T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM TANIMLI AĞLARDA MAKİNE ÖĞRENME TEMELLİ SALDIRI TESPİT SİSTEMİ

YÜKSEK LİSANS TEZİ

Birol EMEKLİ

Bilişim Sistemleri Mühendisliği Anabilim Dalı

OCAK 2024

T.C.
SAKARYA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YAZILIM TANIMLI AĞLARDA MAKİNE ÖĞRENME TEMELLİ SALDIRI TESPİT SİSTEMİ

YÜKSEK LİSANS TEZİ

Birol EMEKLİ

Bilişim Sistemleri Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. İhsan Hakan SELVİ

OCAK 2024

Birol EMEKLİ tarafından hazırlanan "Yazılım Tanımlı Ağlarda Makine Öğrenme Temelli Saldırı Tespit Sistemi" adlı tez çalışması 22.01.2024 tarihinde aşağıdaki jüri tarafından oy birliği/oy çokluğu ile Sakarya Üniversitesi Fen Bilimleri Enstitüsü Bilişim Sistemleri Mühendisliği Anabilim Dalı'nda Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi

Jüri Başkanı :	**Prof. Dr. Numan ÇELEBİ** Sakarya Üniversitesi	.....
Jüri Üyesi :	**Prof. Dr. Orhan ER** Sakarya Üniversitesi	.....
Jüri Üyesi :	**Doç. Dr. İhsan Hakan SELVİ** Sakarya Üniversitesi	.....

ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Sakarya Üniversitesi Fen Bilimleri Enstitüsü Lisansüstü Eğitim-Öğretim Yönetmeliğine ve Yükseköğretim Kurumları Bilimsel Araştırma ve Yayın Etiği Yönergesine uygun olarak hazırlamış olduğum "YAZILIM TANIMLI AĞLARDA MAKİNE ÖĞRENME TEMELLİ SALDIRI TESPİT SİSTEMİ" başlıklı tezin bana ait, özgün bir çalışma olduğunu; çalışmamın tüm aşamalarında yukarıda belirtilen yönetmelik ve yönergeye uygun davrandığımı, tezin içerdiği yenilik ve sonuçları başka bir yerden almadığımı, tezde kullandığım eserleri usulüne göre kaynak olarak gösterdiğimi, bu tezi başka bir bilim kuruluna akademik amaç ve unvan almak amacıyla vermediğimi ve 20.04.2016 tarihli Resmi Gazete'de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince Sakarya Üniversitesi'nin abonesi olduğu intihal yazılım programı kullanılarak Enstitü tarafından belirlenmiş ölçütlere uygun rapor alındığını, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun ortaya çıkması halinde doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi beyan ederim. (22/01/2024).

Birol EMEKLİ

Beni seven herkese…

TEŞEKKÜR

Yüksel lisans eğitimim boyunca, bilgi ve deneyimlerini benden esirgemeyen, bu tez çalışmasında da yardımlarını esirgemeyen danışman hocam Doç. Dr. İhsan Hakan SELVİ'ye teşekkür ederim.

Tez çalışması kapsamında moral ve motivasyon vererek desteklerini esirgemeyen eşim Gözde ALTIN EMEKLİ ve kardeşim Mehmet EMEKLİ'ye teşekkür ederim. Bu süreçte bana gerek motivasyon gerekse desteğini esirgemeyen tüm aileme, arkadaşlarıma şükranlarımı sunuyorum.

Birol EMEKLİ

İÇİNDEKİLER

Sayfa

KISALTMALAR

ACK : Acknowledge
API : Application Programing Interface
AUC : Area Under Curve
CNN : Convolutional Neural Network
CSV : Comma Separated Values
DDoS : Distributed Denial of Service
DoS : Denial of Service
FN : False Negative
FP : False Pozitive
GRU : Gated Recurrent Unit
IP : Internet Protocol
KNN : K-Nearest Neighbors
LR : Logistic Regression
LSTM : Long Short Term Memory
ONOS : Open Networking Operating System
OVS : Open vSwitch
PCA : Principal Component Analysis
PCAP : Packet Capture
R2L : Remote to Local
RF : Random Forest
RNN : Recurrent Neural Networks
ROC : Receiver Operating Characteristic
STS : Saldırı Tespit Sistemi
SYN : Synchronize
TCP : Transmission Control Protocol
TN : True Negative
TP : True Pozitive
U2R : User to Root
YTA : Yazılım Tanımlı Ağ

TABLO LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

YAZILIM TANIMLI AĞLARDA DERİN ÖĞRENME TEMELLİ SALDIRI TESPİT SİSTEMLERİ

ÖZET

Küresel bir ağ olan interneti her geçen gün daha çok insan bağlanıp kullanmaktadır. Artan kullanıcı sayısı ve uygulamalar ağ güvenliği açısından riskleri de içerlnde barındırmaktadır. Her ne kadar güvenlik duvarları izinsiz erişimleri engellese de özel ağ içerisindeki bir servis internet dünyasına açıldığında hem zararlı hem zararsız kullanıcılar tarafından ulaşılabilir olmaktadır. Burada güvenlik duvarları tarafından verilen izinlerden kaynaklanabilecek saldırıları tespit etmede saldırı tespit sistemleri başarı ile kullanılmaktadır. Ayrıca saldırı tespit sistemleri, artan yapay zeka uygulamaları ile birlikte matematiksel ve istatistiksel yöntemler kullanılarak, mevcut verilerden çıkarımlar yapan, bilinmeyenlere dair tahminlerde bulunan, makine öğrenmesi algoritmaları kullanılarak daha başarılı sonuçlar vermektedir.

Bu çalışmada 2020 yılında Yazılım Tanımlı Ağlar için oluşturulmuş olan InSDN veri seti kullanılmıştır. Veri seti içerisinde web atak, DoS, DDoS, bilgi toplama, botnet, kullanıcı hesabının yönetici hesabına yükseltilmesi, kaba kuvvet saldırısı olmak üzere 7 farklı saldırı tipi ve normal trafik verileri bulunmaktadır. Kullanılan veri setini daha anlamlı hale getirmek için veri temizleme, veri dönüştürme, veri normalleştirme ve öznitelik seçim yöntemleri uygulanmıştır.

Veri seti üzerindeki çalışmalarda Python ve kütüphaneleri kullanılmıştır. Veriyi daha anlamlı hale getirmek için Pandas, Numpy, Seaborn ve Matplotlib kütüphanelerinden faydalanılmıştır. Saldırı tespiti için kullanılan makine öğrenme algoritmalarında Sklearn ve derin öğretme algoritmalarında ise Keras kütüphanesiden yararlanılmıştır.

Deneysel çalışmalar için veri işleme aşamalarından öznitelik seçimi için Ki-Kare, Spearmen, Karşılıklı bilgi, Kendall, Anova, Temel bileşen analizi ve sarmal yöntemlerden elde edilen özellikler Lojistik Regresyon, Karar Ağaçları, K En Yakın Komşu, AdaBoost, XGBoost, Rastgele Orman, Evrişimli Sinir Ağları, Tekrarlayan Sinir Ağları, Kısa Uzun Vadeli Bellek olmak üzere dokuz farklı makine öğrenmesi algoritması ile eğitilmiştir.

Deneysel çalışma sırasında kullanılan özellik sayısı azaldıkça derin öğrenme algoritmalarının başarım oranının düştüğü gözlemlenmiştir. Farklı öznitelik seçim yöntemlerindeki özelliklerin kullanılarak eğitildiği bu çalışma da XGBoost algoritması başarım oranı ve diğer metrikler incelendiğinde en iyi performans elde edilen algoritma olduğu görülmüştür. Çalışma sonrasındaki deneysel çıktılar XGBoost algoritmasının InSDN veri seti üzerinde önerilen diğer yöntemlerden daha iyi başarım oranı elde ettiğini göstermektedir.

MACHINE LEARNING BASED INTRUSION DETECTION SYSTEM IN SOFTWARE DEFINED NETWORKS

SUMMARY

The global network, known as the internet, is being accessed and utilized by an increasing number of individuals every day. The growing user base and applications bring along certain risks in terms of network security. While security firewalls prevent unauthorized access, when a service within a private network is exposed to the internet, it becomes accessible to both malicious and benign users. In detecting potential threats arising from permissions granted by security firewalls, intrusion detection systems are successfully employed. Furthermore, intrusion detection systems, in conjunction with the rising applications of artificial intelligence, utilize mathematical and statistical methods. By making inferences from existing data, making predictions about unknowns, and employing machine learning algorithms, they yield more successful results.

Software-defined networks consist of two independent components: the transmission of network traffic and the control mechanisms, which are managed at different layers. Software-defined networking allows for more manageable and extensible control of network traffic. It comprises infrastructure, control, and application layers.

In this study, the InSDN dataset created for Software-Defined Networks in the year 2020 was utilized. The dataset encompasses 7 different types of attacks, including web attacks, DoS, DDoS, information gathering, botnet activities, elevation of user account to administrator account, and brute force attacks, along with normal traffic data. There are a total of 275,515 attack records and 68,424 normal traffic records within the dataset. To enhance the meaningfulness of the dataset, data cleaning, data transformation, data normalization, and feature selection methods were applied.

The dataset comprises a total of 84 features. During the data cleaning stage, features with zero values for Fwd PSH Flags, Fwd URG Flags, CWE Flag Count, ECE Flag Cnt, Fwd Byts/b Avg, Fwd Pkts/b Avg, Fwd Blk Rate Avg, Bwd Byts/b Avg, Bwd Pkts/b Avg, Bwd Blk Rate Avg, and Fwd Seg Size Min were removed from the dataset. Additionally, variables such as Flow ID, Src IP, Dst IP, Timestamp, Src Port, Dst Port, and Protocol were excluded due to their variability, as their inclusion could lead to overfitting in the model.

During the data transformation stage, all the data has been labeled as either attack or normal. This study involves binary classification. The label encoder library of the Pandas library in the Python programming language has been employed for this purpose.

The performance rates of machine learning algorithms are dependent on the quality and values of the data. Therefore, data normalization has been performed to bring the values into a consistent format. The StandardScaler function from the Python sklearn library has been used for this purpose.

Feature selection methods have been applied to choose the minimum number of features that would yield the same result in the dataset. This method aims to select the features that contribute the most to the outcome. Feature selection methods such as Chi-Square, Spearman, Mutual Information, Kendall, ANOVA, Principal Component Analysis, and Recursive Feature Elimination have been employed. The feature output for each method is as follows: Chi-Square 39, Spearman 40, Mutual Information 39, Kendall 27, ANOVA 40, Principal Component Analysis 30, Recursive Forward Selection 50, and Recursive Backward Selection 59 features. The output of each feature selection method has been trained with different machine learning algorithms, and their performance rates have been compared.

The studies on the dataset have been conducted using Python and its libraries. To make the data more meaningful, the Pandas, NumPy, Seaborn, and Matplotlib libraries have been utilized. For machine learning algorithms used in intrusion detection, the Scikit-learn (Sklearn) library has been employed, and for deep learning algorithms, the Keras library has been utilized.

In the experimental study, the parameters of nine different machine learning algorithms were adjusted. These algorithms include Logistic Regression, Decision Trees, K-Nearest Neighbors, AdaBoost, XGBoost, Random Forest, Convolutional Neural Networks, Recurrent Neural Networks, and Long Short-Term Memory. To observe the models' performance, the dataset was divided into training and testing sets. After training the models, their performance was evaluated using the allocated test data.

The highest accuracy rates achieved by the algorithms on the test data, along with the feature selection methods used, are as follows: Logistic Regression, Sequential Forward Selection feature selection algorithm with achieved an accuracy of 0.9967. Decision Trees, Sequential Forward Selection feature selection algorithm, achieved an accuracy of 0.9999. K-Nearest Neighbors, Sequential Backward Selection feature selection algorithm and Chi-Square feature selection algorithm, achieved an accuracy of 0.9998. AdaBoost, Sequential Backward Selection feature selection algorithm, achieved an accuracy of 0.9990. XGBoost, Anova, Chi-Square, Mutual Information, Sequential Backward and Sequential Forward feature selection algorithm, achieved an accuracy of 0.9999. Random Forest, Chi-Square and Sequential Forward Selection feature selection algorithms, achieved an accuracy of 0.9999. Convolutional Neural Networks, Mutual Information feature selection algorithm, achieved an accuracy of 0.8743. Recurrent Neural Networks, Anova and Chi-Square feature selection algorithm, achieved an accuracy of 0.9935. Long Short-Term Memory, Mutual Information feature selection algorithm, achieved an accuracy of 0.9927. These accuracy rates were obtained in the study.

During the experimental study, it was observed that as the number of features used decreased, the performance of deep learning algorithms decreased. Convolutional Neural Networks exhibited the lowest performance among the algorithms in this dataset. Recurrent Neural Networks and Long Short-Term Memory achieved lower performance rates compared to machine learning algorithms.

Considering the time and performance values obtained after the study, it was observed that the Decision Trees algorithm yielded the highest performance with the least number of features when using the Kendall feature selection algorithm. The Kendall feature selection algorithm demonstrated that the same result could be achieved with

27 features. The Decision Trees algorithm, with 27 features, showed the highest performance rate of 0.9878, making it the most successful algorithm in this context.

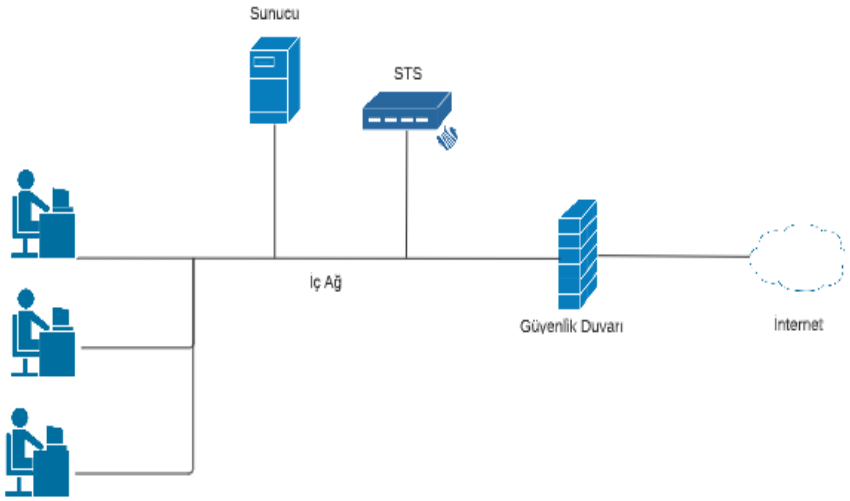
This study aims to create an intelligent system for software-defined networks, and in this context, the performance of various machine learning algorithms has been investigated. Despite variations in the size, content of the dataset, and parameter settings of the algorithms, it has been observed that deep learning algorithms generally achieve lower performance rates. On the contrary, machine learning algorithms tend to achieve higher performance rates. Specifically, for the design of an intrusion detection system in software-defined networks, the XGBoost algorithm has been identified as a potential choice due to its relatively higher performance.

1. GİRİŞ

1.1. Saldırı Tespit Sistemi

Saldırı Tespit Sistemleri(STS), bilgi sistemlerinde meydana gelen olayları görüntüleyerek güvenlik sorunu yaratabilecek olayları yorumlar , analiz eden ve aksiyonlar alabilen yazılım ya da donanım sistemlerine denilmektedir. Son zamanlarda bilgi sistemlerindeki artan saldırı sayıları ve risklerinin artması ile birlikte, STS'ler tüm bilgi güvenliği sistemlerinde tamamlayıcı bir bileşen olmaya başlamıştır (AYDIN, 2016).

Bilgi sistemlerine izinsiz erişimler, bir tehdit ve ya saldırı olarak nitelendirilmektedir ve güvenlik mekanizmalarından geçerek gizlilik, bütüklük ve erişilebilirlik için sağlanması gereken güvenli ortamı da riske atmaktadır. Web servisi üzerinden erişilebilen sistemlerde saldırılar sistem üzerinde oturum açma yetkisine sahip olan kullanıcılar ile daha yüksek yetkili halkara erişimi olan kullanıcılar tarafından da yapılabilirler. STS'ler sistemlere entegre olabilen, denetim izleri ile izleme, analiz etme, alarm üretme, aksiyon alma gibi modülleri ile geliştirilmiş olan ürünlerdir (AYDIN, 2016).



Şekil 1.1. Saldırı tespit sistemi.

Saldırı tespit sisteminin görevleri aşağıdaki gibidir.

- Tüm sistemin analiz edilmesi
- Saldırıları tespit edebilmek
- İz kayıtları üzerinden analizler ve anormallikleri tespit etmek
- Sistemlerdeki konfigürasyonlardaki açıklıkları inceleme
- Tüm bileşenleri bir bütün görüp değerlendirerek tüm kayıtları analiz etmek

STS'ler gelebilecek saldırılarla karşı sistemlerin kendisini koruyabilmesi için destekleyici sistemlerdir. Topladığı bilgileri yorumlar ve analiz eder. Eski verilerde benzer bir denetim izi ile karşılaşıp karşılaşmadığını ve bilgiler içerisinde farklı matematiksel formullerle analizler yaparak saldırı olup olmadığını tespit etmeye çalışır.

1.2. Saldırı Tespit Sistemi Tarihçesi

Saldırı tespit ifadesi Anderson tarafındından 1980 yılında Computer Security Threat Monitoring and Survaillance makalesi içerisinde ilk olarak bahsedilmiştir (Schultz, 2000). STS bu çalışma sonrasında daha çok duyulmaya ve araştırmalara dahil olmaya başlamıştır.

STS'ler ilk ortaya çıktıklarında basit bilgi sistemleri için düşünülmüştür. Daha sonraki gelişmelerde ise STS'ler denetim izi için büyük önem arz etmeye başlamıştır. Bununla birlikte veri tabanı kavramının bilgi güvenliğindeki önemli bir bileşen olduğu görülmüştür. Daha sonraki çalışmalarda güvenlik alanındaki uygulamalarda denetim izlerninin otomatik sistemlerle elde edilecek sistemler üzerinde çalışılmıştır. 1985'deki çalışmalarda birlikte denetim verileri üzerindeki çalışmalar özenle geliştirilmiş ve matematiksel yaklaşımlarla saldırıların tespit edilmesi hedeflenmiştir.

STS'ler bilgi sistemleri için yeni nesil bir güvenlik önlemi sunmaktadır. STS'lerin temel amacı, sisteme zarar verebilecek saldırı girişimlerinin tespit ve teşhis edilmesi ile ilgilenirler. Günümüzde artan bilgi sistemleri ve bu sistemlere girişimde bulunulan saldırıların hızlı bir şekilde çoğalması ile birlikte STS'leri kullanım oranı önem kazanmaya başlamıştır (Levitt, 1994).

