınıflandırma algoritmasının

değerlendirmesinden yoksun olması düşük sınıflandırma doğruluğu sonucunu doğurabilir.

Fisher skor, t-Skor, korelasyon tabanlı öznitelik seçimi (Correlation based Feature Selection-CFS) , ki-kare testi (Chi-square), bilgi kazancı (information gain-IG), karşılıklı bilgi (mutual information-MI), kazanç oranı (gain ratio-GR), relief, one-R yaygın kullanılan filtreleme yöntemleridir.

Bilgi Kazancı (Information Gain): Entropi tabanlı bir öznitelik seçim yöntemidir. Entropi ise bir sistemdeki belirsizliğin ölçüsüdür ve 0 ile 1 arasında bir değer alır. Yüksek entropi değeri, söz konusu sistemin daha fazla bilgi içerdiği anlamına gelir. Bilgi kazancı (IG) yönteminde amaç, X özelliğini gözlemleyerek Y özelliği hakkında bilgi kazanmak ve Y özelliğinin entropi değerindeki azalmayı ölçmektir. IG bir rastgele değişkenin başka bir rastgele değişken hakkında içerdiği bilgi miktarının bir ölçüsüdür. Bu yöntemde, sınıf değişkenlerinin entropi değerleri hesaplanır (Denklem 2). Daha sonra veri setindeki her bir öznitelik için entropi değerleri hesaplanır (Denklem 3). Son olarak bulunan entropi değerlerinin farkı alınarak bilgi kazancı hesaplanır (Denklem 4). Çıkan sonuç ne kadar yüksek ise ilgili öznitelik veri setini temsil etmede o kadar başarılıdır. IG değeri düşük, veri setini temsil etmede yetersiz öznitelikler elenir (Kaynar ve ark., 2018). IG, X gözlemlenerek Y hakkında kazanılmış bilgi ile Y gözlemlenerek X hakkında kazanılmış bilgi birbirine eşit olduğu için simetrik bir değerlendirme ölçütüdür. Yöntemin dezavantajı ise, daha fazla bilgi içermediğinde bile yüksek kardinaliteye sahip öznitelikler lehine önyargılı sonuç vermesidir (Budak, 2018). Bu da aşırı öğrenmeye neden olmaktadır.

$$H(Y)= -\sum_{y \in Y} p(y) \log_2 (p(y)) \qquad (2)$$

$$H(Y/X)= -\sum_{x \in X} p(x) \sum_{y \in Y} p(y\backslash x) \log_2 (p(y\backslash x)) \qquad (3)$$

$$\text{Bilgi Kazancı} = H(Y)-H(Y\backslash X) \qquad (4)$$

Kazanım Oranı (Gain Raito): IG yönteminin, yüksek kardinaliteye sahip öznitelikleri seçme eğiliminin önüne geçmek için geliştirilmiş bir yöntemdir. Yöntemde önce her bir öznitelik için bölünme bilgisi hesaplanır, daha sonra bilgi kazancı değeri, bölünme bilgisi değerine bölünerek kazanım oranı (GR) bulunur (Denklem 5).

Kazanım Oranı= Bilgi Kazancı \ H(X) (5)

Karşılıklı Bilgi (Mutual İnformation): Karşılıklı bilgi (MI), bir öznitelik ile sınıf değişkeni arasında ölçülebilir bir bağlantı kurmanın yoludur. "*y'nin değerinin öğrenilmesinden kaynaklanan, x hakkındaki belirsizlikteki ortalama azalmayı ölçer; veya tam tersi, x'in y hakkında ilettiği ortalama bilgi miktarını ölçer"* (MacKay, 2003). MI denklem 6 ile hesaplanır.

$$I(X ; Y) = H(X) - H(X \mid Y) \quad (6)$$

I=MI (Karşılıklı Bilgi), X=Öznitelik, Y=Sınıf Değişkeni, H=Entropi

MI puanı 0 ile 1 arasındadır. Değer ne kadar yüksek olursa, bu öznitelik ile sınıf değişkeni arasındaki bağlantı o kadar güçlü olur ve bu özniteliğin seçilmesi gerektiğini düşündürür. Hesaplanan sonuç sıfır ise, öznitelik ile sınıf değişkeni birbirinden bağımsızdır. IG ve MI'nın hesaplama yöntemleri benzer olup çoğu zaman birbirlerinin yerine kullanılsa da, öznitelik seçimi bağlamında her ikisi de, bir özniteliğin belirli bir sınıf değişkeni ile ne kadar alakalı olduğunu ölçer, ancak MI yalnızca olumlu öznitelikleri ölçerken, IG hem olumsuz hem de olumlu öznitelikleri ölçer. Bu nedenle, MI tek taraflı, IG iki taraflı bir metriktir.

Doğrusal bağımlılığa dayalı yöntemler (korelasyon vb.), model koordinatları ve farklı sınıf değişkenleri arasındaki keyfi ilişkilerle ilgilenemez; fakat MI değişkenler arasındaki keyfi ilişkileri ölçebilir ve farklı değişkenler üzerinde etkili olan dönüşümlere bağlı değildir (Battiti, 1994).

2.1.3.1.2. Sarmal Yöntemler

Sarmal yöntemler, öznitelikleri seçerken sınıflandırma algoritmalarını kullandıkları için yüksek bir sınıflandırma doğruluğu elde eder. Ancak hesaplama süresi ve kaynak tüketimi filtreleme yöntemlerine kıyasla oldukça fazladır. Sarmal yöntemler, öznitelikler alanında bir arama prosedürü kullanır ve ardından veri setini en iyi temsil eden öznitelikleri bulmak için çeşitli alt kümeler oluşturur, değerlendirir. Belirli bir öznitelik alt kümesinin değerlendirilmesi, belirli bir sınıflandırma modelinin tekrar tekrar eğitilmesi ve test edilmesiyle yapılır (Bouaguel, 2015). Ardışık ileri yönde seçim (SFS), ardışık geri yönde seçim (SBS), L ekle -R çıkar, ardışık ileri yönde kayan seçim (SFFS), ardışık geri yönde kayan seçim (SBFS), genetik algoritma (GA) yaygın kullanılan sarmal tekniklerdendir.

Ardışık İleri Yönde Seçim (Sequential Forward Selection-SFS): SFS, işleme boş öznitelik kümesi ile başlar. Her döngüde ilgili öznitelik, sınıflandırma başarısına olan etkisine göre öznitelik alt kümesine eklenir ya da elenir. Böylece veri setini en iyi temsil eden öznitelik alt kümesine ulaşılır.

Ardışık Geri Yönde Seçim (Sequential Backward Selection-SBFS): SFS algoritmasının tersine SBS, öznitelik seçim işlemine tüm öznitelikler kümesi ile başlar. Her döngüde kümeden bir öznitelik çıkarılır ve bu işlem sınıflandırma başarısında bir artış söz konusu olmayana dek sürer.

Genetik Algoritma (Genetic Algorithm-GA): GA'lar doğal evrimden esinlenen sezgisel bir arama ve optimizasyon tekniğidir. GA'da her bir kromozom bir problemin çözümünü temsil eder, en iyi çözümü arama işlemine, rastgele üretilmiş bir kromozom popülasyonu ile başlar. Bu popülasyon bir durdurma kriterine ulaşıncaya kadar çaprazlama, mutasyon işlemleri ile güncellenir. Bu şekilde bir GA, belirli bir problemin en iyi çözümünü üretir (McCall, 2005). GA, birçok öznitelik seçimi algoritmasına göre yavaş olsa da başarı oranı genel olarak oldukça yüksektir.

2.1.3.1.3. Gömülü Yöntemler

Gömülü yöntemler, filtreleme ve sarmal yöntemleri aynı anda kullanır. Öznitelik seçme sürecini ve sınıflandırma algoritmasının eğitimini bütünleştirerek sınıflandırma eğitimi ile eş zamanlı olarak öznitelik seçimini gerçekleştirir. Böylece sarmal yöntemlerde yapılan farklı alt kümeleri yeniden sınıflandırmak için harcanan hesaplama maliyeti ve süresi azaltılır. Gömülü yöntemlerde öznitelik seçiminin sınıflandırma algoritmalarıyla bağlantısı, sarmal yöntemlerde olduğundan daha güçlüdür (Liu ve ark., 2019).

Gömülü Yöntemlerin Avantajları: Sarmal yöntemlerde olduğu gibi özniteliklerin birbirleriyle olan etkileşimini dikkate alırlar. Filtreleme yöntemleri gibi hızlıdırlar ve filtreleme yöntemlerine kıyasla daha başarılıdırlar. Aşırı öğrenmeye, sarmal ve filtreleme yöntemlerine kıyasla daha az eğilimlidirler.

Gömülü Yöntemlerin Dezavantajları: Sınıflandırma algoritmalarına bağımlıdırlar ve sınıflandırma algoritmasındaki en ufak değişiklik kötü performansa yol açabilir.
Bu çalışmada, gömülü yöntemlerde kullanılan düzenlilik yaklaşımı ve algoritma temelli yaklaşımdan ikincisi kullanılacaktır. Algoritma temelli yaklaşımda öznitelik seçimi işlemi çoğunlukla karar ağacı, randomforest, ExtraTree, XGBoost gibi ağaç tabanlı algoritmalar kullanılarak yapılır. Çizelge 2.2'de öznitelik seçimi yöntemleri karşılaştırılmıştır.

Çizelge 2.2. Öznitelik seçim yöntemlerinin karşılaştırılması.

Öznitelik seçim yöntemi	**Çalışma prensibi**	**Avantajları**	**Dezavantajları**
Filtreleme	Öznitelik seçimi makine öğrenimi algoritmasından bağımsız olarak, ilgili özniteliğin sınıf değişkeni ile ilişkisine bakılarak yapılır.	Hızlı, kaynak tüketimi az, makine öğrenimi algoritmalarından bağımsız.	Düşük başarı oranı, aşırı öğrenmeye yatkınlık.
Sarmal	Öznitelikler seçilirken sınıflandırma algoritmasının değerlendirmesinden geçer. Sınıflandırmaya katkısı olan öznitelikler seçilir, diğerleri elenir.	Yüksek başarı oranı.	Yavaş, kaynak tüketimi fazla, aşırı öğrenmeye yatkınlık
Gömülü	Öznitelik seçim sürecini ve sınıflandırma algoritmasının eğitimini bütünleştirirerek sınıflandırma eğitimi ile eş zamanlı olarak öznitelik seçimini gerçekleştirir.	Hızlı, yüksek başarı oranı, aşırı öğrenmeye dirençli.	Makine öğrenimi algoritmalarına bağımlılık.

2.1.3.1.4. Hibrit Yöntemler

Hibrit yöntemler, filtreleme, sarmal ve gömülü yöntemleri bir arada kullanır. Amaç her bir yöntemin avantajlarından faydalanırken, eksikliklerini bir diğer yöntem ile tamamlamaktır. Örneğin, ilk olarak aday öznitelikler, hesaplama açısından avantajlı olduğu için filtreleme yöntemleri kullanılarak orijinal öznitelik kümesinden seçilir. Sonra seçilen aday öznitelikler, sınıflandırma doğruluğu açısından avantajlı sarmal yöntemler kullanılarak değerlendirilir ve veri setini en iyi temsil eden öznitelik alt kümesi bulunur (Hsu ve ark., 2011).

Bu çalışmada, öznitelik seçimi ön işlemi yapılırken hibrit öznitelik seçim yöntemi kullanılmıştır. Hibrit öznitelik seçim yöntemi ile filtreleme, sarmal ve gömülü öznitelik seçim yöntemlerinin birlikte kullanımlarının birbirlerinin dezavantajlarının yok edip, eksikliklerini tamamlayarak, bu yöntemlerin hız, sınıflandırma başarısı gibi avantajlarından maksimum düzeyde yararlanmak hedeflenmiştir. Hibrit öznitelik seçim yöntemi için filtreleme, sarmal, gömülü öznitelik seçimi yöntemlerini temsilen MI, GA ve XGBoost yöntemleri kullanılmıştır. Çalışmada bu yöntemlerin seçilmesinin nedenleri; MI'ın değişkenler arasındaki keyfi ilişkileri ölçebilmesi ve farklı değişkenler üzerinde etkili olan dönüşümlere bağlı olmaması, GA'nın yavaş olmasına ve kaynak tüketiminin fazla olmasına rağmen başarı oranının oldukça tatmin edici olması, gömülü yöntemleri temsilen seçilen XGBoost'un ise hızlı olması, topluluk öğrenme algoritmaları içerisinde ve çevrim içi veri bilimcileri ile makine öğrenimi uygulayıcıları topluluğu olan kaggle'ın düzenlediği yarışmalarda açık ara başarılı olmasıdır.

Çalışmada kullanılan öznitelik seçim yöntemlerini iki gruba ayırabiliriz. MI ve XGBoost yöntemleri öznitelikleri önemlerine göre sıralarken, GA öznitelik seçimi yapar. MI, her bir öznitelik için bir önem değeri hesaplar, daha sonra öznitelikleri bu önem değerine göre yeniden sıralar. XGBoost gömülü öznitelik seçim yöntemi ile de en uygun öznitelikler sınıflandırma performansına göre yeniden sıralanır ama sarmal yöntemlerden farklı olarak öznitelik seçimi, sınıflandırma eğitimi ile eş zamanlı olarak

gerçekleştirilir. GA yönteminde ise öznitelikler, kullanılan sınıflandırma algoritmasının performansına bağlı olarak seçilir ya da elenir. Bu çalışmada GA ile öznitelik seçimi yapılırken sınıflandırıcı olarak hızlı eğitilebilmesinden dolayı lojistik regresyon (LR), özniteliklerin birbirleriyle olan etkileşimlerini ve öznitelik seçimini etkileyebilecek diğer değişkenleri hesaplayabilme yeteneğinden dolayı ağaç tabanlı bir algoritma olan RF kullanılmıştır.

MI ve XGBoost öznitelik seçim yöntemleri ile elde edilen önem değerleri için bir eşik değeri belirlenerek bu eşik değerinin üzerinde önem derecesine sahip öz nitelikler seçilebilmektedir; fakat eşik değerinin ne olacağı bilgisi açık olmadığı için çalışmada, en iyi başarı oranlarını veren eşik değerleri deneme yanılma yoluyla belirlenmiştir.

2.2. Saldırı Tespit Modelleri Oluşturulması

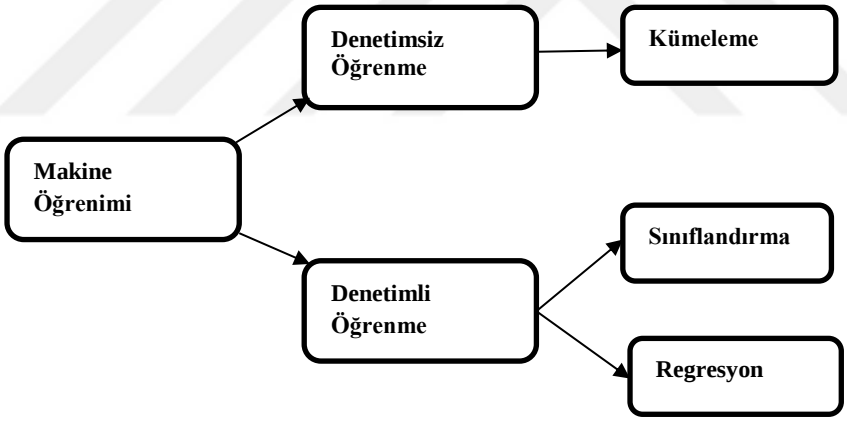
Önerilen metodolojinin ikinci aşamasında, birinci aşamasında ön işlemler uygulanarak elde edilen veri setleri ve ön işlem uygulanmamış veri setleri ile KNN, MLP, RF, XGBoost, LightGBM makine öğrenimi algoritmaları kullanılarak saldırı tespit modelleri oluşturulmuştur. Bu bölümde makine öğrenimi sınıflandırma yöntemleri anlatılacaktır.

2.2.1. Makine Öğrenimi Sınıflandırma Yöntemleri

Bilgi ve iletişim teknolojilerinin ulaşılabilirliğinin kolaylaşması, hayatın her alanında kullanımının artması ile bu teknolojilerin ürettiği verilerin çeşitliliği ve boyutu da çok hızlı artmaktadır. Bu veri yığınları içerisinden gelecek ile ilgili tahminlerde bulunabilmek için ihtiyaç duyulan anlamlı bilgiye ulaşma sürecine veri madenciliği denir.

Makine öğrenimi ise bir veri madenciliği yöntemidir. İstatistik, yapay zekâ ve bilgisayar biliminin kesişme noktasındaki bir araştırma alanıdır, aynı zamanda tahmine dayalı analitik veya istatistiksel öğrenme olarak da bilinir. Hangi filmlerin izleneceğine, hangi ürünlerin satın alınacağına ilişkin kullanıcılara isabetli önerilerde bulunmak için, Facebook, Amazon, Netflix gibi web sitelerinde, çok sayıda makine öğrenimi modelleri kullanılır (Müller ve Guido, 2017).

Makine öğrenimi denetimli (supervised) öğrenme ve denetimsiz (unsupervised) öğrenme olarak ikiye ayrılır. Denetimli öğrenmede verilerin sınıfı bellidir ve yeni gelecek veri bu sınıflardan birine dâhil edilir. Sınıflandırma ve regresyon birer denetimli öğrenme yöntemidir. Veri sınıfı sayısal değilse sınıflandırma, sayısal ise regresyon kullanılır. Denetimsiz öğrenmede ise verilerin sınıfı belli değildir, veriler arasında örüntüler oluşturulur. Kümeleme denetimsiz öğrenme yöntemidir. Şekil 2.6'da makine öğrenimi yöntemleri gösterilmektedir.



Şekil 2.6. Makine öğrenimi yöntemleri.

Bu çalışmada kullanacağımız NSL-KDD veri setinin sınıf değişkeni kategorik (normal, anormal) olduğu ve amacımız yeni gelen verileri bu sınıflardan birine dâhil etmek olduğu için denetimli öğrenme yöntemi olan sınıflandırma yöntemini kullanacağız. Çalışmada kullanılacak sınıflandırma algoritmaları:

KNN (K-Nearest Neighbors) Algoritması: 1951'de Evelyn Fix ve Joseph Hodges tarafından geliştirilmiştir. En basit sınıflandırma algoritmalarından biridir ve lazy

(tembel) bir öğrenme algoritmasıdır, eğitim verilerini öğrenmez sadece ezberler. KNN, yeni bir veriyi daha önce ezberlediği eğitim verileriyle olan yakınlık ilişkisine bakarak sınıflandırır. KNN algoritması kullanılırken, kontrol edilecek veri sayısını ifade eden bir k değeri belirlenir. Örneğin k=5 olarak belirlenirse, yeni gelen verinin eski verilere uzaklıkları ölçülür ve en yakın 5 tanesi belirlenir. Yeni bir verinin sınıfı belirlenirken, en yakın k kadar eleman ile yeni veri arasındaki uzaklıklar hesaplanır ve yeni veri en fazla örnek bulunduran sınıfa dâhil edilir. K değerinin tek tam sayı seçilmesinin nedeni, k değerinin çift tamsayı seçilmesi durumunda sınıflar arası eşitlik durumunun oluşma olasılığını ortadan kaldırmaktır. Uzaklık hesaplanırken genelde Öklid fonksiyonu, altenatif olarak da Manhattan, Minkowski ve Hamming fonksiyonları kullanılır.

