Journal of the Faculty of Engineering and Architecture of Gazi University 39:2 (2024) 679-692

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University
Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Increasing the performance of intrusion detection models developed using machine learning method with preprocessing applied to the dataset

Esen Gül İlgün*[1], Refik Samet[2]
[1]Forensic Informatics Program, Institute of Forensic Sciences, Ankara University, 06590, Dikimevi, Ankara, Türkiye
[2]Department of Computer Engineering, Ankara University, 06830, Gölbaşı, Ankara, Türkiye

Highlights:

- Methodology to increase the speed and accuracy of anomaly-based intrusion detection models is proposed
- Effect of the preprocesses applied to the datasets on the anomaly-based intrusion detection models is examined
- Categorical data encoding, scaling and hybrid feature selection preprocesses are used

Keywords:

- Intrusion detection models
- Intrusion detection performance
- Dataset preprocessing
- Machine learning
- Hyper-parameter optimization

Article Info:

Research Article
Received: 26.05.2022
Accepted: 11.04.2023

DOI:

10.17341/gazimmfd.1122021

Correspondence:

Author: Esen Gül İlgün
e-mail: ilgunesengul@gmail.com
phone: +90 544 760 8695

Graphical/Tabular Abstract

In this study; In order to increase the detection rate and rate of anomaly-based intrusion detection models, the effect of preprocessing applied to the datasets on the performance of the models was examined. A four-stage methodology (Figure A) has been proposed: pre-processing of the data set, creating intrusion detection models with pre-processed data sets and machine learning algorithms, pre-evaluating the models, and improving the models.

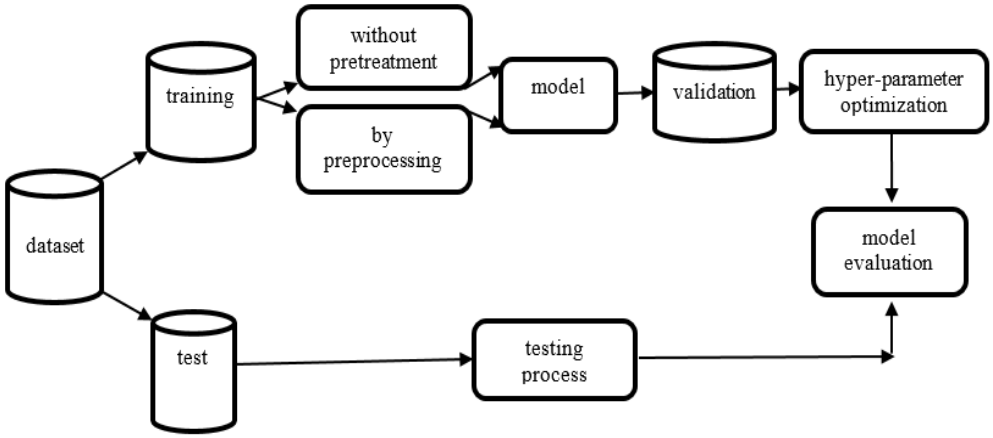


Figure A. Flowchart of the proposed methodology

Purpose:

The aim of the study is to increase the performance (speed and accuracy) of intrusion detection models developed with machine learning methods.

Theory and Methods:

In this study; A four-stage methodology has been proposed to examine the effect of preprocesses applied to the data sets on the success of intrusion detection models developed using machine learning models. Firstly, preprocesses such as the categorical data encoding, scaling and hybrid feature selection were applied to the data sets, then, the intrusion detection models were developed with preprocessed datasets and machine learning algorithms, and, finally, the performance of the models was evaluated and improved by performing hyper-parameter optimization.

Results:

Experimental results showed that the proposed methodology increased the performance (speed and accuracy) of the intrusion detection models, namely, the accuracy of 96.1% and the speed of 0.373 second were obtained in the training data set and the accuracy of 100% and the speed of 0.005 second were obtained in the test dataset.

Conclusion:

In this study; Experimental results showed that preprocessing and hyper-parameter optimization in models increase the intrusion detection speed and accuracy. The most successful results were obtained with the datasets in which all preprocesses were applied together. It has been concluded that in order to achieve more successful results the up-to-date and different data sets should be used, different methods should be tried for each algorithm used, and hyper-parameter optimization should be done.

Journal of the Faculty of Engineering and Architecture of Gazi University 39:2 (2024) 679-692

Mühendislik Mimarlık Fakültesi Dergisi
Journal of The Faculty of Engineering and Architecture of Gazi University

Elektronik / Online ISSN : 1304 - 4915
Basılı / Printed ISSN : 1300 - 1884

Veri setine uygulanan ön işlemler ile makine öğrenimi yöntemi kullanılarak geliştirilen saldırı tespit modellerinin performanslarının arttırılması

Esen Gül İlgün*[1], Refik Samet[2]

[1]Ankara Üniversitesi, Adli Bilimler Enstitüsü, Adli Bilişim Programı, 06590, Dikimevi, Ankara, Türkiye
[2]Ankara Üniversitesi, Bilgisayar Mühendisliği Bölümü, 06830, Gölbaşı, Ankara, Türkiye

ÖNEÇIKANLAR

- Anomali tabanlı saldırı tespit modellerinin hızını ve doğruluğunu artırmak için metodoloji önerildi
- Veri setlerine uygulanan ön işlemlerin anomali tabanlı saldırı tespit modelleri üzerindeki etkisi incelendi
- Kategorik veri kodlama, ölçeklendirme ve hibrit öznitelik seçimi ön işlemleri kullanıldı

Makale Bilgileri

Araştırma Makalesi
Geliş: 26.05.2022
Kabul: 11.04.2023

DOI:
10.17341/gazimmfd.1122021

Anahtar Kelimeler:
Saldırı tespit modelleri, ön işlemler, saldırı tespit performansı, makine öğrenimi, hiper-parametre optimizasyonu

ÖZ

Son yıllarda yapay zekâ teknikleri kullanılarak geliştirilen siber saldırılar sızdıkları sistemin kullanıcı davranışlarını öğrenerek sisteme başarılı bir şekilde entegre olabilmekte ve bu sayede geleneksel güvenlik yazılımları tarafından tespit edilememektedir. Çeşidi ve sayısı hızla artan bu tür siber saldırılar anomali tabanlı Saldırı Tespit Sistemleri (STS) tarafından tespit edilebilmektedir. Ancak bu tür STS'lerin performansları yeterli olmadığı için STS'lerin performanslarının iyileştirilmesi ile ilgili yapılan araştırmaların önemi de artmaktadır. Bu çalışmada, anomali tabanlı saldırı tespit modellerinin tespit hızını ve doğruluğunu arttırmak için dört aşamalı bir metodoloji önerilmiştir. Bu metodoloji kapsamında kullanılan NSL-KDD veri setine ilk önce ön işlem uygulanmadan, daha sonra sırasıyla kategorik veri kodlama, ölçeklendirme, hibrit öznitelik seçimi ön işlemleri ayrı ayrı ve birlikte uygulanarak farklı veri setleri elde edilmiştir. Elde edilen veri setleri ve K-Nearest Neighbor (KNN), Multi Layer Perceptron (MLP), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM) makine öğrenimi algoritmaları kullanılarak çok sayıda saldırı tespit modeli oluşturulmuştur. Son olarak en başarılı sonuçların elde edildiği modellerde hiper-parametre optimizasyonu yapılarak modellerin performansları iyileştirilmiştir. Çalışmanın sonunda eğitim veri seti üzerinde 0,373 s sürede %96,1 saldırı tespit başarısına, test veri seti üzerinde ise 0,005 s sürede %100 saldırı tespit başarısına ulaşılmıştır.

Increasing the performance of intrusion detection models developed using machine learning method with preprocessing applied to the dataset

HIGHLIGHTS

- Methodology to increase the speed and accuracy of anomaly-based intrusion detection models is proposed
- Effect of the preprocesses applied to the datasets on the anomaly-based intrusion detection models is examined
- Categorical data encoding, scaling and hybrid feature selection preprocesses are used

Article Info

Research Article
Received: 26.05.2022
Accepted: 11.04.2023

DOI:
10.17341/gazimmfd.1122021

Keywords:
Intrusion detection models, preprocessing, intrusion detection performance, machine learning, hyperparameter optimization

ABSTRACT

Cyber-attacks developed using artificial intelligence techniques in recent years can be successfully integrated into the system by learning the user behavior of the system they infiltrated, and thus cannot be detected by traditional security software. Such cyber-attacks, of which type and number are increasing rapidly, can be detected by anomaly-based Intrusion Detection Systems (STS). However, since the performance of such STSs is not sufficient, the importance of research on improving the performance of STSs is increasing. In this study, a four-stage methodology is proposed to increase the detection speed and accuracy of anomaly-based intrusion detection models. Different datasets were obtained by applying categorical data coding, scaling, and hybrid feature selection preprocesses separately and together, respectively, to the NSL-KDD dataset used within the scope of this methodology. A large number of intrusion detection models were created using the obtained datasets and machine learning algorithms of K-Nearest Neighbor (KNN), Multi-Layer Perceptron (MLP), Random Forest (RF), eXtreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM). Finally, the performance of the models was improved by performing hyper-parameter optimization in the models where the most successful results were obtained. At the end of the study, 96.1% intrusion detection success was achieved in 0.373 s on the training dataset, and 100% intrusion detection success in 0.005 s on the test dataset.