1.3. STS ile Güvenlik Duvarı Arasındaki Farklar

Bilgi sistemlerinde genel bir bakış atıldığında yaygın olarak güvenlik duvarlarının kullanıldığı görüntülenir. Güvenlik duvarı temelde, birbirinden bağımsız ama iletişimde bulunan ağlar yerleştirilen ve ağlar arasında belirlenen bazı politikalar çerçevesinden ağların birbirinden izole edilmesini sağlamaktadır. Güvenlik duvarları tek başına tam güvenlilir bir çözüm oluşturamazlar. Özel bir ağda bulunan bir web servisine gelen istekleri bu güvenlik duvarları geçirmek zorundadır. Bu nedenle saldırganlar tarafından web servisinin terhic edildiği zayıflıklarda güvenlik duvarları bir koruma sağlayamamış olurlar.

Güvenlik Duvarları bütünleşik sistemlerde ilk erişilen bileşendir. İlk iletişimin kurulmasında trafiğin geçip geçmemesi ile ilgilenirler. Sistem içerisine geçmiş bir trafikle ilgilenmezler. STS'leri iz kaynaklarından aldığı özellikleri çoklu inceleme yaparlar ve saldırılara karşı alarmları vardır. Şüpheli bir durum analiz ettiğinde durumu güvenlik birimlerine bildirir.

1.4. Saldırı Tespit Sistemi Çalışmaları

STS çalışmaları incelendiğinde;

- Hareketlerin davranışlarının çıkartılması
- Davranış hareketlerinin sınıflandırılması
- Veri depolama ve toplama
- Veri filtreleme
- Alarm üretme ve aksiyo alma

gibi farklı konularda araştırmalar olduğu görülmektedir. Bu tehditlerin oluşabilmesi için analiz edilmesi gereken veri büyüklüğü önemlidir.Karar verme aşamasında hızlı ve doru tespit edebilecek yöntemlere ihtiyaç duyulmaktadır. Tasarım yapılırken öğrenim ve tespit yöntemi için saldırı logları kullanılmaktadır. Uygulamalarda kullanılabilir bir log dosyası olması gerçekleştirilmesi planlanan istemin hızlı sonuç üretmesi açısından önemlidir.

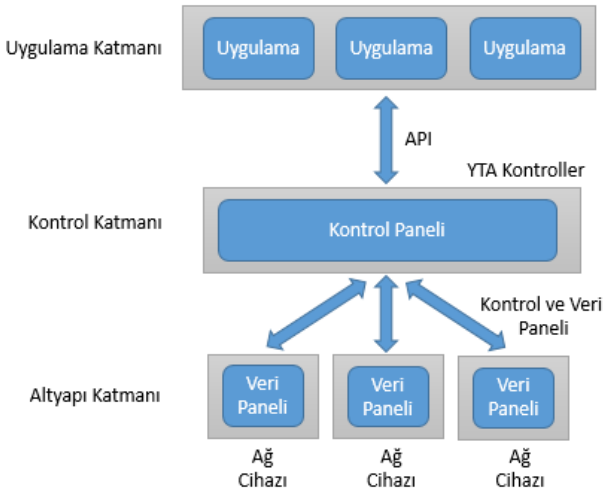
Mevcut çalışmalarda önceden oluşturulmuş veri setlerinin halen kullanımda olduğu görülmektedir. DARPA'nın 1998 ve 1999 yıllarında STS'lerin başarısını değerlendirmesini yaptığı çalışma da özel olarak kullandığı KDD'99 veri seti halen

araştırmacılar tarafından tercih edilen veri kümeleridir fakat yazılım tanımlı ağlar için bu veri seti uygun değildir.

1.5. Yazılım Tanımlı Ağ

Yazılım tanımlı ağ trafiği iletme ve kontrol mekanizmasının bağımsız olarak yönetilebilmesini sağlayan yeni bir ağ mimarisidir. YTA ile geleneksel ağ yapısındaki kısıtlı sınırlamaları daha genişletmiş ve yönetilebilir olmasına imkân tanımayı amaçlamaktadır. YTA tasarımı gereği trafiği kontroller mekanizması ile daha önceden tanımlanmış algoritmalara göre altyapı cihazlarına ileten yapıya sahiptir. Bu işlemi gerçekleştirirken kontroller yakın ve ya uzak bir sunucuda çalışabilirler. İletim mekanizması ile kontroller arasında güvenli bir bağlantı sağlanarak trafiğin istenildiği şekilde ulaştırılmasına olanak sağlamaktadır (ZAVRAK & İSKEFİYELİ, 2016). YTA bant genişliği yönetimi, güvenlik, hizmet kalitesi, çok noktaya yayın, yönlendirme gibi hizmetleri API aracılığı ile yönetilmesine imkân sağlar (Pathan, 2014).

YTA altyapı katmanı, kontrol katmanı ve uygulama katmanı olarak üç farklı fonksiyonel katman mimarisi içermektedir.



Şekil 1.2. Yazılım tanımlı ağ yapısı.

Altyapı Katmanı: YTA'daki en alt katmanıdır. Geleneksel ağ mimarisindeki gibi fiziksel ve ya sanal ağ cihazlarından oluşmaktadır (Xia et al., 2015). Bu katmandaki cihazlar sadece yönlendirme işlemi yapmaktadır.

Kontrol Katmanı: YTA'nın merkezi ve önemli yapısıdır. Ağ trafiğini kontrol eden akıllı bir sistem üzerine inşa edilmiştir (Selmic et al., 2016). Bu katmanın asıl görevi altyapı katmanı ile uygulama katmanı arasındaki iletişimin sağlanabilmesi için köprü görevi görmektir. YTA sadece bir kontrol katmanı tarafından kontrol edilmesi gerektiği bazı çalışmalarda söylenmektedir (Selmic et al., 2016).

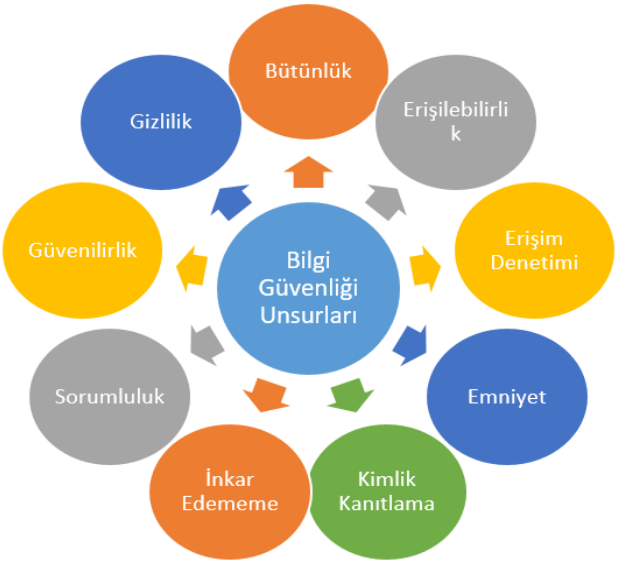
Uygulama Katmanı: Altyapı katmanında bulunan ağ cihazlarının kontrol edilebilmesi için tasarlanmış, kullanıcı tarafından erişilebilen yapıdır. Bu katman aracılığıyla ağın durumu ve özel gereksinimleri ile ilgili bilgilere ulaşılabilmektedir. Bu katmandaki YTA kontrol düzlemi modelin birincil bileşeni olarak kabul edilir (Cui et al., 2016).

2. SALDIRI TESPİT SİSTEMLERİ VE MAKİNE ÖĞRENMESİ

2.1. Ağ ve Bilgi Güvenliği

Bilgisayar güvenliğinde sistemler tehditlerden korunmadığı sürece güvenlik öneminden söz edilemez. Gizlilik, bütünlük ve erişilebilirlik bilgi sistemlerindeki güvenlik önlemlerinin başlıca yapı taşlarıdır.

- Gizlilik: Kullanıcıların erişmemesi gereken bilgiyi engelleme yöntemidir
- Bütünlük: Sistemin toplu olarak korunması gerektiği bileşendir
- Erişilebilirlik: Bilgi sistemlerinin işleyişlerine devam edebilmesidir. Yetkili kişiler tarafından sistemlere erişilebilmesidir (AYDIN, 2016).



Şekil 2.1. Bilgi güvenliği unsurları.

2.1.1. Tehdit nedir

Bir sisteme zarar verebilecek her türlü işlem tehdit olarak adlandırılabilir. Bu tehditler sistem üzerinde zarar verme, veri toplama, veriler üzerinde yapılabilecek değişiklikler şeklinde gerçekleştirilebilir. Bilgi sistemlerinde tehditler bilgisayar saldırganları, zararlı yazılımlar, doğal afetler gibi sayılabilir.

2.1.2. Zayıflık nedir

Bilgisayar sistemleri üzerindeki zafiyetlerden güvenlik açıkları oluşur. Bu açıkların sistemlerde var olmasına ise zayıflık denilir. Diğer tüm sistemlerde zayıflıklar içerebilir. Bunlardan bazıları güvenlik süreçlerinde, kontrol ve yapılandırmalarda ya da yönetimlerdei hassasiyetlerden kaynaklanabilir. Tehditler ve zayıflıklar birbirlerine çok yakın olmalarına ragmen aynı değildirler. Tehdit var olan zayıflığın veya zayıflıkların sömürülmesidir. STS'ler entegre oldukların sistemlerde tehdit ve zayıflıkları tespit ederek onlara karşılık verebilen yapılardır.

2.2. Bilgi Güvenliği Süreçleri

Ağ ve bilgi güvenliği kamsapmında, tasarlanacak olan güvenlik sistemlerinin temel bileşenleri hazırlanırken doğru bir şekilde tüm güvenlik ilkelerinin kapsam içerisinde dahil edilmek gerekir. Bu süreçler genellikle sistemler üzerindeki güvenlik açıklarının kapatılarak saldırganlar tarafından gelebilecek tehditlerin önceden öngörerek risklerin minimilize edilmesini hedeflemektedir.

Kapsam dahilinde her türlü bilgi ve tehdit kendi kapsamında değerlendirilerek güvenlik önleme sistemi olarak maliyetlerde göz önüne alınması gerekmektedir. Sistemlerin %100 güvenli olmayacağı düşünülecek olursa, güvenliğin ideal yapılandırılmasında şu üç süreç dikkati çekmektedir.

2.2.1. Önleme

Bilgisayar sistemine yönelik, yapılabilecek her saldırı ve tehdite karşı, sistemdeki açıklıkları güvenilir şekle getirmek için alınmış olan her türlü aksiyon önleme faaliyetine denir. Örneğin kişisel bilgisayar güvenliğimizde, açılışta şifre girilmesi, en güncel işletim sistemi ve programların kullanılması, internetten indirilen dosyaların kullanılmadan önce virus programları tarafından taranması bazı önleme faaliyetleridir. Alınan her önlem güvenliğin tam olarak sağlandığından söz edilemez.Bu nedenle farklı tedbirlerede başvurulması gerekmektedir.

2.2.2. Saptama

Saptamada temel amaç sistemin bütün durumunun analiz edilerek sistemdeki tüm faaliyetleri izleyerek kayıt altında tutmaktır. Bu kayıtlar kanıt olarak sunulabilir ayrıca bu kayıtlardan gelebilecek benzer saldırılara karşı tedbir almak amaçlı kullanılabilir.

Güvenlik duvarı, saldırı tespit sistemleri, virus programları, ağ trafiği izleyiciler saptama aşamasıda kullanılan en temel araçlardır.

2.2.3. Karşı koyma

Bu aşama da önleme adımındaki zafiyetlerden dolayı durdurulamayan ve saptama adımında tespit edilerek saldırganlar tarafından girişimde bulunulan atağa cevap verebilecek aksiyonlar alınmasıdır. Saptama sürecinde kullanılan STS, tespit ettiği faaliyetlere karşı aksiyon alabilecek fonksiyonları ile daha çok işlevsellik kazanır. STS'lerin saldırı engelleme gibi bir fonksiyonu olmadığı için karşı koyma aşaması ağ ve bilgi güvenliğinde önemli bir adımdır.

Şekil 2.2. Ağ ve bilgi güvenliği adımları.

Tüm bu süreçler Şekil 2.2.'de gözüktüğü gibi karşılık verme aşamasında bir saldırı engellenmese bile sistemin tekrar çalışır hale getirilmesi, saldırılar hakkında bilgi toplama ve güvenlik zafiyetlerinin giderilmesi için 3 aşama önem arz etmektedir.

2.3. Ağ Üzerinden Yapılabilecek Saldırı Türleri

2.3.1. Bilgi toplama (Probe)

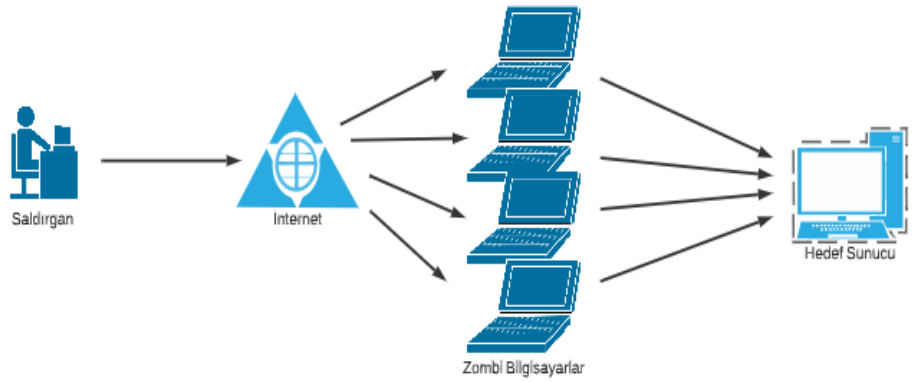
Bilgi toplama saldırı türü hedef alınan bir ya da daha fazla sistemin erişim bilgilerini tespit etmek, topolojideki konumunu belirlemek, sistem üzerindeki kullanıcı ve yetkilerinin tespiti, üzerinde çalışan servislerin versiyon bilgilerini tespit etmek ve buradaki zafiyetlerden yararlanmak gibi bilgileri elde edebilek için yapılır.

- Bir port grubuna tarama (ipsweep)
- Hedef alınan sistemdeki servislerin tespiti yapabilmek için tüm portlara tarama başlatmak (portsweep,nmap)

2.3.2. Hizmet engelleme (DoS)

TCP/IP protocol yapısından faydalanmak isteyen saldırganlar hedef aldıkları sistemlere fazla sayıda istekler göndererek sunucunun erişelemez hale getirilmesi için seçtkleri saldırı tipidir. Bu saldırı türünde saldırganlar farklı noktalardan hedef aldıkları sistemlere kaldırabileceğinden fazla işlem göndererek, üzerindeki hizmetleri çalışamaz hale getirmek ve saldırılar sırasında sistemin açıklıklarını bularak içeriye sızabilmeyi hedeflerler. Hedef sistemdei kaynakların tüketilmesi, ağ trafiğinde yoğunluk yaratarak sistemlerin birbirleri arasında iletişim sorunu yaratmak, yaratılan bu yük fazlalığından ötürü siztemlerdeki zayıflıklardan yararlanmak DoS saldırısı için örnek olarak söylenebilir. Bu saldırı tipini normal trafikten ayırt etmek çok zor olduğu için en tehlikeli saldırı yöntemlerinden bir tanesidir. Anomali tespitli STS'ler bu tarz tespiti zor saldırı tipleri tespitinde sıklıkla kullanılmaktadır.

Saldırganlar DOS ile sistemlerin iletişimlerini kısıtlar ve hizmet alabilecek tüm servisleri engellemek için çeşitli saldırı tipleri uygularlar. Sistemdeki var olan zafiyetlerden faydalanmak için ağı derinmelesine analiz ederler ve ağ yolu ile DoS saldırıları gerçekleştirirler. DoS saldırıları iki çeşittir. Savunmasızlık yöntemini tercih eden saldırganlar, hedef seçtikleri sistemdeki uygulama ve ya ağ protokolleri üzerindeki güvenlik açıklarını kullanarak sistem üzerinde yük yaratmak ve erişilemez duruma getirilmesini hedefler. Bu tarz saldırılar genellikle kaynak tüketimin de artış meydana getirir ve sistemi yavaşlatma ve ya yeniden başlatmaya zorlarlar. Bir diğer yöntem olan taşkın yönteminde hedef sisteme daha fazla iş yükü gönderirler ve sistemin kaynaklarını tüketerek hizmetlerin aksatılmasına sebep olunur.

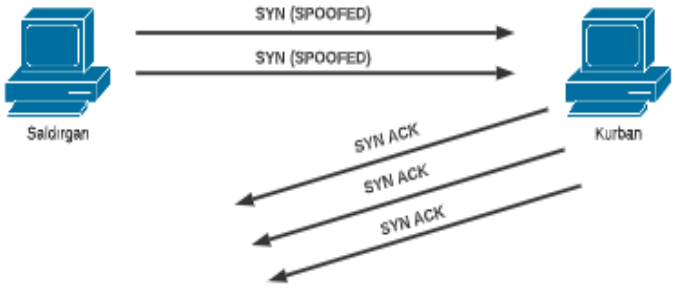


Şekil 2.3. DDoS saldırısı.

DOS saldırıları gerçekleştirilirken, farklı kaynaklardan yapılmasına da Dağıtılmış Hizmet Reddi(DDoS) denilmektedir. Şekil 2.3'de görüldüğü gibi, saldırganlar tarafından ele geçirdikleri farklı bilgisayarları zombi olarak kullanırlar. Uzaktan kontrol edebildikleri bu bilgisayarları hedef aldıkları sistemlere saldırmak için kullanırlar. DoS saldırılar hedef sistem üzerindeki açıklardan faydalanma ve sistemleri etkisiz hale getirmeyi amaçlar. DoS saldırı çeşitlerini aşaıdaki gibi açıklayabiliriz.