Multi Layer Perceptron (MLP): MLP, derin yapay sinir ağıdır. Sinyali almak için bir girdi katmanından, tahminde bulunmak için hesaplama işlemlerini gerçekleştirdiği gizli katmandan, girdi hakkında tahminde bulunan bir çıktı katmanından oluşur. Gizli katman sayısı, isteğe veya probleme bağlı olarak değişebilir. Genel olarak denetimli öğrenme problemlerine uygulanırlar. Bir dizi girdi-çıktı çifti üzerinde eğitim alırlar ve bu girdiler-çıktılar arasındaki doğrusal olmayan ilişkileri modellemeyi öğrenirler. Girdiler, başlangıç ağırlıkları ile ağırlıklı bir toplamda birleştirilir ve aktivasyon fonksiyonuna tabi tutulur. İşlem sonrası elde edilen kombinasyonlar bir sonraki katmana yayılır. Bu yüzden MLP ileri beslemeli bir algoritmadır; fakat maliyet fonksiyonunu en aza indirmek amacıyla ağdaki ağırlıkların yinelemeli olarak ayarlamasına izin veren, geri yayılım adı verilen öğrenme mekanizması kullanır. Denetimli öğrenme çalışmalarında tahmin doğruluğu başarısından dolayı sıklıkla kullanılır. Gizli katman sayısı, aktivasyon fonksiyonu, ağırlık öğrenme oranı gibi parametreler ayarlanarak MLP ile kurulan modelin başarısı arttırılabilir.

2.2.1.1. Topluluk Öğrenme (Ensemble Learning) Yöntemleri

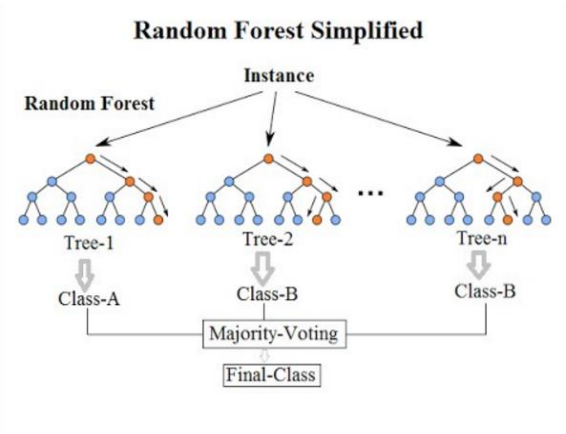
Topluluk öğrenme, daha başarılı bir tahmin sonucuna ulaşabilmek için birden fazla algoritmayı aynı anda kullanan makine öğrenimi yöntemidir. Bir topluluk, temel öğreniciler (base learner) ya da zayıf öğreniciler (weak learner) olarak adlandırılan bir dizi öğreniciyi (karar ağacı, sinir ağı vb.) içerir ve iki adımda oluşturulur: Temel öğrenicileri seçmek ve ardından bu öğrenicilerin tahmin sonuçlarını birleştirmek (Zhou, 2012). En iyi tahmin sonuçlarını elde etmek için, topluluktaki temel öğreniciler probleme uygun seçilmeli ve çeşitli olmalıdır. Farklı veri noktalarında farklı hatalar yapan temel öğreniciler birbirinden farklı yani çeşitlidir (Fawagreh ve ark., 2014). Hangi temel öğrenicilerin birlikte kullanılarak, hangi derecede tahmin gücü yüksek, güçlü öğrenicilere dönüştürülebileceği makine öğrenimi alanında çalışılan önemli bir konudur ve AdaBoost, Bagging gibi ünlü topluluk öğrenme yöntemlerinin doğmasını sağlamıştır (Zhou, 2012). Bagging, Boosting, Random Forest, Gradient Boosting, XGBoost ve LightGBM bu bölümde anlatılacak topluluk öğrenme yöntemleridir. Çalışmamızda ise Random Forest, XGBoost ve LightGBM kullanılacaktır.

2.2.1.1.1. Bagging (Bootstrap Aggregating) Algoritması

Bagging (torbalama, önyükleme toplaması) 1996 yılında Breiman tarafından geliştirilmiştir (Breiman, 1996). Yöntemde temel öğreniciler, eğitim setinden rastgele seçilen farklı alt kümeleriyle eğitilir. Seçilen her alt küme yerine tekrar konulduğu için bazı örnekler eğitim setinde birden fazla kez yer alabilir. Eğitim setlerinin çeşitlendirilmesindeki amaç, toplam tahmin başarısını arttırmaktır. Son olarak, öğrenicilerin tüm tahminleri, regresyon problemleri için ortalama alınarak, sınıflandırma problemleri için ağırlıklı oylama yöntemi ile birleştirilir. Random Forest bir bagging algoritmasıdır.

Random Forest (Rastgele Orman) Algoritması: Rastgele orman (RF), her bir ağacın ormandaki tüm ağaçlar için aynı dağılıma sahip bağımsız olarak örneklenen rastgele

bir vektörün değerlerine dayandığı bir ağaç tahmin edicilerin birleşimidir (Breiman, 2001).



Şekil 2.7. Rastgele orman algoritması akış şeması (Jagannath, 2020)

Şekil 2.7'de görüldüğü gibi, RF algoritması veriyi farklı parçalara böler, bu parçalar üzerinden farklı karar ağaçları oluşturur ve bu karar ağaçlarının tahmin sonuçlarını çoğunluğun oyu yöntemi ile birleştirerek bir tahmin sonucuna ulaşır. RF algoritmasının, birçok problemde yüksek doğruluk sağlaması, hızlı olması, hata, güç, korelasyon ve değişken önemi hakkında yararlı tahminlerde bulunabilmesi, basit bir yapısının olması, kolayca paralelleştirilebilmesi gibi birçok avantajı vardır (Breiman, 2001).

2.2.1.1.2. Boosting (Güçlendirme) Algoritması

Bu yöntemde, eğitim için ayrılan veri setinden rastgele seçilen verilerle temel öğrenicinin eğitimi gerçekleştirilir. Ardından oluşturulan model test edilir. Test sonucu yanlış sınıflandırılan örnekler belirlenir ve bunlar bir sonraki öğrenici için eğitim verisi seçiminde önceliklendirilir. Her eğitimde bu seçim güncellenir. Bu şekilde yanlış verilen kararlar üzerine odaklanılarak doğru tahmin oranı arttırılır. Boosting ile amaç, temel öğrenicilerin birleştirilerek güçlü bir öğrenici oluşturulmasıdır (Kearns, 1988). Bagging yönteminde her bir iterasyonda tüm örneklerin eğitim veri setine seçilme olasılıkları aynı iken boosting'de her iterasyonda veri örneklerinin seçilme olasılıkları

güncellenmektedir. Boosting modelleri: AdaBoost, Gradient Boosting, XGBoost, LightGBM ve CatBoost algoritmalarıdır.

AdaBoost (Adaptive Boosting) Algoritması: Temel öğrenicilerin birleştirilerek güçlü bir öğrenici oluşturulması fikri, 1995 yılında Freund ve Schapire tarafından AdaBoost algoritması geliştirilerek hayata geçirilmiştir (Freund ve Schapire, 1997). Algoritmanın ana fikri, eğitim seti üzerinde bir ağırlık kümesinin sürdürülmesidir. Başlangıçta tüm ağırlıklar eşit olarak ayarlanır, ancak her iterasyonda yanlış sınıflandırılmış örneklerin ağırlıkları arttırılır. Böylece temel öğrenici eğitim setindeki sınıflandırılması zor örneklere odaklanmaya zorlanır, adapte edilir (Freund ve Schapire, 1999). Bu şekilde zayıf öğrenici çıkışı, diğer öğreniciye giriş olacak şekilde ardışık hesaplamalar ile eğitime devam edilir ve en sonunda zayıf öğrenicilerin ulaştığı sonuçlar birleştirilerek son karara ulaşılır.

Gradient Boosting (Gradyan Arttırma) Algoritması: Gradient Boosting algoritması ağaç temelli bir topluluk öğrenme yöntemidir, 2001'de Freidman geliştirmiştir (Freidman, 2001). AdaBoost ve Gradient Boosting arasındaki en büyük fark, iki algoritmanın zayıf öğrenicilerin eksikliklerini belirleme şeklidir. AdaBoost, eksiklikleri yüksek ağırlıklı veri noktalarını kullanarak tespit ederken, Gradient Boosting, aynı görevi kayıp fonksiyonundaki (loss function) gradyanları kullanarak gerçekleştirir. Yani yeni öngörücüyü önceki tahminci tarafından yapılan artık hatalara uydurmaya çalışır. Kayıp fonsiyonu temelde, modelin tahmininin olması gereken değerden farkını, yani hatayı hesaplamaktadır. Başarılı bir modelden beklenti 0'a yakın kayıp değerine sahip olmasıdır. Kayıp fonksiyonu hata hesaplarken, hata problemini bir optimizasyon problemine dönüştürerek yapar. Optimize edilecek hataya bağlı olarak uygun kayıp fonksiyonu seçilmelidir. Gradient Boosting, bir kayıp fonksiyonunun ulaştığı değeri en aza indirmeye dayanır; fakat daha az kontrol sunan ve esasen gerçek dünya uygulamalarıyla uyuşmayan bir kayıp fonksiyon yerine, kullanıcının belirlediği bir kayıp fonksiyonunun optimize edilmesine izin verir.

XGBoost (Extreme Gradient Boosting) Algoritması: XGBoost, karar ağacı tabanlı bir topluluk öğrenme algoritmasıdır. Gradient Boosting algoritmasının sistem optimizasyonu ve algoritmik geliştirmeler ile güncellenmiş yüksek performanslı versiyonudur. XGBoost, Washington Üniversitesi'nde bir araştırma projesi olarak geliştirilmiştir. İlk olarak Tianqi Chen ve Carlos Guestrin tarafından, 2016'da yayınlanan "XGBoost: A Scalable Tree Boosting System" adlı makale ile tanıtılmıştır (Chen ve Guestrin, 2016). Kaggle'ın düzenlediği yarışmada 29 problemin 17'sinde en yüksek performansı XGBoost algoritması göstermiştir.

LightGBM (Light Gradient Boosting Machine) Algoritması: LightGBM, Microsoft DMTK (Distributed Machine Learning Toolkit) projesi ile 2017 yılında geliştirilmiş bir boosting algoritmasıdır. Hesaplama hızı, bellek tüketimi ve tahmin oranı açısından diğer boosting algoritmalarına göre daha avantajlıdır. Farklı veri setleri ile yapılan deneylerden alınan sonuçlara göre LightGBM'in eğitim süresi, GBDT'ye (Gradient Boosting Decision Tree) göre 20 kat daha hızlıdır (Ke ve ark., 2017). LightGBM, diğer boosting algoritmalarından farklı olarak, karar ağaçlarının eğitiminde kullanılan seviye odaklı (level-wise or depth-wise) ve yaprak odaklı (leaf-wise) stratejilerinden, yaprak odaklı stratejiyi kullanarak, bölünme işlemine kaybı azaltan yapraklardan devam edildiği için hatayı azaltır, öğrenme hızını arttırır.

2.3. Modellerin Değerlendirilmesi

Literatürde STS modelleri geliştirilirken iki yaklaşımla karşılaşılmaktadır. İlk yaklaşımda çapraz doğrulama yöntemleri kullanılarak, modellerin hem eğitimi hem de test edilmesi için aynı veri seti kullanılmaktadır. Bu yaklaşımı kullanan çalışmalarda kurulan modeller, modelin veri setine aşırı uyum göstermesi; fakat yeni verilere uyum sağlayamaması olarak tanımlanan aşırı öğrenmeye maruz kaldığı için, çok yüksek başarı değerlerine ulaşmıştır fakat sonuçlar gerçekçi değildir. İkinci yaklaşımda ise, model eğitim sırasında eğitim veri seti ile eğitilmekte, ardından saldırı türü ve sayısı eğitim setinden farklı olan test veri seti ile test edilmektedir. İkinci yaklaşımda model

aşırı öğrenmeye daha az maruz kaldığı için birinci yaklaşıma göre daha düşük başarı değerleri elde edilse de sonuçlar daha gerçekçidir (Naseer ve Saleem, 2018). Bu çalışmada ikinci yaklaşım kullanılmış ve NSL-KDD veri setinde yer alan eğitim veri seti ile makine öğrenimi algoritmaları eğitilerek oluşturulan modellerin başarısı, saldırı türü ve sayısı eğitim setinden farklı olan test veri seti ile test edilmiştir. Bu bölümde kurulan modellerin başarısını değerlendirmek için kullanılan karışıklık matrisinden ve değerlendirme metriklerinden bahsedilmiştir.

2.3.1. Karışıklık Matrisi (Confusion Matrix)

Ön işlemlerden geçirilmiş ve ön işlem uygulanmamış eğitim veri setleri kullanılarak eğitilen sınıflandırma algoritmaları ile oluşturulan modellerin test veri seti ile test edilirken ne derecede başarılı olduklarını belirlemek için çeşitli değerlendirme metrikleri kullanılır ve bu metriklerde kullanılan değerler karışıklık matrisi'nden (Confusion Matrix) elde edilir.

Çizelge 2.3. Karışıklık matrisi.

Karışıklık Matrisi	**Tahmin edilen pozitif**	**Tahmin edilen negatif**
Gerçek pozitif	True positive(TP)	False negative(FN)
Gerçek negatif	False positive(FP)	True negative(TN)

Karışıklık matrisi, bir ekseninde gerçek değerlerin, diğer ekseninde tahmin edilen değerlerin olduğu matristir (Çizelge 2.3). Bu değerler kullanılarak, kurulan modelin başarısının yorumlanmasını sağlayan değerlendirme metrikleri hesaplanır.
Karışıklık matrisi'nde yer alan değerlerin tanımları şu şekildedir:

True positive (TP): Veri setinde saldırı sınıfındaki verilerin, kurulan model tarafından saldırı olarak tahmin edilmesi durumu.
False Negative (FN): Veri setinde saldırı sınıfındaki verilerin, kurulan model tarafından normal olarak tahmin edilmesi durumu.
False positive (FP): Veri setinde normal sınıfındaki verilerin, kurulan model tarafından saldırı olarak tahmin edilmesi durumu

True negative (TN): Veri setinde normal sınıfındaki verilerin, kurulan model tarafından normal olarak tahmin edilmesi durumu.

2.3.2. Değerlendirme Metrikleri

Kurulan modelin, problemi çözme konusunda ne kadar başarılı olduğuna karar vermek için, karışıklık matrisindeki değerler kullanılarak hesaplanan değerlendirme metrikleri:

Doğruluk (Accuracy): Doğru tahminlerin sayısının tüm örneklerin sayısına bölünmesiyle elde edilir (Denklem 7). Modelin genel performansı hakkında bilgi verir. Doğruluk skoru 0 ile 1 arasında olup 1'e yaklaştıkça modelin başarısı artar. Sınıf değişkeninin veri seti içinde dengeli dağıldığı durumlarda kullanılmalıdır. Dengesiz dağılmış bir veri setinde, model çoğunluk olan sınıfın değerini tahmin etme eğilimindedir, bu da taraflı bir yüksek doğruluk sonucuna neden olur. Bu sebeple dengesiz dağılmış veri setleri ile kurulan modeli değerlendirmek için doğruluğun yanında başka metriklere de bakılmalıdır.

Doğruluk= (TP+TN) / (TP+TN+FP+FN) **(7)**

Kesinlik (Precision): Saldırı olarak tahmin edilen örneklerin gerçekte kaç tanesinin saldırı olduğunu ölçer (Denklem 8). Kesinlik, özellikle FP tahminin maliyetinin, yüksek olduğu durumlarda kritik öneme sahiptir.

Kesinlik= TP / (TP+FP) **(8)**

Özgüllük (Specificity): Normal olan veri örneklerini normal olarak tahmin etme oranıdır (Denklem 9). Özgüllük değerinin yüksek olması, modelimizin gerçek negatifleri tahmin etmede iyi olduğu anlamına gelir.

Özgüllük= TN / (TN+FP) **(9)**

Duyarlılık (Recall): Saldırı örneklerin kaçının saldırı olarak tahmin edildiğini ölçer (Denklem 10). Sınıflandırıcıların bütünlüğünün bir ölçüsü olarak düşünülebilir. Duyarlılık, FN tahminin maliyetinin, yüksek olduğu durumlarda kritik öneme sahip bir metriktir. Mümkün olduğunca yüksek olması gereklidir.

Duyarlılık= TP / (TP+FN) **(10)**

F-ölçütü (F-skor): Kesinlik ve duyarlılık değerlerinin harmonik ortalamasıdır (Denklem 11). F-ölçütü, kesinlik ve duyarlılık arasındaki dengeyi sağlar. Tüm hata maliyetlerini de içerecek bir değerlendirme metriğine olan ihtiyacı karşılaması bakımından F-ölçütü önemlidir. Aritmetik ortalama yerine, harmonik ortalama kullanılmasının sebebi ise uç değerlere sahip veri örneklerinin göz ardı edilmesini engellemektir. Özellikle dengesiz dağılan veri setlerinde doğruluk metriğinden daha iyi bir yol göstericidir. F-ölçütü, literatürde f-skor, f-ölçütü, f1 puanı olarak da isimlendirilir.

F-ölçütü= (2*kesinlik*duyarlılık) / (kesinlik+duyarlılık) **(11)**

Matthews Correlation Coefficient (MMC): Gerçek veriler ile tahmin edilen veriler arasındaki korelasyon ilişkisine bakarak değerlendirme yapar (Denklem 12). Karışıklık matrisindeki tüm değerleri kullanarak -1 ile +1 arasında bir sonuca ulaşır. -1 tümü ile yanlış tahminleme yapıldığını gösterirken, +1 tümü ile doğru tahminleme yapıldığını gösterir.

MCC = (TP*TN) - (FP*FN) / √(TP+FP)(TP+FN)(TN+FP)(TN+FN) **(12)**

ROC Eğrisi (Receiver Operating Characteristic): . ROC eğrisi bir t eşiğinin değişen değerleri için X ekseni üzerindeki 1-özgüllük değerlerine karşılık gelen, Y eksenindeki duyarlılık değerlerinin kesişiminden oluşan noktaların birleştirilmesi ile çizilir. Eğri altında kalan alan ile AUC (Area Under the Curve) puanı hesaplanır. Eğri sol üst köşeye yaklaştıkça sonuç iyileşiyor demektir. En iyi sonuç için ROC eğrisi, (0,0)' ı (0,1)' e ve (0,1)' i (1,1)' e bağlayan çizgi ile elde edilir.

AUC (Area Under Curve): AUC çizilen ROC eğrisinin altında kalan alandır ve AUC ile tahmin doğruluğunun ortalaması hesaplanır. Çizilen ROC eğrisi solt üst köşeye

yani (0,1) noktasına ne kadar yakınsa altında kalan alan 1 değerine o kadar yaklaşacaktır. AUC için maksimum değer 1' dir ve bu değer, modelin ilgili veri seti için mükemmel sınıflandırma yapabildiğini gösterir. AUC 0.5 ise modelin sınıflandırma başarısı 0'dır.

PRC (Precision-Recall Curve): PRC eğrisi, kesinlik ve duyarlılık arasındaki ilişkiyi gösterir ve X eksinindeki duyarlılık, Y eksinindeki kesinlik değerlerinin kesişiminden geçerek oluşan bir eğridir. Dengesiz ve iki sınıflı veri setlerinde sınıflandırıcı performansını değerlendirirken PRC eğrisi ROC eğrisinden daha bilgilendiricidir (Saito ve Rehmsmeier, 2015). ROC eğirisinde olduğu gibi PRC eğrisinin altında kalan alana da AUC denir ve 1'e yaklaşması modelin sınıflandırma başarısının arttığı anlamına gelir.