*Sorumlu Yazar/Yazarlar / Corresponding Author/Authors : *ilgunesengul@gmail.com, samet@eng.ankara.edu.tr / Tel: +90 544 760 8695

1. Giriş (Introduction)

İnternetin kesintisiz olarak birçok alanda sunduğu hizmetlerle, Dünya genelinde 4,8 milyar olan İnternet kullanıcı sayısının [1] 2022 yılına kadar 6 milyara, 2030 yılına kadar ise 7,5 milyara yükseleceği tahmin ediliyor ki bu da 2030 yılında öngörülen 8,5 milyarlık dünya nüfusunun %90'ına denk gelmektedir [2]. Bu hızlı artış, dijital ortamda kritik öneme sahip bilginin boyutunun artması ve yetkisiz kişilerin bu bilgilere ulaşma yönündeki istikrarlı çabaları ile sonuçlanmaktadır. Pratikte tamamen güvenilir bir sistem kurmanın zorluğu, işletim sistemlerindeki açıkları çoğunlukla ilk olarak saldırganların fark etmeleri ve kullanmaları, güvenlik düzeyi yüksek bir sistemin dahi, kullanıcı ayrıcalıklarına sahip yetkili kişiler tarafından suiistimal edilmesi, erişim kontrolü düzeyi yükseldikçe kullanıcı verimliliğinin düşmesi, tüm kripto sistemlerin kırılabilmeleri ve de yazılım teknolojisinin hızlı gelişimi gibi nedenler ise saldırganların bu verilere ulaşma yönündeki çabalarını olumlu sonuçlandırmaktadır [3]. Siber saldırıların verdiği zararın 2025 yılında 10,5 trilyon dolara ulaşabileceği öngörülüyor ki bu zarar, doğal afetlerin ve yasadışı uyuşturucu ticaretinin toplamının verdiği zarardan kat be kat daha büyüktür [4]. 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 [5]. Tüm bu nedenlerden dolayı dünya çapında bilgi güvenliği ve risk yönetimi ürün ve hizmetlerine yapılan harcamaların %11,3 artarak 2023'te 188,3 milyar doları aşacağı tahmin ediliyor [6]. Bu 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, antivirüs yazılımları gibi çekirdek modunda çalışan güvenlik yazılımlarını ve sadece daha önce karşılaşılmış saldırıları tespit edebilen imza tabanlı STS'leri atlatabilen, yeni nesil kötü amaçlı yazılımların artışı ile yakından ilgilidir [7]. Saldırı biçimini, hedef aldığı sisteme göre güncelleyerek sürekli gelişme gösteren kötü amaçlı yazılımların tespiti ve bulaştıkları sistemlerle olan etkileşimlerini anlamak için kapsamlı olarak analiz edilmeleri gerekmektedir [8]. Gelişmiş tekniklerin kullanıldığı siber saldırıların verdikleri zararların engellenmesi aşamasında, makine öğrenimi yöntemleri kullanılarak geliştirilen ve ağ trafiğini analiz ederek, sıfır gün atakları olarak adlandırılan daha önce karşılaşılmamış saldırıların tespitinde oldukça başarılı olabilen anomali tabanlı STS'ler gittikçe daha çok önem kazanmaktadır [9]. Anomali tabanlı STS'lerde algılama bir sınıflandırma görevidir. Bir sınıflandırma görevinde ise amaç, makine öğrenimi algoritmasını örnek verilerle eğiterek yeni gelen örneklerin sınıfının doğru tahmin edilmesini sağlamaktır ve eğitimde kullanılan verilerin niteliği son derece önemlidir. Makine öğrenimi algoritmalarından en üst düzeyde sınıflandırma performansı elde etmenin etkili bir yolu, kullanılan veri setini ön işlemektir.

Bu çalışmanın amacı, veri setlerine uygulanan ön işlemlerin, anomali tabanlı saldırı tespit modellerinin performansları üzerindeki etkisini incelemektir. Bir başka deyişle bu çalışmada, makine öğrenimi yöntemi ile anomali tabanlı saldırı tespit modeli geliştirilirken kullanılan veri setini, en doğru şekilde ön işleyerek modelin tespit başarısını ve hızını arttırmak hedeflenmektedir. Bu amacı gerçekleştirmek için çalışmada, veri setlerine uygulanan ön işlemlerin anomali tabanlı saldırı tespit modelleri üzerindeki etkisi geleneksel yöntemlerle detaylı bir şekilde araştırılmaktadır. Önerilen dört aşamalı metodolojide (1) veri setine uygulanan ön işlemler için çok sayıda yöntem denenmiş, (2) elde edilen bu farklı veri setleri çeşitli makine öğrenimi algoritmaları ile sınanmıştır. (3) Bu algoritmaların kıyaslanması ve (4) en başarılı modellerde hiper-parametre optimizasyonu yapılarak performansların iyileştirilmesi amaçlanmıştır [10].

Çalışmanın 2. bölümünde, ilgili çalışmalar özetlenmiştir. 3. bölümde, önerilen metodoloji detaylı olarak anlatılmıştır. 4. bölümde, önerilen metodolojinin uygulaması ve alınan sonuçlar verilmiştir. 5. bölümde ise sonuçların analizi ve değerlendirilmesi yapılmıştır.

2. İlgili Çalışmalar (Related Works)

Literatür incelendiğinde; Davis ve Clark, makine öğrenimi yöntemi ile geliştirilen anomali tabanlı STS'lerin performanslarını arttırmak ve çeşidi artan saldırıların tespiti için istek paketlerin derinlemesine incelenmesini önermiş ve de veri ön işlemenin anomali tabanlı STS'lerin başarısı üzerinde büyük bir etkiye sahip olduğunu vurgulamıştır [11]. Naseer ve Saleem çalışmalarında, çeşitli kategorik veri kodlama yöntemlerini deneyerek, kullandıkları veri setine en uygun yöntemi seçmiş, Derin Evrişimli Sinir Ağı (DCNN) algoritması ile kurulan modellerde rastgele arama yöntemi kullanarak, hiper-parametre optimizasyonu yapmıştır. Ön işlemlerin ve hiper-parametre optimizasyonunun, oluşturulan modellerin saldırı tespit oranını ve hızını kayda değer ölçüde iyileştirdiğini belirtmiştir [12]. Hancock ve Khoshgoftaar, kararlı kategorik veri kodlama tekniklerinin (leave one out encoder, hashing encoder vs.) düşük çalışma süresi ve düşük hesaplama karmaşıklığına sahip olmalarından dolayı büyük boyutlu veri setleri için uygun olduğu vurgulamıştır [13]. Tang vd., one hot encoding yöntemi ile kategorik veri kodlama ve LightGBM algoritması ile öznitelik seçimi ön işlemleri uyguladıkları veri seti ve Autoencoder (AE) algoritması ile oluşturdukları saldırı tespit modelinde %89, 82 doğruluk oranına ulaşmıştır [14]. Aslan vd., zararlı yazılımların sistem içinde gösterdikleri davranışları analiz ederek, bu yazılımları tespit etmek için Eksiltici Merkezi Davranış Modeli (EMDM) önermiştir. Önerilen modelde, zararlı yazılım davranışları ve davranışların gerçekleştirildiği sistem analiz edilerek öznitelikler oluşturulmuştur. Ayrıca, yeni bir öznitelik seçim algoritması önerilerek elde edilen öznitelikler azaltılmıştır. Önerilen model ile algılama oranı, yanlış pozitif oranı ve doğruluk metriklerinde sırasıyla %99,9, %0,2 ve %99,8 oranlarına ulaşılmıştır [15]. Mazini vd., veri setine kategorik veri kodlama ve ölçeklendirme ön işlemlerinden sonra hiper-parametre optimizasyonu uygulamış ve yapay arı kolonisi (ABC) algoritması ile öznitelik seçimi yapmıştır. Elde edilen veri seti ve AdaBoost.M2 sınıflandırıcısı ile oluşturulan modelde %99,61 algılama, %0,01 yanlış algılama, %98,90 doğru tespit oranlarına ulaşılmıştır [16]. Balakrishnan vd., bilgi kazanım oranına göre öznitelik seçimi yaparak, elde edilen öznitelik alt kümesine sahip veri seti ve SVM sınıflandırma algoritması ile DoS (Denial of Service Attack) saldırılarında % 99,11 tespit başarısı, 2,08 s tespit süresi sonuçlarına ulaşmıştır [17]. Torabi vd., geliştirilen saldırı tespit modellerinin genelleme başarısını kanıtlamak için farklı ve güncel veri setleri ile kullanılmasının öneminden bahsetmiştir [18]. Özkan Okay vd., hibrit saldırı tespit modelini önermiş, Feature Selection Approach (FSAP) algoritması ile öznitelik seçimi ön işlemi uygulanan KDD'99 ve UNSW-NB15 veri setleri ile %99,65 ve %99,17 doğruluk oranları elde etmiştir [19]. Ambusaidi vd., karşılıklı bilgi (MI) ve sarmal ardışık ileri yönde seçim (SFS) yöntemlerini kullanarak hibrit öznitelik seçimi uyguladıkları veri setleri ile %98,90 saldırı tespit başarısı ve %0,521 yanlış pozitif oranına ulaşmıştır [20]. Chen vd., Principal Component Analysis (PCA), GA ve C4.5 tekniklerinin seçtiği özniteliklerin farklı birleşim ve kesişim kombinasyonlarından oluşan veri setlerini kullanarak, PCA ve GA tekniklerin ortak seçtiği öznitelikler ile en başarılı sonuca ulaşmıştır [21]. Song, geleneksel öznitelik seçimi algoritmalarının, değişken boyutlu veri setleri için yeterli olmadığından, bu sorunun çevrimiçi öznitelik seçimi algoritmaları ile çözülebileceğinden bahsetmiştir [22]. Kanimozhi ve Jacob, MLP algoritması kullanarak oluşturdukları

anomali tabanlı saldırı tespit modelinde, grid search yöntemi kullanarak gizli katman sayısı ve alfa parametreleri için hiper-parametre optimizasyonu yapmış ve %99,97 doğruluk, %0,001 yanlış pozitif, %99 F-ölçütü oranlarına ulaşmıştır [23]. Latah ve Toker çalışmalarında, yazılım tanımlı ağlarda (SDN), anomali tabanlı saldırı tespiti için NSL-KDD veri seti, 12 farklı sınıflandırıcı ve veri setinden öznitelik çıkarımı için PCA yaklaşımını kullanmıştır. Yapılan deneyler sonucunda, Decision Tree (DT) algoritması ile kurulan model doğruluk, kesinlik, F1-ölçütü, AUC ve McNemar metriklerinde en iyi performansı göstermiştir. Bagging ve boosting yaklaşımları, KNN, ELM, NN, RF, SVM ve LDA gibi diğer geleneksel makine öğrenimi yöntemlerini, %99,5'lik bir güven düzeyiyle geride bırakırken, LogitBoost ile FAR ve recall metriklerinde en iyi sonuçlara ulaşılmıştır. En iyi test süresi ise ELM ile elde edilmiştir [24]. Uğurlu vd. çalışmalarında, karanlık ağ trafiğinin tespiti ve sınıflandırılması için kullandıkları CICDarknet2020 veri seti içerisinde bulunan 82 adet öznitelik içerisinden, ağırlıklandırma işlemi ile 30 adet öznitelik seçmiştir. Çalışmada hiper-parametre ayarı için ızgara yöntemi kullanılmış, yapılan deneysel çalışmalar sonucunda, Karar Ağacı algoritması ile %93,32 doğruluk oranına ulaşılmıştır [25]. Literatür incelemesi sonucunda (Tablo 1), kullanılan veri setlerine uygulanan kategorik

Tablo 1. İlgili çalışmalar (Related works)