2.3.2.1. SYN flood

Bilgisayar sistemi TCP/IP "stack" adı verilen bir bölgede tüm bağlantı bilgilerini tutar. TCP bağlantı protokolü üçlü el sıkışma adımlarını tamamlanması gerekir. Bağlantı isteğini ilk başlatan taraf bağlantı kuracağı sisteme SYN paketi gönderir ve bağlantı kurulacak taraf isteği kabul etmek için SYN/ACK gönderir. Daha sonra ilk talep eden karşı taraftan aldığı SYN/ACK paketine ACK paketi göndererek üçlü el sıkışma adımlarını tamamlamış olur. TCP bu çalışma yapısı kötü amaçlar için kullanılabilmektedir. SYN paketi gönderen kişi paket içerisindeki source ip kısmına kendi IP adresini değilde varolmayan herhangi IP adresini ekleyerek gönderir. Bu değişiklik sonrasında hedef sistem SYN içeriği gönderilen paketi alır ve üçlü el sıkışma işlemi için SYN/ACK paketini gönderir. Hedef sistem ACK mesajını var olmayan bir IP adresinden beklediği için cevap gelene kadar bu varolmayan IP adresini stack bölgesinde tutmaya devam eder. Doğal olarak bu saldırı türü tekrarlanmaya devam ederse varolmayan IP adreslerinden bir ACK mesajı gelmeyeceğinden bir süre sonra stack bölgesi taşar ve ağ işlemez hale gelir. Yani sonuç olarak sunucu saldırganlar tarafından gönderilen SYN isteklerine SYN/ACK mesajı gönderemez hale gelir ve bağlantı kurulamaz.



Şekil 2.4. SYN flood saldırısı.

2.3.2.2. Ping taşma saldırısı

Ping taşma saldırı hedef sistemlerin bant geniliğini işgal ederek iç ağdaki iletişimin erişilemez olması için gerçekleştirilir. Saldırganlar ICMP paketindeki IP tespit etme yöntemi için kullanılan paket boyutunu değiştirerek istekte bulunurlar. Hedef sistemler gelen büyük paketleri işlemek isterler ve çok fazla gelmesi durumunda sistem üzerinde kaynak tüketiminin artmasına ve çökme durumuna kadar gidebilirler.

2.3.2.3. Kaba kuvvet saldırısı (brute force)

Bu saldırı türü hedef aldığı bilgisayar sistemlerini bir çok veri göndererek meşgül ederler. Saldırganlar bu tip girişimlerde hedef adresin yayın adresine paketler gönderirler. Bilgisayar sistemleri içerisinde paket trafiğinin iletilmesi için sistemler arasında ICMP istek paketleri yoğunluğu artacaktır. Oluşan bu yoğunluk ile ağ trafiğindeki bant genişliği kullanımı arttığı için iç ağdaki haberleşme gecikmelerine ve ya erişilememesine neden olacaktır.

2.3.3. Uzaktan yerel oturum açma (R2L)

Saldırganların sistemler üzerinde bir kullanıcısı olmadan sisteme paket gönderdiği saldırı türüdür. Bunlara örnek olarak:

- FTP atak
- İmap atağı
- Basit şifreler kullanılması ve tespiti ile sisteme girilmesi (Guest)

2.3.4. Kullanıcı hesabinin yönetici hesabina yükseltilmesi (U2R)

Bu saldırı tipinde sistem üzerinde kullanıcı bulunan fakat kısıtlı yetkilere sahip kullanıcılar hedef alınır. Bu kullanıcıların daha yetkili kullanıcıların yapabileceği yetkileri alabilmek için çaba gösterildiği saldırı türüdür. Örnek olarak şunlar verilebilir.

- Sosyal mühendislik saldırıları
- Rootkit yükleme saldırıları
- Buffer owerflow atakları
- Veri tabanını bulunun bir sistemdeki kullanıcıların belirli komutlar kullanılarak yönetici hakkında sahip olması
- Sistemlerdeki local kullanıcılar root yüksek kullanıcılı hesaplara yükseltilmesi

2.4. Saldırı Tespit Sistemin Kullanılan Yöntemler

STS'ler iz kanıtları üzerinden hareketlerin anormal ya da kötüye kullanımını tespit edebilmek üzere iki yöntem kullanırlar. Anomalinin tespit edilmesi için iz kanıtlarındadaki hareketler bir model oluşturulur. Kötüye kullanımın tespitinde ise saldırı tipindeki hareketler kullanılır (Budak et al., 2013). Gelen bir saldının kynağının tespit edilememesi durumunda engellemek mümkün olmayabilir. STS iz kanıtları üzerinden bu bilgiye ulaşmak için çalışırlar. Topladığı iz kanıtları üzerinden derinlemesine yararlanma sağlayarak en hızlı şekilde tespit etmek üzerine inşa edilmiştir. STS tespit yöntemine göre farklı iki grupta incelenir. STS'ler hangi amaçı kullanılacağı öncesinden belirlenmelidir. Anomali temelli STS'ler tüm normal olmayan davranışların tespitinde kullanılır. Kötü kullanım STS'ler ise daha çok bilinen davranışların tespitinde kullanılır. Bu yöntemlerde her ikisi de kendi içerisinde farklı öne çıkan özelliklere sahiptir. Daha başarılı sonuçlar elde edebilmek için her iki yaklaşım modelini barındıran hibrit sistemlerin oluşturulması daha faydalı ve akılcı bir sistem olacağını düşündürmektedir (Budak et al., 2013).

Şekil 2.5. Saldırı tespit sistemi tipleri.

STS'ler kullanılan dşğer gvenlik cihazlarındaki bulunan kurallarda tespit edilememiş ve ya daha önceden bilinmeyen bir saldırı olup olmadığı tespit etmeye çalışan daha akılcı bir üründür.

Saldırı tespiti için geçmişten beri istatistiksel yöntemler kullanılmıştır fakat bunlardan farklı olarak durum geçiş diyagramları (state transition diagrams), yapay sinir ağları (artificial neural networks), veri madenciliği (data mining), yapay bağışıklık sistemi (artificial immune system), örüntü eşleme, bulanık mantık (fuzzy logic) farklı birçok yöntemin de uygulandığı görülmüştür (Budak et al., 2013).

2.4.1. Sistemdeki konumuna göre sts

Sistemdeki iz kanıtlarının görüntülenmesinde ve davranışların analiz edilebilmesi için kullanılırlar. Bileşen ve ağ temelli olmak üzere iki gruba ayrılır.

2.4.1.1. Ağ temelli sts

Ağ temelli STS'ler adından da anlaşılacağı üzere ağ katmanından topladığı verileri analiz eder ve yorumlarlar. Üç farklı birimden oluşur.

Veri toplama modülü iz kanıtlarının toplanabilmesi için ağ trafiği üzerinde dinlemeler gerçekleştirir. Toplanan iz kanıtları belirli bir formatta normalize edildikten sonra normal olmayan davranış hareketleri ile karşılaştırılır ve buradaki çıktı ile saldırı ayrımı yapılabilir.

2.4.1.2. Bileşen temelli sts

Sistemler üzerindeki hareketlerin ve değişkenliklerin görüntülenmesi için kullanılan bir STS modelidir. Bu modelde iz kanıtları üzerinden sistemlerdeki değişik davranış ve hareketleri izlerler. Bileşen temelli STS ile bağlantı kurulan her sistem bir iz kanıtı gönderir ve bu iz kanıtları üzerindeki değişik davranışların kontrolünü sağlarlar.

2.4.2. Saldırı algilama yöntemlerine göre sts

2.4.2.1. Imza tabanlı sts

Bu tarz STS'ler genellikle bir kuralı olan, tespitinin keskin çilgilerle ayrımının yapılabildiği, saldırı tipi ve yönteminin bilindiği sistemlerde tasarlanmıştır. Her saldırı için kendi içerisinde daha önceden belirtilmiş saldırı ya da normal bir hareket olup olmadığınıtespit edebilen kuralları vardır. Kuralların oluşturulabilmesi için tespit zamanına kadar gelen saldırıların hareketlerinden faydalanılır. Bu hareketlerin özelliklerinde protokol bilgi ve işaretleri, IP adresi bilgileri, paket içerisindeki veri ve port numarası vardır. Bu özellikler izmaların oluşturulabilmesi için yol göstericidir.

İzma tabanlı STS'ler zayıf yönlerinden birisi gelebilecek farklı ve yeni saldırı türlerinin tespit edilecek bir kabiliyetinin olmamasıdır. Yeni üretilmiş saldırı tipi daha önceden izmalı kurallar oluşturulması için belirlenmesi için hareketin saldırı olduğu algılayamaz. İzma tabanlı STS'lerdeki bu zayıflıklar ve ileriye yönelik tespitlerindeki eksikliklerin giderilebilmesi için anormalik tabanlı STS'ler geliştirilmiştir (Pathan, 2014).

2.4.2.2. Anormallik tabanlı sts

STS'lerde anormallik tespitinde ağ ya da sunucudaki genel davranışlarda ve iz kanıtlarından varlığından haberdar olur. İzlediği sistemlerdeki anormal hareketler normal davranışlarda farklılık gösterir ve bu sayede ayrım yapabilirler. STS'ler bir süredir izlediği sistemlerin davranış hareketlerini düzenli bir şekilde profil oluştururlar. Profillerin oluşturulabilmesi için bir süre zaman geçmesi ve izlere genel bir bakış açısı kazanması beklenir. Herhangi bir farklılık geldiği taktirde topladığı tüm verilerdeki değerlerden sapmanın olup olmadığını kontrol eder ve eğer bir sapma oldu ise anomali tespiti yaparlar. Saldırı içermeyen hareketlerin tespiti yapıldıktan sonra, davranışları normal kanıtlardan sapan her iz bir saldırı olduğu düşünülür. Herhangi bir sapma olduğunda ve anomali tespit edildiğimde ilgili yöneticileri uyarabilmek için alarm ve bildirim iletirler.

2.4.3. Veri işleme zamanina göre sts

Bir saldırı, zafiyet ve ya güvenlikle ilgili ihlal tespit edildikten sonra ilgili kişilere bildirim gönderilmesine kadar geçen süre veri işleme zamanı olarak adlandırılır. Gerçek ve gerçek zamanlı olmayan sistemler diye iki ayrılır.

2.4.3.1. Gerçek zamanlı sts

Herhangi bir saldırı tespit ettiği durumda saldırı tipine karşı aksiyon alan sistemlerdir. Gerçek zamanlı STS'ler üzerine günümüzde çalışmalar yapılmakta ve adına saldırı engelleme sistemi(IPS) denilmektedir.

2.4.3.2. Gerçek zamanlı olmayan sts

Saldırı tespit ettiğinde alarm üreten, sistem yöneticilerinde bildirim göndererek tespit ettiği olayı incelemek için detayları ileten sistemlerdir. Bu sistemler tehditleri kayıt ederler ve daha sonra incelemek üzere saklarlar.

2.5. STS'lerde Makine Öğrenmesi

Makine öğrenmesi, tespit etme ve karar verme aşamalarında matematiksel ve istatistiksel formüllerden faydalanır. Bu hesaplamalar için bilgisayarın işlem yapabilmek kabiliyetinden faydalanır. Sınıflandırma çalışmalarında bir çok makine öğrenme algoritmaları başarılı sonuçlar vermektedir.

Makine öğrenimi tabanlı sistemlerde, yeni verilere maruz kaldıklarında otomatik olarak değişen bir sistemi veya programı tanımlarız. Böylece program daha önce karşılaşmamış saldırıları tespit edebilmektedir.

Literatürdedeki çalışmalar incelendiğinde, STS'lerde en çok tercih edilen makine öğrenme algoritmalarının aşağıdaki algoritmalar olduğu görülmüştür.

- Lojistik Regresyon
- XGBoost
- Rastgele Orman
- AdaBoost
- Karar Ağaçları
- Evrişimli Sinir Ağları
- Tekrarlayan Sinir Ağları
- Uzun Kısa Vadeli Bellek

2.5.1. Lojistik regresyon (LR)

Bu algoritmadaki temel hedef, özellikleri verilmiş değerin hangi sınıf içerisinde olduğunu bulmayı amaçlamaktadır. Lojistik regresyon, bağımlı değişkenlerin değerlerinin matematiksel yöntem ile olasılığının hesaplandıktan sonra, uygun sınıflandırma yapabilen bir algoritmadır (LaValley, 2008). Veri içerisindeki en az değişkeni kullanarak doğru sonuca ulaşmayı hedeflemektedir.

2.5.2. XGBoost

XGBoost algoritması, aşırı öğrenmenin önüne geçen, yüksek tahmin gücü ve diğer algoritmalara göre daha hızlı çalışması ile son zamanlarda çok tercih edilen bir algoritmadır. 2016 yılında Chen ve Carlos tarafından kullanılan bu yöntemin diğer algoritmalardan daha iyi performans sergilediğini belirtmişlerdir (Agarwal et al., 1994). Veri kaybının en aza indirilmesi, boş olan değerler ile çalışabilmek, aşırı uyumu

azaltması, maliyetin azaltılması ve kaynakların optimizasyonunu yapabilen başarılı bir yöntemdir (Agarwal et al., 1994). Algoritmanın hızlı olması ve veri seti üzerinde yaptıkları ile birçok makine öğrenmesi yarışmalarında ilk sıralarda başarısını göstermektedir.

2.5.3. Rastgele orman (RF)

Breiman 2001 yılında öneride bulunduğu algoritmadır (Jin et al., 2020). Bu algoritmanın temelinde istenilen sayı kadar sınıflandırıcı ağaç oluşturarak içerisinden seçilen bir alt küme ile verilerin sınıflandırılmasını sağlayan algoritmadır (Jin et al., 2020). Adından da anlaşılacağı gibi rastgele torbalama yöntemi ile eğitilmiş alt küme ağaçları oluşturulur. Bu algoritma kategorik ve süreklilik içeren küçük ya da büyük veri setlerinde çalışabildiği için yaygın olarak kullanılan sınıflandırıcı bir yöntemdir.

2.5.4. AdaBoost

Freund ve Schapire 1996'da önerdikleri bir algoritmadır. Uygulaması basit ve sağlam temeller üzerine kurulmuş bir algoritmadır. Algoritma eğitim aşamasında veriler üzerindeki ağırlıkları koruyarak zayıf değerler dizisi oluşturur ve oluşturulan bu zayıf değerleri birleştirerek güçlü sonuçlar elde etmeyi hedefler. Başlangıçta tüm ağırlıklar eşit olarak ayarlanır ve bir sınıflandırma yapılır. Eğitim aşamasında zayıf değere sahip olan özellikler artırılmaya çalışılır ve böylece daha anlamlı değerlerle sınıflandırmadaki başarımın artırılması sağlanır (Wang et al., 2021).

2.5.5. Kara ağaçlari (DC)

Makine öğrenmesi, görüntülerin işlenmesi ve örüntülerin tanımlanması gibi birçok alanda kullanılmaktadır (Charbuty & Abdulazeez, 2021). Bu algoritma ağacın kökünden başlayarak yaprak düğümdeki veri ayrımına kadarki sürecin tanımlandığı bir tekniktir. Kategorik veya sayısal veri setleri için uygundur (Lee et al., 2022). Karar ağaçları girdi verilerindeki sınıflara bakarak daha küçük gruplara ayırarak, tüm düğümlerin bölünmesi ile anlamsız verilerin anlamlı verilerden ayrıştırılmasını sağlar (Lee et al., 2022).

2.5.6. Evrişimli sinir ağlari (CNN)

Evrişimli Sinir Ağları, veri setinden öznitelik çıkarılmasına imkân veren katmanlardan oluşan derin öğrenme algoritmasıdır (Yilmaz, 2021). Nesne algılama, görüntülerin

sınıflandırılması, veri üzerinden özellik çıkarımı gibi alanlarda sıklıkla kullanılmaktadır (Çekiç & Çavdar, 2023). Algoritma temelinde yapay sinir ağları barınmaktadır. İlk katmanlarında verilerden özellik çıkarımı gerçekleştirilir. Bu aşamadan sonra bir sonraki katmana iletmek ve performans artırımı için boyutun düşürülmesine yönelik yöntemler kullanılır. Tüm girdiler özellik seçim aşamasından geçtikten sonra bir boyutlu vektöre dönüştürülerek, tam bağlanmış katmanlara girdi olarak verilir ve verilerin sınıflandırılması sağlanır (Elmas, 2022).

2.5.7. Tekrarlayan sinir ağlari (RNN)

Tekrarlayan Sinir Ağları ileri beslemeli ağ türlerinden birisidir. Çalışma aşamasında başarımı artırmak için daha önceki sonuçları kısa süreliğine dahili belleğinde tutarak tekrar girdi olarak alır ve daha güçlü bir derin öğrenme yöntemi için çalışır (Cui, Z. , R. KE, n.d.). Model aktif olarak çalışırken t zamanındaki verilerle işlem yaparken t-1 zamanındaki verileri dahili belleğinde tutarak buradaki sonuçlarını da girdi olarak almaktadır.

2.5.8. Uzun kisa vadeli bellek (LSTM)

Uzun Kısa Vadeli Bellek algoritması 1997'de Hochreiter ve Schmidhuber tarafından ortaya atılmıştır. TSA yapısındaki eksikliklerin giderilmesi için geliştirilmiş bir algoritmadır (Hochreiter & Schmidhuber, 1997). TSA yapısında kısa süreli bellek aktarmaları sağlıklı bir şekilde yapılırken, daha uzun süreli bilgilerin aktarmasında sorunlar ortaya çıkmaktaydı. UKVB algoritması ile bilgilerin saklanması, güncellenmesi ve unutulması daha başarılı bir şekilde sağlandığı belirtilmiştir (Staudemeyer & Morris, 2019).

2.6. Özellik Seçim ve Azaltma Yöntemleri

Özellik seçimi ve azaltma yöntemleri, çok fazla özellik barındıran bir veri kümesi içerisinden aynı sonucu verecek en küçük özelliklerin barındırıldığı alt kümelerin matematiksel yöntemlerle seçilmesi olarak tanımlanmaktadır. Bu metot ile veri setinden daha çok bilgiye ulaşabilmek, anlaşılabilirliği artırmak, ölçeklenebilirlik ve doğruluğu geliştirmek için güzel bir yöntemdir. Temel olarak veri kümesi içerisinde özelliklerin sonuca en çok katkı sağlayanlarının seçilmesini amaçlar (Forman, 2000). Çok büyük boyutlu veri kümelerinde, daha az özellik ile aynı çıktıları veren alt

kümeleri bulma da sıklıkla tercih edilmektedir. Makine öğrenmesi algoritmalarında başarım oranını artımak, daha hızlı çıktılar üretmek için çok sık kullanılmaktadır.

2.6.1. Ki-kare (Chi-squared)

Bu testin asıl amacı, gözlenen ile beklenen frekansların istatistiksel yöntemlerle farklarının ne kadar anlamlı olduğunu ortaya çıkarmayı hedefler (BUDAK, 2018). Ki-kare testinde önce tüm özelliklerin sınıflara karşı istatistiklerini hesaplayarak başlar. Sonrasında hesaplanan değerin belirlenmiş önemlilik seviyesine olan tutarlılığına bakılarak özelliklerin ayrıştırılması sağlanır (Kavzoğlu et al., n.d.). Bu hesaplama sonucunda ki-kare değeri sıfır çıkarsa bağımsız değişken olarak değerlendirilir. Yüksek değerde olması ise değişkenin daha anlamlı olduğunu ifade etmektedir. Ki-kare formülü aşağıdaki gibi tanımlanmıştır.