Bu çalışmada kullanılan NSL-KDD gibi sınıf değişkeni dengesiz dağılan veri setleri ile geliştirilen saldırı tespit modellerinde doğruluk metriği, çoğunluk sınıfın lehine taraflı bir yüksek doğruluk sonucu vereceği için tercih edilmemiştir. Üzerinde çalışılan sınıflandırma problemi bir saldırı tespiti olduğu ve saldırı tespit modelinin FP ile FN tahmininin maliyetinin yüksek olduğu kritik durumlar göz önünde bulundurulduğu için, geliştirilen saldırı tespit modellerinin tespit başarısını değerlendirmek için kesinlik ve duyarlılık metriklerinin harmonik ortalaması olan F-ölçütü dikkate alınmıştır.

Bu bölümde, çalışmada kullanılan yazılım ve donanım materyalleri ve önerilen üç aşamalı metodoloji tanıtılmıştır. Metodolojinin 1.aşamasında veri setine uygulanan ön işlemler olan kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi, 2.aşamasında kullanılacak makine öğrenimi algoritmaları, 3. aşamasında kurulan saldırı tespit modellerinin performansını analiz etmek ve değerlendirmek için kullanılan değerlendirme metrikleri anlatılmıştır.

3.BULGULAR

Bu bölümde, önerilen üç aşamalı metodolojinin uygulaması: Ön işlem uygulanmayan, kategorik veri kodlama, ölçeklendirme, öznitelik seçimi ön işlemlerinin uygulandığı ve tüm ön işlemlerin uygulandığı veri setlerinin kullanıldığı beş farklı senaryo uygulanarak yapılmıştır. Metodolojinin 1. aşamasında veri setine, kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemleri uygulanmıştır. Metodolojinin 2. aşaması uygulanırken, ön işlem uygulanmayan veri setleri ve 1. aşamada elde edilen veri setleri ile KNN, MLP, RF, XGBoost ve LightGBM makine öğrenimi algoritmaları kullanılarak saldırı tespit modelleri oluşturulmuştur. Metodolojinin 3.aşaması uygulanırken, 2.aşamada oluşturulan modellerin başarısı, test veri seti kullanılarak eğitim süresi, test süresi, doğruluk, kesinlik, duyarlılık, F-ölçütü, MMC, ROC ve PRC metriklerine göre değerlendirilmiş ve karşılaştırılmıştır. Modellerin saldırı tespit başarısını değerlendirmek için, çalışmada kullanılan NSL-KDD gibi sınıf değişkeni dengesiz dağılan veri setlerinde doğruluk metriğinden daha sağlıklı fikir verdiği ve FP ile FN tahmininin maliyetinin yüksek olduğu kritik durumlar göz önünde bulundurulduğu için F-ölçütü dikkate alınmıştır. Uygulamalar süresince oluşturulan modeller, algoritma ve değerlendirme prosedürünün stokastik doğası veya sayısal hassasiyetteki farklılıklar nedeniyle, deneme yanılma ile ulaşılan, değerlerin stabile yaklaştığı çalıştırma sayısı kadar (beş kez) çalıştırılmış ve sonuçların ortalaması dikkate alınmıştır.

3.1. Veri Setinin Ön İşlenmesi Uygulamaları

Önerilen metodolojinin veri setinin ön işlenmesi aşamasında, NSL-KDD veri setine en uygun ön işlemi belirlemek amacıyla çeşitli kategorik veri kodlama teknikleri ve öznitelik seçim yöntemleri, ölçeklendirme ön işlemi için ise min-max yöntemi uygulanmıştır.

3.1.1. Veri Setleri Üzerinde Kategorik Kodlama Tekniklerinin Uygulanması ve Sonuçların Karşılaşılaştırılması

Kullanılacak NSL-KDD veri setine en uygun kategorik veri kodlama tekniğini belirleyebilmek için, veri setindeki kategorik verilere, sekiz farklı kategorik veri kodlama tekniği uygulanmış ve sınıflandırıcı olarak dengesiz veri setlerindeki başarısı, eğitim ve test hızının yüksek olmasından dolayı RF algoritması kullanılmıştır. Her kodlama tekniğinin, veri setine uygulanması sonucunda elde edilen, yeni eğitim ve test veri setinin öznitelik sayıları birbirinden farklı olacağı için, oluşturulan modellerin değerlendirilmesinde eğitim ve test için aynı veri setinin kullanıldığı 10'lu çapraz doğrulama yöntemi kullanılmıştır. K-çapraz doğrulama yöntemi ile veri seti belirlenen k sayısına göre eşit parçalara bölünür, her bir parçanın hem eğitim hem de test için kullanılması sağlanır. Oluşturulan modellerde değerlendirmeler boyut (öznitelik sayısı), eğitim süresi, ortalama sınıflandırma başarısı ve başarının standart sapması ölçütlerine göre yapılmıştır.

Çizelge 3.1. Kategorik veri kodlama tekniklerinin veri setine uygulanması ile alınan sonuçların karşılaştırılması.

Encoding tekniği	**Boyut (öznitelik sayısı)**	**Eğitim süresi (s)**	**Ortalama başarı**	**Başarının standart sapması**
Label Encoding	3	111.415914	0.998968	0.000263
One Hot Encoding	84	136.313674	0.998960	0.000228
Target Encoding	3	108.701497	0.998976	0.000214
Leave One Out Encoding	3	81.962925	1.0	0.0
Weight Of Evidence Encoding	3	109.278701	0.998968	0.000222
James Stein Encoding	3	107.59389	0.998984	0.000194
CatBoost Encoding	3	143.380519	0.998833	0.000198
M-Estimate Encoding	3	108.857498	0.998976	0.000196

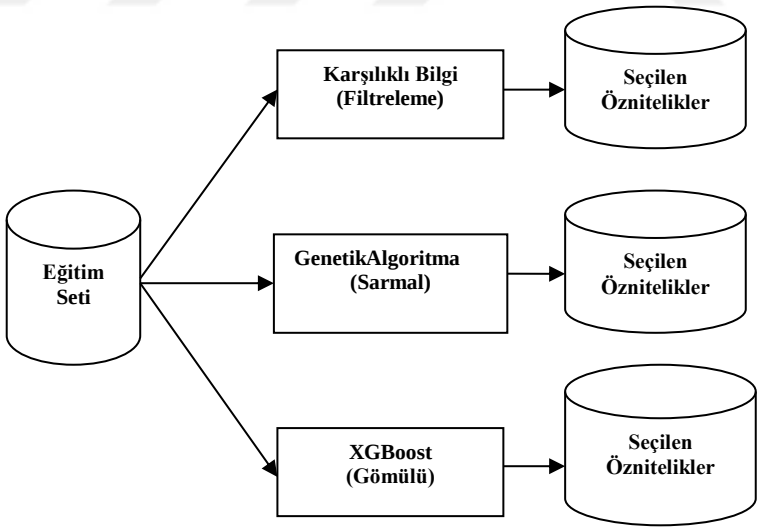
Çizelge 3.1'deki sonuçlar değerlendirildiğinde, öznitelik sayısını arttırmadığı, dolayısıyla işlem süresini uzatmadığı için, kullanılan diğer tekniklere kıyasla eğitim süresi kısa olduğu ve aşırı öğrenmeye karşı dirençli olduğu için, NSL-KDD veri setine en uygun kodlama tekniği leave one out encoding olarak belirlenmiştir.

3.1.2. Veri Setine Ölçeklendirme Ön İşleminin Uygulanması

Bu çalışmada NSL-KDD veri setine, literatürdeki STS çalışmalarında da yaygın olarak kullanılan min-max ölçeklendirme ön işlemi uygulanarak, tüm verilerin değerleri, en küçük değer 0 en büyük değer 1 olacak şekilde 0-1 aralığına yayılmıştır. Ölçeklendirme ön işleminin veri setine uygulanması için, scikit-learn ön işleme algoritmalarından 'Min-Max()' fonksiyonu kullanılmıştır.

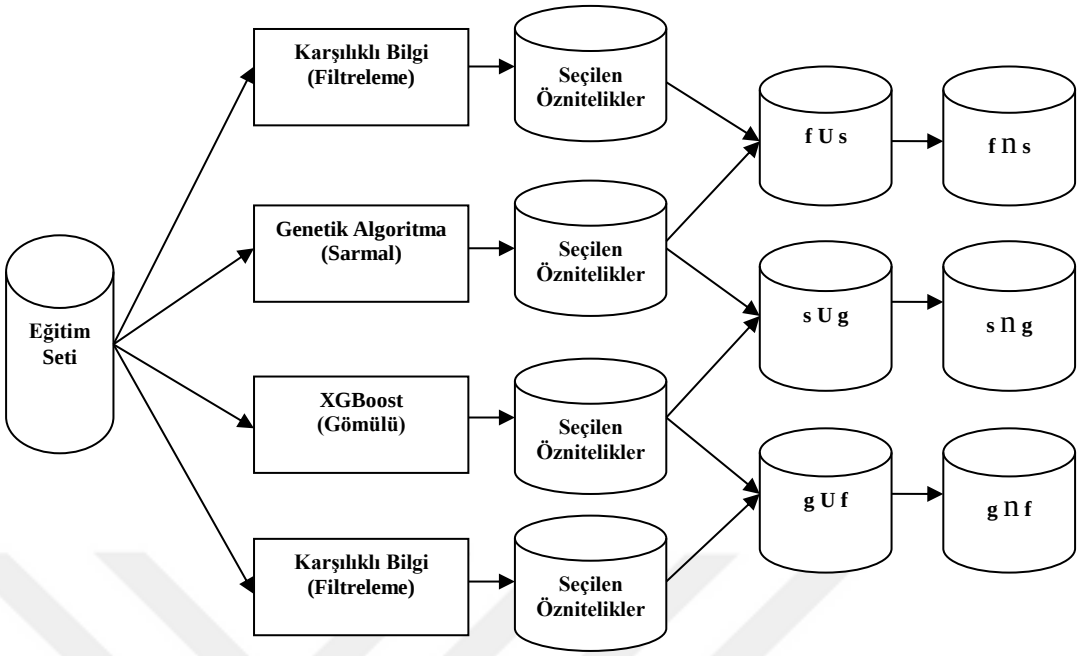
3.1.3. Veri Setine Hibrit Öznitelik Seçimi Ön İşleminin Uygulanması

Çalışmada, kümelerin keşişim ve birleşim özelliklerinden esinlenerek, hibrit öznitelik seçimi yöntemi NSL-KDD veri seti üzerinde üç adımda uygulanmıştır. Birinci adımda NSL-KDD veri seti üzerinde MI, GA ve XGBoost öznitelik seçim yöntemleri ayrı ayrı uygulanmış ve her bir yöntemin seçtiği özniteliklerden oluşan üç farklı veri seti elde edilmiştir (Şekil 3.1).



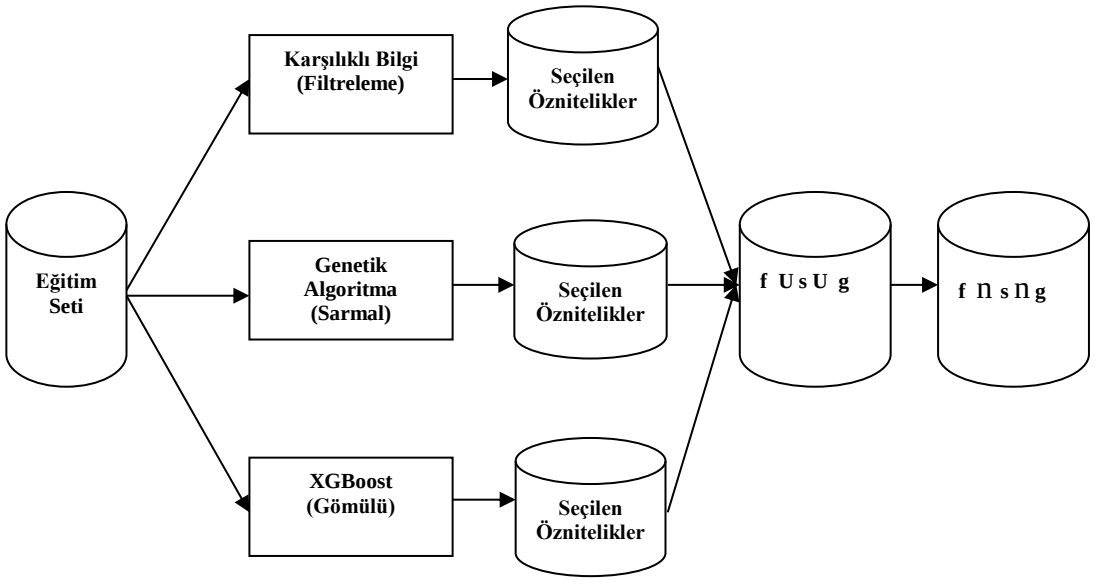
Şekil. 3.1. Kullanılan öznitelik seçim yöntemleri ile öznitelik seçimi.

Öznitelik seçimi ön işleminin ikinci adımında, birinci adımda elde edilen farklı özniteliklere ve öznitelik sayılarına sahip veri setlerindeki özniteliklerin ikili kesişim ve ikili birleşimlerinden oluşan, altı farklı veri seti elde edilmiştir (Şekil 3.2).



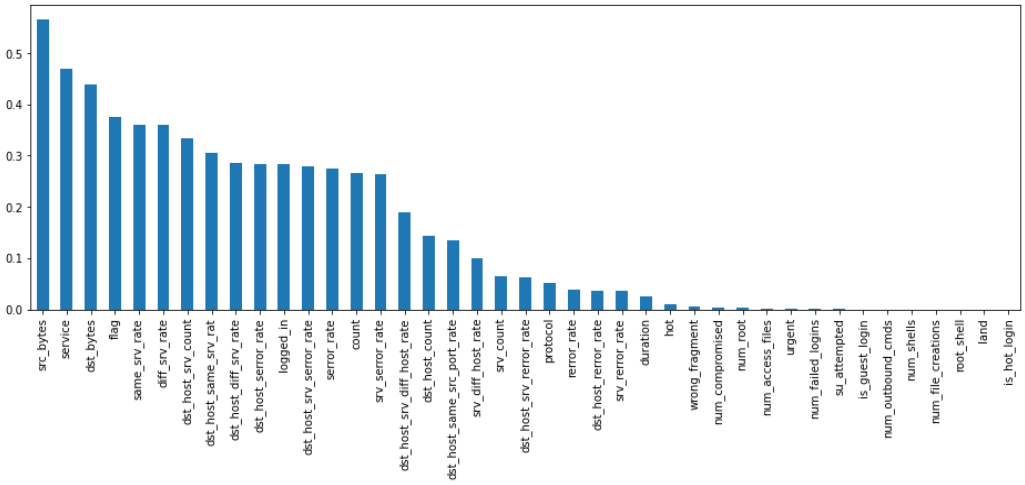
Şekil.3.2. Öznitelik seçim yöntemlerinin ürettiği öznitelik alt kümelerinin ikili kesişimleri ve birleşimleri. (f: filtreleme yöntemi MI, s: sarmal yöntemi GA, g: gömülü yöntemi XGBoost)

Öznitelik seçimi ön işleminin üçüncü adımında, birinci adımda elde edilen farklı özniteliklere ve öznitelik sayılarına sahip veri setlerindeki özniteliklerin üçlü kesişim ve üçlü birleşimlerinden iki farklı veri seti elde edilmiştir (Şekil 3.3).



Şekil.3.3. Öznitelik seçim yöntemlerinin ürettiği öznitelik alt kümelerinin üçlü kesişimleri ve birleşimleri.

MI öznitelik seçim yönteminin NSL-KDD veri setindeki öznitelik önem sıralaması Şekil 3.4'te, seçilen öznitelikler ise Çizelge 3.2'de gösterilmiştir.



Şekil.3.4. MI'ın öznitelik önem sıralaması.

Çizelge 3.2. MI ile seçilen öznitelikler ve numaraları

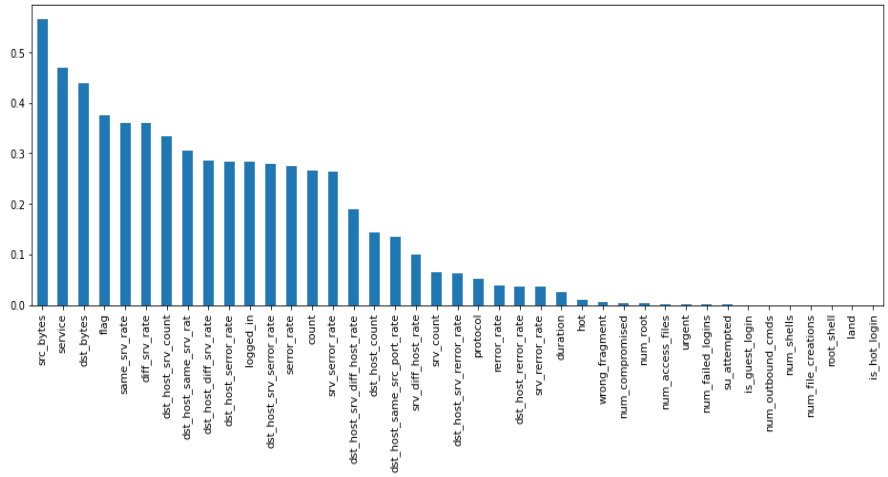
Kullanılan Yöntem	**Seçilen öznitelikler ve numaraları (26 öznitelik)**
MI (Filtreleme)	'src_bytes', 'service', 'dst_bytes', 'flag', 'same_srv_rate', 'diff_srv_rate', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_serror_rate', 'logged_in', 'dst_host_srv_serror_rate', 'serror_rate','count', 'srv_serror_rate', 'dst_host_srv_diff_host_rate', 'dst_host_count', 'dst_host_same_src_port_rate', 'srv_diff_host_rate', 'srv_count', ' dst_host_srv_rerror_rate', 'protocol_type', 'rerror_rate', 'dst_host_rerror_rate', 'srv_rerror_rate', 'duration',
	5,3,6,4,29,30,33,34,35,38,12,39,25,23,26,37,32,36,31,24,41,2,27,40,28,1

GA sarmal yöntemin, LR sınıflandırma algoritmasını kullanarak seçtiği öznitelikler Çizelge 3.3' te gösterilmiştir.

Çizelge 3.3. GA'nın seçtiği öznitelikler

Kullanılan Yöntem	**Seçilen öznitelikler ve numaları (21 öznitelik)**
GA(Logistic Regression)	'protocol_type', 'flag', 'dst_bytes', 'wrong_fragment', 'num_failed_logins', 'logged_in', 'num_compromised', 'su_attempted', 'num_root', 'num_file_creations', 'num_shells', 'num_outbound_cmds', 'serror_rate', 'rerror_rate', 'srv_diff_host_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_diff_srv_rate', 'dst_host_srv_diff_host_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate'
	2,4,6,8,11,12,13,15,16,17,18,20,25,27,31,32,33,35,37,38,39

XGBoost gömülü öznitelik seçim yönteminin veri setindeki öznitelik önem sıralaması Şekil 3.5'te, seçilen öznitelikler ise Çizelge 3.4'te gösterilmiştir.



Şekil 3.5. XGBoost'un öznitelik önem sıralaması.