Makale Bilgileri	Kullanılan Yöntem	Ulaşılan Sonuç
Davis ve Clark (2011)	Anomali tabanlı STS kullanılan ağ trafiği özellikleri ve veri seti ön işleme tekniklerinin kapsamlı bir incelemesi	Çeşidi artan saldırıların etkili tespiti için, hazır veri setlerini ön işlemek yerine, ağ trafiği içerisindeki istek paketleri derinlemesine incelenmeli ve paket içeriğinden öznitelikler türetilmelidir.
Balakrishnan vd. (2014)	Öznitelik seçimi ve sınıflandırma teknikleri kullanılarak saldırı tespiti	DoS (Denial of Service Attack) saldırılarında % 99,11 tespit başarısı, 2,08 s tespit süresi sonuçlarına ulaşılmıştır.
Ambusaidi vd. (2014)	İzinsiz giriş tespiti için hibrid öznitelik seçimi yaklaşımı	%98,90 saldırı tespit oranı ve %0,521 yanlış pozitif oranına ulaşılmıştır
Song (2016)	Saldırı tespit sistemi için öznitelik seçimi	Geleneksel öznitelik seçimi algoritmaları değişken boyutlu veri setleri için yeterli değildir. Bu sorun çevrimiçi öznitelik seçimi algoritmaları ile çözülebilir.
Naseer and Saleem (2018)	Derin Evrişimli Sinir Ağlarını (DCNN) kullanarak ağ saldırı tespiti	Leave-One-Out kategorik veri kodlama yöntemi kullanılarak ve Random-search yöntemi ile hiper parametre optimizasyonu yapılarak NSLKDDTest+ ve NSLKDDTest21 için sırasıyla %85.22 ve %69.56 sınıflandırma doğruluğu elde edilmiştir.
Latah ve Toker (2018)	Öznitelik çıkarımı ve 12 farklı sınıflandırıcı kullanılarak saldırı tespiti	NSL-KDD veri setine PCA yöntemi ile öznitelik çıkarımı ve NN, LDA, DT, RF, Linear SVM, KNN, NB, ELM, AdaBoost, RUSBoost, LogitBoost, BaggingTrees sınıflandırıcıları ile kurulan modellerde, DT ile %99.70 eğitim, %88.74 test doğruluğuna ulaşılmıştır.
Kanimozhi and Jacob (2019)	Bulut bilişim kullanarak CSE-CICIDS2018 veri seti üzerinde hiper-parametre optimizasyonu ile yapay zekâ tabanlı ağ saldırı tespiti	GridSearchCV yöntemi ile hiper-parametre optimizasyonu yapılarak %99,97 doğruluk, %0,001 yanlış pozitif, %99 F-ölçütü oranlarına ulaşılmıştır
Mazini vd. (2019)	Hibrit yapay arı kolonisi ve AdaBoost algoritmaları kullanılarak anormali tabanlı saldırı tespiti	%99,61 algılama, %0,01 yanlış algılama, %98,90 doğru tespit oranlarına ulaşılmıştır.
Hancock and Khoshgoftaar (2020)	Derin öğrenme algoritmaları için kategorik veri kodlama tekniklerinin kapsamlı incelenmesi	Büyük veri setlerinde, kategorik verileri kodlamak için hesaplama süresinin uzunluğundan dolayı algoritmik teknikler önerilmez iken, kararlı tekniklerin (leave one out encoder, hashing encoder vs.) düşük çalışma süresinden dolayı büyük boyutlu veri setleri için uygun olduğu görülmüştür.
Tang vd. (2020)	LightGBM ve Autoencoder tabanlı etkili bir saldırı tespit yöntemi	Min-max ölçeklendirme, One Hot Encoding kategorik veri kodlama ve LightGBM algoritması ile öznitelik seçimi ön işlemleri uygulanarak Autoencoder (AE) algoritması ile %89, 82 doğruluk oranına ulaşılmıştır.
Chen vd. (2020)	Veri setlerinde filtreleme, sarmal ve gömülü öznitelik seçim yöntemlerini birleştirerek topluluk öznitelik seçim yöntemi yaklaşımı	PCA (filtreleme) ve GA (sarmal) tekniklerin ortak seçtiği öznitelikler ile en başarılı sonuca ulaşılmıştır.
Aslan vd. (2020)	Kötü amaçlı yazılımları tespit etmek için çıkarımsal merkez davranış modeli kullanımı	Önerilen model ile algılama oranı, yanlış pozitif oranı ve doğruluk metrikleri sırasıyla %99,9, %0,2 ve %99,8 olarak ölçülmüştür.
Torabi vd. (2021)	Saldırı tespit sistemi için öznitelik seçimi ve topluluk teknikleri üzerine bir inceleme	Optimizasyona dayalı öznitelik seçim yöntemleri ve topluluk öğrenme yöntemleri kullanılarak algoritmaların ya da yöntemlerin tek tek kullanımına oranla daha başarılı sonuçlara ulaşılmıştır.
Özkan-Okay vd. (2021)	WLAN'da siber saldırı tespiti için hibrit saldırı tespiti yaklaşımı	KDD'99, UNSW-NB15 veri setlerinde %99,65 ve %99,17 doğruluk oranları elde edilmiştir
Uğurlu vd. (2023)	Öznitelik seçimi ve hiper-parametre optimizasyonu yapılarak, karanlık ağ trafiğinin makine öğrenimi yöntemleri ile tespiti	CICDarknet2020 veri seti ve Karar Ağacı algoritması ile %93,32 doğruluk oranına ulaşılmıştır

veri kodlama, ölçeklendirme, öznitelik seçimi ön işlemlerine aynı çalışmada odaklanan, bu ön işlemlerin ayrı ayrı ve birlikte kullanımının saldırı tespit performansı üzerindeki etkisini ayrıntılı olarak inceleyen, elde edilen modellerde hiper-parametre optimizasyonunun yapıldığı yeterince çalışma olmadığı fark edilmiştir. Bu nedenle bu çalışmada, bahsedilen ön işlemlere odaklanılmış, kullanılan veri setine en uygun ön işlemin seçilebilmesi için çok sayıda yöntem ve teknik denenmiştir. Bu ön işlemlerin ayrı ayrı ve birlikte kullanımının saldırı tespit modellerinin performansları üzerindeki etkisi incelenerek, saldırı tespit başarısı ile hızının arttırılması ve son olarak elde edilen en başarılı modellerde hiper-parametre optimizasyonu yapılarak modellerin performanslarının iyileştirilmesi hedeflenmiştir.

3. Önerilen Metodoloji (Recommended Methodology)

Bu çalışmada, makine öğrenimi yöntemi ile geliştirilen anomali tabanlı saldırı tespit modellerinin saldırı tespit performanslarını arttırmak için veri setine uygulanan ön işlemlere ve hiper-parametre optimizasyonuna odaklanan dört aşamalı bir metodoloji önerilmiştir:

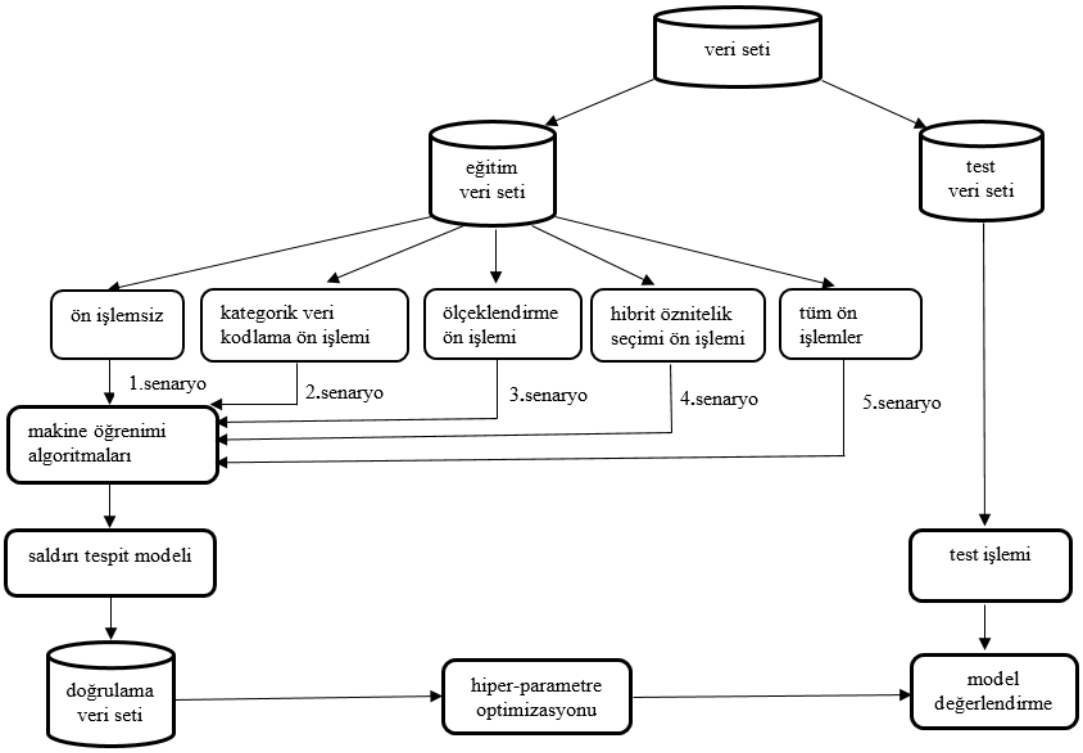
1) Veri setinin ön işlenmesi;
2) Ön işlenmiş veri setleri ve makine öğrenimi algoritmaları ile saldırı tespit modelleri oluşturulması;
3) Modellerin ön değerlendirilmesi;
4) Model iyileştirilmesi.

Çalışmada önerilen metodoloji Şekil 1'de gösterildiği gibi beş farklı senaryo ile uygulanmıştır.

İlk senaryoda NSL-KDD eğitim veri setine ön işlem uygulanmadan (Şekil 1'de 1. senaryo), ikinci senaryoda kategorik veri kodlama ön işlemi uygulanarak (Şekil 1'de 2. senaryo), üçüncü senaryoda min-max ölçeklendirme ön işlemi uygulanarak (Şekil 1'de 3. senaryo), dördüncü senaryoda hibrit öznitelik seçimi ön işlemi uygulanarak (Şekil 1'de 4. senaryo) ve beşinci senaryoda ise eğitim veri setine kategorik veri kodlama, ölçeklendirme ön işlemleri birlikte uygulandıktan sonra elde edilen veri setinden öznitelik seçimi ön işlemi yapılarak (Şekil 1'de 5. senaryo) elde edilen veri setleri ile KNN, MLP, RF, XGBoost ve LightGBM makine öğrenimi algoritmaları eğitilerek çok sayıda saldırı tespit modeli oluşturulmuştur. Daha sonra test veri setleri kullanılarak modellerin performansları analiz edilmiştir. Son aşamada elde edilen en başarılı modellerde, eğitim veri setinin %20 sini içeren doğrulama veri seti kullanılarak hiper-parametre optimizasyonu yapılmıştır. Böylece algoritmaların kullanılan veri seti ile maksimum başarı sağlayacak parametre kombinasyonuna ulaşılmış ve modellerin performansları iyileştirilmiştir.