2.6.2. Spearman

Veri setindeki tüm değişkenleri temsil edebilecek en yararlı özelliklerden altkümeler oluşturur. Bu işlem gerçekleştirilirken sonuca en yararlı değişkenler ve seçilen bu değişkenlerin birbiri ile ilişkilerinin düşük olması hedeflenir (Zhang & Zhao, 2008). Bu işlem ile özellik sayısını en aza indirerek veri setini en doğru şekilde temsil etmek amaçlanır.

2.6.3. Karşılıklı bilgi (Mutual info)

İki değişken arasındaki karşılıklı bilgiyi ölçer ve bu rasgele değişkenler sıfıra eşit olması durumunda düşük bağımlı değişken, daha yüksek bir sayıya eşit ise bağımlılığı yüksek değişkenler olarak değerlendirilir (Tanuj Joshi: Tiwari, 2019).

2.6.4. Kendall

Kendall sıra korelasyonu katsayısı iki değişkenin nitelik sıralamaları arasındaki benzerlik derecelerini ölçmek için kullanılır (Van Hulse et al., 2009). Katsayı değeri her bir değişkenin tüm değişkenler içerisindeki sırasındaki ters çevrilme sayısına bağlıdır. Bu işlemi gerçekleştirirken her değişken sırası için tüm nesne çiftlerinin kümesi ile temsil edilir (Abdi, 2008).

2.6.5. Anova

Birden fazla değişkenin arasındaki farkı test etmek için kullanılır. Öncelikle tüm özelliklerin nitelik sıralamaları bulunur ve sonrasında en yüksek nitelik sırasındaki değerler ile özellik alt kümesi oluşturulur (Ding et al., 2016). Bu özellik alt kümesi, veri setinden en az kayıpla çıktıya ulaşmayı sağlayan en anlamlı özellikleri içermektedir.

2.6.6. Temel bileşen analizi (PCA)

Temel bileşen analizi, tüm veri setindeki özelliklerin varyanslarını hesaplayarak en az veri kaybı ile aynı çıktıya ulaşabilecek yeni bir özellik kümesi oluşturur. Gereksiz verilerin ortadan kaldırılmasını amaçlayan bu yöntem ile makine öğrenmesi gibi çalışmalardaki hesaplamalarda daha kolay işlemler gerçekleştirilmektedir (Malhi & Gao, 2004).

2.6.7. Ardışık ileri yönde seçim (forward selection)

Ardışık İleri Yönde Seçim Algoritması, Whitney'in 1971 yılındaki çalışmasında özniteliklerin hepsi ile başarılı çıktılar elde edilemeyeceği ve bu soruna istinaden önerdiği öznitelik seçim yöntemlerinden birisidir (Whitney, 1971). Bu yöntemde seçim aşaması boş elemanlı küme olarak başlayıp, küme de bulunmayan tüm özellikleri kümeye ekleyerek ilerler. Kümeye dâhil edilen her özellik için sonuca etki eden katkısına bakılır. Çıktıya katkısı çok olan ve seçilen her özellik, kalıcı olarak kümede bulunur. Bu şekilde tüm özniteliklerin çıktıya olan katkısına bakılarak, en anlamlı özniteliklerin seçimi gerçekleştirilir (BUDAK, 2018).

2.6.8. Ardışık geri yönde seçim (backward selection)

Ardışık Geri Yönde Seçim Algoritması, 1963 yılında Maril ve Green'in çalışmasında ilk defa bahsedilmiştir (Lewis, 1963). Bu seçim algoritması ileri yönde seçim algoritmasının tersi şeklindedir. Başlangıç aşamasında kümede tüm öznitelikler bulundurularak başlanır. Bu aşama da çıktıya en iyi değer verecek öznitelik her aşamada kümeden çıkartılır ve bir defa kümeden çıkan öznitelik tekrar kümeye dâhil edilemez (Oral & Ozkan, 2013). Bu yöntemin sonucunda algoritmanın en iyi alt kümesi oluşturulur ve çıktıya katkısı yüksek olan öznitelikler ayrıştırılmış olur.

3. VERI SETI VE DENEYSEL ÇALIŞMA

Bu bölüm içerisinde çalışmada kullanılan veri seti üzerinde uygulanılan özellik seçim yöntemleri, veri seti içeriği hakkında bilgiler yer almaktadır. Özellik seçim yöntemlerinden sonra ortaya çıkan anlamlı veriler ile farklı makine öğrenmesi algoritmaların performanslarının değerlendirilmesi yer almaktadır.

3.1. Veri Seti

Bu çalışma YTA yönelik olduğu yazılım tanımlı ağlarda elde edilmiş InSDN verileri kullanılmıştır. Bu veri seti YTA için Mahmoud, Nhien ve Anca tarafından 2020 yılında oluşturulmuş yeni bir veri setidir (Elsayed et al., 2020). Bu veri setinin amacı YTA yönelik yapılan saldırıların tespiti için daha anlamlı veriler içermektedir. Daha önceki birçok çalışma içerisinde KDD 99 veri seti kullanılmaktadır fakat bu veri seti YTA için uygun özellikleri ve yapıyı içermemektedir. Veri seti içerisinde web atak, bilgi toplama, DoS, botnet, DDoS, kullanıcı hesabının yönetici hesabına yükseltilmesi, kaba kuvvet saldırı olmak üzere 7 farklı saldırı tipi ve normal trafik verilerini içermektedir. Bu saldırı tipleri bölüm 2.3'de açıklanmıştır.

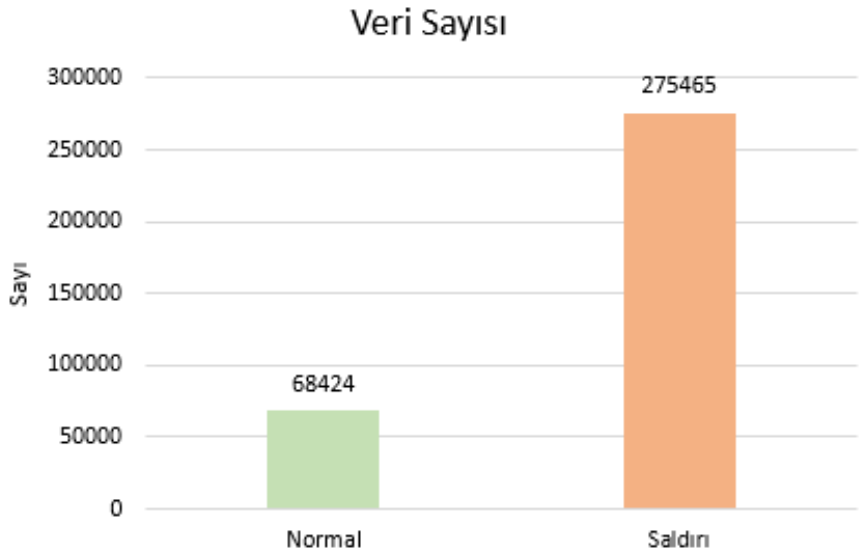
Tablo 3.1. InSDN veri seti içerisindeki farklı türdeki verilerin sayıları.

Veri Türü	Saldırı Türü	Sayısı
Normal	Normal	68424
Saldırı	Web Atak	192
	Bilgi Toplama	98129
	DoS	53616
	Botnet	164
	DDoS	121942
	Hesap Yükseltme	17
	Kaba Kuvvet	1405

Mahmoud ve arkadaşları buradaki eksikliği gidermek adına kurdukları labaratuvar ortamında bu veri setini elde etmişlerdir. Bu çalışmada dört farklı sanal makine kullanmışlardır. Saldırgan olarak Kali Linux işletim sistemi olan birinci makine, ikinci makinede ise Ubuntu üzerinde ONOS denetleyicisi bulunmaktadır.

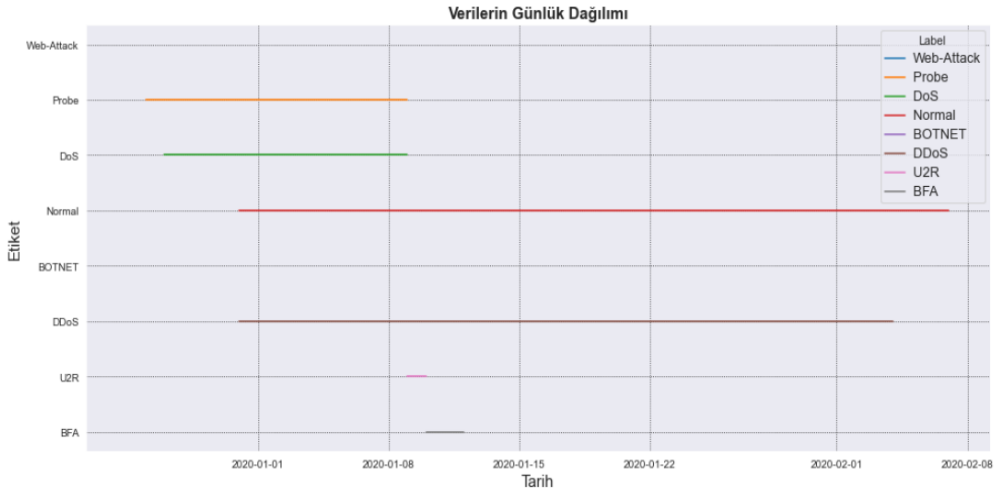
Üçüncü makine ise mininet ve OVS anahtarının çalıştığı Ubuntu işletim sistemi olan bir farklı makinedir. Dördüncü makine ise savunmasız olarak bırakılan ve yapılacak saldırıların hedefi olarak düşünülmüştür (Elsayed et al., 2020).

Veri setini Wireshark aracı ile PCAP formatında tüm logları yakalayarak hazırlamışlardır. Bu veri seti oluşturulurken Kanada Siber Güvenlik Enstitüsü ekibi tarafından geliştirilmiş olan CICFlowMeter kullanılmışlar. Bu uygulama Java tabanlı ve PCAP dosyalarından ağ akış trafiğini CSV dosyasına çıkartılmasını sağlamaktadır.



Şekil 3.1. Veri tipi sayısı.

Bu şekilde toplam 343 939 veri elde edilmiş ve bunların 275 515 saldırı özellikli, 68 424'ü ise normal trafiği içermektedir. Şekil 3.1'de veri sayılarının görselleştirilmiş hali ve Şekil 3.2'de ise verilerin oluşturulduğu tarihlere ait bir görselleştirme bulunmaktadır.



Şekil 3.2. Verilerin günlük dağılımı.

InSDN veri seti Table 3.1'de gösterildiği gibi toplam 7 farklı saldırı türü içermektedir. Bu veri setinde her bir veri için 84 öznitelik bulunmaktadır. Bu özellikleri açıklamaları EK A'da bulunmaktadır.

Tablo 3.2. InSDN veri seti öznitelikleri.

No.	Öznitelik İsmi	No.	Öznitelik İsmi	No.	Öznitelik İsmi
1	Flow ID	29	Fwd IAT Std	57	ECE Flag Cnt
2	Src IP	30	Fwd IAT Max	58	Down/Up Ratio
3	Src Port	31	Fwd IAT Min	59	Pkt Size Avg
4	Dst IP	32	Bwd IAT Tot	60	Fwd Seg Size Avg
5	Dst Port	33	Bwd IAT Mean	61	Bwd Seg Size Avg
6	Protocol	34	Bwd IAT Std	62	Fwd Byts/b Avg
7	Timestamp	35	Bwd IAT Max	63	Fwd Pkts/b Avg
8	Flow Duration	36	Bwd IAT Min	64	Fwd Blk Rate Avg
9	Tot Fwd Pkts	37	Fwd PSH Flags	65	Bwd Byts/b Avg
10	Tot Bwd Pkts	38	Bwd PSH Flags	66	Bwd Pkts/b Avg
11	TotLen Fwd Pkts	39	Fwd URG Flags	67	Bwd Blk Rate Avg
12	TotLen Bwd Pkts	40	Bwd URG Flags	68	Subflow Fwd Pkts
13	Fwd Pkt Len Max	41	Fwd Header Len	69	Subflow Fwd Byts
14	Fwd Pkt Len Min	42	Bwd Header Len	70	Subflow Bwd Pkts
15	Fwd Pkt Len Mean	43	Fwd Pkts/s	71	Subflow Bwd Byts
16	Fwd Pkt Len Std	44	Bwd Pkts/s	72	Init Fwd Win Byts
17	Bwd Pkt Len Max	45	Pkt Len Min	73	Init Bwd Win Byts
18	Bwd Pkt Len Min	46	Pkt Len Max	74	Fwd Act Data Pkts
19	Bwd Pkt Len Mean	47	Pkt Len Mean	75	Fwd Seg Size Min

Tablo 3.3. (Devamı) InSDN veri seti öznitelikleri.

No.	Öznitelik İsmi	No.	Öznitelik İsmi	No.	Öznitelik İsmi
20	Bwd Pkt Len Std	48	Pkt Len Std	76	Active Mean
21	Flow Byts/s	49	Pkt Len Var	77	Active Std
22	Flow Pkts/s	50	FIN Flag Cnt	78	Active Max
23	Flow IAT Mean	51	SYN Flag Cnt	79	Active Min
24	Flow IAT Std	52	RST Flag Cnt	80	Idle Mean
25	Flow IAT Max	53	PSH Flag Cnt	81	Idle Std
26	Flow IAT Min	54	ACK Flag Cnt	82	Idle Max
27	Fwd IAT Tot	55	URG Flag Cnt	83	Idle Min
28	Fwd IAT Mean	56	CWE Flag Count	84	Label

3.2. Veri İşleme

Bu adım içerisinde veriyi algoritmalardaki başarım oranını yükseltmek için bazı yöntemler uygulanacaktır.

3.2.1. Veri temizleme

Makine öğrenmesi ve derin öğrenme algoritmaların başarım oranlarını artırmak, maliyetin azaltılması, daha anlamlı hale getirilmesi için veri seti üzerinde bazı temizlik işlemleri yapılması gerekmektedir. Veri seti içerisinde anlamsız, birbirinden farklı, boş değerler içeren gereksiz öğreler temizlenmesi veri setini daha anlamlı hale getirecektir. Veri temizleme aşamasında başlıca şu adımlar uygulanabilir;

- Tekrar eden verilerin kaldırılması
- Veriler içerisindeki yazım ve diğer hataların temizlenmesi
- Veri setine uygun anlam içermeyen verilerin temizlenmesi
- Boş veri seti içeren sütunların temizlenmesi

Bu adım içerisinde InSDN veri kümesindeki tüm öznitelikler incelenmiştir. Öznitelikler içerisinde 11 özniteliğin tüm değerlerinin 0 olduğu görülmüştür. Bu özniteliklerin isimleri "Fwd PSH Flags, Fwd URG Flags, CWE Flag Count, ECE Flag Cnt, Fwd Byts/b Avg, Fwd Pkts/b Avg, Fwd Blk Rate Avg, Bwd Byts/b Avg, Bwd Pkts/b Avg, Bwd Blk Rate Avg, Fwd Seg Size Min"'dır. Bu özniteliklerin çıktıyı elde edebilmek için herhangi bir katkısı olmayacağından 11 öznitelik veri setinden kaldırılmıştır.

Veri setindeki “Flow ID, Src IP, Dst IP, Timestamp, Src Port, Dst Port, Protocol” 7 öznitelik dinamik değişkenler olduğundan ve modelin yanlış eğitilmesine yol açmaması için veri setinden kaldırılmıştır.

Veri seti üzerindeki veri temizleme işleminden sonra öznitelik sayısı 65 düşmüştür ve bu öznitelikler Tablo 3.3’de verilmiştir. İlerleyen bölümlerde yapılacak veri dönüştürme, veri normalleştirme ve öznitelik seçim yöntemleri bu 65 özellik ile yapılmıştır.

Tablo 3.3. Veri temizleme sonrası öznitelikler.

No.	Öznitelik İsmi	No.	Öznitelik İsmi	No.	Öznitelik İsmi
E1	ACK Flag Cnt	E23	Flow Duration	E45	Idle Std
E2	Active Max	E24	Flow IAT Max	E46	Init Bwd Win Byts
E3	Active Mean	E25	Flow IAT Mean	E47	Init Fwd Win Byts
E4	Active Min	E26	Flow IAT Min	E48	PSH Flag Cnt
E5	Active Std	E27	Flow IAT Std	E49	Pkt Len Max
E6	Bwd Header Len	E28	Flow Pkts/s	E50	Pkt Len Mean
E7	Bwd IAT Max	E29	Fwd Act Data Pkts	E51	Pkt Len Min
E8	Bwd IAT Mean	E30	Fwd Header Len	E52	Pkt Len Std
E9	Bwd IAT Min	E31	Fwd IAT Max	E53	Pkt Len Var
E10	Bwd IAT Std	E32	Fwd IAT Mean	E54	Pkt Size Avg
E11	Bwd IAT Tot	E33	Fwd IAT Min	E55	RST Flag Cnt
E12	Bwd PSH Flags	E34	Fwd IAT Std	E56	SYN Flag Cnt
E13	Bwd Pkt Len Max	E35	Fwd IAT Tot	E57	Subflow Bwd Byts
E14	Bwd Pkt Len Mean	E36	Fwd Pkt Len Max	E58	Subflow Bwd Pkts
E15	Bwd Pkt Len Min	E37	Fwd Pkt Len Mean	E59	Subflow Fwd Byts
E16	Bwd Pkt Len Std	E38	Fwd Pkt Len Min	E60	Subflow Fwd Pkts
E17	Bwd Pkts/s	E39	Fwd Pkt Len Std	E61	Tot Bwd Pkts
E18	Bwd Seg Size Avg	E40	Fwd Pkts/s	E62	Tot Fwd Pkts
E19	Bwd URG Flags	E41	Fwd Seg Size Avg	E63	TotLen Bwd Pkts
E20	Down/Up Ratio	E42	Idle Max	E64	TotLen Fwd Pkts
E21	FIN Flag Cnt	E43	Idle Mean	E65	URG Flag Cnt
E22	Flow Byts/s	E44	Idle Min		

3.2.2. Veri dönüştürme

Veri setindeki anlamsız değerler ve öznitelikler temizlendikten sonra verileri normalize etme süreci başlatılmıştır. Ilk süreçte 1 normal etiket ve 7 saldırı etiketi

olmak üzere toplam 8 farklı sonuç vardır. Bu çalışma da sadece "Normal" ya da "Saldırı" olduğu tespit edileceği için, tüm etiketlerin normal ve saldırı olarak değiştirilmesi yapılmıştır. Veri sayıları Şekil 3.1'de gösterilmiştir.