Çizelge 3.4. XGBoost ile seçilen öznitelikler ve numaraları.

Kullanılan yöntem	**Seçilen öznitelikler ve numaraları (24 öznitelik)**
XGBoost(Gömülü)	'src_bytes', 'count', 'hot', 'wrong_fragment', 'srv_count', 'dst_host_srv_rerror_rate', 'dst_host_srv_count', 'dst_bytes', 'dst_host_same_src_port_rate', 'logged_in', 'dst_host_same_srv_rate', 'num_failed_logins', 'flag', 'dst_host_srv_diff_host_rate', 'service', 'duration', 'srv_rerror_rate', 'dst_host_srv_serror_rate', 'dst_host_rerror_rate', 'dst_host_serror_rate', 'dst_host_diff_srv_rate', 'dst_host_count', 'num_shells', 'root_shell', 'su_attempted', 'serror_rate', 'srv_serror_rate'
	5,23,10,8,24,33,6,36,12,34,11,4,37,3,1,28,39,35,32,18,14,15,25,26.

Çizelge 3.5. Öznitelik seçim tekniklerinin ürettiği öznitelik alt kümeleri ile bu alt kümelerin ikili, üçlü kesişim ve birleşimleri. (f:MI(filtreleme), s: GA(sarmal), g: XGBoost(gömülü), Küme: Öznitelik alt kümesi)

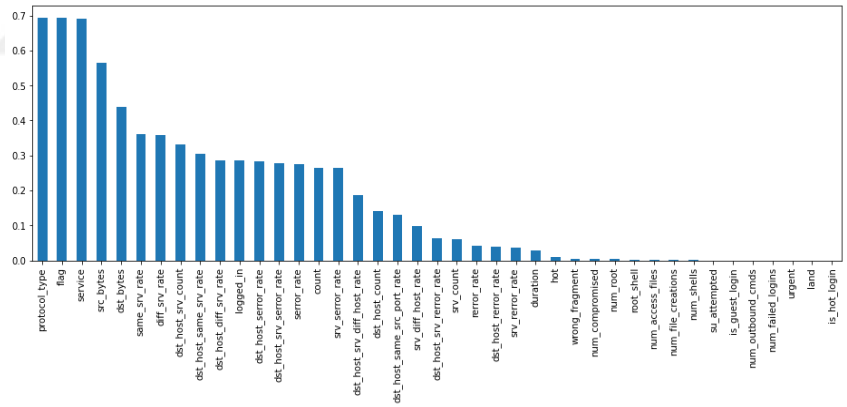
Küme	**Öznitelik No:**
f	1,2,3,4,5,6,12,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41.
s	2,4,6,8,11,12,13,15,16,17,18,20,25,27,31,32,33,35,37,38,39.
g	1,3,4,5,6,8,10,11,12,14,15,18,23,24,25,26,28,32,33,34,35,36,37,39.
fns	2,4,6,12,25,27,31,32,33,35,37,38,39.
fng	1,3,4,5,6,12,23,24,25,26,28,32,33,34,35,36,37,39.
gns	4,6,8,11,12,15,18,25,32,33,35,37,39.
fus	1,2,3,4,5,6,8,11,12,13,15,16,17,18,20,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39, 40,41.
fug	1,2,3,4,5,6,8,10,11,12,14,15,18,23, 24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41.
gus	1,2,3,4,5,6,8,10,11,12,13,14,15,16,17,18,20,23,24,25,26,27,28,31,32,33,34,35,36,37,38,39.
fnsng	4,6,12,25,32,33,35,37,39.
fusug	1,2,3,4,5,6,8,10,11,12,13,14,15,16,17,18,20,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37, 38,39,40,41.

Öznitelik seçimi ön işleminin son adımında karşılıklı bilgi, genetik algoritma ve XGBoost gömülü öznitelik seçim yöntemlerinin ürettiği öznitelik alt kümelerine ve bu

öznitelik alt kümelerinin ikili ve üçlü kesişim ile birleşimlerine sahip 11 farklı veri seti (Çizelge 3.5) elde edilmiştir.

3.1.4. Kategorik Veri Kodlama ve Ölçeklendirme Ön İşlemleri Yapılan Veri Setinden Hibrit Öznitelik Seçim Yöntemi ile Öznitelik Seçimi Ön İşlemi

Çalışmanın bu aşamasında, NSL-KDD veri setlerine, sekiz farklı kategorik veri kodlama tekniğinin uygulandığı testler sonucunda elde edilen Çizelge 3.1'deki sonuçlara göre en başarılı kodlama tekniği seçilen leave one out encoding tekniği, daha sonra min-max ölçeklendirme ön işlemleri uygulamıştır. Daha sonra bu iki ön işlemin de uygulandığı veri setleri kullanılarak MI, GA ve XGBoost öznitelik seçim yöntemleri ile öznitelik seçimi yinelenmiştir. MI'nın öznitelik seçim yönteminin kategorik veri kodlama ve ölçeklendirme ön işlemlerinden geçmiş NSL-KDD veri setindeki öznitelik önem sıralaması Şekil 3.6'da, seçtiği öznitelikler Çizelge 3.6'da gösterilmiştir.



Şekil.3.6. Ön işlemlerden geçmiş veri setinde MI'ın öznitelik önem sıralaması.

Çizelge 3.6. Ön işlemlerden geçmiş veri setinden MI ile seçilen öznitelikler ve numaraları.

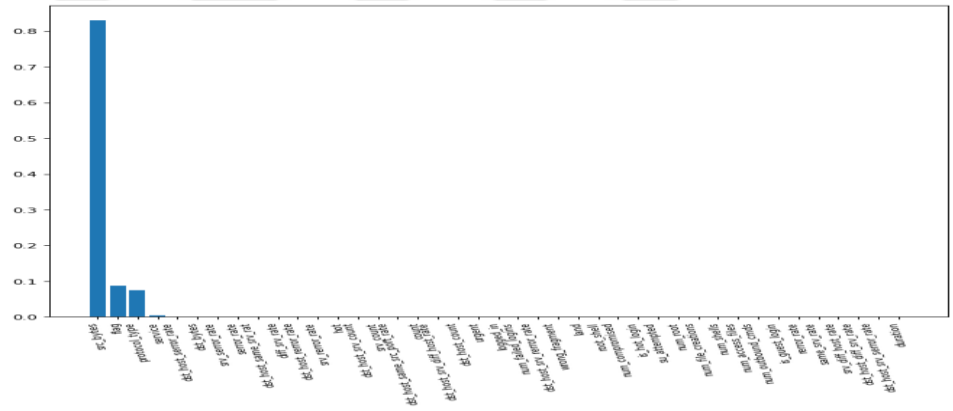
Kullanılan yöntem	**Seçilen öznitelikler ve numaraları (28 öznitelik)**
MI (Filtre)	'duration', 'protocol_type', 'service', 'flag', 'src_bytes', 'dst_bytes', 'wrong_fragment', 'hot', 'logged_in', 'count', 'srv_count', 'serror_rate', 'srv_serror_rate', 'rerror_rate', 'srv_rerror_rate', 'same_srv_rate', 'diff_srv_rate', 'srv_diff_host_rate', 'dst_host_count', 'dst_host_srv_count', 'dst_host_same_srv_rate', 'dst_host_diff_srv_rate', 'dst_host_same_src_port_rate', 'dst_host_srv_diff_host_rate', 'dst_host_serror_rate', 'dst_host_srv_serror_rate', 'dst_host_rerror_rate', 'dst_host_srv_rerror_rate'
	1,2,3,4,5,6,8,10,12,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41

GA öznitelik seçim yönteminin, RF sınıflandırma algoritması kullanarak, kategorik veri kodlama ve ölçeklendirme ön işlemlerinden geçmiş veri setinden seçtiği öznitelikler Çizelge 3.7'de gösterilmiştir.

Çizelge 3.7. Ön işlemlerden geçmiş veri setinden GA'nın seçtiği öznitelikler.

Kullanılan yöntem	**Seçilen öznitelikler ve numaları (14 öznitelik)**
GA (Random Forest)	'src_bytes', 'wrong_fragment', 'hot', 'num_root', 'num_shells', 'num_outbound_cmds', 'is_hot_login', 'serror_rate', 'rerror_rate', 'diff_srv_rate', 'dst_host_same_srv_rate', 'dst_host_srv_diff_host_rate', 'dst_host_rerror_rate', 'dst_host_srv_rerror_rate'
	5,8,10,16,18,20,21,25,27,30,34,37,40,41

XGBoost öznitelik seçim yönteminin kategorik veri kodlama ve ölçeklendirme ön işlemlerinden geçmiş veri setindeki öznitelik önem sıralaması Şekil 3.7'de, seçtiği öznitelikler Çizelge 3.8'de gösterilmiştir.



Şekil 3.7. Ön işlemlerden geçmiş veri setinde XGBoost'un öznitelik önem sıralaması.

Çizelge 3.8. Ön işlemlerden geçmiş veri setinden XGBoost ile seçilen öznitelikler ve numaraları.

Kullanılan yöntem	**Seçilen öznitelikler ve numaraları (18 öznitelik)**
XGBoost (Gömülü)	'src_bytes', 'flag', 'protocol_type', 'service', 'dst_host_serror_rate', 'dst_bytes', 'srv_serror_rate', 'serror_rate', 'dst_host_same_srv _rate', 'diff_srv_rate', 'dst_host_rerror_rate', 'srv_rerror_rate', 'hot', 'dst_host_srv_count', 'srv_count', 'dst_host_same_src_port_rate', 'count', 'dst_host_srv_diff_host_rate'
	5,4,2,3,38,6,26,25,34,30,40,28,10,33,24,36,23,37

MI, GA ve XGBoost öznitelik seçim yöntemlerinin, ön işlemlerden geçmiş veri setinden ürettikleri öznitelik alt kümelerine ve bu öznitelik alt kümelerinin ikili ve üçlü kesişim ile birleşimlerine sahip 11 farklı veri seti elde edilmiştir (Çizelge 3.9).

Çizelge 3.9. Ön işlemlerden geçmiş veri setinden öznitelik seçim tekniklerinin ürettiği öznitelik alt kümeleri ile bu alt kümelerin ikili, üçlü kesişim ve birleşimleri.
(f:MI(filtreleme), s: GA(sarmal), g: XGBoost(gömülü), Küme: Öznitelik alt kümesi)

Küme	**Öznitelik No:**
f	1,2,3,4,5,6,8,10,12,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41
s	5,8,10,16,18,20,21,25,27,30,34,37,40,41
g	2,3,4,5,6,10,23,24,25,26,28,30,33,34,36,37,38,40
fns	5,8,10,25,27,30,34,37,40,41
fng	2,3,4,5,6,10,23,24,25,26,28,30,33,34,36,37,38,40,
gns	5,10,25,30,34,37,40
fus	1,2,3,4,5,6,8,10,12,16,18,20,21,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41
fug	1,2,3,4,5,6,8,10,12,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41
gus	2,3,4,5,6,8,10,16,18,20,21,23,24,25,26,27,28,30,33,34,36,37,38,40,41
fnsng	5,10,25,30,34,37,40
fusug	1,2,3,4,5,6,8,10,12,16,18,20,21,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41

Çizelge 3.9 ve Çizelge 3.5 karşılaştırıldığında, kategorik kodlama ve ölçeklendirme ön işlemleri uygulanmış veri setinden, öznitelik seçim yöntemleri ile seçilen öznitelikler, ön işlem uygulanmamış veri setinden seçilen özniteliklerden farklıdır. Dolayısıyla bu iki farklı öznitelik seçimi uygulamasının, saldırı tespit başarısına etkilerinin farklı olması beklenmektedir.

3.2. Saldırı Tespit Modellerinin Kurulması ve Değerlendirilmesi

Beş farklı senaryo kullanılarak ön işlem uygulanmayan ve ön işlemlerden geçirilerek oluşturulan veri setleri kullanılarak KNN, MLP, RF, XGBoost, LightGBM makine öğrenimi algoritmaları ile saldırı tespit modelleri oluşturulmuş, bu modellerin başarısı test veri seti kullanılarak, eğitim süresi, test süresi, doğruluk, hassasiyet, F-ölçütü, MMC, ROC, PRC metrikleriyle analiz edilmiş ve değerlendirilmiştir.

3.2.1. 1.Senaryo: Makine Öğrenimi Algoritmalarının Ön İşlem Yapılmayan Veri Setleri Üzerinde Uygulanması

Birinci senaryo uygulanırken, veri setine ön işlem uygulanmadan KNN, MLP, RF, XGBoost, LightGBM algoritmaları ile modeller oluşturulmuş ve test veri seti ile test edilmiştir. Yapılan işlem sonunda elde edilen, Çizelge 3.10'da görülen sonuçlar

değerlendirdiğinde, veri seti üzerinde ön işlem yapılmadığında, eğitim veri seti üzerinde en kısa sürede saldırı tespitini, 0,031 s ile KNN algoritması yapabilmişken; en uzun saldırı tespit süresi 18,675 s ile MLP algoritmasına aittir. Test veri seti üzerinde ise en kısa saldırı tespiti süresine 0,031 s ile yine MLP algoritmasında, en uzun saldırı tespit süresine ise 64,146 s ile KNN algoritmasında ulaşılmıştır. Saldırı tespit başarısı için F-ölçütü'ne bakıldığında %78,5 ile XGBoost ve LightGBM algoritmaları ile alınan sonuçlar diğer metriklerde de olduğu gibi, diğer algoritmalar ile alınan sonuçlara göre daha başarılıdır; fakat genel olarak elde edilen saldırı tespit başarıları ve hızları yeterli değildir. Uygulama sonucunda veri setine ön işlem uygulanmadığında, makine öğrenimi algoritmalarından yeterince performans alınamadığı açıkça görülmektedir.

Çizelge 3.10. Veri setleri üzerinde ön işlem yapılmadan alınan sonuçlar

Algoritmalar	Eğitim süresi (s)	Test süresi (s)	Doğruluk	Kesinlik	Duyarlılık	F-ölçütü	MMC	ROC alanı	PRC alanı
KNN	0,031	64,146	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
MLP	18,675	0,031	77,3%	93,0%	65,1%	76,6%	59,3%	89,4%	92,1%
RF	12,531	0,334	77,1%	96,7%	61,9%	75,5%	60,8%	94,2%	95,9%
XGBoost	16,010	0,047	79,4%	96,7%	66,1%	78,5%	64,1%	96,6%	97,0%
LightGBM	1,582	0,091	79,4%	96,6%	66,1%	78,5%	63,9%	96,4%	96,9%

3.2.2. 2.Senaryo: Makine Öğrenimi Algoritmalarının Kategorik Veri Kodlama Ön İşlemi Yapılan Veri Setleri Üzerinde Uygulanması

İkinci senaryo uygulanırken, NSL-KDD eğitim veri setine kategorik veri kodlama ön işlemi yapılarak elde edilen Çizelge 3.1'deki sonuçlara göre en başarılı kodlama yöntemi seçilen leave one out encoding kategorik veri kodlama tekniği uygulanarak elde edilen veri setleri üzerinde makine öğrenimi algoritmaları çalıştırılarak oluşturulan modeller test veri seti ile test edilmiş ve Çizelge 3.11'deki sonuçlara ulaşılmıştır.

Çizelge 3.11. Kategorik veri kodlama ön işlemi uygulanarak alınan sonuçlar

Algoritmalar	Eğitim süresi (s)	Test süresi (s)	Doğruluk	Kesinlik	Duyarlılık	F-ölçütü	MMC	ROC alanı	PRC alanı
KNN	0,051	50,899	76,6%	96,6%	61,1%	74,8%	60,1%	82,3%	91,2%
MLP	13,377	0,028	78,2%	93,6%	66,3%	77,6%	60,8%	89,9%	92,5%
RF	8,349	0,182	80,7%	96,9%	68,3%	80,1%	66,1%	97,1%	97,4%
XGBoost	4,816	0,018	83,0%	97,1%	72,3%	82,9%	69,6%	97,4%	97,7%
LightGBM	0,573	0,036	84,2%	96,9%	74,7%	84,4%	71,4%	97,0%	97,5%

Veri setine kategorik veri kodlama ön işlemi uygulandığında ve Çizelge 3.11'deki sonuçlar, ön işlem uygulanmamış veri seti ile alınan Çizelge 3.10'daki sonuçlarla karşılaştırılarak değerlendirildiğinde, eğitim veri seti üzerindeki saldırı tespit hızı yaklaşık olarak, MLP'de %28, RF'de %33, XGBoost'da %70, LightGBM'de %64 oranlarında artarken, KNN'de %65 oranında düşmüştür. Test veri seti üzerindeki saldırı tespit hızı ise yaklaşık olarak, KNN'de %21, MLP'de %10, RF'de %45, XGBoost'ta % 62, LightGBM'de %60 oranlarında artmıştır. Saldırı tespit başarısını analiz etmek için F-ölçütü metriğine baktığımızda ise, KNN'de %0,9 oranında bir düşüş olmuştur, MLP'de %1, RF'de %4,6, XGBoost'ta %4,4, LightGBM'de %5,9 oranlarında bir yükseliş olmuştur. Uygulama sonucunda, veri setine kategorik veri kodlama ön işlemi uygulandığında KNN dışında, eğitim ve test veri setleri üzerindeki saldırı tespit hızlarında kayda değer bir artış, saldırı tespit başarısında ise yaklaşık %0,9 ile %5,9 arasında bir iyileşme görülmüştür.

3.2.3. 3.Senaryo: Makine Öğrenimi Algoritmalarının Ölçeklendirme Ön İşlemi Yapılan Veri Setleri Üzerinde Uygulanması

Üçüncü senaryo uygulanırken, Min-Max yöntemiyle ölçeklendirme ön işlemi yapılan veri setleri üzerinde, KNN, MLP, RF, XGBoost, LightGBM algoritmaları çalıştırılmıştır ve tespit süreleri ve tespit oranları analiz edilmiştir. Çizelge 3.12'de görülen sonuçlar ön işlem uygulanmamış veri seti ile alınan sonuçlarla kıyaslanmıştır. Eğitim veri seti üzerindeki saldırı tespit hızı yaklaşık olarak, KNN'de %48, LightGBM'de %16 oranlarında artmış, RF'de %4 ve XGBoost'ta %1 oranlarında, MLP ise yaklaşık 6 kat düşmüştür. Test veri seti üzerindeki saldırı tespit hızı ise yaklaşık olarak, RF'de %3, XGBoost'ta %13, LightGBM'de %14 oranlarında artmış,

KNN'de %4, MLP'de %123 oranlarında düşmüştür. F-ölçütüne baktığımızda ise; KNN'de %0,3, MLP'de %2, XGBoost'ta %0,4 oranlarında artış olmuş, RF'de %0,3 ve LightGBM'de %1,5 oranlarında bir düşüş olmuştur. Uygulama sonucunda, veri setine ölçeklendirme ön işlemi uygulandığında MLP algoritmasının saldırı tespit hızlarında ciddi bir düşüş görülürken LightGBM'nin eğitim ve test veri seti üzerindeki saldırı tespit hızı kayda değer ölçüde artmıştır. KNN algoritmasının eğitim veri üzerindeki saldırı tespit hızı kayda değer oranda artarken, test veri setindeki tespit hızı küçük oranda azalmıştır. F-ölçütünde ise %0,3 ile %2 arasında düşük oranlarda bir artış veya azalış söz konusudur. Uygulama sonunda ölçeklendirme ön işlemi, saldırı tespit hızı üzerinde bazı algoritmalarda olumlu bir etki yaparken, bazı algoritmalarda tespit hızını düşürmüştür. Saldırı tespit başarılarında ise çok küçük oranlarda bir etki söz konusudur.