3.1. Veri Setinin Ön İşlenmesi (Preprocessing of The Dataset)

KDD99 veri seti, ABD Hava Kuvvetleri yerel alan ağı'nın simülasyonu kurularak toplanan, dokuz haftalık TCP dökümü verileri ile elde edilen DARPA veri setinin alt kümesidir. Bu çalışmada kullanılan NSL-KDD veri seti ise, KDD99 veri seti üzerinde makine öğrenimi algoritmalarının daha yüksek performansta çalışabilmeleri için tekrar eden bağlantı kayıtlarının silinerek, veri boyutunun düşürülmesiyle elde edilmiştir. Bu şekilde veri setinin, özellikle hesaplama maliyetinden dolayı parçalara bölünmesi zorunluluğu da ortadan kaldırılmıştır [26]. Fakat NSL-KDD veri seti, bir sanal bilgisayar ağı simülasyonudur ve sonuç olarak, deneyler gerçek ortam söz konusu olduğunda çelişkili sonuçlar verebilir [27]. Kyoto 2006+ gibi gerçek ağ trafiği verilerini içeren veri setlerinin varlığına rağmen, çalışmada NSL-KDD veri setinin kullanılmasının sebepleri: literatürde bu veri seti ile yapılmış çok sayıda çalışma olmasından dolayı, kapsamlı performans karşılaştırma olanağı olması, veri setinin boyutunun, kısıtlı kaynak söz konusu olduğu için hesaplama maliyeti bakımından sonuca ulaşmaya imkân tanıması ile çalışmanın veri setine uygulanan ön işlemler ile saldırı tespit modellerinin performanslarını arttırmaya yönelik, başka veri setleri ile de test



Şekil 1. Önerilen metodolojinin akış diyagramı (Flowchart of the proposed methodology)

edilebilecek özgün bir yöntem önerebilmesi olmuştur. Veri setindeki her bağlantı kaydı 9 temel ve 32 adet türetilmiş olmak üzere 41 adet özniteliğe ve bağlantının normal ya da anormal olduğunu belirleyen bir de sınıf değişkenine sahiptir. Bu 41 öznitelik aşağıdaki gibi dört kategoriye ayrılarak ifade edilmiştir [22].

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.

Bu 41 öznitelik Tablo 2'de gösterildiği gibidir. Bu özniteliklerin ilk 1-9 arasındaki öznitelikler temel öznitelikler, 10-22 arasındakiler içerik öznitelikleri, 23-31 arasındakiler Zaman Bağlı Trafik Öznitelikleri, 32-41 arasındakiler ise Bağlantı Tabanlı Trafik Öznitelikler kategorilerine aittir.

Veri setlerindeki sınıfların dengesiz dağılması, kategorik değerli veriler, farklı aralıklardaki veri değerleri, veri setini temsil etmede yetersiz öznitelikler, makine öğrenimi algoritmalarının sınıflandırma performansını düşürebilir. Bu çalışmada, kullanılan makine öğrenimi algoritmalarından en üst düzeyde performans elde edebilmek için NSL-KDD veri setine sırasıyla kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemleri uygulanmıştır.

Kategorik Veri Kodlama: Çoğu makine öğrenimi algoritması kategorik değere sahip verileri, aralarında matematiksel ya da mantıksal bir ilişki bulunmadığı için işleyemez. Bu nedenle kategorik değerler sayısal değerlere dönüştürülmelidir. Bu çalışmada, NSL-KDD veri setindeki kategorik değerli verileri, sayısal formata dönüştürürken en uygun tekniği seçmek adına Label Encoding, One Hot Encoding, Target Encoding, Leave One Out Encoding, Weights of Evidence Encoding, James Stein Encoding, CatBoost Encoding, M-Estimate Encoding teknikleri uygulanarak tekniklerin başarısı boyut (öznitelik sayısı), eğitim süresi, ortalama sınıflandırma başarısı ve başarının standart sapması ölçütlerine göre kıyaslanmıştır.

Ölçeklendirme: Çalışmada, NSL-KDD veri setindeki farklı aralıklardaki veri değerlerinin ortak bir ölçekte eşleştirilerek söz konusu verilerin daha objektif karşılaştırılabilmesi için literatürdeki STS çalışmalarında da yaygın kullanılan, en küçük değere sahip veri 0, en büyük değere sahip veri 1 olacak şekilde, diğer bütün veri değerlerinin bu 0-1 aralığına yayıldığı min-max ölçeklendirme yöntemi kullanılmıştır [28-31].

Öznitelik Seçimi: Çalışmada, veri setini temsil etmede yetersiz, algoritmanın sınıflandırma performansını düşürecek öznitelikleri veri setinden çıkarmak için hibrit öznitelik seçim yöntemi kullanılmıştır. Hibrit öznitelik seçim yöntemi ile filtreleme, sarmal ve gömülü öznitelik seçim yöntemlerinin birlikte kullanımlarının birbirlerinin dezavantajlarının yok edip, eksikliklerini tamamlaması hedeflenmiştir [32]. Filtreleme, sarmal, gömülü öznitelik seçimi yöntemlerini temsilen kullanılan MI, GA, XGBoost yöntemlerinin ayrı ayrı seçtiği özniteliklere ve bu özniteliklerin ikili, üçlü kesişim ve birleşimlerinden oluşan özniteliklere sahip 11 farklı veri seti elde edilmiştir. MI ve XGBoost yöntemleri öznitelikleri önemlerine göre sıralarken, GA yöntemi öznitelikleri, kullanılan sınıflandırma algoritmasının performansına bağlı olarak seçer ya da eler [33-35]. 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.

3.2. Saldırı Tespit Modelleri Oluşturulması
(Creation of Intrusion Detection Models)

Önerilen metodolojinin 2. aşamasında KNN, MLP, RF, XGBoost ve LightGBM makine öğrenimi algoritmaları, metodolojinin 1. aşamasında uygulanan ön işlemler sonucu elde edilen veri setleri ve ön işlem uygulanmamış veri seti ile eğitilerek çok sayıda saldırı tespit modeli oluşturulmuştur.

3.3. Modellerin Ön Değerlendirilmesi
(Preliminary Evaluation of Models)

Önerilen metodolojinin 2. aşamasında, ön işlem uygulanmamış ve ön işlenmiş veri setleri ile makine öğrenimi algoritmaları kullanılarak oluşturulan modellerin performanslarının, hiper-parametre optimizasyonu öncesi ön değerlendirilmesi, saldırı türü ve sayısı eğitim setinden farklı olan test veri seti ile Tablo 3'te gösterilen karışıklık matrisinden elde edilen değerlendirme metrikleri dikkate alınarak yapılmıştır.

Doğruluk= (DP + DN) \ (DP + DN + YP + YN)
Kesinlik= DP \ (DP + YP)
Duyarlılık= DP \ (DP + YN)
F-ölçütü= 2 * ((kesinlik * duyarlılık) \ (kesinlik + duyarlılık))

Tablo 2. NSL-KDD veri seti öznitelikleri (NSL-KDD dataset attributes)

Öznitelik adı ve numarası
duration (1), protocol_type (2), service (3), flag (4) src_bytes (5), dst_bytes (6), land (7), wrong_fragment (8), urgent (9), hot (10), num_failed_logins (11), logged_in (12), num_compromised (13), root_shell (14), su_attempted (15), num_root (16), num_file_creations (17), num_shells (18), num_access_files (19), num_outbound_cmds (20), is_host_login (21), is_guest_login (22), 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)

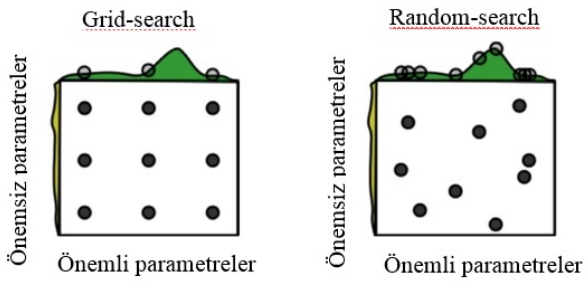
Tablo 3. Karışıklık Matrisi (Confusion Matrix)

Karışıklık matrisi	Tahmin edilen pozitif (saldırı)	Tahmin edilen negatif (normal)
Gerçek Pozitif	Doğru Pozitif (DP)	Yanlış Negatif (YN)
Gerçek negatif	Yanlış Pozitif (YP)	Doğru Negatif (DN)

3.4. Model İyileştirilmesi - Hiper-parametre Optimizasyonu
(Model Improvement – Hyperparameters Optimization)

Model parametreleri, değeri eğitim verilerinden öğrenilen yapılandırma değişkenleridir. Örneğin, Yapay Sinir Ağları'ndaki (YSA) gizli katman sayısı, KNN'deki komşu sayısı, XGBoost'taki maksimum derinlik sayısı. Model hiper-parametresi ise, değeri veri seti ya da problem durumu gibi etkenlere göre değişiklik gösteren, modeli tasarlayan kişi tarafından belirlenen ve modelin performansına optimum katkı sağlaması beklenen parametre kombinasyonudur. Örneğin, YSA ile kurulan bir modelin performansına optimum katkı sağlayacak gizli katman, öğrenme oranı, alfa değeri, iterasyon sayısı değerlerinin belirlenmesi işlemi hiper-parametre optimizasyonudur. Hiper-parametre optimizasyonu kayıp fonksiyonun optimize edilme sürecinde gerçekleştirilir ve bunun için çeşitli arama yöntemleri kullanılır. Bu yöntemlerden en bilinenlerin ikisi: grid search ve random search yöntemleridir.

Grid search yönteminde hiper-parametreler belirlenirken, verilen parametrelerin tüm kombinasyonları için oluşturulan modeller değerlendirilirken; Random search yönteminde hiper-parametre araması, verilen parametrelerin rastgele kombinasyonları kullanılarak yapılır (Şekil 3).