Veri seti içerisindeki "Label" dışındaki tüm öznitelikler sayısal değerlere sahiptir. Kullanılacak bazı algoritmaların kategorik değişkenleri desteklese de karşılaştırmaların yapılabilmesi için label özniteliğinin de sayısal değere dönüştürülmesi yapılmıştır. Bu işlem Python programlama dilinde Pandas kütüphanesinin label encoder fonksiyonu ile gerçekleştirilmiştir.

Tablo 3.4. InSDN veri kümesi label özniteliği sayısal değeri.

Veri Türü	Değeri
Normal	0
Saldırı	1

3.2.3. Veri normalleştirme

Bu adımda veri dönüştürme işleminden sonra, verilerin arasındaki farklılıkların azaltılması ve verilerin tek bir düzen haline getirilmesi ele alınmıştır. Bu aşamada Python sklearn kütüphanesi standardscaler fonksiyonundan yararlanılmıştır. Makine öğrenme algoritmalarının başarısı verinin kalitesine bağlıdır. Veri normalleştrme, her özniteliğe eşit katkı sağlayacak şekilde ölçeklendirildiği bir veri ön işleme yaklaşımıdır (Singh & Singh, 2020).

Standardscaler, elde edilen dağılımın ortalama değerinin sıfır ve standart sapma olacak şekilde veri seti içerisinde dönüştürmesidir. Bu değer orjinal değerden ortalama değerin çıkartılması ve ardından standart sapmaya bölümünden elde edilmektedir. Şekil 3.3'deki formül ile hesaplanır.

$$z = \frac{x - \mu}{\sigma}$$

Şekil 3.3. Standardscaler formül.

Şekil 3.3'deki formülde x orjinal değer, μ ortalama ve σ standart sapmadır.

3.2.4. Öznitelik seçimi

Bu adımda Bölüm 2.6'daki bahsedilen özellik seçim yöntemleri uygulanmıştır. Table 3.5'de özellik seçim yöntemlerinin aynı çıktıyı elde edebilmek için kullanılmasını önerdiği öznitelik sayıları verilmiştir. Ayrıca Şekil 3.5'de ise Tablo 3.3'de verilen Veri Setindeki Öznitelikler içerisinden hangi özniteliklerin seçildiği görselleştirilerek verilmiştir.

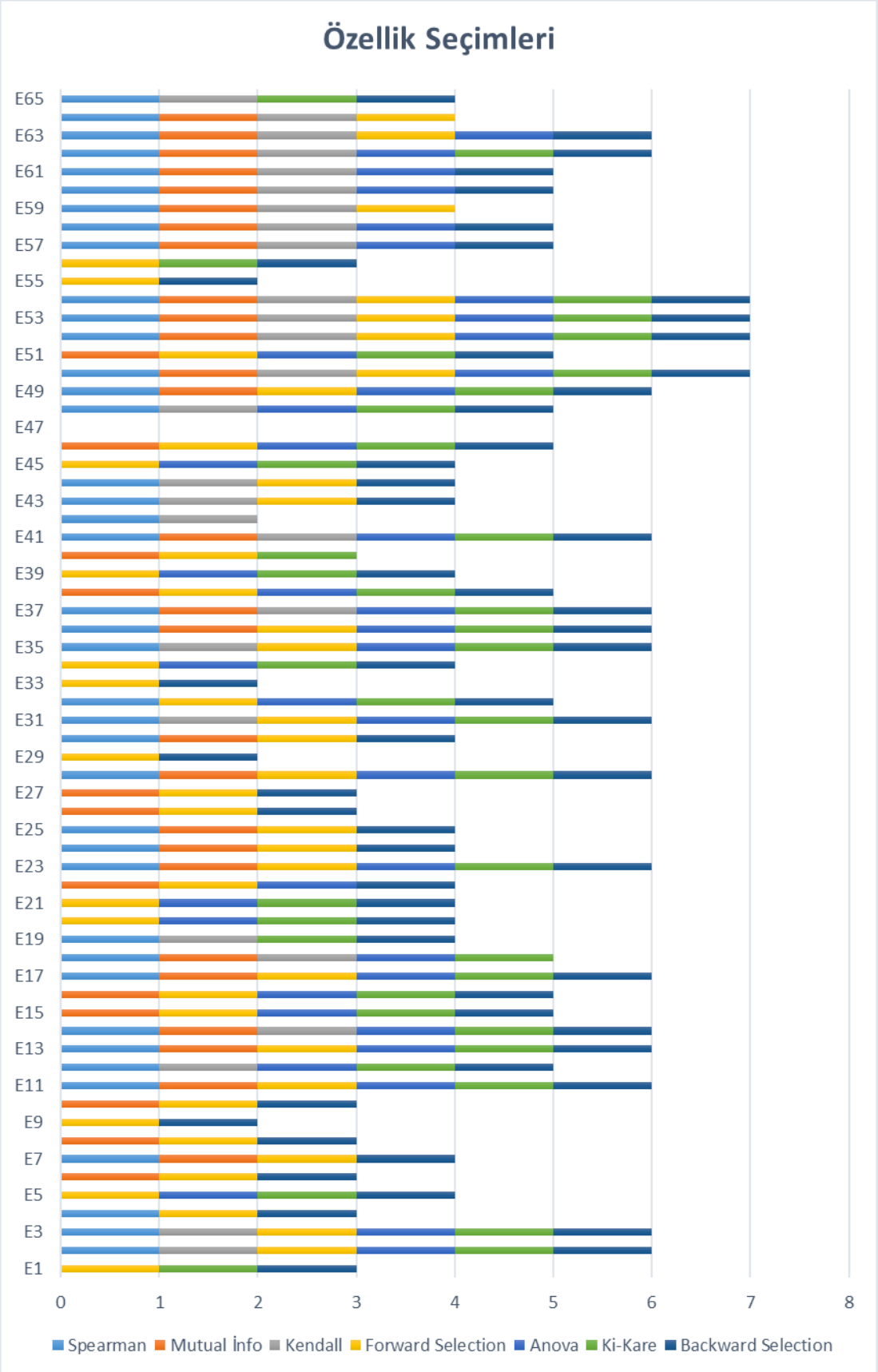
Tablo 3.5. Özellik seçim yöntemi sonrası öznitelik sayıları.

Özellik Seçim Yöntemi	Öznitelik Çıktısı
Spearman	40
Kendall	27
Anova	40
Ki-Kare	39
Mutual İnfo	39
Forward Selection	50
Backward Selection	59
PCA	30

3.3. Deneysel Çalışma

Bu adımda Şekil 3.4'deki adımlar izlenerek, veri ön işleme adımlarından sonra farklı sınıflandırma algoritmalarının özellik seçim yöntemlerindeki öznitelikler ile başarım oranları karşılaştırılmıştır.

Şekil 3.4. Tasarlanan eğitim modeli.



Şekil 3.5. Özellik seçim yöntemleri öznitelikleri.

3.3.1. Uygulama

Bu çalışma içerisindeki tüm işlemler kişisel bilgisayarım üzerinde gerçekleştirilmiştir. İşlemlerin yüklü RAM ve CPU tüketmesinden dolayı adım adım ilerlenmiştir. Kişisel bilgisayarım 16 GB RAM, i5-10500H CPU ve 4GB ekran kartına sahiptir. Python programlama ve kütüphaneleri ile çalışılmıştır. Verileri ön işlemek Pandas ve Numpy, verilerin görsel olarak anlamlandırılması için Seaborn ve Matplotlib, makine öğrenme algoritmalarında ise Sklearn ve derin öğrenme algoritmaları için ise Keras kütüphaneleri kullanılmıştır.

3.3.2. Makine öğrenmesi algoritmalari hiper parametreleri

Bu bölümde makine ve derin öğrenme algoritmaları parametrelerinde yapılan ayarlar gösterilmiştir. Bu parametreler eğitim sıraında kullanılan algoritmadaki ölçüm oranlarını etkilediği için her çalışmaya özel olarak ayarlanmalıdır.

3.3.2.1. Lojistik regresyon

Tablo 3.6. Lojistik resresyon kullanılan hiper parametreler.

Hiperparametreler	Değer	Özelliği
penalty	L2	Modele az katkısı olan değerleri sıfıra doğru yakınlaştırır
tol	0.0001	Optimizasyon için durma kriteri
C	1	Optimizasyon düzeltmeleri için
solver	lbfgs	Optimizasyonda kullanılacak algoritma
max_iter	100	Çözücülerin yakınsaması için maksimum yineleme sayısı

3.3.2.2. XGBoost

Tablo 3.7. XGBoost kullanılan hiper parametreler.

Hiperparametreler	Değer	Özelliği
n_estimators	1000	Tahmin edici ağaç sayısı
learning_rate	0.5	Yeni eklenen ağaçların öğrenme hızı ayarları
N_jobs	4	Pararlelde çalışan iş parçacığı sayısı
enable_categorical	false	Sonuçlar numerik olduğu için false ayarlandı

3.3.2.3. Rastgele orman

Tablo 3.8. Rastgele orman kullanılan hiper parametreler.

Hiperparametreler	Değer	Özelliği
n_estimators	100	Tahmin edici ağaç sayısı
random_state	1	Ağaç oluştururken örneklerin rastgeleliği kontrolü
Min_samples_leaf	1	Uç düğümde kalacak en düşük örnek adedi
Min_samples_split	2	Düğümün bölünmesi için örnek adedi
Oob_score	false	Genelleme puanı için dışarıdan veri kullanılmayacağı

3.3.2.4. Karar ağaçları

Tablo 3.9. Karar ağaçları kullanılan hiper parametreler.

Hiperparametreler	Değer	Özelliği
Criterion	mse	Ağaçlanmanın kalitesini ölçmek için
Splitter	best	Ağaçlanmayı seçmek için kullanılan strateji
Min_sample_leaf	1	Uç düğümde kalacak en düşük örnek sayısı
Min_samples_split	2	Dâhili düğümü bölmek için gereken örnek sayısı
random_state	0	Ağaç oluştururken örneklerin rastgeleliği kontrolü

3.3.2.5. K en yakın komşu

Tablo 3.10. K en yakın komşu kullanılan hiper parametreler.

Hiperparametreler	Değer	Özelliği
Leaf_size	30	Yaprak boyutu
metric	Minkowski	Mesafe hesaplaması için kullanılan metric
N_neighbours	3	Kullanılacak komşu sayısı
p	2	Minkowski algoritmasının güç parametresi
weights	uniform	Tahminde kullanılan ağırlık fonksiyonu

3.3.2.6. AdaBoost

Tablo 3.11. AdaBoost kullanılan hiper parametreler.

Hiperparametreler	Değer	Özelliği
Algorithm	SAMME.R	Güçlendirme algoritması
Learning_rate	1	Yükseltme yinelemesinde öğrenme ağırlık sayısı
N_estimators	50	Maksimum tahmin edicis sayısı

3.3.2.7. Tekrarlayan sinir ağları

Tablo 3.12. Tekrarlayan sinir ağları parametreler.

Layer	Değer	Özelliği
Embedding	40,40	Girdi vektör oluşturulması katmanı
Simple Rnn	40,40	RNN veri iletişim katmanı
Dropout	40,40	Aşırı öğrenmeden kaçınmak için veri azaltma
Simpe Rnn	40,32	RNN veri iletişim katmanı
Dropout	40,32	Aşırı öğrenmeden kaçınmak için veri azaltma
Simple Rnn	40,24	RNN veri iletişim katmanı
Dropout	40,24	Aşırı öğrenmeden kaçınmak için veri azaltma
Simple Rnn	40,16	RNN veri iletişim katmanı
Dropout	40,16	Aşırı öğrenmeden kaçınmak için veri azaltma
Simple Rnn	4	RNN veri iletişim katmanı
Dense	1	Derin öğrenme iletişim katmanı

3.3.2.8. Evrişimli sinir ağları

Tablo 3.13. Evrişimli sinir ağları parametreler.

Layer	Değer	Özelliği
Embedding	40,40	Girdi vektör oluşturulması katmanı
Conv1d	40,32	CNN özellik saptama katmanı
Dropout	40,32	Aşırı öğrenmeden kaçınmak için veri azaltma

Tablo 3.13.(Devamı) Evrişimli sinir ağları parametreler.

Layer	Değer	Özelliği
Conv1d	40,16	CNN özellik saptama katmanı
Dropout	40,16	Aşırı öğrenmeden kaçınmak için veri azaltma
Conv1d	40,4	CNN özellik saptama katmanı
Maxpooling	4	Ağdaki uyumsuzluk için max değerler alınması
Dense	1	Derin öğrenme iletişim katmanı

3.3.2.9. Uzun kısa vadeli bellek

Tablo 3.14. Uzun kısa vadeli bellek ağları parametreler.

Layer	Değer	Özelliği
Embedding	40,40	Girdi vektör oluşturulması katmanı
Lstm	40,32	LSTM iletişim katmanı
Dropout	40,32	Aşırı öğrenmeden kaçınmak için veri azaltma
Lstm	40,16	LSTM iletişim katmanı
Dropout	40,16	Aşırı öğrenmeden kaçınmak için veri azaltma
Lstm	4	LSTM iletişim katmanı
Dense	1	Derin öğrenme iletişim katmanı

3.3.3. Uygulamalar ve Analiz Sonuçları

Bu adımda özellik seçim ve azaltma yöntemleri ile belirlenen özniteliklerin, makine öğrenmesi algoritmalarının başarımları ölçülmüştür. Başarım ölçütü olarak karmaşıklık matrisi, doğruluk(accuracy), kesinlik(precision), duyarlılık(recall), F1 skor ve eğri altında kalan alan (AUC) değerlerine bakılmıştır. InSDN veri setindeki veri dağılımı dengesizliği göz önüne alındığında F1 skor başarım kriteri olarak değerlendirilmiştir. Özellik seçim yöntemleri ile belirlenen öznitelikler Şekil 3.5'de verilmiştir. Veri setinden %25 test veri seti olarak ayrılmıştır.

3.3.3.1. Spearman öznitelikleri ile eğitim

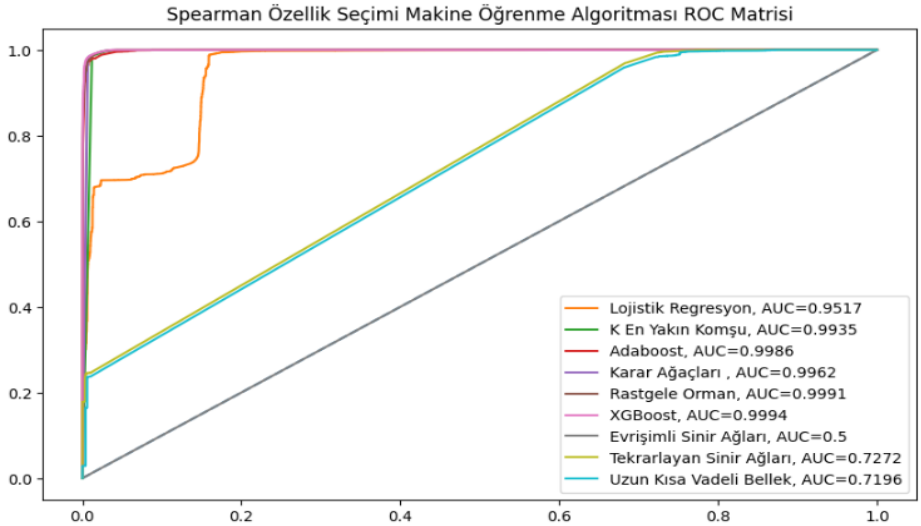
Bu aşamada spearman özellik seçim yöntemi sonucundaki 40 öznitelik ile başarımlar ölçülmüştür. 40 öznitelik Table 3.3'de verildiği gibi E2, E3, E4, E7, E11, E12, E13,

E14, E17, E18, E19, E23, E24, E25, E28, E30, E31, E32, E35, E36, E37, E41, E42, E43, E44, E48, E49, E50, E52, E53, E54, E57, E58, E59, E60, E61, E62, E63, E64, E65 değişkenlerini oluşturmaktadır.

Tablo 3.15. Spearman öznitelikleri ile başarım oranları.

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68439	297	3313	13924	0.9580	0.9538	0.9957	0.9743
Karar Ağaçları	68463	273	416	16821	0.9919	0.9940	0.9960	0.9950
K En Yakın Komşu	68457	279	493	16744	0.9910	0.9928	0.9959	0.9944
AdaBoost	68474	262	862	16375	0.9869	0.9876	0.9962	0.9919
XGBoost	68660	76	571	16666	0.9924	0.9918	0.9989	0.9953
Rastgele Orman	68686	50	621	16616	0.9921	0.9910	0.9993	0.9951
Evrişimli Sinir Ağları	68736	0	17237	0	0.8009	0.7995	1.0000	0.8886
Tekrarlayan Sinir Ağları	68595	141	12664	4573	0.8477	0.8442	0.9979	0.9146
Uzun Kısa Vadeli Bellek	68307	429	13311	3926	0.8422	0.8369	0.9938	0.9086

Tablo 3.15'de görüleceği üzere spearman öznitelikleri ile XGBoost 0.9924 doğruluk oranı ile en yüksek başarımı elde etmiştir. Rastgele orman algoritması 0.9921 doğruluk oranı ile ikinci yüksek başarılı algoritma olduğu görülmüştür.



Şekil 3.6. Spearman öznitelikleri ile roc matrisi.

Şekil 3.6'da görüleceği üzere ROC eğirisine bakıldığında (0,1) noktası en yakın 0.9994 AUC değeri ile XGBoost algoritmasının olduğu görülmektedir. F1 skor ve AUC değerleri bize spearman özellik seçim yönteminde XGBoost algoritmasının daha başarılı olduğunu göstermektedir.