Çizelge 3.12. Veri setleri üzerinde ölçeklendirme ön işlemi yapılarak alınan sonuçlar

Algoritmalar	Eğitim süresi (s)	Test süresi (s)	Doğruluk	Kesinlik	Duyarlılık	F-ölçütü	MMC	ROC alanı	PRC alanı
KNN	0,016	66,670	77,6%	97,3%	62,3%	76,0%	61,8%	83,5%	91,9%
MLP	133,871	0,069	79,5%	96,7%	66,4%	78,7%	64,3%	95,6%	96,4%
RF	13,004	0,325	76,9%	96,6%	61,6%	75,2%	60,5%	92,6%	94,8%
XGBoost	16,194	0,041	79,7%	96,5%	66,6%	78,9%	64,4%	92,4%	95,0%
LightGBM	1,326	0,078	78,2%	96,2%	64,2%	77,0%	62,1%	92,8%	95,3%

3.2.4. 4.Senaryo: Makine Öğrenimi Algoritmalarının Öznitelik Seçimi Ön İşlemi Yapılan Veri Setleri Üzerinde Uygulanması

Dördüncü senaryo uygulanırken, ön işlem yapılmayan veri setinde, MI, GA ve XGBoost gömülü öznitelik seçim yöntemleri uygulanarak elde edilen 11 veri seti (Çizelge 3.5) ve KNN, MLP, RF, XGBoost, LightGBM makine öğrenimi algoritmaları ile toplamda 55 adet model oluşturulmuştur. Modeller test veri seti ile test edilmiş, eğitim süresi, test süresi, doğruluk, kesinlik, duyarlılık, F-ölçütü, MMC, ROC ve PRC metrikleriyle değerlendirilmiştir. Alınan sonuçlar KNN kullanan modeller için Çizelge 3.13'de, MLP kullanan modeller için Çizelge 3.14'te, RF kullanan modeller için Çizelge 3.15'te, XGBoost kullanan modeller için Çizelge 3.16'da, LightGBM kullanan modeller için Çizelge 3.17'de gösterilmiştir.

Çizelge 3.13. Öznitelik seçimi ön işlemi uygulanan veri setleri ile KNN analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC Alanı
f	0,020	50,516	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
s	0,017	50,347	75,4%	95,4%	59,6%	73,4%	57,8%	81,9%	90,8%
g	0,019	50,143	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
fns	14,386	1,397	72,6%	94,7%	54,9%	69,5%	53,5%	79,2%	89,2%
fng	0,017	51,320	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
gns	15,440	1,391	76,1%	95,2%	61,1%	74,4%	58,6%	83,1%	91,2%
fus	0,031	53,262	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
fug	0,032	53,453	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
gus	0,025	56,099	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
fnsng	11,723	1,164	76,2%	94,8%	61,5%	74,6%	58,6%	83,1%	91,2%
fusug	0,026	55,000	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%

Çizelge 3.14. Öznitelik seçimi ön işlemi uygulanan veri setleri ile MLP analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F – ölçütü	MMC	ROC alanı	PRC Alanı
f	11,380	0,020	76,0%	91,7%	63,7%	75,2%	56,2%	87,2%	90,5%
s	6,004	0,020	72,1%	92,2%	59,0%	71,9%	54,1%	88,2%	86,9%
g	12,165	0,021	77,7%	93,6%	65,4%	77,0%	60,2%	90,8%	93,0%
fns	7,742	0,018	69,2%	94,5%	48,9%	63,5%	49,0%	88,5%	88,3%
fng	9,910	0,021	75,6%	93,1%	61,9%	74,3%	56,9%	87,3%	91,1%
gns	7,842	0,019	74,3%	90,0%	61,4%	72,9%	53,5%	87,0%	84,8%
fus	10,362	0,022	77,4%	94,2%	64,3%	76,4%	60,0%	88,7%	92,1%
fug	10,067	0,025	77,6%	95,4%	63,7%	76,4%	60,8%	88,5%	92,3%
gus	11,200	0,022	78,2%	95,1%	65,0%	77,2%	61,6%	89,9%	92,8%
fnsng	7,945	0,018	73,7%	90,4%	60,2%	72,3%	52,9%	87,0%	86,2%
fusug	13,864	0,024	77,5%	94,0%	64,9%	76,5%	60,2%	88,9%	92,3%

Çizelge 3.15. Öznitelik seçimi ön işlemi uygulanan veri setleri ile RF analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC Alanı
f	8,582	0,179	77,5%	96,7%	62,7%	76,0%	61,4%	93,7%	95,7%
s	5,968	0,215	73,1%	96,4%	54,9%	69,9%	55,2%	94,7%	96,4%
g	7,073	0,184	76,9%	96,8%	61,5%	75,2%	60,7%	93,5%	95,5%
fns	5,518	0,197	69,4%	94,9%	48,9%	64,5%	49,4%	92,2%	94,3%
fng	0,937	0,181	76,7%	96,7%	61,2%	75,0%	60,3%	92,6%	94,7%
gns	5,509	0,215	74,4%	95,4%	57,9%	72,0%	56,4%	94,3%	95,7%
fus	7,663	0,186	77,2%	96,7%	62,0%	75,6%	61,0%	94,3%	96,0%
fug	7,960	0,181	76,9%	96,7%	61,5%	75,2%	60,5%	93,7%	95,7%
gus	7,804	0,190	76,0%	96,6%	59,9%	73,9%	59,2%	93,4%	95,5%
fnsng	5,522	0,203	74,6%	95,4%	58,2%	72,3%	56,7%	93,0%	95,1%
fusug	8,850	0,181	77,3%	96,6%	62,2%	75,7%	61,0%	94,2%	95,9%

Çizelge 3.16. Öznitelik seçimi ön işlemi uygulanan veri setleri ile XGBoost analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F - ölçütü	MMC	ROC alanı	PRC alanı
f	3,759	0,018	79,3%	96,7%	65,9%	78,4%	63,9%	96,0%	96,6%
s	2,940	0,015	75,3%	96,5%	58,7%	73,0%	58,2%	96,1%	96,3%
g	3,595	0,020	78,8%	96,5%	65,1%	77,8%	63,2%	96,4%	96,8%
fns	2,376	0,014	71,0%	94,6%	52,1%	67,2%	51,4%	95,2%	95,6%
fng	2,942	0,015	77,8%	95,6%	64,0%	76,6%	61,2%	95,7%	96,3%
gns	2,305	0,014	75,5%	95,2%	59,9%	73,5%	57,8%	95,3%	95,7%
fus	4,227	0,020	79,9%	96,7%	66,9%	79,1%	64,8%	96,7%	97,0%
fug	4,158	0,018	80,3%	96,7%	67,7%	79,7%	65,4%	96,4%	96,9%
gus	4,181	0,019	78,5%	96,6%	64,5%	77,4%	62,7%	96,8%	97,3%
fnsng	2,112	0,014	76,3%	95,2%	61,5%	74,7%	58,9%	95,4%	96,1%
fusug	4,441	0,019	79,3%	96,7%	65,9%	78,4%	64,0%	96,7%	97,2%

Çizelge 3.17. Öznitelik seçimi ön işlemi uygulanan veri setleri ile LightGBM analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC alanı
f	0,530	0,033	79,4%	96,6%	66,1%	78,5%	64,0%	96,0%	96,8%
s	0,361	0,038	74,9%	96,7%	57,8%	72,4%	57,7%	95,6%	96,0%
g	0,480	0,043	77,9%	96,3%	63,6%	76,6%	61,7%	96,0%	96,7%
fns	0,354	0,030	72,8%	94,7%	55,3%	69,8%	53,8%	92,2%	93,8%
fng	0,487	0,035	76,9%	95,4%	62,5%	75,5%	59,9%	95,1%	96,2%
gns	0,312	0,030	76,2%	95,0%	61,3%	74,6%	58,7%	94,6%	95,2%
fus	0,599	0,037	80,0%	96,7%	67,1%	79,3%	64,9%	96,5%	97,0%
fug	0,556	0,038	78,7%	96,4%	65,1%	77,7%	62,9%	96,2%	96,7%
gus	0,510	0,033	76,8%	96,4%	61,6%	75,1%	60,2%	95,8%	96,5%
fnsng	0,319	0,031	76,4%	95,0%	61,8%	74,9%	58,9%	94,9%	95,7%
fusug	0,554	0,038	80,0%	96,7%	67,3%	79,3%	65,9%	96,3%	96,9%

Algoritmaların en başarılı oldukları öznitelik alt kümeleri Çizelge 3.18'de gösterilmiştir.

Çizelge 3.18. En başarılı öznitelik alt kümesi ile algoritma birleşimi ve analiz sonuçları

Öznitelik alt kümesi+ algoritma	Eğitim süresi (s)	Test süresi (s)	Doğ-ruluk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC alanı
fng + KNN	0,017	51,320	77,2%	96,4%	62,3%	75,7%	60,8%	82,8%	91,3%
gus + MLP	11,200	0,022	78,2%	95,1%	65,0%	77,2%	61,6%	89,9%	92,8%
fng + RF	0,937	0,181	76,7%	96,7%	61,2%	75,0%	60,3%	92,6%	94,7%
fug + XGBoost	4,158	0,018	80,3%	96,7%	67,7%	79,7%	65,4%	96,4%	96,9%
fusug + LightGBM	0,554	0,038	80,0%	96,7%	67,3%	79,3%	65,9%	96,3%	96,9%

Çizelge 3.18'deki sonuçlara göre KNN algoritmasının en başarılı modeli kurduğu veri seti, filtreleme yöntemlerinden olan MI ve gömülü yöntemlerden olan XGBoost öznitelik seçim yöntemlerinin ortak seçtiği özniteliklerden oluşan özniteliklere sahip veri setidir. MLP algoritması en başarılı modeli; XGBoost ve sarmal yöntemlerden olan GA'nın seçtiği özniteliklerin birleşiminden oluşan özniteliklere sahip veri seti ile RF algoritması; MI ve XGBoost'un seçtikleri özniteliklerin kesişiminden, XGBoost algortiması; MI ve XGBoost'un seçtikleri özniteliklerin birleşiminden, LightGBM

algoritması ise en başarılı modeli; MI, GA ve XGBoost'un seçtikleri özniteliklerin birleşiminden oluşan özniteliklere sahip veri seti ile kurmuşlardır. Çizelge 3.18'deki sonuçlar, ön işlem uygulanmamış veri setinin kullanıldığı Çizelge 3.10'daki sonuçlarla karşılaştırılarak değerlendirildiğinde, eğitim veri seti üzerindeki saldırı tespit hızı yaklaşık olarak, KNN'de %45 MLP'de %40, RF'de %92, XGBoost'da %74, LightGBM'de %65 oranlarında artmıştır. Test veri seti üzerindeki ise saldırı tespit hızı yaklaşık olarak, KNN'de %20, MLP'de %29, RF'de %46, XGBoost'ta %62, LightGBM'de %58 oranlarında artmıştır. F-ölçütüne baktığımızda ise, KNN'de bir değişiklik olmamış, RF'de ise %0,5 oranında bir düşüş olmuştur. MLP'de %0,6, XGBoost'ta %1,2, LightGBM'de %0,8 oranlarında bir yükseliş olmuştur. Uygulama sonucunda, veri setine sadece öznitelik seçimi ön işlemi uygulandığında kullanılan tüm algoritmaların saldırı tespit hızında kayda değer bir artış görülürken, F-ölçütünde %0,5 ile %1,2 arasında küçük oranlarda artış ve azalış görülmüştür. Diğer metriklerin değerinde ise genel olarak küçük oranlarda artış gözlemlenmiştir.

3.2.5. 5.Senaryo: Makine Öğrenimi Algoritmalarının Tüm Ön İşlemlerden Geçmiş Veri Seti Üzerinde Uygulanması

Beşinci ve son senaryo uygulanırken, kategorik kodlama ve ölçeklendirme ön işlemlerinin uygulandığı veri setinden seçilen öznitelikler kullanılarak, kategorik veri kodlama, ölçeklendirme, öznitelik seçimi olmak üzere, üç ön işlemin birlikte uygulandığı veri setleri ve KNN, MLP, RF, XGBoost, LightGBM algoritmaları ile saldırı tespit modelleri oluşturulmuştur. Oluşturulan modeller test veri seti kullanılarak eğitim süresi, test süresi, doğruluk, kesinlik, duyarlılık, F-ölçütü, MMC, ROC, PRC metrikleriyle analiz edilip değerlendirilmiştir.

Alınan sonuçlar KNN kullanan modeller için Çizelge 3.19'da, MLP kullanan modeller için Çizelge 3.20'de, RF kullanan modeller için Çizelge 3.21'de, XGBoost kullanan modeller için Çizelge 3.22'de, LightGBM kullanan modeller için Çizelge 3.23'te gösterilmiştir.

Çizelge 3.19. Tüm ön işlemlerden geçmiş veri setleri ile KNN analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F - ölçütü	MMC	ROC alanı	PRC alanı
f	0,012	51,468	80,5%	96,5%	68,2%	79,9%	65,6%	86,0%	92,8%
s	18,547	14,118	78,1%	95,5%	64,5%	77,0%	61,6%	84,7%	91,9%
g	0,009	49,489	79,4%	94,2%	67,9%	78,9%	62,8%	83,8%	91,5%
fns	16,202	10,577	78,0%	95,5%	64,5%	77,0%	61,5%	84,7%	91,9%
fng	0,009	49,490	79,4%	94,2%	67,9%	78,9%	62,8%	83,8%	91,5%
gns	8,332	5,133	78,9%	95,8%	65,9%	78,1%	62,9%	86,0%	92,6%
fus	0,009	50,173	80,5%	96,6%	68,2%	79,9%	65,6%	86,4%	93,0%
fug	0,009	49,899	80,5%	96,6%	68,2%	79,9%	65,6%	86,4%	93,0%
gus	0,006	48,971	79,2%	94,2%	67,6%	78,7%	62,5%	83,1%	91,2%
fnsng	8,206	5,193	78,9%	95,8%	65,9%	78,1%	62,9%	86,0%	92,6%
fusug	0,013	49,485	72,6%	96,8%	53,8%	69,1%	54,7%	80,4%	90,1%

Çizelge 3.20. Tüm ön işlemlerden geçmiş veri setleri ile MLP analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC alanı
f	49,845	0,031	81,5%	92,8%	73,2%	81,8%	65,5%	94,5%	95,4%
s	59,792	0,022	76,5%	95,4%	61,8%	75,0%	59,4%	89,7%	91,4%
g	65,051	0,037	82,1%	92,7%	74,4%	82,4%	66,4%	92,8%	94,4%
fns	56,016	0,019	76,2%	95,6%	60,9%	74,4%	58,9%	89,7%	91,5%
fng	63,309	0,028	80,0%	91,7%	71,3%	80,2%	62,6%	90,2%	92,7%
gns	64,377	0,025	75,7%	96,5%	59,5%	73,6%	58,7%	92,6%	94,3%
fus	53,053	0,034	81,8%	92,9%	73,6%	82,1%	65,8%	94,5%	95,3%
fug	52,371	0,025	81,2%	93,2%	72,3%	81,4%	65,1%	93,8%	95,0%
gus	57,500	0,031	79,4%	91,6%	70,2%	79,5%	61,7%	91,4%	92,9%
fnsng	83,470	0,019	75,4%	96,7%	58,8%	73,1%	58,4%	92,3%	94,2%
fusug	50,199	0,025	81,6%	92,2%	74,0%	82,1%	65,4%	93,4%	94,8%

Çizelge 3.21. Tüm ön işlemlerden geçmiş veri setleri ile RF analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC alanı
f	6,207	0,136	79,4%	98,7%	64,7%	78,1%	65,0%	92,1%	94,6%
s	5,593	0,219	76,3%	96,6%	60,4%	74,4%	59,6%	90,0%	92,9%
g	5,111	0,130	99,5%	99,1%	100,0%	99,6%	99,0%	99,7%	99,6%
fns	5,779	0,202	75,9%	96,5%	59,8%	73,8%	59,0%	89,6%	93,0%
fng	5,100	0,123	99,5%	99,2%	100,0%	99,6%	99,1%	99,7%	99,6%
gns	4,832	0,206	74,0%	96,8%	56,2%	71,1%	56,5%	91,6%	94,4%
fus	5,822	0,136	99,3%	98,9%	100,0%	99,4%	98,7%	99,7%	99,6%
fug	6,180	0,142	99,5%	99,1%	100,0%	99,5%	98,9%	99,7%	99,7%
gus	5,440	0,134	99,6%	99,2%	100,0%	99,6%	99,1%	99,7%	99,7%
fnsng	4,729	0,200	74,0%	96,7%	56,2%	71,1%	56,5%	92,0%	94,5%
fusug	5,751	0,139	99,4%	99,0%	100,0%	99,5%	98,8%	99,7%	99,6%

Çizelge 3.22. Tüm ön işlemlerden geçmiş veri setleri ile XGBoost analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC alanı
f	2,168	0,009	78,8%	96,4%	65,2%	77,8%	63,1%	94,4%	95,9%
s	2,218	0,015	77,2%	96,8%	62,0%	75,6%	61,0%	94,4%	95,8%
g	1.432	0,008	97,9%	96,5%	100,0%	98,2%	95,8%	99,5%	99,4%
fns	2,097	0,012	76,9%	96,8%	61,5%	75,2%	60,6%	94,6%	95,9%
fng	1,414	0,007	97,9%	96,5%	100,0%	98,2%	95,8%	99,5%	99,4%
gns	1,747	0,012	77,8%	96,8%	63,0%	76,3%	61,8%	94,6%	95,9%
fus	2,150	0,009	97,9%	96,5%	100,0%	98,2%	95,8%	99,4%	99,4%
fug	1,903	0,009	97,9%	96,5%	100,0%	98,2%	95,8%	99,4%	99,4%
gus	1,732	0,008	97,9%	96,5%	100,0%	98,2%	95,8%	99,6%	99,6%
fnsng	1,732	0,012	77,8%	96,8%	63,0%	76,3%	61,8%	94,6%	95,9%
fusug	2,136	0,008	97,9%	96,5%	100,0%	98,2%	95,8%	99,4%	99,4%

Çizelge 3.23. Tüm ön işlemlerden geçmiş veri setleri ile LightGBM analiz sonuçları

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin-Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC alanı
f	0,515	0,027	79,3%	97,9%	65,1%	78,2%	64,6%	81,4%	91,1%
s	0,305	0,037	77,9%	96,9%	63,1%	76,5%	62,0%	95,3%	96,3%
g	0,373	0,019	95,6%	96,7%	95,5%	96,1%	91,1%	97,8%	98,3%
fns	0,317	0,033	77,5%	96,9%	62,4%	75,9%	61,4%	95,2%	96,3%
fng	0,383	0,021	95,6%	96,7%	95,5%	96,1%	91,1%	97,8%	98,3%
gns	0,262	0,029	78,2%	96,9%	63,7%	76,9%	62,5%	95,3%	96,4%
fus	0,496	0,022	95,6%	96,7%	95,5%	96,1%	91,1%	97,8%	98,3%
fug	0,425	0,021	95,6%	96,7%	95,5%	96,1%	91,1%	97,8%	98,3%
gus	0,390	0,022	95,6%	96,7%	95,5%	96,1%	91,1%	97,8%	98,3%
fnsng	0,276	0,031	78,2%	96,9%	63,7%	76,9%	62,5%	95,3%	96,4%
fusug	0,467	0,021	95,6%	96,7%	95,5%	96,1%	91,1%	97,8%	98,3%

Tüm ön işlemlerden geçmiş veri setleri ile en başarılı sonuçlara ulaşılan, öznitelik alt kümesi ve makine öğrenimi algoritması birlikteliği Çizelge 3.24'te gösterilmiştir.