Şekil 3. Gridsearch ve random search yöntemleri
(Grid search and random search methods-Bergstra ve Bengio, 2012)

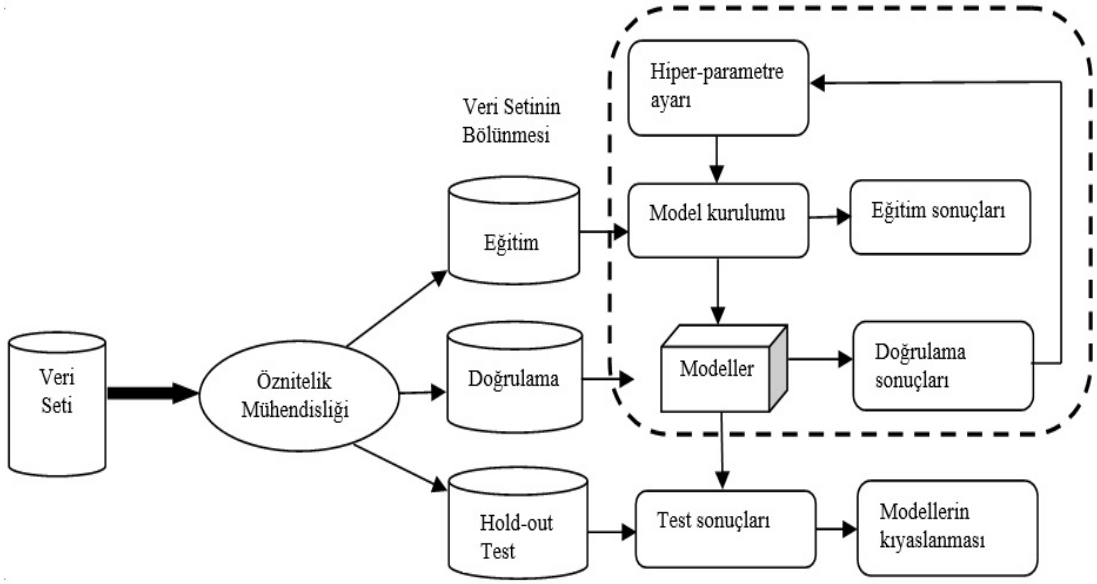
Derin sinir ağlarında hiper-parametre optimizasyonunu random search yöntemi ile gerçekleştiren bir çalışmadan alınan sonuçlar [36], grid search and manual search yöntemlerini kullanan çalışmaların sonuçları ile kıyaslandığında, aynı etki alanı üzerinde random search yönteminin, diğer iki yönteme göre daha düşük bir hesaplama süresinde yedi veri setinden dördünde istatistiksel olarak eşit performans, birinde üstün performans göstermiştir. Gauss süreç analizi, çoğu veri seti için hiper-parametrelerin sadece birkaçının gerçekten önemli olduğunu ve farklı veri setlerinde farklı hiper-parametrelerin önemli olduğunu ortaya koymaktadır. Bu da yeni veri setleri için algoritmaları yapılandırmada grid search yöntemini kötü bir seçim haline getirmektedir. Bu çalışmada hiper-parametre optimizasyonu, özellikle büyük boyutlu veri setlerinde düşük hesaplama maliyeti ile başarılı sonuçlara ulaşabilmesinden dolayı random search yöntemi ile yapılmıştır.

4. Uygulama Sonuçları ve Tartışmalar
(Implementation Results and Discussions)

Yapılan çalışmada önerilen metodoloji beş farklı senaryo ile uygulanmış, her senaryo uygulaması sonucunda saldırı tespit modelleri oluşturulmuş ve test veri seti kullanılarak modellerin başarı durumları, doğruluk, kesinlik, duyarlılık ve F-ölçütü metrikleriyle analiz edilmiştir. Modellerin saldırı tespit başarısını değerlendirmek için, çalışmada kullanılan NSL-KDD gibi sınıf değişkeni dengesiz dağılan veri setlerinde doğruluk metriğinden daha sağlıklı fikir verdiği [37] ve FP ile FN tahmininin maliyetinin yüksek olduğu kritik durumlar göz önünde bulundurulduğu için F-ölçütü dikkate alınmıştır.

4.1. 1.Senaryo: Ön İşlem Yapılmamış Veri Setlerinin Makine Öğrenimi Algoritmaları Üzerinde Uygulanması
(Scenario 1: Implementation of Unprocessed Datasets on Machine Learning Algorithms)

NSL-KDD veri setine ön işlem yapılmadan, KNN, MLP, RF, XGBoost, LightGBM algoritmaları ile modeller oluşturulmuş ve test veri seti ile test edilmiştir. Birinci senaryonun uygulanması ile alınan sonuçlar (Tablo 4) değerlendirildiğinde; 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 2. Hiper-parametre optimizasyonu akış şeması (Hyper-parameter optimization flowchart)

En yüksek 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. İlk senaryo sonunda, makine öğrenimi algoritmalarının, ön işlem uygulanmamış veri setleri üzerinde uygulandığında yeterli performansı gösteremedikleri gözlemlenmiştir.

4.2. 2.senaryo: Kategorik Veri Kodlama Ön İşlemi Yapılan Veri Setlerinin Makine Öğrenimi Algoritmaları Üzerinde Uygulanması
(Scenario 2: Application of Categorical Data Encoding Pre-Processed Datasets on Machine Learning Algorithms)

Kullanılan NSL-KDD veri setine en uygun kategorik veri kodlama tekniğini belirleyebilmek için veri setindeki kategorik verilere, sekiz farklı kategorik veri kodlama tekniği uygulanırken sınıflandırıcı olarak dengesiz veri setlerindeki başarısı, eğitim ve test hızının yüksek olmasından dolayı RF algoritması kullanılmıştır [38]. Tablo 5'teki sonuçlar değerlendirildiğinde, NSL-KDD veri setine en uygun kodlama tekniğinin "Birini Dışarıda Bırak (Leave One Out Encoding)" olduğu görülmüştür. Birini Dışarıda Bırak kategorik veri kodlama tekniği uygulanarak elde edilen veri setleri üzerinde makine öğrenimi algoritmaları çalıştırılarak oluşturulan modeller, test veri seti kullanılarak test edilmiş ve Tablo 6'daki sonuçlara ulaşılmıştır.

Tablo 6'daki sonuçlar değerlendirildiğinde: Ön işlem olarak, veri setine uygun olarak seçilen Birini Dışarıda Bırak 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. Eğitim veri seti ve test veri seti üzerinde en yüksek saldırı tespit hızı artışı %70 ve %62 ile XGBoost ile kurulan modelde görülmüştür. En yüksek F-ölçütü değeri artışı ise %5,9 ile LightGBM ile kurulan modelde görülmüştür. KNN algoritmasında ise eğitim veri seti üzerinde saldırı tespit hızı yaklaşık %65 oranında, F-ölçütü değeri ise %0,9 oranında düşmüştür. 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.

4.3. 3.Senaryo: Ölçeklendirme Ön İşlemi Yapılan Veri Setlerinin Makine Öğrenimi Algoritmaları Üzerinde Uygulanması
(Scenario 3: Application of Scaling Pre-processed Datasets on Machine Learning Algorithms)

Min-Max yöntemiyle ölçeklendirme ön işlemi yapılan veri setleri üzerinde KNN, MLP, RF, XGBoost, LightGBM algoritmaları çalıştırılmış ve Tablo 7'deki sonuçlara ulaşılmıştır.

Tablo 4. Ön işlem yapılmadan alınan sonuçlar (Results without pre-process)

Algoritmalar	Eğitim Süresi (s)	Test Süresi (s)	Doğruluk	Kesinlik	Duyarlılık	F -ölçütü
KNN	0,031	64,146	77,2%	96,4%	62,3%	75,7%
MLP	18,675	0,031	77,3%	93,0%	65,1%	76,6%
RF	12,531	0,334	77,1%	96,7%	61,9%	75,5%
XGBoost	16,010	0,047	79,4%	96,7%	66,1%	78,5%
LightGBM	1,582	0,091	79,4%	96,6%	66,1%	78,5%

Tablo 5. Kategorik veri kodlama tekniklerinin kıyaslanması (Comparison of categorical data coding techniques)

Kategorik veri kodlama teknikleri	Boyut (öznitelik sayısı)	Eğitim süresi (s)	Ortalama başarı	Başarının standart sapması
Etiket Kodlama (Label Encoding)	3	111.415914	0.998968	0.000263
Tek Sıcak Kodlama (One Hot Encoding)	84	136.313674	0.998960	0.000228
Hedef Kodlama (Target Encoding)	3	108.701497	0.998976	0.000214
Birini Dışarıda Bırak Kodlaması (Leave One Out Encoding)	3	81.962925	1.0	0.0
Kanıt Ağırlığı Kodlaması (Weight of Evidence Encoding)	3	109.278701	0.998968	0.000222
James Stein Kodlama (James Stein Encoding)	3	107.59389	0.998984	0.000194
CatBoost Kodlama (CatBoost Encoding)	3	143.380519	0.998833	0.000198
M-Tahmini Kodlama (M-Estimate Encoding)	3	108.857498	0.998976	0.000196

Tablo 6. Kategorik veri kodlama ön işlemi yapılarak alınan sonuçlar
(The results obtained by pre-processing categorical data coding)

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

Tablo 7. Ölçeklendirme ön işlemi yapılarak alınan sonuçlar (Results obtained by performing scaling preprocessing)

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

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. 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. 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 kullanılması ve de oluşturulan modellerde hiper-parametre optimizasyonu ile sağlanabilir.

4.4. 4.Senaryo: Hibrit Öznitelik Seçimi Ön İşlemi Yapılan Veri Setleri Üzerinde Makine Öğrenimi Algoritmalarının Uygulanması
(Scenario 4: Application of Machine Learning Algorithms on Hybrid Feature Selection Preprocessed Datasets)

Çalışmada, kümelerin keşişim ve birleşim özelliklerinden esinlenerek, NSL-KDD veri seti üzerinde hibrit öznitelik seçimi yöntemi uygulanırken, ilk önce MI, GA ve XGBoost öznitelik seçim yöntemleri ayrı ayrı uygulanmış ve her bir yöntemin seçtiği özniteliklerden oluşan üç farklı veri seti, daha sonra bu özniteliklerin kesişim ve birleşimlerinden oluşan 8 farklı veri seti olmak üzere 11 farklı veri seti elde edilmiştir (Tablo 8'de yer alan öznitelik numaralarına karşılık gelen öznitelikler için Tablo 2). Tablo 8'e bakıldığında filtreleme, sarmal ve gömülü öznitelik seçimi yöntemlerinin ortak olarak seçtiği öznitelikler (f∩s∩g); 4 (flag), 6 (dst bytes) (temel öznitelikler), 12 (logged in) (içerik özniteliği), 25 (serror rate) (Zaman Bağlı Trafik Öznitelikleri), 32 (dst host count), 33(dst host srv count), 35(dst_host_diff_srv_rate), 37(dst_host_srv_diff_host_rate), 39 (dst_host_srv_serror_rate) (Bağlantı Tabanlı Trafik Öznitelikleri). En çok sayıda öznitelik, Bağlantı Tabanlı Trafik Öznitelikleri kategorisinden seçilmiştir. Üç öznitelik seçim yönteminin de seçmediği öznitelikler ise: 7(land), 9(urgent), 19(num_access_files), 21(is_hot_login), 22(is_guest_login)'dir ve bu öznitelikler, temel öznitelikler ve içerik öznitelikleri kategorilerine aittir.

Elde edilen 11 faklı veri seti ve KNN, MLP, RF, XGBoost, LightGBM makine öğrenimi algoritmaları ile oluşturulan modellerin test veri seti kullanılarak eğitim süresi, test süresi, doğruluk, kesinlik, duyarlılık ve F-ölçütü metrikleriyle değerlendirilmesi ile en başarılı algoritma ve öznitelik alt kümesi birlikteliğinin yer aldığı Tablo 9'daki sonuçlara ulaşılmıştır.