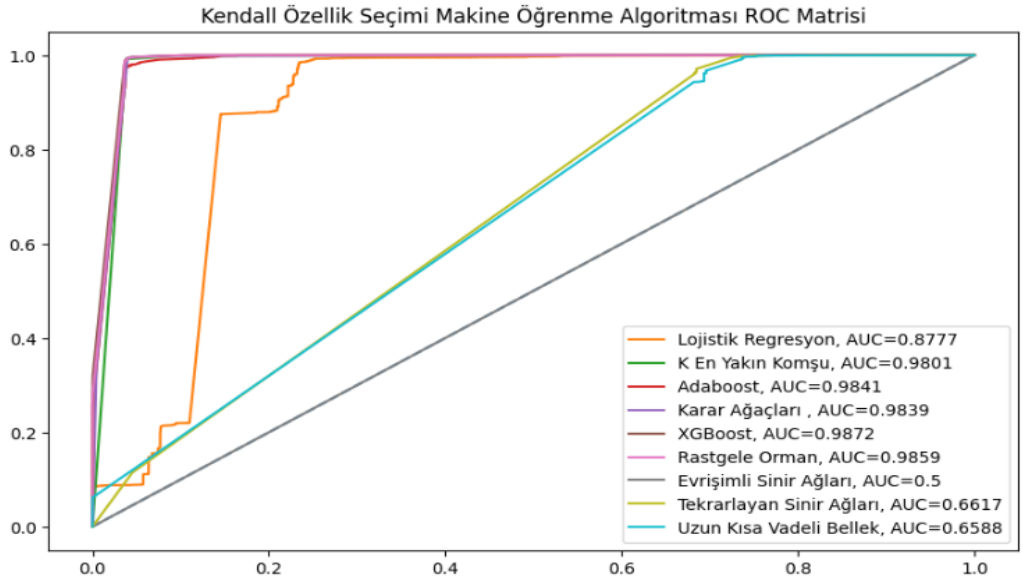
3.3.3.2. Kendall öznitelikleri ile eğitim

Bu aşamada kendall özellik seçim yöntemi sonucundaki 27 öznitelik ile başarımlar ölçülmüştür. 27 öznitelik Tablo 3.3'de verildiği gibi E2, E3, E12, E14, E18, E19, E31, E35, E37, E41, E42, E43, E44, E48, E50, E52, E53, E54, E57, E58, E59, E60, E61, E62, E63, E64, E65 değişkenlerini oluşturmaktadır.

Tablo 3.16. Kendall öznitelikleri ile başarım oranları.

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68229	507	4353	12884	0.9434	0.9400	0.9926	0.9656
Karar Ağaçları	68426	310	736	16501	0.9878	0.9894	0.9955	0.9924
K En Yakın Komşu	68512	224	1341	15896	0.9817	0.9808	0.9967	0.9887
AdaBoost	68285	451	1970	15267	0.9718	0.9720	0.9934	0.9826
XGBoost	68422	314	784	16453	0.9872	0.9887	0.9954	0.9920
Rastgele Orman	68399	337	771	16466	0.9871	0.9889	0.9951	0.9920
Evrişimli Sinir Ağları	68736	0	17237	0	0.8009	0.7995	1.0000	0.8886
Tekrarlayan Sinir Ağları	68554	182	12646	4591	0.8503	0.8443	0.9974	0.9144
Uzun Kısa Vadeli Bellek	68440	296	12740	4497	0.8446	0.8431	0.9957	0.9130

Şekil 3.16'da görüleceği üzere 0.9878 doğruluk ve 0.9924 F1 skor oranı ile Karar ağaçları en başarılı algoritma olduğu görülmektedir. Öznitelik sayısının az olması saldırı tespitinde FN değerlerin diğer özellik seçim yöntemlerine göre yüksek olduğu görülmüştür.



Şekil 3.7. Kendall öznitelikleri ile roc matrisi.

Şekilde 3.7'de eğitilen modelin ROC matrisi incelendiğinde Tablo 3.16'da en başarılı algoritma Karar ağaçları olarak görünse de Şekil 3.7'de görüleceği üzere 0.9872 AUC değeri ile XGBoost algoritması (0,1) noktası en yakın algoritma olmuştur. Karar ağaçları 0.9839 AUC değeri ile dördüncü en başarılı çıktıyı veren algoritma olduğu görülmüştür. F1 skor değeri ölçüm kriteri olarak alındığında Karar ağaçları kendall özellik seçim yönteminde daha başarılı algoritma olarak görülmektedir.

3.3.3.3. Anova öznitelikleri ile eğitim

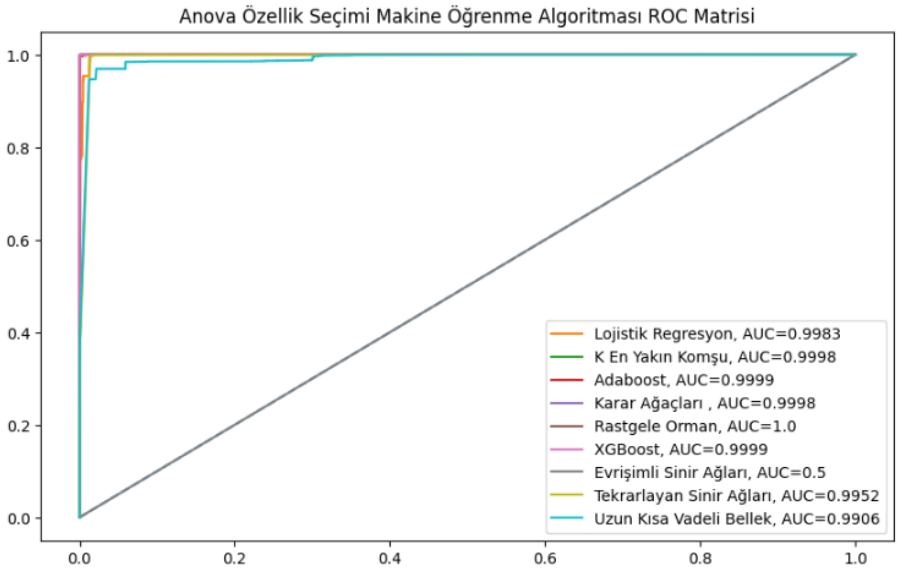
Bu aşamada anova özellik seçim yöntemi sonucundaki 40 öznitelik ile başarımlar ölçülmüştür. 40 öznitelik Tablo 3.3'de verildiği gibi E2, E3, E5, E11, E12, E13, E14, E15, E16, E17, E18, E20, E21, E22, E23, E28, E31, E32, E34, E35, E36, E37, E38, E39, E41, E45, E46, E48, E49, E50, E51, E52, E53, E54, E57, E58, E60, E61, E62, E63 değişkenlerini oluşturmaktadır.

Tablo 3.17. Anova öznitelikleri ile başarım oranları.

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68660	76	276	16961	0.9959	0.9960	0.9989	0.9974
Karar Ağaçları	68731	5	6	17231	0.9998	0.9999	0.9999	0.9999
K En Yakın Komşu	68723	13	11	17226	0.9997	0.9998	0.9998	0.9998
AdaBoost	68632	104	93	17144	0.9977	0.9986	0.9985	0.9986
XGBoost	68734	2	3	17234	0.9999	1.0000	1.0000	1.0000
Rastgele Orman	68734	2	8	17229	0.9998	0.9999	1.0000	0.9999
Evrişimli Sinir Ağları	68736	0	17237	0	0.8009	0.7995	1.0000	0.8886
Tekrarlayan Sinir Ağları	68533	203	251	16986	0.9935	0.9964	0.9970	0.9967
Uzun Kısa Vadeli Bellek	68635	101	5696	11541	0.8650	0.9234	0.9985	0.9595

Tablo 3.17'de görüleceği üzere XGBoost algoritması 0.9999 doğruluk ve 1.0 F1 skor değerleri ile diğer makine öğrenme algoritmalarından daha yüksek başarım elde ettiği görülmektedir. Derin öğrenme algoritmalarından Tekrarlayan sinir ağları ve Uzun kısa vadeli bellek, spearman ve kendall özellik seçim yöntemine göre anova özellik seçim yönteminde daha yüksek başarımlara ulaştığı görülmüştür.

Şekilde 3.8'de ROC matrisi incelendiğinde Rastgele orman algoritması 1 AUC değeri ile XGBoost algoritmasından daha yüksek olduğu görülmektedir. F1 skor ölçüm kriteri olarak değerlendirildiğinde XGBoost algoritmasının başarım oranının daha yüksek olduğu görülmüştür.



Şekil 3.8. Anova öznitelikleri ile roc matrisi.

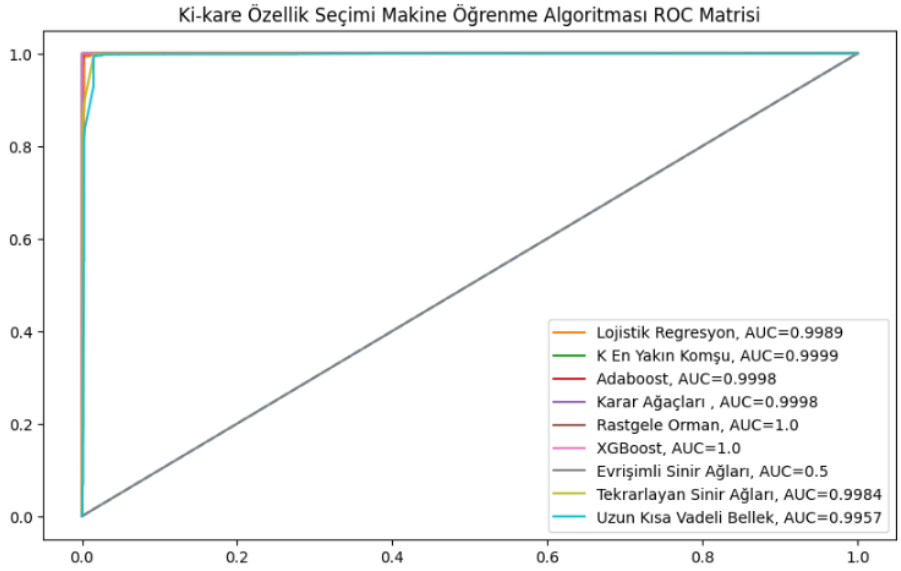
3.3.3.4. Ki-Kare öznitelikleri ile eğitim

Bu aşamada ki-kare özellik seçim yöntemi sonucundaki 39 öznitelik ile başarımlar ölçülmüştür. 39 öznitelik Tablo 3.3'de verildiği gibi E1, E2, E3, E5, E11, E12, E13, E14, E15, E16, E17, E18, E19, E20, E21, E23, E28, E31, E32, E34, E35, E36, E37, E38, E39, E40, E41, E45, E46, E48, E49, E50, E51, E52, E53, E54, E56, E62, E65 değişkenlerini oluşturmaktadır.

Tablo 3.18. Ki-kare öznitelikleri ile başarım oranları.

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68706	30	264	16973	0.9965	0.9962	0.9996	0.9979
Karar Ağaçları	68727	9	5	17232	0.9998	0.9999	0.9999	0.9999
K En Yakın Komşu	68730	6	8	17229	0.9998	0.9999	0.9999	0.9999
AdaBoost	68633	103	132	17105	0.9972	0.9981	0.9985	0.9983
XGBoost	68734	2	3	17234	0.9999	1.0000	1.0000	1.0000
Rastgele Orman	68734	2	2	17235	0.9999	1.0000	1.0000	1.0000
Evrişimli Sinir Ağları	68736	0	17237	0	0.8009	0.7995	1.0000	0.8886
Tekrarlayan Sinir Ağları	68497	239	259	16978	0.9935	0.9962	0.9965	0.9964
Uzun Kısa Vadeli Bellek	68407	329	314	16923	0.9802	0.9954	0.9952	0.9953

Tablo 3.18'de analiz sonuçlarından da görüleceği üzere XGBoost ve Rastgele orman algoritması 0.9999 doğruluk ve 1.0 F1 skor oranı ile benzer sonuçlar vermişlerdir.



Şekil 3.9. Ki-kare öznitelikleri ile roc matrisi.

Şekilde 3.9'da ROC matrisinde görüleceği üzere XGBoost ve Rastgele orman algoritmaları 1.0 AUC değeri ile en başarılı algoritma olduğu görülmüştür.F1 skor ve AUC değerleri ki-kare özellik seçim yöntemnde XGBoost ve Rastgele orman algoritmalarının benzer performans sergilediği görülmüştür.

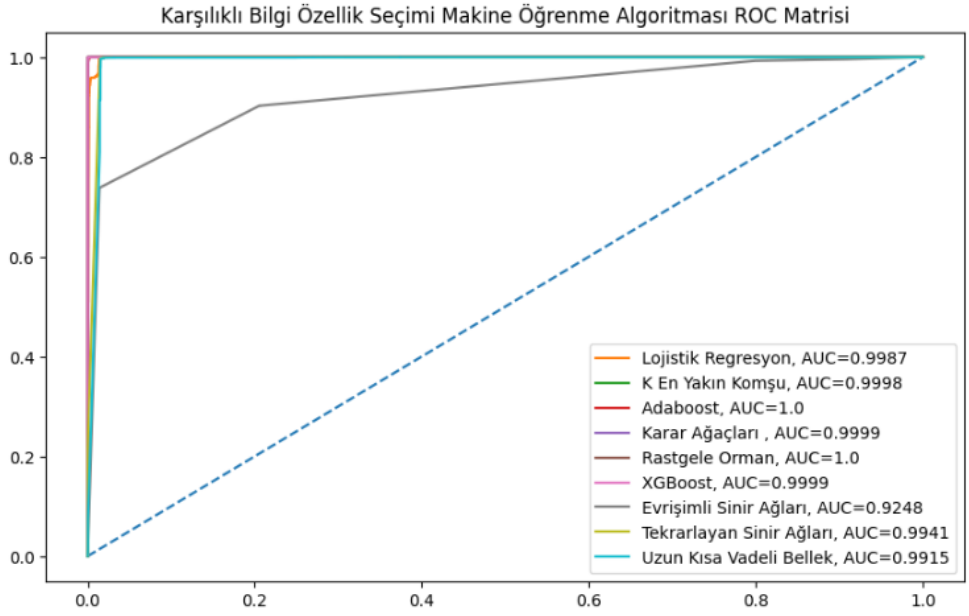
3.3.3.5. Karşılıklı bilgi öznitelikleri ile eğitim

Bu aşamada karşılıklı bilgi özellik seçim yöntemi sonucundaki 39 öznitelik ile başarımlar ölçülmüştür. 39 öznitelik Tablo 3.3'de verildiği gibi E6, E7, E8, E10, E11, E13, E14, E15, E16, E17, E18, E22, E23, E24, E25, E26, E27, E28, E30, E36, E37, E38, E40, E41, E46, E49, E50, E51, E52, E53, E54, E57, E58, E59, E60, E61, E62, E63, E64 değişkenlerini oluşturmaktadır.

Tablo 3.19. Karşılıklı bilgi öznitelikleri ile başarım oranları.

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68666	70	271	16966	0.9960	0.9961	0.9990	0.9975
Karar Ağaçları	68730	6	3	17234	0.9998	1.0000	0.9999	0.9999
K En Yakın Komşu	68729	7	12	17225	0.9997	0.9998	0.9999	0.9999
AdaBoost	68681	55	44	17193	0.9988	0.9994	0.9992	0.9993
XGBoost	68734	2	4	17233	0.9999	0.9999	1.0000	1.0000
Rastgele Orman	68734	2	13	17224	0.9998	0.9998	1.0000	0.9999
Evrişimli Sinir Ağları	62040	6696	3558	13679	0.8743	0.9458	0.9026	0.9237
Tekrarlayan Sinir Ağları	68643	93	321	16916	0.9913	0.9953	0.9986	0.9970
Uzun Kısa Vadeli Bellek	68520	216	281	16956	0.9927	0.9959	0.9969	0.9964

Tablo 3.19'da analiz bulguları incelendiğinde XGBoost algoritmasının 0.9999 doğruluk ve 1.0 F1 skor oranı ile en yüksek başarım oranı elde eden algoritma olduğu görülmektedir.



Şekil 3.10. Karşılıklı bilgi öznitelikleri ile roc matrisi.

Şekilde 3.10'da ROC matrisi incelendiğinde XGboost ve Adaboost algoritmaları (0,1) noktasına en yakın algoritma olduğu görülmüştür. Tablo 3.19'de F1 skor değerleri incelendiğinde XGBoost algoritmasının daha başarılı olduğu görülmüştür.

3.3.3.6. Ardışık ileri yönde öznitelikleri ile eğitim

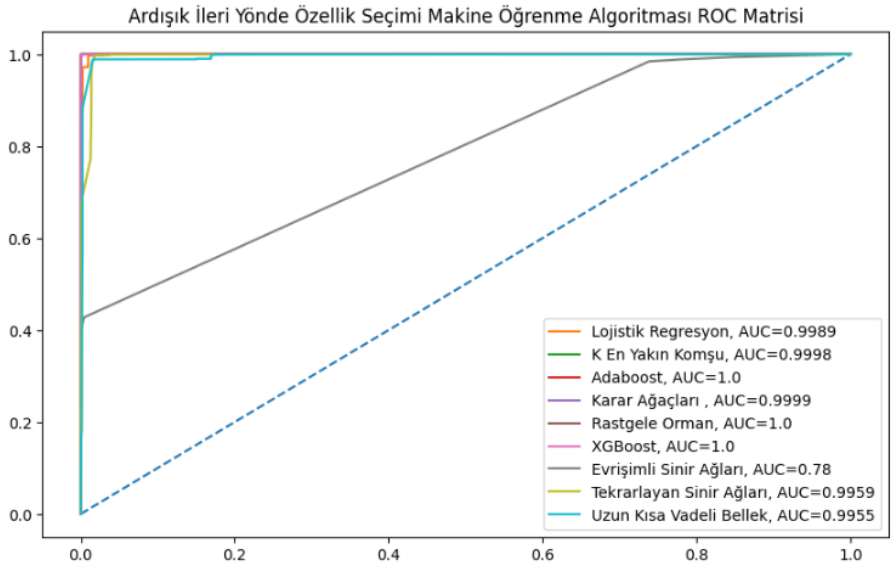
Bu aşamada ardışık ileri yönde özellik seçim yöntemi sonucundaki 50 öznitelik ile başarımlar ölçülmüştür. 50 öznitelik Tablo 3.3'de verildiği gibi E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E13, E15, E16, E17, E20, E21, E22, E23, E24, E25, E26, E27, E28, E29, E30, E31, E32, E33, E34, E35, E36, E38, E39, E40, E43, E44, E45, E46, E49, E50, E51, E52, E53, E54, E55, E56, E59, E63, E64 değişkenleridir.

Tablo 3.20'de görüleceği üzere Karar ağaçları, XGBoost ve Rastgele orman algoritmaları 0.9999 doğruluk oranıyla aynı başarım oranları elde etmişlerdir.