Çizelge 3.24. Tüm ön işlemlerden geçmiş veri seti ile en başarılı öznitelik alt kümesi ve algoritma birleşimi

Küme	Eğitim süresi (s)	Test süresi (s)	Doğru-luk	Kesin Lik	Duyar-lılık	F-ölçütü	MMC	ROC alanı	PRC alanı
fug + KNN	0,009	49,899	80,5%	96,6%	68,2%	79,9%	65,6%	86,4%	93,0%
fusug + MLP	50,199	0,025	81,6%	92,2%	74,0%	82,1%	65,4%	93,4%	94,8%
fng + RF	5,100	0,123	99,5%	99,2%	100,0%	99,6%	99,1%	99,7%	99,6%
fng + XGBoost	1,414	0,007	97,9%	96,5%	100,0%	98,2%	95,8%	99,5%	99,4%
g + LightGBM	0,373	0,019	95,6%	96,7%	95,5%	96,1%	91,1%	97,8%	98,3%

Çizelge 3.24'teki sonuçlara göre KNN algoritmasının en başarılı modeli kurduğu veri seti, filtreleme yöntemlerinden olan MI ve gömülü yöntemlerden olan XGBoost öznitelik seçim yöntemlerinin seçtiği özniteliklerin birleşiminden oluşan özniteliklere sahip veri setidir. En başarılı modeli MLP algoritması; MI, GA ve XGBoost'un seçtikleri özniteliklerin birleşiminden oluşan özniteliklere sahip veri seti ile RF algoritması; MI ve XGBoost'un seçtikleri özniteliklerin kesişimine, XGBoost

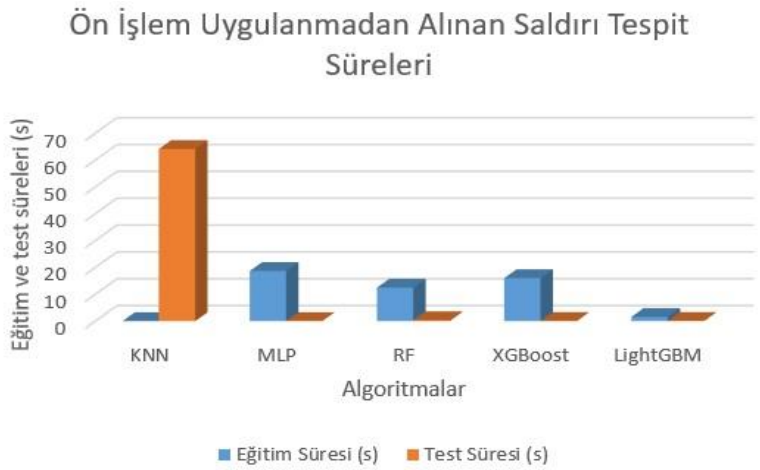
algoritması; MI ve XGBoost'un seçtikleri özniteliklerin kesişimine, LightGBM algoritması ise en başarılı modeli; XGBoost'un seçtiği özniteliklere sahip veri seti ile kurmuştur. Çizelge 3.24'teki sonuçlar, ön işlem uygulanmamış veri setinin kullanıldığı Çizelge 3.10'daki sonuçlarla karşılaştırılarak analiz edildiğinde, eğitim veri seti üzerindeki saldırı tespit hızı yaklaşık olarak KNN'de %71, RF'de %59, XGBoost'da %91, LightGBM'de %76 oranlarında artmış fakat MLP'de %169 oranında azalmıştır. Test veri seti üzerindeki saldırı tespit hızı ise yaklaşık olarak, KNN'de %22, MLP'de %19, RF'de %63, XGBoost'ta %85, LightGBM'de %79 oranlarında artmıştır. F-ölçütüne baktığımızda, KNN'de %4.2, MLP'de %5.5, RF'de %24.1, XGBoost'ta %19.7, LightGBM'de %17.6 oranlarında bir artış olmuştur. Uygulama sonucunda, veri setine kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemlerinin tümünün, birlikte uygulandığı veri setleri ile oluşturulan modellerde, MLP ile kurulan model dışında, saldırı tespit hızlarında %19 ile %91 arasında değişen tatmin edici bir artış görülürken, F-ölçütü'nde %4.2 ile %24.1 arasında değişen kayda değer oranlarda bir artış görülmüştür. Diğer metriklerin değerinde ise genel olarak artış gözlemlenmiştir. Yapılan tüm uygulamalar sonucunda, tüm ön işlemlerin birlikte uygulandığı veri setleriyle oluşturulan saldırı tespit modellerinin, saldırı tespit başarısı ve hızı bakımından, ön işlemlerin ayrı ayrı uygulandığı veri setleriyle oluşturulan saldırı tespit modellerinden açıkça daha başarılı oldukları görülmüştür.

Bu bölümde önerilen metodolojinin uygulaması beş farklı senaryo kullanılarak gerçekleştirilmiştir. Birinci senaryoda veri setine ön işlem uygulanmadan, ikinci senaryoda kategorik veri kodlama ön işlemi, üçüncü senaryoda veri ölçeklendirme ön işlemi, dördüncü senaryoda öznitelik seçimi ön işlemi, beşinci senaryoda tüm ön işlemler uygulanarak elde edilen veri setleri ve makine öğrenimi algoritmaları ile saldırı tespit modelleri oluşturulmuş, oluşturulan modeller saldırı tespit başarısı ve hızı bakımından analiz edilmiştir. En başarılı sonuçlara tüm ön işlemlerin uygulandığı veri setleri kullanılarak oluşturulan modeller ile beşinci senaryonun uygulanması sonucu ulaşılmıştır.

4. TARTIŞMA

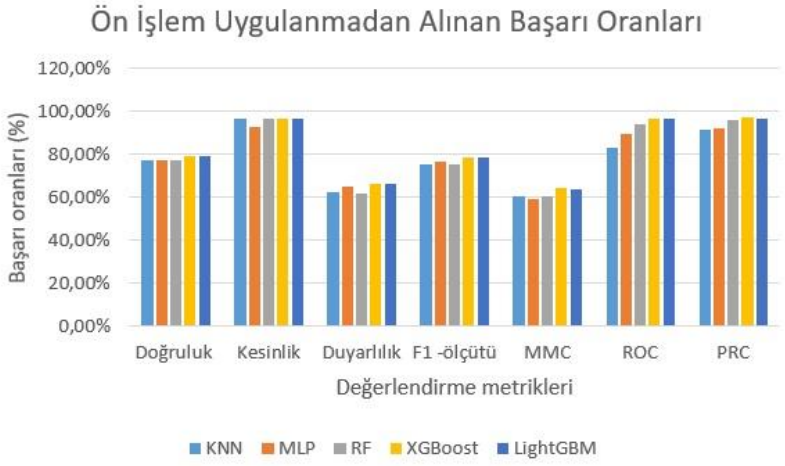
Yapılan çalışmada beş farklı senaryo, beş farklı makine öğrenimi algoritması üzerinde uygulanmış, her senaryo uygulaması sonucunda saldırı tespit modelleri oluşturulmuş ve test veri seti kullanılarak modellerin başarı durumları, doğruluk, kesinlik, duyarlılık, F-ölçütü, MMC, ROC ve PRC metrikleriyle analiz edilmiştir. Yapılan analizlerde, modellerin saldırı tespit başarısını değerlendirmek için, çalışmada kullanılan NSL-KDD gibi sınıf değişkeni dengesiz dağılan veri setlerinde doğruluk metriğinden daha sağlıklı fikir verdiği ve FP ile FN tahmininin maliyetinin yüksek olduğu kritik durumlar göz önünde bulundurulduğu için F-ölçütü dikkate alınmıştır.

Birinci senaryoda veri setlerine ön işlem uygulanmamıştır. Senaryonun uygulama sonuçları değerlendirildiğinde; makine öğrenimi algoritmalarının, veri setlerine ön işlem uygulanmadığında saldırı tespit oranlarının ve saldırı tespit hızlarının düşük olduğu gözlemlenmiştir. En hızlı tespit oranı, eğitim veri seti üzerinde 0,031 s ile KNN algoritması, test veri seti üzerinde ise 0,031 s ile MLP algoritması kullanılarak elde edilmiştir. En düşük saldırı tespit hızı ise, eğitim veri seti üzerinde 18,675 s ile MLP, test veri seti üzerinde 64,146 s ile KNN algoritması kullanılarak elde edilmiştir (Şekil 4.1).



Şekil 4.1. Ön işlem yapılmadan alınan saldırı tespit süreleri.

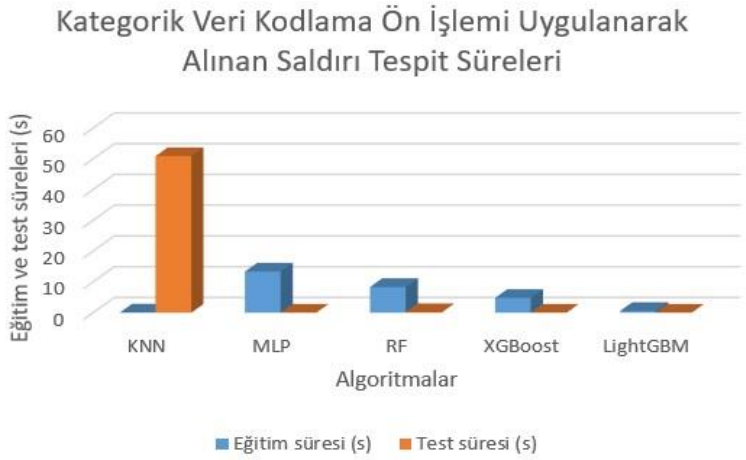
En başarılı saldırı tespit başarısı, %78,5 F-ölçütü ile XGBoost ve LightGBM algoritmaları kullanılarak elde edilmiştir. En düşük tespit başarısı ise %75,5 F-ölçütü ile RF algoritması ile kurulan modele aittir (Şekil 4.2). İlk senaryo sonunda, makine öğrenimi algoritmalarının, ön işlem uygulanmamış veri setleri üzerinde uygulandığında başarısız olduğu gözlemlenmiştir.



Şekil 4.2. Ön işlem yapılmadan alınan başarı oranları.

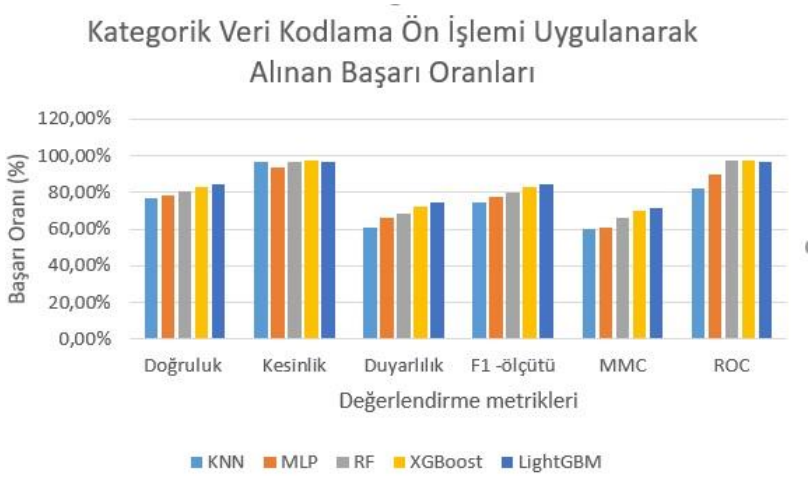
İkinci senaryoda veri setlerine kategorik değere sahip veriler için kategorik veri kodlama tekniklerinden olan Leave One Out Encoding ön işlemi uygulanmıştır.

Senaryonun uygulama sonuçları değerlendirildiğinde: Ön işlem olarak, veri setine uygun olarak seçilen leave one out encoding kategorik veri kodlama tekniği uygulandığında, ön işlem uygulanmayan veri setleriyle oluşturulan modellere kıyasla saldırı tespit hızlarında %10 ile %70 arasında bir artış, F-ölçütü'nde ise %0,9 ile %5,9 arasında bir artış gözlemlenmiştir (Şekil 4.3). Eğitim veri seti ve test veri seti üzerinde en yüksek saldırı tespit hızı artışı %70 ve %62 ile XGBoost ile kurulan modelde görülmüştür.



Şekil 4.3. Kategorik kodlama ön işlemi uygulanarak alınan saldırı tespit süreleri.

En yüksek saldırı tespit başarısı artışı ise %5,9 ile LightGBM ile kurulan modelde görülmüştür. KNN algoritmasında ise eğitim veri seti üzerinde saldırı tespit hızı yaklaşık %65 oranında, saldırı tespit başarısı ise %0,9 oranında düşmüştür (Şekil 4.4). İkinci senaryo sonunda, veri setine kategorik veri kodlama ön işlemi uygulandığında makine öğrenimi algoritmalarının saldırı tespit başarısı (F-ölçütü) ve hızının genel olarak arttığı gözlemlenmiştir.



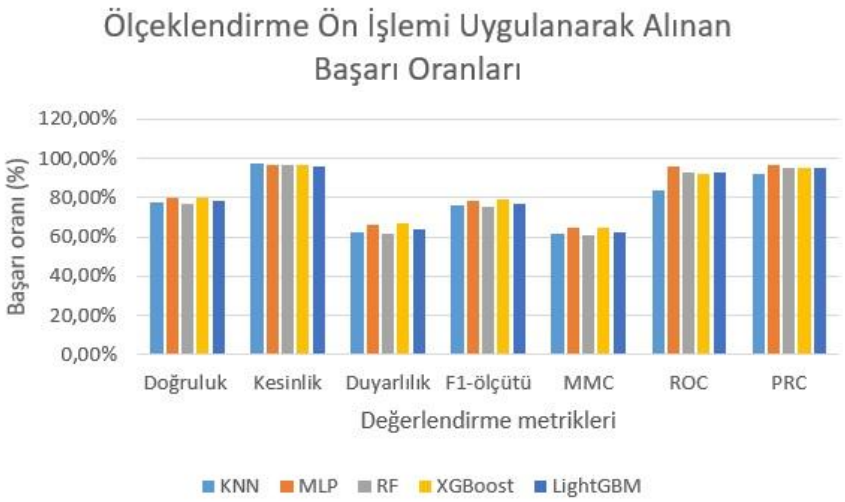
Şekil 4.4. Kategorik kodlama ön işlemi uygulanarak alınan başarı oranları.

Üçüncü senaryoda veri setlerine min-max ölçeklendirme ön işlemi uygulanmıştır. Senaryonun uygulama sonuçları değerlendirildiğinde: Ön işlem uygulanmayan veri setleriyle oluşturulan modellere kıyasla eğitim veri setleri üzerindeki saldırı tespit hızında KNN ve LightGBM algoritmaları ile %48 ve %16 oranlarında artış, RF ve

XGBoost ile %4 ve %1 oranlarında düşüş, MLP ile ise yaklaşık altı kat düşüş gözlemlenmiştir. Test veri setindeki saldırı tespit hızında ise, RF, XGBoost ve LightGBM ile, %3, %13 ve %14 oranlarında artış, KNN ve MLP algoritmaları ile %4 ve %123 oranlarında bir düşüş gözlemlenmiştir (Şekil 4.5).

Şekil 4.5. Ölçeklendirme ön işlemi uygulanarak alınan saldırı tespit süreleri.

Saldırı tespit başarısı için F-ölçütü'ne bakıldığında, KNN, MLP ve XGBoost ile %0,3, %2 ve %0,4 oranlarında bir artış, RF ve LightGBM ile ise, %0,3 ve %1,5 oranlarında bir düşüş gözlemlenmiştir. Eğitim veri setinde saldırı tespit hızındaki en büyük artış KNN'de görülürken, test veri seti üzerinde en büyük artış LightGBM ile gözlemlenmiştir; fakat saldırı tespit hızı MLP ile kurulan modellerde ciddi oranda düşmüştür. F-ölçütü'nde ise %0,3-%2 arasında küçük oranlarda düşüş ve artışlar gözlemlenmiştir (Şekil 4.6). Üçüncü senaryo sonunda, veri setine ölçeklendirme ön işlemi uygulandığında, saldırı tespit hızının artması beklenirken özellikle MLP algoritmasında düşüş görülmüştür. Bu düşüşün nedeninin uygun ölçeklendirme yönteminin seçilmemiş olması olabileceği ile birlikte sorunun çözümü, kullanılan veri setine ve algoritmaya uygun ölçeklendirme yönteminin ve de oluşturulan modellerde hiper parametrelerin belirlenerek kullanılması ile sağlanabilir.

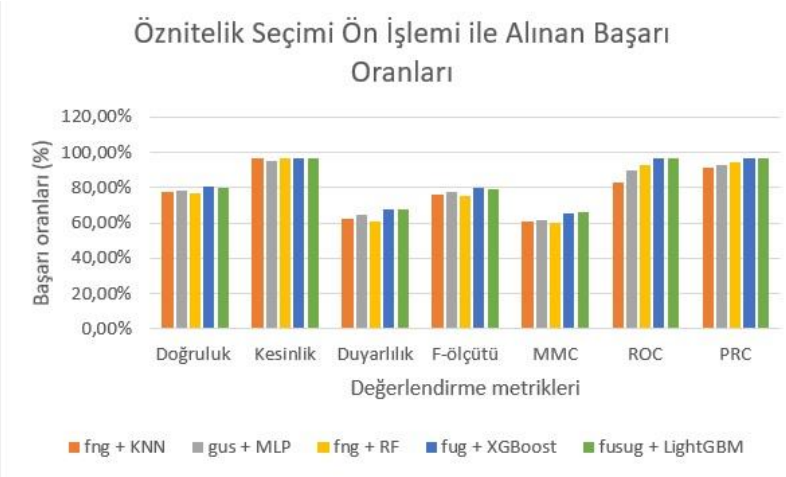


Şekil 4.6. Ölçeklendirme ön işlemi uygulanarak alınan başarı oranları.

Dördüncü senaryoda veri setlerine hibrit öznitelik seçimi ön işlemi uygulanmıştır. Senaryonun uygulama sonuçları değerlendirildiğinde: Algoritmaların en başarılı modelleri kurdukları veri setlerinin hibrit öznitelik seçimi yönteminin uygulanması ile elde edilen, KNN için fng, MLP için gus, RF için fng, XGBoost için fug, LightGBM için ise fusug öznitelik alt kümelerine sahip veri setleri olduğu görülmüştür. Ön işlem uygulanmayan veri setleri ile oluşturulan modellere kıyasla eğitim veri setleri üzerindeki saldırı tespit hızında KNN, MLP, RF, XGBoost ve LightGBM algoritmalarında sırasıyla %45, %40, %92, %74 ve %65 oranlarında artış, test veri setindeki saldırı tespit hızında ise, %20, %29, %46, %62 ve %58 oranlarında artış gözlemlenmiştir (Şekil 4.7).

Şekil 4.7. Öznitelik seçimi ön işlemi uygulanarak alınan saldırı tespit süreleri.