Senaryonun uygulama sonuçları değerlendirildiğinde: Algoritmaların en başarılı modelleri kurdukları veri setlerinin hibrit öznitelik seçimi yönteminin uygulanması ile elde edilen, KNN için f∩g, MLP için g∪s, RF için f∩g, XGBoost için f∪g, LightGBM için ise f∪s∪g 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. 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. 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

Tablo 8. Öznitelik seçim yöntemleri ile elde edilen öznitelik alt kümeleri: Küme: Öznitelik alt kümesi; f: MI (filtreleme); **s**: GA (sarmal); g: XGBoost (gömülü); ∩: kesişim; ∪: birleşim (Feature subsets obtained by feature selection methods: Set: Subset of attributes; f: MI (filtering); s: GA (spiral); g: XGBoost (embedded); ∩: intersection; ∪: conjunction)

Küme	Öznitelik numarası
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.
f∩s	2,4,6,12,25,27,31,32,33,35,37,38,39.
f∩g	1,3,4,5,6,12,23,24,25,26,28,32,33,34,35,36,37,39.
g∩s	4,6,8,11,12,15,18,25,32,33,35,37,39.
f∪s	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.
f∪g	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.
g∪s	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.
f∩s∩g	4,6,12,25,32,33,35,37,39.
f∪s∪g	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.

Tablo 9. En başarılı öznitelik alt kümesi ile algoritma birleşimi ve analiz sonuçları
(Algorithm combination and analysis results with the most successful feature subset)

Öznitelik alt kümesi+ algoritma	Eğitim Süresi (s)	Test Süresi (s)	Doğruluk	Kesinlik	Duyarlılık	F-ölçütü
f∩g+ KNN	0,017	51,320	77,2%	96,4%	62,3%	75,7%
g∪s+ MLP	11,200	0,022	78,2%	95,1%	65,0%	77,2%
f∩g + RF	0,937	0,181	76,7%	96,7%	61,2%	75,0%
f∪g + XGBoost	4,158	0,018	80,3%	96,7%	67,7%	79,7%
f∪s∪g + LightGBM	0,554	0,038	80,0%	96,7%	67,3%	79,3%

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.

4.5. 5.senaryo: Ön İşlemlerden Geçmiş Veri Setinden Öznitelik Seçimi ve Tüm Ön İşlemlerin Yapıldığı Veri Setleri Üzerinde Makine Öğrenimi Algoritmalarının Uygulanması

(Scenario 5: Feature Selection from the Preprocessed Dataset and Application of Machine Learning Algorithms on All Preprocessed Datasets)

5. senaryonun uyglanmasında, Birini Dışarıda Bırak kategorik veri kodlama tekniği ve min-max ölçeklendirme ön işlemleri uygulanmış NSL-KDD veri setinden MI, GA ve XGBoost öznitelik seçim yöntemleri ile öznitelik seçimi ön işlemi yapılmıştır. Elde edilen öznitelik alt kümeleri Tablo 10'da gösterildiği gibidir. Tablo 10'a bakıldığında filtreleme, sarmal ve gömülü öznitelik seçimi yöntemlerinin ortak olarak seçtiği öznitelikler (f∩s∩g); 5 (src_bytes), (temel öznitelikler), 10 (Hot) (içerik özniteliği), 25 (serror rate, 30 (srv_rerror_rate), (Zaman Bağlı Trafik Öznitelikleri), 34 (dst_host_same_srv_rate), 37 (dst_host_srv_diff_host_rate), 40 (dst_host_rerror_rate), (Bağlantı Tabanlı Trafik Öznitelikleri). En çok sayıda öznitelik, bağlantı tabanlı trafik öznitelikleri kategorisinden seçilmiştir. Üç öznitelik seçim yönteminin de seçmediği öznitelikler ise: 7 (land), 9 (urgent), 11 (num_failed_logins), 13(num_compromised), 14 (root_shell), 15 (su_attempted), 17 (num_file_creations), 19 (num_access_files), 22 (is_guest_login) 'dir ve bu öznitelikler, temel öznitelikler ve içerik öznitelikleri kategorilerine aittir. Ön işlem uygulanmamış ve ön işlem uygulanmış veri setlerinden öznitelik seçiminin yapıldığı 4. ve 5.senaryolar ile farklı öznitelikler seçilmiş olmasına rağmen 7 (land), 9 (urgent), 22 (is_guest_login), 19 (num_access_files), özniteliklerinin her iki senaryoda da üç farklı öznitelik seçim yöntemi tarafından da seçilmemiş olmaması dikkat çekicidir.

Tüm ön işlemlerden geçmiş 11 farklı veri seti ve KNN, MLP, RF, XGBoost, LightGBM makine öğrenimi algoritmaları ile oluşturulan 55 adet modelin değerlendirilmesi ile en başarılı algoritma ve öznitelik alt kümesi birlikteliğinin yer aldığı Tablo 11'deki sonuçlara ulaşılmıştır.

Beşinci senaryo sonunda, algoritmaların en başarılı modelleri kurdukları veri setlerinin hibrit öznitelik seçimi yönteminin uygulanması ile elde edilen, KNN için f∪g, MLP için f∪s∪g, RF için f∩g, XGBoost için f∩g, LightGBM için ise g öznitelik alt kümelerine sahip veri setleri olduğu görülmüştür. En başarılı tüm öznitelik alt kümelerinde, gömülü yöntem ile seçilen özniteliklerin oluşturduğu, g kümesinin bulunması dikkat çekicidir. Ön işlem uygulanmayan veri setleriyle oluşturulan modellere kıyasla eğitim veri setleri üzerindeki saldırı tespit hızında KNN, 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. F-ölçütünde ise 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.

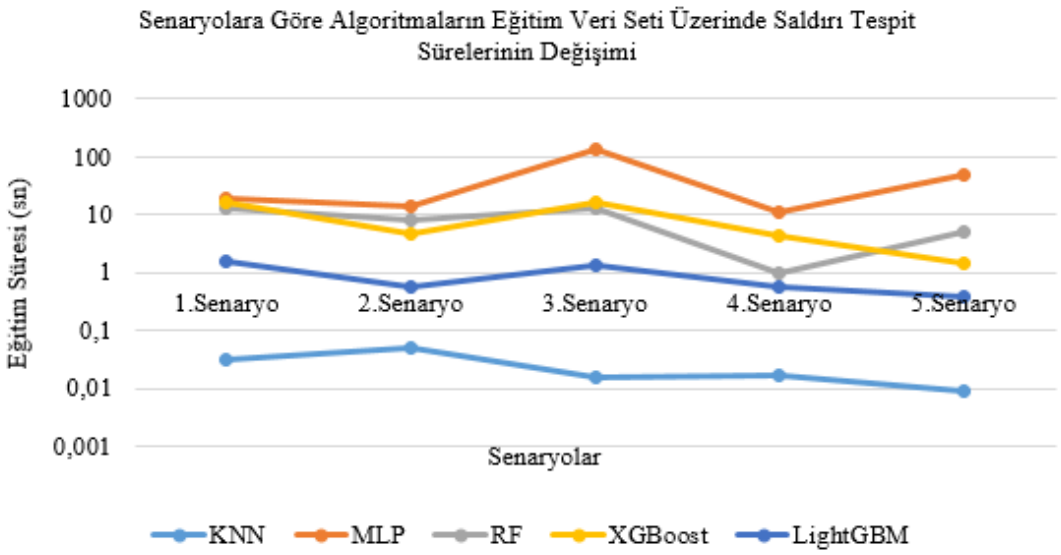
Bütün senaryoların sonuçları karşılaştırıldığında: Eğitim veri seti üzerindeki saldırı tespit sürelerinde en yüksek azalış, dolayısıyla en büyük hız artışı sırasıyla XGBoost:14,596 s; RF:7,431 s; LightGBM:1,209 s; KNN:0,022 s şeklinde sıralanırken, MLP'de tespit süresi 31,524 s artarak tespit hızı düşmüştür (Şekil 4).

Tablo 10. Ön işlemlerden geçmiş veri setinden elde edilen öznitelik alt kümeleri
(Attribute subsets obtained from the preprocessed dataset)

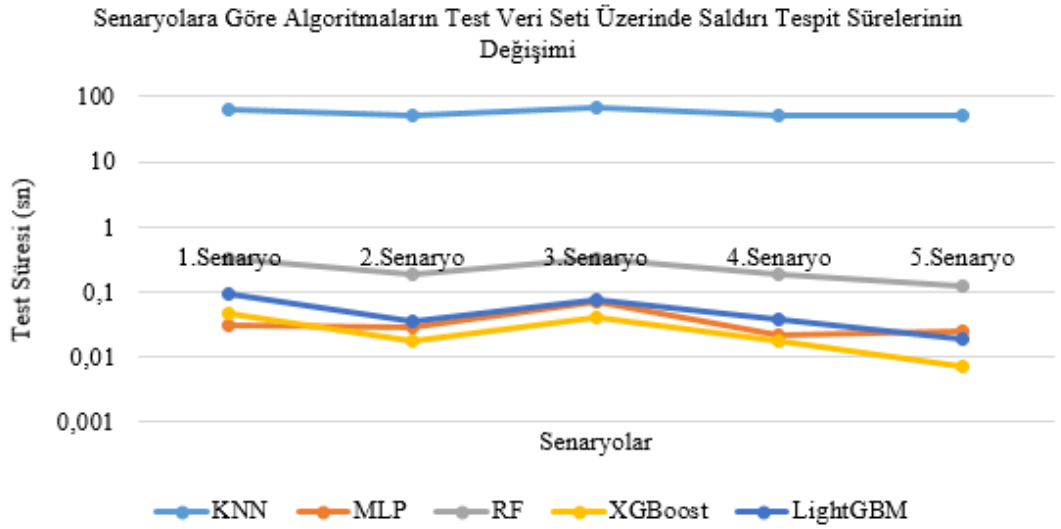
Küme	Öznitelik numarası
f	1,2,3,4,5,6,8,10,12,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41
s	5,8,10,16,18,20,21,25,27,30,34,37,40,41
g	2,3,4,5,6,10,23,24,25,26,28,30,33,34,36,37,38,40
f∩s	5,8,10,25,27,30,34,37,40,41
f∩g	2,3,4,5,6,10,23,24,25,26,28,30,33,34,36,37,38,40,
g∩s	5,10,25,30,34,37,40
f∪s	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
f∪g	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
g∪s	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
f∩s∩g	5,10,25,30,34,37,40
f∪s∪g	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

Tablo 11. Tüm ön işlemlerden geçmiş veri seti ile en başarılı öznitelik alt kümesi ve algoritma birleşimi
(The most successful feature subset and algorithm combination with all preprocessed dataset)

Küme	Eğitim Süresi (s)	Test Süresi (s)	Doğruluk	Kesinlik	Duyarlılık	F-ölçütü
f∪g + KNN	0,009	49,899	80,5%	96,6%	68,2%	79,9%
f∪s∪g + MLP	50,199	0,025	81,6%	92,2%	74,0%	82,1%
f∩g + RF	5,100	0,123	99,5%	99,2%	100,0%	99,6%
f∩g + XGBoost	1,414	0,007	97,9%	96,5%	100,0%	98,2%
g + LightGBM	0,373	0,019	95,6%	96,7%	95,5%	96,1%