Şekilde 3.11'den ROC matrisi incelemeleri yapıldığında XGboost, Rastgele orman ve Adaboost algoritmalarının AUC değerlerinin birbirine eşit olduğu görülmüştür. Tablo 3.20'deki başarım oranları incelendiğinde F1 skor XGBoost algoritmasının daha başarılı olduğunu göstermektedir.

Tablo 3.20. Ardışık ileri yönde seçim öznitelikleri ile başarım oranları.

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68717	19	258	16979	0.9967	0.9963	0.9997	0.9980
Karar Ağaçları	68733	3	4	17233	0.9999	0.9999	1.0000	0.9999
K En Yakın Komşu	68727	9	9	17228	0.9997	0.9999	0.9999	0.9999
AdaBoost	68698	38	49	17188	0.9989	0.9993	0.9994	0.9994
XGBoost	68734	2	3	17234	0.9999	1.0000	1.0000	1.0000
Rastgele Orman	68734	2	5	17232	0.9999	0.9999	1.0000	0.9999
Evrişimli Sinir Ağları	67666	1070	12727	4510	0.8349	0.8417	0.9844	0.9075
Tekrarlayan Sinir Ağları	68558	178	455	16782	0.9932	0.9934	0.9974	0.9954
Uzun Kısa Vadeli Bellek	68057	679	2870	14367	0.9907	0.9595	0.9901	0.9746



Şekil 3.11. Ardışık ileri yönde seçim öznitelikleri ile roc matrisi.

3.3.3.7. Ardışık geri yönde öznitelikleri ile eğitim

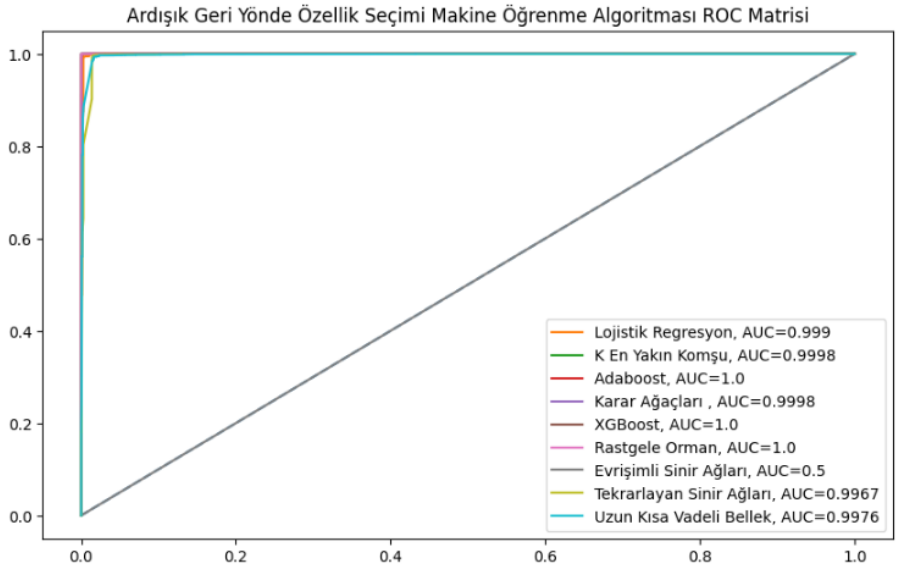
Bu aşamada ardışık geri yönde özellik seçim yöntemi sonucundaki 59 öznitelik ile başarımlar ölçülmüştür. 59 öznitelik Tablo 3.3'de verildiği gibi E1, E2, E3, E4, E5, E6, E7, E8, E9, E10, E11, E12, E13, E14, E15, E16, E17, E19, E20, E21, E22, E23, E24, E25, E26, E27, E28, E29, E30, E31, E32, E33, E34, E35, E36, E37, E38, E39, E41, E43, E44, E45, E46, E48, E49, E50, E51, E52, E53, E54, E55, E56, E57, E58, E60, E61, E62, E63, E65 değişkenlerini oluşturmaktadır.

Tablo 3.21'de görüleceği üzere XGBoost algoritmasının doğruluk ve F1 skor oranları diğer algoritmalardan yüksek olduğu görülmektedir. Karar ağaçları, K en yakın komşu

algoritması ve Rastgele orman algoritmaları başarım oranları da birbirine çok yakın olduğu görülmektedir.

Tablo 3.21. Ardışık geri yönde seçim öznitelikleri ile başarım oranları.

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68713	23	262	16975	0.9966	0.9962	0.9997	0.9979
Karar Ağaçları	68733	3	6	17231	0.9998	0.9999	1.0000	0.9999
K En Yakın Komşu	68728	8	9	17228	0.9998	0.9999	0.9999	0.9999
AdaBoost	68688	48	35	17202	0.9990	0.9995	0.9993	0.9994
XGBoost	68734	2	3	17234	0.9999	1.0000	1.0000	1.0000
Rastgele Orman	68734	2	7	17230	0.9998	0.9999	1.0000	0.9999
Evrişimli Sinir Ağları	68736	0	17237	0	0.8009	0.7995	1.0000	0.8886
Tekrarlayan Sinir Ağları	64028	4708	4716	12521	0.9925	0.9314	0.9315	0.9315
Uzun Kısa Vadeli Bellek	67922	814	443	16794	0.9688	0.9935	0.9882	0.9908



Şekil 3.12. Ardışık geri yönde seçim öznitelikleri ile roc matrisi.

Şekil 3.12'deki ROC matrisi değerleri inelendiğinde (0,1) noktasından geçen Adaboost, XGBoost ve Rastgele orman algoritmalarının başarımlarının yüksek olduğu görülmektedir fakat Tablo 3.21'deki başarım değerleri incelendiğide F1 skor ve AUC değerleri XGBoost algoritması başarımı daha yüksek olduğu göstermektedir.

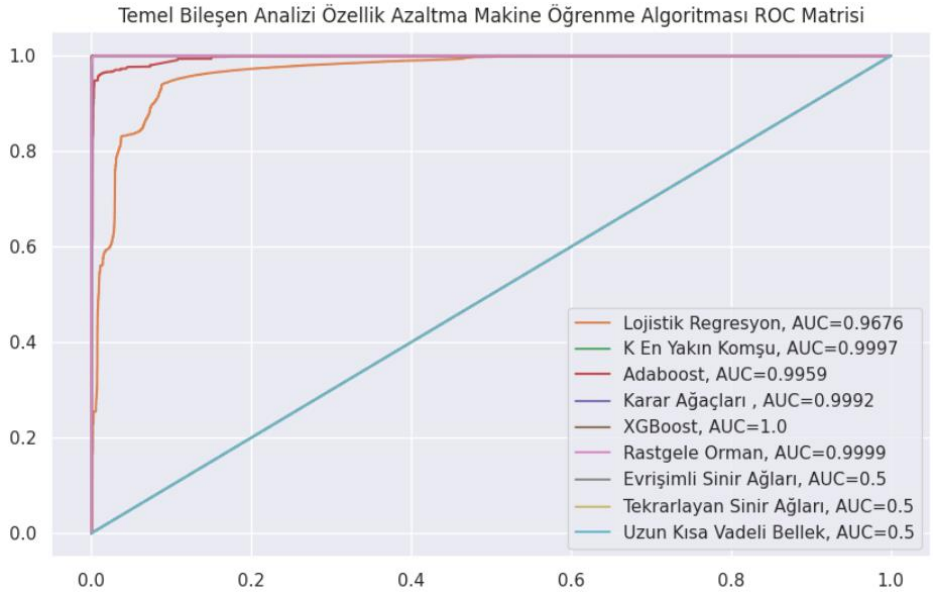
3.3.3.8. Temel bileşen analizi öznitelikleri ile eğitim

Bu aşamada temel bileşen analizi özellik indirgeme yöntemi sonucundaki 30 öznitelik ile başarımlar ölçülmüştür. Temel bileşen analizi tamamen yeni öznitelikler ortaya çıkarmaktadır.

Tablo 3.22. Temel bileşen analizi öznitelikleri ile başarım oranları

Algoritmalar	TP	FN	FP	TN	Doğruluk	Kesinlik	Duyarlılık	F1 Skor
Lojistik Regresyon	68682	98	8883	8310	0.8955	0.8855	0.9986	0.9386
Karar Ağaçları	68765	15	22	17171	0.9995	0.9997	0.9998	0.9997
K En Yakın Komşu	68770	10	34	17159	0.9994	0.9995	0.9999	0.9997
AdaBoost	67139	1641	774	16419	0.9719	0.9886	0.9761	0.9823
XGBoost	68766	14	14	17179	0.9996	0.9998	0.9998	0.9998
Rastgele Orman	68767	13	98	17095	0.9987	0.9986	0.9998	0.9992
Evrişimli Sinir Ağları	68780	0	17193	0	0.8005	0.8000	1.0000	0.8889
Tekrarlayan Sinir Ağları	68780	0	17193	0	0.8005	0.8000	1.0000	0.8889
Uzun Kısa Vadeli Bellek	68780	0	17193	0	0.8005	0.8000	1.0000	0.8889

Tablo 3.22'de analiz sonuçları incelendiğinde XGBoost algoritması 0.9996 doğruluk ve 0.9998 F1 Skor oranı ile diğer algoritmalardan daha başarılı performans sergilediği görülmektedir.



Şekil 3.13. Temel bileşen analizi öznitelikleri ile roc matrisi.

Şekil 3.13'deki ROC matrisi incelemelerine bakıldığında XGBoost algoritmasının (0,1) noktasından geçen ve en başarılı algoritma olduğu görülmektedir. F1 skor ve

AUC değeri bize XGBoost algoritmasının daha iyi performans sergilediği göstermektedir.

3.3.4. Araştırma bulgulari

Bölüm 3.3.3'de yapılan uygulamalarda farklı özellik seçim algoritmaları ile 9 farklı makine öğrenmesi algoritması çalışmaları ve başarım metrikleri verilmiştir.

Tablo 3.23. Algoritmaların özniteliklere göre doğruluk oranları.

Özellik Seçim Yöntemleri	Spearman	Kendall	Anova	Ki-Kare	Karşılıklı Bilgi	Ardışık İleri	Ardışık Geri	PCA
Öznitelik Sayısı	40	27	40	39	39	50	59	30
Lojistic Regresyon	0.9580	0.9434	0.9959	0.9965	0.9960	0.9967	0.9966	0.8955
Karar Ağaçları	0.9919	0.9878	0.9998	0.9998	0.9998	0.9999	0.9998	0.9995
K En Yakın Komşu	0.9910	0.9817	0.9997	0.9998	0.9997	0.9997	0.9998	0.9994
AdaBoost	0.9869	0.9718	0.9977	0.9972	0.9988	0.9989	0.9990	0.9719
XGBoost	0.9924	0.9872	0.9999	0.9999	0.9999	0.9999	0.9999	0.9996
Rastgele Orman	0.9921	0.9871	0.9998	0.9999	0.9998	0.9999	0.9998	0.9987
Evrişimli Sinir Ağları	0.8009	0.8009	0.8009	0.8009	0.8743	0.8349	0.8009	0.8005
Tekrarlayan Sinir Ağları	0.8477	0.8503	0.9935	0.9935	0.9913	0.9932	0.9925	0.8005
Uzun Kısa Vadeli Bellek	0.8422	0.8446	0.8650	0.9802	0.9927	0.9907	0.9688	0.8005

Yapılan farklı algoritma ve özellik seçim yöntemlerinin doğruluk oranları tek tablo halinde Tablo 3.23'de gösterilmiştir. Doğruluk oranları ile deneysel sonuçlarda incelendiğinde Spearman, Anova, Karşılıklı bilgi, ardışık geri yönde özellik seçim yöntemleri ve temel bileşen analizi özellik azaltma yöntemlerinde en başarılı algoritmanın XGBoost olduğu görülmüştür. Kendall özellik seçim yönteminde Karar ağaçları daha başarılı sonuçlar ortaya koymuştur. Ki-kare ve ardışık ileri yönde özellik seçim algoritmalarında ise Rastgele orman ve XGBoost algoritmalarının başarım oranları birbirine çok yakın olduğu görülmüştür.

Tablo 3.24'de ise deneysel çalışma sonucunda özellik seçim yöntemleri ile algoritmaların F1 skor değerleri verilmiştir. Veri seti içerisindeki dengesiz saldırı ve normal veri dağılımı göz önüne alındığında başarım metriği olarak F1 skor daha doğru

sonuç vereceği düşünülmüştür. F1 skorları ile algoritma başarım oranları karşılaştırıldığında Spearman, Anova, Karşılıklı bilgi, Ardışık ileri yönde, Ardışık geri yönde ve Temel bileşen analizi özellik seçimi ve azaltma yöntemlerinde XGBoost en başarılı algoritma olduğu görülmüştür. Kendall özelli seçim yönteminde ise Karar ağaçları daha başarılı sonuç verdiği görülmüştür. Ki-kare özelik seçim yönteminde ise XGboost v Rastgele orman algoritması aynı başarım oranlarına sahiptirler.

Tablo 3.24. Algoritmaların özniteliklere göre f1 skor oranları.

Özellik Seçim Yöntemleri	Spearman	Kendall	Anova	Ki-Kare	Karşılıklı Bilgi	Ardışık İleri	Ardışık Geri	PCA
Öznitelik Sayısı	40	27	40	39	39	50	59	30
Lojistic Regresyon	0.9743	0.9656	0.9974	0.9979	0.9975	0.9980	0.9979	0.9386
Karar Ağaçları	0.9950	0.9924	0.9999	0.9999	0.9999	0.9999	0.9999	0.9997
K En Yakın Komşu	0.9944	0.9887	0.9998	0.9999	0.9999	0.9999	0.9999	0.9997
AdaBoost	0.9919	0.9826	0.9986	0.9983	0.9993	0.9994	0.9994	0.9823
XGBoost	0.9953	0.9920	1.0000	1.0000	1.0000	1.0000	1.0000	0.9998
Rastgele Orman	0.9951	0.9920	0.9999	1.0000	0.9999	0.9999	0.9999	0.9992
Evrişimli Sinir Ağları	0.8886	0.8886	0.8886	0.8886	0.9237	0.9075	0.8886	0.8889
Tekrarlayan Sinir Ağları	0.9146	0.9144	0.9967	0.9964	0.9970	0.9954	0.9315	0.8889
Uzun Kısa Vadeli Bellek	0.9086	0.9130	0.9595	0.9953	0.9964	0.9746	0.9908	0.8889

Çalışma içerisindeki metrikler incelendiğinde genel olarak derin öğrenme algoritmalarının başarım oranlarının daha düşük olduğu görülmüştür. Öznitelik sayısı arttığında derin öğrenme algoritmalarının başarım oranlarının yükseldiği görülmüştür.

Makine öğrenme algoritmalarından XGBoost, Karar ağaçları ve Rastgele orman algoritması başarım oranları diğer algoritmalara göre daha yüksek olduğu görülmüştür.

4. SONUÇ VE ÖNERİLER

Bu çalışma içerisinde YTA için makine ve derin öğrenme destekli STS geliştirmek üzerinde çalışılmıştır. Model oluştururken YTA için oluşturulmuş InSDN veri seti kullanılmıştır. InSDN veri seti YTA için tasarlanmış yeni bir veri setidir. Veri seti ile ilgili bilgilere Bölüm 3.1'de bahsedilmiştir. InSDN veri setini belirli işlemlerden geçirdikten sonra başarım oranları çıktıları Bölüm 3.3.3'de uygulama çıktılarında gösterilmiştir.

Deneysel çalışmalar sonucunda XGBoost algoritmasının Kendall özellik seçim yöntemi dışında kullanılan Spearman, Anova, Ki-kare, karşılıklı bilgi, ardışık ileri yönde, ardışık geri yönde özellik seçim yöntemi ve Temel bileşen analizi özellik azaltma yöntemlerinde en başarılı algoritma olduğu görülmüştür.

Qizhao ve arkadaşları yaptıkları çalışmada InSDN veri setindeki 56 öznitelikten 16 özniteliği kullanarak bir çalışma gerçekleştirmişlerdir. Seçtikleri 16 öznitelik ile farklı makine öğrenmesi algoritmaları başarımlarını ölçmüşlerdir. XGBoost algoritması ile 16 öznitelik ile 0.9783 oranında bir başarım elde etmişlerdir (Zhou et al., 2021).

Mahmoud ve arkadaşları yaptıkları çalışmada özniteliklerin belirlenmesinde CNN kullanmışlardır. Veri seti içerisindeki özniteliklerin her saldırı için kabul edilebilir olmadıklarını ileri sürmüşlerdir. Yaptıkları bu deneysel çalışma sonucunda 48 öznitelik ile CNN ve LSTM hibrit şekilde bir model oluşturmuşlardır. Oluşturulan model sonucunda oluşturulan hibrit modelin kullanılması için öneri sunmuşlardır. Bu deneysel çalışmalarında 0.9632 oranında bir başarım oranı elde etmişlerdir (Abdallah et al., 2021).

Kung ve arkadaşları InSDN veri seti kullanarak 2021 yılında yaptıkları çalışma da CIC-Flowmeter kullanmışlar ve veri seti içerisindeki 48 özelliği kullanmışlardır. Veri normalleştirme aşamasında mix-max normalizasyonu kullanmışlardır. Tasarladıkları derin öğrenme modelini önerdikleri bu çalışma da çok frekanslı derin öğrenme modeli ile en yüksek başarım oranını DDoS saldırıları için 99.92 başarım oranı elde ettikleri görünmektedir (Wang et al., 2021).

Abdullah ve arkadaşları yaptıkları çalışmada ise veri seti içerisinden 48 ve 6 öznitelik ile deneysel çalışmalarını gerçekleştirmişlerdir. Çalışma içerisinde saldırı tespiti için RNN, LSTM ve GRU algoritmalarını önermişlerdir. Yaptıkları çalışma da 48 öznitelikle 0.9884, 6 öznitelikle ise 0.9257 oranında LSTM ile yüksek başarım elde etmişlerdir (Alshra'A et al., 2021).

Literatürdeki bir başka çalışmada Mahmoud Said ve arkadaşları tarafından 77 öznitelik ile deneysel çalışma gerçekleştirmişlerdir. LSTM ve OC-SVM modellerini önerdikleri çalışmalarında en yüksek 0.9050 oranında bir başarım elde etmişlerdir (Said Elsayed et al., 2020).

Bu çalışma da ise 8 farklı özellik seçim yöntemi ve dokuz farklı makine öğrenmesi algoritması kullanılmıştır. XGBoost algoritması genel olarak diğer tüm makine öğrenme algoritmalarından başarılı sonuçlar elde ettiği görülmüştür.