Saldırı tespit başarısı için F-ölçütü'ne bakıldığında MLP, XGBoost ve LightGBM algoritmalarında sırasıyla %0,6, %1,2 ve %0,8 oranlarında bir artış olduğu, RF'de ise %0,5 oranında bir düşüş olduğu, KNN'de bir değişiklik olmadığı gözlemlenmiştir (Şekil 4.8).

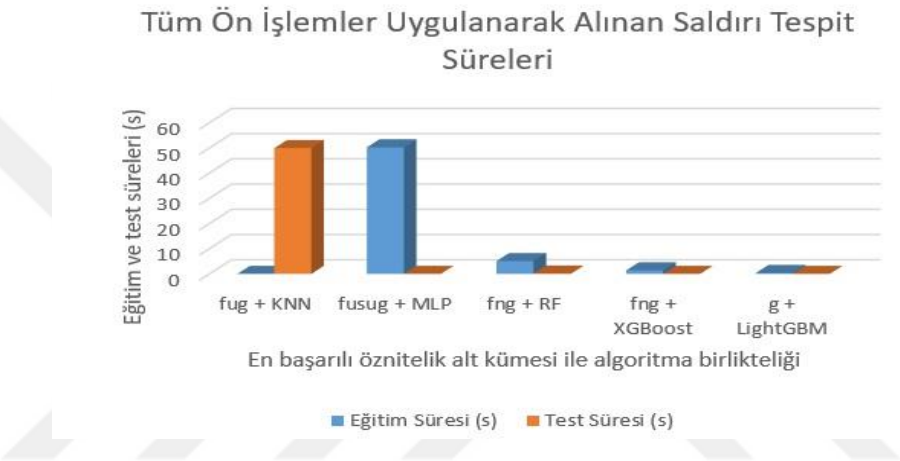


Şekil 4.8. Öznitelik seçimi ön işlemi uygulanarak alınan başarı oranları.

Dördüncü senaryo sonunda, hibrit öznitelik seçimi yönteminin klasik öznitelik seçimi yöntemlerine göre daha başarılı olduğu gözlemlenmiştir. Veri setine öznitelik seçimi ön işlemi uygulandığında öznitelik sayısı azaldığı için tüm algoritmalar ile saldırı tespit hızı artmıştır; fakat öznitelik seçiminin veri setini en iyi temsil eden öznitelikleri seçmesi hedeflendiği için, tespit hızları ile birlikte tespit başarısının da arttırması beklenirken, bu artış çok küçük oranlarda olmuştur. RF ile ise tespit başarısında küçük oranda bir düşüş görülmüştür. Bu problemin çözümü için, öznitelik seçiminin kategorik veri kodlama ve ölçeklendirme ön işlemlerinden geçmiş veri seti ile yapılması öngörülmüş ve beşinci senaryo uygulanmıştır. Fakat farklı öznitelik seçim yöntemleri ve eşik değerleri denenerek de saldırı tespit hızı ve başarısını arttırmak mümkündür.

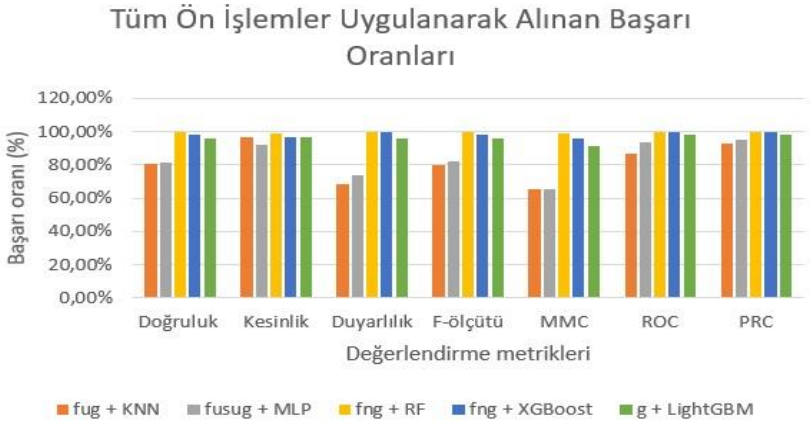
Beşinci ve son senaryonun uygulanması için kategorik veri kodlama ve ölçeklendirme ön işlemlerinden geçmiş veri setinden seçilen özniteliklere sahip, kısaca tüm ön işlemlerin birlikte uygulandığı veri seti kullanılmıştır. Algoritmaların en başarılı modelleri kurdukları veri setlerinin hibrit öznitelik seçimi yönteminin

uygulanması ile elde edilen, KNN için fug, MLP için fusug, RF için fng, XGBoost için fng, LightGBM için ise g öznitelik alt kümelerine sahip veri setleri olduğu görülmüştür. Ön işlem uygulanmayan veri setleriyle oluşturulan modellere kıyasla eğitim veri setleri üzerindeki saldırı tespit hızında KNN, MLP, RF, XGBoost ve LightGBM algoritmaları ile sırasıyla %71, %59, %91 ve %76 oranlarında artış, MLP ile %169 oranında bir düşüş gözlemlenmiştir. Test veri setindeki saldırı tespit hızında ise, sırasıyla %22, %19, %63, %85 ve %79 oranlarında artış gözlemlenmiştir (Şekil 4.9).



Şekil 4.9. Tüm ön işlemler uygulanarak alınan saldırı tespit süreleri.

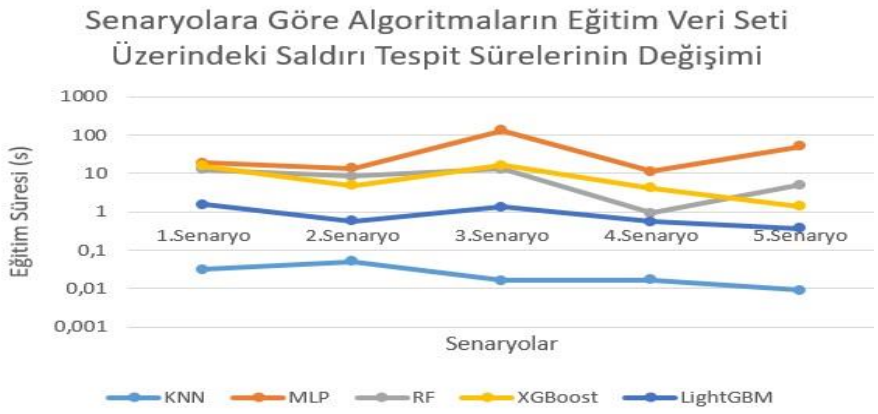
Saldırı tespit başarısı için F-ölçütüne bakıldığında KNN, MLP, RF, XGBoost ve LightGBM algoritmaları ile sırasıyla %4,2, %5,5 ve %24,1 %19,7 ve %17,6 oranlarında bir artış gözlemlenmiştir (Şekil 4.10).



Şekil 4.10. Tüm ön işlemler uygulanarak alınan başarı oranları.

Beşinci senaryo sonunda, literatürdeki araştırmalara benzer bir sonuca ulaşılmıştır (Torabi ve ark., 2021; Kamarudin ve Watson, 2019; Mohammad, 2021; Ambusaidi ve ark., 2014; Chih Wen ve ark., 2020). Hibrit öznitelik seçimi yönteminin klasik öznitelik seçim yöntemlerine göre daha başarılı olduğu gözlemlenmiştir. Dördüncü senaryonun uygulamasında tatmin edici ölçüde doğrulanamayan, öznitelik seçimi ön işlemi, saldırı tespit hızı ve tespit başarısını arttırır hipotezi beşinci senaryonun uygulanması sonunda doğrulanmış ve saldırı tespit hızının MLP dışında tüm algoritmalar ile dikkat çekecek ölçüde arttığı, saldırı tespit başarısının (F-ölçütü) ise ön işlem uygulanmayan veri seti ve ön işlemlerin ayrı ayrı uygulandığı veri setleri ile kurulan modellere kıyasla tüm algoritmalarda kayda değer oranlarda arttığı gözlemlenmiştir. MLP ile kurulan modellerde tüm ön işlemlerin uygulandığı veri setinin kullanıldığı 5. senaryo ile sadece öznitelik seçimi ön işleminin uygulandığı 4.senaryodan özellikle saldırı tespit hızı için daha kötü sonuçlar alınmasının sebebi 5.senaryoda veri setine ölçeklendirme ön işleminin de uygulanmasıdır ve MLP için farklı bir ölçeklendirme yöntemi kullanılması bu sorunun çözümü olabilir.

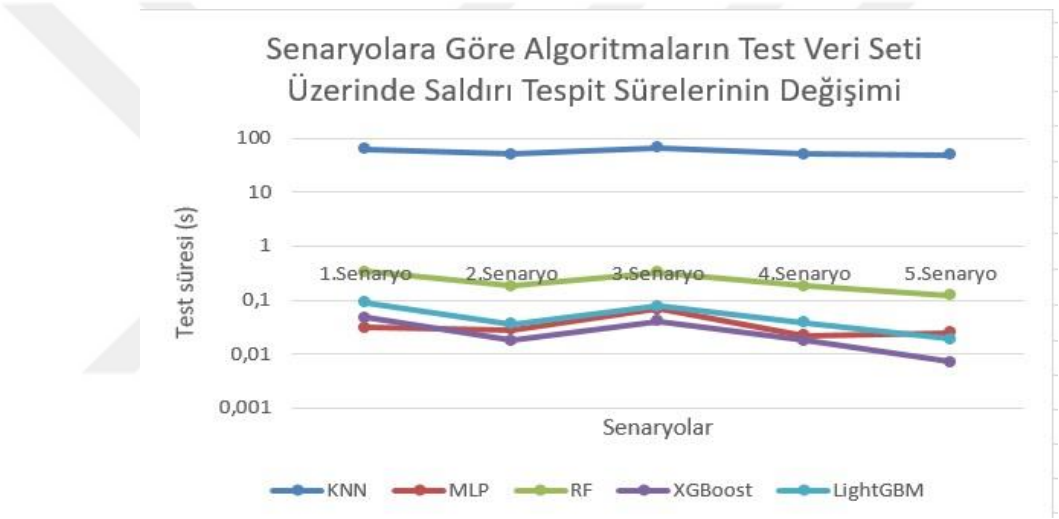
Şekil 4.11'de görüldüğü gibi, eğitim seti üzerindeki saldırı tespit süreleri, senaryoların uygulanması süresince, MLP algoritması ve ölçeklendirme ön işleminin uygulandığı 3.senaryo dışında azalan bir değişim göstermiştir. Son senaryoda, ilk senaryoya göre saldırı tespit süresinin kayda değer ölçüde azaldığı, saldırı tespit hızının arttığı görülmektedir (Çizelge 4.1).



Şekil 4.11. Eğitim veri seti üzerinde saldırı tespit sürelerinin senaryolara göre değişimi

İlk senaryonun ve son senaryonun uygulanması sonunda alınan sonuçlara göre eğitim veri seti üzerindeki saldırı tespit sürelerindeki değişim karşılaştırıldığında en yüksek azalış, dolayısıyla en büyük hız artışı sırasıyla XGBoost:14,596 s; RF:7,431 s; LightGBM:1,209 s; KNN:0,022 s şeklinde sıralanırken, MLP'de tespit süresi 31,524 s artmış, tespit hızı düşmüştür (Şekil 4.11).

Eğitim veri seti üzerinde en yüksek saldırı tespit hızına 0,009 s ile KNN algoritmasının fug öznitelik alt kümesini kullanarak kurduğu model ile ulaşılırken, test veri seti üzerindeki saldırı tespit hızı 49,899 s ile en yavaş model, yine KNN'nin fug öznitelik alt kümesi ile kurduğu model olmuştur (Şekil 4.12).



Şekil 4.12. Test veri seti üzerinde saldırı tespit sürelerinin senaryolara göre değişimi.

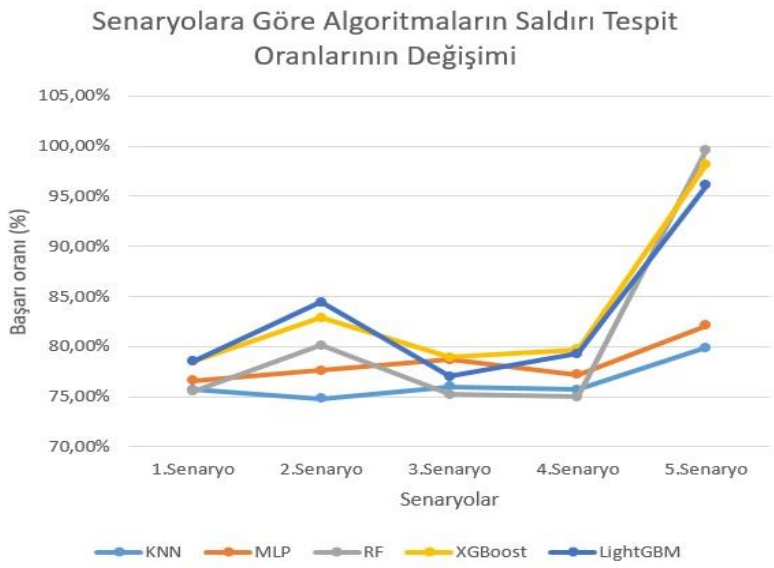
Çizelge 4.1. Ön işlem adımlarının saldırı tespit süreleri üzerindeki etkisi

	Ön işlem uygulanmadan		**Kategorik veri kodlama uygulanarak**		**Ölçeklendirme uygulanarak**		**Öznitelik seçimi uygulanarak**		**Tüm ön işlemler uygulanarak**	
Algorit-Malar	Eğitim süresi (s)	Test Süresi (s)	Eğitim süresi (s)	Test Süresi (s)	Eğitim süresi (s)	Test Süresi (s)	Eğitim süresi (s)	Test Süresi (s)	Eğitim süresi (s)	Test Süresi (s)
KNN	0,031	64,146	0,051	50,899	0,016	66,670	0,017	51,320	0,009	49,899
MLP	18,675	0,031	13,377	0,028	133,871	0,069	11,200	0,022	50,199	0,025
RF	12,531	0,334	8,349	0,182	13,004	0,325	0,937	0,181	5,100	0,123
XGBoost	16,010	0,047	4,816	0,018	16,194	0,041	4,158	0,018	1,414	0,007
Light-GBM	1,582	0,091	0,573	0,036	1,326	0,078	0,554	0,038	0,373	0,019

İlk senaryonun ve son senaryonun uygulanması sonunda alınan sonuçlara göre test veri seti üzerindeki saldırı tespit sürelerindeki değişim karşılaştırıldığında en yüksek

azalış, dolayısıyla en büyük hız artışı sırasıyla KNN:14,247 s; RF:0,211 s; LightGBM:0,072 s; XGBoost:0,040 s; MLP:0,006 s şeklinde sıralanırken, en yüksek hıza 0,007 s ile XGBoost algoritmasının fng öznitelik alt kümesini kullanarak kurduğu model ulaşmıştır (Şekil 4.12).

Şekil 4.13 ve Çizelge 4.2'de görüldüğü gibi algoritmaların saldırı tespit başarılarında senaryoların uygulanması süresince artan bir değişim olmuştur. En büyük oranda artış, veri setine tüm ön işlemlerin uygulandığı 5.senaryoda gerçekleşmiştir. İlk senaryonun ve son senaryonun uygulanması sonunda alınan sonuçlara göre, saldırı tespit başarıları için baz aldığımız F-ölçütü'ndeki değişim karşılaştırıldığında en yüksek artış sırasıyla, RF:%24,1; XGboost:%19,7; LightGBM:%17,6; MLP: %5,5; KNN:%4,2 şeklinde sıralanabilir. Son senaryo sonunda en yüsek saldırı tespit değerleri, fng öznitelik alt kümesi ve RF algoritması ile oluşturulan modelde %96,6, fng öznitelik alt kümesi ve XGBoost algoritması ile oluşturulan modelde %98,2, g öznitelik alt kümesi ve LightGBM algoritması ile oluşturulan modelde %96,1'dir.



Şekil 4.13. Senaryolara göre saldırı tespit başarılarının (F-ölçütü) değişimi.

Çizelge 4.2. Ön işlem adımlarının F-ölçütü üzerindeki etkisi

	Ön işlem uygulanmadan (1.Senaryo)	**Kategorik veri kodlama uygulanarak (2.Senaryo)**	**Ölçeklendirme uygulanarak (3.Senaryo)**	**Öznitelik seçimi uygulanarak (4.Senaryo)**	**Tüm ön işlemler uygulanarak (5.Senaryo)**
Algorit-malar	F-ölçütü	F-ölçütü	F-ölçütü	F-ölçütü	F-ölçütü
KNN	75,7%	74,8%	76,0%	75,7%	79,9%
MLP	76,6%	77,6%	78,7%	77,2%	82,1%
RF	75,5%	80,1%	75,2%	75,0%	99,6%
XGBoost	78,5%	82,9%	78,9%	79,7%	98,2%
Light-GBM	78,5%	84,4%	77,0%	79,3%	96,1%

Bu bölümde, yapılan çalışma ile ulaşılan sonuçların, konu ile ilgili literatür çalışmalarındaki sonuçlarla benzer olup olmadığı ve çalışmanın amacına ne denli ulaşıldığı tartışılmıştır.

5. SONUÇ VE ÖNERİLER

Bu çalışmada, literatürdeki STS çalışmalarının birçoğunda kullanılan NSL-KDD veri setine uygulanan ön işlemlerin, makine öğrenimi yöntemiyle geliştirilen saldırı tespit modellerinin performansı üzerindeki etkisi incelenerek, yüksek oranda ve hızda saldırı tespit yapabilen saldırı tespit modeli geliştirilmesi hedeflenmiştir. Çalışmanın başında literatür taraması yapılmış ve tarama sonucunda yapılan çalışmaların çoğunlukla veri setine uygulanan tek bir ön işleme odaklandığı, birden fazla ön işlemi uygulayan çalışmaların ise, kullanılan veri setine ve makine öğrenimi algoritmasına en uygun ön işlemleri bulabilmek için çok sayıda seçeneği denemek yerine, literatürde sık kullanılan ön işlemleri kullandığı, öznitelik seçimi ön işleminde ise filtreleme ya da sarmal yöntemlerin kullanıldığı görülmüştür. Son yıllarda yapılan siber saldırıların çeşidi ve sayısı arttığı, kullandıkları teknoloji sayesinde hedefledikleri sisteme kolayca entegre oldukları için, imza tabanlı STS'ler tarafından tespit edilmeleri zorlaşmıştır. Bu nedenle saldırı tespit modellerinin çeşidi ve sayısı artan, daha önce karşılaşmadıkları saldırıları, yüksek oranda ve hızda tespit edebilmesi, siber saldırıların verdikleri zararların büyüklüğü düşünülünce bu saldırıların yüksek oranda ve hızda tespit edilebilmesi için yapılan anomali tabanlı STS çalışmaları son derece önemlidir. Tüm bu sebeplerden dolayı, bu çalışma ile birlikte;

- Veri setine uygulanan ön işlemler, farklı ön işlemler uygulanarak elde edilmiş veri setleri ve makine öğrenimi algoritmalarıyla saldırı tespit modellerinin oluşturulması ve oluşturulan modellerin değerlendirilmesi olmak üzere üç aşamadan oluşan bir metodoloji önerilmiş ve uygulanmıştır.

- Metodolojinin ilk aşamasında veri setine ön işlemler uygulanırken, kategorik verileri kodlama ön işleminde, veri setine en uygun yöntemin seçilmesinin oluşturulan modellerin tespit başarısı ve hızındaki olumlu etkisi yapılan testler ile açıkça görülmüştür.