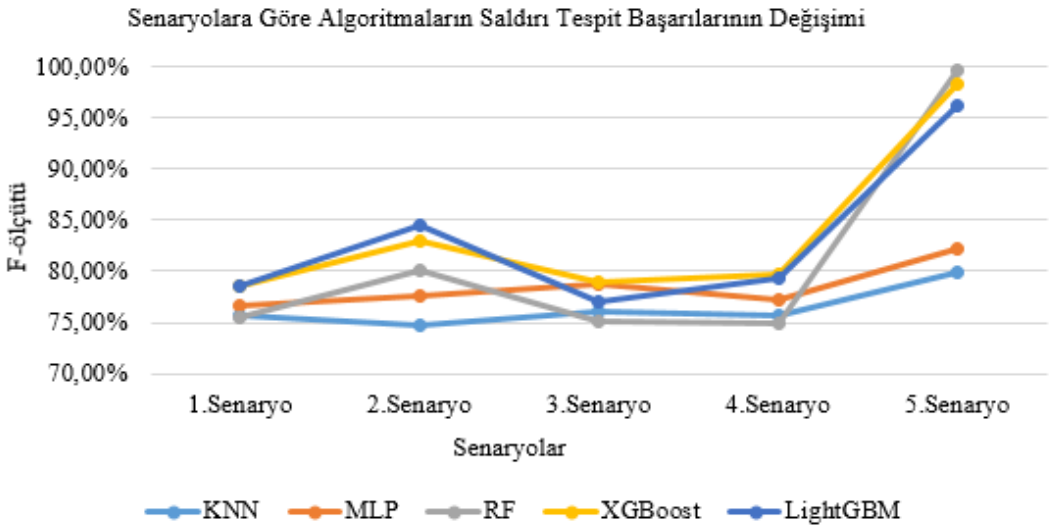
Şekil 4. Eğitim veri seti üzerinde saldırı tespit sürelerinin bütün senaryoların uygulanma sonuçlarına göre kıyaslanması
(Comparison of the attack detection times on the training dataset according to the results of the implementation of all scenarios)



Şekil 5. Test veri seti üzerinde saldırı tespit sürelerinin bütün senaryoların uygulanma sonuçlarına göre kıyaslanması
(Comparison of the attack detection times on the test dataset according to the results of the implementation of all scenarios)

Bütün senaryoların sonuçları karşılaştırıldığında: Test veri seti üzerindeki saldırı tespit sürelerinde 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 5.senaryonun uygulanması sonucu 0,007 s ile XGBoost algoritmasının fng öznitelik alt kümesini kullanarak kurduğu model ile ulaşılmıştır (Şekil 5). Şekil 6'da görüldüğü gibi algoritmaların saldırı tespit başarıları için baz alınan F-ölçütü'nün değerinde senaryoların uygulanması süresince artan bir değişim olmuştur. En büyük oranda artış, veri setine tüm ön işlemlerin uygulandığı 5.senaryoda gerçekleşmiştir. Bütün senaryoların uygulanması sonunda alınan sonuçlara göre, F-ölçütündeki değişim karşılaştırıldığında en yüksek artış sırasıyla, RF:%24,1; XGboost:%19,7; LightGBM:%17,6; MLP: %5,5; KNN:%4,2 şeklinde sıralanabilir. 5. ve son senaryonun uygulanması sonucu en yüksek saldırı tespit değerleri, f∩g öznitelik alt kümesi ve RF algoritması ile oluşturulan modelde %96,6, f∩g öznitelik alt kümesi ve XGBoost algoritması ile oluşturulan modelde %98,2, g öznitelik alt kümesi ve LightGBM algoritması ile oluşturulan modelde %96,1'dir.

Çalışmada, veri setine uygulanan ön işlem sırası ve sayısı değiştirilerek, 5 farklı senaryo ile uygulanan metodolojinin başarısının, çalışma zamanları ve F-ölçütü açısından değerlendirilmesi için Şekil 3-4-5 bakıldığında: Ön işlenmemiş veri seti ile kurulan modellere kıyasla (1.senaryo), modellerin çalışma hızında ve F-ölçütünde en büyük orandaki artışlar, 5.senaryonun uygulanması ile gerçekleşmiştir. 5.senaryoda kategorik veri kodlama, ölçeklendirme ön işlemleri uygulanan veri setinden, hibrit öznitelik seçimi yapılmıştı. Bu da gösteriyor ki 4. Senaryoda hedeflenen başarıya ulaşılamamasının sebebi, ön işlem uygulanmamış veri setinden, hibrit öznitelik seçimi yapılması ile en doğru özniteliklerin seçilememiş olmasıdır. Çalışmada uygulanan testler sonucunda, veri setine uygulanan ön işlemlerin sırasının ve birlikteliğinin, kurulan modellerin performansları açısından kritik öneme sahip olduğu görülmüştür.



Şekil 6. Saldırı tespit başarılarının (F-ölçütü) bütün senaryoların uygulanma sonuçlarına göre kıyaslanması
(Comparison of intrusion detection successes (F-criterion) against the implementation results of all scenarios)

Tablo 12. Random search yöntemi ile hiper-parametre optimizasyonu sonucu elde edilen parametreler
Parameters obtained as a result of hyper-parameter optimization with Random Search method)

Küme	Hiper-parametreler
f∪g + KNN	weights= 'distance', p= 2, n_neighbors=51, metric='euclidean', leaf_size= 51, algorithm= 'auto'
f∪s∪g + MLP	activation= 'logistic', alpha= 0.2201, batch_size= 256, hidden_layer_sizes=(89, 198), learning_rate='adaptive', momentum=0.8, solver='sgd', max_iter=1300, learning_rate_init= 0.36000000000000004
f∩g + RF	n_estimators= 70, min_samples_split= 5, min_samples_leaf= 2, max_features= 'auto', max_depth= 40
f∩g + XGBoost	n_estimators= 110, subsample=0.8, learning_rate= 0.05, min_child_weight= 2, max_depth=3, random_state= 5, reg_alpha= 0, reg_lambda= 1
g + LightGBM	feature_fraction= 0.8, boosting='dart',min_child_weight=5,max_depth=3, bagging_fraction= 0.6, num_leaves= 20, n_estimators=110, min_data_in_leaf= 30

Tablo 13. 5.senaryo sonunda elde edilen modellerde hiper-parametre optimizasyonu ile alınan sonuçlar
(Results obtained with hyper-parameter optimization in the models obtained at the end of the 5th scenario)

Küme	Eğitim Süresi (s)	Test Süresi (s)	Doğruluk	Kesinlik	Duyarlılık	F-ölçütü
f∪g + KNN	0,012	51,779	80,7%	96,6%	68,4%	80,1%
f∪s∪g + MLP	115,404	0,071	85,8%	92,1%	83,2%	87,4%
f∩g + RF	3,530	0,086	100%	100%	100,0%	100%
f∩g + XGBoost	1,207	0,005	100%	96,7%	100,0%	100%
g + LightGBM	1,059	0,024	100%	95,9%	100%	100%

4.6. Model İyileştirme: Random Search Yöntemi ile Hiper-parametre Optimizasyonu
(Model Optimization: Hyperparameter Optimization with Random Search Method)

5. senaryo sonucu elde edilen veri setleriyle ulaşılan en başarılı modellerde Random Search yöntemi kullanılarak Tablo 12'deki hiper-parametre kombinasyonlarına ulaşılmıştır.Tablo 12'de yer alan hiper-parametreler kullanılarak saldırı tespit modelleri tekrar eğitilip test edilmiş ve Tablo 13'te görülen sonuçlara ulaşılmıştır.

Tablo 13'te yer alan sonuçlar, veri setine tüm ön işlemlerin uygulandığı 5. senaryonun sonuçlarını içeren Tablo 10 ile karşılaştırıldığında; Eğitim veri seti üzerindeki saldırı tespit hızında KNN'de %33, MLP'de %130, LightGBM'de ise %184 oranlarında bir düşüş, RF'de %44.48, XGBoost'ta %14.6 oranlarında bir artış gözlemlenmiştir. Test veri seti üzerindeki saldırı tespit hızında ise KNN'de %3.8, MLP'de %184, LightGBM'de %26.3 oranlarında bir düşüş, RF'de %43, XGBoost'ta %40 oranlarında bir artış olmuştur. Saldırı tespit başarılarını değerlendirmek için F-ölçütüne baktığımızda KNN, MLP, RF, XGBoost ve LightGBM algoritmalarında sırasıyla %0.2, %5.3, %0.4, %1.8, %3.9 oranlarında bir artış gözlemlenmiştir. Hiper-parametre optimizasyonu yapılırken genel olarak saldırı tespit başarısının arttırılmasına karşılık eğitim ve test süresinin uzadığı görülürken, saldırı tespit başarısı ile hızını birlikte arttırabilmek için yeterli kaynak karşılanabildiği takdirde daha kapsamlı bir parametre araması yapılması gerektiği sonucuna varılmıştır.

Bu bölümde; veri setine uygulanan ön işlemler, farklı ön işlemler uygulanarak elde edilmiş veri setleri ve makine öğrenimi algoritmalarıyla saldırı tespit modellerinin oluşturulması, oluşturulan modellerin değerlendirilmesi, modellerin iyileştirilmesi olmak üzere dört aşamadan oluşan bir metodoloji, beş farklı senaryo kullanılarak ve son olarak da hiper-parametre optimizasyonu yapılarak uygulanmıştır. Veri setine ön işlemin uygulanmadığı birinci senaryonun uygulama sonuçları; kategorik veri kodlama, ölçeklendirme ve öznitelik seçimi ön işlemlerin ayrı ayrı uygulandığı ikinci, üçüncü ve dördüncü senaryoların uygulama sonuçları ile ve de ön işlemlerin birlikte uygulandığı beşinci senaryonun uygulama sonuçları ile kıyaslanmıştır. En başarılı sonuçlara beşinci senaryo ile ulaşılmıştır. Son olarak veri setine tüm ön işlemlerin uygulandığı 5.senaryo sonunda elde edilen en başarılı modellerde hiper-parametre optimizasyonu yapılarak modellerin performansları iyileştirilmiştir.