Zamandan ve kaynak tüketimi açısından değerlendirildiğinde en az özellik ve en yüksek başarım oranı olarak Kendall öznitelik seçimi algoritması sonucunda çıkan 27 özellik ile Karar ağaçları 0.9878 ve XGBoost 0.9872 başarım oranları elde edilmiştir.

Tablo 3.23'de elde ettiğimiz tüm çıktılar gösteriyor ki, özellik seçim yönteminde çıkan öznitelik sayıları arttıkça derin öğrenme algoritma yöntemlerinin başarım oranı da artmaktadır. Daha az öznitelik ile bir eğitim modelinde ise makine öğrenmesi algoritmalarının başarım oranlarının daha yüksek olduğu elde edilmiştir.

Tüm makine öğrenmesi algoritmaları öznitelik seçim algoritmaları ile eğitildiğinde Tablo 3.23 ve Tablo 2.24'deki başarım metriklerinden görüleceği üzere bu çalışma da en başarılı algoritma XGBoost algoritması oldu söylenebilmektedir. Çalışma ile ilgili veri seti ve kodlara https://github.com/birolemekli/SDN-IDS web adresindeki github repositorysinden erişebilirsiniz.

Çalışma içerisinde kullanılan algoritmaların hiper parametreleri bölüm 3.3.2'de verildiği şekilde ayarlanmıştır. Buradaki hiper parametrelerin değiştirilerek modellerin başarım oranları gelecek çalışmalarda araştırılabilir.

Literatüre yeni girmiş olan InSDN veri setinin daha geliştirilmesi için çalışmalar yapılabilir. Daha çok veri ile yapılan modellemelerde gelebilecek saldırıları tespit etmek daha mümkün olacaktır. Bu yüzden InSDN veri setine katkıda bulunmak daha sonraki çalışmalarda değerlendirilebilir.

InSDN veri seti 7 farklı saldırı türünü barındırmaktadır fakat bu deneysel çalışma içerisinde ikili sınıflandırma üzerinde çalışılmıştır. Bu ikili sınıflandırma da saldırı olup olmadığı tespit edilirken, gelecek çalışmalarda saldırı türünün ne olduğu hakkında çalışmalar yapılabilir.

Bu tez çalışması YTA için oluşturulmuş olan yeni bir veri seti üzerindeki başarımların tespit edilmesi üzerine çalışılmıştır. Bilimsel çalışmalara katkısı, gelecek diğer çalışmalara bir yol gösterici olacağı, YTA ağlara yönelik yenilikçi ve akıllı bir system tasarımı inşa etmek, saldırı tespit sistemlerine farklı bir bakış açısı katacağına inanıyoruz.

KAYNAKLAR

Abdallah, M., An Le Khac, N., Jahromi, H., & Delia Jurcut, A. (2021). A Hybrid CNN-LSTM Based Approach for Anomaly Detection Systems in SDNs. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3465481.3469190

Abdi, H. (2008). Kendall Rank Correlation Coefficient. *The Concise Encyclopedia of Statistics*, 278–281. https://doi.org/10.1007/978-0-387-32833-1_211

Agarwal, A. K., Wadhwa, S., & Chandra, S. (1994). Diagnosis of tuberculosis--newer tests. *The Journal of the Association of Physicians of India*, *42*(8), 665.

Alshra'A, A. S., Farhat, A., & Seitz, J. (2021). Deep Learning Algorithms for Detecting Denial of Service Attacks in Software-Defined Networks. *Procedia Computer Science*, *191*(2019), 254–263. https://doi.org/10.1016/j.procs.2021.07.032

AYDIN, M. A. (2016). *Bilgisayar Ağlarında Saldırı Tespiti için İstatistiksel Yöntem Kullanılması*. *15*(2), 1–23.

BUDAK, H. (2018). Özellik Seçim Yöntemleri ve Yeni Bir Yaklaşım. *Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, *22*(Özel), 21. https://doi.org/10.19113/sdufbed.01653

Budak, İ., Şen, B., & Yıldırım, M. Z. (2013). Lojistik Regresyon ile Bilgisayar Ağlarında Anomali Tespiti. *Akademik Bilişim*, 1–10.

Çekiç, İ., & Çavdar, K. (2023). Detection of the cracks in metal sheets using convolutional neural network (CNN). *Journal of the Faculty of Engineering and Architecture of Gazi University*, *38*(1), 153–162. https://doi.org/10.17341/gazimmfd.873479

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28. https://doi.org/10.38094/jastt20165

Cui, Z. , R. KE, Z. P. (n.d.). *Stacked Bidirectional and Unidirectional LSTM RNN for spped prediction (1)*.

Cui, L., Yu, F. R., & Yan, Q. (2016). When big data meets software-defined networking: SDN for big data and big data for SDN. *IEEE Network*, *30*(1), 58–65. https://doi.org/10.1109/MNET.2016.7389832

Ding, H., Liang, Z. Y., Guo, F. B., Huang, J., Chen, W., & Lin, H. (2016). Predicting bacteriophage proteins located in host cell with feature selection technique. *Computers in Biology and Medicine*, *71*, 156–161. https://doi.org/10.1016/j.compbiomed.2016.02.012

Elmas, B. (2022). Classification varieties of marble and granite by convolutional neural networks with transfer learning method. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *37*(2), 985–1001. https://doi.org/10.17341/gazimmfd.936835

Elsayed, M. S., Le-Khac, N. A., & Jurcut, A. D. (2020). InSDN: A novel SDN intrusion dataset. *IEEE Access*, *8*, 165263–165284. https://doi.org/10.1109/ACCESS.2020.3022633

Forman, G. (2000). 10.1162/153244303322753670. *CrossRef Listing of Deleted DOIs*, *1*, 1289–1305. https://doi.org/10.1162/153244303322753670

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Jin, Z., Shang, J., Zhu, Q., Ling, C., Xie, W., & Qiang, B. (2020). RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12343 LNCS*, 503–515. https://doi.org/10.1007/978-3-030-62008-0_35

Kavzoğlu, T., Şahİn, E. K., & Çölkesen, İ. (n.d.). *Heyelan DuyarliliAnalizindeKi-KareTestine Dayali FaktöSeçimi*. *Uzalcbs 2014*, 14–17.

LaValley, M. P. (2008). Logistic regression. *Circulation*, *117*(18), 2395–2399. https://doi.org/10.1161/CIRCULATIONAHA.106.682658

Lee, C. S., Cheang, P. Y. S., & Moslehpour, M. (2022). Predictive Analytics in Business Analytics: Decision Tree. *Advances in Decision Sciences*, *26*(1), 1–29. https://doi.org/10.47654/V26Y2022I1P1-30

Levitt, K. N. (1994). The Role of Customer-Premises Bandwidth Management: In the evolving era of high-speed wide-area networking, customer premises bandwidth management should offer economic advantages well into the future. *IEEE Network*, *8*(3), 26–41. https://doi.org/10.1109/65.283931

Lewis, P. M. (1963). R63-92 On the Effectiveness of Receptors in Recognition Systems. *IEEE Transactions on Electronic Computers*, *EC-12*(5), 582–582. https://doi.org/10.1109/PGEC.1963.263678

Malhi, A., & Gao, R. X. (2004). PCA-based feature selection scheme for machine defect classification. *IEEE Transactions on Instrumentation and Measurement*, *53*(6), 1517–1525. https://doi.org/10.1109/TIM.2004.834070

Oral, L., & Ozkan, K. (2013). *Suboptimal optimization method for dominant point detection*. 1–4. https://doi.org/10.1109/siu.2013.6531295

Pathan, A. K. (2014). The State of the Art in Intrusion Prevention and Detection. In *The State of the Art in Intrusion Prevention and Detection*. https://doi.org/10.1201/b16390

Said Elsayed, M., Le-Khac, N. A., Dev, S., & Jurcut, A. D. (2020). Network Anomaly Detection Using LSTM Based Autoencoder. *Q2SWinet 2020 - Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, 37–45. https://doi.org/10.1145/3416013.3426457

Schultz, E. E. (2000). Intrusion Detection. In *Network Security* (Vol. 2000, Issue 8, p. 19). https://doi.org/10.1016/s1353-4858(00)08021-1

Selmic, R. R., Phoha, V. V., & Serwadda, A. (2016). WSN Architecture. *Wireless Sensor Networks*, *1*, 37–81. https://doi.org/10.1007/978-3-319-46769-6_3

Singh, D., & Singh, B. (2020). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*, 105524. https://doi.org/10.1016/j.asoc.2019.105524

Staudemeyer, R. C., & Morris, E. R. (2019). *Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks*. 1–42. http://arxiv.org/abs/1909.09586

Tanuj Joshi: Tiwari, R. (2019). Efficient System for Cancer Detection Using Feature Selection Algorithms. *Ijsrcsams*, *8*(5), 1–5.

Van Hulse, J., Khoshgoftaar, T. M., Napolitano, A., & Wald, R. (2009). Feature selection with high-dimensional imbalanced data. *ICDM Workshops 2009 - IEEE International Conference on Data Mining*, 507–514. https://doi.org/10.1109/ICDMW.2009.35

Wang, K., Fu, Y., Duan, X., Liu, T., & Xu, J. (2021). *Abnormal traffic detection system in SDN based on deep learning hybrid models*.

Whitney, A. W. (1971). A Direct Method of Nonparametric Measurement Selection. *IEEE Transactions on Computers*, *C–20*(9), 1100–1103. https://doi.org/10.1109/T-C.1971.223410

Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2015). A Survey on Software-Defined Networking. *IEEE Communications Surveys and Tutorials*, *17*(1), 27–51. https://doi.org/10.1109/COMST.2014.2330903

Yilmaz, A. (2021). Diagnosing COVID-19 from X-Ray images with using multi-channel CNN architecture. *Journal of the Faculty of Engineering and Architecture of Gazi University*, *36*(4), 1761–1773. https://doi.org/10.17341/gazimmfd.746883

ZAVRAK, S., & İSKEFİYELİ, M. (2016). Yazilim-Tanimli Ağlar Ve SaldiriTespit Ve ÖnlemSistemleriÜzerineBirİnceleme a Review of Software-Defined Networking. *Academia.Edu*, *December 2017*. https://www.academia.edu/download/55079726/Pages_from_Pages_from_isc-BildiriKitabi2016.pdf

Zhang, S., & Zhao, Z. (2008). Feature selection filtering methods for emotion recognition in chinese speech signal. *International Conference on Signal Processing Proceedings, ICSP*, 1699–1702. https://doi.org/10.1109/ICOSP.2008.4697464

Zhou, Q., Yu, J., & Li, D. (2021). A dynamic and lightweight framework to secure source addresses in the SDN-based networks. *Computer Networks*, *193*(July 2020), 108075. https://doi.org/10.1016/j.comnet.2021.108075

EKLER

EK A: Veri setindeki öznitelikler.

Öznitelik İsmi	Açıklama
ACK Flag Cnt	ACK edilmiş paket sayısı
Active Max	Bir işlemin boşta kalmadan önceki etkin olduğu maksimum süre
Active Mean	Bir işlemin boşta kalmadan önceki aktif olduğu ortalama süre
Active Min	Bir işlemin boşta kalmadan önceki etkin olduğu minimum süre
Active Std	Bir işlemin boşta kalmadan önceki etkin olduğu standart sapma süresi
Bwd Header Len	Geri gönderilen başlıklar için kullanılan toplam bayt sayısı
Bwd IAT Max	Geri yönde gönderilen iki paket arasında maksimum süre
Bwd IAT Mean	Geri yönde gödnerilen iki paket arasındaki ortalama süre
Bwd IAT Min	Geri yönde gönderilen iki paket arasında minimum süre
Bwd IAT Std	Geri yönde gönderilen iki paket arasında standart sapma
Bwd IAT Tot	Geri yönde gönderilen iki paket arasında toplam süre
Bwd PSH Flags	Geri yönde gönderilen paketlerdeki PHS bayrağı ayarlanma sayısı
Bwd Pkt Len Max	Geri yönde maksimum paket boyutu
Bwd Pkt Len Mean	Geri yönde ortalama paket boyutu
Bwd Pkt Len Min	Geri yönde minimum paket boyutu
Bwd Pkt Len Std	Geri yönde standrt sapma paket boyutu
Bwd Pkts/s	Saniyedeki geri gönderilen paket sayısı
Bwd Seg Size Avg	Geri yönde gözlemlenen ortalama boyut
Bwd URG Flags	Geri yönde hareket eden paketlerde URG bayrağının ayarlanma sayısı
Down/Up Ratio	İndirme ve yükleme oranı
FIN Flag Cnt	FIN'li paket sayısı
Flow Byts/s	Saniyedeki akış bayt sayısı
Flow Duration	Akışın mikrosaniye cinsinden süresi
Flow IAT Max	Gönderilen iki paket arasındaki maksimum süre
Flow IAT Mean	Gönderilen iki paket arasındaki ortalama süre
Flow IAT Min	Gönderilen iki paket arasındaki maksimum süre
Flow IAT Std	Gönderilen iki paket arasındaki standart sapma süre
Flow Pkts/s	Saniyedeki akış paketi sayısı
Fwd Act Data Pkts	İleri yönde en az 1 baytlık TCP veri paket sayısı

EK A: (Devamı) Veri setindeki öznitelikler.

Öznitelik İsmi	Açıklama
Fwd Header Len	İleri yöndeki başlıklar için kullanulan toplam bayt sayısı
Fwd IAT Max	İleri yönde gönderilen iki paket arasındaki maksimum süre
Fwd IAT Mean	İleri yönde gönderilen iki paket arasındaki ortalama süre
Fwd IAT Min	İleri yönde gönderilen iki paket arasındaki minimum süre
Fwd IAT Std	İleri yönde gönderilen iki paket arasındaki startdart sapma süre
Fwd IAT Tot	İleri yönde gönderilen iki paket arasındaki toplam süre
Fwd Pkt Len Max	İleri yönde paketin maksimum boyutu
Fwd Pkt Len Mean	İleri yönde paketin ortalama boyutu
Fwd Pkt Len Min	İleri yönde paketin minimum boyutu
Fwd Pkt Len Std	İleri yönde paketin standart sapma boyutu
Fwd Pkts/s	Saniyedeki ileri gönderilen pket sayısı
Fwd Seg Size Avg	İleri yönde gözlemlenen ortalama boyut
Idle Max	Akışın aktif olmadan önce boşta kaldığı maksimum süre
Idle Mean	Akışın aktif olmadan önce boşta kaldığı ortalama süre
Idle Min	Akışın aktif olmadan önce boşta kaldığı minimum süre
Idle Std	Akışın aktif olmadan önce boşta kaldığı standart sapma süre
Init Bwd Win Byts	Başlangıç penceresinde geri yönde gönderilen toplam bayt sayısı
Init Fwd Win Byts	Başlangıç penceresinde ileri yönde gönderilen toplam bayt sayısı
PSH Flag Cnt	Push'lu paket sayısı
Pkt Len Max	Bir paketin maksimum uzunluğu
Pkt Len Mean	Bir paketin ortamala uzunluğu
Pkt Len Min	Bir paketin minimum uzunluğu
Pkt Len Std	Bir paketin standart sapma uzunluğu
Pkt Len Var	Bir paketin varyans uzunluğu
Pkt Size Avg	Ortalama Paket Sayısı
RST Flag Cnt	RST'li paket sayısı
SYN Flag Cnt	SYN'li paket sayısı
Subflow Bwd Byts	Bir alt akışta geri yöndeki ortalama bayt sayısı
Subflow Bwd Pkts	Bir alt akışta geri yöndeki ortalama paket sayısı
Subflow Fwd Byts	Bir alt akışta ileri yöndeki ortalama bayt sayısı
Subflow Fwd Pkts	Bir alt akışta ileri yöndeki ortalama paket sayısı
Tot Bwd Pkts	Geri yöndeki toplam paketler
Tot Fwd Pkts	İleri yöndeki toplam paketler

EK A: (Devamı) Veri setindeki öznitelikler.

Öznitelik İsmi	Açıklama
TotLen Bwd Pkts	Geri yöndeki paketlerin toplam boyutu
TotLen Fwd Pkts	İleri yöndeki paketlerin toplam boyutu
URG Flag Cnt	RUG'li paket sayısı
Fwd PSH Flags	İleri yönde hareket eden paketlerde PHS bayrağının ayarlanma sayısı
Fwd URG Flags	İleri yönde hareket eden paketlerde URG bayrağının ayarlanma sayısı
CWE Flag Count	CWE'li paket sayısı
ECE Flag Cnt	ECE'li paket sayısı
Fwd Byts/b Avg	İleri yöndeki ortalama bayt sayısı toplu hızı
Fwd Pkts/b Avg	İleri yöndeki ortalama paket sayısı toplu hızı
Fwd Blk Rate Avg	İleri yönde ortalama toplu oran sayısı
Bwd Byts/b Avg	Geri yöndeki ortalama bayt sayısı toplu oranı
Bwd Pkts/b Avg	Geri yönde ortalama paket sayısı toplu oranı
Bwd Blk Rate Avg	Geri yöndeki ortalama toplu oran sayısı
Fwd Seg Size Min	İleri yönde gözlenen minimum segment boyutu
Flow ID	Akış kimliği bilgisi
Src IP	Kaynak IP adresi
Dst IP	Hedef IP adresi
Timestamp	Akış zaman damgası
Src Port	Kaynağın TCP UDP bağlantı noktası
Dst Port	Hedefin TCP UDP bağlantı noktası
Protocol	Akışla ilgili protokol
Label	Saldırı etiketi

ÖZGEÇMİŞ

Ad-Soyad : Birol EMEKLİ

ÖĞRENİM DURUMU:

- **Önlisans** **:** 2012, Aksaray Üniversitesi, Bilgisayar Programcılığı
- **Lisans** **:** 2016, Anadolu Üniversitesi, İşletme
- **Lisans** **:** 2019, Sakarya Üniversitesi, Bilgisayar Müühendisliği

MESLEKİ DENEYİMLER:

- Ağustos 2013 – Mart 2021 yılları arasında Sakarya Üniversitesi Bilgi İşlem Daire Başkanlığında Bilgisayar İşletmeni olarak çalıştı.
- Mart 2021 – Aralık 2022 yılları arasında Havelsan Deniz Komuta Merkezinde DevOps Mühendisi olarak çalıştı
- Aralık 2022'den beri Payten Firmasında Linux ve Altyapı Takım Lideri olarak görev yapmaktadır.

Yayın Listesi:

- Gsm Operatörlerine Yönelik Atilan Türkçe Tweetlerin Derin Öğrenme Yöntemleriyle Duygu Analizi *4. Uluslararası Marmara Fen Bilimleri Kongresi, Sayfa 3*