- Ölçeklendirme ön işleminin kullanılan her algoritmanın performansını olumlu etkilemediği, bazı algoritmaların performansını düşürdüğü sonucuna varılmıştır.

- Öznitelik seçimi ön işleminin, ön işlem uygulanmayan veri setine uygulandığında beklenen saldırı tespit başarısı ve hız artışını gerçekleştiremediği, fakat kategorik kodlama ve ölçeklendirme ön işlemlerinden geçmiş veri setine uygulanan öznitelik seçim işleminin seçtiği özniteliklerle hedeflenen tespit başarısı artışına ve tespit hızı artışına daha çok yaklaşıldığı görülmüştür.

- Filtreleme, sarmal ve gömülü öznitelik seçim yöntemlerinin ayrı ayrı kullanılmasına karşın her birinin avantajlarını kullanıp, bir yöntemin eksiklerinin diğer yöntemle telafi edilmesini hedefleyen hibrit öznitelik seçim yönteminin kullanılmasının, tespit başarısı ve hızı bakımından daha başarılı modelleri oluşturduğu gözlemlenmiştir.

- Ön işlemlerin veri setine birlikte uygulanmasının, ayrı ayrı uygulanmasına kıyasla saldırı tespit modellerinde başarıyı daha büyük oranda arttırdığı gözlemlenmiştir.

Çalışma sürecinde gözlemlenenler sonucunda önerilerde bulunmak gerekirse;

- Veri setine uygulanan ön işlemlerin, kullanılacak veri setine ve makine öğrenimi algoritmalarına uygun seçilmesi, saldırı tespit modelinin performansı açısından son derece önemlidir.

- Çevrim içi ortamda değişken veri setleri söz konusu olduğunda, öznitelik seçim algoritmaları yetersiz olacağı için çevrim içi öznitelik seçimi algoritmaları geliştirme çalışmalarının artması önemlidir.

- STS çalışmalarında kullanılan benzetim veri setlerindeki saldırı kayıtlarının oranının çevrim içi trafikte olduğundan daha fazla olması, aşırı öğrenmeyi dolayısıyla yanlış alarm oranını arttırabileceği için, gerçeğe daha yakın veri setleri oluşturulmalıdır.

- Başarılı saldırı tespit modelleri oluşturmak için kapsamlı ve güncel veri setlerine ihtiyaç vardır; fakat veri seti oluşturma süreci zahmetli ve maliyetli olduğu için yeterince kapsamlı ve güncel veri seti yoktur. Bu durum STS çalışmalarını yavaşlatmaktadır. Bu önemli sorunun çözümü için bütçe ve zaman ayrılması son derece önemlidir.

Bu çalışmada birden fazla ön işleme odaklanılarak çalışmanın kapsamı genişletildiği için ön işlemler, kullanılan beş makine öğrenimi algoritması için ortak seçilmiş ve uygulanmıştır. Hesaplama maliyetinin yüksek olmasından dolayı ise algoritmalarda parametre ayarı yapılamamıştır. Veri seti oluşturmak, maliyetli ve zahmetli bir süreç olduğu için benzetim veri seti kullanılmıştır. Gelecekteki çalışmalarda saldırı tespit başarısını ve hızını arttırmak için ön işlemler, kullanılacak her bir algoritmaya ve veri setine uygun seçilebilir, algoritmalarda parametre ayarı yapılabilir, oluşturulan saldırı tespit modellerinin başarısı, saldırı ve normal bağlantı kayıtlarının oranları gerçeğe daha yakın olan güncel ve de farklı veri setleri ile daha güvenilir test edilebilir. Böylece başarısı birçok açıdan kanıtlanmış, güvenilirliği yüksek saldırı tespit modelleri geliştirilebilir.

ÖZET

Veri Setine Uygulanan Ön İşlemlerin Anomali Tabanlı Saldırı Tespit Modellerinin Performansları Üzerindeki Etkisinin İncelenmesi

Bu çalışma, veri setine uygulan ön işlemlerin, saldırı tespit modellerinin performansı üzerindeki etkisini inceleyerek, saldırı tespit başarısı ve saldırı tespit hızı yüksek bir saldırı tespit modelini oluşturmak için bir metodoloji önerilmesi, metodolojisinin uygulanması, alınan sonuçları değerlendirerek, karşılaşılan problemlere çözüm önerisi sunmak amacıyla yapılmıştır.

Çalışmanın ilk aşamasında yapılan literatür taraması sonucunda; anomali tabanlı saldırı tespit modeli geliştirilirken, veri setine uygulanan birden fazla ön işleme aynı anda odaklanan ve veri setine en uygun ön işlemleri seçmek adına çok sayıda yöntemi deneyip test eden yeterince çalışma olmadığı görülmüştür. Bu nedenle kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemlerinin üçünün ayrı ayrı ve birlikte uygulanmasının saldırı tespit modelinin başarısı üzerindeki etkisine odaklanan bir çalışma yapmak istenmiştir.

Önerilen metodoloji kullanılarak ölçeklendirme ön işlemi dışında, her ön işlemin modellerin başarısı üzerinde olumlu bir etki yaptığı; fakat en yüksek saldırı tespit hızına ve başarısına, tüm ön işlemlerin birlikte uygulandığı veri seti ile kurulan modellerle ulaşıldığı görülmüştür. Veri setine uygun kategorik kodlama yönteminin seçilmesinin ve hibrit öznitelik seçim yönteminin model başarısına kayda değer bir katkıda bulunduğu görülmüştür.

Anahtar Sözcükler: Ön İşlemler, Saldırı Tespit Hızı, Saldırı Tespit Modeli, Saldırı Tespit Başarısı, Veri Seti.

SUMMARY

Examination of the Effect of Preprocessing Applied to the Dataset on the Performances of Anomaly Based İntrusion Detection Models

This study examines the effect of pre-processings applied to the data set on the performance of anomaly-based intrusion detection models developed with machine learning method, to suggest a methodology to create an intrusion detection model with high intrusion detection success and intrusion detection speed, to apply the methodology, to evaluate the results and to address the problems encountered made to propose a solution.

As a result of the literatüre review made in the first stage of the study; while developing anomaly-based intrusion detection model, it has been observed that there are not enough studies that focus on more than one preprocessing applied to the data set at the same time and that try and test a vaste number of methods in order to select the most suitable preprocesses for the data set. Therefore, it is asked to conduct a study focusing on the effect of applying categorical data coding, scaling and feature selection preprocesses separately and together on the success of the intrusion detection model.

By using the suggested methodology, it has been observed that except for the scaling pre-process, each pre-process has a positive effect on the success of the models, however the highest attack detection speed and success is achieved with the models established with the data set in which all the pre-processes are applied together. It was seen that choosing the categorical coding method suitable for the data set and the hybrid feature selection method contributed considerably to the model success.

Keywords: Dataset, Intrusion Detection Models, Intrusion Detection Speed, Intrusion Detection Success, Pre-Processes.

KAYNAKLAR

ALTERYX (2019). Alteryx/Categorical_Encoding. Erişim Adresi: [https://github.com/alteryx/categorical_encoding/blob/main/guides/flowchart/Categorical%20Encoding%20Flowchart.png]. Erişim Tarihi: 09/06/2021.

AMBUSAİDİ M A, HE X, TAN Z, NANDA P, LU L F, NAGAR U T (2014). A Novel Feature Selection Approach for Intrusion Detection Data Classification. 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, pp.: 82-89.

ASLAN Ö, SAMET R (2020). A Comprehensive Review on Malware Detection Approaches. IEEE Access, **8** (1-1): 6249-6271.

ANDERSON J. P (1980). "Computer Security Threat Monitoring and Surveillance". Technical Report, James P. Anderson Company, Fort Washington, 1980.

AYDIN M A, ÖRENCİK M B (2005). Bilgisayar Ağlarında Saldırı Tespiti İçin İstatistiksel Yöntem Kullanılması. Yüksek Lisans Tezi, İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2005, İstanbul.

BACE M, MELL P (2001). NIST Special Publication on Intrusion Detection Systems.

BALAKRİSHNAN S M, VENKATALAKSHMİ K, KANNAN A (2014). Intrusion Detection System Using Feature Selection and Classification Technique. *International Journal of Computer Science and Applications* (IJCSA), **3** (4): 145.

BATTİTİ R (1994). Using mutual information for selecting features in supervised neural net learning. *İn IEEE Transactions on Neural Networks*, **5** (4): 537-550, July 1994, doi: 10.1109/72.298224.

BERNASCHİ M, AİUTOLO E D, RUGHETTİ P (1999). Enforcing network security: a real cease study in a research organization. *Computers and Security*, **18** (6): 533–543, January 1999, https://doi.org/10.1016/S0167-4048(99)80117-5.

BEULAH J R, PUNİTHAVATHANİ D S (2015). Simple Hybrid Feature Selection (SHFS) for Enhancing Network Intrusion Detection with NSL-KDD Dataset. *International Journal of Applied Engineering Research* ISSN 0973-4562, **10** (19): 40498-40505.

BOU-HARB E, DEBBABİ M, ASSİ C (2014). "Cyber Scanning: A Comprehensive Survey," in *IEEE Communications Surveys & Tutorials*, **16** (3): 1496-1519, Third Quarter 2014, doi: 10.1109/SURV.2013.102913.00020.

BREİMAN L (1996). Bagging predictors. *Machine Learning*, **24:** 123–140. https://doi.org/10.1007/BF00058655

BREİMAN L (2001). Random Forests. *Machine Learning* **45:** 5–32. https://doi.org/10.1023/A:1010933404324

BUDAK H (2018). Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, **22**:10, DOI:10.19113/sdufbed.01653

CANBEK G, SAĞIROĞLU Ş (2006). Bilgi, Bilgi Güvenliği ve Süreçleri Üzerine Bir İnceleme. *Politeknik Dergisi*, **9**: 165-174.

CHAHAR V, CHHİKARA R, GİGRAS Y, SİNGH L (2016). Significance of Hybrid Feature Selection Technique for Intrusion Detection Systems. *Indian Journal of Science and Technology*, **9** (48): 1-7.

CHEN CW, TSAİ Y H, CHANG F R, LİN W C (2020). Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results. *Expert Systems*, **37** (5): e12553.

CHEN T, GUESTRİN C (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 785–794. DOI:https://doi.org/10.1145/2939672.2939785

CYBERSECURİTY VENİTURES (2021). 2021 REPORT: CYBERWARFARE IN THE C-SUIT. ErişimAdresi: [https://cybersecurityventures.com/wp-content/uploads/2021/01/Cyberwarfare-2021-Report.pdf] Erişim Tarihi: 19/11/2021.

CYBERSECURİTY VENİTURES (2020). Cybercrime To Cost The World $10.5 Trillion Annually By 2025. Erişim Adresi: [https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/] Erişim Tarihi: 3/11/2021.

DAVİS J J, CLARK A J (2011). Data preprocessing for anomaly based network intrusion detection: A review. *Computers & Security*, **30** (6-7): 353-375

DEVİ R R, ABUALKİBASH M(2019). Intrusion Detection System Classification Using Different Machine Learning Algorithms on KDD-99 and NSL-KDD Datasets - A Review Paper. *International Journal of Computer Science and Information Technology*, **11** (3): 65-80, DOI:10.5121/IJCSIT.2019.11306

FAWAGREH K, GABER M M, ELYAN E (2014). Random forests: from early developments to recent advancements. *Systems Science & Control Engineering*, **2** (1): 602-609, DOI: 10.1080/21642583.2014.956265

FREUNDY, SCHAPİRE R E (1997). A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *COLT 1997*. DOI:10.1006/JCSS.1997.1504. Corpus ID: 11742444

FREUNDY, SCHAPİRE R E (1999). "A Short Introduction to Boosting." *IJCAI'99: Proceedings of the 16th international joint conference on Artificial intelligence,* **2**: 1401–1406.

FRİEDMAN J H(2001). "Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, **29**: 1189-1232.

GARTNER (2021). Gartner Forecasts Worldwide Security and Risk Management Spending To Exceed $150 Billion in 2021. Erişim Adresi: [https://www.gartner.com/en/newsroom/press-releases/2021-05-17-gartner-forecasts-worldwide-security-and-risk-managem] Erişim Tarihi: 25.11.2021.

HANCOCK J T, KHOSHGOFTAAR T M (2020). Survey on categorical data for neural networks. *Journal of Big Data*, **7**: 1-41.

HSU H, HSİEH C W, LU M (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, **38** (7): 8144-8150.

JAGANNATH V (2020). Random Forest Template for TIBCO Spotfire®. Erişim Adresi: [https://community.tibco.com/wiki/random-forest-template-tibco-spotfire]. Erişim Tarihi: 17/12/2021.

KAMARUDİN, M H, MAPLE C, WATSON T (2019). Hybrid feature selection technique for intrusion detection system. *International Journal of High Performance Computing and Networking*, **13**: 232-240.

KAYNAR O, ARSLAN H, GÖRMEZ Y, IŞIK Y E (2018). Makine Öğrenmesi ve Öznitelik Seçim Yöntemleriyle Saldırı Tespiti. *Bilişim Teknolojileri Dergisi,* **11** (2): 175-185.

KE G, MENG Q, FİNLEY T, WANG T, CHEN W, MA W, LİU T Y (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, **30**: 3146-3154.

KEARNS M (1988). Thoughts on Hypothesis Boosting, Machine Learning class project, Dec. 1988.

Lİ X, Yİ P, WEİ W, JİANG Y, TİAN L (2021). LNNLS-KH: A Feature Selection Method for Network Intrusion Detection, Security and Communication Networks, **2021** (3): 1-22, Article ID 8830431. https://doi.org/10.1155/2021/8830431

LİU H, ZHOU M, LİU Q (2019). An embedded feature selection method for imbalanced data classification. *İn IEEE/CAA Journal of Automatica Sinica*, **6** (3): 703-715, May 2019, doi: 10.1109/JAS.2019.1911447.

MCCALL J (2005). Genetic algorithms for modelling and optimisation. *Journal of Computational and Applied Mathematics*, **184** (1): 205-222, ISSN 0377-0427. https://doi.org/10.1016/j.cam.2004.07.034.

MACKAY D J C (2003). Information Theory, Inference and Learning Algorithms, 4 nd Ed, Part 2, Chapter 8, pp.: 139.

MAZİNİ M, SHİRAZİ B, MAHDAVİ I (2019). Anomaly network-based intrusion detection system using a reliable hybrid artificial bee colony and AdaBoost algorithms. *Journal of King Saud University - Computer and Information Sciences*, **32** (10): 1206-1207.

MOHAMMAD A H (2021). Intrusion Detection Using a New Hybrid Feature Selection Model. *Intelligent Automation & Soft Computing*, **30** (1): 65–80.

MÜLLER A C, GUİDO S (2016). Introduction to Machine Learning with Python. 1nd Ed, Chapter 5, pp.:260.

MÜLLER A C, GUİDO S (2017). Introduction to Machine Learning with Python. 2nd Ed, Chapter 1, pp.:1.

NASEER S, SALEEM Y (2018). Enhanced Network Intrusion Detection using Deep Convolutional Neural Networks. *KSII Transactions on Internet and Information Systems* **12** (10): 5159-5178.

ÖZGÜR A, ERDEM H (2018). Saldırı Tespit Sistemlerinde Genetik Algoritma Kullanarak Nitelik Seçimi ve Çoklu Sınıflandırıcı Füzyonu. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi*, **33**(1): 75-87, Mart 2018, doi:10.17341/gazimmfd.406781

ÖZKAN-OKAY M, ASLAN Ö, ERYİĞİT R, SAMET R (2021). SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN. in *IEEE Access*, **9**: 157639-157653, 2021, doi: 10.1109/ACCESS.2021.3129600.

RAMAN M R, SOMU N, KİRTHİVASAN K, LİSCANO R, SRİRAM V S (2017). An efficient intrusion detection system based on hypergraph - Genetic algorithm for parameter optimization and feature selection in support vector machine. *Knowledge-Based Systems* **134**: 1-12.

SİNGH S, SİLAKARİ S (2009). A Survey of Cyber Attack Detection Systems. *IJCSNS International Journal of Computer Science and Network Security*, **9** (5), May 2009.

SONG J (2016). Feature selection for intrusion detection system. Ph.D. Thesis, Department of Computer Science Institute of Mathematics, Physics and Computer Science Aberystwyth University.

SONG J, TAKAKURA H, OKABE Y, KWON Y (2007). A Robust Feature Normalization Scheme and an Optimized Clustering Method for Anomaly-Based Intrusion Detection System. *Lecture Notes in Computer Science*, **4443**: 140-151, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-71703-4_14

SUNDARAM A (1996). An introduction to intrusion detection. *XRDS*, **2**: 3-7.

SUNG A H, MUKKAMALA S (2004). The Feature Selection and Intrusion Detection Problems. *In: Maher M.J. (eds)* Advances in Computer Science - ASIAN 2004. Higher-Level Decision Making. ASIAN 2004. *Lecture Notes in Computer Science*, **3321**: 468-482, ISBN: 978-3-540-24087-7.

STOLFO S J, LEE W, PRODROMİDİS A, CHAN P K (2000). "Cost-Based modeling and evaluation for data mining with application to fraud and intrusion detection: Results from the JAM Project", Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00, **2**: 130-144, doi: 10.1109/DISCEX.2000.821515.

TAMA B A, COMUZZİ M, RHEE K H (2019) . TSE-IDS: A Two-Stage Classifier Ensemble for Intelligent Anomaly-Based Intrusion Detection System. İn *IEEE Access*, **7**: 94497-94507, doi: 10.1109/ACCESS.2019.2928048.

TANG C, LUKTARHAN N, ZHAO Y (2020). An Efficient Intrusion Detection Method Based on LightGBM and Autoencoder. *Symmetry*, **12** (9): 1458.

TANRIKULU H (2009). Saldırı Tespit Sistemlerinde Yapay Sinir Ağların Kullanılması, Yüksek Lisans Tezi, Ankara Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

TORABİ M, UDZİR N I, ABDULLAH M T, YAAKOB R (2021). A Review on Feature Selection and Ensemble Techniques for Intrusion Detection System. *International Journal of Advanced Computer Science and Applications (IJACSA)*, **12** (5): 538-553.

WAAD B (2015). On Feature Selection Methods for Credit Scoring. Ph.D. Thesis, Institut Supérieur de Gestion de Tunis. DOI:10.13140/2.1.3354.1443.

WE ARE SOCİAL (2021). Digital 2021 July Global Statshot Report. Erişim Adresi: [https://wearesocial.com/blog/2021/07/digital-2021-i-dati-di-luglio/]ErişimTarihi: 19/11/2021.

WORLD ECONOMİC FORUM (2020). The Global Risks Report 2021. Erişim Adresi: [https://www3.weforum.org/docs/WEF_Global_Risk_Report_2020.pdf]ErişimTarihi: 25/11/2021.

ZHANG O (2015). Tips for data science competitions. Erişim Adresi: [https://www.slideshare.net/OwenZhang2/tips-for-data-science-competitions] Erişim Tarihi: 11.12.2021.

ZHOU Y, CHENG G, JİANG S, DAİ M (2020). Building an efficient intrusion detection system based on feature selection and ensemble classifier. *Computer Networks*, **174** (3), 107247/ISSN: 1389-1286.

ZHOU Z H (2012). Ensemble Methods: Foundations and Algorithms. Chapter 1, pp.:15-17.