5. Sonuçlar (Conclusions)

Bu çalışmada, veri setine uygulanan ön işlemlerin, makine öğrenimi yöntemiyle geliştirilen saldırı tespit modellerinin performansı üzerindeki etkisi incelenerek, yüksek hızda ve oranda tespit yapabilen saldırı tespit modeli geliştirilmesi hedeflenmiştir. Yapılan bu çalışma ile birlikte; veri setine uygulanan ön işlemler, ön işlemler uygulanarak elde edilen veri setleri ile makine öğrenimi algoritmaları kullanılarak saldırı tespit modellerinin oluşturulması, oluşturulan modellerin değerlendirilmesi, modellerin iyileştirilmesi olmak üzere dört aşamadan oluşan bir metodoloji önerilmiştir. Metodolojinin 1.aşamasında, veri setine ön işlem uygulanmadan ve kategorik veri kodlama, ölçeklendirme, öznitelik seçimi ön işlemleri ayrı ayrı ve de birlikte uygulanarak 5 farklı senaryo için 5 farklı veri seti oluşturulmuştur. Metodolojinin 2.aşamasında, 1.aşamada oluşturulan veri setleri ve KNN, MLP, RF, XGBoost, LightGBM makine öğrenimi algoritmaları kullanılarak saldırı tespit modelleri oluşturulmuştur. Metodolojinin 3.aşamasında oluşturulan modellerin performansları eğitim süresi, test süresi, doğruluk, kesinlik, duyarlılık ve F-ölçütü metrikleri ile değerlendirilmiştir. Metodolojinin 4.aşamasında ise 3.aşamada elde edilen en başarılı modellerde Random Search yöntemi kullanılarak hiper-parametre optimizasyonu yapılmış ve modellerin performansları iyileştirilmiştir. Tüm senaryoların uygulanması ile alınan sonuçlar, saldırı tespit hızı ve başarısı birlikte dikkate alınarak değerlendirildiğinde; Tüm ön işlemlerin birlikte uygulandığı 5.senaryoda, 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 başarısı elde edilmiştir. Yine 5.senaryoda, hibrit öznitelik seçim yöntemi kullanılarak MI ve XGBoost öznitelik seçim yöntemlerinin ortak seçtiği öznitelikler ve XGBoost algoritması ile oluşturulan modelde Random Search yöntemi kullanılarak yapılan hiper-parametre optimizasyonu sonucu elde edilen parametreler ile söz konusu modelin, test veri seti üzerindeki tespit hızı 5.senaryo ile alınan sonuçlara kıyasla %28,6 oranında, saldırı tespit başarısı için baz alınan F-ölçütü ise %1,8 arttırılarak 0,005 s sürede %100 saldırı tespit başarısı elde edilmiştir. Sonraki çalışmalarda, saldırı tespit başarısı ile hızını arttırmak için veri setine uygulanan ön işlemlerin kullanılacak her bir algoritmaya ve veri setine uygun seçilmesi, çevrim içi ortamda değişken veri setleri söz konusu olduğunda, öznitelik seçim algoritmaları yetersiz kalacağı için çevrim içi öznitelik seçimi algoritmaları kullanılması, en doğru hiper-parametrelerin belirlenebilmesi için çok sayıda parametre kombinasyonunun denenmesine olanak sağlayacak kaynakların temin edilmesi, oluşturulacak saldırı tespit modellerinin başarısının, saldırı ve normal bağlantı kayıtları oranları gerçeğe daha yakın olan güncel ve farklı veri setleri ile daha güvenilir test edilerek başarısı birçok açıdan kanıtlanmış, güvenilirliği yüksek saldırı tespit modelleri geliştirilmesi, ayrıca son yıllarda oldukça başarılı olan derin öğrenme modellerinin kullanımı da hedeflenmektedir.

Kaynaklar (References)

1. We Are Social, Digital 2021July Global Statshot Report. https://wearesocial.com/blog/2021/07/digital-2021-i-dati-di-luglio/. Yayın tarihi Temmuz 23, 2021. Erişim tarihi Kasım 19, 2021.
2. Cybersecurity Venitures, 2021 Report: Cyberwarfare in the C-suit. https://cybersecurityventures.com/wp-content/uploads/2021/01/Cyberwarfare-2021-Report.pdf. Yayın tarihi Ocak 21, 2021. Erişim tarihi Kasım 19, 2021.
3. Sundaram A., An introduction to intrusion detection, XRDS, *2*, 3-7, 1996.
4. Cybersecurity Venitures, Cybercrime To Cost The World $10.5 Trillion Annually By 2025. https://cybersecurityventures.com/cybercrime-damages-6-trillion-by-2021/. Yayın tarihi Kasım 13, 2020. Erişim tarihi Kasım 3, 2021.
5. World Economic Forum, The Global Risks Report 2021. https://www3.weforum.org/docs/WEF_Global_Risk_Report_2020.pdf. Yayın tarihi Ocak 15, 2020. Erişim tarihi Kasım 25, 2021.
6. Gartner Identifies Three Factors Influencing Growth in Security Spending.https://www.gartner.com/en/newsroom/press-releases/2022-10-13-gartner-identifies-three-factors-influencing-growth-i. Yayın tarihi Ekim 13, 2022. Erişim Tarihi 07.03.2023.
7. Aslan Ö., Samet R., A Comprehensive Review on Malware Detection Approaches, IEEE Access, 8 (1-1), 6249-6271, 2020.
8. Samet R., Aslan Ö., Bölüm 8, Kötü Amaçlı Yazılımlar ve Analizi, Siber güvenlik ve savunma (Farkındalık ve Caydırıcılık), Baskı 1, Editörler: Sağıroğlu Ş., Alkan M., Grafiker Yayınları, Ankara-Türkiye, 225-251, 2018.
9. Bou-Harb E., Debbabi M., Assi C., Cyber Scanning: A Comprehensive Survey, in IEEE Communications Surveys & Tutorials, 16 (3), 1496-1519, 2014.
10. İlgün E., Veri Setine Uygulanan Ön İşlemlerin Anomali Tabanlı Saldırı Tespit Modellerinin Performansları Üzerindeki Etkisinin İncelenmesi, Yüksek Lisans Tezi, Ankara Üniversitesi, Sağlık Bilimleri Enstitüsü, Ankara, 2022.
11. Davis J.J., Clark A.J., Data preprocessing for anomaly based network intrusion detection: A review, Computers & Security, 30 (6-7), 353-375, 2011.
12. Naseer S., Saleem Y., Enhanced Network Intrusion Detection Using Deep Convolutional Neural Networks, KSII Trans. Internet Inf. Syst, 12 (10), 5159-5178, 2018.
13. Hancock J.T., Khoshgoftaar T.M., Survey on categorical data for neural networks, Journal of Big Data, 7, 1-41, 2020.
14. Tang C., Luktarhan N., Zhao Y., An Efficient Intrusion Detection Method Based on LightGBM and Autoencoder. Symmetry, 12 (9), 1458, 2020.
15. Aslan, Ö., Samet, R., Tanriöver, Ö.Ö, Using a Subtractive Center Behavioral Model to Detect Malware, Secur. Commun. Networks, 7501894, 1-17, 2020.
16. Mazini M., Shirazi B., Mahdavi I., 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, 2019.
17. Balakrishnan S.M., Venkatalakshmi K., Kannan A., Intrusion Detection System Using Feature Selection and Classification Technique, IJCSA, 3 (4), 145, 2014.
18. Torabi M., Udzir N.I., Abdullah M.T., Yaakob R.A., Review on Feature Selection and Ensemble Techniques for Intrusion Detection System, IJACSA, 12 (5), 538-553, 2021.
19. Özkan Okay M., Aslan Ö., Eryiğit R., Samet R., SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN, IEEE Access, 9, 157639-157653, 2021.
20. Ambusaidi M.A., He X., Tan Z., Nanda P., Lu L.F., Nagar U.T., A Novel Feature Selection Approach for Intrusion Detection Data Classification, 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications, Beijing-China, 82-89, 24-26 Eylül, 2014.
21. Chen C.W., Tsai Y.H., Chang F.R., Lin W.C., Ensemble feature selection in medical datasets: Combining filter, wrapper, and embedded feature selection results, Expert Systems, 37 (5), e12553, 2020.
22. Song J., Feature selection for intrusion detection system, Ph.D. Thesis, Aberystwyth University, Department of Computer Science Institute of Mathematics, Physics and Computer Science, Penglais-UK, 2016.
23. Kanimozhi V., Jacob P., Artificial Intelligence based Network Intrusion Detection with hyper-parameter optimization tuning on the realistic

cyber dataset CSE-CIC-IDS2018 using cloud computing, ICT Express, 5 (3), 211-214, 2019.
24. Latah M., Toker L., Towards an efficient anomaly-based intrusion detection for software-defined networks, IET Netw., 7, 453-459, 2018.
25. Uğurlu M., Doğru İ. A., Arslan R.S., Detection and classification of darknet traffic using machine learning methods, Journal of the Faculty of Engineering and Architecture of Gazi University, 38 (3), 1737-1746, 2023.
26. Özgür A., Erdem H., Feature selection and multiple classifier fusion using genetic algorithms in intrusion, detection systems, Journal of the Faculty of Engineering and Architecture of Gazi University, 33 (1), 75-87, 2018.
27. Protic D., Review of KDD Cup '99, NSL-KDD and Kyoto 2006+ datasets, Vojnotehnicki Glasnik, 66, 580-596, 2018.
28. Li W., Liu Z., A method of SVM with Normalization in Intrusion Detection. Procedia environmental sciences, 11, 256-262, 2011.
29. Yadav M.S., Kalpana R., Data Preprocessing for Intrusion Detection System Using Encoding and Normalization Approaches, 2019 11th International Conference on Advanced Computing (ICoAC), Chennai-India, 265-269, 18-20 Aralık, 2019.
30. Kasongo S.M., Sun Y., Performance Analysis of Intrusion Detection Systems Using a Feature Selection Method on the UNSW-NB15 Dataset, J Big Data, 7, 105, 2020.
31. Khare N., Devan P., Chowdhary C.L., Bhattacharya S., Singh G., Singh S, Yoon B., SMO-DNN: Spider Monkey Optimization and Deep Neural Network Hybrid Classifier Model for Intrusion Detection, Electronics, 9 (4), 692, 2020.
32. Hsu H., Hsieh C.W., Lu M., Hybrid feature selection by combining filters and wrappers, Expert Systems with Applications, 38 (7), 8144-8150, 2011.
33. Mackay D.J.C., Part 2, Chapter 8, Information Theory, Inference and Learning Algorithms, 4 nd Ed, Cambridge University, 139, 2003.
34. Liu H., Zhou M., Liu Q., An embedded feature selection method for imbalanced data classification, in IEEE/CAA Journal of Automatica Sinica, 6 (3), 703-715, 2019.
35. Mccall J., Genetic algorithms for modelling and optimisation, Journal of Computational and Applied Mathematics, 184 (1), 205-222, 2005.
36. Bergstra J., Bengio Y., Random Search for Hyper-Parameter Optimization, Journal of Machine Learning Research, 13 (1), 281-305, 2012.
37. Kartal E., Ozen Z., Dengesiz Veri Setlerinde Sınıflandırma, Baskı 1, Bölüm 8, Editörler: Torkul O., Gülseçen S., Uyaroğlu Y., Çağıl G., Uçar M.K., Sakarya Üniversitesi Kütüphanesi Yayınevi, Sakarya-Türkiye, 109-131, 2017.
38. İlgün E., Samet R., Ön İşlemlerin Makine Öğrenmesi Yöntemi ile Geliştirilen Saldırı Tespit Modellerinin Performansları Üzerindeki Etkisi, 7. Uluslararası Erciyes Bilimsel Araştırmalar Kongresi, Kayseri-Türkiye, 48-58, 9-10 Mart, 2022.